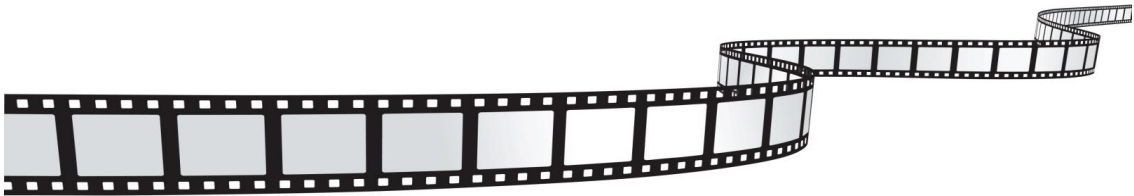


18.08.10



Contents

1	Introduction	66
2	Module Index	67
2.1	Modules	67
3	Data Structure Index	72
3.1	Data Structures	72
4	File Index	79
4.1	File List	79
5	Module Documentation	96
5.1	Median	96
5.1.1	Detailed Description	96
5.2	Lens Shading Correction	97
5.2.1	Detailed Description	97
5.3	Raw	98
5.3.1	Detailed Description	98
5.4	Debayer	99
5.4.1	Detailed Description	99
5.5	Sharpen	100
5.5.1	Detailed Description	100
5.6	Luma Denoise	101
5.6.1	Detailed Description	101
5.7	Chroma Denoise	102
5.7.1	Detailed Description	102
5.8	Look-up table	103
5.8.1	Detailed Description	103
5.9	Color Combination	104
5.9.1	Detailed Description	104

5.10	Convolution	105
5.10.1	Detailed Description	105
5.11	Harris Corner Detector	106
5.11.1	Detailed Description	106
5.12	Polyphase FIR Scaler	107
5.12.1	Detailed Description	107
5.12.2	Enumeration Type Documentation	108
5.13	Edge operator	109
5.13.1	Detailed Description	109
5.14	Sigma Denoise	110
5.14.1	Detailed Description	110
5.15	Chroma Generation	111
5.15.1	Detailed Description	111
5.16	DoG LTM	112
5.16.1	Detailed Description	112
5.17	MIPI Rx	113
5.17.1	Detailed Description	113
5.18	MIPI Tx	114
5.18.1	Detailed Description	114
5.19	Absolute difference	115
5.19.1	Detailed Description	115
5.19.2	Function Documentation	115
5.20	Accumulate Square	116
5.20.1	Detailed Description	116
5.20.2	Function Documentation	116
5.21	Accumulate Weighted	117
5.21.1	Detailed Description	117
5.21.2	Function Documentation	117
5.22	Arithmetic addition	118
5.22.1	Detailed Description	118
5.22.2	Function Documentation	118
5.23	Arithmetic addition with mask	119
5.23.1	Detailed Description	119
5.23.2	Function Documentation	119
5.24	Arithmetic subtraction	120
5.25	Arithmetic	121

5.25.1 Detailed Description	126
5.25.2 Function Documentation	126
5.26 Arithmetic subtraction fp16	127
5.26.1 Detailed Description	127
5.26.2 Function Documentation	127
5.27 Arithmetic subtraction with mask	128
5.27.1 Detailed Description	128
5.27.2 Function Documentation	128
5.28 Average	129
5.28.1 Detailed Description	129
5.28.2 Function Documentation	129
5.29 bilateral5x5	130
5.29.1 Detailed Description	130
5.29.2 Function Documentation	130
5.30 Bitwise And	131
5.30.1 Detailed Description	131
5.30.2 Function Documentation	131
5.31 Bitwise And with mask	132
5.31.1 Detailed Description	132
5.31.2 Function Documentation	132
5.32 Bitwise Not	133
5.32.1 Detailed Description	133
5.32.2 Function Documentation	133
5.33 Bitwise Or	134
5.33.1 Detailed Description	134
5.33.2 Function Documentation	134
5.34 Bitwise Or with mask	135
5.34.1 Detailed Description	135
5.34.2 Function Documentation	135
5.35 Bitwise Xor	136
5.35.1 Detailed Description	136
5.35.2 Function Documentation	136
5.36 Bitwise Xor with mask	137
5.36.1 Detailed Description	137
5.36.2 Function Documentation	137
5.37 Generic Box Filter	138

5.37.1	Detailed Description	138
5.37.2	Enumeration Type Documentation	138
5.37.3	Function Documentation	139
5.38	Box Filter 11x11	140
5.38.1	Detailed Description	140
5.38.2	Function Documentation	140
5.39	Box Filter 13x13	141
5.39.1	Detailed Description	141
5.39.2	Function Documentation	141
5.40	Box Filter 15x15	142
5.40.1	Detailed Description	142
5.40.2	Function Documentation	142
5.41	Box Filter 3x3	143
5.41.1	Detailed Description	143
5.41.2	Function Documentation	143
5.42	Box Filter 5x5	144
5.42.1	Detailed Description	144
5.42.2	Function Documentation	144
5.43	Box Filter 7x7	145
5.43.1	Detailed Description	145
5.43.2	Function Documentation	145
5.44	Box Filter 9x9	146
5.44.1	Detailed Description	146
5.44.2	Function Documentation	146
5.45	Canny Edge Detection	147
5.45.1	Detailed Description	147
5.45.2	Function Documentation	147
5.46	censusMatching16	148
5.46.1	Detailed Description	148
5.46.2	Function Documentation	148
5.47	censusMatching32	149
5.47.1	Detailed Description	149
5.47.2	Function Documentation	149
5.48	censusMatching64	150
5.48.1	Detailed Description	150
5.48.2	Function Documentation	150

5.49	censusMatching65	151
5.49.1	Detailed Description	151
5.49.2	Function Documentation	151
5.50	censusMatchingPyr	152
5.50.1	Detailed Description	152
5.50.2	Function Documentation	152
5.51	censusMin16	153
5.51.1	Detailed Description	153
5.51.2	Function Documentation	153
5.52	censusMin64	154
5.52.1	Detailed Description	154
5.52.2	Function Documentation	154
5.53	censusMin65	155
5.53.1	Detailed Description	155
5.53.2	Function Documentation	155
5.54	censusMin7	156
5.54.1	Detailed Description	156
5.54.2	Function Documentation	156
5.55	CensusTransform5x5	157
5.55.1	Detailed Description	157
5.55.2	Function Documentation	157
5.56	channelExtract	158
5.56.1	Detailed Description	158
5.56.2	Function Documentation	158
5.57	Chroma Block	159
5.57.1	Detailed Description	159
5.57.2	Function Documentation	159
5.58	Contrast	160
5.58.1	Detailed Description	160
5.58.2	Function Documentation	160
5.59	Convolution 11x11	161
5.59.1	Detailed Description	161
5.59.2	Function Documentation	161
5.60	Convolution 15x1	162
5.60.1	Detailed Description	162
5.60.2	Function Documentation	162

5.61	Convolution 1x15	163
5.61.1	Detailed Description	163
5.61.2	Function Documentation	163
5.62	Convolution 1x5	164
5.62.1	Detailed Description	164
5.62.2	Function Documentation	164
5.63	Convolution 1x5 Fp16ToFp16	165
5.63.1	Detailed Description	165
5.63.2	Function Documentation	165
5.64	Convolution 1x7	166
5.64.1	Detailed Description	166
5.64.2	Function Documentation	166
5.65	Convolution 1x7 Fp16ToFp16	167
5.65.1	Detailed Description	167
5.65.2	Function Documentation	167
5.66	Convolution 1x9	168
5.66.1	Detailed Description	168
5.66.2	Function Documentation	168
5.67	Convolution 3x3	169
5.67.1	Detailed Description	169
5.67.2	Function Documentation	169
5.68	Convolution 3x3 Fp16ToFp16	170
5.68.1	Detailed Description	170
5.68.2	Function Documentation	170
5.69	Convolution 5x1	171
5.69.1	Detailed Description	171
5.69.2	Function Documentation	171
5.70	Convolution 5x1 Fp16ToFp16	172
5.70.1	Detailed Description	172
5.70.2	Function Documentation	172
5.71	Convolution 5x5	173
5.71.1	Detailed Description	173
5.71.2	Function Documentation	173
5.72	Convolution 5x5 Fp16ToFp16	174
5.72.1	Detailed Description	174
5.72.2	Function Documentation	174

5.73	Convolution 7x1	175
5.73.1	Detailed Description	175
5.73.2	Function Documentation	175
5.74	Convolution 7x1 Fp16ToFp16	176
5.74.1	Detailed Description	176
5.74.2	Function Documentation	176
5.75	Convolution 7x7	177
5.75.1	Detailed Description	177
5.75.2	Function Documentation	177
5.76	Convolution 7x7 Fp16ToFp16	178
5.76.1	Detailed Description	178
5.76.2	Function Documentation	178
5.77	Convolution 7x7 Fp16ToU8	179
5.77.1	Detailed Description	179
5.77.2	Function Documentation	179
5.78	Convolution 9x1	180
5.78.1	Detailed Description	180
5.78.2	Function Documentation	180
5.79	Convolution 9x9	181
5.79.1	Detailed Description	181
5.79.2	Function Documentation	181
5.80	Convolution 9x9 Fp16ToFp16	182
5.80.1	Detailed Description	182
5.80.2	Function Documentation	182
5.81	Convert 16bpp To 8bpp	183
5.81.1	Detailed Description	183
5.81.2	Function Documentation	183
5.82	Convert F16 To U8	184
5.82.1	Detailed Description	184
5.82.2	Function Documentation	184
5.83	12Bpp to 8Bpp conversion	185
5.83.1	Detailed Description	185
5.83.2	Function Documentation	185
5.84	Convert Fp16 to U16	186
5.84.1	Detailed Description	186
5.84.2	Function Documentation	186

5.85	Convert U16 to Fp16	187
5.85.1	Detailed Description	187
5.85.2	Function Documentation	187
5.86	Convert U8 To F16	188
5.86.1	Detailed Description	188
5.86.2	Function Documentation	188
5.87	YUV400 to YUV422 conversion	189
5.87.1	Detailed Description	189
5.87.2	Function Documentation	189
5.88	Generic Convolution	190
5.88.1	Detailed Description	190
5.88.2	Function Documentation	190
5.89	Convolution Separable 11x11	191
5.89.1	Detailed Description	191
5.89.2	Function Documentation	191
5.90	Convolution Separable 11x11 Fp16ToFp16	192
5.90.1	Detailed Description	192
5.90.2	Function Documentation	192
5.91	Convolution Separable 3x3	193
5.91.1	Detailed Description	193
5.91.2	Function Documentation	193
5.92	Convolution Separable 3x3 Fp16ToFp16	194
5.92.1	Detailed Description	194
5.92.2	Function Documentation	194
5.93	Convolution Separable 5x5	195
5.93.1	Detailed Description	195
5.93.2	Function Documentation	195
5.94	Convolution Separable 5x5 Fp16ToFp16	196
5.94.1	Detailed Description	196
5.94.2	Function Documentation	196
5.95	Convolution Separable 7x7	197
5.95.1	Detailed Description	197
5.95.2	Function Documentation	197
5.96	Convolution Separable 7x7 Fp16ToFp16	198
5.96.1	Detailed Description	198
5.96.2	Function Documentation	198

5.97 Convolution Separable 9x9	199
5.97.1 Detailed Description	199
5.97.2 Function Documentation	199
5.98 Convolution Separable 9x9 Fp16ToFp16	200
5.98.1 Detailed Description	200
5.98.2 Function Documentation	200
5.99 Convert to YUV444	201
5.99.1 Detailed Description	201
5.99.2 Function Documentation	201
5.100Copy	202
5.100.1 Detailed Description	202
5.100.2 Function Documentation	202
5.101Corner Min Eigenvalue	203
5.101.1 Detailed Description	203
5.101.2 Function Documentation	203
5.102Corner Min Eigenvalue Patched	204
5.102.1 Detailed Description	204
5.102.2 Function Documentation	204
5.103Crop	205
5.104CV	206
5.104.1 Detailed Description	208
5.104.2 Function Documentation	208
5.105YUV to NV12 chroma conversion	209
5.106NV21 to RGB conversion	210
5.106.1 Detailed Description	210
5.106.2 Function Documentation	210
5.107RGB(fp16) to Luma(u8) conversion	211
5.107.1 Detailed Description	211
5.107.2 Function Documentation	211
5.108RGB(fp16) to UV420(u8) conversion	212
5.108.1 Detailed Description	212
5.108.2 Function Documentation	212
5.109RGB to Chroma NV12 conversion	213
5.109.1 Detailed Description	213
5.109.2 Function Documentation	213
5.110RGB to Luma conversion	214

5.111ISP	215
5.111.1 Detailed Description	217
5.111.2 Function Documentation	218
5.112RGB to Luma NV12 conversion	219
5.112.1 Detailed Description	219
5.112.2 Function Documentation	219
5.113RGB to UV conversion	220
5.113.1 Detailed Description	220
5.113.2 Function Documentation	220
5.114RGB to UV420 conversion	221
5.114.1 Detailed Description	221
5.114.2 Function Documentation	221
5.115RGB to YUV422 conversion	222
5.115.1 Detailed Description	222
5.115.2 Function Documentation	222
5.116YUV422 to RGB conversion	223
5.116.1 Detailed Description	223
5.116.2 Function Documentation	223
5.117YUV to RGB conversion	224
5.117.1 Detailed Description	224
5.117.2 Function Documentation	224
5.118Dilate 3x3	225
5.118.1 Detailed Description	225
5.118.2 Function Documentation	225
5.119Dilate 5x5	226
5.119.1 Detailed Description	226
5.119.2 Function Documentation	226
5.120Dilate 7x7	227
5.120.1 Detailed Description	227
5.120.2 Function Documentation	227
5.121Generic Dilate	228
5.121.1 Detailed Description	228
5.121.2 Function Documentation	228
5.122Equalize Histogram	229
5.122.1 Detailed Description	229
5.122.2 Function Documentation	229

5.123Erode 3x3	230
5.123.1 Detailed Description	230
5.123.2 Function Documentation	230
5.124Erode 5x5	231
5.124.1 Detailed Description	231
5.124.2 Function Documentation	231
5.125Erode 7x7	232
5.125.1 Detailed Description	232
5.125.2 Function Documentation	232
5.126AF Stats	233
5.126.1 Detailed Description	233
5.126.2 Function Documentation	233
5.127Fast9M2	234
5.127.1 Detailed Description	234
5.127.2 Function Documentation	234
5.128Fast9ScoreCv	235
5.128.1 Detailed Description	235
5.128.2 Function Documentation	235
5.129Gauss Blur	236
5.129.1 Detailed Description	236
5.129.2 Function Documentation	236
5.130GaussHx2	237
5.130.1 Detailed Description	237
5.130.2 Function Documentation	237
5.131GaussHx2_fp16	238
5.131.1 Detailed Description	238
5.131.2 Function Documentation	238
5.132GaussVx2	239
5.132.1 Detailed Description	239
5.132.2 Function Documentation	239
5.133GaussVx2_fp16	240
5.133.1 Detailed Description	240
5.133.2 Function Documentation	240
5.134Generate Chroma	241
5.134.1 Detailed Description	241
5.134.2 Function Documentation	241

5.135Generate Chroma with subsampling	242
5.135.1 Detailed Description	242
5.135.2 Function Documentation	242
5.136Generate Reference for Luma Denoise	243
5.136.1 Detailed Description	243
5.136.2 Function Documentation	243
5.137Generate Reference for Luma Denoise(fp16 input)	244
5.137.1 Detailed Description	244
5.137.2 Function Documentation	244
5.138Luma Blur	245
5.138.1 Detailed Description	245
5.138.2 Function Documentation	245
5.139Generate Luma U8 to Fp16	246
5.139.1 Detailed Description	246
5.139.2 Function Documentation	246
5.140greyDesat	247
5.140.1 Detailed Description	247
5.140.2 Function Documentation	247
5.141hammingDistance	248
5.141.1 Detailed Description	248
5.141.2 Function Documentation	248
5.142harrisResponse	249
5.142.1 Detailed Description	249
5.142.2 Function Documentation	249
5.143Histogram	250
5.143.1 Detailed Description	250
5.143.2 Function Documentation	250
5.144histogramStat	251
5.144.1 Detailed Description	251
5.144.2 Function Documentation	251
5.145Homography	252
5.145.1 Detailed Description	252
5.145.2 Function Documentation	252
5.146Integral Image Square Sum(f32)	253
5.146.1 Detailed Description	253
5.146.2 Function Documentation	253

5.147Integral Image Square Sum(U32)	254
5.147.1 Detailed Description	254
5.147.2 Function Documentation	254
5.148Integral Image Sum(f32)	255
5.148.1 Detailed Description	255
5.148.2 Function Documentation	255
5.149Integral Image Sum(U16toU32)	256
5.149.1 Detailed Description	256
5.149.2 Function Documentation	256
5.150Integral Image Sum(U32)	257
5.150.1 Detailed Description	257
5.150.2 Function Documentation	257
5.151interpolatePixelBilinear	258
5.151.1 Detailed Description	258
5.151.2 Function Documentation	258
5.152Laplacian 3x3	259
5.152.1 Detailed Description	259
5.152.2 Function Documentation	259
5.153Laplacian 5x5	260
5.153.1 Detailed Description	260
5.153.2 Function Documentation	260
5.154Laplacian 5x5 Fp16 To Fp16	261
5.154.1 Detailed Description	261
5.154.2 Function Documentation	261
5.155Laplacian 7x7	262
5.155.1 Detailed Description	262
5.155.2 Function Documentation	262
5.156Laplacian 7x7 Fp16 To Fp16	263
5.156.1 Detailed Description	263
5.156.2 Function Documentation	263
5.157localMaxMin3x3_fp16	264
5.157.1 Detailed Description	264
5.157.2 Function Documentation	264
5.158localTM	265
5.158.1 Detailed Description	265
5.158.2 Function Documentation	265

5.159	Low Level Correction	266
5.159.1	Detailed Description	266
5.159.2	Function Documentation	266
5.159.3	Variable Documentation	267
5.160	Low Level Correction on Multiple Planes	268
5.160.1	Detailed Description	268
5.160.2	Function Documentation	268
5.161	Luma Blur	269
5.161.1	Detailed Description	269
5.161.2	Function Documentation	269
5.162	LUT 10 to 16	270
5.162.1	Detailed Description	270
5.162.2	Function Documentation	270
5.163	LUT 10 to 8	271
5.163.1	Detailed Description	271
5.163.2	Function Documentation	271
5.164	LUT 12 to 16	272
5.164.1	Detailed Description	272
5.164.2	Function Documentation	272
5.165	LUT 12 to 8	273
5.165.1	Detailed Description	273
5.165.2	Function Documentation	273
5.166	LUT 8 to 8	274
5.166.1	Detailed Description	274
5.166.2	Function Documentation	274
5.167	LUT 16 to 8	275
5.167.1	Detailed Description	275
5.167.2	Function Documentation	275
5.168	maxTest3x3_fp16	276
5.168.1	Detailed Description	276
5.168.2	Function Documentation	276
5.169	meanStdDev	277
5.169.1	Detailed Description	277
5.169.2	Function Documentation	277
5.170	Min/Max Value Position	278
5.170.1	Detailed Description	278

5.170.2 Function Documentation	278
5.171 Min/Max Value	279
5.171.1 Detailed Description	279
5.171.2 Function Documentation	279
5.172 minTest3x3_fp16	280
5.172.1 Detailed Description	280
5.172.2 Function Documentation	280
5.173 Mix Median	281
5.173.1 Detailed Description	281
5.173.2 Function Documentation	281
5.174 MonoImbalance	282
5.175 Negative	283
5.175.1 Detailed Description	283
5.175.2 Function Documentation	283
5.176 nonMaxFp32	284
5.176.1 Detailed Description	284
5.176.2 Function Documentation	284
5.177 nonMax3x3U8	285
5.177.1 Detailed Description	285
5.177.2 Function Documentation	285
5.178 padBayer5Frame	286
5.178.1 Detailed Description	286
5.178.2 Function Documentation	286
5.179 Pixel packer	287
5.179.1 Detailed Description	287
5.179.2 Function Documentation	287
5.180 Pixel Unpacker	288
5.180.1 Detailed Description	288
5.180.2 Function Documentation	288
5.181 Pixel Unpacker Mipi 10b	289
5.181.1 Detailed Description	289
5.181.2 Function Documentation	289
5.182 Pixel Unpacker WB	290
5.182.1 Detailed Description	290
5.182.2 Function Documentation	290
5.183 Pixel Position	291

5.183.1 Detailed Description	291
5.183.2 Function Documentation	291
5.184purpleFlare	292
5.184.1 Detailed Description	292
5.184.2 Function Documentation	292
5.185Pyramid Downscale	293
5.185.1 Detailed Description	293
5.185.2 Function Documentation	293
5.186Random Noise	294
5.186.1 Detailed Description	294
5.186.2 Function Documentation	294
5.187Random Noise (high speed)	295
5.187.1 Detailed Description	295
5.187.2 Function Documentation	295
5.188Sum of Absolute Differences 11x11	296
5.188.1 Detailed Description	296
5.188.2 Function Documentation	296
5.189Sum of Absolute Differences 5x5	297
5.189.1 Detailed Description	297
5.189.2 Function Documentation	297
5.190Downscale by 2	298
5.190.1 Detailed Description	298
5.190.2 Function Documentation	298
5.191Downscale by 2 (fp16/u8)	299
5.191.1 Detailed Description	299
5.191.2 Function Documentation	299
5.192Downscale by 2 (fp16/fp16)	300
5.192.1 Detailed Description	300
5.192.2 Function Documentation	300
5.193Lanczos Downscale by 2 (6 taps)	301
5.193.1 Detailed Description	301
5.193.2 Function Documentation	301
5.194Lanczos Downscale by 2 (7 taps)	302
5.194.1 Detailed Description	302
5.194.2 Function Documentation	302
5.195Upscale by 2	303

5.195.1 Detailed Description	303
5.195.2 Function Documentation	303
5.196Upscale by 2 with phases 0.25 and 0.75 fp16 to fp16	304
5.196.1 Detailed Description	304
5.196.2 Function Documentation	304
5.197Upscale by 2 with phases 0.25 and 0.75 u16 to u16	305
5.197.1 Detailed Description	305
5.197.2 Function Documentation	305
5.198Upscale by 2 with phases 0.25 and 0.75 fp16 to u8	306
5.198.1 Detailed Description	306
5.198.2 Function Documentation	306
5.199Upscale by 2 with phases 0.25 and 0.75 u8 to u8	307
5.199.1 Detailed Description	307
5.199.2 Function Documentation	307
5.200Lanczos Horizontal Upscale by 2	308
5.200.1 Detailed Description	308
5.200.2 Function Documentation	308
5.201Lanczos Upscale by 2	309
5.201.1 Detailed Description	309
5.201.2 Function Documentation	309
5.202Lanczos Vertical Upscale by 2	310
5.202.1 Detailed Description	310
5.202.2 Function Documentation	310
5.203Arbitrary Downscale	311
5.203.1 Detailed Description	311
5.203.2 Function Documentation	311
5.204scharr_fp16	312
5.204.1 Detailed Description	312
5.204.2 Function Documentation	312
5.205Laplacian 3x3 Fp16 To Fp16	313
5.205.1 Detailed Description	313
5.205.2 Function Documentation	313
5.206Sobel	314
5.206.1 Detailed Description	314
5.206.2 Function Documentation	314
5.207Sum of Squared Differences 11x11	315

5.207.1 Detailed Description	315
5.207.2 Function Documentation	315
5.208 Sum of Squared Differences 5x5	316
5.208.1 Detailed Description	316
5.208.2 Function Documentation	316
5.209 Sum of Squared Differences 7x7 (U8 to U32)	317
5.209.1 Detailed Description	317
5.209.2 Function Documentation	317
5.210 Sum of Squared Differences 7x7	318
5.210.1 Detailed Description	318
5.210.2 Function Documentation	318
5.211 Threshold	319
5.211.1 Detailed Description	319
5.211.2 Enumeration Type Documentation	319
5.211.3 Function Documentation	320
5.212 Threshold Binary Range	321
5.212.1 Detailed Description	321
5.212.2 Function Documentation	321
5.213 Threshold Binary U8	322
5.213.1 Detailed Description	322
5.213.2 Function Documentation	322
5.214 ThresholdFilter	323
5.214.1 Detailed Description	323
5.214.2 Function Documentation	323
5.215 Undistort	324
5.215.1 Detailed Description	324
5.215.2 Function Documentation	324
5.216 White Balance Bayer GBRG	325
5.216.1 Detailed Description	325
5.216.2 Function Documentation	325
5.217 White Balance RGB	326
5.217.1 Detailed Description	326
5.217.2 Function Documentation	326
5.218 XY Generator	327
5.218.1 Detailed Description	327
5.218.2 Function Documentation	327

5.219DMA	328
5.219.1 Detailed Description	328
6 Data Structure Documentation	329
6.1 AccumulateWeightedParam Struct Reference	329
6.1.1 Detailed Description	329
6.1.2 Field Documentation	329
6.2 ae_patch_stats Struct Reference	329
6.2.1 Field Documentation	330
6.3 AeAwbStatsCfg Struct Reference	330
6.3.1 Field Documentation	330
6.4 AF_paxel_statistics Struct Reference	331
6.4.1 Field Documentation	331
6.5 Bilateral5x5Param Struct Reference	331
6.5.1 Field Documentation	331
6.6 BoxFilter11x11Param Struct Reference	332
6.6.1 Detailed Description	332
6.6.2 Field Documentation	332
6.7 BoxFilter13x13Param Struct Reference	332
6.7.1 Detailed Description	332
6.7.2 Field Documentation	332
6.8 BoxFilter15x15Param Struct Reference	333
6.8.1 Detailed Description	333
6.8.2 Field Documentation	333
6.9 BoxFilter3x3Param Struct Reference	333
6.9.1 Detailed Description	333
6.9.2 Field Documentation	333
6.10 BoxFilter5x5Param Struct Reference	334
6.10.1 Detailed Description	334
6.10.2 Field Documentation	334
6.11 BoxFilter7x7Param Struct Reference	334
6.11.1 Detailed Description	334
6.11.2 Field Documentation	334
6.12 BoxFilter9x9Param Struct Reference	335
6.12.1 Detailed Description	335
6.12.2 Field Documentation	335
6.13 BoxFilterParam Struct Reference	335

6.13.1 Detailed Description	335
6.13.2 Field Documentation	336
6.14 cannyEdgeDetectionParam Struct Reference	336
6.14.1 Detailed Description	336
6.14.2 Field Documentation	336
6.15 CensusMatching32Param Struct Reference	337
6.15.1 Detailed Description	337
6.15.2 Field Documentation	337
6.16 CensusMatching64Param Struct Reference	337
6.16.1 Detailed Description	337
6.16.2 Field Documentation	337
6.17 CensusMatchingPyrParam Struct Reference	338
6.17.1 Detailed Description	338
6.17.2 Field Documentation	338
6.18 ChannelExtractParam Struct Reference	338
6.18.1 Detailed Description	338
6.18.2 Field Documentation	338
6.19 ChrDnsParam Struct Reference	339
6.19.1 Detailed Description	339
6.19.2 Field Documentation	339
6.20 ChrGenParam Struct Reference	340
6.20.1 Detailed Description	340
6.20.2 Field Documentation	340
6.21 ChrGenSSParam Struct Reference	340
6.21.1 Detailed Description	340
6.21.2 Field Documentation	340
6.22 ChromaBlkParam Struct Reference	341
6.22.1 Detailed Description	341
6.22.2 Field Documentation	341
6.23 cmxRegUsage Struct Reference	341
6.23.1 Field Documentation	341
6.24 ColCombParam Struct Reference	342
6.24.1 Detailed Description	342
6.24.2 Field Documentation	342
6.25 CombDecimAwbGainsParam Struct Reference	343
6.25.1 Field Documentation	343

6.26	CombDecimStatsGainsParam Struct Reference	343
6.26.1	Field Documentation	344
6.27	CommInfo Struct Reference	344
6.27.1	Field Documentation	345
6.28	ContrastParam Struct Reference	345
6.28.1	Detailed Description	346
6.28.2	Field Documentation	346
6.29	Conv11x11Param Struct Reference	346
6.29.1	Detailed Description	346
6.29.2	Field Documentation	346
6.30	Conv15x1Param Struct Reference	347
6.30.1	Detailed Description	347
6.30.2	Field Documentation	347
6.31	Conv1x15Param Struct Reference	347
6.31.1	Detailed Description	348
6.31.2	Field Documentation	348
6.32	Conv1x5Fp16ToFp16Param Struct Reference	348
6.32.1	Detailed Description	348
6.32.2	Field Documentation	348
6.33	Conv1x5Param Struct Reference	349
6.33.1	Detailed Description	349
6.33.2	Field Documentation	349
6.34	Conv1x7Fp16ToFp16Param Struct Reference	349
6.34.1	Detailed Description	349
6.34.2	Field Documentation	350
6.35	Conv1x7Param Struct Reference	350
6.35.1	Detailed Description	350
6.35.2	Field Documentation	350
6.36	Conv1x9Param Struct Reference	350
6.36.1	Detailed Description	351
6.36.2	Field Documentation	351
6.37	Conv3x3Fp16ToFp16Param Struct Reference	351
6.37.1	Detailed Description	351
6.37.2	Field Documentation	351
6.38	Conv3x3Param Struct Reference	352
6.38.1	Detailed Description	352

6.38.2	Field Documentation	352
6.39	Conv5x1Fp16ToFp16Param Struct Reference	352
6.39.1	Detailed Description	353
6.39.2	Field Documentation	353
6.40	Conv5x1Param Struct Reference	353
6.40.1	Detailed Description	353
6.40.2	Field Documentation	353
6.41	Conv5x5Fp16ToFp16Param Struct Reference	354
6.41.1	Detailed Description	354
6.41.2	Field Documentation	354
6.42	Conv5x5Param Struct Reference	354
6.42.1	Detailed Description	354
6.42.2	Field Documentation	355
6.43	Conv7x1Fp16ToFp16Param Struct Reference	355
6.43.1	Detailed Description	355
6.43.2	Field Documentation	355
6.44	Conv7x1Param Struct Reference	355
6.44.1	Detailed Description	356
6.44.2	Field Documentation	356
6.45	Conv7x7Param Struct Reference	356
6.45.1	Detailed Description	356
6.45.2	Field Documentation	356
6.46	Conv7x7ParamFp16ToFp16 Struct Reference	357
6.46.1	Detailed Description	357
6.46.2	Field Documentation	357
6.47	Conv7x7ParamFp16ToU8 Struct Reference	357
6.47.1	Detailed Description	358
6.47.2	Field Documentation	358
6.48	Conv9x1Param Struct Reference	358
6.48.1	Detailed Description	358
6.48.2	Field Documentation	358
6.49	Conv9x9Fp16ToFp16Param Struct Reference	359
6.49.1	Detailed Description	359
6.49.2	Field Documentation	359
6.50	Conv9x9Param Struct Reference	359
6.50.1	Detailed Description	360

6.50.2	Field Documentation	360
6.51	ConvGenericParam Struct Reference	360
6.51.1	Detailed Description	360
6.51.2	Field Documentation	360
6.52	ConvParam Struct Reference	361
6.52.1	Detailed Description	361
6.52.2	Field Documentation	361
6.53	ConvSeparable11x11Fp16ToFp16Param Struct Reference	362
6.53.1	Detailed Description	362
6.53.2	Field Documentation	362
6.54	ConvSeparable11x11Param Struct Reference	362
6.54.1	Detailed Description	362
6.54.2	Field Documentation	362
6.55	ConvSeparable3x3Fp16ToFp16Param Struct Reference	363
6.55.1	Detailed Description	363
6.55.2	Field Documentation	363
6.56	ConvSeparable3x3Param Struct Reference	363
6.56.1	Detailed Description	363
6.56.2	Field Documentation	364
6.57	ConvSeparable5x5Fp16ToFp16Param Struct Reference	364
6.57.1	Detailed Description	364
6.57.2	Field Documentation	364
6.58	ConvSeparable5x5Param Struct Reference	364
6.58.1	Detailed Description	364
6.58.2	Field Documentation	365
6.59	ConvSeparable7x7Fp16ToFp16Param Struct Reference	365
6.59.1	Detailed Description	365
6.59.2	Field Documentation	365
6.60	ConvSeparable7x7Param Struct Reference	365
6.60.1	Detailed Description	365
6.60.2	Field Documentation	366
6.61	ConvSeparable9x9Fp16ToFp16Param Struct Reference	366
6.61.1	Detailed Description	366
6.61.2	Field Documentation	366
6.62	ConvSeparable9x9Param Struct Reference	366
6.62.1	Detailed Description	366

6.62.2	Field Documentation	367
6.63	CropParam Struct Reference	367
6.63.1	Detailed Description	367
6.63.2	Field Documentation	367
6.64	cvtColorChromaNV12Param Struct Reference	367
6.64.1	Detailed Description	367
6.64.2	Field Documentation	368
6.65	CvtColorChromaYUVToNV12Param Struct Reference	368
6.65.1	Detailed Description	368
6.65.2	Field Documentation	368
6.66	cvtColorLumaNV12Param Struct Reference	368
6.66.1	Detailed Description	369
6.66.2	Field Documentation	369
6.67	DbpParam Struct Reference	369
6.67.1	Detailed Description	369
6.67.2	Field Documentation	370
6.68	Dilate3x3Param Struct Reference	370
6.68.1	Detailed Description	370
6.68.2	Field Documentation	370
6.69	Dilate5x5Param Struct Reference	371
6.69.1	Detailed Description	371
6.69.2	Field Documentation	371
6.70	Dilate7x7Param Struct Reference	371
6.70.1	Detailed Description	371
6.70.2	Field Documentation	371
6.71	DilateGenericParam Struct Reference	372
6.71.1	Detailed Description	372
6.71.2	Field Documentation	372
6.72	Disp2depthParam Struct Reference	372
6.72.1	Detailed Description	372
6.72.2	Member Function Documentation	373
6.72.3	Field Documentation	373
6.73	DmaDesc Struct Reference	373
6.73.1	Field Documentation	373
6.74	DMAExtCfg Union Reference	374
6.74.1	Field Documentation	374

6.75	DmaParam Struct Reference	374
6.75.1	Detailed Description	375
6.75.2	Field Documentation	375
6.76	DMAPartialCfg Union Reference	376
6.76.1	Field Documentation	376
6.77	DmaTaskList Struct Reference	376
6.77.1	Field Documentation	376
6.78	DogLtmParam Struct Reference	377
6.78.1	Field Documentation	377
6.79	EdgeParam Struct Reference	377
6.79.1	Detailed Description	378
6.79.2	Field Documentation	378
6.80	EqualizeHistParam Struct Reference	378
6.80.1	Detailed Description	378
6.80.2	Field Documentation	379
6.81	Erode3x3Param Struct Reference	379
6.81.1	Detailed Description	379
6.81.2	Field Documentation	379
6.82	Erode5x5Param Struct Reference	379
6.82.1	Detailed Description	379
6.82.2	Field Documentation	380
6.83	Erode7x7Param Struct Reference	380
6.83.1	Detailed Description	380
6.83.2	Field Documentation	380
6.84	ExtStatsSatPixelsU32Param Struct Reference	380
6.84.1	Field Documentation	381
6.85	Fast9M2Param Struct Reference	382
6.85.1	Detailed Description	383
6.85.2	Field Documentation	383
6.86	Fast9ScoreCvParam Struct Reference	383
6.86.1	Detailed Description	383
6.86.2	Field Documentation	383
6.87	GenChrParam Struct Reference	384
6.87.1	Detailed Description	384
6.87.2	Field Documentation	384
6.88	GenLumaU8Fp16Param Struct Reference	385

6.88.1 Detailed Description	385
6.88.2 Field Documentation	385
6.89 GreyDesatParam Struct Reference	385
6.89.1 Detailed Description	385
6.89.2 Field Documentation	385
6.90 HammingDistanceParam Struct Reference	386
6.90.1 Detailed Description	386
6.90.2 Field Documentation	386
6.91 HarrisParam Struct Reference	386
6.91.1 Detailed Description	386
6.91.2 Field Documentation	387
6.92 HarrisSwParam Struct Reference	387
6.92.1 Detailed Description	387
6.92.2 Field Documentation	387
6.93 HistogramParam Struct Reference	387
6.93.1 Detailed Description	388
6.93.2 Field Documentation	388
6.94 HistogramStatParam Struct Reference	388
6.94.1 Detailed Description	388
6.94.2 Field Documentation	388
6.95 HomographyParam Struct Reference	388
6.95.1 Detailed Description	388
6.95.2 Field Documentation	389
6.96 HorizPaddingOffS Struct Reference	389
6.96.1 Field Documentation	389
6.97 HPadInfoS Struct Reference	389
6.97.1 Field Documentation	390
6.98 InterpolatePixelBilinearParam Struct Reference	390
6.98.1 Detailed Description	390
6.98.2 Field Documentation	390
6.99 LocalTMPParam Struct Reference	391
6.99.1 Detailed Description	391
6.99.2 Field Documentation	391
6.100LowLvlCorrNPiParam Struct Reference	391
6.100.1 Detailed Description	391
6.100.2 Field Documentation	391

6.101LowLvlCorrParam Struct Reference	392
6.101.1 Detailed Description	392
6.102LscParam Struct Reference	392
6.102.1 Detailed Description	393
6.102.2 Field Documentation	393
6.103Lut10to16Param Struct Reference	394
6.103.1 Detailed Description	394
6.103.2 Field Documentation	394
6.104Lut10to8Param Struct Reference	394
6.104.1 Detailed Description	394
6.104.2 Field Documentation	394
6.105Lut12to16Param Struct Reference	395
6.105.1 Detailed Description	395
6.105.2 Field Documentation	395
6.106Lut12to8Param Struct Reference	395
6.106.1 Detailed Description	395
6.106.2 Field Documentation	395
6.107Lut8to8Param Struct Reference	396
6.107.1 Detailed Description	396
6.107.2 Field Documentation	396
6.108LutParam Struct Reference	396
6.108.1 Detailed Description	397
6.108.2 Field Documentation	397
6.109MaxTest3x3fp16Param Struct Reference	397
6.109.1 Detailed Description	398
6.109.2 Field Documentation	398
6.110MedParam Struct Reference	398
6.110.1 Detailed Description	398
6.110.2 Field Documentation	399
6.111memRegDescriptor Struct Reference	399
6.111.1 Field Documentation	399
6.112MinMaxPosParam Struct Reference	399
6.112.1 Detailed Description	400
6.112.2 Field Documentation	400
6.113minMaxValParam Struct Reference	400
6.113.1 Detailed Description	400

6.113.2 Field Documentation	400
6.114MinTest3x3fp16Param Struct Reference	401
6.114.1 Detailed Description	401
6.114.2 Field Documentation	401
6.115MipiRxLoopbackParam Struct Reference	401
6.115.1 Field Documentation	402
6.116MipiRxParam Struct Reference	402
6.116.1 Detailed Description	402
6.116.2 Field Documentation	403
6.117MipiTxLoopbackParam Struct Reference	404
6.117.1 Field Documentation	404
6.118MipiTxParam Struct Reference	405
6.118.1 Detailed Description	405
6.118.2 Field Documentation	406
6.119MixMedianParam Struct Reference	407
6.119.1 Detailed Description	408
6.119.2 Field Documentation	408
6.120MonoImbalanceParam Struct Reference	408
6.120.1 Detailed Description	408
6.120.2 Field Documentation	408
6.121nonMax3x3Fp32Param Struct Reference	409
6.121.1 Field Documentation	409
6.122ParentInfoS Struct Reference	409
6.122.1 Field Documentation	409
6.123PixelUnpackerMipi10bParam Struct Reference	409
6.123.1 Detailed Description	410
6.123.2 Field Documentation	410
6.124PixelUnpackerParam Struct Reference	410
6.124.1 Detailed Description	410
6.124.2 Field Documentation	410
6.125PixelUnpackerWBParam Struct Reference	411
6.125.1 Detailed Description	411
6.125.2 Field Documentation	411
6.126PolyFirParam Struct Reference	411
6.126.1 Detailed Description	412
6.126.2 Field Documentation	412

6.127	positionKernelParam Struct Reference	414
6.127.1	Detailed Description	414
6.127.2	Field Documentation	414
6.128	PpAf Struct Reference	415
6.128.1	Detailed Description	415
6.128.2	Field Documentation	415
6.129	PurpleFlareParam Struct Reference	417
6.129.1	Detailed Description	417
6.129.2	Field Documentation	417
6.130	RandNoiseFp16Param Struct Reference	417
6.130.1	Detailed Description	417
6.130.2	Field Documentation	417
6.131	RandNoiseParam Struct Reference	417
6.131.1	Detailed Description	418
6.131.2	Field Documentation	418
6.132	RawParam Struct Reference	418
6.132.1	Detailed Description	419
6.132.2	Field Documentation	419
6.133	SchedInfoS Struct Reference	421
6.133.1	Field Documentation	421
6.134	Semaphore Class Reference	422
6.134.1	Constructor & Destructor Documentation	422
6.134.2	Member Function Documentation	423
6.134.3	Field Documentation	423
6.135	SigmaParam Struct Reference	423
6.135.1	Detailed Description	424
6.135.2	Field Documentation	424
6.136	SIPP_ACCESS_SCHEDULER Struct Reference	425
6.136.1	Field Documentation	425
6.137	SIPP_ACCESS_SCHEDULER_QU Struct Reference	426
6.137.1	Field Documentation	426
6.138	SIPP_ACCESS_SCHEDULER_QU_ENTRY Struct Reference	427
6.138.1	Field Documentation	427
6.139	SIPP_HW_SESSION Struct Reference	427
6.139.1	Field Documentation	428
6.140	SIPP_PAL_QU Struct Reference	429

6.140.1 Field Documentation	429
6.141SIPP_PAL_THREAD Struct Reference	429
6.141.1 Field Documentation	430
6.142SIPP_PIPELINE_FINALISED_DATA Struct Reference	430
6.142.1 Field Documentation	430
6.143SIPP_TRACE_FLAGS Struct Reference	430
6.143.1 Field Documentation	430
6.144SippCmxBufferMapS Struct Reference	430
6.144.1 Field Documentation	431
6.145SippFilterS Struct Reference	431
6.145.1 Field Documentation	433
6.146SippFilterSchedInfoS Struct Reference	442
6.146.1 Field Documentation	443
6.147SippGlobals Struct Reference	443
6.147.1 Field Documentation	443
6.148SippHeapCB Struct Reference	443
6.148.1 Field Documentation	444
6.149SippHwBufS Struct Reference	444
6.149.1 Field Documentation	444
6.150SippHwIOBuf Struct Reference	445
6.150.1 Field Documentation	445
6.151SippManagedBufSchedInfo Struct Reference	446
6.151.1 Field Documentation	446
6.152SippMemRegion Struct Reference	446
6.152.1 Field Documentation	446
6.153SippMemRegionListNode Struct Reference	447
6.153.1 Field Documentation	447
6.154sippOpipeBufInfo Struct Reference	448
6.154.1 Field Documentation	448
6.155SippOseS Struct Reference	448
6.155.1 Field Documentation	449
6.156SippPipelineS Struct Reference	449
6.156.1 Field Documentation	451
6.157SippPixelChunkPos Struct Reference	460
6.157.1 Field Documentation	460
6.158SippSchEntS Struct Reference	460

6.158.1 Field Documentation	461
6.159SippVPhysMapS Struct Reference	464
6.159.1 Field Documentation	464
6.160sSchedIBufUsageInfoS Struct Reference	464
6.160.1 Field Documentation	465
6.161sSchLineBufferS Struct Reference	466
6.161.1 Field Documentation	466
6.162sSippCdmaQuEntryS Struct Reference	467
6.162.1 Field Documentation	468
6.163sSippCdmaQuS Struct Reference	468
6.163.1 Field Documentation	468
6.164sSippCMDQuEntryS Struct Reference	468
6.164.1 Field Documentation	469
6.165sSippCMDQuS Struct Reference	469
6.165.1 Field Documentation	469
6.166StartBicubicParam Struct Reference	469
6.166.1 Field Documentation	470
6.167StatsAwbSatPixelsParam Struct Reference	470
6.167.1 Field Documentation	471
6.168StatsAwbSatPixelsParamU32 Struct Reference	472
6.168.1 Field Documentation	473
6.169SubpixelFilterParam Struct Reference	474
6.169.1 Detailed Description	474
6.169.2 Field Documentation	474
6.170ThresholdBinaryRangeParam Struct Reference	475
6.170.1 Detailed Description	475
6.170.2 Field Documentation	475
6.171ThresholdBinaryU8Param Struct Reference	475
6.171.1 Detailed Description	476
6.171.2 Field Documentation	476
6.172ThresholdFilterParam Struct Reference	476
6.172.1 Detailed Description	476
6.172.2 Field Documentation	476
6.173ThresholdParam Struct Reference	476
6.173.1 Detailed Description	477
6.173.2 Field Documentation	477

6.174tMLPISStartCQCtrl Struct Reference	477
6.174.1 Field Documentation	478
6.175TripleConv3x3 Struct Reference	478
6.175.1 Field Documentation	478
6.176tRTStats Struct Reference	478
6.176.1 Field Documentation	479
6.177tSippFramework Struct Reference	479
6.177.1 Field Documentation	479
6.178tSippMCB Struct Reference	480
6.178.1 Field Documentation	480
6.179tSippPhysicalPool Struct Reference	480
6.179.1 Field Documentation	481
6.180tSippPipelineSuper Struct Reference	481
6.180.1 Field Documentation	481
6.181tsSippHeap Struct Reference	482
6.181.1 Field Documentation	482
6.182UndistortBParam Struct Reference	483
6.182.1 Detailed Description	483
6.182.2 Field Documentation	483
6.183UnitInfo Struct Reference	484
6.183.1 Field Documentation	484
6.184UsmParam Struct Reference	484
6.184.1 Detailed Description	485
6.184.2 Field Documentation	485
6.185WhiteBalanceBayerGBRGParam Struct Reference	486
6.185.1 Detailed Description	486
6.185.2 Field Documentation	486
6.186WhiteBalanceRGBParam Struct Reference	486
6.186.1 Detailed Description	487
6.186.2 Field Documentation	487
6.187XYGenParam Struct Reference	487
6.187.1 Detailed Description	487
6.187.2 Field Documentation	488
6.188YDnsParam Struct Reference	488
6.188.1 Detailed Description	488
6.188.2 Field Documentation	489

6.189	YDnsRefFp16Param Struct Reference	489
6.189.1	Detailed Description	490
6.189.2	Field Documentation	490
6.190	YDnsRefLut10bppParam Struct Reference	490
6.190.1	Detailed Description	491
6.190.2	Field Documentation	491
6.191	YDnsRefParam Struct Reference	491
6.191.1	Detailed Description	491
6.191.2	Field Documentation	491
7	File Documentation	493
7.1	absdiff.h File Reference	493
7.1.1	Detailed Description	493
7.2	accumulateSquare.h File Reference	493
7.2.1	Detailed Description	494
7.3	accumulateWeighted.h File Reference	494
7.3.1	Detailed Description	494
7.4	arithmeticAdd.h File Reference	494
7.4.1	Detailed Description	495
7.5	arithmeticAddmask.h File Reference	495
7.5.1	Detailed Description	495
7.6	arithmeticSub.h File Reference	495
7.6.1	Detailed Description	495
7.7	arithmeticSubFp16ToFp16.h File Reference	496
7.7.1	Detailed Description	496
7.8	arithmeticSubmask.h File Reference	496
7.8.1	Detailed Description	496
7.9	avg.h File Reference	497
7.9.1	Detailed Description	497
7.10	bilateral5x5.h File Reference	497
7.10.1	Detailed Description	497
7.11	bitwiseAnd.h File Reference	498
7.11.1	Detailed Description	498
7.12	bitwiseAndMask.h File Reference	498
7.12.1	Detailed Description	498
7.13	bitwiseNot.h File Reference	498
7.13.1	Detailed Description	499

7.14	bitwiseOr.h File Reference	499
7.14.1	Detailed Description	499
7.15	bitwiseOrMask.h File Reference	499
7.15.1	Detailed Description	500
7.16	bitwiseXor.h File Reference	500
7.16.1	Detailed Description	500
7.17	bitwiseXorMask.h File Reference	500
7.17.1	Detailed Description	500
7.18	boxFilter.h File Reference	501
7.18.1	Detailed Description	501
7.18.2	Macro Definition Documentation	501
7.19	boxFilter11x11.h File Reference	502
7.19.1	Detailed Description	502
7.20	boxFilter13x13.h File Reference	502
7.20.1	Detailed Description	503
7.21	boxFilter15x15.h File Reference	503
7.21.1	Detailed Description	503
7.22	boxFilter3x3.h File Reference	503
7.22.1	Detailed Description	504
7.23	boxFilter5x5.h File Reference	504
7.23.1	Detailed Description	504
7.24	boxFilter7x7.h File Reference	504
7.24.1	Detailed Description	505
7.25	boxFilter9x9.h File Reference	505
7.25.1	Detailed Description	505
7.26	cannyEdgeDetection.h File Reference	506
7.26.1	Detailed Description	506
7.27	censusMatching16.h File Reference	506
7.27.1	Detailed Description	506
7.28	censusMatching32.h File Reference	507
7.28.1	Detailed Description	507
7.29	censusMatching64.h File Reference	507
7.29.1	Detailed Description	508
7.30	censusMatching65.h File Reference	508
7.30.1	Detailed Description	508
7.31	censusMatchingPyr.h File Reference	508

7.31.1 Detailed Description	509
7.32 censusMin16.h File Reference	509
7.32.1 Detailed Description	509
7.33 censusMin64.h File Reference	509
7.33.1 Detailed Description	509
7.34 censusMin65.h File Reference	510
7.34.1 Detailed Description	510
7.35 censusMin7.h File Reference	510
7.35.1 Detailed Description	510
7.36 censusTransform5x5.h File Reference	511
7.36.1 Detailed Description	511
7.37 channelExtract.h File Reference	511
7.37.1 Detailed Description	511
7.38 chromaBlock.h File Reference	512
7.38.1 Detailed Description	512
7.39 combDecimDemosaicAwbGains.h File Reference	512
7.39.1 Enumeration Type Documentation	513
7.39.2 Function Documentation	513
7.40 combDecimDemosaicAwbGainsStats.c File Reference	513
7.40.1 Macro Definition Documentation	513
7.40.2 Function Documentation	513
7.40.3 Variable Documentation	514
7.41 combDecimDemosaicAwbGainsStats.h File Reference	514
7.41.1 Function Documentation	514
7.42 contrast.h File Reference	514
7.42.1 Detailed Description	515
7.43 conv11x11.h File Reference	515
7.43.1 Detailed Description	515
7.44 conv15x1.h File Reference	516
7.44.1 Detailed Description	516
7.45 conv1x15.h File Reference	516
7.45.1 Detailed Description	517
7.46 conv1x5.h File Reference	517
7.46.1 Detailed Description	517
7.47 conv1x5Fp16ToFp16.h File Reference	517
7.47.1 Detailed Description	518

7.48	conv1x7.h File Reference	518
7.48.1	Detailed Description	518
7.49	conv1x7Fp16ToFp16.h File Reference	518
7.49.1	Detailed Description	519
7.50	conv1x9.h File Reference	519
7.50.1	Detailed Description	519
7.51	conv3x3.h File Reference	520
7.51.1	Detailed Description	520
7.52	conv3x3Fp16ToFp16.h File Reference	520
7.52.1	Detailed Description	521
7.53	conv5x1.h File Reference	521
7.53.1	Detailed Description	521
7.54	conv5x1Fp16ToFp16.h File Reference	521
7.54.1	Detailed Description	522
7.55	conv5x5.h File Reference	522
7.55.1	Detailed Description	522
7.56	conv5x5Fp16ToFp16.h File Reference	522
7.56.1	Detailed Description	523
7.57	conv7x1.h File Reference	523
7.57.1	Detailed Description	523
7.58	conv7x1Fp16ToFp16.h File Reference	524
7.58.1	Detailed Description	524
7.59	conv7x7.h File Reference	524
7.59.1	Detailed Description	525
7.60	conv7x7Fp16ToFp16.h File Reference	525
7.60.1	Detailed Description	525
7.61	conv7x7Fp16ToU8.h File Reference	525
7.61.1	Detailed Description	526
7.62	conv9x1.h File Reference	526
7.62.1	Detailed Description	526
7.63	conv9x9.h File Reference	526
7.63.1	Detailed Description	527
7.64	conv9x9Fp16ToFp16.h File Reference	527
7.64.1	Detailed Description	527
7.65	convert16bppTo8bpp.h File Reference	528
7.65.1	Detailed Description	528

7.66	convertF16ToU8.h File Reference	528
7.66.1	Detailed Description	528
7.67	convertFrom12BppTo8Bpp.h File Reference	528
7.67.1	Detailed Description	529
7.68	convertPFp16U16.h File Reference	529
7.68.1	Detailed Description	529
7.69	convertPU16Fp16.h File Reference	529
7.69.1	Detailed Description	530
7.70	convertU8ToF16.h File Reference	530
7.70.1	Detailed Description	530
7.71	convertYUV400ToYUV422.h File Reference	530
7.71.1	Detailed Description	530
7.72	convGeneric.h File Reference	531
7.72.1	Detailed Description	531
7.73	convSeparable11x11.h File Reference	531
7.73.1	Detailed Description	532
7.74	convSeparable11x11Fp16ToFp16.h File Reference	532
7.74.1	Detailed Description	532
7.75	convSeparable3x3.h File Reference	532
7.75.1	Detailed Description	533
7.76	convSeparable3x3Fp16ToFp16.h File Reference	533
7.76.1	Detailed Description	533
7.77	convSeparable5x5.h File Reference	533
7.77.1	Detailed Description	534
7.78	convSeparable5x5Fp16ToFp16.h File Reference	534
7.78.1	Detailed Description	534
7.79	convSeparable7x7.h File Reference	535
7.79.1	Detailed Description	535
7.80	convSeparable7x7Fp16ToFp16.h File Reference	535
7.80.1	Detailed Description	536
7.81	convSeparable9x9.h File Reference	536
7.81.1	Detailed Description	536
7.82	convSeparable9x9Fp16ToFp16.h File Reference	536
7.82.1	Detailed Description	537
7.83	convYuv444.h File Reference	537
7.83.1	Detailed Description	537

7.84	copy.h File Reference	537
7.84.1	Detailed Description	538
7.85	cornerMinEigenVal.h File Reference	538
7.85.1	Detailed Description	538
7.86	cornerMinEigenValpatched.h File Reference	538
7.86.1	Detailed Description	538
7.87	crop.h File Reference	539
7.87.1	Detailed Description	539
7.88	cropCvtPlaneMode.h File Reference	539
7.88.1	Detailed Description	540
7.89	cvtColorChromaYUVToNV12.h File Reference	540
7.89.1	Detailed Description	540
7.89.2	Function Documentation	540
7.90	cvtColorNV21toRGB.h File Reference	541
7.90.1	Detailed Description	541
7.91	cvtColorRGBfp16ToLumaU8.h File Reference	541
7.91.1	Detailed Description	541
7.92	cvtColorRGBfp16ToUV420U8.h File Reference	541
7.92.1	Detailed Description	542
7.93	cvtColorRGBtoChromaNV12.h File Reference	542
7.93.1	Detailed Description	542
7.94	cvtColorRGBtoLuma.h File Reference	542
7.94.1	Detailed Description	543
7.95	cvtColorRGBtoLumaNV12.h File Reference	543
7.95.1	Detailed Description	543
7.96	cvtColorRGBtoUV.h File Reference	543
7.96.1	Detailed Description	544
7.97	cvtColorRGBtoUV420.h File Reference	544
7.97.1	Detailed Description	544
7.98	cvtColorRGBToYUV422.h File Reference	544
7.98.1	Detailed Description	545
7.99	cvtColorYUV422ToRGB.h File Reference	545
7.99.1	Detailed Description	545
7.100	cvtColorYUVToRGB.h File Reference	545
7.100.1	Detailed Description	545
7.101	dilate3x3.h File Reference	546

7.101.1 Detailed Description	546
7.102dilate5x5.h File Reference	546
7.102.1 Detailed Description	547
7.103dilate7x7.h File Reference	547
7.103.1 Detailed Description	547
7.104dilateGeneric.h File Reference	547
7.104.1 Detailed Description	548
7.105disp2depth.h File Reference	548
7.105.1 Detailed Description	548
7.105.2 Function Documentation	548
7.106disp2depth_exec.c File Reference	549
7.106.1 Function Documentation	549
7.107disp2depth_exec.h File Reference	549
7.107.1 Function Documentation	550
7.108equalizeHist.h File Reference	550
7.108.1 Detailed Description	551
7.109erode3x3.h File Reference	551
7.109.1 Detailed Description	551
7.110erode5x5.h File Reference	551
7.110.1 Detailed Description	552
7.111erode7x7.h File Reference	552
7.111.1 Detailed Description	552
7.112extAfStats.h File Reference	552
7.112.1 Detailed Description	553
7.113extStatsSatPixelsU32.h File Reference	553
7.113.1 Function Documentation	553
7.114fast9M2.h File Reference	553
7.114.1 Detailed Description	554
7.115fast9ScoreCv.h File Reference	554
7.115.1 Detailed Description	554
7.116gauss.h File Reference	555
7.116.1 Detailed Description	555
7.117gaussHx2.h File Reference	555
7.117.1 Detailed Description	555
7.118gaussHx2_fp16.h File Reference	555
7.118.1 Detailed Description	556

7.119gaussVx2.h File Reference	556
7.119.1 Detailed Description	556
7.120gaussVx2_fp16.h File Reference	556
7.120.1 Detailed Description	557
7.121genChroma.h File Reference	557
7.121.1 Detailed Description	557
7.122genChromaSS.h File Reference	557
7.122.1 Detailed Description	558
7.123genDnsRef.h File Reference	558
7.123.1 Detailed Description	558
7.124genDnsRefFp16.h File Reference	558
7.124.1 Detailed Description	559
7.125genLuma.h File Reference	559
7.125.1 Detailed Description	559
7.126genLumaU8Fp16.h File Reference	559
7.126.1 Detailed Description	560
7.127greyDesat.h File Reference	560
7.127.1 Detailed Description	560
7.128hammingDistance.h File Reference	561
7.128.1 Detailed Description	561
7.129harrisResponse.h File Reference	561
7.129.1 Detailed Description	562
7.130histogram.h File Reference	562
7.130.1 Detailed Description	562
7.131histogramStat.h File Reference	562
7.131.1 Detailed Description	563
7.132homography.h File Reference	563
7.132.1 Detailed Description	563
7.133integralImageSqSumF32M2.h File Reference	563
7.133.1 Detailed Description	564
7.134integralImageSqSumU32M2.h File Reference	564
7.134.1 Detailed Description	564
7.135integralImageSumF32M2.h File Reference	564
7.135.1 Detailed Description	565
7.136integralImageSumU16U32.h File Reference	565
7.136.1 Detailed Description	565

7.137integralImageSumU32M2.h File Reference	565
7.137.1 Detailed Description	565
7.138interpolatePixelBilinear.h File Reference	566
7.138.1 Detailed Description	566
7.139laplacian3x3.h File Reference	566
7.139.1 Detailed Description	567
7.140laplacian5x5.h File Reference	567
7.140.1 Detailed Description	567
7.141laplacian5x5Fp16ToFp16.h File Reference	567
7.141.1 Detailed Description	567
7.142laplacian7x7.h File Reference	568
7.142.1 Detailed Description	568
7.143laplacian7x7Fp16ToFp16.h File Reference	568
7.143.1 Detailed Description	568
7.144localMaxMin3x3_fp16.h File Reference	569
7.144.1 Detailed Description	569
7.145localTM.h File Reference	569
7.145.1 Detailed Description	569
7.146lowLvlCorr.h File Reference	570
7.146.1 Detailed Description	570
7.147lowLvlCorrMultiplePlanes.c File Reference	570
7.147.1 Macro Definition Documentation	571
7.148lowLvlCorrMultiplePlanes.h File Reference	571
7.148.1 Detailed Description	571
7.149lumaBlur.h File Reference	571
7.149.1 Detailed Description	572
7.150lut10to16.h File Reference	572
7.150.1 Detailed Description	572
7.151lut10to8.h File Reference	572
7.151.1 Detailed Description	573
7.152lut12to16.h File Reference	573
7.152.1 Detailed Description	573
7.153lut12to8.h File Reference	573
7.153.1 Detailed Description	574
7.154lut8to8.h File Reference	574
7.154.1 Detailed Description	574

7.155lutP10BppU16inU8out.h File Reference	575
7.155.1 Detailed Description	575
7.156maxTest3x3_fp16.h File Reference	575
7.156.1 Detailed Description	576
7.157MDKdox-Sipp-intro.txt File Reference	576
7.158meanStdDev.h File Reference	576
7.158.1 Detailed Description	576
7.159minMaxPos.h File Reference	576
7.159.1 Detailed Description	577
7.160minMaxValue.h File Reference	577
7.160.1 Detailed Description	577
7.161minTest3x3_fp16.h File Reference	577
7.161.1 Detailed Description	578
7.162mixMedian.h File Reference	578
7.162.1 Detailed Description	578
7.163monoImbalance.h File Reference	579
7.163.1 Detailed Description	579
7.163.2 Function Documentation	579
7.164myriad2SippDefs.inc File Reference	579
7.165negative.h File Reference	579
7.165.1 Detailed Description	580
7.166nonMax3x3Fp32.h File Reference	580
7.166.1 Detailed Description	580
7.167nonMax3x3U8.h File Reference	580
7.167.1 Detailed Description	581
7.168padBayer5.h File Reference	581
7.168.1 Detailed Description	581
7.168.2 Function Documentation	581
7.169padBayer5Frame.h File Reference	581
7.169.1 Detailed Description	582
7.170pixelPacker10b.h File Reference	582
7.170.1 Detailed Description	582
7.171pixelUnpacker.h File Reference	582
7.171.1 Detailed Description	583
7.172pixelUnpackerMipi10b.h File Reference	583
7.172.1 Detailed Description	583

7.173pixelUnpackerWB.h File Reference	583
7.173.1 Detailed Description	584
7.174positionKernel.h File Reference	584
7.174.1 Detailed Description	584
7.175purpleFlare.h File Reference	584
7.175.1 Detailed Description	585
7.176pyrDown.h File Reference	585
7.176.1 Detailed Description	585
7.177randNoise.h File Reference	585
7.177.1 Detailed Description	586
7.178randNoiseFp16.h File Reference	586
7.178.1 Detailed Description	586
7.179sad11x11.h File Reference	587
7.179.1 Detailed Description	587
7.180sad5x5.h File Reference	587
7.180.1 Detailed Description	587
7.181scale05BilinHV.h File Reference	587
7.181.1 Detailed Description	588
7.182scale05BilinHV_Fp16U8.h File Reference	588
7.182.1 Detailed Description	588
7.183scale05BilinHVFp16.h File Reference	588
7.183.1 Detailed Description	589
7.184scale05Lanc6HV.h File Reference	589
7.184.1 Detailed Description	589
7.185scale05Lanc7HV.h File Reference	589
7.185.1 Detailed Description	589
7.186scale2xBilinHV.h File Reference	590
7.186.1 Detailed Description	590
7.187scale2xBilinHV_025_075_Fp16ToFp16.h File Reference	590
7.187.1 Detailed Description	590
7.188scale2xBilinHV_025_075_U16ToU16.h File Reference	591
7.188.1 Detailed Description	591
7.189scale2xBilinHV_Fp16U8_phase025_075.h File Reference	591
7.189.1 Detailed Description	591
7.190scale2xBilinHV_U8ToU8_phase025_075.h File Reference	592
7.190.1 Detailed Description	592

7.191scale2xLancH.h File Reference	592
7.191.1 Detailed Description	592
7.192scale2xLancHV.h File Reference	592
7.192.1 Detailed Description	593
7.193scale2xLancV.h File Reference	593
7.193.1 Detailed Description	593
7.194scaleBilinArb.h File Reference	593
7.194.1 Detailed Description	594
7.195scharr_fp16.h File Reference	594
7.195.1 Detailed Description	594
7.196sipp.h File Reference	594
7.196.1 Detailed Description	594
7.197sipp_ma2x5x.h File Reference	594
7.197.1 Detailed Description	597
7.197.2 Macro Definition Documentation	597
7.197.3 Function Documentation	599
7.198sippAccessScheduler.c File Reference	606
7.198.1 Detailed Description	606
7.198.2 Macro Definition Documentation	607
7.198.3 Function Documentation	607
7.198.4 Variable Documentation	607
7.199sippAccessScheduler.h File Reference	608
7.199.1 Detailed Description	608
7.199.2 Function Documentation	608
7.200sippAccessSchedulerTypes.h File Reference	608
7.200.1 Detailed Description	609
7.200.2 Macro Definition Documentation	610
7.200.3 Typedef Documentation	610
7.200.4 Enumeration Type Documentation	610
7.201sippAnalysePipema2x5x.c File Reference	610
7.201.1 Detailed Description	611
7.201.2 Function Documentation	611
7.202sippApi.c File Reference	611
7.202.1 Detailed Description	612
7.202.2 Macro Definition Documentation	612
7.202.3 Function Documentation	612

7.202.4 Variable Documentation	613
7.203sippApiInternal.c File Reference	614
7.203.1 Function Documentation	614
7.203.2 Variable Documentation	615
7.204sippBaseTypes.h File Reference	615
7.204.1 Macro Definition Documentation	616
7.204.2 Typedef Documentation	616
7.205sippCfg.h File Reference	617
7.205.1 Macro Definition Documentation	617
7.206sippCmxDmaIf.c File Reference	620
7.206.1 Detailed Description	621
7.206.2 Macro Definition Documentation	621
7.206.3 Function Documentation	621
7.206.4 Variable Documentation	623
7.207sippCoreApi.c File Reference	623
7.207.1 Detailed Description	624
7.207.2 Function Documentation	624
7.207.3 Variable Documentation	624
7.208sippCoreGlobals.c File Reference	625
7.208.1 Detailed Description	625
7.209sippCoreHw.c File Reference	625
7.209.1 Function Documentation	625
7.210sippCoreUtils.c File Reference	626
7.210.1 Detailed Description	626
7.210.2 Function Documentation	626
7.210.3 Variable Documentation	628
7.211sippDbg.c File Reference	628
7.211.1 Detailed Description	629
7.211.2 Function Documentation	629
7.211.3 Variable Documentation	630
7.212sippDefines.h File Reference	630
7.212.1 Macro Definition Documentation	631
7.213sippDriverCmxDmaIf.c File Reference	634
7.213.1 Detailed Description	635
7.214sippError.c File Reference	635
7.214.1 Detailed Description	635

7.214.2 Function Documentation	636
7.214.3 Variable Documentation	636
7.215sippEvents.h File Reference	637
7.215.1 Typedef Documentation	637
7.215.2 Enumeration Type Documentation	637
7.216sippFastExeUpd.h File Reference	638
7.216.1 Detailed Description	638
7.217sippFilterAccesors.c File Reference	638
7.217.1 Function Documentation	638
7.218sippGenericRuntime.c File Reference	641
7.218.1 Detailed Description	641
7.218.2 Function Documentation	641
7.218.3 Variable Documentation	642
7.219sippGenericRuntimema2x5x.c File Reference	642
7.219.1 Detailed Description	643
7.219.2 Function Documentation	643
7.220sippGenericSchApi.c File Reference	643
7.220.1 Function Documentation	644
7.221sippGenericSchDebug.c File Reference	645
7.221.1 Function Documentation	645
7.222sippGenericSchReq.c File Reference	645
7.222.1 Detailed Description	646
7.222.2 Function Documentation	646
7.223sippGenericSchWrite.c File Reference	647
7.223.1 Detailed Description	647
7.223.2 Function Documentation	647
7.224sippHeap.c File Reference	647
7.224.1 Detailed Description	648
7.224.2 Macro Definition Documentation	648
7.224.3 Function Documentation	648
7.224.4 Variable Documentation	649
7.225sippHpad.c File Reference	649
7.225.1 Detailed Description	650
7.225.2 Function Documentation	650
7.225.3 Variable Documentation	650
7.226sippHwBitfieldDefs.h File Reference	650

7.226.1 Macro Definition Documentation	675
7.226.2 Enumeration Type Documentation	723
7.227sippHwChromaDns.c File Reference	724
7.227.1 Detailed Description	724
7.228sippHwColComb.c File Reference	724
7.228.1 Detailed Description	724
7.229sippHwCommon.c File Reference	724
7.229.1 Detailed Description	725
7.229.2 Macro Definition Documentation	725
7.229.3 Function Documentation	725
7.230sippHwCommon_ma2x5x.h File Reference	726
7.230.1 Macro Definition Documentation	729
7.230.2 Enumeration Type Documentation	734
7.231sippHwConv.c File Reference	735
7.231.1 Detailed Description	735
7.232sippHwDebayer.c File Reference	735
7.232.1 Detailed Description	735
7.233sippHwDefs.h File Reference	735
7.233.1 Detailed Description	735
7.234sippHwDefs_ma2x5x.h File Reference	736
7.234.1 Detailed Description	737
7.234.2 Function Documentation	737
7.235sippHwDogLtm.c File Reference	738
7.235.1 Detailed Description	738
7.236sippHwEdge.c File Reference	738
7.236.1 Detailed Description	738
7.237sippHwGenChroma.c File Reference	738
7.237.1 Detailed Description	738
7.238sippHwHarris.c File Reference	739
7.238.1 Detailed Description	739
7.239sippHwIds.h File Reference	739
7.239.1 Detailed Description	741
7.239.2 Macro Definition Documentation	741
7.240sippHwLsc.c File Reference	746
7.240.1 Detailed Description	747
7.241sippHwLumaDns.c File Reference	747

7.241.1 Detailed Description	747
7.242sippHwLut.c File Reference	747
7.242.1 Detailed Description	747
7.243sippHwMedian.c File Reference	748
7.243.1 Detailed Description	748
7.244sippHwMipiRx.c File Reference	748
7.244.1 Detailed Description	748
7.244.2 Function Documentation	749
7.245sippHwMipiTx.c File Reference	749
7.245.1 Detailed Description	749
7.245.2 Function Documentation	750
7.246sippHwPolyFir.c File Reference	750
7.246.1 Detailed Description	750
7.247sippHwRaw.c File Reference	750
7.247.1 Detailed Description	750
7.248sippHWSessionControl.c File Reference	751
7.248.1 Detailed Description	751
7.248.2 Function Documentation	751
7.248.3 Variable Documentation	752
7.249sippHwSigma.c File Reference	752
7.249.1 Detailed Description	752
7.249.2 Function Documentation	752
7.250sippHwUnsharp.c File Reference	753
7.250.1 Detailed Description	753
7.251sippInternal.h File Reference	753
7.251.1 Detailed Description	756
7.251.2 Function Documentation	757
7.252sippIoPtrs.c File Reference	766
7.252.1 Detailed Description	766
7.252.2 Function Documentation	767
7.252.3 Variable Documentation	769
7.253sippIsr.c File Reference	769
7.253.1 Detailed Description	769
7.253.2 Function Documentation	770
7.253.3 Variable Documentation	770
7.254sippManagerApi.c File Reference	770

7.254.1 Detailed Description	770
7.254.2 Function Documentation	771
7.254.3 Variable Documentation	771
7.255sippMem.c File Reference	771
7.255.1 Detailed Description	772
7.255.2 Macro Definition Documentation	772
7.255.3 Function Documentation	772
7.255.4 Variable Documentation	774
7.256sippMemLineBuffer.c File Reference	774
7.256.1 Detailed Description	775
7.256.2 Function Documentation	775
7.256.3 Variable Documentation	776
7.257sippMLPIRuntime.c File Reference	776
7.257.1 Detailed Description	776
7.258sippMLPIRuntimeema2x5x.c File Reference	777
7.258.1 Detailed Description	777
7.259sippMLPISchApi.c File Reference	777
7.260sippMLPISchDebug.c File Reference	777
7.261sippMLPISchReq.c File Reference	777
7.262sippMLPISchWrite.c File Reference	777
7.262.1 Detailed Description	777
7.263sippOPipeRuntime.c File Reference	777
7.263.1 Detailed Description	777
7.264sippOPipeSchApi.c File Reference	778
7.265sippOPipeSchedulingEntity.c File Reference	778
7.265.1 Detailed Description	779
7.265.2 Macro Definition Documentation	779
7.265.3 Function Documentation	780
7.266sippPal.h File Reference	781
7.266.1 Function Documentation	782
7.267sippPalTypes.h File Reference	783
7.267.1 Macro Definition Documentation	786
7.267.2 Typedef Documentation	791
7.268sippPalTypes.h File Reference	791
7.268.1 Macro Definition Documentation	794
7.268.2 Typedef Documentation	798

7.269sippPalTypes.h File Reference	798
7.269.1 Macro Definition Documentation	801
7.269.2 Typedef Documentation	806
7.270sippPipelineValidate.c File Reference	806
7.270.1 Detailed Description	806
7.270.2 Function Documentation	806
7.271sippPipeSessionControl.c File Reference	807
7.271.1 Detailed Description	807
7.271.2 Function Documentation	807
7.271.3 Variable Documentation	807
7.272sippPlatform.h File Reference	807
7.272.1 Detailed Description	808
7.273sippPlatform_ma2x5x.h File Reference	808
7.273.1 Detailed Description	809
7.273.2 Macro Definition Documentation	809
7.273.3 Function Documentation	810
7.274sippPlatformAbstractionLayer.c File Reference	811
7.274.1 Macro Definition Documentation	811
7.274.2 Function Documentation	811
7.275sippPlatformAbstractionLayer.c File Reference	812
7.275.1 Macro Definition Documentation	812
7.275.2 Function Documentation	813
7.275.3 Variable Documentation	813
7.276sippPlatformAbstractionLayer.c File Reference	813
7.276.1 Detailed Description	814
7.276.2 Function Documentation	814
7.277sippScheduleIsr.c File Reference	815
7.277.1 Detailed Description	816
7.277.2 Function Documentation	816
7.277.3 Variable Documentation	817
7.278sippSchTypes.h File Reference	817
7.278.1 Typedef Documentation	817
7.278.2 Enumeration Type Documentation	817
7.279sippSessionControl.h File Reference	818
7.279.1 Detailed Description	818
7.279.2 Macro Definition Documentation	818

7.279.3 Enumeration Type Documentation	818
7.280sippShave.c File Reference	818
7.280.1 Detailed Description	819
7.280.2 Function Documentation	819
7.280.3 Variable Documentation	819
7.281sippShaveIf.c File Reference	820
7.281.1 Detailed Description	820
7.281.2 Function Documentation	820
7.282sippShaveMacros.h File Reference	820
7.282.1 Detailed Description	821
7.283sippShaveSym.h File Reference	821
7.284sippShvDbg.c File Reference	821
7.284.1 Detailed Description	821
7.284.2 Function Documentation	821
7.284.3 Variable Documentation	821
7.285sippTestCommon.c File Reference	821
7.285.1 Detailed Description	822
7.285.2 Function Documentation	822
7.286sippTestCommon.h File Reference	822
7.286.1 Detailed Description	822
7.287sippTestCommon_ma2x5x.h File Reference	822
7.287.1 Detailed Description	823
7.287.2 Function Documentation	823
7.287.3 Variable Documentation	823
7.288sippThread.c File Reference	823
7.288.1 Detailed Description	823
7.289sippTypes.h File Reference	823
7.289.1 Macro Definition Documentation	826
7.289.2 Typedef Documentation	827
7.289.3 Enumeration Type Documentation	829
7.290sippTypesPrivate.h File Reference	832
7.290.1 Macro Definition Documentation	833
7.290.2 Typedef Documentation	834
7.290.3 Enumeration Type Documentation	834
7.290.4 Variable Documentation	835
7.291sippUtils.c File Reference	835

7.291.1 Detailed Description	836
7.291.2 Macro Definition Documentation	836
7.291.3 Function Documentation	836
7.292sLaplacian3x3Fp16ToFp16.h File Reference	837
7.292.1 Detailed Description	837
7.293sobel.h File Reference	837
7.293.1 Detailed Description	838
7.294ssd11x11.h File Reference	838
7.294.1 Detailed Description	838
7.295ssd5x5.h File Reference	838
7.295.1 Detailed Description	838
7.296ssd7x7U8ToU32.h File Reference	839
7.296.1 Detailed Description	839
7.297ssdPointLine7x7U8U32.h File Reference	839
7.297.1 Detailed Description	839
7.298startBicubic.h File Reference	840
7.298.1 Function Documentation	840
7.299statsAwbSatPixels.h File Reference	840
7.299.1 Function Documentation	840
7.300statsAwbSatPixelsU32.h File Reference	840
7.300.1 Function Documentation	841
7.301subpixelFilter.h File Reference	841
7.301.1 Detailed Description	841
7.301.2 Typedef Documentation	842
7.301.3 Function Documentation	842
7.302svuAbsdiff.c File Reference	842
7.302.1 Function Documentation	842
7.303svuAccumulateSquare.c File Reference	842
7.303.1 Function Documentation	843
7.304svuAccumulateWeighted.c File Reference	843
7.304.1 Function Documentation	843
7.305svuArithmeticAdd.c File Reference	844
7.305.1 Function Documentation	844
7.306svuArithmeticAddmask.c File Reference	844
7.306.1 Function Documentation	845
7.307svuArithmeticSub.c File Reference	845

7.307.1 Function Documentation	845
7.308svuArithmeticSubFp16ToFp16.c File Reference	845
7.308.1 Function Documentation	846
7.309svuArithmeticSubmask.c File Reference	846
7.309.1 Function Documentation	846
7.310svuAvg.c File Reference	847
7.311svuBilateral5x5.c File Reference	847
7.311.1 Function Documentation	847
7.312svuBitwiseAnd.c File Reference	848
7.312.1 Function Documentation	848
7.313svuBitwiseAndMask.c File Reference	848
7.313.1 Function Documentation	849
7.314svuBitwiseNot.c File Reference	849
7.314.1 Function Documentation	849
7.315svuBitwiseOr.c File Reference	850
7.315.1 Function Documentation	850
7.316svuBitwiseOrMask.c File Reference	850
7.316.1 Function Documentation	851
7.317svuBitwiseXor.c File Reference	851
7.317.1 Function Documentation	851
7.318svuBitwiseXorMask.c File Reference	851
7.318.1 Function Documentation	852
7.319svuBoxFilter.c File Reference	852
7.319.1 Macro Definition Documentation	852
7.319.2 Function Documentation	853
7.320svuBoxFilter11x11.c File Reference	853
7.320.1 Function Documentation	853
7.321svuBoxFilter13x13.c File Reference	854
7.321.1 Function Documentation	854
7.322svuBoxFilter15x15.c File Reference	854
7.322.1 Function Documentation	854
7.323svuBoxFilter3x3.c File Reference	855
7.323.1 Function Documentation	855
7.324svuBoxFilter5x5.c File Reference	855
7.324.1 Function Documentation	856
7.325svuBoxFilter7x7.c File Reference	856

7.325.1 Function Documentation	856
7.326svuBoxFilter9x9.c File Reference	857
7.326.1 Function Documentation	857
7.327svuCannyEdgeDetection.c File Reference	857
7.327.1 Macro Definition Documentation	858
7.327.2 Function Documentation	858
7.327.3 Variable Documentation	858
7.328svuCensusMatching16.c File Reference	858
7.328.1 Function Documentation	859
7.329svuCensusMatching32.c File Reference	859
7.329.1 Function Documentation	859
7.330svuCensusMatching64.c File Reference	860
7.330.1 Function Documentation	860
7.331svuCensusMatching65.c File Reference	860
7.331.1 Function Documentation	861
7.332svuCensusMatchingPyr.c File Reference	861
7.332.1 Function Documentation	861
7.333svuCensusMin16.c File Reference	862
7.333.1 Function Documentation	862
7.334svuCensusMin64.c File Reference	862
7.334.1 Function Documentation	862
7.335svuCensusMin65.c File Reference	863
7.335.1 Function Documentation	863
7.336svuCensusMin7.c File Reference	863
7.336.1 Function Documentation	864
7.337svuCensusTransform5x5.c File Reference	864
7.337.1 Function Documentation	864
7.338svuChannelExtract.c File Reference	865
7.338.1 Function Documentation	865
7.339svuChromaBlock.c File Reference	865
7.340svuCombDecimDemosaicAwbGains.c File Reference	866
7.340.1 Macro Definition Documentation	866
7.340.2 Function Documentation	866
7.341svuContrast.c File Reference	867
7.342svuConv11x11.c File Reference	867
7.342.1 Function Documentation	867

7.343svuConv15x1.c File Reference	868
7.343.1 Function Documentation	868
7.344svuConv1x15.c File Reference	868
7.344.1 Function Documentation	868
7.345svuConv1x5.c File Reference	869
7.345.1 Function Documentation	869
7.346svuConv1x5Fp16ToFp16.c File Reference	869
7.346.1 Function Documentation	870
7.347svuConv1x7.c File Reference	870
7.347.1 Function Documentation	870
7.348svuConv1x7Fp16ToFp16.c File Reference	870
7.348.1 Function Documentation	871
7.349svuConv1x9.c File Reference	871
7.349.1 Function Documentation	871
7.350svuConv3x3.c File Reference	872
7.350.1 Function Documentation	872
7.351svuConv3x3Fp16ToFp16.c File Reference	872
7.351.1 Function Documentation	872
7.352svuConv5x1.c File Reference	873
7.352.1 Function Documentation	873
7.353svuConv5x1Fp16ToFp16.c File Reference	873
7.353.1 Function Documentation	874
7.354svuConv5x5.c File Reference	874
7.354.1 Function Documentation	874
7.355svuConv5x5Fp16ToFp16.c File Reference	874
7.355.1 Function Documentation	875
7.356svuConv7x1.c File Reference	875
7.356.1 Function Documentation	875
7.357svuConv7x1Fp16ToFp16.c File Reference	876
7.357.1 Function Documentation	876
7.358svuConv7x7.c File Reference	876
7.358.1 Function Documentation	876
7.359svuConv7x7Fp16ToFp16.c File Reference	877
7.359.1 Function Documentation	877
7.360svuConv7x7Fp16ToU8.c File Reference	877
7.360.1 Function Documentation	878

7.361svuConv9x1.c File Reference	878
7.361.1 Function Documentation	878
7.362svuConv9x9.c File Reference	878
7.362.1 Function Documentation	879
7.363svuConv9x9Fp16ToFp16.c File Reference	879
7.363.1 Function Documentation	879
7.364svuConvert16bppTo8bpp.c File Reference	880
7.364.1 Macro Definition Documentation	880
7.365svuConvertF16ToU8.c File Reference	880
7.365.1 Function Documentation	880
7.366svuConvertFrom12BppTo8Bpp.c File Reference	881
7.366.1 Function Documentation	881
7.367svuConvertPFp16U16.c File Reference	881
7.367.1 Macro Definition Documentation	882
7.368svuConvertPU16Fp16.c File Reference	882
7.368.1 Macro Definition Documentation	882
7.369svuConvertU8ToF16.c File Reference	882
7.369.1 Function Documentation	882
7.370svuConvertYUV400ToYUV422.c File Reference	883
7.370.1 Function Documentation	883
7.371svuConvGeneric.c File Reference	883
7.371.1 Function Documentation	884
7.372svuConvSeparable11x11.c File Reference	884
7.372.1 Function Documentation	884
7.373svuConvSeparable11x11Fp16ToFp16.c File Reference	884
7.373.1 Function Documentation	885
7.374svuConvSeparable3x3.c File Reference	885
7.374.1 Function Documentation	885
7.375svuConvSeparable3x3Fp16ToFp16.c File Reference	885
7.375.1 Function Documentation	886
7.376svuConvSeparable5x5.c File Reference	886
7.376.1 Function Documentation	886
7.377svuConvSeparable5x5Fp16ToFp16.c File Reference	887
7.377.1 Function Documentation	887
7.378svuConvSeparable7x7.c File Reference	887
7.378.1 Function Documentation	888

7.379svuConvSeparable7x7Fp16ToFp16.c File Reference	888
7.379.1 Function Documentation	888
7.380svuConvSeparable9x9.c File Reference	888
7.380.1 Function Documentation	889
7.381svuConvSeparable9x9Fp16ToFp16.c File Reference	889
7.381.1 Function Documentation	889
7.382svuConvYuv444.c File Reference	890
7.382.1 Macro Definition Documentation	890
7.383svuCopy.c File Reference	890
7.384svuCornerMinEigenVal.c File Reference	890
7.384.1 Macro Definition Documentation	891
7.384.2 Function Documentation	891
7.384.3 Variable Documentation	891
7.385svuCornerMinEigenValpatched.c File Reference	892
7.385.1 Function Documentation	892
7.386svuCrop.c File Reference	892
7.387svuCropCvtPlaneMode.c File Reference	892
7.387.1 Function Documentation	893
7.388svuCvtColorChromaYUVToNV12.c File Reference	893
7.388.1 Function Documentation	893
7.389svucvtColorNV21toRGB.c File Reference	894
7.389.1 Function Documentation	894
7.390svuCvtColorRGBfp16ToLumaU8.c File Reference	895
7.390.1 Function Documentation	895
7.391svuCvtColorRGBfp16ToUV420U8.c File Reference	895
7.391.1 Function Documentation	895
7.392svuCvtColorRGBtoChromaNV12.c File Reference	896
7.392.1 Function Documentation	896
7.393svuCvtColorRGBtoLuma.c File Reference	896
7.393.1 Function Documentation	897
7.394svuCvtColorRGBtoLumaNV12.c File Reference	897
7.394.1 Function Documentation	897
7.395svuCvtColorRGBtoUV.c File Reference	898
7.395.1 Function Documentation	898
7.396svuCvtColorRGBtoUV420.c File Reference	898
7.396.1 Function Documentation	898

7.397svuCvtColorRGBToYUV422.c File Reference	900
7.397.1 Function Documentation	900
7.398svuCvtColorYUV422ToRGB.c File Reference	900
7.398.1 Function Documentation	901
7.399svuCvtColorYUVToRGB.c File Reference	901
7.399.1 Function Documentation	901
7.400svuDilate3x3.c File Reference	901
7.400.1 Function Documentation	902
7.401svuDilate5x5.c File Reference	902
7.401.1 Function Documentation	902
7.402svuDilate7x7.c File Reference	902
7.402.1 Function Documentation	903
7.403svuDilateGeneric.c File Reference	903
7.403.1 Function Documentation	903
7.404svudisp2depth.c File Reference	903
7.404.1 Function Documentation	904
7.405svuEqualizeHist.c File Reference	904
7.405.1 Function Documentation	904
7.406svuErode3x3.c File Reference	904
7.406.1 Function Documentation	905
7.407svuErode5x5.c File Reference	905
7.407.1 Function Documentation	905
7.408svuErode7x7.c File Reference	905
7.408.1 Function Documentation	906
7.409svuExtAfStats.c File Reference	906
7.409.1 Macro Definition Documentation	907
7.410svuExtStatsSatPixelsU32.c File Reference	908
7.410.1 Macro Definition Documentation	908
7.410.2 Function Documentation	909
7.411svuFast9M2.c File Reference	909
7.411.1 Macro Definition Documentation	909
7.411.2 Function Documentation	909
7.412svuFast9ScoreCv.c File Reference	910
7.412.1 Macro Definition Documentation	911
7.412.2 Function Documentation	911
7.412.3 Variable Documentation	912

7.413svuGauss.c File Reference	912
7.413.1 Function Documentation	912
7.414svuGaussHx2.c File Reference	913
7.414.1 Function Documentation	913
7.415svuGaussHx2_fp16.c File Reference	913
7.415.1 Function Documentation	914
7.416svuGaussVx2.c File Reference	914
7.416.1 Function Documentation	914
7.417svuGaussVx2_fp16.c File Reference	914
7.417.1 Function Documentation	915
7.418svuGenChroma.c File Reference	915
7.418.1 Function Documentation	915
7.419svuGenChromaSS.c File Reference	915
7.419.1 Function Documentation	916
7.420svuGenDnsRef.c File Reference	916
7.420.1 Function Documentation	916
7.421svuGenDnsRefFp16.c File Reference	916
7.421.1 Function Documentation	917
7.422svuGenLuma.c File Reference	917
7.422.1 Function Documentation	917
7.423svuGenLumaU8Fp16.c File Reference	917
7.423.1 Function Documentation	917
7.424svuGreyDesat.c File Reference	918
7.424.1 Function Documentation	918
7.425svuHammingDistance.c File Reference	918
7.425.1 Function Documentation	918
7.426svuHarrisResponse.c File Reference	919
7.426.1 Macro Definition Documentation	919
7.426.2 Function Documentation	919
7.427svuHistogram.c File Reference	920
7.427.1 Function Documentation	920
7.428svuHistogramStat.c File Reference	920
7.428.1 Function Documentation	921
7.429svuHomography.c File Reference	921
7.429.1 Function Documentation	921
7.430svuIntegralImageSqSumF32M2.c File Reference	922

7.430.1 Macro Definition Documentation	922
7.430.2 Function Documentation	922
7.431svuIntegralImageSqSumU32M2.c File Reference	922
7.431.1 Macro Definition Documentation	923
7.431.2 Function Documentation	923
7.432svuIntegralImageSumF32M2.c File Reference	923
7.432.1 Macro Definition Documentation	924
7.432.2 Function Documentation	924
7.433svuIntegralImageSumU16U32.c File Reference	924
7.433.1 Macro Definition Documentation	924
7.433.2 Function Documentation	924
7.434svuIntegralImageSumU32M2.c File Reference	926
7.434.1 Macro Definition Documentation	926
7.434.2 Function Documentation	926
7.435svuInterpolatePixelBilinear.c File Reference	927
7.435.1 Function Documentation	927
7.436svuLaplacian3x3.c File Reference	927
7.436.1 Function Documentation	928
7.437svuLaplacian5x5.c File Reference	929
7.437.1 Function Documentation	929
7.438svuLaplacian5x5Fp16ToFp16.c File Reference	929
7.438.1 Function Documentation	930
7.439svuLaplacian7x7.c File Reference	930
7.439.1 Function Documentation	930
7.440svuLaplacian7x7Fp16ToFp16.c File Reference	930
7.440.1 Function Documentation	931
7.441svuLocalMaxMin3x3_fp16.c File Reference	931
7.441.1 Function Documentation	931
7.442svuLocalTM.c File Reference	932
7.442.1 Function Documentation	932
7.443svuLowLvlCorr.c File Reference	932
7.443.1 Macro Definition Documentation	933
7.444svuLumaBlur.c File Reference	933
7.444.1 Function Documentation	933
7.445svuLut10to16.c File Reference	933
7.445.1 Function Documentation	934

7.446svuLut10to8.c File Reference	935
7.446.1 Function Documentation	935
7.447svuLut12to16.c File Reference	936
7.447.1 Function Documentation	936
7.448svuLut12to8.c File Reference	936
7.448.1 Function Documentation	937
7.449svuLut8to8.c File Reference	937
7.449.1 Function Documentation	937
7.450svuLutP10BppU16inU8out.c File Reference	938
7.450.1 Macro Definition Documentation	938
7.451svuMaxTest3x3_fp16.c File Reference	938
7.451.1 Function Documentation	939
7.452svuMeanStdDev.c File Reference	939
7.452.1 Function Documentation	939
7.453svuMinMaxPos.c File Reference	940
7.453.1 Function Documentation	940
7.454svuMinMaxValue.c File Reference	941
7.454.1 Function Documentation	941
7.455svuMinTest3x3_fp16.c File Reference	941
7.455.1 Function Documentation	942
7.456svuMixMedian.c File Reference	942
7.456.1 Function Documentation	942
7.457svuMonoImbalance.c File Reference	943
7.457.1 Function Documentation	943
7.458svuNegative.c File Reference	943
7.458.1 Detailed Description	944
7.458.2 Function Documentation	944
7.459svuNonMax3x3Fp32.c File Reference	944
7.459.1 Function Documentation	944
7.460svuNonMax3x3U8.c File Reference	945
7.460.1 Function Documentation	945
7.461svuPadBayer5.c File Reference	945
7.461.1 Function Documentation	946
7.462svuPadBayer5Frame.c File Reference	946
7.462.1 Function Documentation	946
7.463svuPixelPacker10b.c File Reference	946

7.463.1 Function Documentation	947
7.464svuPixelUnpacker.c File Reference	947
7.464.1 Function Documentation	947
7.465svuPixelUnpackerMipi10b.c File Reference	947
7.465.1 Function Documentation	948
7.466svuPixelUnpackerWB.c File Reference	948
7.466.1 Function Documentation	948
7.467svuPositionKernel.c File Reference	949
7.467.1 Function Documentation	949
7.468svuPurpleFlare.c File Reference	950
7.468.1 Macro Definition Documentation	950
7.468.2 Function Documentation	950
7.469svuPyrDown.c File Reference	950
7.469.1 Function Documentation	951
7.470svuRandNoise.c File Reference	951
7.470.1 Macro Definition Documentation	951
7.470.2 Function Documentation	951
7.470.3 Variable Documentation	952
7.471svuRandNoiseFp16.c File Reference	952
7.471.1 Function Documentation	952
7.471.2 Variable Documentation	953
7.472svuSAD11x11.c File Reference	953
7.472.1 Function Documentation	954
7.473svuSAD5x5.c File Reference	954
7.473.1 Function Documentation	954
7.474svuScale05BilinHV.c File Reference	954
7.474.1 Function Documentation	955
7.475svuScale05BilinHV_Fp16U8.c File Reference	955
7.475.1 Function Documentation	955
7.476svuScale05BilinHVFp16.c File Reference	956
7.476.1 Function Documentation	956
7.477svuScale05Lanc6HV.c File Reference	956
7.477.1 Function Documentation	957
7.478svuScale05Lanc7HV.c File Reference	957
7.478.1 Function Documentation	957
7.479svuScale2xBilinHV.c File Reference	957

7.480svuScale2xBilinHV_025_075_Fp16ToFp16.c File Reference	958
7.480.1 Function Documentation	958
7.481svuScale2xBilinHV_025_075_U16ToU16.c File Reference	958
7.481.1 Function Documentation	959
7.482svuScale2xBilinHV_Fp16U8_phase025_075.c File Reference	959
7.482.1 Function Documentation	959
7.483svuScale2xBilinHV_U8ToU8_phase025_075.c File Reference	960
7.483.1 Function Documentation	960
7.484svuScale2xLancH.c File Reference	960
7.485svuScale2xLancHV.c File Reference	960
7.485.1 Macro Definition Documentation	961
7.485.2 Function Documentation	961
7.485.3 Variable Documentation	961
7.486svuScale2xLancV.c File Reference	962
7.487svuScaleBilinArb.c File Reference	962
7.488svuScharr_fp16.c File Reference	962
7.488.1 Detailed Description	962
7.488.2 Function Documentation	963
7.489svusLaplacian3x3Fp16ToFp16.c File Reference	963
7.489.1 Function Documentation	963
7.490svuSobel.c File Reference	963
7.490.1 Detailed Description	964
7.490.2 Function Documentation	964
7.491svuSSD11x11.c File Reference	964
7.491.1 Function Documentation	964
7.492svuSSD5x5.c File Reference	965
7.492.1 Function Documentation	965
7.493svuSSD7x7U8ToU32.c File Reference	965
7.493.1 Function Documentation	966
7.494svuSsdPointLine7x7U8U32.c File Reference	966
7.494.1 Function Documentation	966
7.495svuStartBicubic.c File Reference	966
7.495.1 Macro Definition Documentation	967
7.495.2 Function Documentation	967
7.496svuStatsAwbSatPixels.c File Reference	967
7.496.1 Macro Definition Documentation	967

7.496.2 Function Documentation	968
7.497svuStatsAwbSatPixelsU32.c File Reference	968
7.497.1 Function Documentation	968
7.498svuSubpixelFilter.c File Reference	968
7.498.1 Function Documentation	969
7.499svuThreshold.c File Reference	969
7.499.1 Function Documentation	969
7.500svuThresholdBinaryRange.c File Reference	970
7.500.1 Function Documentation	970
7.501svuThresholdBinaryU8.c File Reference	970
7.501.1 Function Documentation	971
7.502svuThresholdFilter.c File Reference	971
7.502.1 Function Documentation	971
7.503svuUndistortBrown.c File Reference	971
7.503.1 Function Documentation	972
7.504svuWhiteBalanceBayerGBRG.c File Reference	972
7.504.1 Function Documentation	972
7.505svuWhiteBalanceRGB.c File Reference	973
7.505.1 Function Documentation	973
7.506svuXYgen.c File Reference	974
7.506.1 Function Documentation	974
7.507threshold.h File Reference	974
7.507.1 Detailed Description	975
7.508thresholdBinaryRange.h File Reference	975
7.508.1 Detailed Description	975
7.509thresholdBinaryU8.h File Reference	975
7.509.1 Detailed Description	976
7.510thresholdFilter.h File Reference	976
7.510.1 Detailed Description	976
7.511tripleConv3x3.c File Reference	976
7.511.1 Function Documentation	977
7.512tripleConv3x3.h File Reference	977
7.512.1 Function Documentation	977
7.513undistortBrown.h File Reference	977
7.513.1 Detailed Description	977
7.514whiteBalanceBayerGBRG.h File Reference	978

- 7.514.1 Detailed Description 978
- 7.515whiteBalanceRGB.h File Reference 978
 - 7.515.1 Detailed Description 979
- 7.516wrapperSem.cpp File Reference 979
- 7.517wrapperSem.cpp File Reference 979
- 7.518wrapperSem.h File Reference 979
- 7.519wrapperSem.h File Reference 979
- 7.520xyGen.h File Reference 979
 - 7.520.1 Detailed Description 980

Index **981**

Chapter 1

Introduction

The document describes the Movidius Streaming Image Processing Pipeline (SIPP) Framework API and the related SIPP filters. For the general description of the SIPP Framework please read the MDK Programmer's Guide.

Chapter 2

Module Index

2.1 Modules

Here is a list of all modules:

Arithmetic	121
12Bpp to 8Bpp conversion	185
Absolute difference	115
Accumulate Square	116
Accumulate Weighted	117
Arbitrary Downscale	311
Arithmetic addition	118
Arithmetic addition with mask	119
Arithmetic subtraction	120
Arithmetic subtraction fp16	127
Arithmetic subtraction with mask	128
Average	129
Bitwise And	131
Bitwise And with mask	132
Bitwise Not	133
Bitwise Or	134
Bitwise Or with mask	135
Bitwise Xor	136
Bitwise Xor with mask	137
CensusTransform5x5	157
Convert 16bpp To 8bpp	183
Convert F16 To U8	184
Convert Fp16 to U16	186
Convert U16 to Fp16	187
Convert U8 To F16	188
Convolution 11x11	161
Convolution 15x1	162
Convolution 1x15	163
Convolution 1x5	164
Convolution 1x5 Fp16ToFp16	165
Convolution 1x7	166
Convolution 1x7 Fp16ToFp16	167
Convolution 1x9	168

Convolution 3x3	169
Convolution 3x3 Fp16ToFp16	170
Convolution 5x1	171
Convolution 5x1 Fp16ToFp16	172
Convolution 5x5	173
Convolution 5x5 Fp16ToFp16	174
Convolution 7x1	175
Convolution 7x1 Fp16ToFp16	176
Convolution 7x7	177
Convolution 7x7 Fp16ToFp16	178
Convolution 7x7 Fp16ToU8	179
Convolution 9x1	180
Convolution 9x9	181
Convolution 9x9 Fp16ToFp16	182
Convolution Separable 11x11	191
Convolution Separable 11x11 Fp16ToFp16	192
Convolution Separable 3x3	193
Convolution Separable 3x3 Fp16ToFp16	194
Convolution Separable 5x5	195
Convolution Separable 5x5 Fp16ToFp16	196
Convolution Separable 7x7	197
Convolution Separable 7x7 Fp16ToFp16	198
Convolution Separable 9x9	199
Convolution Separable 9x9 Fp16ToFp16	200
Copy	202
Downscale by 2	298
Downscale by 2 (fp16/fp16)	300
Downscale by 2 (fp16/u8)	299
Gauss Blur	236
GaussHx2	237
GaussHx2_fp16	238
GaussVx2	239
GaussVx2_fp16	240
Generic Convolution	190
Lanczos Downscale by 2 (6 taps)	301
Lanczos Downscale by 2 (7 taps)	302
Lanczos Horizontal Upscale by 2	308
Lanczos Upscale by 2	309
Lanczos Vertical Upscale by 2	310
Laplacian 3x3	259
Laplacian 3x3 Fp16 To Fp16	313
Laplacian 5x5	260
Laplacian 5x5 Fp16 To Fp16	261
Laplacian 7x7	262
Laplacian 7x7 Fp16 To Fp16	263
Luma Blur	269
MonoImbalance	282
Negative	283
Pyramid Downscale	293
Random Noise	294
Random Noise (high speed)	295

Sum of Absolute Differences 11x11	296
Sum of Absolute Differences 5x5	297
Sum of Squared Differences 11x11	315
Sum of Squared Differences 5x5	316
Sum of Squared Differences 7x7	318
Sum of Squared Differences 7x7 (U8 to U32)	317
Upscale by 2	303
XY Generator	327
bilateral5x5	130
censusMatching16	148
censusMatching32	149
censusMatching64	150
censusMatching65	151
censusMatchingPyr	152
censusMin16	153
censusMin64	154
censusMin65	155
censusMin7	156
channelExtract	158
hammingDistance	248
localMaxMin3x3_fp16	264
maxTest3x3_fp16	276
meanStdDev	277
minTest3x3_fp16	280
nonMax3x3U8	285
nonMaxFp32	284
CV	206
Box Filter 11x11	140
Box Filter 13x13	141
Box Filter 15x15	142
Box Filter 3x3	143
Box Filter 5x5	144
Box Filter 7x7	145
Box Filter 9x9	146
Canny Edge Detection	147
Corner Min Eigenvalue	203
Corner Min Eigenvalue Patched	204
Crop	205
Dilate 3x3	225
Dilate 5x5	226
Dilate 7x7	227
Erode 3x3	230
Erode 5x5	231
Erode 7x7	232
Fast9M2	234
Fast9ScoreCv	235
Generic Box Filter	138
Generic Dilate	228
Homography	252
Integral Image Square Sum(U32)	254
Integral Image Square Sum(f32)	253

Integral Image Sum(U16toU32)	256
Integral Image Sum(U32)	257
Integral Image Sum(f32)	255
Min/Max Value	279
Min/Max Value Position	278
Pixel Position	291
Sobel	314
Threshold	319
Threshold Binary Range	321
Threshold Binary U8	322
ThresholdFilter	323
harrisResponse	249
interpolatePixelBilinear	258
scharr_fp16	312
Chroma Denoise	102
Chroma Generation	111
Color Combination	104
Convolution	105
DMA	328
Debayer	99
DoG LTM	112
Edge operator	109
Harris Corner Detector	106
ISP	215
AF Stats	233
Chroma Block	159
Contrast	160
Convert to YUV444	201
Equalize Histogram	229
Generate Chroma	241
Generate Chroma with subsampling	242
Generate Luma U8 to Fp16	246
Generate Reference for Luma Denoise	243
Generate Reference for Luma Denoise(fp16 input)	244
Histogram	250
LUT 10 to 16	270
LUT 10 to 8	271
LUT 12 to 16	272
LUT 12 to 8	273
LUT 16 to 8	275
LUT 8 to 8	274
Low Level Correction	266
Low Level Correction on Multiple Planes	268
Luma Blur	245
Mix Median	281
NV21 to RGB conversion	210
Pixel Unpacker	288
Pixel Unpacker Mipi 10b	289
Pixel Unpacker WB	290
Pixel packer	287
RGB to Chroma NV12 conversion	213

RGB to Luma NV12 conversion	219
RGB to Luma conversion	214
RGB to UV conversion	220
RGB to UV420 conversion	221
RGB to YUV422 conversion	222
RGB(fp16) to Luma(u8) conversion	211
RGB(fp16) to UV420(u8) conversion	212
Undistort	324
Upscale by 2 with phases 0.25 and 0.75 fp16 to fp16	304
Upscale by 2 with phases 0.25 and 0.75 fp16 to u8	306
Upscale by 2 with phases 0.25 and 0.75 u16 to u16	305
Upscale by 2 with phases 0.25 and 0.75 u8 to u8	307
White Balance Bayer GBRG	325
White Balance RGB	326
YUV to NV12 chroma conversion	209
YUV to RGB conversion	224
YUV400 to YUV422 conversion	189
YUV422 to RGB conversion	223
greyDesat	247
histogramStat	251
localTM	265
padBayer5Frame	286
purpleFlare	292
Lens Shading Correction	97
Look-up table	103
Luma Denoise	101
MIPI Rx	113
MIPI Tx	114
Median	96
Polyphase FIR Scaler	107
Raw	98
Sharpen	100
Sigma Denoise	110

Chapter 3

Data Structure Index

3.1 Data Structures

Here are the data structures with brief descriptions:

AccumulateWeightedParam	
Parameter structure of the Accumulate Weighted filter	329
ae_patch_stats	329
AeAwbStatsCfg	330
AF_paxel_statistics	331
Bilateral5x5Param	331
BoxFilter11x11Param	
Parameter structure of the Box Filter 11x11 filter	332
BoxFilter13x13Param	
Parameter structure of the Box Filter 13x13 filter	332
BoxFilter15x15Param	
Parameter structure of the Box Filter 15x15 filter	333
BoxFilter3x3Param	
Parameter structure of the Box Filter 3x3 filter	333
BoxFilter5x5Param	
Parameter structure of the Box Filter 5x5 filter	334
BoxFilter7x7Param	
Parameter structure of the Box Filter 7x7 filter	334
BoxFilter9x9Param	
Parameter structure of the Box Filter 9x9 filter	335
BoxFilterParam	
Parameter structure of the Generic Box Filter filter	335
cannyEdgeDetectionParam	
Parameter structure of the Canny Edge Detection filter	336
CensusMatching32Param	
Parameter structure of the censusMatching32 filter	337
CensusMatching64Param	
Parameter structure of the censusMatching64 filter	337
CensusMatchingPyrParam	
Parameter structure of the censusMatchingPyr filter	338
ChannelExtractParam	
Parameter structure of the channelExtract filter	338

ChrDnsParam	Parameter structure of the chormadns filter	339
ChrGenParam	Parameter structure of the filter	340
ChrGenSSParam	Parameter structure of the Generate Chroma with subsampling filter	340
ChromaBlkParam	Parameter structure of the Chroma Block filter	341
cmxRegUsage	341
ColCombParam	Parameter structure of the colorcomb filter	342
CombDecimAwbGainsParam	343
CombDecimStatsGainsParam	343
CommInfo	344
ContrastParam	Parameter structure of the Contrast filter	345
Conv11x11Param	Parameter structure of the Convolution 11x11 filter	346
Conv15x1Param	Parameter structure of the Convolution 15x1 filter	347
Conv1x15Param	Parameter structure of the Convolution 1x15 filter	347
Conv1x5Fp16ToFp16Param	Parameter structure of the Convolution 1x5 Fp16ToFp16 filter	348
Conv1x5Param	Parameter structure of the Convolution 1x5 filter	349
Conv1x7Fp16ToFp16Param	Parameter structure of the Convolution 1x7 Fp16ToFp16 filter	349
Conv1x7Param	Parameter structure of the Convolution 1x7 filter	350
Conv1x9Param	Parameter structure of the Convolution 1x9 filter	350
Conv3x3Fp16ToFp16Param	Parameter structure of the Convolution 3x3 Fp16ToFp16 filter	351
Conv3x3Param	Parameter structure of the Convolution 3x3 filter	352
Conv5x1Fp16ToFp16Param	Parameter structure of the Convolution 5x1 Fp16ToFp16 filter	352
Conv5x1Param	Parameter structure of the Convolution 5x1 filter	353
Conv5x5Fp16ToFp16Param	Parameter structure of the Convolution 5x5 Fp16ToFp16 filter	354
Conv5x5Param	Parameter structure of the Convolution 5x5 filter	354
Conv7x1Fp16ToFp16Param	Parameter structure of the Convolution 7x1 Fp16ToFp16 filter	355
Conv7x1Param	Parameter structure of the Convolution 7x1 filter	355
Conv7x7Param	Parameter structure of the Convolution 7x7 filter	356

Conv7x7ParamFp16ToFp16	
Parameter structure of the Convolution 7x7 Fp16ToFp16 filter	357
Conv7x7ParamFp16ToU8	
Parameter structure of the Convolution 7x7 Fp16ToU8 filter	357
Conv9x1Param	
Parameter structure of the Convolution 9x1 filter	358
Conv9x9Fp16ToFp16Param	
Parameter structure of the Convolution 9x9 Fp16ToFp16 filter	359
Conv9x9Param	
Parameter structure of the Convolution 9x9 filter	359
ConvGenericParam	
Parameter structure of the Generic Convolution filter	360
ConvParam	
Parameter structure of the convolution filter	361
ConvSeparable11x11Fp16ToFp16Param	
Parameter structure of the Convolution Separable 11x11 Fp16ToFp16 filter	362
ConvSeparable11x11Param	
Parameter structure of the Convolution Separable 11x11 filter	362
ConvSeparable3x3Fp16ToFp16Param	
Parameter structure of the Convolution Separable 3x3 Fp16ToFp16 filter	363
ConvSeparable3x3Param	
Parameter structure of the Convolution Separable 3x3 filter	363
ConvSeparable5x5Fp16ToFp16Param	
Parameter structure of the Convolution Separable 5x5 Fp16ToFp16 filter	364
ConvSeparable5x5Param	
Parameter structure of the Convolution Separable 5x5 filter	364
ConvSeparable7x7Fp16ToFp16Param	
Parameter structure of the Convolution Separable 7x7 Fp16ToFp16 filter	365
ConvSeparable7x7Param	
Parameter structure of the Convolution Separable 7x7 filter	365
ConvSeparable9x9Fp16ToFp16Param	
Parameter structure of the Convolution Separable 9x9 Fp16ToFp16 filter	366
ConvSeparable9x9Param	
Parameter structure of the Convolution Separable 9x9 filter	366
CropParam	
Parameter structure of the Crop filter	367
cvtColorChromaNV12Param	
Parameter structure of the RGB to Chroma NV12 conversion filter	367
CvtColorChromaYUVToNV12Param	
Parameter structure of the YUV to NV12 chroma conversion filter	368
cvtColorLumaNV12Param	
Parameter structure of the RGB to Luma NV12 conversion filter	368
DbyrParam	
Parameter structure of the debayer filter	369
Dilate3x3Param	
Parameter structure of the Dilate 3x3 filter	370
Dilate5x5Param	
Parameter structure of the Dilate 5x5 filter	371
Dilate7x7Param	
Parameter structure of the Dilate 7x7 filter	371

DilateGenericParam	
Parameter structure of the Generic Dilate filter	372
Disp2depthParam	
Parameter structure of the Threshold filter	372
DmaDesc	373
DMAExtCfg	374
DmaParam	
Parameter structure of the DMA filter	374
DMAPartialCfg	376
DmaTaskList	376
DogLtmParam	377
EdgeParam	
Parameter structure of the edgeoperator filter	377
EqualizeHistParam	
Parameter structure of the Equalize Histogram filter	378
Erode3x3Param	
Parameter structure of the Erode 3x3 filter	379
Erode5x5Param	
Parameter structure of the Erode 5x5 filter	379
Erode7x7Param	
Parameter structure of the Erode 7x7 filter	380
ExtStatsSatPixelsU32Param	380
Fast9M2Param	
Parameter structure of the Fast9M2 filter	382
Fast9ScoreCvParam	
Parameter structure of the Fast9ScoreCv filter	383
GenChrParam	
Parameter structure of the edgeoperator filter	384
GenLumaU8Fp16Param	
Shave function of the Generate Luma U8 to Fp16 filter	385
GreyDesatParam	
Parameter structure of the greyDesat filter	385
HammingDistanceParam	
Parameter structure of the hammingDistance filter	386
HarrisParam	
Parameter structure of the harrisorners filter	386
HarrisSwParam	
Parameter structure of the harrisResponse filter	387
HistogramParam	
Parameter structure of the Histogram filter	387
HistogramStatParam	
Parameter structure of the histogramStat filter	388
HomographyParam	
Parameter structure of the Homography filter	388
HorizPaddingOffS	389
HPadInfoS	389
InterpolatePixelBilinearParam	
Parameter structure of the interpolatePixelBilinear filter	390
LocalTMPParam	
Parameter structure of the localTM filter	391

LowLvlCorrNPiParam	
Parameter structure of the Low Level Correction on Multiple Planes filter	391
LowLvlCorrParam	
Parameter structure of the Low Level Correction filter	392
LscParam	
Parameter structure of the lsc filter	392
Lut10to16Param	
Parameter structure of the LUT 10 to 16 filter	394
Lut10to8Param	
Parameter structure of the LUT 10 to 8 filter	394
Lut12to16Param	
Parameter structure of the LUT 12 to 16 filter	395
Lut12to8Param	
Parameter structure of the LUT 12 to 8 filter	395
Lut8to8Param	
Parameter structure of the LUT 8 to 8 filter	396
LutParam	
Parameter structure of the lut filter	396
MaxTest3x3fp16Param	
Parameter structure of the maxTest3x3_fp16 filter	397
MedParam	
Parameter structure of the median filter	398
memRegDescriptor	399
MinMaxPosParam	
Parameter structure of the Min/Max Value Position filter	399
minMaxValParam	
Parameter structure of the Min/Max Value filter	400
MinTest3x3fp16Param	
Parameter structure of the minTest3x3_fp16 filter	401
MipiRxLoopbackParam	401
MipiRxParam	
Parameter structure of the MIPI Rx filter	402
MipiTxLoopbackParam	404
MipiTxParam	
Parameter structure of the MIPI Rx filter	405
MixMedianParam	
Parameter structure of the Mix Median filter	407
MonoImbalanceParam	
Parameter structure of the monoImbalance filter	408
nonMax3x3Fp32Param	409
ParentInfoS	409
PixelUnpackerMipi10bParam	
Parameter structure of the Pixel Unpacker Mipi 10b filter	409
PixelUnpackerParam	
Parameter structure of the Pixel Unpacker filter	410
PixelUnpackerWBParam	
Parameter structure of the Pixel Unpacker WB filter	411
PolyFirParam	
Parameter structure of the polyphasefir filter	411
positionKernelParam	
Parameter structure of the Pixel Position filter	414

PpAf	
Parameter structure of the AF Stats filter	415
PurpleFlareParam	
Parameter structure of the purpleFlare filter	417
RandNoiseFp16Param	
Parameter structure of the Random Noise (high speed) filter	417
RandNoiseParam	
Parameter structure of the Random Noise filter	417
RawParam	
Parameter structure of the raw filter	418
SchedInfoS	421
Semaphore	422
SigmaParam	
Parameter structure of the edgeoperator filter	423
SIPP_ACCESS_SCHEDULER	425
SIPP_ACCESS_SCHEDULER_QU	426
SIPP_ACCESS_SCHEDULER_QU_ENTRY	427
SIPP_HW_SESSION	427
SIPP_PAL_QU	429
SIPP_PAL_THREAD	429
SIPP_PIPELINE_FINALISED_DATA	430
SIPP_TRACE_FLAGS	430
SippCmxBufferMapS	430
SippFilterS	431
SippFilterSchedInfoS	442
SippGlobals	443
SippHeapCB	443
SippHwBufS	444
SippHwIOBuf	445
SippManagedBufSchedInfo	446
SippMemRegion	446
SippMemRegionListNode	447
sippOpipBufInfo	448
SippOseS	448
SippPipelineS	449
SippPixelChunkPos	460
SippSchEntS	460
SippVPhysMapS	464
sSchedIBufUsageInfoS	464
sSchLineBufferS	466
sSippCdmaQuEntryS	467
sSippCdmaQuS	468
sSippCMDQuEntryS	468
sSippCMDQuS	469
StartBicubicParam	469
StatsAwbSatPixelsParam	470
StatsAwbSatPixelsParamU32	472
SubpixelFilterParam	
Parameter structure of the Threshold filter	474
ThresholdBinaryRangeParam	
Parameter structure of the Threshold Binary Range filter	475

ThresholdBinaryU8Param	
Parameter structure of the Threshold Binary U8 filter	475
ThresholdFilterParam	
Parameter structure of the ThresholdFilter filter	476
ThresholdParam	
Parameter structure of the Threshold filter	476
tMLPISStartCQCtrl	477
TripleConv3x3	478
tRTStats	478
tSippFramework	479
tSippMCB	480
tSippPhysicalPool	480
tSippPipelineSuper	481
tsSippHeap	482
UndistortBParam	
Parameter structure of the Undistort filter	483
UnitInfo	484
UsmParam	
Parameter structure of the sharpen filter	484
WhiteBalanceBayerGBRGParam	
Parameter structure of the White Balance Bayer GBRG filter	486
WhiteBalanceRGBParam	
Parameter structure of the White Balance RGB filter	486
XYGenParam	
Parameter structure of the XY Generator filter	487
YDnsParam	
Parameter structure of the ydns filter	488
YDnsRefFp16Param	
Parameter structure of the Generate Reference for Luma Denoise(fp16 input) filter .	489
YDnsRefLut10bppParam	
Parameter structure of the LUT 16 to 8 filter	490
YDnsRefParam	
Parameter structure of the Generate Reference for Luma Denoise filter	491

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

absdiff.h	This file contains the declaration of the Absolute difference SIPP filter API	493
accumulateSquare.h	This file contains the declaration of the Accumulate Square SIPP filter API	493
accumulateWeighted.h	This file contains the declaration of the Accumulate Weighted SIPP filter API	494
arithmeticAdd.h	This file contains the declaration of the Arithmetic addition SIPP filter API	494
arithmeticAddmask.h	This file contains the declaration of the Arithmetic addition with mask SIPP filter API	495
arithmeticSub.h	This file contains the declaration of the Arithmetic subtraction SIPP filter API	495
arithmeticSubFp16ToFp16.h	This file contains the declaration of the Arithmetic subtraction fp16 SIPP filter API .	496
arithmeticSubmask.h	This file contains the declaration of the Arithmetic subtraction with mask SIPP filter API	496
avg.h	This file contains the declaration of the Average SIPP filter API	497
bilateral5x5.h	This file contains the declaration of the bilateral5x5 SIPP filter API	497
bitwiseAnd.h	This file contains the declaration of the Bitwise And SIPP filter API	498
bitwiseAndMask.h	This file contains the declaration of the Bitwise And with mask SIPP filter API	498
bitwiseNot.h	This file contains the declaration of the Bitwise Not SIPP filter API	498
bitwiseOr.h	This file contains the declaration of the Bitwise Or SIPP filter API	499
bitwiseOrMask.h	This file contains the declaration of the Bitwise Or with mask SIPP filter API	499
bitwiseXor.h	This file contains the declaration of the Bitwise Xor SIPP filter API	500

bitwiseXorMask.h	
	This file contains the declaration of the Bitwise Xor with mask SIPP filter API 500
boxFilter.h	
	This file contains the declaration of the Generic Box Filter SIPP filter API 501
boxFilter11x11.h	
	This file contains the declaration of the Box Filter 11x11 SIPP filter API 502
boxFilter13x13.h	
	This file contains the declaration of the Box Filter 13x13 SIPP filter API 502
boxFilter15x15.h	
	This file contains the declaration of the Box Filter 15x15 SIPP filter API 503
boxFilter3x3.h	
	This file contains the declaration of the Box Filter 3x3 SIPP filter API 503
boxFilter5x5.h	
	This file contains the declaration of the Box Filter 5x5 SIPP filter API 504
boxFilter7x7.h	
	This file contains the declaration of the Box Filter 7x7 SIPP filter API 504
boxFilter9x9.h	
	This file contains the declaration of the Box Filter 9x9 SIPP filter API 505
cannyEdgeDetection.h	
	This file contains the declaration of the Canny Edge Detection SIPP filter API 506
censusMatching16.h	
	This file contains the declaration of the censusMatching16 SIPP filter API 506
censusMatching32.h	
	This file contains the declaration of the censusMatching32 SIPP filter API 507
censusMatching64.h	
	This file contains the declaration of the censusMatching64 SIPP filter API 507
censusMatching65.h	
	This file contains the declaration of the censusMatching65 SIPP filter API 508
censusMatchingPyr.h	
	This file contains the declaration of the censusMatchingPyr SIPP filter API 508
censusMin16.h	
	This file contains the declaration of the censusMin16 SIPP filter API 509
censusMin64.h	
	This file contains the declaration of the censusMin64 SIPP filter API 509
censusMin65.h	
	This file contains the declaration of the censusMin65 SIPP filter API 510
censusMin7.h	
	This file contains the declaration of the censusMin7 SIPP filter API 510
censusTransform5x5.h	
	This file contains the declaration of the CensusTransform5x5 SIPP filter API 511
channelExtract.h	
	This file contains the declaration of the channelExtract SIPP filter API 511
chromaBlock.h	
	This file contains the declaration of the Chroma Block SIPP filter API 512
combDecimDemosaicAwbGains.h 512
combDecimDemosaicAwbGainsStats.c 513
combDecimDemosaicAwbGainsStats.h 514
contrast.h	
	This file contains the declaration of the Contrast SIPP filter API 514
conv11x11.h	
	This file contains the declaration of the Convolution 11x11 SIPP filter API 515

conv15x1.h	This file contains the declaration of the Convolution 15x1 SIPP filter API	516
conv1x15.h	This file contains the declaration of the Convolution 1x15 SIPP filter API	516
conv1x5.h	This file contains the declaration of the Convolution 1x5 SIPP filter API	517
conv1x5Fp16ToFp16.h	This file contains the declaration of the Convolution 1x5 Fp16ToFp16 SIPP filter API	517
conv1x7.h	This file contains the declaration of the Convolution 1x7 SIPP filter API	518
conv1x7Fp16ToFp16.h	This file contains the declaration of the Convolution 1x7 Fp16ToFp16 SIPP filter API	518
conv1x9.h	This file contains the declaration of the Convolution 1x9 SIPP filter API	519
conv3x3.h	This file contains the declaration of the Convolution 3x3 SIPP filter API	520
conv3x3Fp16ToFp16.h	This file contains the declaration of the Convolution 3x3 Fp16ToFp16 SIPP filter API	520
conv5x1.h	This file contains the declaration of the Convolution 5x1 SIPP filter API	521
conv5x1Fp16ToFp16.h	This file contains the declaration of the Convolution 5x1 Fp16ToFp16 SIPP filter API	521
conv5x5.h	This file contains the declaration of the Convolution 5x5 SIPP filter API	522
conv5x5Fp16ToFp16.h	This file contains the declaration of the Convolution 5x5 Fp16ToFp16 SIPP filter API	522
conv7x1.h	This file contains the declaration of the Convolution 7x1 SIPP filter API	523
conv7x1Fp16ToFp16.h	This file contains the declaration of the Convolution 7x1 Fp16ToFp16 SIPP filter API	524
conv7x7.h	This file contains the declaration of the Convolution 7x7 SIPP filter API	524
conv7x7Fp16ToFp16.h	This file contains the declaration of the Convolution 7x7 Fp16ToFp16 SIPP filter API	525
conv7x7Fp16ToU8.h	This file contains the declaration of the Convolution 7x7 Fp16ToU8 SIPP filter API .	525
conv9x1.h	This file contains the declaration of the Convolution 9x1 SIPP filter API	526
conv9x9.h	This file contains the declaration of the Convolution 9x9 SIPP filter API	526
conv9x9Fp16ToFp16.h	This file contains the declaration of the Convolution 9x9 Fp16ToFp16 SIPP filter API	527
convert16bppTo8bpp.h	This file contains the declaration of the Convert 16bpp To 8bpp SIPP filter API . . .	528
convertF16ToU8.h	This file contains the declaration of the Convert F16 To U8 SIPP filter API	528
convertFrom12BppTo8Bpp.h	This file contains the declaration of the 12Bpp to 8Bpp conversion SIPP filter API .	528
convertPFp16U16.h	This file contains the declaration of the Convert Fp16 to U16 SIPP filter API	529

convertPU16Fp16.h	
	This file contains the declaration of the Convert U16 to Fp16 SIPP filter API 529
convertU8ToF16.h	
	This file contains the declaration of the Convert U8 To F16 SIPP filter API 530
convertYUV400ToYUV422.h	
	This file contains the declaration of the YUV400 to YUV422 conversion SIPP filter API 530
convGeneric.h	
	This file contains the declaration of the Generic Convolution SIPP filter API 531
convSeparable11x11.h	
	This file contains the declaration of the Convolution Separable 11x11 SIPP filter API 531
convSeparable11x11Fp16ToFp16.h	
	This file contains the declaration of the Convolution Separable 11x11 Fp16ToFp16 SIPP filter API 532
convSeparable3x3.h	
	This file contains the declaration of the Convolution Separable 3x3 SIPP filter API . 532
convSeparable3x3Fp16ToFp16.h	
	This file contains the declaration of the Convolution Separable 3x3 Fp16ToFp16 SIPP filter API 533
convSeparable5x5.h	
	This file contains the declaration of the Convolution Separable 5x5 SIPP filter API . 533
convSeparable5x5Fp16ToFp16.h	
	This file contains the declaration of the Convolution Separable 5x5 Fp16ToFp16 SIPP filter API 534
convSeparable7x7.h	
	This file contains the declaration of the Convolution Separable 7x7 SIPP filter API . 535
convSeparable7x7Fp16ToFp16.h	
	This file contains the declaration of the Convolution Separable 7x7 Fp16ToFp16 SIPP filter API 535
convSeparable9x9.h	
	This file contains the declaration of the Convolution Separable 9x9 SIPP filter API . 536
convSeparable9x9Fp16ToFp16.h	
	This file contains the declaration of the Convolution Separable 9x9 Fp16ToFp16 SIPP filter API 536
convYuv444.h	
	This file contains the declaration of the Convert to YUV444 SIPP filter API 537
copy.h	
	This file contains the declaration of the Copy SIPP filter API 537
cornerMinEigenVal.h	
	This file contains the declaration of the Corner Min Eigenvalue SIPP filter API . . . 538
cornerMinEigenValpatched.h	
	This file contains the declaration of the Corner Min Eigenvalue Patched SIPP filter API 538
crop.h	
	This file contains the declaration of the Crop SIPP filter API 539
cropCvtPlaneMode.h	
	This file contains the declaration of the CropCvtPlaneMode SIPP filter API 539
cvtColorChromaYUVToNV12.h	
	This file contains the declaration of the YUV to NV12 chroma conversion SIPP filter API 540

cvtColorNV21toRGB.h	
This file contains the declaration of the NV21 to RGB conversion SIPP filter API	541
cvtColorRGBfp16ToLumaU8.h	
This file contains the declaration of the RGB(fp16) to Luma(u8) conversion SIPP filter API	541
cvtColorRGBfp16ToUV420U8.h	
This file contains the declaration of the RGB(fp16) to UV420(u8) conversion SIPP filter API	541
cvtColorRGBtoChromaNV12.h	
This file contains the declaration of the RGB to Chroma NV12 conversion SIPP filter API	542
cvtColorRGBtoLuma.h	
This file contains the declaration of the RGB to Luma conversion SIPP filter API	542
cvtColorRGBtoLumaNV12.h	
This file contains the declaration of the RGB to Luma NV12 conversion SIPP filter API	543
cvtColorRGBtoUV.h	
This file contains the declaration of the RGB to UV conversion SIPP filter API	543
cvtColorRGBtoUV420.h	
This file contains the declaration of the RGB to UV420 conversion SIPP filter API	544
cvtColorRGBtoYUV422.h	
This file contains the declaration of the RGB to YUV422 conversion SIPP filter API	544
cvtColorYUV422ToRGB.h	
This file contains the declaration of the YUV422 to RGB conversion SIPP filter API	545
cvtColorYUVToRGB.h	
This file contains the declaration of the YUV to RGB conversion SIPP filter API	545
dilate3x3.h	
This file contains the declaration of the Dilate 3x3 SIPP filter API	546
dilate5x5.h	
This file contains the declaration of the Dilate 5x5 SIPP filter API	546
dilate7x7.h	
This file contains the declaration of the Dilate 7x7 SIPP filter API	547
dilateGeneric.h	
This file contains the declaration of the Generic Dilate SIPP filter API	547
disp2depth.h	
This file contains the declaration of the Threshold SIPP filter API	548
disp2depth_exec.c	549
disp2depth_exec.h	549
equalizeHist.h	
This file contains the declaration of the Equalize Histogram SIPP filter API	550
erode3x3.h	
This file contains the declaration of the Erode 3x3 SIPP filter API	551
erode5x5.h	
This file contains the declaration of the Erode 5x5 SIPP filter API	551
erode7x7.h	
This file contains the declaration of the Erode 7x7 SIPP filter API	552
extAfStats.h	
This file contains the declaration of the AF Stats SIPP filter API	552
extStatsSatPixelsU32.h	553
fast9M2.h	
This file contains the declaration of the Fast9M2 SIPP filter API	553

fast9ScoreCv.h	This file contains the declaration of the Fast9ScoreCv SIPP filter API	554
gauss.h	This file contains the declaration of the Gauss Blur SIPP filter API	555
gaussHx2.h	This file contains the declaration of the GaussHx2 SIPP filter API	555
gaussHx2_fp16.h	This file contains the declaration of the GaussHx2_fp16 SIPP filter API	555
gaussVx2.h	This file contains the declaration of the GaussVx2 SIPP filter API	556
gaussVx2_fp16.h	This file contains the declaration of the GaussVx2_fp16 SIPP filter API	556
genChroma.h	This file contains the declaration of the Generate Chroma SIPP filter API	557
genChromaSS.h	This file contains the declaration of the Generate Chroma with subsampling SIPP filter API	557
genDnsRef.h	This file contains the declaration of the Generate Reference for Luma Denoise SIPP filter API	558
genDnsRefFp16.h	This file contains the declaration of the Generate Reference for Luma Denoise(fp16 input) SIPP filter API	558
genLuma.h	This file contains the declaration of the Luma Blur SIPP filter API	559
genLumaU8Fp16.h	This file contains the declaration of the Generate Luma U8 to Fp16 SIPP filter API	559
greyDesat.h	This file contains the declaration of the greyDesat SIPP filter API	560
hammingDistance.h	This file contains the declaration of the censusTransform7x7 SIPP filter API	561
harrisResponse.h	This file contains the declaration of the harrisResponse SIPP filter API	561
histogram.h	This file contains the declaration of the Histogram SIPP filter API	562
histogramStat.h	This file contains the declaration of the histogramStat SIPP filter API	562
homography.h	This file contains the declaration of the Homography SIPP filter API	563
integralImageSqSumF32M2.h	This file contains the declaration of the Integral Image Square Sum(f32) SIPP filter API	563
integralImageSqSumU32M2.h	This file contains the declaration of the Integral Image Square Sum(U32) SIPP filter API	564
integralImageSumF32M2.h	This file contains the declaration of the Integral Image Sum(f32) SIPP filter API	564
integralImageSumU16U32.h	This file contains the declaration of the Integral Image Sum(U16toU32) SIPP filter API	565

integralImageSumU32M2.h	
	This file contains the declaration of the Integral Image Sum(U32) SIPP filter API . . . 565
interpolatePixelBilinear.h	
	This file contains the declaration of the interpolatePixelBilinear SIPP filter API . . . 566
laplacian3x3.h	
	This file contains the declaration of the Laplacian 3x3 SIPP filter API 566
laplacian5x5.h	
	This file contains the declaration of the Laplacian 5x5 SIPP filter API 567
laplacian5x5Fp16ToFp16.h	
	This file contains the declaration of the Laplacian 5x5 Fp16 To Fp16 SIPP filter API 567
laplacian7x7.h	
	This file contains the declaration of the Laplacian 7x7 SIPP filter API 568
laplacian7x7Fp16ToFp16.h	
	This file contains the declaration of the Laplacian 7x7 Fp16 To Fp16 SIPP filter API 568
localMaxMin3x3_fp16.h	
	This file contains the declaration of the localMaxMin3x3_fp16 SIPP filter API . . . 569
localTM.h	
	This file contains the declaration of the localTM SIPP filter API 569
lowLvlCorr.h	
	This file contains the declaration of the Low Level Correction SIPP filter API 570
lowLvlCorrMultiplePlanes.c 570
lowLvlCorrMultiplePlanes.h	
	This file contains the declaration of the Low Level Correction on Multiple Planes SIPP filter API 571
lumaBlur.h	
	This file contains the declaration of the Luma Blur SIPP filter API 571
lut10to16.h	
	This file contains the declaration of the LUT 10 to 16 SIPP filter API 572
lut10to8.h	
	This file contains the declaration of the LUT 10 to 8 SIPP filter API 572
lut12to16.h	
	This file contains the declaration of the LUT 12 to 16 SIPP filter API 573
lut12to8.h	
	This file contains the declaration of the LUT 12 to 8 SIPP filter API 573
lut8to8.h	
	This file contains the declaration of the LUT 8 to 8 SIPP filter API 574
lutP10BppU16inU8out.h	
	This file contains the declaration of the LUT 16 to 8 SIPP filter API 575
maxTest3x3_fp16.h	
	This file contains the declaration of the maxTest3x3_fp16 SIPP filter API 575
meanStdDev.h	
	This file contains the declaration of the meanStdDev SIPP filter API 576
minMaxPos.h	
	This file contains the declaration of the Min/Max Value Position SIPP filter API . . . 576
minMaxValue.h	
	This file contains the declaration of the Min/Max Value SIPP filter API 577
minTest3x3_fp16.h	
	This file contains the declaration of the minTest3x3_fp16 SIPP filter API 577
mixMedian.h	
	This file contains the declaration of the Mix Median SIPP filter API 578

monoImbalance.h	
This file contains the declaration of the MonoImbalance SIPP filter API	579
myriad2SippDefs.inc	579
negative.h	
This file contains the declaration of the Negative SIPP filter API	579
nonMax3x3Fp32.h	
This file contains the declaration of the nonMaxFp32 SIPP filter API	580
nonMax3x3U8.h	
This file contains the declaration of the nonMax3x3U8 SIPP filter API	580
padBayer5.h	581
padBayer5Frame.h	581
pixelPacker10b.h	
This file contains the declaration of the Pixel packer SIPP filter API	582
pixelUnpacker.h	
This file contains the declaration of the Pixel Unpacker SIPP filter API	582
pixelUnpackerMipi10b.h	
This file contains the declaration of the Pixel Unpacker Mipi 10b SIPP filter API	583
pixelUnpackerWB.h	
This file contains the declaration of the Pixel Unpacker WB SIPP filter API	583
positionKernel.h	
This file contains the declaration of the Pixel Position SIPP filter API	584
purpleFlare.h	
This file contains the declaration of the purpleFlare SIPP filter API	584
pyrDown.h	
This file contains the declaration of the Pyramid Downscale SIPP filter API	585
randNoise.h	
This file contains the declaration of the Random Noise SIPP filter API	585
randNoiseFp16.h	
This file contains the declaration of the Random Noise (high speed) SIPP filter API	586
sad11x11.h	
This file contains the declaration of the Sum of Absolute Differences 11x11 SIPP filter API	587
sad5x5.h	
This file contains the declaration of the Sum of Absolute Differences 5x5 SIPP filter API	587
scale05BilinHV.h	
This file contains the declaration of the scale05bilinHV SIPP filter API	587
scale05BilinHV_Fp16U8.h	
This file contains the declaration of the Downscale by 2 (fp16/u8) SIPP filter API	588
scale05BilinHVFp16.h	
This file contains the declaration of the Downscale by 2 (fp16/fp16) SIPP filter API	588
scale05Lanc6HV.h	
This file contains the declaration of the Lanczos Downscale by 2 (6 taps) SIPP filter API	589
scale05Lanc7HV.h	
This file contains the declaration of the Lanczos Downscale by 2 (7 taps) SIPP filter API	589
scale2xBilinHV.h	
This file contains the declaration of the Upscale by 2 SIPP filter API	590

scale2xBilinHV_025_075_Fp16ToFp16.h	
This file contains the declaration of the Upscale by 2 with phases 0.25 and 0.75 fp16 to fp16 SIPP filter API	590
scale2xBilinHV_025_075_U16ToU16.h	
This file contains the declaration of the Upscale by 2 with phases 0.25 and 0.75 u16 to u16 SIPP filter API	591
scale2xBilinHV_Fp16U8_phase025_075.h	
This file contains the declaration of the Upscale by 2 with phases 0.25 and 0.75 fp16 to u8 SIPP filter API	591
scale2xBilinHV_U8ToU8_phase025_075.h	
This file contains the declaration of the Upscale by 2 with phases 0.25 and 0.75 u8 to u8 SIPP filter API	592
scale2xLancH.h	
This file contains the declaration of the Lanczos Horizontal Upscale by 2 SIPP filter API	592
scale2xLancHV.h	
This file contains the declaration of the Lanczos Upscale by 2 SIPP filter API	592
scale2xLancV.h	
This file contains the declaration of the Lanczos Vertical Upscale by 2 SIPP filter API	593
scaleBilinArb.h	
This file contains the declaration of the Arbitrary Downscale SIPP filter API	593
scharr_fp16.h	
This file contains the declaration of the scharr_fp16 SIPP filter API	594
sipp.h	
SIPP engine	594
sipp_ma2x5x.h	
SIPP engine	594
sippAccessScheduler.c	
SIPP framework API Platform(s) supported : MA2x5x	606
sippAccessScheduler.h	
SIPP framework API Platform(s) supported : MA2x5x	608
sippAccessSchedulerTypes.h	
SIPP framework API Platform(s) supported : MA2x5x	608
sippAnalysePipema2x5x.c	
Examine a pipeline for validity & features pertinent to implementation Platform(s) supported : MA2x5x	610
sippApi.c	
SIPP framework API Platform(s) supported : MA2x5x	611
sippApiInternal.c	614
sippBaseTypes.h	615
sippCfg.h	617
sippCmxDmaIff.c	
SIPP cmx dma interface - abstracted to this file for ease of switching cmx dma interfacing mechanisms - potentially to use generic driver	620
sippCoreApi.c	
SIPP engine	623
sippCoreGlobals.c	
SIPP engine	625
sippCoreHw.c	625
sippCoreUtils.c	
SIPP engine	626

sippDbg.c	
SIPP engine	628
sippDefines.h	630
sippDriverCmxDmaIf.c	
SIPP cmx dma driver interface	634
sippError.c	
SIPP framework Error management With a concurrent pipeline API, the minimum aim is that an error on one pipeline does NOT effect other pipelines also running at that time	635
sippEvents.h	637
sippFastExeUpd.h	
SIPP core : fast update of filters execution numbers	638
sippFilterAccesors.c	638
sippGenericRuntime.c	
SIPP engine	641
sippGenericRuntimema2x5x.c	
SIPP engine	642
sippGenericSchApi.c	643
sippGenericSchDebug.c	645
sippGenericSchReq.c	
SIPP engine	645
sippGenericSchWrite.c	
SIPP engine	647
sippHeap.c	647
sippHpad.c	
SIPP engine	649
sippHwBitfieldDefs.h	650
sippHwChromaDns.c	
SIPP engine	724
sippHwColComb.c	
SIPP engine	724
sippHwCommon.c	
SIPP engine	724
sippHwCommon_ma2x5x.h	726
sippHwConv.c	
SIPP engine	735
sippHwDebayer.c	
SIPP engine	735
sippHwDefs.h	
SIPP engine	735
sippHwDefs_ma2x5x.h	
Config data structures for MA2x5x SIPP HW filters. Most registers exposed through these data structures are described in detail in the MDK Programmer's Guide	736
sippHwDogLtm.c	
SIPP engine	738
sippHwEdge.c	
SIPP engine	738
sippHwGenChroma.c	
SIPP engine	738
sippHwHarris.c	
SIPP engine	739

sippHwIds.h	
HW filter related macros	739
sippHwLsc.c	
SIPP engine	746
sippHwLumaDns.c	
SIPP engine	747
sippHwLut.c	
SIPP engine	747
sippHwMedian.c	
SIPP engine	748
sippHwMipiRx.c	
SIPP engine	748
sippHwMipiTx.c	
SIPP engine	749
sippHwPolyFir.c	
SIPP engine	750
sippHwRaw.c	
SIPP engine	750
sippHWSessionControl.c	
Establishes context for and makes calls to sipp HW and shaves This file provides the hardware facing aspect to the interface. It is part of a set with sippPipeSessionControl.c which provides pipeline functionality Platform(s) supported : ma2x5x	751
sippHwSigma.c	
SIPP engine	752
sippHwUnsharp.c	
SIPP engine	753
sippInternal.h	
SIPP core : internal definitions	753
sippIoPtrs.c	
SIPP engine	766
sippIsr.c	
SIPP framework API - asynchronous runtime API Platform(s) supported : ma2x5x	769
sippManagerApi.c	
SIPP framework API Platform(s) supported : ma2x5x	770
sippMem.c	
SIPP engine	771
sippMemLineBuffer.c	
SIPP engine	774
sippMLPIRuntime.c	
Multiple Liner Per Iteration runtime	776
sippMLPIRuntimeema2x5x.c	
Multiple Line Per Iteration runtime, ma2x5x specific aspect	777
sippMLPISchApi.c	777
sippMLPISchDebug.c	777
sippMLPISchReq.c	777
sippMLPISchWrite.c	
SIPP engine	777
sippOPipeRuntime.c	
SIPP engine	777
sippOPipeSchApi.c	778

sippOPipeSchedulingEntity.c	
Establish the OPipe Scheduling Entities within a pipeline Platform(s) supported -	
: ma2x5x	778
sippPal.h	781
BM/src/leon/sippPalTypes.h	783
PC/sippPalTypes.h	791
RTEMS/src/leon/sippPalTypes.h	798
sippPipelineValidate.c	
Examine a pipeline for validity & features pertinent to implementation Platform(s)	
supported : MA2x5x	806
sippPipeSessionControl.c	
Processes commands from sipp access scheduler Establishes context for and makes	
calls to sipp HW and shaves This file provides the pipeline aspect to the interface. It	
is part of a set with SippHWSessionControl.c which provides the HW functionality	
Platform(s) supported : ma2x5x	807
sippPlatform.h	
SIPP engine	807
sippPlatform_ma2x5x.h	
SIPP engine	808
BM/src/leon/sippPlatformAbstractionLayer.c	811
PC/sippPlatformAbstractionLayer.c	812
RTEMS/src/leon/sippPlatformAbstractionLayer.c	
SIPP engine	813
sippScheduleIsr.c	
SIPP framework ISRs used by the schedule based runtimes in asyml mode Plat-	
form(s) supported : ma2x5x	815
sippSchTypes.h	817
sippSessionControl.h	
SIPP framework API Platform(s) supported : ma2x5x	818
sippShave.c	
SIPP engine	818
sippShaveIf.c	
SIPP engine	820
sippShaveMacros.h	
SIPP engine	820
sippShaveSym.h	821
sippShvDbg.c	
SIPP engine	821
sippTestCommon.c	
SIPP engine	821
sippTestCommon.h	
SIPP engine	822
sippTestCommon_ma2x5x.h	
SIPP engine	822
sippThread.c	
SIPP engine, RTOS thread	823
sippTypes.h	823
sippTypesPrivate.h	832
sippUtils.c	
SIPP engine	835

sLaplacian3x3Fp16ToFp16.h	
This file contains the declaration of the Laplacian 3x3 Fp16 To Fp16 SIPP filter API	837
sobel.h	
This file contains the declaration of the Sobel SIPP filter API	837
ssd11x11.h	
This file contains the declaration of the Sum of Squared Differences 11x11 SIPP filter API	838
ssd5x5.h	
This file contains the declaration of the Sum of Squared Differences 5x5 SIPP filter API	838
ssd7x7U8ToU32.h	
This file contains the declaration of the Sum of Squared Differences 7x7 (U8 to U32) SIPP filter API	839
ssdPointLine7x7U8U32.h	
This file contains the declaration of the Sum of Squared Differences 7x7 SIPP filter API	839
startBicubic.h	840
statsAwbSatPixels.h	840
statsAwbSatPixelsU32.h	840
subpixelFilter.h	
This file contains the declaration of the subpixel SIPP filter API	841
svuAbsdiff.c	842
svuAccumulateSquare.c	842
svuAccumulateWeighted.c	843
svuArithmeticAdd.c	844
svuArithmeticAddmask.c	844
svuArithmeticSub.c	845
svuArithmeticSubFp16ToFp16.c	845
svuArithmeticSubmask.c	846
svuAvg.c	847
svuBilateral5x5.c	847
svuBitwiseAnd.c	848
svuBitwiseAndMask.c	848
svuBitwiseNot.c	849
svuBitwiseOr.c	850
svuBitwiseOrMask.c	850
svuBitwiseXor.c	851
svuBitwiseXorMask.c	851
svuBoxFilter.c	852
svuBoxFilter11x11.c	853
svuBoxFilter13x13.c	854
svuBoxFilter15x15.c	854
svuBoxFilter3x3.c	855
svuBoxFilter5x5.c	855
svuBoxFilter7x7.c	856
svuBoxFilter9x9.c	857
svuCannyEdgeDetection.c	857
svuCensusMatching16.c	858
svuCensusMatching32.c	859
svuCensusMatching64.c	860
svuCensusMatching65.c	860

svuCensusMatchingPyr.c	861
svuCensusMin16.c	862
svuCensusMin64.c	862
svuCensusMin65.c	863
svuCensusMin7.c	863
svuCensusTransform5x5.c	864
svuChannelExtract.c	865
svuChromaBlock.c	865
svuCombDecimDemosaicAwbGains.c	866
svuContrast.c	867
svuConv11x11.c	867
svuConv15x1.c	868
svuConv1x15.c	868
svuConv1x5.c	869
svuConv1x5Fp16ToFp16.c	869
svuConv1x7.c	870
svuConv1x7Fp16ToFp16.c	870
svuConv1x9.c	871
svuConv3x3.c	872
svuConv3x3Fp16ToFp16.c	872
svuConv5x1.c	873
svuConv5x1Fp16ToFp16.c	873
svuConv5x5.c	874
svuConv5x5Fp16ToFp16.c	874
svuConv7x1.c	875
svuConv7x1Fp16ToFp16.c	876
svuConv7x7.c	876
svuConv7x7Fp16ToFp16.c	877
svuConv7x7Fp16ToU8.c	877
svuConv9x1.c	878
svuConv9x9.c	878
svuConv9x9Fp16ToFp16.c	879
svuConvert16bppTo8bpp.c	880
svuConvertF16ToU8.c	880
svuConvertFrom12BppTo8Bpp.c	881
svuConvertPFp16U16.c	881
svuConvertPU16Fp16.c	882
svuConvertU8ToF16.c	882
svuConvertYUV400ToYUV422.c	883
svuConvGeneric.c	883
svuConvSeparable11x11.c	884
svuConvSeparable11x11Fp16ToFp16.c	884
svuConvSeparable3x3.c	885
svuConvSeparable3x3Fp16ToFp16.c	885
svuConvSeparable5x5.c	886
svuConvSeparable5x5Fp16ToFp16.c	887
svuConvSeparable7x7.c	887
svuConvSeparable7x7Fp16ToFp16.c	888
svuConvSeparable9x9.c	888
svuConvSeparable9x9Fp16ToFp16.c	889
svuConvYuv444.c	890

svuCopy.c	890
svuCornerMinEigenVal.c	890
svuCornerMinEigenValpatched.c	892
svuCrop.c	892
svuCropCvtPlaneMode.c	892
svuCvtColorChromaYUVToNV12.c	893
svucvtColorNV21toRGB.c	894
svuCvtColorRGBfp16ToLumaU8.c	895
svuCvtColorRGBfp16ToUV420U8.c	895
svuCvtColorRGBtoChromaNV12.c	896
svuCvtColorRGBtoLuma.c	896
svuCvtColorRGBtoLumaNV12.c	897
svuCvtColorRGBtoUV.c	898
svuCvtColorRGBtoUV420.c	898
svuCvtColorRGBToYUV422.c	900
svuCvtColorYUV422ToRGB.c	900
svuCvtColorYUVToRGB.c	901
svuDilate3x3.c	901
svuDilate5x5.c	902
svuDilate7x7.c	902
svuDilateGeneric.c	903
svudisp2depth.c	903
svuEqualizeHist.c	904
svuErode3x3.c	904
svuErode5x5.c	905
svuErode7x7.c	905
svuExtAfStats.c	906
svuExtStatsSatPixelsU32.c	908
svuFast9M2.c	909
svuFast9ScoreCv.c	910
svuGauss.c	912
svuGaussHx2.c	913
svuGaussHx2_fp16.c	913
svuGaussVx2.c	914
svuGaussVx2_fp16.c	914
svuGenChroma.c	915
svuGenChromaSS.c	915
svuGenDnsRef.c	916
svuGenDnsRefFp16.c	916
svuGenLuma.c	917
svuGenLumaU8Fp16.c	917
svuGreyDesat.c	918
svuHammingDistance.c	918
svuHarrisResponse.c	919
svuHistogram.c	920
svuHistogramStat.c	920
svuHomography.c	921
svuIntegralImageSqSumF32M2.c	922
svuIntegralImageSqSumU32M2.c	922
svuIntegralImageSumF32M2.c	923
svuIntegralImageSumU16U32.c	924

svuIntegralImageSumU32M2.c	926
svuInterpolatePixelBilinear.c	927
svuLaplacian3x3.c	927
svuLaplacian5x5.c	929
svuLaplacian5x5Fp16ToFp16.c	929
svuLaplacian7x7.c	930
svuLaplacian7x7Fp16ToFp16.c	930
svuLocalMaxMin3x3_fp16.c	931
svuLocalTM.c	932
svuLowLvlCorr.c	932
svuLumaBlur.c	933
svuLut10to16.c	933
svuLut10to8.c	935
svuLut12to16.c	936
svuLut12to8.c	936
svuLut8to8.c	937
svuLutP10BppU16inU8out.c	938
svuMaxTest3x3_fp16.c	938
svuMeanStdDev.c	939
svuMinMaxPos.c	940
svuMinMaxValue.c	941
svuMinTest3x3_fp16.c	941
svuMixMedian.c	942
svuMonoImbalance.c	943
svuNegative.c	
Main leon file	943
svuNonMax3x3Fp32.c	944
svuNonMax3x3U8.c	945
svuPadBayer5.c	945
svuPadBayer5Frame.c	946
svuPixelPacker10b.c	946
svuPixelUnpacker.c	947
svuPixelUnpackerMipi10b.c	947
svuPixelUnpackerWB.c	948
svuPositionKernel.c	949
svuPurpleFlare.c	950
svuPyrDown.c	950
svuRandNoise.c	951
svuRandNoiseFp16.c	952
svuSAD11x11.c	953
svuSAD5x5.c	954
svuScale05BilinHV.c	954
svuScale05BilinHV_Fp16U8.c	955
svuScale05BilinHVFp16.c	956
svuScale05Lanc6HV.c	956
svuScale05Lanc7HV.c	957
svuScale2xBilinHV.c	957
svuScale2xBilinHV_025_075_Fp16ToFp16.c	958
svuScale2xBilinHV_025_075_U16ToU16.c	958
svuScale2xBilinHV_Fp16U8_phase025_075.c	959
svuScale2xBilinHV_U8ToU8_phase025_075.c	960

svuScale2xLancH.c	960
svuScale2xLancHV.c	960
svuScale2xLancV.c	962
svuScaleBilinArb.c	962
svuScharr_fp16.c	
Main leon file	962
svuLaplacian3x3Fp16ToFp16.c	963
svuSobel.c	
Main leon file	963
svuSSD11x11.c	964
svuSSD5x5.c	965
svuSSD7x7U8ToU32.c	965
svuSsdPointLine7x7U8U32.c	966
svuStartBicubic.c	966
svuStatsAwbSatPixels.c	967
svuStatsAwbSatPixelsU32.c	968
svuSubpixelFilter.c	968
svuThreshold.c	969
svuThresholdBinaryRange.c	970
svuThresholdBinaryU8.c	970
svuThresholdFilter.c	971
svuUndistortBrown.c	971
svuWhiteBalanceBayerGBRG.c	972
svuWhiteBalanceRGB.c	973
svuXYgen.c	974
threshold.h	
This file contains the declaration of the Threshold SIPP filter API	974
thresholdBinaryRange.h	
This file contains the declaration of the Threshold Binary Range SIPP filter API	975
thresholdBinaryU8.h	
This file contains the declaration of the Threshold Binary U8 SIPP filter API	975
thresholdFilter.h	
This file contains the declaration of the ThresholdFilter SIPP filter API	976
tripleConv3x3.c	976
tripleConv3x3.h	977
undistortBrown.h	
This file contains the declaration of the Undistort SIPP filter API	977
whiteBalanceBayerGBRG.h	
This file contains the declaration of the White Balance Bayer GBRG SIPP filter API	978
whiteBalanceRGB.h	
This file contains the declaration of the White Balance RGB SIPP filter API	978
ma2150/testUtils/wrapperSem.cpp	979
ma2x5x/testUtils/wrapperSem.cpp	979
ma2150/testUtils/wrapperSem.h	979
ma2x5x/testUtils/wrapperSem.h	979
xyGen.h	
This file contains the declaration of the XY Generator SIPP filter API	979

Chapter 5

Module Documentation

5.1 Median

Median Filter.

Data Structures

- struct [MedParam](#)

Parameter structure of the median filter.

5.1.1 Detailed Description

Median Filter.

Myriad architecture:

MA2x5x

Output data type(s):

UInt8

Filter type:

hw

Filter function:

SIPP_MED_ID

Inputs:

- name: input0; datatypes: UInt8; kernels: 3x0, 3x3, 5x5, 7x7

5.2 Lens Shading Correction

Lens shading correction (or anti-vignetting) compensates for the effect produced by camera optics whereby the light intensity of pixels reduces the further away from the centre of the image they are.

Data Structures

- struct [LscParam](#)

Parameter structure of the lsc filter.

5.2.1 Detailed Description

Lens shading correction (or anti-vignetting) compensates for the effect produced by camera optics whereby the light intensity of pixels reduces the further away from the centre of the image they are.

Myriad architecture:

MA2x5x

Output data type(s):

UInt8, UInt16

Filter type:

hw

Filter function:

SIPP_LSC_ID

Inputs:

- datatypes: UInt8, UInt16; kernels: 1x0, 1x1

5.3 Raw

The Raw filter performs a number of functions on raw CFA data, prior to demosaicing, including hot and cold pixel suppression, Gr/Gb imbalance correction, digital gain, and statistics collection.

Data Structures

- struct [RawParam](#)

Parameter structure of the raw filter.

5.3.1 Detailed Description

The Raw filter performs a number of functions on raw CFA data, prior to demosaicing, including hot and cold pixel suppression, Gr/Gb imbalance correction, digital gain, and statistics collection.

Myriad architecture:

MA2x5x

Preserve:

numPlanes, imgSize

Output data type(s):

UInt8, UInt16

Filter type:

hw

Filter function:

SIPP_RAW_ID

Inputs:

- datatypes: UInt8, UInt16; kernels: 5x5, 5x0

5.4 Debayer

This filter converts raw Bayer data into 3-channels-per-pixel RGB data.

Data Structures

- struct [DbyrParam](#)

Parameter structure of the debayer filter.

5.4.1 Detailed Description

This filter converts raw Bayer data into 3-channels-per-pixel RGB data.

Myriad architecture:

MA2x5x

Filter type:

hw

Filter function:

SIPP_DBYR_ID

Inputs:

- datatypes: UInt8, UInt16; kernels: 11x11

Output buffers:

- datatypes: UInt8, UInt16; preserve: imgSize
- datatypes: UInt8, fp16; preserve: imgSize, numPlains

5.5 Sharpen

The Sharpen filter enhances image sharpness. Programmable (separable, symmetric) blur filter kernel. Sharpening functionality can be disabled to use filter kernel on its own.

Data Structures

- struct [UsmParam](#)

Parameter structure of the sharpen filter.

5.5.1 Detailed Description

The Sharpen filter enhances image sharpness. Programmable (separable, symmetric) blur filter kernel. Sharpening functionality can be disabled to use filter kernel on its own.

Myriad architecture:

MA2x5x

Output data type(s):

UInt8, fp16

Filter type:

hw

Filter function:

SIPP_SHARPEN_ID

Preserve:

dataType, imgSize, numPlanes

Inputs:

- datatypes: UInt8, fp16; kernels: 3x3, 5x5, 7x7

5.6 Luma Denoise

This filter is designed to remove noise from a single image plane.

Data Structures

- struct `YDnsParam`

Parameter structure of the ydns filter.

5.6.1 Detailed Description

This filter is designed to remove noise from a single image plane.

Myriad architecture:

MA2x5x

Preserve:

numPlanes, imgSize

Output data type(s):

UInt8, fp16

Filter type:

hw

Filter function:

SIPP_LUMA_ID

Mandatory inputs:

1

Inputs:

- name: input; datatypes: UInt8, fp16; kernels: 11x11

5.7 Chroma Denoise

This filter performs chroma denoise using wide cascaded, thresholded box filters.

Data Structures

- struct [ChrDnsParam](#)

Parameter structure of the chormadns filter.

5.7.1 Detailed Description

This filter performs chroma denoise using wide cascaded, thresholded box filters.

Myriad architecture:

MA2x5x

Output data type(s):

UInt8

Filter type:

hw

Filter function:

SIPP_CHROMA_ID

Inputs:

- datatypes: UInt8; kernels: 3x3

5.8 Look-up table

The LUT is a highly flexible lookup table which can be used for tonal curve or gamma application, or other complex functions.

Data Structures

- struct [LutParam](#)

Parameter structure of the [lut](#) filter.

5.8.1 Detailed Description

The LUT is a highly flexible lookup table which can be used for tonal curve or gamma application, or other complex functions.

Myriad architecture:

MA2x5x

Preserve:

numPlanes, imgSize

Output data type(s):

UInt8, UInt16, half, fp16

Filter type:

hw

Filter function:

SIPP_LUT_ID

Inputs:

- datatypes: UInt8, UInt16, half, fp16; kernels: 1x1

5.9 Color Combination

The Color Combination filter takes Chrominance and Luminance data that was separated from RGB previously, for the purposes of independent filtering.

Data Structures

- struct `ColCombParam`

Parameter structure of the colorcomb filter.

5.9.1 Detailed Description

The Color Combination filter takes Chrominance and Luminance data that was separated from RGB previously, for the purposes of independent filtering.

Myriad architecture:

MA2x5x

Output data type(s):

UInt8, UInt16, half

Filter type:

hw

Filter function:

SIPP_CC_ID

Mandatory inputs:

2

Preserve:

Inputs:

- name: luma; datatypes: UInt8, fp16; kernels: 1x1
- name: chroma; datatypes: UInt8; kernels: 5x5

5.10 Convolution

Convolution kernel is used to apply arbitrary (i.e. non-separable) convolutions.

Data Structures

- struct [ConvParam](#)

Parameter structure of the convolution filter.

5.10.1 Detailed Description

Convolution kernel is used to apply arbitrary (i.e. non-separable) convolutions.

Myriad architecture:

MA2x5x

Output data type(s):

UInt8, half

Filter type:

hw

Filter function:

SIPP_CONV_ID

Inputs:

- datatypes: UInt8, half; kernels: 3x3, 5x5

5.11 Harris Corner Detector

The Harris corners filter performs corner detection on U8F image data.

Data Structures

- struct [HarrisParam](#)

Parameter structure of the harriscorners filter.

5.11.1 Detailed Description

The Harris corners filter performs corner detection on U8F image data.

Myriad architecture:

MA2x5x

Output data type(s):

half,fp16,fp32,float

Filter type:

hw

Preserve:

numPlanes, imgSize

Filter function:

SIPP_HARRIS_ID

Inputs:

- datatypes: UInt8; kernels: 5x5, 7x7, 9x9

5.12 Polyphase FIR Scaler

The poly-phase FIR filter scaler is suitable for high-quality implementations of scaling using e.g. Lanczos resampling.

Data Structures

- struct [PolyFirParam](#)

Parameter structure of the polyphasefir filter.

Enumerations

- enum [PolyModes](#) { [POLY_MODE_AUTO](#) = 0, [POLY_MODE_ADVANCE](#) = 1 }
- enum [PolyScalerType](#) { [POLY_LANCZOS](#) = 0, [POLY_BICUBIC](#) = 1, [POLY_BILINEAR](#) = 2 }
- enum [PolyPlaneMode](#) { [POLY_PLANE_ALL](#) = 0, [POLY_PLANE_Y](#) = 1, [POLY_PLANE_U](#) = 2, [POLY_PLANE_V](#) = 3, [POLY_PLANE_UV](#) = 4 }

5.12.1 Detailed Description

The poly-phase FIR filter scaler is suitable for high-quality implementations of scaling using e.g. Lanczos resampling.

Myriad architecture:

MA2x5x

Output data type(s):

UInt8, half

Filter type:

hw

Filter function:

SIPP_UPFIRDN_ID

Flags:

SIPP_RESIZE

Preserve:

Inputs:

- datatypes: UInt8, half; kernels: 3x3, 5x5, 7x7

5.12.2 Enumeration Type Documentation

enum **PolyModes**

Enumerator

POLY_MODE_AUTO
POLY_MODE_ADVANCE

enum **PolyPlaneMode**

Enumerator

POLY_PLANE_ALL
POLY_PLANE_Y
POLY_PLANE_U
POLY_PLANE_V
POLY_PLANE_UV

enum **PolyScalerType**

Enumerator

POLY_LANCZOS
POLY_BICUBIC
POLY_BILINEAR

5.13 Edge operator

Flexible 3x3 edge-detection operator suitable for implementation of e.g. Sobel filter.

Data Structures

- struct [EdgeParam](#)

Parameter structure of the edgeoperator filter.

5.13.1 Detailed Description

Flexible 3x3 edge-detection operator suitable for implementation of e.g. Sobel filter.

Myriad architecture:

MA2x5x

Preserve:

imgSize, numPlains

Output data type(s):

UInt8, UInt16

Filter type:

hw

Filter function:

SIPP_EDGE_OP_ID

Inputs:

- datatypes: UInt8; kernels: 3x3

5.14 Sigma Denoise

Sigma Denoise & Black level correction Filter.

Data Structures

- struct [SigmaParam](#)
Parameter structure of the edgeoperator filter.

5.14.1 Detailed Description

Sigma Denoise & Black level correction Filter.

Myriad architecture:

MA2x5x

Output data type(s):

UInt8, UInt16

Filter type:

hw

Filter function:

SIPP_SIGMA_ID

Inputs:

- datatypes: UInt8, UInt16; kernels: 5x5

5.15 Chroma Generation

Spatial sub-sampling, Purple Flare reduction, Desaturation & Chroma Generation Filter.

Data Structures

- struct [GenChrParam](#)

Parameter structure of the edgeoperator filter.

5.15.1 Detailed Description

Spatial sub-sampling, Purple Flare reduction, Desaturation & Chroma Generation Filter.

Myriad architecture:

MA2x5x

Output data type(s):

UInt8

Filter type:

hw

Filter function:

SIPP_CGEN_ID

Preserve:

numPlanes

Flags:

SIPP_RESIZE

Inputs:

- datatypes: UInt8, UInt16; kernels: 3x3, 6x6

5.16 DoG LTM

Spatial Local Tone Mapping plus Noise reduction based on a Difference of Gaussians.

Data Structures

- struct [DogLtmParam](#)

5.16.1 Detailed Description

Spatial Local Tone Mapping plus Noise reduction based on a Difference of Gaussians.

Myriad architecture:

MA2x5x

Output data type(s):

UInt8, fp16

Filter type:

hw

Filter function:

SIPP_DOGL_ID

Inputs:

- datatypes: UInt8, fp16; kernels: 3x3, 5x5, 7x7, 9x9, 11x11, 13x13, 15x15

5.17 MIPI Rx

Flexible streaming processing of input directly from MIPI Rx including windowing, sub-sampling, data selection, black level subtraction (for RAW input) and data format conversion.

Data Structures

- struct [MipiRxParam](#)

Parameter structure of the [MIPI Rx](#) filter.

5.17.1 Detailed Description

Flexible streaming processing of input directly from MIPI Rx including windowing, sub-sampling, data selection, black level subtraction (for RAW input) and data format conversion.

Output data type(s):

UInt8, UInt16, UInt32, half, fp16

Preserve:

Filter type:

hw

Filter functions:

SIPP_MIPI_RX0_ID, SIPP_MIPI_RX1_ID, SIPP_MIPI_RX2_ID, SIPP_MIPI_RX3_ID

Inputs:

Mandatory inputs:

0

5.18 MIPI Tx

Timing generation for MIPI Tx controller parallel interface for CSI-2/DSI output.

Data Structures

- struct [MipiTxParam](#)

Parameter structure of the [MIPI Rx](#) filter.

5.18.1 Detailed Description

Timing generation for MIPI Tx controller parallel interface for CSI-2/DSI output.

Output data type(s):

Filter type:

hw

Filter functions:

SIPP_MIPI_TX0_ID, SIPP_MIPI_TX1_ID

Inputs:

- datatypes: UInt8, UInt16, UInt32, half, fp16; kernels: 1x1

5.19 Absolute difference

This filter computes the absolute difference of two images.

Functions

- void **SVU_SYM()** **svuAbsdiff** (**SippFilter** *fptr)
Shave function of the [Absolute difference](#) filter.
- **SHAVE_SYM_EXPORT** (**svuAbsdiff**)

5.19.1 Detailed Description

This filter computes the absolute difference of two images.

Output data type(s):

UInt8

Inputs:

- name: input1; datatypes: UInt8; kernels: 1x1
- name: input2; datatypes: UInt8; kernels: 1x1

Mandatory inputs:

2

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/absoluteDiff/arch//shave/src/absoluteDiff.asm

5.19.2 Function Documentation

SHAVE_SYM_EXPORT (**svuAbsdiff**)

void **SVU_SYM()** **svuAbsdiff** (**SippFilter** * fptr)

Shave function of the [Absolute difference](#) filter.

5.20 Accumulate Square

This filter adds the square of the source image to the accumulator.

Functions

- void **SVU_SYM()** **svuAccumulateSquare** (**SippFilter** *fptr)
Shave function of the [Accumulate Square](#) filter.
- **SHAVE_SYM_EXPORT** (**svuAccumulateSquare**)

5.20.1 Detailed Description

This filter adds the square of the source image to the accumulator.

Preserve:

imgSize, numPlanes

Output data type(s):

float

Inputs:

- datatypes: UInt8; kernels: 1x1
- datatypes: UInt8; kernels: 1x1
- datatypes: float; kernels: 1x1

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/accumulateSquare/arch//shave/src/accumulateSquare.asm

5.20.2 Function Documentation

SHAVE_SYM_EXPORT (**svuAccumulateSquare**)

void **SVU_SYM()** **svuAccumulateSquare** (**SippFilter** * fptr)

Shave function of the [Accumulate Square](#) filter.

5.21 Accumulate Weighted

This filter calculates the weighted sum of the input image and the accumulator so that accumulator becomes a running average of frame sequence.

Data Structures

- struct `AccumulateWeightedParam`
Parameter structure of the `Accumulate Weighted` filter.

Functions

- void `SVU_SYM()` `svuAccumulateWeighted` (`SippFilter *fptr`)
Shave function of the `Accumulate Weighted` filter.
- `SHAVE_SYM_EXPORT` (`svuAccumulateWeighted`)

5.21.1 Detailed Description

This filter calculates the weighted sum of the input image and the accumulator so that accumulator becomes a running average of frame sequence.

Output data type(s):

fp32, float

Preserve:

numPlanes,imgSize

Inputs:

- datatypes: UInt8; kernels: 1x1
- datatypes: UInt8; kernels: 1x1
- datatypes: float; kernels: 1x1

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/accumulateWeighted/arch//shave/src/accumulateWeighted.asm

5.21.2 Function Documentation

`SHAVE_SYM_EXPORT` (`svuAccumulateWeighted`)

void `SVU_SYM()` `svuAccumulateWeighted` (`SippFilter * fptr`)

Shave function of the `Accumulate Weighted` filter.

5.22 Arithmetic addition

This filter performs addition two input images.

Functions

- void **SVU_SYM()** **svuArithmeticAdd** (**SippFilter** *fptr)
Shave function of the [Arithmetic addition](#) filter.
- **SHAVE_SYM_EXPORT** (**svuArithmeticAdd**)

5.22.1 Detailed Description

This filter performs addition two input images.

Output data type(s):

UInt8

Inputs:

- name: input1; datatypes: UInt8; kernels: 1x1
- name: input2; datatypes: UInt8; kernels: 1x1

Mandatory inputs:

2

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/arithmeticAdd/arch//shave/src/arithmeticAdd.asm

5.22.2 Function Documentation

SHAVE_SYM_EXPORT (**svuArithmeticAdd**)

void **SVU_SYM()** **svuArithmeticAdd** (**SippFilter** * fptr)

Shave function of the [Arithmetic addition](#) filter.

5.23 Arithmetic addition with mask

This filter performs addition two input images depending on the mask value.

Functions

- void **SVU_SYM()** **svuArithmeticAddmask** (**SippFilter** *fptr)
Shave function of the [Arithmetic addition with mask](#) filter.
- **SHAVE_SYM_EXPORT** (**svuArithmeticAddmask**)

5.23.1 Detailed Description

This filter performs addition two input images depending on the mask value.

Output data type(s):

UInt8

Inputs:

- name: input1 datatypes: UInt8; kernels: 1x1
- name: input2 datatypes: UInt8; kernels: 1x1
- name: input3 datatypes: UInt8; kernels: 1x1

Mandatory inputs:

3

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/arithmeticAddmask/arch//shave/src/arithmeticAddmask.asm

5.23.2 Function Documentation

SHAVE_SYM_EXPORT (**svuArithmeticAddmask**)

void **SVU_SYM()** **svuArithmeticAddmask** (**SippFilter** * fptr)

Shave function of the [Arithmetic addition with mask](#) filter.

5.24 Arithmetic subtraction

This filter performs subtraction two input images.

This filter performs subtraction two input images.

5.25 Arithmetic

Modules

- [12Bpp to 8Bpp conversion](#)

This filter performs a conversion on the input image from 12bpp to 8bpp.

- [Absolute difference](#)

This filter computes the absolute difference of two images.

- [Accumulate Square](#)

This filter adds the square of the source image to the accumulator.

- [Accumulate Weighted](#)

This filter calculates the weighted sum of the input image and the accumulator so that accumulator becomes a running average of frame sequence.

- [Arbitrary Downscale](#)

This filter performs bilinear scale, arbitrary X and Y scale factors.

- [Arithmetic addition](#)

This filter performs addition two input images.

- [Arithmetic addition with mask](#)

This filter performs addition two input images depending on the mask value.

- [Arithmetic subtraction](#)

This filter performs subtraction two input images.

- [Arithmetic subtraction fp16](#)

This filter performs subtraction of two fp16 input images.

- [Arithmetic subtraction with mask](#)

This filter performs subtraction of two input images depending on the mask value.

- [Average](#)

This filter outputs the average of two input images.

- [Bitwise And](#)

This filter performs per-element bit-wise logical conjunction(AND) for two input images.

- [Bitwise And with mask](#)

This filter performs per-element bit-wise logical conjunction(AND) for two input images, depending on the mask value.

- [Bitwise Not](#)

This filter performs per-element bit-wise NOT.

- [Bitwise Or](#)

This filter performs per-element bit-wise logical disjunction(OR) for two input images.

- [Bitwise Or with mask](#)

This filter performs per-element bit-wise logical disjunction(OR) for two input images, depending on the mask value.

- [Bitwise Xor](#)

This filter performs per-element bit-wise logical conjunction(XOR) for two input images.

- [Bitwise Xor with mask](#)

This filter performs per-element bit-wise logical conjunction(XOR) for two input images, depending on the mask value.

- [CensusTransform5x5](#)

- [Convert 16bpp To 8bpp](#)

- This filter performs conversion from 16 bpp input array to 8 bpp.*

 - **Convert F16 To U8**

This filter performs F16 to U8 conversion.
 - **Convert Fp16 to U16**

This filter performs FP16 to U16 conversion.
 - **Convert U16 to Fp16**

This filter performs U16 to Fp16 conversion.
 - **Convert U8 To F16**

This filter performs a conversion from U8 to F16.
 - **Convolution 11x11**

This filter performs a convolution on the input image using the given 11x11 matrix.
 - **Convolution 15x1**

This filter performs a convolution on the input image using the given 15x1 matrix.
 - **Convolution 1x15**

This filter performs a convolution on the input image using the given 1x15 matrix.
 - **Convolution 1x5**

This filter performs a convolution on the input image using the given 1x5 matrix.
 - **Convolution 1x5 Fp16ToFp16**

This filter performs a convolution on the fp16 input image using the given 1x5 matrix.
 - **Convolution 1x7**

This filter performs a convolution on the input image using the given 1x7 matrix.
 - **Convolution 1x7 Fp16ToFp16**

This filter performs a convolution on the fp16 input image using the given 1x7 matrix.
 - **Convolution 1x9**

This filter performs a convolution on the input image using the given 1x9 matrix.
 - **Convolution 3x3**

This filter performs a convolution on the input image using the given 3x3 matrix.
 - **Convolution 3x3 Fp16ToFp16**

This filter performs a convolution on the fp16 input image using the given 3x3 matrix.
 - **Convolution 5x1**

This filter performs a convolution on the input image using the given 5x1 matrix.
 - **Convolution 5x1 Fp16ToFp16**

This filter performs a convolution on the fp16 input image using the given 5x1 matrix.
 - **Convolution 5x5**

This filter performs a convolution on the input image using the given 5x5 matrix.
 - **Convolution 5x5 Fp16ToFp16**

This filter performs a convolution on the fp16 input image using the given 5x5 matrix.
 - **Convolution 7x1**

This filter performs a convolution on the input image using the given 7x1 matrix.
 - **Convolution 7x1 Fp16ToFp16**

This filter performs a convolution on the fp16 input image using the given 7x1 matrix.
 - **Convolution 7x7**

This filter performs a convolution on the input image using the given 7x7 matrix.
 - **Convolution 7x7 Fp16ToFp16**

This filter performs a convolution on the fp16 input image using the given 7x7 matrix.

- **Convolution 7x7 Fp16ToU8**

This filter performs a convolution on the fp16 input image using the given 7x7 matrix and stores the result in the U8 output image.

- **Convolution 9x1**

This filter performs a convolution on the input image using the given 9x1 matrix.

- **Convolution 9x9**

This filter performs a convolution on the input image using the given 9x9 matrix.

- **Convolution 9x9 Fp16ToFp16**

This filter performs a convolution on the fp16 input image using the given 9x9 matrix.

- **Convolution Separable 11x11**

This filter performs a separable convolution on the input image using the given 11x11 matrix.

- **Convolution Separable 11x11 Fp16ToFp16**

This filter performs a separable convolution on the fp16 input image using the given 11x11 matrix.

- **Convolution Separable 3x3**

This filter performs a separable convolution on the fp16 input image using the given 3x3 matrix.

- **Convolution Separable 3x3 Fp16ToFp16**

This filter performs a separable convolution on the fp16 input image using the given 3x3 matrix.

- **Convolution Separable 5x5**

This filter performs a separable convolution on the fp16 input image using the given 5x5 matrix.

- **Convolution Separable 5x5 Fp16ToFp16**

This filter performs a separable convolution on the fp16 input image using the given 5x5 matrix.

- **Convolution Separable 7x7**

This filter performs a separable convolution on the fp16 input image using the given 7x7 matrix.

- **Convolution Separable 7x7 Fp16ToFp16**

This filter performs a separable convolution on the fp16 input image using the given 7x7 matrix.

- **Convolution Separable 9x9**

This filter performs a separable convolution on the fp16 input image using the given 9x9 matrix.

- **Convolution Separable 9x9 Fp16ToFp16**

This filter performs a separable convolution on the fp16 input image using the given 9x9 matrix.

- **Copy**

This filter copies input image to output.

- **Downscale by 2**

This filter performs bilinear downscale with 0.5 factor – Horizontal and Vertical directions.

- **Downscale by 2 (fp16/fp16)**

This filter performs bilinear downscale with 0.5 factor – Horizontal and Vertical directions. fp16 in/out variant.

- **Downscale by 2 (fp16/u8)**

This filter performs bilinear downscale with 0.5 factor – Horizontal and Vertical directions. fp16 in/u8 out variant.

- **Gauss Blur**

This filter applies gaussian blur on Luma channel.

- **GaussHx2**

This filter applies downscale 2x horizontal gaussian blur. Has to be used in combination with GaussVx2 to obtain correct output.

- **GaussHx2_fp16**

This filter applies downscale 2x horizontal gaussian blur. Has to be used in combination with GaussVx2 to obtain correct output.

- **GaussVx2**

This filter applies downscale 2x vertical gaussian blur. Has to be used in combination with GaussHx2 to obtain correct output.

- **GaussVx2_fp16**

This filter applies downscale 2x vertical gaussian blur. Has to be used in combination with GaussHx2 to obtain correct output.

- **Generic Convolution**

This filter performs a generic convolution on the input image using the kernel size given by the user.

- **Lanczos Downscale by 2 (6 taps)**

This filter applies a lanczos downscale, with factor 0.5, and 6 taps; Horizontal and vertical directions.

- **Lanczos Downscale by 2 (7 taps)**

This filter applies a lanczos downscale, with factor 0.5, and 7 taps; Horizontal and vertical directions.

- **Lanczos Horizontal Upscale by 2**

This filter applies a lanczos upscale, with factor 2; Horizontal direction.

- **Lanczos Upscale by 2**

This filter applies a lanczos upscale, with factor 2; Horizontal and Vertical direction.

- **Lanczos Vertical Upscale by 2**

This filter applies a lanczos upscale, with factor 2; Vertical direction.

- **Laplacian 3x3**

The filter applies a Laplacian filter with custom size.

- **Laplacian 3x3 Fp16 To Fp16**

The filter applies a Laplacian filter with custom size.

- **Laplacian 5x5**

The filter applies a Laplacian filter with custom size.

- **Laplacian 5x5 Fp16 To Fp16**

The filter applies a Laplacian filter with custom size.

- **Laplacian 7x7**

The filter applies a Laplacian filter with custom size.

- **Laplacian 7x7 Fp16 To Fp16**

The filter applies a Laplacian filter with custom size.

- **Luma Blur**

This filter applies the blur operator on Luma channel.

- **MonoImbalance**

- **Negative**

This filter creates the negative of the image.

- **Pyramid Downscale**

This filter applies a pyramid operator using 5x5 gauss downscale operator.

- **Random Noise**

This filter generates random noise.

- **Random Noise (high speed)**

This filter generates random noise using high speed algorithm.

- **Sum of Absolute Differences 11x11**

The filter computes the sum of absolute differences between two input images on a 11x11 kernel.

- **Sum of Absolute Differences 5x5**

The filter computes the sum of absolute differences between two input images on a 5x5 kernel.

- [Sum of Squared Differences 11x11](#)

This filter performs sum of squared differences (SSD), the differences are squared and aggregated within a square window (11x11 in this case).

- [Sum of Squared Differences 5x5](#)

This filter performs sum of squared differences (SSD), the differences are squared and aggregated within a square window (5x5 in this case).

- [Sum of Squared Differences 7x7](#)

This filter performs sum of squared differences (SSD), the differences are squared and aggregated within a square window (7x7 in this case).

- [Sum of Squared Differences 7x7 \(U8 to U32\)](#)

This filter performs sum of squared differences (SSD) to a U32 output array, the differences are squared and aggregated within a square window (7x7 in this case).

- [Upscale by 2](#)

This filter performs bilinear upscale with 2 factor; Horizontal and Vertical directions.

- [XY Generator](#)

This filter performs xyGenerator for Bicubic.

- [bilateral5x5](#)

This filter performs per-element bit-wise logical conjunction(AND) for two input images.

- [censusMatching16](#)

*mvCvCensusMatching16 - performs an XOR operation between pixel one pixel in *in1 and 16 pixels from *in2 and counts up how many values of 1 are in the result*

- [censusMatching32](#)

*mvCvCensusMatching32 - performs an XOR operation between pixel one pixel in *in1 and 16 pixels from *in2 and counts up how many values of 1 are in the result*

- [censusMatching64](#)

*mvCvCensusMatching64 - performs an XOR operation between pixel one pixel in *in1 and 16 pixels from *in2 and counts up how many values of 1 are in the result*

- [censusMatching65](#)

*mvCvCensusMatching65 - performs an XOR operation between pixel one pixel in *in1 and 16 pixels from *in2 and counts up how many values of 1 are in the result*

- [censusMatchingPyr](#)

*mvCvCensusMatchingPyr - performs an XOR operation between pixel one pixel in *in1 and 16 pixels from *in2 and counts up how many values of 1 are in the result*

- [censusMin16](#)

mvCvCensusMin16 - computes minimum of 16 disparity costs values

- [censusMin64](#)

mvCvCensusMin64 - computes minimum of 64 disparity costs values

- [censusMin65](#)

mvCvCensusMin65 - computes minimum of 65 disparity costs values

- [censusMin7](#)

mvCvCensusMin7 - computes minimum of 7 disparity costs values

- [channelExtract](#)

This kernel extracts one of the R, G, B, plane from an interleaved RGB line.

- [hammingDistance](#)

hammingDistance kernel finds matches between two descriptors

- [localMaxMin3x3_fp16](#)

/// This kernel will find the points which are minimums or maximums in their 3x3 zone.

- [maxTest3x3_fp16](#)

This kernel will compare the points from inBufferCandidates to the corresponding 3x3 zone of inBuffer.

- [meanStdDev](#)

This kernel calculates mean and standard deviation of an array of elements.

- [minTest3x3_fp16](#)

This kernel will compare the points from inBufferCandidates to the corresponding 3x3 zone of inBuffer.

- [nonMax3x3U8](#)

This filter verifies whether each element from the central line is the maximum within a 3x3 range.

- [nonMaxFp32](#)

This filter verifies whether each element from the central line is the maximum within a 3x3 range.

Functions

- void [SVU_SYM\(\)](#) [svuArithmeticSub](#) ([SippFilter](#) *fptr)

Shave function of the [Arithmetic subtraction](#) filter.

- [SHAVE_SYM_EXPORT](#) ([svuArithmeticSub](#))

5.25.1 Detailed Description

Output data type(s):

UInt8

Inputs:

- name: input1 datatypes: UInt8; kernels: 1x1
- name: input2 datatypes: UInt8; kernels: 1x1

Mandatory inputs:

2

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/arithmeticSub/arch//shave/src/arithmeticSub.asm

5.25.2 Function Documentation

[SHAVE_SYM_EXPORT](#) ([svuArithmeticSub](#))

void [SVU_SYM\(\)](#) [svuArithmeticSub](#) ([SippFilter](#) * fptr)

Shave function of the [Arithmetic subtraction](#) filter.

5.26 Arithmetic subtraction fp16

This filter performs subtraction of two fp16 input images.

Functions

- void **SVU_SYM()** **svuArithmeticSubFp16ToFp16** (**SippFilter** *fptr)
Shave function of the [Arithmetic subtraction fp16](#) filter.
- **SHAVE_SYM_EXPORT** (**svuArithmeticSubFp16ToFp16**)

5.26.1 Detailed Description

This filter performs subtraction of two fp16 input images.

Output data type(s):

UInt16

Inputs:

- name: input1 datatypes: UInt16; kernels: 1x1
- name: input2 datatypes: UInt16; kernels: 1x1

Mandatory inputs:

2

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/arithmeticSubFp16ToFp16/arch//shave/src/arithmeticSub-Fp16ToFp16.asm

5.26.2 Function Documentation

SHAVE_SYM_EXPORT (**svuArithmeticSubFp16ToFp16**)

void **SVU_SYM()** **svuArithmeticSubFp16ToFp16** (**SippFilter** * fptr)

Shave function of the [Arithmetic subtraction fp16](#) filter.

5.27 Arithmetic subtraction with mask

This filter performs subtraction of two input images depending on the mask value.

Functions

- void **SVU_SYM()** **svuArithmeticSubmask** (**SippFilter** *fptr)
Shave function of the [Arithmetic subtraction with mask](#) filter.
- **SHAVE_SYM_EXPORT** (**svuArithmeticSubmask**)

5.27.1 Detailed Description

This filter performs subtraction of two input images depending on the mask value.

Output data type(s):

UInt8

Inputs:

- name: input1 datatypes: UInt8; kernels: 1x1
- name: input2 datatypes: UInt8; kernels: 1x1

Mandatory inputs:

2

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/arithmeticSubmask/arch//shave/src/arithmeticSubmask.asm

5.27.2 Function Documentation

SHAVE_SYM_EXPORT (**svuArithmeticSubmask**)

void **SVU_SYM()** **svuArithmeticSubmask** (**SippFilter** * fptr)

Shave function of the [Arithmetic subtraction with mask](#) filter.

5.28 Average

This filter outputs the average of two input images.

Functions

- void **SVU_SYM()** **svuAvg** (**SippFilter** *fptr)
Shave function of the [Average](#) filter.
- **SHAVE_SYM_EXPORT** (**svuAvg**)

5.28.1 Detailed Description

This filter outputs the average of two input images.

Output data type(s):

UInt8

Inputs:

- name: input1; datatypes: UInt8; kernels: 1x1
- name: input2; datatypes: UInt8; kernels: 1x1

Mandatory inputs:

2

5.28.2 Function Documentation

SHAVE_SYM_EXPORT (**svuAvg**)

void **SVU_SYM()** **svuAvg** (**SippFilter** * fptr)

Shave function of the [Average](#) filter.

5.29 bilateral5x5

This filter performs per-element bit-wise logical conjunction(AND) for two input images.

Data Structures

- struct [Bilateral5x5Param](#)

Functions

- void [SVU_SYM\(\)](#) [svuBilateral5x5](#) ([SippFilter](#) *fptr)
Shave function of the [bilateral5x5](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuBilateral5x5](#))

5.29.1 Detailed Description

This filter performs per-element bit-wise logical conjunction(AND) for two input images.

Preserve:

numPlanes

Output data type(s):

UInt16

Inputs:

- datatypes: UInt16; kernels: 5x5

Mandatory inputs:

1

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/bilateral5x5/shave/myriad2/bilateral5x5.asm

5.29.2 Function Documentation

[SHAVE_SYM_EXPORT](#) ([svuBilateral5x5](#))

void [SVU_SYM\(\)](#) [svuBilateral5x5](#) ([SippFilter](#) * fptr)

Shave function of the [bilateral5x5](#) filter.

5.30 Bitwise And

This filter performs per-element bit-wise logical conjunction(AND) for two input images.

Functions

- void **SVU_SYM()** **svuBitwiseAnd** (**SippFilter** *fptr)
*Shave function of the **Bitwise And** filter.*
- **SHAVE_SYM_EXPORT** (**svuBitwiseAnd**)

5.30.1 Detailed Description

This filter performs per-element bit-wise logical conjunction(AND) for two input images.

Output data type(s):

UInt8

Inputs:

- name: input1 datatypes: UInt8; kernels: 1x1
- name: input2 datatypes: UInt8; kernels: 1x1

Mandatory inputs:

2

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/bitwiseAnd/arch//shave/src/bitwiseAnd.asm

5.30.2 Function Documentation

SHAVE_SYM_EXPORT (**svuBitwiseAnd**)

void **SVU_SYM()** **svuBitwiseAnd** (**SippFilter** * fptr)

Shave function of the **Bitwise And** filter.

5.31 Bitwise And with mask

This filter performs per-element bit-wise logical conjunction(AND) for two input images, depending on the mask value.

Functions

- void **SVU_SYM()** **svbitwiseAndMask** (**SippFilter** *fptr)
*Shave function of the **Bitwise And with mask** filter.*
- **SHAVE_SYM_EXPORT** (**svbitwiseAndMask**)

5.31.1 Detailed Description

This filter performs per-element bit-wise logical conjunction(AND) for two input images, depending on the mask value.

Output data type(s):

UInt8

Inputs:

- name: input1 datatypes: UInt8; kernels: 1x1
- name: input2 datatypes: UInt8; kernels: 1x1

Mandatory inputs:

2

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/bitwiseAndMask/arch//shave/src/bitwiseAndMask.asm

5.31.2 Function Documentation

SHAVE_SYM_EXPORT (**svbitwiseAndMask**)

void **SVU_SYM()** **svbitwiseAndMask** (**SippFilter** * fptr)

Shave function of the **Bitwise And with mask** filter.

5.32 Bitwise Not

This filter performs per-element bit-wise NOT.

Functions

- void **SVU_SYM()** **svuBitwiseNot** (**SippFilter** *fptr)
*Shave function of the **Bitwise Not** filter.*
- **SHAVE_SYM_EXPORT** (**svuBitwiseNot**)

5.32.1 Detailed Description

This filter performs per-element bit-wise NOT.

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 1x1

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/bitwiseNot/arch//shave/src/bitwiseNot.asm

5.32.2 Function Documentation

SHAVE_SYM_EXPORT (**svuBitwiseNot**)

void **SVU_SYM()** **svuBitwiseNot** (**SippFilter** * fptr)

Shave function of the **Bitwise Not** filter.

5.33 Bitwise Or

This filter performs per-element bit-wise logical disjunction(OR) for two input images.

Functions

- void **SVU_SYM()** **svuBitwiseOr** (**SippFilter** *fptr)
*Shave function of the **Bitwise Or** filter.*
- **SHAVE_SYM_EXPORT** (**svuBitwiseOr**)

5.33.1 Detailed Description

This filter performs per-element bit-wise logical disjunction(OR) for two input images.

Output data type(s):

UInt8

Inputs:

- name: input1 datatypes: UInt8; kernels: 1x1
- name: input2 datatypes: UInt8; kernels: 1x1

Mandatory inputs:

2

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/bitwiseOr/arch//shave/src/bitwiseOr.asm

5.33.2 Function Documentation

SHAVE_SYM_EXPORT (**svuBitwiseOr**)

void **SVU_SYM()** **svuBitwiseOr** (**SippFilter** * fptr)

Shave function of the **Bitwise Or** filter.

5.34 Bitwise Or with mask

This filter performs per-element bit-wise logical disjunction(OR) for two input images, depending on the mask value.

Functions

- void **SVU_SYM()** **svuBitwiseOrMask** (**SippFilter** *fptr)
*Shave function of the **Bitwise Or with mask** filter.*
- **SHAVE_SYM_EXPORT** (**svuBitwiseOrMask**)

5.34.1 Detailed Description

This filter performs per-element bit-wise logical disjunction(OR) for two input images, depending on the mask value.

Output data type(s):

UInt8

Inputs:

- name: input1 datatypes: UInt8; kernels: 1x1
- name: input2 datatypes: UInt8; kernels: 1x1

Mandatory inputs:

2

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/bitwiseOrMask/arch//shave/src/bitwiseOrMask.asm

5.34.2 Function Documentation

SHAVE_SYM_EXPORT (**svuBitwiseOrMask**)

void **SVU_SYM()** **svuBitwiseOrMask** (**SippFilter** * fptr)

Shave function of the **Bitwise Or with mask** filter.

5.35 Bitwise Xor

This filter performs per-element bit-wise logical conjunction(XOR) for two input images.

Functions

- void **SVU_SYM()** **svuBitwiseXor** (**SippFilter** *fptr)
*Shave function of the **Bitwise Xor** filter.*
- **SHAVE_SYM_EXPORT** (**svuBitwiseXor**)

5.35.1 Detailed Description

This filter performs per-element bit-wise logical conjunction(XOR) for two input images.

Output data type(s):

UInt8

Inputs:

- name: input1 datatypes: UInt8; kernels: 1x1
- name: input2 datatypes: UInt8; kernels: 1x1

Mandatory inputs:

2

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/bitwiseXor/arch//shave/src/bitwiseXor.asm

5.35.2 Function Documentation

SHAVE_SYM_EXPORT (**svuBitwiseXor**)

void **SVU_SYM()** **svuBitwiseXor** (**SippFilter** * fptr)

Shave function of the **Bitwise Xor** filter.

5.36 Bitwise Xor with mask

This filter performs per-element bit-wise logical conjunction(XOR) for two input images, depending on the mask value.

Functions

- void **SVU_SYM()** **svuBitwiseXorMask** (**SippFilter** *fptr)
*Shave function of the **Bitwise And with mask** filter.*
- **SHAVE_SYM_EXPORT** (**svuBitwiseXorMask**)

5.36.1 Detailed Description

This filter performs per-element bit-wise logical conjunction(XOR) for two input images, depending on the mask value.

Output data type(s):

UInt8

Inputs:

- name: input1 datatypes: UInt8; kernels: 1x1
- name: input2 datatypes: UInt8; kernels: 1x1

Mandatory inputs:

2

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/bitwiseXorMask/arch//shave/src/bitwiseXorMask.asm

5.36.2 Function Documentation

SHAVE_SYM_EXPORT (**svuBitwiseXorMask**)

void **SVU_SYM()** **svuBitwiseXorMask** (**SippFilter** * fptr)

Shave function of the **Bitwise And with mask** filter.

5.37 Generic Box Filter

This filter calculates average on variable kernel size, on kernel size number of input lines.

Data Structures

- struct [BoxFilterParam](#)
Parameter structure of the [Generic Box Filter](#) filter.

Enumerations

- enum [boxDataFmt](#) {
 [FMT_U8](#), [FMT_U16](#), [FMT_U32](#), [FMT_F16](#),
 [FMT_F32](#) }

Functions

- void [SVU_SYM\(\)](#) [svuBoxFilter](#) ([SippFilter](#) *fptr)
Shave function of the [Generic Box Filter](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuBoxFilter](#))

5.37.1 Detailed Description

This filter calculates average on variable kernel size, on kernel size number of input lines.

Output data type(s):

UInt8, UInt16, UInt32, half, float

Inputs:

- datatypes: UInt8, UInt16, UInt32, half, float; kernels: 1x1, 3x3, 5x5, 7x7, 9x9, 11x11, 13x13, 15x15

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/boxFilter/arch//shave/src/boxFilter.asm

5.37.2 Enumeration Type Documentation

enum [boxDataFmt](#)

Enumerator

FMT_U8
FMT_U16
FMT_U32
FMT_F16
FMT_F32

5.37.3 Function Documentation

SHAVE_SYM_EXPORT (**svuBoxFilter**)

void **SVU_SYM**() **svuBoxFilter** (**SippFilter** * fptr)

Shave function of the **Generic Box Filter** filter.

5.38 Box Filter 11x11

This filter applies the box filter on the source image using the specified structuring element.

Data Structures

- struct **BoxFilter11x11Param**
*Parameter structure of the **Box Filter 11x11** filter.*

Functions

- void **SVU_SYM()** **svuBoxFilter11x11** (**SippFilter** *fptr)
*Shave function of the **Box Filter 11x11** filter.*
- **SHAVE_SYM_EXPORT** (**svuBoxFilter11x11**)

5.38.1 Detailed Description

This filter applies the box filter on the source image using the specified structuring element.

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 11x11

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/boxFilter11x11/arch//shave/src/boxFilter11x11.asm

5.38.2 Function Documentation

SHAVE_SYM_EXPORT (**svuBoxFilter11x11**)

void **SVU_SYM()** **svuBoxFilter11x11** (**SippFilter** * fptr)

Shave function of the **Box Filter 11x11** filter.

5.39 Box Filter 13x13

This filter applies the box filter on the source image using the specified structuring element.

Data Structures

- struct **BoxFilter13x13Param**
*Parameter structure of the **Box Filter 13x13** filter.*

Functions

- void **SVU_SYM()** **svuBoxFilter13x13** (**SippFilter** *fptr)
*Shave function of the **Box Filter 13x13** filter.*
- **SHAVE_SYM_EXPORT** (**svuBoxFilter13x13**)

5.39.1 Detailed Description

This filter applies the box filter on the source image using the specified structuring element.

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 13x13

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/boxFilter13x13/arch//shave/src/boxFilter13x13.asm

5.39.2 Function Documentation

SHAVE_SYM_EXPORT (**svuBoxFilter13x13**)

void **SVU_SYM()** **svuBoxFilter13x13** (**SippFilter** * fptr)

Shave function of the **Box Filter 13x13** filter.

5.40 Box Filter 15x15

This filter applies the box filter on the source image using the specified structuring element.

Data Structures

- struct **BoxFilter15x15Param**
*Parameter structure of the **Box Filter 15x15** filter.*

Functions

- void **SVU_SYM()** **svuBoxFilter15x15** (**SippFilter** *fptr)
*Shave function of the **Box Filter 15x15** filter.*
- **SHAVE_SYM_EXPORT** (**svuBoxFilter15x15**)

5.40.1 Detailed Description

This filter applies the box filter on the source image using the specified structuring element.

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 15x15

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/boxFilter15x15/arch//shave/src/boxFilter15x15.asm

5.40.2 Function Documentation

SHAVE_SYM_EXPORT (**svuBoxFilter15x15**)

void **SVU_SYM()** **svuBoxFilter15x15** (**SippFilter** * fptr)

Shave function of the **Box Filter 15x15** filter.

5.41 Box Filter 3x3

This filter applies the box filter on the source image using the specified structuring element.

Data Structures

- struct **BoxFilter3x3Param**
*Parameter structure of the **Box Filter 3x3** filter.*

Functions

- void **SVU_SYM()** **svuBoxFilter3x3** (**SippFilter** *fptr)
*Shave function of the **Box Filter 3x3** filter.*
- **SHAVE_SYM_EXPORT** (**svuBoxFilter3x3**)

5.41.1 Detailed Description

This filter applies the box filter on the source image using the specified structuring element.

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 3x3

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/boxFilter3x3/arch//shave/src/boxFilter3x3.asm

5.41.2 Function Documentation

SHAVE_SYM_EXPORT (**svuBoxFilter3x3**)

void **SVU_SYM()** **svuBoxFilter3x3** (**SippFilter** * fptr)

Shave function of the **Box Filter 3x3** filter.

5.42 Box Filter 5x5

This filter applies the box filter on the source image using the specified structuring element.

Data Structures

- struct **BoxFilter5x5Param**
*Parameter structure of the **Box Filter 5x5** filter.*

Functions

- void **SVU_SYM()** **svuBoxFilter5x5** (**SippFilter** *fptr)
*Shave function of the **Box Filter 5x5** filter.*
- **SHAVE_SYM_EXPORT** (**svuBoxFilter5x5**)

5.42.1 Detailed Description

This filter applies the box filter on the source image using the specified structuring element.

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 5x5

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/boxFilter5x5/arch//shave/src/boxFilter5x5.asm

5.42.2 Function Documentation

SHAVE_SYM_EXPORT (**svuBoxFilter5x5**)

void **SVU_SYM()** **svuBoxFilter5x5** (**SippFilter** * fptr)

Shave function of the **Box Filter 5x5** filter.

5.43 Box Filter 7x7

This filter applies the box filter on the source image using the specified structuring element.

Data Structures

- struct **BoxFilter7x7Param**
*Parameter structure of the **Box Filter 7x7** filter.*

Functions

- void **SVU_SYM()** **svuBoxFilter7x7** (**SippFilter** *fptr)
*Shave function of the **Box Filter 7x7** filter.*
- **SHAVE_SYM_EXPORT** (**svuBoxFilter7x7**)

5.43.1 Detailed Description

This filter applies the box filter on the source image using the specified structuring element.

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 7x7

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/boxFilter7x7/arch//shave/src/boxFilter7x7.asm

5.43.2 Function Documentation

SHAVE_SYM_EXPORT (**svuBoxFilter7x7**)

void **SVU_SYM()** **svuBoxFilter7x7** (**SippFilter** * fptr)

Shave function of the **Box Filter 7x7** filter.

5.44 Box Filter 9x9

This filter applies the box filter on the source image using the specified structuring element.

Data Structures

- struct **BoxFilter9x9Param**
*Parameter structure of the **Box Filter 9x9** filter.*

Functions

- void **SVU_SYM()** **svuBoxFilter9x9** (**SippFilter** *fptr)
*Shave function of the **Box Filter 9x9** filter.*
- **SHAVE_SYM_EXPORT** (**svuBoxFilter9x9**)

5.44.1 Detailed Description

This filter applies the box filter on the source image using the specified structuring element.

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 9x9

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/boxFilter9x9/arch//shave/src/boxFilter9x9.asm

5.44.2 Function Documentation

SHAVE_SYM_EXPORT (**svuBoxFilter9x9**)

void **SVU_SYM()** **svuBoxFilter9x9** (**SippFilter** * fptr)

Shave function of the **Box Filter 9x9** filter.

5.45 Canny Edge Detection

The filter finds edges in the input image and marks them in the output map edges using the Canny algorithm.

Data Structures

- struct `cannyEdgeDetectionParam`
Parameter structure of the Canny Edge Detection filter.

Functions

- void `SVU_SYM() svuCannyEdgeDetection (SippFilter *fptr)`
Shave function of the Canny Edge Detection filter.
- `SHAVE_SYM_EXPORT (svuCannyEdgeDetection)`

5.45.1 Detailed Description

The filter finds edges in the input image and marks them in the output map edges using the Canny algorithm.

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 9x9

Path to external ASM file(s) used in the filter implementation:

`/components/kernelLib/MvCV/kernels/canny/arch//shave/src/canny.asm`

5.45.2 Function Documentation

`SHAVE_SYM_EXPORT (svuCannyEdgeDetection)`

void `SVU_SYM() svuCannyEdgeDetection (SippFilter * fptr)`

Shave function of the Canny Edge Detection filter.

5.46 censusMatching16

mvCvCensusMatching16 - performs an XOR operation between pixel one pixel in *in1 and 16 pixels from *in2 and counts up how many values of 1 are in the result

Functions

- void **SVU_SYM()** svuCensusMatching16 (SippFilter *fptr)
Parameter structure of the censusMatching16 filter.
- **SHAVE_SYM_EXPORT** (svuCensusMatching16)

5.46.1 Detailed Description

mvCvCensusMatching16 - performs an XOR operation between pixel one pixel in *in1 and 16 pixels from *in2 and counts up how many values of 1 are in the result

Preserve:

numPlanes

Output data type(s):

UInt8

Inputs:

- datatypes: UInt32; kernels: 1x1
- datatypes: UInt32; kernels: 1x32

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/censusMatching16/arch//shave/src/censusMatching16.asm

5.46.2 Function Documentation

SHAVE_SYM_EXPORT (svuCensusMatching16)

void **SVU_SYM()** svuCensusMatching16 (**SippFilter** * fptr)

Parameter structure of the censusMatching16 filter.

Shave function of the censusMatching16 filter.

5.47 censusMatching32

mvCvCensusMatching32 - performs an XOR operation between pixel one pixel in *in1 and 16 pixels from *in2 and counts up how many values of 1 are in the result

Data Structures

- struct **CensusMatching32Param**
Parameter structure of the censusMatching32 filter.

Functions

- void **SVU_SYM()** svuCensusMatching32 (SippFilter *fptr)
Shave function of the censusMatching32 filter.
- **SHAVE_SYM_EXPORT** (svuCensusMatching32)

5.47.1 Detailed Description

mvCvCensusMatching32 - performs an XOR operation between pixel one pixel in *in1 and 16 pixels from *in2 and counts up how many values of 1 are in the result

Preserve:

numPlanes

Output data type(s):

UInt8

Inputs:

- datatypes: UInt32; kernels: 1x1
- datatypes: UInt32; kernels: 1x64

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/censusMatching32/arch//shave/src/censusMatching32.asm

5.47.2 Function Documentation

SHAVE_SYM_EXPORT (**svuCensusMatching32**)

void **SVU_SYM()** svuCensusMatching32 (**SippFilter** * fptr)

Shave function of the **censusMatching32** filter.

5.48 censusMatching64

mvCvCensusMatching64 - performs an XOR operation between pixel one pixel in *in1 and 16 pixels from *in2 and counts up how many values of 1 are in the result

Data Structures

- struct [CensusMatching64Param](#)
Parameter structure of the [censusMatching64](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuCensusMatching64](#) ([SippFilter](#) *fptr)
Shave function of the [censusMatching64](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuCensusMatching64](#))

5.48.1 Detailed Description

mvCvCensusMatching64 - performs an XOR operation between pixel one pixel in *in1 and 16 pixels from *in2 and counts up how many values of 1 are in the result

Preserve:

numPlanes

Output data type(s):

UInt8

Inputs:

- datatypes: UInt32; kernels: 1x1
- datatypes: UInt32; kernels: 1x128

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/censusMatching64/arch//shave/src/censusMatching64.asm

5.48.2 Function Documentation

[SHAVE_SYM_EXPORT](#) ([svuCensusMatching64](#))

void [SVU_SYM\(\)](#) [svuCensusMatching64](#) ([SippFilter](#) * fptr)

Shave function of the [censusMatching64](#) filter.

5.49 censusMatching65

mvCvCensusMatching65 - performs an XOR operation between pixel one pixel in *in1 and 16 pixels from *in2 and counts up how many values of 1 are in the result

Functions

- void **SVU_SYM()** svuCensusMatching65 (SippFilter *fptr)
Parameter structure of the censusMatching65 filter.
- **SHAVE_SYM_EXPORT** (svuCensusMatching65)

5.49.1 Detailed Description

mvCvCensusMatching65 - performs an XOR operation between pixel one pixel in *in1 and 16 pixels from *in2 and counts up how many values of 1 are in the result

Preserve:

numPlanes

Output data type(s):

UInt8

Inputs:

- datatypes: UInt32; kernels: 1x1
- datatypes: UInt32; kernels: 1x130

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/censusMatching65/arch//shave/src/censusMatching65.asm

5.49.2 Function Documentation

SHAVE_SYM_EXPORT (svuCensusMatching65)

void **SVU_SYM()** svuCensusMatching65 (**SippFilter** * fptr)

Parameter structure of the censusMatching65 filter.

Shave function of the censusMatching65 filter.

5.50 censusMatchingPyr

mvCvCensusMatchingPyr - performs an XOR operation between pixel one pixel in *in1 and 16 pixels from *in2 and counts up how many values of 1 are in the result

Data Structures

- struct [CensusMatchingPyrParam](#)
Parameter structure of the [censusMatchingPyr](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuCensusMatchingPyr](#) ([SippFilter](#) *fptr)
Shave function of the [censusMatchingPyr](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuCensusMatchingPyr](#))

5.50.1 Detailed Description

mvCvCensusMatchingPyr - performs an XOR operation between pixel one pixel in *in1 and 16 pixels from *in2 and counts up how many values of 1 are in the result

Preserve:

numPlanes

Output data type(s):

UInt8

Inputs:

- datatypes: UInt32; kernels: 1x1
- datatypes: UInt32; kernels: 1x15

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/censusMatchingPyr/arch//shave/src/censusMatchingPyr.asm

5.50.2 Function Documentation

[SHAVE_SYM_EXPORT](#) ([svuCensusMatchingPyr](#))

void [SVU_SYM\(\)](#) [svuCensusMatchingPyr](#) ([SippFilter](#) * fptr)

Shave function of the [censusMatchingPyr](#) filter.

5.51 censusMin16

mvcvCensusMin16 - computes minimum of 16 disparity costs values

Functions

- void **SVU_SYM()** svuCensusMin16 (**SippFilter** *fptr)
Parameter structure of the censusMin16 filter.
- **SHAVE_SYM_EXPORT** (svuCensusMin16)

5.51.1 Detailed Description

mvcvCensusMin16 - computes minimum of 16 disparity costs values

Output data type(s):

UInt8

Preserve:

numPlanes

Inputs:

- datatypes: UInt8; kernels: 1x1

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/censusMin16/arch//shave/src/censusMin16.asm

5.51.2 Function Documentation

SHAVE_SYM_EXPORT (svuCensusMin16)

void **SVU_SYM()** svuCensusMin16 (**SippFilter** * fptr)

Parameter structure of the censusMin16 filter.

Shave function of the censusMin16 filter.

5.52 censusMin64

mvcvCensusMin64 - computes minimum of 64 disparity costs values

Functions

- void **SVU_SYM()** **svuCensusMin64** (**SippFilter** *fptr)
*Parameter structure of the **censusMin64** filter.*
- **SHAVE_SYM_EXPORT** (**svuCensusMin64**)

5.52.1 Detailed Description

mvcvCensusMin64 - computes minimum of 64 disparity costs values

Preserve:

numPlanes

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 1x1

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/censusMin64/arch//shave/src/censusMin64.asm

5.52.2 Function Documentation

SHAVE_SYM_EXPORT (**svuCensusMin64**)

void **SVU_SYM()** **svuCensusMin64** (**SippFilter** * fptr)

Parameter structure of the **censusMin64** filter.

Shave function of the **censusMin64** filter.

5.53 censusMin65

mvcvCensusMin65 - computes minimum of 65 disparity costs values

Functions

- void **SVU_SYM()** **svuCensusMin65** (**SippFilter** *fptr)
*Parameter structure of the **censusMin65** filter.*
- **SHAVE_SYM_EXPORT** (**svuCensusMin65**)

5.53.1 Detailed Description

mvcvCensusMin65 - computes minimum of 65 disparity costs values

Preserve:

numPlanes

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 1x1

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/censusMin65/arch//shave/src/censusMin65.asm

5.53.2 Function Documentation

SHAVE_SYM_EXPORT (**svuCensusMin65**)

void **SVU_SYM()** **svuCensusMin65** (**SippFilter** * fptr)

Parameter structure of the **censusMin65** filter.

Shave function of the **censusMin65** filter.

5.54 censusMin7

mvcvCensusMin7 - computes minimum of 7 disparity costs values

Functions

- void **SVU_SYM()** **svuCensusMin7** (**SippFilter** *fptr)
*Parameter structure of the **censusMin7** filter.*
- **SHAVE_SYM_EXPORT** (**svuCensusMin7**)

5.54.1 Detailed Description

mvcvCensusMin7 - computes minimum of 7 disparity costs values

Preserve:

numPlanes

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 1x1

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/censusMin7/arch//shave/src/censusMin7.asm

5.54.2 Function Documentation

SHAVE_SYM_EXPORT (**svuCensusMin7**)

void **SVU_SYM()** **svuCensusMin7** (**SippFilter** * fptr)

Parameter structure of the **censusMin7** filter.

Shave function of the **censusMin7** filter.

5.55 CensusTransform5x5

Functions

- void **SVU_SYM()** **svuCensusTransform5x5** (**SippFilter** *fptr)
*Parameter structure of the **CensusTransform5x5** filter.*
- **SHAVE_SYM_EXPORT** (**svuCensusTransform5x5**)

5.55.1 Detailed Description

Preserve:

imgSize

Output data type(s):

UInt32

Inputs:

- datatypes: UInt8; kernels 5x5:

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/censusTransform5x5/arch//shave/src/censusTransform5x5.asm

5.55.2 Function Documentation

SHAVE_SYM_EXPORT (**svuCensusTransform5x5**)

void **SVU_SYM()** **svuCensusTransform5x5** (**SippFilter** * fptr)

Parameter structure of the **CensusTransform5x5** filter.

Shave function of the **CensusTransform5x5** filter.

5.56 channelExtract

This kernel extracts one of the R, G, B, plane from an interleaved RGB line.

Data Structures

- struct **ChannelExtractParam**
*Parameter structure of the **channelExtract** filter.*

Functions

- void **SVU_SYM()** **svuChannelExtract** (**SippFilter** *fptr)
*Shave function of the **channelExtract** filter.*
- **SHAVE_SYM_EXPORT** (**svuChannelExtract**)

5.56.1 Detailed Description

This kernel extracts one of the R, G, B, plane from an interleaved RGB line.

Preserve:

dataType

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 1x1

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/channelExtract/arch//shave/src/channelExtract.asm

5.56.2 Function Documentation

SHAVE_SYM_EXPORT (**svuChannelExtract**)

void **SVU_SYM()** **svuChannelExtract** (**SippFilter** * fptr)

Shave function of the **channelExtract** filter.

5.57 Chroma Block

This filter applies chroma desaturation and 3x3 color correction matrix.

Data Structures

- struct **ChromaBlkParam**

*Parameter structure of the **Chroma Block** filter.*

Functions

- void **SVU_SYM()** **svuChromaBlock** (**SippFilter** *fptr)
*Shave function of the **Chroma Block** filter.*
- **SHAVE_SYM_EXPORT** (**svuChromaBlock**)

5.57.1 Detailed Description

This filter applies chroma desaturation and 3x3 color correction matrix.

Output data type(s):

UInt8, half

Preserve:

imgSize

Inputs:

- name:rgb; datatypes: UInt8; kernels: 1x1
- name:luma; datatypes: UInt8, half; kernels: 1x1

5.57.2 Function Documentation

SHAVE_SYM_EXPORT (**svuChromaBlock**)

void **SVU_SYM()** **svuChromaBlock** (**SippFilter** * fptr)

Shave function of the **Chroma Block** filter.

5.58 Contrast

This filter applies contrast on pixel element.

Data Structures

- struct **ContrastParam**
*Parameter structure of the **Contrast** filter.*

Functions

- void **SVU_SYM()** **svuContrast** (**SippFilter** *fptr)
*Shave function of the **Contrast** filter.*
- **SHAVE_SYM_EXPORT** (**svuContrast**)

5.58.1 Detailed Description

This filter applies contrast on pixel element.

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 1x1

5.58.2 Function Documentation

SHAVE_SYM_EXPORT (**svuContrast**)

void **SVU_SYM()** **svuContrast** (**SippFilter** * fptr)

Shave function of the **Contrast** filter.

5.59 Convolution 11x11

This filter performs a convolution on the input image using the given 11x11 matrix.

Data Structures

- struct `Conv11x11Param`
Parameter structure of the [Convolution 11x11](#) filter.

Functions

- void `SVU_SYM() svuConv11x11 (SippFilter *fptr)`
Shave function of the [Convolution 11x11](#) filter.
- `SHAVE_SYM_EXPORT (svuConv11x11)`

5.59.1 Detailed Description

This filter performs a convolution on the input image using the given 11x11 matrix.

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 11x11

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/convolution11x11/arch//shave/src/convolution11x11.asm

5.59.2 Function Documentation

`SHAVE_SYM_EXPORT (svuConv11x11)`

void `SVU_SYM() svuConv11x11 (SippFilter * fptr)`

Shave function of the [Convolution 11x11](#) filter.

5.60 Convolution 15x1

This filter performs a convolution on the input image using the given 15x1 matrix.

Data Structures

- struct **Conv15x1Param**
*Parameter structure of the **Convolution 15x1** filter.*

Functions

- void **SVU_SYM()** **svuConv15x1** (**SippFilter** *fptr)
*Shave function of the **Convolution 15x1** filter.*
- **SHAVE_SYM_EXPORT** (**svuConv15x1**)

5.60.1 Detailed Description

This filter performs a convolution on the input image using the given 15x1 matrix.

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 15x1

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/convolution15x1/arch//shave/src/convolution15x1.asm

5.60.2 Function Documentation

SHAVE_SYM_EXPORT (**svuConv15x1**)

void **SVU_SYM()** **svuConv15x1** (**SippFilter** * fptr)

Shave function of the **Convolution 15x1** filter.

5.61 Convolution 1x15

This filter performs a convolution on the input image using the given 1x15 matrix.

Data Structures

- struct **Conv1x15Param**
*Parameter structure of the **Convolution 1x15** filter.*

Functions

- void **SVU_SYM()** **svuConv1x15** (**SippFilter** *fptr)
*Shave function of the **Convolution 1x15** filter.*
- **SHAVE_SYM_EXPORT** (**svuConv1x15**)

5.61.1 Detailed Description

This filter performs a convolution on the input image using the given 1x15 matrix.

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 1x15

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/convolution1x15/arch//shave/src/convolution1x15.asm

5.61.2 Function Documentation

SHAVE_SYM_EXPORT (**svuConv1x15**)

void **SVU_SYM()** **svuConv1x15** (**SippFilter** * fptr)

Shave function of the **Convolution 1x15** filter.

5.62 Convolution 1x5

This filter performs a convolution on the input image using the given 1x5 matrix.

Data Structures

- struct **Conv1x5Param**
*Parameter structure of the **Convolution 1x5** filter.*

Functions

- void **SVU_SYM()** **svuConv1x5** (**SippFilter** *fptr)
*Shave function of the **Convolution 1x5** filter.*
- **SHAVE_SYM_EXPORT** (**svuConv1x5**)

5.62.1 Detailed Description

This filter performs a convolution on the input image using the given 1x5 matrix.

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 1x5

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/convolution1x5/arch//shave/src/convolution1x5.asm

5.62.2 Function Documentation

SHAVE_SYM_EXPORT (**svuConv1x5**)

void **SVU_SYM()** **svuConv1x5** (**SippFilter** * fptr)

Shave function of the **Convolution 1x5** filter.

5.63 Convolution 1x5 Fp16ToFp16

This filter performs a convolution on the fp16 input image using the given 1x5 matrix.

Data Structures

- struct **Conv1x5Fp16ToFp16Param**
*Parameter structure of the **Convolution 1x5 Fp16ToFp16** filter.*

Functions

- void **SVU_SYM()** **svuConv1x5Fp16ToFp16** (**SippFilter** *fptr)
*Shave function of the **Convolution 1x5 Fp16ToFp16** filter.*
- **SHAVE_SYM_EXPORT** (**svuConv1x5Fp16ToFp16**)

5.63.1 Detailed Description

This filter performs a convolution on the fp16 input image using the given 1x5 matrix.

Output data type(s):

half

Inputs:

- datatypes: half; kernels: 1x5

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/convolution1x5Fp16ToFp16/arch//shave/src/convolution1x5-Fp16ToFp16.asm

5.63.2 Function Documentation

SHAVE_SYM_EXPORT (**svuConv1x5Fp16ToFp16**)

void **SVU_SYM()** **svuConv1x5Fp16ToFp16** (**SippFilter** * fptr)

Shave function of the **Convolution 1x5 Fp16ToFp16** filter.

5.64 Convolution 1x7

This filter performs a convolution on the input image using the given 1x7 matrix.

Data Structures

- struct `Conv1x7Param`
Parameter structure of the Convolution 1x7 filter.

Functions

- void `SVU_SYM() svuConv1x7 (SippFilter *fptr)`
Shave function of the Convolution 1x7 filter.
- `SHAVE_SYM_EXPORT (svuConv1x7)`

5.64.1 Detailed Description

This filter performs a convolution on the input image using the given 1x7 matrix.

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 1x7

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/convolution1x7/arch//shave/src/convolution1x7.asm

5.64.2 Function Documentation

`SHAVE_SYM_EXPORT (svuConv1x7)`

void `SVU_SYM() svuConv1x7 (SippFilter * fptr)`

Shave function of the Convolution 1x7 filter.

5.65 Convolution 1x7 Fp16ToFp16

This filter performs a convolution on the fp16 input image using the given 1x7 matrix.

Data Structures

- struct **Conv1x7Fp16ToFp16Param**
*Parameter structure of the **Convolution 1x7 Fp16ToFp16** filter.*

Functions

- void **SVU_SYM()** **svuConv1x7Fp16ToFp16** (**SippFilter** *fptr)
*Shave function of the **Convolution 1x7 Fp16ToFp16** filter.*
- **SHAVE_SYM_EXPORT** (**svuConv1x7Fp16ToFp16**)

5.65.1 Detailed Description

This filter performs a convolution on the fp16 input image using the given 1x7 matrix.

Output data type(s):

half

Inputs:

- datatypes: half; kernels: 1x7

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/convolution1x7Fp16ToFp16/arch//shave/src/convolution1x7-Fp16ToFp16.asm

5.65.2 Function Documentation

SHAVE_SYM_EXPORT (**svuConv1x7Fp16ToFp16**)

void **SVU_SYM()** **svuConv1x7Fp16ToFp16** (**SippFilter** * fptr)

Shave function of the **Convolution 1x7 Fp16ToFp16** filter.

5.66 Convolution 1x9

This filter performs a convolution on the input image using the given 1x9 matrix.

Data Structures

- struct `Conv1x9Param`
Parameter structure of the `Convolution 1x9` filter.

Functions

- void `SVU_SYM() svuConv1x9 (SippFilter *fptr)`
Shave function of the `Convolution 1x9` filter.
- `SHAVE_SYM_EXPORT (svuConv1x9)`

5.66.1 Detailed Description

This filter performs a convolution on the input image using the given 1x9 matrix.

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 1x9

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/convolution1x9/arch//shave/src/convolution1x9.asm

5.66.2 Function Documentation

`SHAVE_SYM_EXPORT (svuConv1x9)`

void `SVU_SYM() svuConv1x9 (SippFilter * fptr)`

Shave function of the `Convolution 1x9` filter.

5.67 Convolution 3x3

This filter performs a convolution on the input image using the given 3x3 matrix.

Data Structures

- struct **Conv3x3Param**
*Parameter structure of the **Convolution 3x3** filter.*

Functions

- void **SVU_SYM()** **svuConv3x3** (**SippFilter** *fptr)
*Shave function of the **Convolution 3x3** filter.*
- **SHAVE_SYM_EXPORT** (**svuConv3x3**)

5.67.1 Detailed Description

This filter performs a convolution on the input image using the given 3x3 matrix.

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 3x3

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/convolution3x3/arch//shave/src/convolution3x3.asm

5.67.2 Function Documentation

SHAVE_SYM_EXPORT (**svuConv3x3**)

void **SVU_SYM()** **svuConv3x3** (**SippFilter** * fptr)

Shave function of the **Convolution 3x3** filter.

Referenced by createTripleConv3x3().

5.68 Convolution 3x3 Fp16ToFp16

This filter performs a convolution on the fp16 input image using the given 3x3 matrix.

Data Structures

- struct **Conv3x3Fp16ToFp16Param**
*Parameter structure of the **Convolution 3x3 Fp16ToFp16** filter.*

Functions

- void **SVU_SYM()** **svuConv3x3Fp16ToFp16** (**SippFilter** *fptr)
*Shave function of the **Convolution 3x3 Fp16ToFp16** filter.*
- **SHAVE_SYM_EXPORT** (**svuConv3x3Fp16ToFp16**)

5.68.1 Detailed Description

This filter performs a convolution on the fp16 input image using the given 3x3 matrix.

Output data type(s):

half

Inputs:

- datatypes: half; kernels: 3x3

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/convolution3x3Fp16ToFp16/arch//shave/src/convolution3x3-Fp16ToFp16.asm

5.68.2 Function Documentation

SHAVE_SYM_EXPORT (**svuConv3x3Fp16ToFp16**)

void **SVU_SYM()** **svuConv3x3Fp16ToFp16** (**SippFilter** * fptr)

Shave function of the **Convolution 3x3 Fp16ToFp16** filter.

5.69 Convolution 5x1

This filter performs a convolution on the input image using the given 5x1 matrix.

Data Structures

- struct **Conv5x1Param**
*Parameter structure of the **Convolution 5x1** filter.*

Functions

- void **SVU_SYM()** **svuConv5x1** (**SippFilter** *fptr)
*Shave function of the **Convolution 5x1** filter.*
- **SHAVE_SYM_EXPORT** (**svuConv5x1**)

5.69.1 Detailed Description

This filter performs a convolution on the input image using the given 5x1 matrix.

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 5x1

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/convolution5x1/arch//shave/src/convolution5x1.asm

5.69.2 Function Documentation

SHAVE_SYM_EXPORT (**svuConv5x1**)

void **SVU_SYM()** **svuConv5x1** (**SippFilter** * fptr)

Shave function of the **Convolution 5x1** filter.

5.70 Convolution 5x1 Fp16ToFp16

This filter performs a convolution on the fp16 input image using the given 5x1 matrix.

Data Structures

- struct **Conv5x1Fp16ToFp16Param**
*Parameter structure of the **Convolution 5x1 Fp16ToFp16** filter.*

Functions

- void **SVU_SYM()** **svuConv5x1Fp16ToFp16** (**SippFilter** *fptr)
*Shave function of the **Convolution 5x1 Fp16ToFp16** filter.*
- **SHAVE_SYM_EXPORT** (**svuConv5x1Fp16ToFp16**)

5.70.1 Detailed Description

This filter performs a convolution on the fp16 input image using the given 5x1 matrix.

Output data type(s):

half

Inputs:

- datatypes: half; kernels: 5x1

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/convolution5x1Fp16ToFp16/arch//shave/src/convolution5x1-Fp16ToFp16.asm

5.70.2 Function Documentation

SHAVE_SYM_EXPORT (**svuConv5x1Fp16ToFp16**)

void **SVU_SYM()** **svuConv5x1Fp16ToFp16** (**SippFilter** * fptr)

Shave function of the **Convolution 5x1 Fp16ToFp16** filter.

5.71 Convolution 5x5

This filter performs a convolution on the input image using the given 5x5 matrix.

Data Structures

- struct **Conv5x5Param**
Parameter structure of the Convolution 5x5 filter.

Functions

- void **SVU_SYM()** **svuConv5x5** (**SippFilter** *fptr)
Shave function of the Convolution 5x5 filter.
- **SHAVE_SYM_EXPORT** (**svuConv5x5**)

5.71.1 Detailed Description

This filter performs a convolution on the input image using the given 5x5 matrix.

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 5x5

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/convolution5x5/arch//shave/src/convolution5x5.asm

5.71.2 Function Documentation

SHAVE_SYM_EXPORT (**svuConv5x5**)

void **SVU_SYM()** **svuConv5x5** (**SippFilter** * fptr)

Shave function of the Convolution 5x5 filter.

5.72 Convolution 5x5 Fp16ToFp16

This filter performs a convolution on the fp16 input image using the given 5x5 matrix.

Data Structures

- struct **Conv5x5Fp16ToFp16Param**
*Parameter structure of the **Convolution 5x5 Fp16ToFp16** filter.*

Functions

- void **SVU_SYM()** **svuConv5x5Fp16ToFp16** (**SippFilter** *fptr)
*Shave function of the **Convolution 5x5 Fp16ToFp16** filter.*
- **SHAVE_SYM_EXPORT** (**svuConv5x5Fp16ToFp16**)

5.72.1 Detailed Description

This filter performs a convolution on the fp16 input image using the given 5x5 matrix.

Output data type(s):

half, fp16

Inputs:

- datatypes: half, fp16; kernels: 5x5

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/convolution5x5Fp16ToFp16/arch//shave/src/convolution5x5-Fp16ToFp16.asm

5.72.2 Function Documentation

SHAVE_SYM_EXPORT (**svuConv5x5Fp16ToFp16**)

void **SVU_SYM()** **svuConv5x5Fp16ToFp16** (**SippFilter** * fptr)

Shave function of the **Convolution 5x5 Fp16ToFp16** filter.

5.73 Convolution 7x1

This filter performs a convolution on the input image using the given 7x1 matrix.

Data Structures

- struct **Conv7x1Param**
*Parameter structure of the **Convolution 7x1** filter.*

Functions

- void **SVU_SYM()** **svuConv7x1** (**SippFilter** *fptr)
*Shave function of the **Convolution 7x1** filter.*
- **SHAVE_SYM_EXPORT** (**svuConv7x1**)

5.73.1 Detailed Description

This filter performs a convolution on the input image using the given 7x1 matrix.

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 7x1

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/convolution7x1/arch//shave/src/convolution7x1.asm

5.73.2 Function Documentation

SHAVE_SYM_EXPORT (**svuConv7x1**)

void **SVU_SYM()** **svuConv7x1** (**SippFilter** * fptr)

Shave function of the **Convolution 7x1** filter.

5.74 Convolution 7x1 Fp16ToFp16

This filter performs a convolution on the fp16 input image using the given 7x1 matrix.

Data Structures

- struct **Conv7x1Fp16ToFp16Param**
Parameter structure of the Convolution 7x1 Fp16ToFp16 filter.

Functions

- void **SVU_SYM()** **svuConv7x1Fp16ToFp16** (**SippFilter** *fptr)
Shave function of the Convolution 7x1 Fp16ToFp16 filter.
- **SHAVE_SYM_EXPORT** (**svuConv7x1Fp16ToFp16**)

5.74.1 Detailed Description

This filter performs a convolution on the fp16 input image using the given 7x1 matrix.

Output data type(s):

half

Inputs:

- datatypes: half; kernels: 7x1

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/convolution7x1Fp16ToFp16/arch//shave/src/convolution7x1-Fp16ToFp16.asm

5.74.2 Function Documentation

SHAVE_SYM_EXPORT (**svuConv7x1Fp16ToFp16**)

void **SVU_SYM()** **svuConv7x1Fp16ToFp16** (**SippFilter** * fptr)

Shave function of the **Convolution 7x1 Fp16ToFp16** filter.

5.75 Convolution 7x7

This filter performs a convolution on the input image using the given 7x7 matrix.

Data Structures

- struct **Conv7x7Param**
Parameter structure of the Convolution 7x7 filter.

Functions

- void **SVU_SYM()** **svuConv7x7** (**SippFilter** *fptr)
Shave function of the Convolution 7x7 filter.
- **SHAVE_SYM_EXPORT** (**svuConv7x7**)

5.75.1 Detailed Description

This filter performs a convolution on the input image using the given 7x7 matrix.

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 7x7

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/convolution7x7/arch//shave/src/convolution7x7.asm

5.75.2 Function Documentation

SHAVE_SYM_EXPORT (**svuConv7x7**)

void **SVU_SYM()** **svuConv7x7** (**SippFilter** * fptr)

Shave function of the Convolution 7x7 filter.

5.76 Convolution 7x7 Fp16ToFp16

This filter performs a convolution on the fp16 input image using the given 7x7 matrix.

Data Structures

- struct **Conv7x7ParamFp16ToFp16**
*Parameter structure of the **Convolution 7x7 Fp16ToFp16** filter.*

Functions

- void **SVU_SYM()** **svuConv7x7Fp16ToFp16** (**SippFilter** *fptr)
*Shave function of the **Convolution 7x7 Fp16ToFp16** filter.*
- **SHAVE_SYM_EXPORT** (**svuConv7x7Fp16ToFp16**)

5.76.1 Detailed Description

This filter performs a convolution on the fp16 input image using the given 7x7 matrix.

Output data type(s):

half

Inputs:

- datatypes: half; kernels: 7x7

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/convolution7x7Fp16ToFp16/arch//shave/src/convolution7x7-Fp16ToFp16.asm

5.76.2 Function Documentation

SHAVE_SYM_EXPORT (**svuConv7x7Fp16ToFp16**)

void **SVU_SYM()** **svuConv7x7Fp16ToFp16** (**SippFilter** * fptr)

Shave function of the **Convolution 7x7 Fp16ToFp16** filter.

5.77 Convolution 7x7 Fp16ToU8

This filter performs a convolution on the fp16 input image using the given 7x7 matrix and stores the result in the U8 output image.

Data Structures

- struct `Conv7x7ParamFp16ToU8`

Parameter structure of the `Convolution 7x7 Fp16ToU8` filter.

Functions

- void `SVU_SYM()` `svuConv7x7Fp16ToU8` (`SippFilter` *fptr)
Shave function of the `Convolution 7x7 Fp16ToU8` filter.
- `SHAVE_SYM_EXPORT` (`svuConv7x7Fp16ToU8`)

5.77.1 Detailed Description

This filter performs a convolution on the fp16 input image using the given 7x7 matrix and stores the result in the U8 output image.

Preserve:

`imgSize`

Output data type(s):

`UInt8`

Inputs:

- datatypes: half; kernels: 7x7

Path to external ASM file(s) used in the filter implementation:

`/components/kernelLib/MvCV/kernels/convolution7x7Fp16ToU8/arch//shave/src/convolution7x7-Fp16ToU8.asm`

5.77.2 Function Documentation

`SHAVE_SYM_EXPORT` (`svuConv7x7Fp16ToU8`)

void `SVU_SYM()` `svuConv7x7Fp16ToU8` (`SippFilter` * fptr)

Shave function of the `Convolution 7x7 Fp16ToU8` filter.

5.78 Convolution 9x1

This filter performs a convolution on the input image using the given 9x1 matrix.

Data Structures

- struct **Conv9x1Param**
*Parameter structure of the **Convolution 9x1** filter.*

Functions

- void **SVU_SYM()** **svuConv9x1** (**SippFilter** *fptr)
*Shave function of the **Convolution 9x1** filter.*
- **SHAVE_SYM_EXPORT** (**svuConv9x1**)

5.78.1 Detailed Description

This filter performs a convolution on the input image using the given 9x1 matrix.

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 9x1

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/convolution9x1/arch//shave/src/convolution9x1.asm

5.78.2 Function Documentation

SHAVE_SYM_EXPORT (**svuConv9x1**)

void **SVU_SYM()** **svuConv9x1** (**SippFilter** * fptr)

Shave function of the **Convolution 9x1** filter.

5.79 Convolution 9x9

This filter performs a convolution on the input image using the given 9x9 matrix.

Data Structures

- struct **Conv9x9Param**
*Parameter structure of the **Convolution 9x9** filter.*

Functions

- void **SVU_SYM()** **svuConv9x9** (**SippFilter** *fptr)
*Shave function of the **Convolution 9x9** filter.*
- **SHAVE_SYM_EXPORT** (**svuConv9x9**)

5.79.1 Detailed Description

This filter performs a convolution on the input image using the given 9x9 matrix.

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 9x9

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/convolution9x9/arch//shave/src/convolution9x9.asm

5.79.2 Function Documentation

SHAVE_SYM_EXPORT (**svuConv9x9**)

void **SVU_SYM()** **svuConv9x9** (**SippFilter** * fptr)

Shave function of the **Convolution 9x9** filter.

5.80 Convolution 9x9 Fp16ToFp16

This filter performs a convolution on the fp16 input image using the given 9x9 matrix.

Data Structures

- struct **Conv9x9Fp16ToFp16Param**
*Parameter structure of the **Convolution 9x9 Fp16ToFp16** filter.*

Functions

- void **SVU_SYM()** **svuConv9x9Fp16ToFp16** (**SippFilter** *fptr)
*Shave function of the **Convolution 9x9 Fp16ToFp16** filter.*
- **SHAVE_SYM_EXPORT** (**svuConv9x9Fp16ToFp16**)

5.80.1 Detailed Description

This filter performs a convolution on the fp16 input image using the given 9x9 matrix.

Output data type(s):

half

Inputs:

- datatypes: half; kernels: 9x9

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/convolution9x9Fp16ToFp16/arch//shave/src/convolution9x9-Fp16ToFp16.asm

5.80.2 Function Documentation

SHAVE_SYM_EXPORT (**svuConv9x9Fp16ToFp16**)

void **SVU_SYM()** **svuConv9x9Fp16ToFp16** (**SippFilter** * fptr)

Shave function of the **Convolution 9x9 Fp16ToFp16** filter.

5.81 Convert 16bpp To 8bpp

This filter performs conversion from 16 bpp input array to 8 bpp.

Functions

- void **SVU_SYM()** **svuConvert16bppTo8bpp** (**SippFilter** *fptr)
Shave function of the [Convert 16bpp To 8bpp](#) filter.
- **SHAVE_SYM_EXPORT** (**svuConvert16bppTo8bpp**)

5.81.1 Detailed Description

This filter performs conversion from 16 bpp input array to 8 bpp.

Output data type(s):

UInt8

Preserve:

imgSize, numPlanes

Inputs:

- datatypes: UInt16; kernels: 1x1

5.81.2 Function Documentation

SHAVE_SYM_EXPORT (**svuConvert16bppTo8bpp**)

void **SVU_SYM()** **svuConvert16bppTo8bpp** (**SippFilter** * fptr)

Shave function of the [Convert 16bpp To 8bpp](#) filter.

5.82 Convert F16 To U8

This filter performs F16 to U8 conversion.

Functions

- void **SVU_SYM()** **svuConvertF16ToU8** (**SippFilter** *fptr)
Shave function of the [Convert F16 To U8](#) filter.
- **SHAVE_SYM_EXPORT** (**svuConvertF16ToU8**)

5.82.1 Detailed Description

This filter performs F16 to U8 conversion.

Preserve:

numPlanes, imgSize

Output data type(s):

UInt8

Inputs:

- datatypes: half; kernels: 1x1

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/convert_fp16_u8/arch//shave/src/convert_fp16_u8.asm

5.82.2 Function Documentation

SHAVE_SYM_EXPORT (**svuConvertF16ToU8**)

void **SVU_SYM()** **svuConvertF16ToU8** (**SippFilter** * fptr)

Shave function of the [Convert F16 To U8](#) filter.

5.83 12Bpp to 8Bpp conversion

This filter performs a conversion on the input image from 12bpp to 8bpp.

Functions

- void **SVU_SYM()** **svuConvertFrom12BppTo8Bpp** (**SippFilter** *fptr)
Shave function of the 12Bpp to 8Bpp conversion filter.
- **SHAVE_SYM_EXPORT** (**svuConvertFrom12BppTo8Bpp**)

5.83.1 Detailed Description

This filter performs a conversion on the input image from 12bpp to 8bpp.

Preserve:

numPlanes, dataType

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 1x1

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/convertFrom12BppTo8Bpp/arch//shave/src/convertFrom12-BppTo8Bpp.asm

5.83.2 Function Documentation

SHAVE_SYM_EXPORT (**svuConvertFrom12BppTo8Bpp**)

void **SVU_SYM()** **svuConvertFrom12BppTo8Bpp** (**SippFilter** * fptr)

Shave function of the 12Bpp to 8Bpp conversion filter.

5.84 Convert Fp16 to U16

This filter performs FP16 to U16 conversion.

Functions

- void **SVU_SYM()** **svuConvertPFp16U16** (**SippFilter** *fptr)
Shave function of the [Convert Fp16 to U16](#) filter.
- **SHAVE_SYM_EXPORT** (**svuConvertPFp16U16**)

5.84.1 Detailed Description

This filter performs FP16 to U16 conversion.

Preserve:

numPlanes, imgSize

Output data type(s):

UInt16

Inputs:

- datatypes: half; kernels: 1x1

5.84.2 Function Documentation

SHAVE_SYM_EXPORT (**svuConvertPFp16U16**)

void **SVU_SYM()** **svuConvertPFp16U16** (**SippFilter** * fptr)

Shave function of the [Convert Fp16 to U16](#) filter.

5.85 Convert U16 to Fp16

This filter performs U16 to Fp16 conversion.

Functions

- void **SVU_SYM()** **svuConvertPU16Fp16** (**SippFilter** *fptr)
*Shave function of the **Convert U16 to Fp16** filter.*
- **SHAVE_SYM_EXPORT** (**svuConvertPU16Fp16**)

5.85.1 Detailed Description

This filter performs U16 to Fp16 conversion.

Preserve:

numPlanes, imgSize

Output data type(s):

half

Inputs:

- datatypes: UInt16; kernels: 1x1

5.85.2 Function Documentation

SHAVE_SYM_EXPORT (**svuConvertPU16Fp16**)

void **SVU_SYM()** **svuConvertPU16Fp16** (**SippFilter** * fptr)

Shave function of the **Convert U16 to Fp16** filter.

5.86 Convert U8 To F16

This filter performs a conversion from U8 to F16.

Functions

- void **SVU_SYM()** **svuConvertU8ToF16** (**SippFilter** *fptr)
*Shave function of the **Convert U8 To F16** filter.*
- **SHAVE_SYM_EXPORT** (**svuConvertU8ToF16**)

5.86.1 Detailed Description

This filter performs a conversion from U8 to F16.

Preserve:

numPlanes, imgSize

Output data type(s):

half

Inputs:

- datatypes: UInt8; kernels: 1x1

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/convert_u8_fp16/arch//shave/src/convert_u8_fp16.asm

5.86.2 Function Documentation

SHAVE_SYM_EXPORT (**svuConvertU8ToF16**)

void **SVU_SYM()** **svuConvertU8ToF16** (**SippFilter** * fptr)

Shave function of the **Convert U8 To F16** filter.

5.87 YUV400 to YUV422 conversion

This filter performs a color conversion from a simple yuv400 to yuv422 with value for chromas 0x80.

Functions

- void **SVU_SYM()** **svuConvertYUV400ToYUV422** (**SippFilter** *fptr)
Shave function of the YUV400 to YUV422 conversion filter.
- **SHAVE_SYM_EXPORT** (**svuConvertYUV400ToYUV422**)

5.87.1 Detailed Description

This filter performs a color conversion from a simple yuv400 to yuv422 with value for chromas 0x80.

Output data type(s):

UInt16

Preserve:

Inputs:

- datatypes: UInt8; kernels:1x1

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/convertYUV400ToYUV422/arch//shave/src/convertYUV400ToYUV422.asm

5.87.2 Function Documentation

SHAVE_SYM_EXPORT (**svuConvertYUV400ToYUV422**)

void **SVU_SYM()** **svuConvertYUV400ToYUV422** (**SippFilter** * fptr)

Shave function of the **YUV400 to YUV422 conversion** filter.

5.88 Generic Convolution

This filter performs a generic convolution on the input image using the kernel size given by the user.

Data Structures

- struct **ConvGenericParam**
*Parameter structure of the **Generic Convolution** filter.*

Functions

- void **SVU_SYM()** **svuConvGeneric** (**SippFilter** *fptr)
*Shave function of the **Generic Convolution** filter.*
- **SHAVE_SYM_EXPORT** (**svuConvGeneric**)

5.88.1 Detailed Description

This filter performs a generic convolution on the input image using the kernel size given by the user.

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels:

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/convolution/arch//shave/src/convolution.asm

5.88.2 Function Documentation

SHAVE_SYM_EXPORT (**svuConvGeneric**)

void **SVU_SYM()** **svuConvGeneric** (**SippFilter** * fptr)

Shave function of the **Generic Convolution** filter.

5.89 Convolution Separable 11x11

This filter performs a separable convolution on the input image using the given 11x11 matrix.

Data Structures

- struct **ConvSeparable11x11Param**
Parameter structure of the Convolution Separable 11x11 filter.

Functions

- void **SVU_SYM()** **svuConvSeparable11x11** (**SippFilter** *fptr)
Shave function of the Convolution Separable 11x11 filter.
- **SHAVE_SYM_EXPORT** (**svuConvSeparable11x11**)

5.89.1 Detailed Description

This filter performs a separable convolution on the input image using the given 11x11 matrix.

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 11x11

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/convSeparable11x11/arch//shave/src/convSeparable11x11.asm

5.89.2 Function Documentation

SHAVE_SYM_EXPORT (**svuConvSeparable11x11**)

void **SVU_SYM()** **svuConvSeparable11x11** (**SippFilter** * fptr)

Shave function of the Convolution Separable 11x11 filter.

5.90 Convolution Separable 11x11 Fp16ToFp16

This filter performs a separable convolution on the fp16 input image using the given 11x11 matrix.

Data Structures

- struct **ConvSeparable11x11Fp16ToFp16Param**
*Parameter structure of the **Convolution Separable 11x11 Fp16ToFp16** filter.*

Functions

- void **SVU_SYM()** **svuConvSeparable11x11Fp16ToFp16** (**SippFilter** *fptr)
*Shave function of the **Convolution Separable 11x11 Fp16ToFp16** filter.*
- **SHAVE_SYM_EXPORT** (**svuConvSeparable11x11Fp16ToFp16**)

5.90.1 Detailed Description

This filter performs a separable convolution on the fp16 input image using the given 11x11 matrix.

Output data type(s):

half

Inputs:

- datatypes: half; kernels: 11x11

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/convSeparable11x11Fp16ToFp16/arch//shave/src/conv-Separable11x11Fp16ToFp16.asm

5.90.2 Function Documentation

SHAVE_SYM_EXPORT (**svuConvSeparable11x11Fp16ToFp16**)

void **SVU_SYM()** **svuConvSeparable11x11Fp16ToFp16** (**SippFilter** * fptr)

Shave function of the **Convolution Separable 11x11 Fp16ToFp16** filter.

5.91 Convolution Separable 3x3

This filter performs a separable convolution on the fp16 input image using the given 3x3 matrix.

Data Structures

- struct **ConvSeparable3x3Param**
*Parameter structure of the **Convolution Separable 3x3** filter.*

Functions

- void **SVU_SYM()** **svuConvSeparable3x3** (**SippFilter** *fptr)
*Shave function of the **Convolution Separable 3x3** filter.*
- **SHAVE_SYM_EXPORT** (**svuConvSeparable3x3**)

5.91.1 Detailed Description

This filter performs a separable convolution on the fp16 input image using the given 3x3 matrix.

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 3x3

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/convSeparable3x3/arch//shave/src/convSeparable3x3.asm

5.91.2 Function Documentation

SHAVE_SYM_EXPORT (**svuConvSeparable3x3**)

void **SVU_SYM()** **svuConvSeparable3x3** (**SippFilter** * fptr)

Shave function of the **Convolution Separable 3x3** filter.

5.92 Convolution Separable 3x3 Fp16ToFp16

This filter performs a separable convolution on the fp16 input image using the given 3x3 matrix.

Data Structures

- struct **ConvSeparable3x3Fp16ToFp16Param**
*Parameter structure of the **Convolution Separable 3x3 Fp16ToFp16** filter.*

Functions

- void **SVU_SYM()** **svuConvSeparable3x3Fp16ToFp16** (**SippFilter** *fptr)
*Shave function of the **Convolution Separable 3x3 Fp16ToFp16** filter.*
- **SHAVE_SYM_EXPORT** (**svuConvSeparable3x3Fp16ToFp16**)

5.92.1 Detailed Description

This filter performs a separable convolution on the fp16 input image using the given 3x3 matrix.

Output data type(s):

half

Inputs:

- datatypes: half; kernels: 3x3

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/convSeparable3x3Fp16ToFp16/arch//shave/src/convSeparable3x3-Fp16ToFp16.asm

5.92.2 Function Documentation

SHAVE_SYM_EXPORT (**svuConvSeparable3x3Fp16ToFp16**)

void **SVU_SYM()** **svuConvSeparable3x3Fp16ToFp16** (**SippFilter** * fptr)

Shave function of the **Convolution Separable 3x3 Fp16ToFp16** filter.

5.93 Convolution Separable 5x5

This filter performs a separable convolution on the fp16 input image using the given 5x5 matrix.

Data Structures

- struct **ConvSeparable5x5Param**
*Parameter structure of the **Convolution Separable 5x5** filter.*

Functions

- void **SVU_SYM()** **svuConvSeparable5x5** (**SippFilter** *fptr)
*Shave function of the **Convolution Separable 5x5** filter.*
- **SHAVE_SYM_EXPORT** (**svuConvSeparable5x5**)

5.93.1 Detailed Description

This filter performs a separable convolution on the fp16 input image using the given 5x5 matrix.

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 5x5

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/convSeparable5x5/arch//shave/src/convSeparable5x5.asm

5.93.2 Function Documentation

SHAVE_SYM_EXPORT (**svuConvSeparable5x5**)

void **SVU_SYM()** **svuConvSeparable5x5** (**SippFilter** * fptr)

Shave function of the **Convolution Separable 5x5** filter.

5.94 Convolution Separable 5x5 Fp16ToFp16

This filter performs a separable convolution on the fp16 input image using the given 5x5 matrix.

Data Structures

- struct **ConvSeparable5x5Fp16ToFp16Param**
*Parameter structure of the **Convolution Separable 5x5 Fp16ToFp16** filter.*

Functions

- void **SVU_SYM()** **svuConvSeparable5x5Fp16ToFp16** (**SippFilter** *fptr)
*Shave function of the **Convolution Separable 5x5 Fp16ToFp16** filter.*
- **SHAVE_SYM_EXPORT** (**svuConvSeparable5x5Fp16ToFp16**)

5.94.1 Detailed Description

This filter performs a separable convolution on the fp16 input image using the given 5x5 matrix.

Output data type(s):

half

Inputs:

- datatypes: half; kernels: 5x5

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/convSeparable5x5Fp16ToFp16/arch//shave/src/convSeparable5x5-Fp16ToFp16.asm

5.94.2 Function Documentation

SHAVE_SYM_EXPORT (**svuConvSeparable5x5Fp16ToFp16**)

void **SVU_SYM()** **svuConvSeparable5x5Fp16ToFp16** (**SippFilter** * fptr)

Shave function of the **Convolution Separable 5x5 Fp16ToFp16** filter.

5.95 Convolution Separable 7x7

This filter performs a separable convolution on the fp16 input image using the given 7x7 matrix.

Data Structures

- struct **ConvSeparable7x7Param**
Parameter structure of the Convolution Separable 7x7 filter.

Functions

- void **SVU_SYM()** **svuConvSeparable7x7** (**SippFilter** *fptr)
Shave function of the Convolution Separable 7x7 filter.
- **SHAVE_SYM_EXPORT** (**svuConvSeparable7x7**)

5.95.1 Detailed Description

This filter performs a separable convolution on the fp16 input image using the given 7x7 matrix.

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 7x7

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/convSeparable7x7/arch//shave/src/convSeparable7x7.asm

5.95.2 Function Documentation

SHAVE_SYM_EXPORT (**svuConvSeparable7x7**)

void **SVU_SYM()** **svuConvSeparable7x7** (**SippFilter** * fptr)

Shave function of the Convolution Separable 7x7 filter.

5.96 Convolution Separable 7x7 Fp16ToFp16

This filter performs a separable convolution on the fp16 input image using the given 7x7 matrix.

Data Structures

- struct **ConvSeparable7x7Fp16ToFp16Param**
Parameter structure of the Convolution Separable 7x7 Fp16ToFp16 filter.

Functions

- void **SVU_SYM()** **svuConvSeparable7x7Fp16ToFp16** (**SippFilter** *fptr)
Shave function of the Convolution Separable 7x7 Fp16ToFp16 filter.
- **SHAVE_SYM_EXPORT** (**svuConvSeparable7x7Fp16ToFp16**)

5.96.1 Detailed Description

This filter performs a separable convolution on the fp16 input image using the given 7x7 matrix.

Output data type(s):

half

Inputs:

- datatypes: half; kernels: 7x7

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/convSeparable7x7Fp16ToFp16/arch//shave/src/convSeparable7x7-Fp16ToFp16.asm

5.96.2 Function Documentation

SHAVE_SYM_EXPORT (**svuConvSeparable7x7Fp16ToFp16**)

void **SVU_SYM()** **svuConvSeparable7x7Fp16ToFp16** (**SippFilter** * fptr)

Shave function of the **Convolution Separable 7x7 Fp16ToFp16** filter.

5.97 Convolution Separable 9x9

This filter performs a separable convolution on the fp16 input image using the given 9x9 matrix.

Data Structures

- struct **ConvSeparable9x9Param**
Parameter structure of the Convolution Separable 9x9 filter.

Functions

- void **SVU_SYM()** **svuConvSeparable9x9** (**SippFilter** *fptr)
Shave function of the Convolution Separable 9x9 filter.
- **SHAVE_SYM_EXPORT** (**svuConvSeparable9x9**)

5.97.1 Detailed Description

This filter performs a separable convolution on the fp16 input image using the given 9x9 matrix.

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 9x9

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/convSeparable9x9/arch//shave/src/convSeparable9x9.asm

5.97.2 Function Documentation

SHAVE_SYM_EXPORT (**svuConvSeparable9x9**)

void **SVU_SYM()** **svuConvSeparable9x9** (**SippFilter** * fptr)

Shave function of the **Convolution Separable 9x9** filter.

5.98 Convolution Separable 9x9 Fp16ToFp16

This filter performs a separable convolution on the fp16 input image using the given 9x9 matrix.

Data Structures

- struct **ConvSeparable9x9Fp16ToFp16Param**
*Parameter structure of the **Convolution Separable 9x9 Fp16ToFp16** filter.*

Functions

- void **SVU_SYM()** **svuConvSeparable9x9Fp16ToFp16** (**SippFilter** *fptr)
*Shave function of the **Convolution Separable 9x9 Fp16ToFp16** filter.*
- **SHAVE_SYM_EXPORT** (**svuConvSeparable9x9Fp16ToFp16**)

5.98.1 Detailed Description

This filter performs a separable convolution on the fp16 input image using the given 9x9 matrix.

Output data type(s):

half

Inputs:

- datatypes: half; kernels: 9x9

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/convSeparable9x9Fp16ToFp16/arch//shave/src/convSeparable9x9-Fp16ToFp16.asm

5.98.2 Function Documentation

SHAVE_SYM_EXPORT (**svuConvSeparable9x9Fp16ToFp16**)

void **SVU_SYM()** **svuConvSeparable9x9Fp16ToFp16** (**SippFilter** * fptr)

Shave function of the **Convolution Separable 9x9 Fp16ToFp16** filter.

5.99 Convert to YUV444

This filter performs line conversion to YUV444.

Functions

- void **SVU_SYM()** **svuRgbYuv444** (**SippFilter** *fptr)
Shave function of the [Convert to YUV444](#) filter.
- **SHAVE_SYM_EXPORT** (**svuRgbYuv444**)

5.99.1 Detailed Description

This filter performs line conversion to YUV444.

Output data type(s):

UInt8

Preserve:

imgSize, numPlanes

Inputs:

- datatypes: half; kernels: 1x1

5.99.2 Function Documentation

SHAVE_SYM_EXPORT (**svuRgbYuv444**)

void **SVU_SYM()** **svuRgbYuv444** (**SippFilter** * fptr)

Shave function of the [Convert to YUV444](#) filter.

5.100 Copy

This filter copies input image to output.

Functions

- void **SVU_SYM()** **svuCopy** (**SippFilter** *fptr)
Shave function of the Copy filter.
- **SHAVE_SYM_EXPORT** (**svuCopy**)

5.100.1 Detailed Description

This filter copies input image to output.

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 1x1

5.100.2 Function Documentation

SHAVE_SYM_EXPORT (**svuCopy**)

void **SVU_SYM()** **svuCopy** (**SippFilter** * fptr)

Shave function of the **Copy** filter.

5.101 Corner Min Eigenvalue

This filter performs corner detection using minimum eigenvalue algorithm for a line.

Functions

- void **SVU_SYM()** **svuCornerMinEigenVal** (**SippFilter** *fptr)
Shave function of the [Corner Min Eigenvalue](#) filter.
- **SHAVE_SYM_EXPORT** (**svuCornerMinEigenVal**)

5.101.1 Detailed Description

This filter performs corner detection using minimum eigenvalue algorithm for a line.

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 5x5

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/cornerMinEigenVal/arch//shave/src/cornerMinEigenVal.asm

5.101.2 Function Documentation

SHAVE_SYM_EXPORT (**svuCornerMinEigenVal**)

void **SVU_SYM()** **svuCornerMinEigenVal** (**SippFilter** * fptr)

Shave function of the [Corner Min Eigenvalue](#) filter.

5.102 Corner Min Eigenvalue Patched

This filter performs corner detection using minimum eigenvalue algorithm for a single pixel.

Functions

- void **SVU_SYM()** **svuCornerMinEigenValpatched** (**SippFilter** *fptr)
*Shave function of the **Corner Min Eigenvalue Patched** filter.*
- **SHAVE_SYM_EXPORT** (**svuCornerMinEigenValpatched**)

5.102.1 Detailed Description

This filter performs corner detection using minimum eigenvalue algorithm for a single pixel.

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 5x5

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/cornerMinEigenVal_patched/arch//shave/src/cornerMinEigenVal_patched.asm

5.102.2 Function Documentation

SHAVE_SYM_EXPORT (**svuCornerMinEigenValpatched**)

void **SVU_SYM()** **svuCornerMinEigenValpatched** (**SippFilter** * fptr)

Shave function of the **Corner Min Eigenvalue Patched** filter.

5.103 Crop

This filter performs a crop on the input image.

This filter performs a crop on the input image. This filter performs a crop on the input image and converts RGB planar to interleaved.

Flags:

SIPP_CROP

5.104 CV

Modules

- **Box Filter 11x11**
This filter applies the box filter on the source image using the specified structuring element.
- **Box Filter 13x13**
This filter applies the box filter on the source image using the specified structuring element.
- **Box Filter 15x15**
This filter applies the box filter on the source image using the specified structuring element.
- **Box Filter 3x3**
This filter applies the box filter on the source image using the specified structuring element.
- **Box Filter 5x5**
This filter applies the box filter on the source image using the specified structuring element.
- **Box Filter 7x7**
This filter applies the box filter on the source image using the specified structuring element.
- **Box Filter 9x9**
This filter applies the box filter on the source image using the specified structuring element.
- **Canny Edge Detection**
The filter finds edges in the input image and marks them in the output map edges using the Canny algorithm.
- **Corner Min Eigenvalue**
This filter performs corner detection using minimum eigenvalue algorithm for a line.
- **Corner Min Eigenvalue Patched**
This filter performs corner detection using minimum eigenvalue algorithm for a single pixel.
- **Crop**
This filter performs a crop on the input image.
- **Dilate 3x3**
This filter performs a dilate on the source image using the specified structuring element.
- **Dilate 5x5**
This filter performs a dilate on the source image using the specified structuring element.
- **Dilate 7x7**
This filter performs a dilate on the source image using the specified structuring element.
- **Erode 3x3**
This filter applies the erode filter on the source image using the specified structuring element.
- **Erode 5x5**
This filter applies the erode filter on the source image using the specified structuring element.
- **Erode 7x7**
This filter applies the erode filter on the source image using the specified structuring element.
- **Fast9M2**
The filter performs the Fast9 for Myriad 2 corner detection.
- **Fast9ScoreCv**
The filter performs the Fast9 for Myriad 2 corner detection.
- **Generic Box Filter**
This filter calculates average on variable kernel size, on kernel size number of input lines.

- **Generic Dilate**

This filter performs a generic dilate on the input image using the kernel size given by the user.

- **Homography**

This filter performs a homography transformation.

- **Integral Image Square Sum(U32)**

This filter makes the sum of all pixels before it and on the left of it's column(this particular case makes square sum of pixels in U32 format).

- **Integral Image Square Sum(f32)**

This filter makes the sum of all pixels before it and on the left of it's column(this particular case makes square sum of pixels in f32 format).

- **Integral Image Sum(U16toU32)**

This filter makes the sum of all pixels before it and on the left of it's column(this particular case makes square sum of pixels in U32 format).

- **Integral Image Sum(U32)**

This filter makes the sum of all pixels before it and on the left of it's column(this particular case makes square sum of pixels in U32 format).

- **Integral Image Sum(f32)**

This filter makes the sum of all pixels before it and on the left of it's column(this particular case makes square sum of pixels in f32 format).

- **Min/Max Value**

This filter computes the minimum and the maximum value of a given input image.

- **Min/Max Value Position**

This filter computes the minimum and the maximum value of a given input line and their position.

- **Pixel Position**

This filter returns the position of a given pixel value.

- **Sobel**

This filter performs sobel edge detection operator.

- **Threshold**

This filter computes the output image based on a threshold value and a threshold type.

- **Threshold Binary Range**

This filter sets output to 0xFF if pixel value is in specified range, otherwise output is 0.

- **Threshold Binary U8**

This filter sets output to 0 if threshold value is less then input value and to 0xFF if threshold value is greater then input value.

- **ThresholdFilter**

This filter computes the output image based on a thresholdFilter value.

- **harrisResponse**

This filter computes Harris response over a patch of the image with a radius of 3.

- **interpolatePixelBilinear**

This filter makes the bilinear interpolation of four pixels.

- **scharr_fp16**

This kernel performs scharr edge detection operator.

Data Structures

- struct **CropParam**

*Parameter structure of the **Crop** filter.*

Functions

- void **SVU_SYM()** **svuCrop** (**SippFilter** *fptr)
*Shave function of the **Crop** filter.*
- **SHAVE_SYM_EXPORT** (**svuCrop**)
- void **SVU_SYM()** **svuCropCvtPlaneMode** (**SippFilter** *fptr)
*Shave function of the **Crop** filter.*
- **SHAVE_SYM_EXPORT** (**svuCropCvtPlaneMode**)

5.104.1 Detailed Description

Output data type(s):

UInt8

Preserve:

numPlanes, dataType

Inputs:

- datatypes: UInt8; kernels: 1x1

5.104.2 Function Documentation

SHAVE_SYM_EXPORT (**svuCropCvtPlaneMode**)

SHAVE_SYM_EXPORT (**svuCrop**)

void **SVU_SYM()** **svuCrop** (**SippFilter** * fptr)

Shave function of the **Crop** filter.

void **SVU_SYM()** **svuCropCvtPlaneMode** (**SippFilter** * fptr)

Shave function of the **Crop** filter.

5.105 YUV to NV12 chroma conversion

This filter performs conversion from YUV image format to NV12 for the chroma part only. The luma part is identical between these two formats and needs to be copied separately.

This filter performs conversion from YUV image format to NV12 for the chroma part only. The luma part is identical between these two formats and needs to be copied separately.

Output data type(s):

UInt8

Preserve:

imgSize, dataType

Inputs:

- datatypes: UInt8; kernels: 1x1

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/cvtColorKernelChromaYUVToNV12/arch//shave/src/cvt-ColorKernelChromaYUVToNV12.asm

5.106 NV21 to RGB conversion

This filter performs color space conversion RGBfp16 to LumaU8 for one line in an image.

Functions

- void **SVU_SYM()** **svucvtColorNV21toRGB** (**SippFilter** *fptr)
Shave function of the NV21 to RGB conversion filter.
- **SHAVE_SYM_EXPORT** (**svucvtColorNV21toRGB**)

5.106.1 Detailed Description

This filter performs color space conversion RGBfp16 to LumaU8 for one line in an image.

Preserve:

imgSize, dataType

Output data type(s):

UInt8

Mandatory inputs:

2

Inputs:

- name: yin; datatypes: UInt8; kernels: 1x1
- name: uvin; datatypes: UInt8; kernels: 1x1

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/cvtColorNV21toRGB/arch//shave/src/cvtColorNV21toRGB-B.asm

5.106.2 Function Documentation

SHAVE_SYM_EXPORT (**svucvtColorNV21toRGB**)

void **SVU_SYM()** **svucvtColorNV21toRGB** (**SippFilter** * fptr)

Shave function of the **NV21 to RGB conversion** filter.

5.107 RGB(fp16) to Luma(u8) conversion

This filter performs color space conversion RGB(fp16) to Luma(U8) for one line in an image.

Functions

- void **SVU_SYM()** **svuCvtColorRGBfp16ToLumaU8** (**SippFilter** *fptr)
Shave function of the RGB(fp16) to Luma(u8) conversion filter.
- **SHAVE_SYM_EXPORT** (**svuCvtColorRGBfp16ToLumaU8**)

5.107.1 Detailed Description

This filter performs color space conversion RGB(fp16) to Luma(U8) for one line in an image.

Preserve:

imgSize

Output data type(s):

UInt8

Inputs:

- datatypes: half; kernels: 1x1

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/cvtColorRGBfp16ToLumaU8/arch//shave/src/cvtColorRGBfp16ToLumaU8.asm

5.107.2 Function Documentation

SHAVE_SYM_EXPORT (**svuCvtColorRGBfp16ToLumaU8**)

void **SVU_SYM()** **svuCvtColorRGBfp16ToLumaU8** (**SippFilter** * fptr)

Shave function of the **RGB(fp16) to Luma(u8) conversion** filter.

5.108 RGB(fp16) to UV420(u8) conversion

This filter performs color space conversion RGB(fp16) to UV420(U8) for one line in an image.

Functions

- void **SVU_SYM()** **svuCvtColorRGBfp16ToUV420U8** (**SippFilter** *fptr)
Shave function of the RGB(fp16) to UV420(u8) conversion filter.
- **SHAVE_SYM_EXPORT** (**svuCvtColorRGBfp16ToUV420U8**)

5.108.1 Detailed Description

This filter performs color space conversion RGB(fp16) to UV420(U8) for one line in an image.

Preserve:

imgSize

Output data type(s):

UInt8

Inputs:

- datatypes: half; kernels: 1x1

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/cvtColorRGBfp16ToUV420U8/arch//shave/src/cvtColorRGBfp16ToUV420U8.asm

5.108.2 Function Documentation

SHAVE_SYM_EXPORT (**svuCvtColorRGBfp16ToUV420U8**)

void **SVU_SYM()** **svuCvtColorRGBfp16ToUV420U8** (**SippFilter** * fptr)

Shave function of the **RGB(fp16) to UV420(u8) conversion** filter.

5.109 RGB to Chroma NV12 conversion

This filter performs conversion from RGB image format to Chroma NV12.

Data Structures

- struct `cvtColorChromaNV12Param`
Parameter structure of the RGB to Chroma NV12 conversion filter.

Functions

- void `SVU_SYM() svuCvtColorRGBtoChromaNV12 (SippFilter *fptr)`
Shave function of the RGB to Chroma NV12 conversion filter.
- `SHAVE_SYM_EXPORT (svuCvtColorRGBtoChromaNV12)`

5.109.1 Detailed Description

This filter performs conversion from RGB image format to Chroma NV12.

Output data type(s):

UInt8

Preserve:

imgSize, dataType

Inputs:

- datatypes: UInt8; kernels: 1x1

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvISP/kernels/cvtColorRGBtoChromaNV12/arch//shave/src/cvtColorRGBtoChromaNV12.asm

5.109.2 Function Documentation

`SHAVE_SYM_EXPORT (svuCvtColorRGBtoChromaNV12)`

void `SVU_SYM() svuCvtColorRGBtoChromaNV12 (SippFilter * fptr)`

Shave function of the RGB to Chroma NV12 conversion filter.

5.110 RGB to Luma conversion

This filter performs conversion from RGB image format to Luma.

This filter performs conversion from RGB image format to Luma.

5.111 ISP

Modules

- [AF Stats](#)
This filter gathers auto-focus statistics from a RGB planar image.
- [Chroma Block](#)
This filter applies chroma desaturation and 3x3 color correction matrix.
- [Contrast](#)
This filter applies contrast on pixel element.
- [Convert to YUV444](#)
This filter performs line conversion to YUV444.
- [Equalize Histogram](#)
This filter makes an equalization through an image with a given histogram.
- [Generate Chroma](#)
This filter generates Chroma planes from RGB and Luma planes.
- [Generate Chroma with subsampling](#)
This filter generates Chroma from an FP16 Luma plane(generated from RGB) and a U8 RGB plane. It also subsamples the data by half in each dimension.
- [Generate Luma U8 to Fp16](#)
This filter generates Luminance from UInt8 RGB input to fp16 output.
- [Generate Reference for Luma Denoise](#)
This filter generates an 8-bit reference plane, used by the Luma Denoise filter.
- [Generate Reference for Luma Denoise\(fp16 input\)](#)
This filter generates an 8-bit reference plane, used by the Luma Denoise filter.
- [Histogram](#)
This filter computes a histogram on a given line to be applied to all lines of an image.
- [LUT 10 to 16](#)
*This filter performs a look-up table transform of a line. Destination line is filled with values from the look-up table. Indices of the entries are taken from the source line.
false.*
- [LUT 10 to 8](#)
This filter performs a look-up table transform of a line. Destination line is filled with values from the look-up table. Indices of the entries are taken from the source line.
- [LUT 12 to 16](#)
This filter performs a look-up table transform of a line. Destination line is filled with values from the look-up table. Indices of the entries are taken from the source line.
- [LUT 12 to 8](#)
This filter performs a look-up table transform of a line. Destination line is filled with values from the look-up table. Indices of the entries are taken from the source line.
- [LUT 16 to 8](#)
This filter performs a look-up table transform of a line. Destination line is filled with values from the look-up table. Indices of the entries are taken from the source line.
- [LUT 8 to 8](#)
This filter performs a look-up table transform of a line. Destination line is filled with values from the look-up table. Indices of the entries are taken from the source line.
- [Low Level Correction](#)

This filter performs low level pixel value correction on a single plane. Contains black level correction and correct bad pixels.

- **Low Level Correction on Multiple Planes**

This filter performs low level pixel value correction on multiple planes. Contains black level correction and correct bad pixels.

- **Luma Blur**

This filter generates Luminance from RGB input.

- **Mix Median**

This filter performs mix two 8-bit planes according to a third 8-bit reference plane.

- **NV21 to RGB conversion**

This filter performs color space conversion RGBfp16 to LumaU8 for one line in an image.

- **Pixel Unpacker**

The filter unpacks 10b -> 16b/15b/14b/13b/12b/11b/10b depending on the shift value. Input of this filter need to be the output of the pixelPacker10b filter.

- **Pixel Unpacker Mipi 10b**

The filter unpacks 2 px/cc with 10bpp to 16b with Mipi.

- **Pixel Unpacker WB**

The filter unpacks 10b -> 16b/15b/14b/13b/12b/11b/10b depending on the shift value; also applies White Balance. Input of this filter need to be the output of the pixelPacker10b filter.

- **Pixel packer**

The filter takes 16bits(10bpp) as input and, in order to compress the data, split a pixel in two components.

- **RGB to Chroma NV12 conversion**

This filter performs conversion from RGB image format to Chroma NV12.

- **RGB to Luma NV12 conversion**

This filter performs conversion from RGB image format to Luma NV12.

- **RGB to Luma conversion**

This filter performs conversion from RGB image format to Luma.

- **RGB to UV conversion**

This filter performs color space conversion from RGB to UV.

- **RGB to UV420 conversion**

This filter performs color space conversion from RGB to UV420.

- **RGB to YUV422 conversion**

This filter performs color space conversion from RGB to YUV422.

- **RGB(fp16) to Luma(u8) conversion**

This filter performs color space conversion RGB(fp16) to Luma(U8) for one line in an image.

- **RGB(fp16) to UV420(u8) conversion**

This filter performs color space conversion RGB(fp16) to UV420(U8) for one line in an image.

- **Undistort**

This filter applies undistort using Brown's distortion model for known lens distortion coefficients.

- **Upscale by 2 with phases 0.25 and 0.75 fp16 to fp16**

This filter performs bilinear upscale with 2 factor; Horizontal and Vertical directions; phases 0.25 and 0.75.

- **Upscale by 2 with phases 0.25 and 0.75 fp16 to u8**

This filter performs bilinear upscale with 2 factor; Horizontal and Vertical directions; phases 0.25 and 0.75.

- **Upscale by 2 with phases 0.25 and 0.75 u16 to u16**

This filter performs bilinear upscale with 2 factor; Horizontal and Vertical directions; phases 0.25 and 0.75.

- [Upscale by 2 with phases 0.25 and 0.75 u8 to u8](#)

This filter performs bilinear upscale with 2 factor; Horizontal and Vertical directions; phases 0.25 and 0.75.

- [White Balance Bayer GBRG](#)

This filter applies white balance gains for BayerGBRG input.

- [White Balance RGB](#)

This filter applies white balance gains for RGB input.

- [YUV to NV12 chroma conversion](#)

This filter performs conversion from YUV image format to NV12 for the chroma part only. The luma part is identical between these two formats and needs to be copied separately.

- [YUV to RGB conversion](#)

This filter performs color space conversion from YUV to RGB.

- [YUV400 to YUV422 conversion](#)

This filter performs a color conversion from a simple yuv400 to yuv422 with value for chromas 0x80.

- [YUV422 to RGB conversion](#)

This filter performs color space conversion from YUV422 to RGB.

- [greyDesat](#)

This filter desaturates areas which are already very close to grey.

- [histogramStat](#)

This filter makes histogram for each RGB component input.

- [localTM](#)

This filter applies a tone mapping function to the Luma channel.

- [padBayer5Frame](#)

This filter adds a bayer padding 5 pixels top/bottom/left/right in mirror mode.

- [purpleFlare](#)

This filter applies a modified unsharp mask on the blue channel only to reduce the "purple flare" effect.

Functions

- void [SVU_SYM\(\)](#) [svuCvtColorRGBtoLuma](#) ([SippFilter](#) *fptr)

Shave function of the [RGB to Luma conversion](#) filter.

- [SHAVE_SYM_EXPORT](#) ([svuCvtColorRGBtoLuma](#))

5.111.1 Detailed Description

Preserve:

[imgSize](#)

Output data type(s):

[UInt8](#)

Inputs:

- datatypes: [UInt8](#); kernels: 1x1

Path to external ASM file(s) used in the filter implementation:

```
/components/kernelLib/MvCV/kernels/cvtColorRGBtoLuma/arch//shave/src/cvtColorRGBto-  
Luma.asm
```

5.111.2 Function Documentation

SHAVE_SYM_EXPORT (**svuCvtColorRGBtoLuma**)

void **SVU_SYM()** svuCvtColorRGBtoLuma (**SippFilter** * fptr)

Shave function of the [RGB to Luma conversion](#) filter.

5.112 RGB to Luma NV12 conversion

This filter performs conversion from RGB image format to Luma NV12.

Data Structures

- struct `cvtColorLumaNV12Param`
Parameter structure of the [RGB to Luma NV12 conversion](#) filter.

Functions

- void `SVU_SYM() svuCvtColorRGBtoLumaNV12 (SippFilter *fptr)`
Shave function of the [RGB to Luma NV12 conversion](#) filter.
- `SHAVE_SYM_EXPORT (svuCvtColorRGBtoLumaNV12)`

5.112.1 Detailed Description

This filter performs conversion from RGB image format to Luma NV12.

Output data type(s):

UInt8

Preserve:

imgSize, dataType

Inputs:

- datatypes: UInt8; kernels: 1x1

Path to external ASM file(s) used in the filter implementation:

`/components/kernelLib/MvISP/kernels/cvtColorRGBtoLumaNV12/arch//shave/src/cvtColorRGBtoLumaNV12.asm`

5.112.2 Function Documentation

`SHAVE_SYM_EXPORT (svuCvtColorRGBtoLumaNV12)`

void `SVU_SYM() svuCvtColorRGBtoLumaNV12 (SippFilter * fptr)`

Shave function of the [RGB to Luma NV12 conversion](#) filter.

5.113 RGB to UV conversion

This filter performs color space conversion from RGB to UV.

Functions

- void **SVU_SYM()** **svuCvtColorRGBtoUV** (**SippFilter** *fptr)
*Shave function of the **RGB to UV conversion** filter.*
- **SHAVE_SYM_EXPORT** (**svuCvtColorRGBtoUV**)

5.113.1 Detailed Description

This filter performs color space conversion from RGB to UV.

Preserve:

imgSize, dataType

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 1x1

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/cvtColorRGBtoUV/arch//shave/src/cvtColorRGBtoUV.asm

5.113.2 Function Documentation

SHAVE_SYM_EXPORT (**svuCvtColorRGBtoUV**)

void **SVU_SYM()** **svuCvtColorRGBtoUV** (**SippFilter** * fptr)

Shave function of the **RGB to UV conversion** filter.

5.114 RGB to UV420 conversion

This filter performs color space conversion from RGB to UV420.

Functions

- void **SVU_SYM()** **svuCvtColorRGBtoUV420** (**SippFilter** *fptr)
Shave function of the [RGB to UV420 conversion](#) filter.
- **SHAVE_SYM_EXPORT** (**svuCvtColorRGBtoUV420**)

5.114.1 Detailed Description

This filter performs color space conversion from RGB to UV420.

Preserve:

imgSize, dataType

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 1x1

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/cvtColorRGBtoUV420/arch//shave/src/cvtColorRGBtoUV420.asm

5.114.2 Function Documentation

SHAVE_SYM_EXPORT (**svuCvtColorRGBtoUV420**)

void **SVU_SYM()** **svuCvtColorRGBtoUV420** (**SippFilter** * fptr)

Shave function of the [RGB to UV420 conversion](#) filter.

5.115 RGB to YUV422 conversion

This filter performs color space conversion from RGB to YUV422.

Functions

- void **SVU_SYM()** **svuCvtColorRGBToYUV422** (**SippFilter** *fptr)
Shave function of the [RGB to YUV422 conversion](#) filter.
- **SHAVE_SYM_EXPORT** (**svuCvtColorRGBToYUV422**)

5.115.1 Detailed Description

This filter performs color space conversion from RGB to YUV422.

Preserve:

dataType

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 1x1

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/cvtColorKernelRGBToYUV422/arch//shave/src/cvtColor-
KernelRGBToYUV422.asm

5.115.2 Function Documentation

SHAVE_SYM_EXPORT (**svuCvtColorRGBToYUV422**)

void **SVU_SYM()** **svuCvtColorRGBToYUV422** (**SippFilter** * fptr)

Shave function of the [RGB to YUV422 conversion](#) filter.

5.116 YUV422 to RGB conversion

This filter performs color space conversion from YUV422 to RGB.

Functions

- void **SVU_SYM()** **svuCvtColorYUV422ToRGB** (**SippFilter** *fptr)
Shave function of the cvtColorYUV422ToRGB filter.
- **SHAVE_SYM_EXPORT** (**svuCvtColorYUV422ToRGB**)

5.116.1 Detailed Description

This filter performs color space conversion from YUV422 to RGB.

Preserve:

dataType

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 1x1

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/cvtColorKernelYUV422ToRGB/arch//shave/src/cvtColor-KernelYUV422ToRGB.asm

5.116.2 Function Documentation

SHAVE_SYM_EXPORT (**svuCvtColorYUV422ToRGB**)

void **SVU_SYM()** **svuCvtColorYUV422ToRGB** (**SippFilter** * fptr)

Shave function of the cvtColorYUV422ToRGB filter.

5.117 YUV to RGB conversion

This filter performs color space conversion from YUV to RGB.

Functions

- void **SVU_SYM()** **svuCvtColorYUVToRGB** (**SippFilter** *fptr)
Shave function of the YUV to RGB conversion filter.
- **SHAVE_SYM_EXPORT** (**svuCvtColorYUVToRGB**)

5.117.1 Detailed Description

This filter performs color space conversion from YUV to RGB.

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 1x1

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/cvtColorKernelYUVtoRGB/arch//shave/src/cvtColor-KernelYUVtoRGB.asm

5.117.2 Function Documentation

SHAVE_SYM_EXPORT (**svuCvtColorYUVToRGB**)

void **SVU_SYM()** **svuCvtColorYUVToRGB** (**SippFilter** * fptr)

Shave function of the **YUV to RGB conversion** filter.

The filter is created with the `outputWidth == 3*inputWidth` and `fptr->sliceWidth = outputWidth` so we transmit it with /3 to do the good processing

5.118 Dilate 3x3

This filter performs a dilate on the source image using the specified structuring element.

Data Structures

- struct [Dilate3x3Param](#)
Parameter structure of the [Dilate 3x3](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuDilate3x3](#) ([SippFilter](#) *fptr)
Shave function of the [Dilate 3x3](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuDilate3x3](#))

5.118.1 Detailed Description

This filter performs a dilate on the source image using the specified structuring element.

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 3x3

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/dilate3x3/arch//shave/src/dilate3x3.asm

5.118.2 Function Documentation

[SHAVE_SYM_EXPORT](#) ([svuDilate3x3](#))

void [SVU_SYM\(\)](#) [svuDilate3x3](#) ([SippFilter](#) * fptr)

Shave function of the [Dilate 3x3](#) filter.

5.119 Dilate 5x5

This filter performs a dilate on the source image using the specified structuring element.

Data Structures

- struct [Dilate5x5Param](#)
Parameter structure of the [Dilate 5x5](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuDilate5x5](#) ([SippFilter](#) *fptr)
Shave function of the [Dilate 5x5](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuDilate5x5](#))

5.119.1 Detailed Description

This filter performs a dilate on the source image using the specified structuring element.

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 5x5

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/dilate5x5/arch//shave/src/dilate5x5.asm

5.119.2 Function Documentation

[SHAVE_SYM_EXPORT](#) ([svuDilate5x5](#))

void [SVU_SYM\(\)](#) [svuDilate5x5](#) ([SippFilter](#) * fptr)

Shave function of the [Dilate 5x5](#) filter.

5.120 Dilate 7x7

This filter performs a dilate on the source image using the specified structuring element.

Data Structures

- struct [Dilate7x7Param](#)
Parameter structure of the [Dilate 7x7](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuDilate7x7](#) ([SippFilter](#) *fptr)
Shave function of the [Dilate 7x7](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuDilate7x7](#))

5.120.1 Detailed Description

This filter performs a dilate on the source image using the specified structuring element.

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 7x7

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/dilate7x7/arch//shave/src/dilate7x7.asm

5.120.2 Function Documentation

[SHAVE_SYM_EXPORT](#) ([svuDilate7x7](#))

void [SVU_SYM\(\)](#) [svuDilate7x7](#) ([SippFilter](#) * fptr)

Shave function of the [Dilate 7x7](#) filter.

5.121 Generic Dilate

This filter performs a generic dilate on the input image using the kernel size given by the user.

Data Structures

- struct **DilateGenericParam**
*Parameter structure of the **Generic Dilate** filter.*

Functions

- void **SVU_SYM()** **svuDilateGeneric** (**SippFilter** *fptr)
*Shave function of the **Generic Dilate** filter.*
- **SHAVE_SYM_EXPORT** (**svuDilateGeneric**)

5.121.1 Detailed Description

This filter performs a generic dilate on the input image using the kernel size given by the user.

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 3x3, 5x5, 7x7, 9x9, 11x11, 13x13, 15x15

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/dilate/arch//shave/src/dilate.asm

5.121.2 Function Documentation

SHAVE_SYM_EXPORT (**svuDilateGeneric**)

void **SVU_SYM()** **svuDilateGeneric** (**SippFilter** * fptr)

Shave function of the **Generic Dilate** filter.

5.122 Equalize Histogram

This filter makes an equalization through an image with a given histogram.

Data Structures

- struct `EqualizeHistParam`
Parameter structure of the `Equalize Histogram` filter.

Functions

- void `SVU_SYM() svuEqualizeHist (SippFilter *fptr)`
Shave function of the `Equalize Histogram` filter.
- `SHAVE_SYM_EXPORT (svuEqualizeHist)`

5.122.1 Detailed Description

This filter makes an equalization through an image with a given histogram.

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 1x1

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/equalizeHist/arch//shave/src/equalizeHist.asm

5.122.2 Function Documentation

`SHAVE_SYM_EXPORT (svuEqualizeHist)`

void `SVU_SYM() svuEqualizeHist (SippFilter * fptr)`

Shave function of the `Equalize Histogram` filter.

5.123 Erode 3x3

This filter applies the erode filter on the source image using the specified structuring element.

Data Structures

- struct [Erode3x3Param](#)
Parameter structure of the [Erode 3x3](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuErode3x3](#) ([SippFilter](#) *fptr)
Shave function of the [Erode 3x3](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuErode3x3](#))

5.123.1 Detailed Description

This filter applies the erode filter on the source image using the specified structuring element.

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 3x3

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/erode3x3/arch//shave/src/erode3x3.asm

5.123.2 Function Documentation

[SHAVE_SYM_EXPORT](#) ([svuErode3x3](#))

void [SVU_SYM\(\)](#) [svuErode3x3](#) ([SippFilter](#) * fptr)

Shave function of the [Erode 3x3](#) filter.

5.124 Erode 5x5

This filter applies the erode filter on the source image using the specified structuring element.

Data Structures

- struct [Erode5x5Param](#)
Parameter structure of the [Erode 5x5](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuErode5x5](#) ([SippFilter](#) *fptr)
Shave function of the [Erode 5x5](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuErode5x5](#))

5.124.1 Detailed Description

This filter applies the erode filter on the source image using the specified structuring element.

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 5x5

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/erode5x5/arch//shave/src/erode5x5.asm

5.124.2 Function Documentation

[SHAVE_SYM_EXPORT](#) ([svuErode5x5](#))

void [SVU_SYM\(\)](#) [svuErode5x5](#) ([SippFilter](#) * fptr)

Shave function of the [Erode 5x5](#) filter.

5.125 Erode 7x7

This filter applies the erode filter on the source image using the specified structuring element.

Data Structures

- struct [Erode7x7Param](#)
Parameter structure of the [Erode 7x7](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuErode7x7](#) ([SippFilter](#) *fptr)
Shave function of the [Erode 7x7](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuErode7x7](#))

5.125.1 Detailed Description

This filter applies the erode filter on the source image using the specified structuring element.

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 7x7

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/erode7x7/arch//shave/src/erode7x7.asm

5.125.2 Function Documentation

[SHAVE_SYM_EXPORT](#) ([svuErode7x7](#))

void [SVU_SYM\(\)](#) [svuErode7x7](#) ([SippFilter](#) * fptr)

Shave function of the [Erode 7x7](#) filter.

5.126 AF Stats

This filter gathers auto-focus statistics from a RGB planar image.

Data Structures

- struct [SippHwIOBuf](#)
- struct [PpAf](#)

Parameter structure of the [AF Stats](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuExtAfStats](#) ([SippFilter](#) *fptr)
Shave function of the [AF Stats](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuExtAfStats](#))

5.126.1 Detailed Description

This filter gathers auto-focus statistics from a RGB planar image.

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 1x1

5.126.2 Function Documentation

[SHAVE_SYM_EXPORT](#) ([svuExtAfStats](#))

void [SVU_SYM\(\)](#) [svuExtAfStats](#) ([SippFilter](#) * fptr)

Shave function of the [AF Stats](#) filter.

5.127 Fast9M2

The filter performs the Fast9 for Myriad 2 corner detection.

Data Structures

- struct **Fast9M2Param**
*Parameter structure of the **Fast9M2** filter.*

Functions

- void **SVU_SYM()** **svuFast9M2** (**SippFilter** *fptr)
*Shave function of the **Fast9M2** filter.*
- **SHAVE_SYM_EXPORT** (**svuFast9M2**)

5.127.1 Detailed Description

The filter performs the Fast9 for Myriad 2 corner detection.

Output data type(s):

float

Preserve:

numPlanes, dataType

Inputs:

- datatypes: float; kernels: 7x7

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/fast9M2/arch//shave/src/fast9M2.asm

5.127.2 Function Documentation

SHAVE_SYM_EXPORT (**svuFast9M2**)

void **SVU_SYM()** **svuFast9M2** (**SippFilter** * fptr)

Shave function of the **Fast9M2** filter.

5.128 Fast9ScoreCv

The filter performs the Fast9 for Myriad 2 corner detection.

Data Structures

- struct **Fast9ScoreCvParam**
*Parameter structure of the **Fast9ScoreCv** filter.*

Functions

- void **SVU_SYM()** **svuFast9ScoreCv** (**SippFilter** *fptr)
*Shave function of the **Fast9ScoreCv** filter.*
- **SHAVE_SYM_EXPORT** (**svuFast9ScoreCv**)

5.128.1 Detailed Description

The filter performs the Fast9 for Myriad 2 corner detection.

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 1x1

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/fast9ScoreCv/arch//shave/src/fast9ScoreCv.asm

5.128.2 Function Documentation

SHAVE_SYM_EXPORT (**svuFast9ScoreCv**)

void **SVU_SYM()** **svuFast9ScoreCv** (**SippFilter** * fptr)

Shave function of the **Fast9ScoreCv** filter.

5.129 Gauss Blur

This filter applies gaussian blur on Luma channel.

Functions

- void **SVU_SYM()** **svuGauss** (**SippFilter** *fptr)
*Shave function of the **Gauss Blur** filter.*
- **SHAVE_SYM_EXPORT** (**svuGauss**)

5.129.1 Detailed Description

This filter applies gaussian blur on Luma channel.

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 5x5

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/gauss/arch//shave/src/gauss.asm

5.129.2 Function Documentation

SHAVE_SYM_EXPORT (**svuGauss**)

void **SVU_SYM()** **svuGauss** (**SippFilter** * fptr)

Shave function of the **Gauss Blur** filter.

5.130 GaussHx2

This filter applies downscale 2x horizontal gaussian blur. Has to be used in combination with GaussVx2 to obtain correct output.

Functions

- void **SVU_SYM()** **svuGaussHx2** (**SippFilter** *fptr)
Shave function of the GaussHx2 filter.
- **SHAVE_SYM_EXPORT** (**svuGaussHx2**)

5.130.1 Detailed Description

This filter applies downscale 2x horizontal gaussian blur. Has to be used in combination with GaussVx2 to obtain correct output.

Output data type(s):

UInt8

Preserve:

numPlanes, dataType

Inputs:

- datatypes: UInt8; kernels: 1x5

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/gaussHx2/arch//shave/src/gaussHx2.asm

5.130.2 Function Documentation

SHAVE_SYM_EXPORT (**svuGaussHx2**)

void **SVU_SYM()** **svuGaussHx2** (**SippFilter** * fptr)

Shave function of the **GaussHx2** filter.

5.131 GaussHx2_fp16

This filter applies downscale 2x horizontal gaussian blur. Has to be used in combination with GaussVx2 to obtain correct output.

Functions

- void **SVU_SYM()** **svuGaussHx2_fp16** (**SippFilter** *fptr)
Shave function of the GaussHx2_fp16 filter.
- **SHAVE_SYM_EXPORT** (**svuGaussHx2_fp16**)

5.131.1 Detailed Description

This filter applies downscale 2x horizontal gaussian blur. Has to be used in combination with GaussVx2 to obtain correct output.

Output data type(s):

half

Inputs:

Preserve:

numPlanes, dataType

- datatypes: half; kernels: 1x5

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/gaussHx2_fp16/arch//shave/src/gaussHx2_fp16.asm

5.131.2 Function Documentation

SHAVE_SYM_EXPORT (**svuGaussHx2_fp16**)

void **SVU_SYM()** **svuGaussHx2_fp16** (**SippFilter** * fptr)

Shave function of the **GaussHx2_fp16** filter.

5.132 GaussVx2

This filter applies downscale 2x vertical gaussian blur. Has to be used in combination with GaussHx2 to obtain correct output.

Functions

- void **SVU_SYM()** **svuGaussVx2** (**SippFilter** *fptr)
Shave function of the GaussVx2 filter.
- **SHAVE_SYM_EXPORT** (**svuGaussVx2**)

5.132.1 Detailed Description

This filter applies downscale 2x vertical gaussian blur. Has to be used in combination with GaussHx2 to obtain correct output.

Flags:

SIPP_RESIZE

Output data type(s):

UInt8

Preserve:

numPlanes, dataType

Inputs:

- datatypes: UInt8; kernels: 5x1

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/gaussVx2/arch//shave/src/gaussVx2.asm

5.132.2 Function Documentation

SHAVE_SYM_EXPORT (**svuGaussVx2**)

void **SVU_SYM()** **svuGaussVx2** (**SippFilter** * fptr)

Shave function of the **GaussVx2** filter.

5.133 GaussVx2_fp16

This filter applies downscale 2x vertical gaussian blur. Has to be used in combination with GaussHx2 to obtain correct output.

Functions

- void **SVU_SYM()** **svuGaussVx2_fp16** (**SippFilter** *fptr)
Shave function of the GaussVx2_fp16 filter.
- **SHAVE_SYM_EXPORT** (**svuGaussVx2_fp16**)

5.133.1 Detailed Description

This filter applies downscale 2x vertical gaussian blur. Has to be used in combination with GaussHx2 to obtain correct output.

Flags:

SIPP_RESIZE

Output data type(s):

half

Preserve:

numPlanes, dataType

Inputs:

- datatypes: half; kernels: 5x1

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/gaussVx2_fp16/arch//shave/src/gaussVx2_fp16.asm

5.133.2 Function Documentation

SHAVE_SYM_EXPORT (**svuGaussVx2_fp16**)

void **SVU_SYM()** **svuGaussVx2_fp16** (**SippFilter** * fptr)

Shave function of the **GaussVx2_fp16** filter.

5.134 Generate Chroma

This filter generates Chroma planes from RGB and Luma planes.

Data Structures

- struct [ChrGenParam](#)

Parameter structure of the filter.

Functions

- void [SVU_SYM\(\)](#) [svuGenChroma](#) ([SippFilter](#) *fptr)
Shave function of the [Generate Chroma](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuGenChroma](#))

5.134.1 Detailed Description

This filter generates Chroma planes from RGB and Luma planes.

Output data type(s):

UInt8

Preserve:

imgSize, dataType

Mandatory inputs:

2

Inputs:

- name: RGB; datatypes: UInt8; kernels: 1x1
- name: Luma; datatypes: UInt8; kernels: 1x1

5.134.2 Function Documentation

[SHAVE_SYM_EXPORT](#) ([svuGenChroma](#))

void [SVU_SYM\(\)](#) [svuGenChroma](#) ([SippFilter](#) * fptr)

Shave function of the [Generate Chroma](#) filter.

5.135 Generate Chroma with subsampling

This filter generates Chroma from an FP16 Luma plane(generated from RGB) and a U8 RGB plane. It also subsamples the data by half in each dimension.

Data Structures

- struct [ChrGenSSParam](#)

Parameter structure of the [Generate Chroma with subsampling](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuGenChromaSS](#) ([SippFilter](#) *fptr)
Shave function of the [Generate Chroma with subsampling](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuGenChromaSS](#))

5.135.1 Detailed Description

This filter generates Chroma from an FP16 Luma plane(generated from RGB) and a U8 RGB plane. It also subsamples the data by half in each dimension.

Output data type(s):

UInt8

Preserve:

Flags:

SIPP_RESIZE

Inputs:

- name: RGB; datatypes: UInt8; kernels: 2x2

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvISP/kernels/genChromaSS/arch//shave/src/genChromaSS.asm

5.135.2 Function Documentation

[SHAVE_SYM_EXPORT](#) ([svuGenChromaSS](#))

void [SVU_SYM\(\)](#) [svuGenChromaSS](#) ([SippFilter](#) * fptr)

Shave function of the [Generate Chroma with subsampling](#) filter.

5.136 Generate Reference for Luma Denoise

This filter generates an 8-bit reference plane, used by the Luma Denoise filter.

Data Structures

- struct **YDnsRefParam**

Parameter structure of the [Generate Reference for Luma Denoise](#) filter.

Functions

- void **SVU_SYM()** **svuGenDnsRef** (**SippFilter** *fptr)
Shave function of the [Generate Reference for Luma Denoise](#) filter.
- **SHAVE_SYM_EXPORT** (**svuGenDnsRef**)

5.136.1 Detailed Description

This filter generates an 8-bit reference plane, used by the Luma Denoise filter.

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 1x1

5.136.2 Function Documentation

SHAVE_SYM_EXPORT (**svuGenDnsRef**)

void **SVU_SYM()** **svuGenDnsRef** (**SippFilter** * fptr)

Shave function of the [Generate Reference for Luma Denoise](#) filter.

5.137 Generate Reference for Luma Denoise(fp16 input)

This filter generates an 8-bit reference plane, used by the Luma Denoise filter.

Data Structures

- struct **YDnsRefFp16Param**

Parameter structure of the [Generate Reference for Luma Denoise\(fp16 input\)](#) filter.

Functions

- void **SVU_SYM()** **svuGenDnsRefFp16** (**SippFilter** *fptr)
Shave function of the [Generate Reference for Luma Denoise\(fp16 input\)](#) filter.
- **SHAVE_SYM_EXPORT** (**svuGenDnsRefFp16**)

5.137.1 Detailed Description

This filter generates an 8-bit reference plane, used by the Luma Denoise filter.

Output data type(s):

UInt8

Preserve:

imgSize, numPlanes

Inputs:

- datatypes: half; kernels: 1x1

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvISP/kernels/genDnsRefFp16/arch//shave/src/genDnsRefFp16.asm

5.137.2 Function Documentation

SHAVE_SYM_EXPORT (**svuGenDnsRefFp16**)

void **SVU_SYM()** **svuGenDnsRefFp16** (**SippFilter** * fptr)

Shave function of the [Generate Reference for Luma Denoise\(fp16 input\)](#) filter.

5.138 Luma Blur

This filter generates Luminance from RGB input.

Functions

- void **SVU_SYM()** **svuGenLuma** (**SippFilter** *fptr)
Shave function of the [Luma Blur](#) filter.
- **SHAVE_SYM_EXPORT** (**svuGenLuma**)

5.138.1 Detailed Description

This filter generates Luminance from RGB input.

Output data type(s):

UInt8

Preserve:

imgSize, dataType

Inputs:

- datatypes: UInt8; kernels: 1x0, 1x1

5.138.2 Function Documentation

SHAVE_SYM_EXPORT (**svuGenLuma**)

void **SVU_SYM()** **svuGenLuma** (**SippFilter** * fptr)

Shave function of the [Luma Blur](#) filter.

5.139 Generate Luma U8 to Fp16

This filter generates Luminance from UInt8 RGB input to fp16 output.

Data Structures

- struct `GenLumaU8Fp16Param`
Shave function of the `Generate Luma U8 to Fp16` filter.

Functions

- void `SVU_SYM()` `svuGenLumaU8Fp16` (`SippFilter *fptr`)
- `SHAVE_SYM_EXPORT` (`svuGenLumaU8Fp16`)

5.139.1 Detailed Description

This filter generates Luminance from UInt8 RGB input to fp16 output.

Output data type(s):

half

Preserve:

imgSize

Inputs:

- datatypes: UInt8; kernels: 1x1

Path to external ASM file(s) used in the filter implementation:

`/components/kernelLib/MvISP/kernels/genLumaU8Fp16/arch//shave/src/genLumaU8Fp16.asm`

5.139.2 Function Documentation

`SHAVE_SYM_EXPORT` (`svuGenLumaU8Fp16`)

void `SVU_SYM()` `svuGenLumaU8Fp16` (`SippFilter * fptr`)

5.140 greyDesat

This filter desaturates areas which are already very close to grey.

Data Structures

- struct **GreyDesatParam**
*Parameter structure of the **greyDesat** filter.*

Functions

- void **SVU_SYM()** **svuGreyDesat** (**SippFilter** *fptr)
*Shave function of the **greyDesat** filter.*
- **SHAVE_SYM_EXPORT** (**svuGreyDesat**)

5.140.1 Detailed Description

This filter desaturates areas which are already very close to grey.

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 3x3

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvISP/kernels/greyDesat/arch//shave/src/greyDesat.asm

5.140.2 Function Documentation

SHAVE_SYM_EXPORT (**svuGreyDesat**)

void **SVU_SYM()** **svuGreyDesat** (**SippFilter** * fptr)

Shave function of the **greyDesat** filter.

5.141 hammingDistance

hammingDistance kernel finds matches between two descriptors

Data Structures

- struct [HammingDistanceParam](#)
Parameter structure of the [hammingDistance](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuHammingDistance](#) ([SippFilter](#) *fptr)
Shave function of the [hammingDistance](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuHammingDistance](#))

5.141.1 Detailed Description

hammingDistance kernel finds matches between two descriptors

Output data type(s):

UInt16

Preserve:

numPlanes

Inputs:

- name: input1; datatypes: UInt8; kernels: 1x1
- name: input2; datatypes: UInt8; kernels: 1x1

Mandatory inputs:

2

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/hammingDistance/arch//shave/src/hammingDistance.asm

5.141.2 Function Documentation

[SHAVE_SYM_EXPORT](#) ([svuHammingDistance](#))

void [SVU_SYM\(\)](#) [svuHammingDistance](#) ([SippFilter](#) * fptr)

Shave function of the [hammingDistance](#) filter.

5.142 harrisResponse

This filter computes Harris response over a patch of the image with a radius of 3.

Data Structures

- struct [HarrisSwParam](#)
Parameter structure of the [harrisResponse](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuHarrisResponse](#) ([SippFilter](#) *fptr)
Shave function of the [harrisResponse](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuHarrisResponse](#))

5.142.1 Detailed Description

This filter computes Harris response over a patch of the image with a radius of 3.

Preserve:

numPlanes, imgSize

Output data type(s):

float

Inputs:

- datatypes: UInt8; kernels: 8x8

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/harrisResponse/arch//shave/src/harrisResponse.asm

5.142.2 Function Documentation

[SHAVE_SYM_EXPORT](#) ([svuHarrisResponse](#))

void [SVU_SYM\(\)](#) [svuHarrisResponse](#) ([SippFilter](#) * fptr)

Shave function of the [harrisResponse](#) filter.

5.143 Histogram

This filter computes a histogram on a given line to be applied to all lines of an image.

Data Structures

- struct [HistogramParam](#)
Parameter structure of the [Histogram](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuHistogram](#) ([SippFilter](#) *fptr)
Shave function of the [Histogram](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuHistogram](#))

5.143.1 Detailed Description

This filter computes a histogram on a given line to be applied to all lines of an image.

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 1x1

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/histogram/arch//shave/src/histogram.asm

5.143.2 Function Documentation

[SHAVE_SYM_EXPORT](#) ([svuHistogram](#))

void [SVU_SYM\(\)](#) [svuHistogram](#) ([SippFilter](#) * fptr)

Shave function of the [Histogram](#) filter.

5.144 histogramStat

This filter makes histogram for each RGB component input.

Data Structures

- struct **HistogramStatParam**
*Parameter structure of the **histogramStat** filter.*

Functions

- void **SVU_SYM()** **svuHistogramStat** (**SippFilter** *fptr)
*Shave function of the **histogramStat** filter.*
- **SHAVE_SYM_EXPORT** (**svuHistogramStat**)

5.144.1 Detailed Description

This filter makes histogram for each RGB component input.

Preserve:

imgSize

Output data type(s):

UInt32

Inputs:

- datatypes: UInt8; kernels: 1x1

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvISP/kernels/histogramStat/arch//shave/src/histogramStat.asm

5.144.2 Function Documentation

SHAVE_SYM_EXPORT (**svuHistogramStat**)

void **SVU_SYM()** **svuHistogramStat** (**SippFilter** * fptr)

Shave function of the **histogramStat** filter.

5.145 Homography

This filter performs a homography transformation.

Data Structures

- struct **HomographyParam**
Parameter structure of the [Homography](#) filter.

Functions

- void **SVU_SYM()** **svuHomography** (**SippFilter** *fptr)
Shave function of the [Homography](#) filter.
- **SHAVE_SYM_EXPORT** (**svuHomography**)

5.145.1 Detailed Description

This filter performs a homography transformation.

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 64x4

5.145.2 Function Documentation

SHAVE_SYM_EXPORT (**svuHomography**)

void **SVU_SYM()** **svuHomography** (**SippFilter** * fptr)

Shave function of the [Homography](#) filter.

5.146 Integral Image Square Sum(f32)

This filter makes the sum of all pixels before it and on the left of it's column(this particular case makes square sum of pixels in f32 format).

Functions

- void **SVU_SYM()** **svuIntegralImageSqSumF32M2** (**SippFilter** *fptr)
*Shave function of the **Integral Image Square Sum(f32)** filter.*
- **SHAVE_SYM_EXPORT** (**svuIntegralImageSqSumF32M2**)

5.146.1 Detailed Description

This filter makes the sum of all pixels before it and on the left of it's column(this particular case makes square sum of pixels in f32 format).

Preserve:

imgSize, numPlanes

Output data type(s):

float

Inputs:

- datatypes: UInt8; kernels: 1x1

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/integralImageSquareSumFloatM2/arch//shave/src/integral-ImageSquareSumFloatM2.asm

5.146.2 Function Documentation

SHAVE_SYM_EXPORT (**svuIntegralImageSqSumF32M2**)

void **SVU_SYM()** **svuIntegralImageSqSumF32M2** (**SippFilter** * fptr)

Shave function of the **Integral Image Square Sum(f32)** filter.

5.147 Integral Image Square Sum(U32)

This filter makes the sum of all pixels before it and on the left of it's column(this particular case makes square sum of pixels in U32 format).

Functions

- void **SVU_SYM()** **svuIntegralImageSqSumU32M2** (**SippFilter** *fptr)
*Shave function of the **Integral Image Square Sum(U32)** filter.*
- **SHAVE_SYM_EXPORT** (**svuIntegralImageSqSumU32M2**)

5.147.1 Detailed Description

This filter makes the sum of all pixels before it and on the left of it's column(this particular case makes square sum of pixels in U32 format).

Output data type(s):

UInt32

Preserve:

numPlanes

Inputs:

- datatypes: UInt8; kernels: 1x1

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/integralImageSquareSumM2/arch//shave/src/integralImage-SquareSumM2.asm

5.147.2 Function Documentation

SHAVE_SYM_EXPORT (**svuIntegralImageSqSumU32M2**)

void **SVU_SYM()** **svuIntegralImageSqSumU32M2** (**SippFilter** * fptr)

Shave function of the **Integral Image Square Sum(U32)** filter.

5.148 Integral Image Sum(f32)

This filter makes the sum of all pixels before it and on the left of it's column(this particular case makes square sum of pixels in f32 format).

Functions

- void **SVU_SYM()** **svuIntegralImageSumF32M2** (**SippFilter** *fptr)
Shave function of the [Integral Image Sum\(f32\)](#) filter.
- **SHAVE_SYM_EXPORT** (**svuIntegralImageSumF32M2**)

5.148.1 Detailed Description

This filter makes the sum of all pixels before it and on the left of it's column(this particular case makes square sum of pixels in f32 format).

Preserve:

imgSize, numPlanes

Output data type(s):

float

Inputs:

- datatypes: UInt8; kernels: 1x1

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/integralImageSumFloatM2/arch//shave/src/integralImageSumFloatM2.asm

5.148.2 Function Documentation

SHAVE_SYM_EXPORT (**svuIntegralImageSumF32M2**)

void **SVU_SYM()** **svuIntegralImageSumF32M2** (**SippFilter** * fptr)

Shave function of the [Integral Image Sum\(f32\)](#) filter.

5.149 Integral Image Sum(U16toU32)

This filter makes the sum of all pixels before it and on the left of it's column(this particular case makes square sum of pixels in U32 format).

Functions

- void **SVU_SYM()** **svuIntegralImageSumU16U32** (**SippFilter** *fptr)
*Shave function of the **Integral Image Sum(U16toU32)** filter.*
- **SHAVE_SYM_EXPORT** (**svuIntegralImageSumU16U32**)

5.149.1 Detailed Description

This filter makes the sum of all pixels before it and on the left of it's column(this particular case makes square sum of pixels in U32 format).

Preserve:

imgSize, numPlanes

Output data type(s):

UInt32

Inputs:

- datatypes: UInt16; kernels: 1x1

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/integralImageSumU16U32/arch//shave/src/integralImageSumU16U32.asm

5.149.2 Function Documentation

SHAVE_SYM_EXPORT (**svuIntegralImageSumU16U32**)

void **SVU_SYM()** **svuIntegralImageSumU16U32** (**SippFilter** * fptr)

Shave function of the **Integral Image Sum(U16toU32)** filter.

5.150 Integral Image Sum(U32)

This filter makes the sum of all pixels before it and on the left of it's column(this particular case makes square sum of pixels in U32 format).

Functions

- void **SVU_SYM()** **svuIntegralImageSumU32M2** (**SippFilter** *fptr)
*Shave function of the **Integral Image Sum(U32)** filter.*
- **SHAVE_SYM_EXPORT** (**svuIntegralImageSumU32M2**)

5.150.1 Detailed Description

This filter makes the sum of all pixels before it and on the left of it's column(this particular case makes square sum of pixels in U32 format).

Preserve:

imgSize, numPlanes

Output data type(s):

UInt32

Inputs:

- datatypes: UInt8; kernels: 1x1

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/integralImageSumM2/arch//shave/src/integralImageSum-M2.asm

5.150.2 Function Documentation

SHAVE_SYM_EXPORT (**svuIntegralImageSumU32M2**)

void **SVU_SYM()** **svuIntegralImageSumU32M2** (**SippFilter** * fptr)

Shave function of the **Integral Image Sum(U32)** filter.

5.151 interpolatePixelBilinear

This filter makes the bilinear interpolation of four pixels.

Data Structures

- struct [InterpolatePixelBilinearParam](#)
Parameter structure of the [interpolatePixelBilinear](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuInterpolatePixelBilinear](#) ([SippFilter](#) *fptr)
Shave function of the [interpolatePixelBilinear](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuInterpolatePixelBilinear](#))

5.151.1 Detailed Description

This filter makes the bilinear interpolation of four pixels.

Output data type(s):

half

Inputs:

- datatypes: half; kernels: 1x1

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/interpolatePixelBilinear/arch//shave/src/interpolatePixelBilinear.asm

5.151.2 Function Documentation

[SHAVE_SYM_EXPORT](#) ([svuInterpolatePixelBilinear](#))

void [SVU_SYM\(\)](#) [svuInterpolatePixelBilinear](#) ([SippFilter](#) * fptr)

Shave function of the [interpolatePixelBilinear](#) filter.

5.152 Laplacian 3x3

The filter applies a Laplacian filter with custom size.

Functions

- void **SVU_SYM()** **svuLaplacian3x3** (**SippFilter** *fptr)
*Shave function of the **Laplacian 3x3** filter.*
- **SHAVE_SYM_EXPORT** (**svuLaplacian3x3**)

5.152.1 Detailed Description

The filter applies a Laplacian filter with custom size.

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 3x3

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/sLaplacian3x3/arch//shave/src/sLaplacian3x3.asm

5.152.2 Function Documentation

SHAVE_SYM_EXPORT (**svuLaplacian3x3**)

void **SVU_SYM()** **svuLaplacian3x3** (**SippFilter** * fptr)

Shave function of the **Laplacian 3x3** filter.

5.153 Laplacian 5x5

The filter applies a Laplacian filter with custom size.

Functions

- void **SVU_SYM()** **svuLaplacian5x5** (**SippFilter** *fptr)
*Shave function of the **Laplacian 5x5** filter.*
- **SHAVE_SYM_EXPORT** (**svuLaplacian5x5**)

5.153.1 Detailed Description

The filter applies a Laplacian filter with custom size.

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 5x5

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/sLaplacian5x5/arch//shave/src/sLaplacian5x5.asm

5.153.2 Function Documentation

SHAVE_SYM_EXPORT (**svuLaplacian5x5**)

void **SVU_SYM()** **svuLaplacian5x5** (**SippFilter** * fptr)

Shave function of the **Laplacian 5x5** filter.

5.154 Laplacian 5x5 Fp16 To Fp16

The filter applies a Laplacian filter with custom size.

Functions

- void **SVU_SYM()** **svuLaplacian5x5Fp16ToFp16** (**SippFilter** *fptr)
*Shave function of the **Laplacian 5x5 Fp16 To Fp16** filter.*
- **SHAVE_SYM_EXPORT** (**svuLaplacian5x5Fp16ToFp16**)

5.154.1 Detailed Description

The filter applies a Laplacian filter with custom size.

Output data type(s):

half

Inputs:

- datatypes: half; kernels: 5x5

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/sLaplacian5x5Fp16ToFp16/arch//shave/src/sLaplacian5x5-Fp16ToFp16.asm

5.154.2 Function Documentation

SHAVE_SYM_EXPORT (**svuLaplacian5x5Fp16ToFp16**)

void **SVU_SYM()** **svuLaplacian5x5Fp16ToFp16** (**SippFilter** * fptr)

Shave function of the **Laplacian 5x5 Fp16 To Fp16** filter.

5.155 Laplacian 7x7

The filter applies a Laplacian filter with custom size.

Functions

- void **SVU_SYM()** **svuLaplacian7x7** (**SippFilter** *fptr)
*Shave function of the **Laplacian 7x7** filter.*
- **SHAVE_SYM_EXPORT** (**svuLaplacian7x7**)

5.155.1 Detailed Description

The filter applies a Laplacian filter with custom size.

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 7x7

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/sLaplacian7x7/arch//shave/src/sLaplacian7x7.asm

5.155.2 Function Documentation

SHAVE_SYM_EXPORT (**svuLaplacian7x7**)

void **SVU_SYM()** **svuLaplacian7x7** (**SippFilter** * fptr)

Shave function of the **Laplacian 7x7** filter.

5.156 Laplacian 7x7 Fp16 To Fp16

The filter applies a Laplacian filter with custom size.

Functions

- void **SVU_SYM()** **svuLaplacian7x7Fp16ToFp16** (**SippFilter** *fptr)
*Shave function of the **Laplacian 7x7 Fp16 To Fp16** filter.*
- **SHAVE_SYM_EXPORT** (**svuLaplacian7x7Fp16ToFp16**)

5.156.1 Detailed Description

The filter applies a Laplacian filter with custom size.

Output data type(s):

half

Inputs:

- datatypes: half; kernels: 7x7

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/sLaplacian7x7Fp16ToFp16/arch//shave/src/sLaplacian7x7-Fp16ToFp16.asm

5.156.2 Function Documentation

SHAVE_SYM_EXPORT (**svuLaplacian7x7Fp16ToFp16**)

void **SVU_SYM()** **svuLaplacian7x7Fp16ToFp16** (**SippFilter** * fptr)

Shave function of the **Laplacian 7x7 Fp16 To Fp16** filter.

5.157 localMaxMin3x3_fp16

/// This kernel will find the points which are minimums or maximums in their 3x3 zone.

Functions

- void **SVU_SYM()** **svuLocalMaxMin3x3_fp16** (**SippFilter** *fptr)
*Shave function of the **localMaxMin3x3_fp16** filter.*
- **SHAVE_SYM_EXPORT** (**svuLocalMaxMin3x3_fp16**)

5.157.1 Detailed Description

/// This kernel will find the points which are minimums or maximums in their 3x3 zone.

Preserve:

imgSize

Output data type(s):

UInt32

Inputs:

- datatypes: half; kernels: 3x3

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/localMaxMin3x3_fp16/arch//shave/src/localMaxMin3x3_fp16.asm

5.157.2 Function Documentation

SHAVE_SYM_EXPORT (**svuLocalMaxMin3x3_fp16**)

void **SVU_SYM()** **svuLocalMaxMin3x3_fp16** (**SippFilter** * fptr)

Shave function of the **localMaxMin3x3_fp16** filter.

5.158 localTM

This filter applies a tone mapping function to the Luma channel.

Data Structures

- struct **LocalTMParam**
*Parameter structure of the **localTM** filter.*

Functions

- void **SVU_SYM()** **svuLocalTM** (**SippFilter** *fptr)
*Shave function of the **localTM** filter.*
- **SHAVE_SYM_EXPORT** (**svuLocalTM**)

5.158.1 Detailed Description

This filter applies a tone mapping function to the Luma channel.

Output data type(s):

UInt16

Flags:

SIPP_RESIZE

Inputs:

- name: input1; datatypes: UInt16; kernels: 1x1
- name: input2; datatypes: UInt8; kernels: 2x2

Mandatory inputs:

2

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvISP/kernels/localTM/arch//shave/src/localTM.asm

5.158.2 Function Documentation

SHAVE_SYM_EXPORT (**svuLocalTM**)

void **SVU_SYM()** **svuLocalTM** (**SippFilter** * fptr)

Shave function of the **localTM** filter.

5.159 Low Level Correction

This filter performs low level pixel value correction on a single plane. Contains black level correction and correct bad pixels.

Data Structures

- struct [LowLvlCorrParam](#)
Parameter structure of the [Low Level Correction](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuLowLvlCorr](#) ([SippFilter](#) *fptr)
Shave function of the [Low Level Correction on Multiple Planes](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuLowLvlCorr](#))

Variables

- [UInt8 LowLvlCorrParam::blackLevel](#)
black level
- float [LowLvlCorrParam::alphaBadPixel](#)
level of correction

5.159.1 Detailed Description

This filter performs low level pixel value correction on a single plane. Contains black level correction and correct bad pixels.

Output data type(s):

[UInt16](#)

Inputs:

- datatypes: [UInt16](#); kernels: 3x3

5.159.2 Function Documentation

[SHAVE_SYM_EXPORT](#) ([svuLowLvlCorr](#))

void [SVU_SYM\(\)](#) [svuLowLvlCorr](#) ([SippFilter](#) * fptr)

Shave function of the [Low Level Correction on Multiple Planes](#) filter.

5.159.3 Variable Documentation

`float LowLvlCorrParam::alphaBadPixel`

level of correction

Referenced by `svuLowLvlCorr()`.

`UInt8 LowLvlCorrParam::blackLevel`

black level

Referenced by `svuLowLvlCorr()`.

5.160 Low Level Correction on Multiple Planes

This filter performs low level pixel value correction on multiple planes. Contains black level correction and correct bad pixels.

Data Structures

- struct [LowLvlCorrNPIParam](#)

Parameter structure of the [Low Level Correction on Multiple Planes](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svulowLvlCorrMultiplePlanes](#) ([SippFilter](#) *fptr)
Shave function of the [Low Level Correction on Multiple Planes](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svulowLvlCorrMultiplePlanes](#))

5.160.1 Detailed Description

This filter performs low level pixel value correction on multiple planes. Contains black level correction and correct bad pixels.

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 3x3

5.160.2 Function Documentation

[SHAVE_SYM_EXPORT](#) ([svulowLvlCorrMultiplePlanes](#))

void [SVU_SYM\(\)](#) [svulowLvlCorrMultiplePlanes](#) ([SippFilter](#) * fptr)

Shave function of the [Low Level Correction on Multiple Planes](#) filter.

5.161 Luma Blur

This filter applies the blur operator on Luma channel.

Functions

- void **SVU_SYM()** **svuLumaBlur** (**SippFilter** *fptr)
*Shave function of the **Luma Blur** filter.*
- **SHAVE_SYM_EXPORT** (**svuLumaBlur**)

5.161.1 Detailed Description

This filter applies the blur operator on Luma channel.

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 3x3

5.161.2 Function Documentation

SHAVE_SYM_EXPORT (**svuLumaBlur**)

void **SVU_SYM()** **svuLumaBlur** (**SippFilter** * fptr)

Shave function of the **Luma Blur** filter.

5.162 LUT 10 to 16

This filter performs a look-up table transform of a line. Destination line is filled with values from the look-up table. Indices of the entries are taken from the source line.

false.

Data Structures

- struct [Lut10to16Param](#)
Parameter structure of the LUT 10 to 16 filter.

Functions

- void [SVU_SYM\(\)](#) [svuLut10to16](#) ([SippFilter](#) *fptr)
Shave function of the LUT 10 to 16 filter.
- [SHAVE_SYM_EXPORT](#) ([svuLut10to16](#))

5.162.1 Detailed Description

This filter performs a look-up table transform of a line. Destination line is filled with values from the look-up table. Indices of the entries are taken from the source line.

false.

Output data type(s):

UInt16 false

Inputs:

- datatypes: UInt16; kernels: 1x1

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/lookupTable10to16/arch//shave/src/lookupTable10to16.asm

5.162.2 Function Documentation

[SHAVE_SYM_EXPORT](#) ([svuLut10to16](#))

void [SVU_SYM\(\)](#) [svuLut10to16](#) ([SippFilter](#) * fptr)

Shave function of the [LUT 10 to 16](#) filter.

5.163 LUT 10 to 8

This filter performs a look-up table transform of a line. Destination line is filled with values from the look-up table. Indices of the entries are taken from the source line.

Data Structures

- struct [Lut10to8Param](#)

Parameter structure of the [LUT 10 to 8](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuLut10to8](#) ([SippFilter](#) *fptr)
Shave function of the [LUT 10 to 8](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuLut10to8](#))

5.163.1 Detailed Description

This filter performs a look-up table transform of a line. Destination line is filled with values from the look-up table. Indices of the entries are taken from the source line.

Preserve:

numPlanes, imgSize, dataType

Output data type(s):

UInt8

Inputs:

- datatypes: UInt16; kernels: 1x1

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/lookupTable10to8/arch//shave/src/lookupTable10to8.asm

5.163.2 Function Documentation

[SHAVE_SYM_EXPORT](#) ([svuLut10to8](#))

void [SVU_SYM\(\)](#) [svuLut10to8](#) ([SippFilter](#) * fptr)

Shave function of the [LUT 10 to 8](#) filter.

5.164 LUT 12 to 16

This filter performs a look-up table transform of a line. Destination line is filled with values from the look-up table. Indices of the entries are taken from the source line.

Data Structures

- struct **Lut12to16Param**

Parameter structure of the LUT 12 to 16 filter.

Functions

- void **SVU_SYM()** **svuLut12to16** (**SippFilter** *fptr)
Shave function of the LUT 12 to 16 filter.
- **SHAVE_SYM_EXPORT** (**svuLut12to16**)

5.164.1 Detailed Description

This filter performs a look-up table transform of a line. Destination line is filled with values from the look-up table. Indices of the entries are taken from the source line.

Output data type(s):

UInt16

Inputs:

- datatypes: UInt16; kernels: 1x1

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/lookupTable12to16/arch//shave/src/lookupTable12to16.asm

5.164.2 Function Documentation

SHAVE_SYM_EXPORT (**svuLut12to16**)

void **SVU_SYM()** **svuLut12to16** (**SippFilter** * fptr)

Shave function of the LUT 12 to 16 filter.

5.165 LUT 12 to 8

This filter performs a look-up table transform of a line. Destination line is filled with values from the look-up table. Indices of the entries are taken from the source line.

Data Structures

- struct [Lut12to8Param](#)

Parameter structure of the [LUT 12 to 8](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuLut12to8](#) ([SippFilter](#) *fptr)
Shave function of the [LUT 12 to 8](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuLut12to8](#))

5.165.1 Detailed Description

This filter performs a look-up table transform of a line. Destination line is filled with values from the look-up table. Indices of the entries are taken from the source line.

Preserve:

numPlanes, imgSize, dataType

Output data type(s):

UInt8

Inputs:

- datatypes: UInt16; kernels: 1x1

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/lookupTable12to8/arch//shave/src/lookupTable12to8.asm

5.165.2 Function Documentation

[SHAVE_SYM_EXPORT](#) ([svuLut12to8](#))

void [SVU_SYM\(\)](#) [svuLut12to8](#) ([SippFilter](#) * fptr)

Shave function of the [LUT 12 to 8](#) filter.

5.166 LUT 8 to 8

This filter performs a look-up table transform of a line. Destination line is filled with values from the look-up table. Indices of the entries are taken from the source line.

Data Structures

- struct [Lut8to8Param](#)

Parameter structure of the [LUT 8 to 8](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuLut8to8](#) ([SippFilter](#) *fptr)
Shave function of the [LUT 8 to 8](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuLut8to8](#))

5.166.1 Detailed Description

This filter performs a look-up table transform of a line. Destination line is filled with values from the look-up table. Indices of the entries are taken from the source line.

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 1x1

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/lookupTable8to8/arch//shave/src/lookupTable8to8.asm

5.166.2 Function Documentation

[SHAVE_SYM_EXPORT](#) ([svuLut8to8](#))

void [SVU_SYM\(\)](#) [svuLut8to8](#) ([SippFilter](#) * fptr)

Shave function of the [LUT 8 to 8](#) filter.

5.167 LUT 16 to 8

This filter performs a look-up table transform of a line. Destination line is filled with values from the look-up table. Indices of the entries are taken from the source line.

Data Structures

- struct [YDnsRefLut10bppParam](#)
Parameter structure of the LUT 16 to 8 filter.

Functions

- void [SVU_SYM\(\)](#) [svuLutP10BppU16inU8out](#) ([SippFilter](#) *fptr)
Shave function of the LUT 16 to 8 filter.
- [SHAVE_SYM_EXPORT](#) ([svuLutP10BppU16inU8out](#))

5.167.1 Detailed Description

This filter performs a look-up table transform of a line. Destination line is filled with values from the look-up table. Indices of the entries are taken from the source line.

Preserve:

numPlanes, imgSize

Output data type(s):

UInt8

Inputs:

- datatypes: UInt16; kernels: 1x1

5.167.2 Function Documentation

[SHAVE_SYM_EXPORT](#) ([svuLutP10BppU16inU8out](#))

void [SVU_SYM\(\)](#) [svuLutP10BppU16inU8out](#) ([SippFilter](#) * fptr)

Shave function of the [LUT 16 to 8](#) filter.

5.168 maxTest3x3_fp16

This kernel will compare the points from inBufferCandidates to the corresponding 3x3 zone of inBuffer.

Data Structures

- struct **MaxTest3x3fp16Param**
*Parameter structure of the **maxTest3x3_fp16** filter.*

Functions

- void **SVU_SYM()** **svuMaxTest3x3_fp16** (**SippFilter** *fptr)
*Shave function of the **maxTest3x3_fp16** filter.*
- **SHAVE_SYM_EXPORT** (**svuMaxTest3x3_fp16**)

5.168.1 Detailed Description

This kernel will compare the points from inBufferCandidates to the corresponding 3x3 zone of inBuffer.

Preserve:

imgSize

Output data type(s):

UInt32

Inputs:

- datatypes: half; kernels: 3x3

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/maxTest3x3_fp16/arch//shave/src/maxTest3x3_fp16.asm

5.168.2 Function Documentation

SHAVE_SYM_EXPORT (**svuMaxTest3x3_fp16**)

void **SVU_SYM()** **svuMaxTest3x3_fp16** (**SippFilter** * fptr)

Shave function of the **maxTest3x3_fp16** filter.

5.169 meanStdDev

This kernel calculates mean and standard deviation of an array of elements.

Functions

- void **SVU_SYM()** **svuMeanStdDev** (**SippFilter** *fptr)
*Parameter structure of the **meanStdDev** filter.*
- **SHAVE_SYM_EXPORT** (**svuMeanStdDev**)

5.169.1 Detailed Description

This kernel calculates mean and standard deviation of an array of elements.

Preserve:

numPlanes

Output data type(s):

float

Inputs:

- datatypes: UInt8; kernels: 1x1

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/meanStdDev/arch//shave/src/meanStdDev.asm

5.169.2 Function Documentation

SHAVE_SYM_EXPORT (**svuMeanStdDev**)

void **SVU_SYM()** **svuMeanStdDev** (**SippFilter** * fptr)

Parameter structure of the **meanStdDev** filter.

Shave function of the **meanStdDev** filter.

5.170 Min/Max Value Position

This filter computes the minimum and the maximum value of a given input line and their position.

Data Structures

- struct **MinMaxPosParam**
Parameter structure of the [Min/Max Value Position](#) filter.

Functions

- void **SVU_SYM()** **svuMinMaxPos** (**SippFilter** *fptr)
Shave function of the [Min/Max Value Position](#) filter.
- **SHAVE_SYM_EXPORT** (svuMinMaxPos)

5.170.1 Detailed Description

This filter computes the minimum and the maximum value of a given input line and their position.

Preserve:

numPlanes

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 1x1

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/minMaxPos/arch//shave/src/minMaxPos.asm

5.170.2 Function Documentation

SHAVE_SYM_EXPORT (**svuMinMaxPos**)

void **SVU_SYM()** **svuMinMaxPos** (**SippFilter** * fptr)

Shave function of the [Min/Max Value Position](#) filter.

5.171 Min/Max Value

This filter computes the minimum and the maximum value of a given input image.

Data Structures

- struct `minMaxValParam`
Parameter structure of the [Min/Max Value](#) filter.

Functions

- void `SVU_SYM() svuMinMaxValue (SippFilter *fptr)`
Shave function of the [Min/Max Value](#) filter.
- `SHAVE_SYM_EXPORT (svuMinMaxValue)`

5.171.1 Detailed Description

This filter computes the minimum and the maximum value of a given input image.

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 1x1

Path to external ASM file(s) used in the filter implementation:

`/components/kernelLib/MvCV/kernels/minMaxKernel/arch//shave/src/minMaxKernel.asm`

5.171.2 Function Documentation

`SHAVE_SYM_EXPORT (svuMinMaxValue)`

void `SVU_SYM() svuMinMaxValue (SippFilter * fptr)`

Shave function of the [Min/Max Value](#) filter.

5.172 minTest3x3_fp16

This kernel will compare the points from inBufferCandidates to the corresponding 3x3 zone of inBuffer.

Data Structures

- struct **MinTest3x3fp16Param**
Parameter structure of the minTest3x3_fp16 filter.

Functions

- void **SVU_SYM()** **svuMinTest3x3_fp16** (**SippFilter** *fptr)
Shave function of the minTest3x3_fp16 filter.
- **SHAVE_SYM_EXPORT** (svuMinTest3x3_fp16)

5.172.1 Detailed Description

This kernel will compare the points from inBufferCandidates to the corresponding 3x3 zone of inBuffer.

Preserve:

imgSize

Output data type(s):

UInt32

Inputs:

- datatypes: half; kernels: 3x3

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/minTest3x3_fp16/arch//shave/src/minTest3x3_fp16.asm

5.172.2 Function Documentation

SHAVE_SYM_EXPORT (**svuMinTest3x3_fp16**)

void **SVU_SYM()** **svuMinTest3x3_fp16** (**SippFilter** * fptr)

Shave function of the **minTest3x3_fp16** filter.

5.173 Mix Median

This filter performs mix two 8-bit planes according to a third 8-bit reference plane.

Data Structures

- struct **MixMedianParam**
*Parameter structure of the **Mix Median** filter.*

Functions

- void **SVU_SYM()** **svuMixMedian** (**SippFilter** *fptr)
*Shave function of the **Mix Median** filter.*
- **SHAVE_SYM_EXPORT** (**svuMixMedian**)

5.173.1 Detailed Description

This filter performs mix two 8-bit planes according to a third 8-bit reference plane.

Output data type(s):

UInt8

Inputs:

- name: in1; datatypes: UInt8; kernels: 1x1
- name: in2; datatypes: UInt8; kernels: 1x1
- name: ref; datatypes: UInt8; kernels: 1x1

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvISP/kernels/mixMedian/arch//shave/src/mixMedian.asm

5.173.2 Function Documentation

SHAVE_SYM_EXPORT (**svuMixMedian**)

void **SVU_SYM()** **svuMixMedian** (**SippFilter** * fptr)

Shave function of the **Mix Median** filter.

5.174 MonoImbalance

Output data type(s):

half

Preserve:

numPlanes

Inputs:

- datatypes: UInt16; kernels: 3x3

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/monoImbalance/arch//shave/src/monoImbalance.asm @

5.175 Negative

This filter creates the negative of the image.

Functions

- void **SVU_SYM()** **svuNegative** (**SippFilter** *fptr)
Shave function of the [Negative](#) filter.
- **SHAVE_SYM_EXPORT** (**svuNegative**)

5.175.1 Detailed Description

This filter creates the negative of the image.

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 1x1

5.175.2 Function Documentation

SHAVE_SYM_EXPORT (**svuNegative**)

void **SVU_SYM()** **svuNegative** (**SippFilter** * fptr)

Shave function of the [Negative](#) filter.

5.176 nonMaxFp32

This filter verifies whether each element from the central line is the maximum within a 3x3 range.

Data Structures

- struct `nonMax3x3Fp32Param`

Functions

- void `SVU_SYM()` `svuNonMax3x3Fp32` (`SippFilter` *fptr)
Shave function of the RGB to Luma conversion filter.
- `SHAVE_SYM_EXPORT` (`svuNonMax3x3Fp32`)

5.176.1 Detailed Description

This filter verifies whether each element from the central line is the maximum within a 3x3 range.

Preserve:

`imgSize`

Output data type(s):

`UInt16`

Inputs:

- datatypes: float; kernels: 3x3

Path to external ASM file(s) used in the filter implementation:

`/components/kernelLib/MvCV/kernels/nonMax3x3_fp32/arch//shave/src/nonMax3x3_fp32.asm`

5.176.2 Function Documentation

`SHAVE_SYM_EXPORT` (`svuNonMax3x3Fp32`)

void `SVU_SYM()` `svuNonMax3x3Fp32` (`SippFilter` * fptr)

Shave function of the [RGB to Luma conversion](#) filter.

5.177 nonMax3x3U8

This filter verifies whether each element from the central line is the maximum within a 3x3 range.

Functions

- void **SVU_SYM()** **svuNonMax3x3U8** (**SippFilter** *fptr)
*Shave function of the **nonMax3x3U8** filter.*
- **SHAVE_SYM_EXPORT** (**svuNonMax3x3U8**)

5.177.1 Detailed Description

This filter verifies whether each element from the central line is the maximum within a 3x3 range.

Preserve:

imgSize

Output data type(s):

UInt16

Inputs:

- datatypes: UInt8; kernels: 3x3

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/nonMax3x3_u8/arch//shave/src/nonMax3x3_u8.asm

5.177.2 Function Documentation

SHAVE_SYM_EXPORT (**svuNonMax3x3U8**)

void **SVU_SYM()** **svuNonMax3x3U8** (**SippFilter** * fptr)

Shave function of the **nonMax3x3U8** filter.

5.178 padBayer5Frame

This filter adds a bayer padding 5 pixels top/bottom/left/right in mirror mode.

Functions

- void `clampInLines` (`SippFilter *fptr`)
- void `SVU_SYM()` `svuPadBayer5Frame` (`SippFilter *fptr`)
Shave function of the [Threshold](#) filter.
- `SHAVE_SYM_EXPORT` (`svuPadBayer5Frame`)

5.178.1 Detailed Description

This filter adds a bayer padding 5 pixels top/bottom/left/right in mirror mode.

Preserve:

numPlanes, dataType

Output data type(s):

UInt16

Inputs:

- datatypes: UInt16; kernels: 13x10

5.178.2 Function Documentation

void `clampInLines` (`SippFilter * fptr`)

`SHAVE_SYM_EXPORT` (`svuPadBayer5Frame`)

void `SVU_SYM()` `svuPadBayer5Frame` (`SippFilter * fptr`)

Shave function of the [Threshold](#) filter.

5.179 Pixel packer

The filter takes 16bits(10bpp) as input and, in order to compress the data, split a pixel in two components.

Functions

- void **SVU_SYM()** **svuPixelPacker10b** (**SippFilter** *fptr)
Shave function of the [Pixel packer](#) filter.
- **SHAVE_SYM_EXPORT** (**svuPixelPacker10b**)

5.179.1 Detailed Description

The filter takes 16bits(10bpp) as input and, in order to compress the data, split a pixel in two components.

Output data type(s):

UInt8, UInt32

Preserve:

numPLanes

Inputs:

- datatypes: UInt16; kernels: 1x1

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvISP/kernels/pixelPacker10b/arch//shave/src/pixelPacker10b.asm

5.179.2 Function Documentation

SHAVE_SYM_EXPORT (**svuPixelPacker10b**)

void **SVU_SYM()** **svuPixelPacker10b** (**SippFilter** * fptr)

Shave function of the [Pixel packer](#) filter.

5.180 Pixel Unpacker

The filter unpacks 10b -> 16b/15b/14b/13b/12b/11b/10b depending on the shift value. Input of this filter need to be the output of the pixelPacker10b filter.

Data Structures

- struct **PixelUnpackerParam**
*Parameter structure of the **Pixel Unpacker** filter.*

Functions

- void **SVU_SYM()** **svuPixelUnpacker** (**SippFilter** *fptr)
*Shave function of the **Pixel Unpacker** filter.*
- **SHAVE_SYM_EXPORT** (**svuPixelUnpacker**)

5.180.1 Detailed Description

The filter unpacks 10b -> 16b/15b/14b/13b/12b/11b/10b depending on the shift value. Input of this filter need to be the output of the pixelPacker10b filter.

Output data type(s):

UInt16

Preserve:

numPLanes

Inputs:

- name: ms8b; datatypes: UInt32; kernels: 1x1
- name: ls2b; datatypes: UInt8; kernels: 1x1

Mandatory inputs:

2

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvISP/kernels/pixelUnpacker/arch//shave/src/pixelUnpacker.asm

5.180.2 Function Documentation

SHAVE_SYM_EXPORT (**svuPixelUnpacker**)

void **SVU_SYM()** **svuPixelUnpacker** (**SippFilter** * fptr)

Shave function of the **Pixel Unpacker** filter.

5.181 Pixel Unpacker Mipi 10b

The filter unpacks 2 px/cc with 10bpp to 16b with Mipi.

Data Structures

- struct `PixelUnpackerMipi10bParam`
Parameter structure of the `Pixel Unpacker Mipi 10b` filter.

Functions

- void `SVU_SYM()` `svuPixelUnpackerMipi10b` (`SippFilter *fptr`)
Shave function of the `Pixel Unpacker Mipi 10b` filter.
- `SHAVE_SYM_EXPORT` (`svuPixelUnpackerMipi10b`)

5.181.1 Detailed Description

The filter unpacks 2 px/cc with 10bpp to 16b with Mipi.

Output data type(s):

UInt16

Preserve:

numPLanes

Inputs:

- datatypes: UInt32; kernels: 1x1

Path to external ASM file(s) used in the filter implementation:

`/components/kernelLib/MvISP/kernels/pixelUnpackerMipi10b/arch//shave/src/pixelUnpacker-Mipi10b.asm`

5.181.2 Function Documentation

`SHAVE_SYM_EXPORT` (`svuPixelUnpackerMipi10b`)

void `SVU_SYM()` `svuPixelUnpackerMipi10b` (`SippFilter * fptr`)

Shave function of the `Pixel Unpacker Mipi 10b` filter.

5.182 Pixel Unpacker WB

The filter unpacks 10b -> 16b/15b/14b/13b/12b/11b/10b depending on the shift value; also applies White Balance. Input of this filter need to be the output of the pixelPacker10b filter.

Data Structures

- struct **PixelUnpackerWBParam**
*Parameter structure of the **Pixel Unpacker WB** filter.*

Functions

- void **SVU_SYM()** **svuPixelUnpackerWB** (**SippFilter** *fptr)
*Shave function of the **Pixel Unpacker WB** filter.*
- **SHAVE_SYM_EXPORT** (**svuPixelUnpackerWB**)

5.182.1 Detailed Description

The filter unpacks 10b -> 16b/15b/14b/13b/12b/11b/10b depending on the shift value; also applies White Balance. Input of this filter need to be the output of the pixelPacker10b filter.

Output data type(s):

UInt16

Preserve:

numPLanes

Inputs:

- name: ms8b; datatypes: UInt32; kernels: 1x1
- name: ls2b; datatypes: UInt8; kernels: 1x1

Mandatory inputs:

2

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvISP/kernels/pixelUnpackerWB/arch//shave/src/pixelUnpackerWB.asm

5.182.2 Function Documentation

SHAVE_SYM_EXPORT (**svuPixelUnpackerWB**)

void **SVU_SYM()** **svuPixelUnpackerWB** (**SippFilter** * fptr)

Shave function of the **Pixel Unpacker WB** filter.

5.183 Pixel Position

This filter returns the position of a given pixel value.

Data Structures

- struct `positionKernelParam`
Parameter structure of the [Pixel Position](#) filter.

Functions

- void `SVU_SYM() svuPositionKernel (SippFilter *fptr)`
Shave function of the [Pixel Position](#) filter.
- `SHAVE_SYM_EXPORT (svuPositionKernel)`

5.183.1 Detailed Description

This filter returns the position of a given pixel value.

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 1x1

Path to external ASM file(s) used in the filter implementation:

`/components/kernelLib/MvCV/kernels/pixelPos/arch//shave/src/pixelPos.asm`

5.183.2 Function Documentation

`SHAVE_SYM_EXPORT (svuPositionKernel)`

void `SVU_SYM() svuPositionKernel (SippFilter * fptr)`

Shave function of the [Pixel Position](#) filter.

5.184 purpleFlare

This filter applies a modified unsharp mask on the blue channel only to reduce the "purple flare" effect.

Data Structures

- struct **PurpleFlareParam**
*Parameter structure of the **purpleFlare** filter.*

Functions

- void **SVU_SYM()** **svuPurpleFlare** (**SippFilter** *fptr)
*Shave function of the **purpleFlare** filter.*
- **SHAVE_SYM_EXPORT** (**svuPurpleFlare**)

5.184.1 Detailed Description

This filter applies a modified unsharp mask on the blue channel only to reduce the "purple flare" effect.

Preserve:

numPlanes, imgSize

Output data type(s):

UInt16

Inputs:

- datatypes: UInt16; kernels: 3x3

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvISP/kernels/purpleFlare/arch//shave/src/purpleFlare.asm

5.184.2 Function Documentation

SHAVE_SYM_EXPORT (**svuPurpleFlare**)

void **SVU_SYM()** **svuPurpleFlare** (**SippFilter** * fptr)

Shave function of the **purpleFlare** filter.

5.185 Pyramid Downscale

This filter applies a pyramid operator using 5x5 gauss downscale operator.

Functions

- void **SVU_SYM()** **svuPyrDown** (**SippFilter** *fptr)
Shave function of the [Pyramid Downscale](#) filter.
- **SHAVE_SYM_EXPORT** (**svuPyrDown**)

5.185.1 Detailed Description

This filter applies a pyramid operator using 5x5 gauss downscale operator.

Flags:

SIPP_RESIZE

Preserve:

numPlanes

Output data type(s):

UInt8

Inputs:

- datatypes: **UInt8**; kernels: 5x5

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/pyrdown/arch//shave/src/pyrdown.asm

5.185.2 Function Documentation

SHAVE_SYM_EXPORT (**svuPyrDown**)

void **SVU_SYM()** **svuPyrDown** (**SippFilter** * fptr)

Shave function of the [Pyramid Downscale](#) filter.

5.186 Random Noise

This filter generates random noise.

Data Structures

- struct **RandNoiseParam**
Parameter structure of the [Random Noise](#) filter.

Functions

- void **SVU_SYM()** **svuGenNoise** (**SippFilter** *fptr)
Shave function of the [Random Noise](#) filter.
- **SHAVE_SYM_EXPORT** (**svuGenNoise**)

5.186.1 Detailed Description

This filter generates random noise.

Output data type(s):

half

Inputs:

- datatypes: half; kernels: 1x1

5.186.2 Function Documentation

SHAVE_SYM_EXPORT (**svuGenNoise**)

void **SVU_SYM()** **svuGenNoise** (**SippFilter** * fptr)

Shave function of the [Random Noise](#) filter.

5.187 Random Noise (high speed)

This filter generates random noise using high speed algorithm.

Data Structures

- struct **RandNoiseFp16Param**
*Parameter structure of the **Random Noise (high speed)** filter.*

Functions

- void **SVU_SYM()** **svuGenNoiseFp16** (**SippFilter** *fptr)
*Shave function of the **Random Noise (high speed)** filter.*
- **SHAVE_SYM_EXPORT** (**svuGenNoiseFp16**)

5.187.1 Detailed Description

This filter generates random noise using high speed algorithm.

Output data type(s):

half

Inputs:

- datatypes: half; kernels: 1x1

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvISP/kernels/randNoiseFp16/arch//shave/src/randNoiseFp16.asm

5.187.2 Function Documentation

SHAVE_SYM_EXPORT (**svuGenNoiseFp16**)

void **SVU_SYM()** **svuGenNoiseFp16** (**SippFilter** * fptr)

Shave function of the **Random Noise (high speed)** filter.

5.188 Sum of Absolute Differences 11x11

The filter computes the sum of absolute differences between two input images on a 11x11 kernel.

Functions

- void **SVU_SYM**() **svuSAD11x11** (**SippFilter** *fptr)
Shave function of the [Sum of Absolute Differences 11x11](#) filter.
- **SHAVE_SYM_EXPORT** (**svuSAD11x11**)

5.188.1 Detailed Description

The filter computes the sum of absolute differences between two input images on a 11x11 kernel.

Output data type(s):

UInt8

Inputs:

- name: input1; datatypes: UInt8; kernels: 11x11
- name: input2; datatypes: UInt8; kernels: 11x11

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/sumOfAbsDiff11x11/arch//shave/src/sumOfAbsDiff11x11.asm

5.188.2 Function Documentation

SHAVE_SYM_EXPORT (**svuSAD11x11**)

void **SVU_SYM**() **svuSAD11x11** (**SippFilter** * fptr)

Shave function of the [Sum of Absolute Differences 11x11](#) filter.

5.189 Sum of Absolute Differences 5x5

The filter computes the sum of absolute differences between two input images on a 5x5 kernel.

Functions

- void **SVU_SYM**() **svuSAD5x5** (**SippFilter** *fptr)
*Shave function of the **Sum of Absolute Differences 5x5** filter.*
- **SHAVE_SYM_EXPORT** (**svuSAD5x5**)

5.189.1 Detailed Description

The filter computes the sum of absolute differences between two input images on a 5x5 kernel.

Output data type(s):

UInt8

Inputs:

- name: input1; datatypes: UInt8; kernels: 5x5
- name: input2; datatypes: UInt8; kernels: 5x5

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/sumOfAbsDiff5x5/arch//shave/src/sumOfAbsDiff5x5.asm

5.189.2 Function Documentation

SHAVE_SYM_EXPORT (**svuSAD5x5**)

void **SVU_SYM**() **svuSAD5x5** (**SippFilter** * fptr)

Shave function of the **Sum of Absolute Differences 5x5** filter.

5.190 Downscale by 2

This filter performs bilinear downscale with 0.5 factor – Horizontal and Vertical directions.

Functions

- void **SVU_SYM()** **svuScl05BilinHV** (**SippFilter** *fptr)
Shave function of the [Downscale by 2](#) filter.
- **SHAVE_SYM_EXPORT** (**svuScl05BilinHV**)

5.190.1 Detailed Description

This filter performs bilinear downscale with 0.5 factor – Horizontal and Vertical directions.

Preserve:

numPlanes, dataType

Output data type(s):

UInt8, UInt16

Flags:

SIPP_RESIZE

Inputs:

- datatypes: UInt8, UInt16; kernels: 2x2

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvISP/kernels/scale05BilinHV_U8ToU8/arch//shave/src/scale05BilinHV-
_U8ToU8.asm, /components/kernelLib/MvISP/kernels/scale05BilinHV_U16ToU16/arch//shave/src/scale05-
BilinHV_U16ToU16.asm

5.190.2 Function Documentation

SHAVE_SYM_EXPORT (**svuScl05BilinHV**)

void **SVU_SYM()** **svuScl05BilinHV** (**SippFilter** * fptr)

Shave function of the [Downscale by 2](#) filter.

5.191 Downscale by 2 (fp16/u8)

This filter performs bilinear downscale with 0.5 factor – Horizontal and Vertical directions. fp16 in/u8 out variant.

Functions

- void **SVU_SYM()** **svuScale05BilinHV_Fp16U8** (**SippFilter** *fptr)
Shave function of the [Downscale by 2 \(fp16/u8\)](#) filter.
- **SHAVE_SYM_EXPORT** (**svuScale05BilinHV_Fp16U8**)

5.191.1 Detailed Description

This filter performs bilinear downscale with 0.5 factor – Horizontal and Vertical directions. fp16 in/u8 out variant.

Preserve:

numPlanes, dataType

Flags:

SIPP_RESIZE

Output data type(s):

UInt8

Inputs:

- datatypes: half; kernels: 2x2

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvISP/kernels/scale05BilinHV_Fp16U8/arch//shave/src/scale05BilinHV-_Fp16U8.asm

5.191.2 Function Documentation

SHAVE_SYM_EXPORT (**svuScale05BilinHV_Fp16U8**)

void **SVU_SYM()** **svuScale05BilinHV_Fp16U8** (**SippFilter** * fptr)

Shave function of the [Downscale by 2 \(fp16/u8\)](#) filter.

5.192 Downscale by 2 (fp16/fp16)

This filter performs bilinear downscale with 0.5 factor – Horizontal and Vertical directions. fp16 in/out variant.

Functions

- void **SVU_SYM()** **svuScale05BilinHVFp16** (**SippFilter** *fptr)
Shave function of the [Downscale by 2 \(fp16/fp16\)](#) filter.
- **SHAVE_SYM_EXPORT** (**svuScale05BilinHVFp16**)

5.192.1 Detailed Description

This filter performs bilinear downscale with 0.5 factor – Horizontal and Vertical directions. fp16 in/out variant.

Preserve:

numPlanes, dataType

Flags:

SIPP_RESIZE

Output data type(s):

half

Inputs:

- datatypes: half; kernels: 2x2

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvISP/kernels/scale05BilinHVFp16/arch//shave/src/scale05BilinHVFp16.asm

5.192.2 Function Documentation

SHAVE_SYM_EXPORT (**svuScale05BilinHVFp16**)

void **SVU_SYM()** **svuScale05BilinHVFp16** (**SippFilter** * fptr)

Shave function of the [Downscale by 2 \(fp16/fp16\)](#) filter.

5.193 Lanczos Downscale by 2 (6 taps)

This filter applies a lanczos downscale, with factor 0.5, and 6 taps; Horizontal and vertical directions.

Functions

- void **SVU_SYM()** **svuScl05Lanc6** (**SippFilter** *fptr)
Shave function of the [Lanczos Downscale by 2 \(6 taps\)](#) filter.
- **SHAVE_SYM_EXPORT** (**svuScl05Lanc6**)

5.193.1 Detailed Description

This filter applies a lanczos downscale, with factor 0.5, and 6 taps; Horizontal and vertical directions.

Preserve:

numPlanes, dataType

Flags:

SIPP_RESIZE

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 6x6

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvISP/kernels/scale05Lanc6HV/arch//shave/src/scale05Lanc6HV.asm

5.193.2 Function Documentation

SHAVE_SYM_EXPORT (**svuScl05Lanc6**)

void **SVU_SYM()** **svuScl05Lanc6** (**SippFilter** * fptr)

Shave function of the [Lanczos Downscale by 2 \(6 taps\)](#) filter.

5.194 Lanczos Downscale by 2 (7 taps)

This filter applies a lanczos downscale, with factor 0.5, and 7 taps; Horizontal and vertical directions.

Functions

- void **SVU_SYM()** **svuScl05Lanc7** (**SippFilter** *fptr)
Shave function of the [Lanczos Downscale by 2 \(7 taps\)](#) filter.
- **SHAVE_SYM_EXPORT** (**svuScl05Lanc7**)

5.194.1 Detailed Description

This filter applies a lanczos downscale, with factor 0.5, and 7 taps; Horizontal and vertical directions.

Preserve:

numPlanes, dataType

Flags:

SIPP_RESIZE

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 7x7

5.194.2 Function Documentation

SHAVE_SYM_EXPORT (**svuScl05Lanc7**)

void **SVU_SYM()** **svuScl05Lanc7** (**SippFilter** * fptr)

Shave function of the [Lanczos Downscale by 2 \(7 taps\)](#) filter.

5.195 Upscale by 2

This filter performs bilinear upscale with 2 factor; Horizontal and Vertical directions.

Functions

- void **SVU_SYM()** **svuScl2xBilinHV** (**SippFilter** *fptr)
Shave function of the Upscale by 2 filter.
- **SHAVE_SYM_EXPORT** (**svuScl2xBilinHV**)

5.195.1 Detailed Description

This filter performs bilinear upscale with 2 factor; Horizontal and Vertical directions.

Preserve:

numPlanes, dataType

Flags:

SIPP_RESIZE

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 2x2

5.195.2 Function Documentation

SHAVE_SYM_EXPORT (**svuScl2xBilinHV**)

void **SVU_SYM()** **svuScl2xBilinHV** (**SippFilter** * fptr)

Shave function of the Upscale by 2 filter.

5.196 Upscale by 2 with phases 0.25 and 0.75 fp16 to fp16

This filter performs bilinear upscale with 2 factor; Horizontal and Vertical directions; phases 0.25 and 0.75.

Functions

- void **SVU_SYM()** `svuScale2xBilinHV_025_075_Fp16ToFp16` (**SippFilter** *fptr)
Shave function of the Upscale by 2 with phases 0.25 and 0.75 fp16 to fp16 filter.
- **SHAVE_SYM_EXPORT** (`svuScale2xBilinHV_025_075_Fp16ToFp16`)

5.196.1 Detailed Description

This filter performs bilinear upscale with 2 factor; Horizontal and Vertical directions; phases 0.25 and 0.75.

Flags:

SIPP_RESIZE

Output data type(s):

half

Preserve:

numPlanes, dataType

Inputs:

- datatypes: half; kernels: 2x2

Mandatory inputs:

1

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvISP/kernels/scale2xBilinHV025_Fp16ToFp16/arch//shave/src/scale2xBilinHV025_Fp16ToFp16.asm, /components/kernelLib/MvISP/kernels/scale2xBilinHV075_Fp16ToFp16/arch//shave/src/scale2xBilinHV075_Fp16ToFp16.asm

5.196.2 Function Documentation

SHAVE_SYM_EXPORT (`svuScale2xBilinHV_025_075_Fp16ToFp16`)

void **SVU_SYM()** `svuScale2xBilinHV_025_075_Fp16ToFp16` (**SippFilter** * fptr)

Shave function of the Upscale by 2 with phases 0.25 and 0.75 fp16 to fp16 filter.

5.197 Upscale by 2 with phases 0.25 and 0.75 u16 to u16

This filter performs bilinear upscale with 2 factor; Horizontal and Vertical directions; phases 0.25 and 0.75.

Functions

- void **SVU_SYM()** `svuScale2xBilinHV_025_075_U16ToU16 (SippFilter *fptr)`
Shave function of the Upscale by 2 with phases 0.25 and 0.75 u16 to u16 filter.
- **SHAVE_SYM_EXPORT** (`svuScale2xBilinHV_025_075_U16ToU16`)

5.197.1 Detailed Description

This filter performs bilinear upscale with 2 factor; Horizontal and Vertical directions; phases 0.25 and 0.75.

Flags:

SIPP_RESIZE

Output data type(s):

UInt16

Preserve:

numPlanes, dataType

Inputs:

- datatypes: UInt16; kernels: 2x2

Mandatory inputs:

1

Path to external ASM file(s) used in the filter implementation:

`/components/kernelLib/MvISP/kernels/scale2xBilinHV025_U16ToU16/arch//shave/src/scale2x-BilinHV025_U16ToU16.asm, /components/kernelLib/MvISP/kernels/scale2xBilinHV075_U16ToU16/arch//shave/src/scale2xBilinHV075_U16ToU16.asm`

5.197.2 Function Documentation

SHAVE_SYM_EXPORT (`svuScale2xBilinHV_025_075_U16ToU16`)

void **SVU_SYM()** `svuScale2xBilinHV_025_075_U16ToU16 (SippFilter * fptr)`

Shave function of the Upscale by 2 with phases 0.25 and 0.75 u16 to u16 filter.

5.198 Upscale by 2 with phases 0.25 and 0.75 fp16 to u8

This filter performs bilinear upscale with 2 factor; Horizontal and Vertical directions; phases 0.25 and 0.75.

Functions

- void **SVU_SYM()** **svuScale2xBilinHV_Fp16U8_phase025_075** (**SippFilter** *fptr)
Shave function of the Upscale by 2 with phases 0.25 and 0.75 fp16 to u8 filter.
- **SHAVE_SYM_EXPORT** (**svuScale2xBilinHV_Fp16U8_phase025_075**)

5.198.1 Detailed Description

This filter performs bilinear upscale with 2 factor; Horizontal and Vertical directions; phases 0.25 and 0.75.

Flags:

SIPP_RESIZE

Output data type(s):

UInt8

Preserve:

numPlanes

Inputs:

- datatypes: half; kernels: 2x2

Mandatory inputs:

1

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvISP/kernels/scale2xBilinHV025_Fp16U8/arch//shave/src/scale2x-BilinHV025_Fp16U8.asm, /components/kernelLib/MvISP/kernels/scale2xBilinHV075_Fp16U8/arch//shave/src/scale2xBilinHV075_Fp16U8.asm

5.198.2 Function Documentation

SHAVE_SYM_EXPORT (**svuScale2xBilinHV_Fp16U8_phase025_075**)

void **SVU_SYM()** **svuScale2xBilinHV_Fp16U8_phase025_075** (**SippFilter** * fptr)

Shave function of the Upscale by 2 with phases 0.25 and 0.75 fp16 to u8 filter.

5.199 Upscale by 2 with phases 0.25 and 0.75 u8 to u8

This filter performs bilinear upscale with 2 factor; Horizontal and Vertical directions; phases 0.25 and 0.75.

Functions

- void **SVU_SYM**() **svuScale2xBilinHV_U8ToU8_phase025_075** (**SippFilter** *fptr)
Shave function of the Upscale by 2 with phases 0.25 and 0.75 u8 to u8 filter.
- **SHAVE_SYM_EXPORT** (**svuScale2xBilinHV_U8ToU8_phase025_075**)

5.199.1 Detailed Description

This filter performs bilinear upscale with 2 factor; Horizontal and Vertical directions; phases 0.25 and 0.75.

Flags:

SIPP_RESIZE

Output data type(s):

UInt8

Preserve:

numPlanes, dataType

Inputs:

- datatypes: UInt8; kernels: 2x2

Mandatory inputs:

1

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvISP/kernels/scale2xBilinHV025_U8ToU8/arch//shave/src/scale2x-BilinHV025_U8ToU8.asm, /components/kernelLib/MvISP/kernels/scale2xBilinHV075_U8ToU8/arch//shave/src/scale2xBilinHV075_U8ToU8.asm

5.199.2 Function Documentation

SHAVE_SYM_EXPORT (**svuScale2xBilinHV_U8ToU8_phase025_075**)

void **SVU_SYM**() **svuScale2xBilinHV_U8ToU8_phase025_075** (**SippFilter** * fptr)

Shave function of the Upscale by 2 with phases 0.25 and 0.75 u8 to u8 filter.

5.200 Lanczos Horizontal Upscale by 2

This filter applies a lanczos upscale, with factor 2; Horizontal direction.

Functions

- void **SVU_SYM()** **svuScl2xLancH** (**SippFilter** *fptr)
Shave function of the [Lanczos Horizontal Upscale by 2](#) filter.
- **SHAVE_SYM_EXPORT** (**svuScl2xLancH**)

5.200.1 Detailed Description

This filter applies a lanczos upscale, with factor 2; Horizontal direction.

Preserve:

numPlanes, dataType

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 1x4

5.200.2 Function Documentation

SHAVE_SYM_EXPORT (**svuScl2xLancH**)

void **SVU_SYM()** **svuScl2xLancH** (**SippFilter** * fptr)

Shave function of the [Lanczos Horizontal Upscale by 2](#) filter.

5.201 Lanczos Upscale by 2

This filter applies a lanczos upscale, with factor 2; Horizontal and Vertical direction.

Functions

- void **SVU_SYM()** **svuScl2xLancHV** (**SippFilter** *fptr)
Shave function of the [Lanczos Upscale by 2](#) filter.
- **SHAVE_SYM_EXPORT** (**svuScl2xLancHV**)

5.201.1 Detailed Description

This filter applies a lanczos upscale, with factor 2; Horizontal and Vertical direction.

Preserve:

numPlanes, dataType

Flags:

SIPP_RESIZE

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 4x4

5.201.2 Function Documentation

SHAVE_SYM_EXPORT (**svuScl2xLancHV**)

void **SVU_SYM()** **svuScl2xLancHV** (**SippFilter** * fptr)

Shave function of the [Lanczos Upscale by 2](#) filter.

5.202 Lanczos Vertical Upscale by 2

This filter applies a lanczos upscale, with factor 2; Vertical direction.

Functions

- void **SVU_SYM()** **svuScl2xLancV** (**SippFilter** *fptr)
Shave function of the [Lanczos Vertical Upscale by 2](#) filter.
- **SHAVE_SYM_EXPORT** (**svuScl2xLancV**)

5.202.1 Detailed Description

This filter applies a lanczos upscale, with factor 2; Vertical direction.

Preserve:

numPlanes, dataType

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 4x1

5.202.2 Function Documentation

SHAVE_SYM_EXPORT (**svuScl2xLancV**)

void **SVU_SYM()** **svuScl2xLancV** (**SippFilter** * fptr)

Shave function of the [Lanczos Vertical Upscale by 2](#) filter.

5.203 Arbitrary Downscale

This filter performs bilinear scale, arbitrary X and Y scale factors.

Functions

- void **SVU_SYM()** **svuSclBilinArb** (**SippFilter** *fptr)
Shave function of the [Arbitrary Downscale](#) filter.
- **SHAVE_SYM_EXPORT** (**svuSclBilinArb**)

5.203.1 Detailed Description

This filter performs bilinear scale, arbitrary X and Y scale factors.

Preserve:

numPlanes, dataType

Flags:

SIPP_RESIZE

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 2x2

5.203.2 Function Documentation

SHAVE_SYM_EXPORT (**svuSclBilinArb**)

void **SVU_SYM()** **svuSclBilinArb** (**SippFilter** * fptr)

Shave function of the [Arbitrary Downscale](#) filter.

5.204 scharr_fp16

This kernel performs scharr edge detection operator.

Functions

- void **SVU_SYM()** **svuScharr_fp16** (**SippFilter** *fptr)
*Shave function of the **scharr_fp16** filter.*
- **SHAVE_SYM_EXPORT** (**svuScharr_fp16**)

5.204.1 Detailed Description

This kernel performs scharr edge detection operator.

Output data type(s):

half

Inputs:

- datatypes: half; kernels: 3x3

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/scharr_fp16/arch//shave/src/scharr_fp16.asm

5.204.2 Function Documentation

SHAVE_SYM_EXPORT (**svuScharr_fp16**)

void **SVU_SYM()** **svuScharr_fp16** (**SippFilter** * fptr)

Shave function of the **scharr_fp16** filter.

5.205 Laplacian 3x3 Fp16 To Fp16

The filter applies a Laplacian filter with custom size.

Functions

- void **SVU_SYM()** **svusLaplacian3x3Fp16ToFp16** (**SippFilter** *fptr)
*Shave function of the **Laplacian 3x3 Fp16 To Fp16** filter.*
- **SHAVE_SYM_EXPORT** (**svusLaplacian3x3Fp16ToFp16**)

5.205.1 Detailed Description

The filter applies a Laplacian filter with custom size.

Output data type(s):

half

Inputs:

- datatypes: half; kernels: 3x3

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/sLaplacian3x3Fp16ToFp16/arch//shave/src/sLaplacian3x3-Fp16ToFp16.asm

5.205.2 Function Documentation

SHAVE_SYM_EXPORT (**svusLaplacian3x3Fp16ToFp16**)

void **SVU_SYM()** **svusLaplacian3x3Fp16ToFp16** (**SippFilter** * fptr)

Shave function of the **Laplacian 3x3 Fp16 To Fp16** filter.

5.206 Sobel

This filter performs sobel edge detection operator.

Functions

- void **SVU_SYM()** **svuSobel** (**SippFilter** *fptr)
*Shave function of the **Sobel** filter.*
- **SHAVE_SYM_EXPORT** (**svuSobel**)

5.206.1 Detailed Description

This filter performs sobel edge detection operator.

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 3x3

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/sobel/arch//shave/src/sobel.asm

5.206.2 Function Documentation

SHAVE_SYM_EXPORT (**svuSobel**)

void **SVU_SYM()** **svuSobel** (**SippFilter** * fptr)

Shave function of the **Sobel** filter.

5.207 Sum of Squared Differences 11x11

This filter performs sum of squared differences (SSD), the differences are squared and aggregated within a square window (11x11 in this case).

Functions

- void **SVU_SYM**() **svuSSD11x11** (**SippFilter** *fptr)
*Shave function of the **Sum of Squared Differences 11x11** filter.*
- **SHAVE_SYM_EXPORT** (**svuSSD11x11**)

5.207.1 Detailed Description

This filter performs sum of squared differences (SSD), the differences are squared and aggregated within a square window (11x11 in this case).

Output data type(s):

UInt8

Inputs:

- name: input1; datatypes: UInt8; kernels: 11x11
- name: input2; datatypes: UInt8; kernels: 11x11

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/sumOfSquaredDiff11x11/arch//shave/src/sumOfSquared-Diff11x11.asm

5.207.2 Function Documentation

SHAVE_SYM_EXPORT (**svuSSD11x11**)

void **SVU_SYM**() **svuSSD11x11** (**SippFilter** * fptr)

Shave function of the **Sum of Squared Differences 11x11** filter.

5.208 Sum of Squared Differences 5x5

This filter performs sum of squared differences (SSD), the differences are squared and aggregated within a square window (5x5 in this case).

Functions

- void **SVU_SYM**() **svuSSD5x5** (**SippFilter** *fptr)
Shave function of the [Sum of Squared Differences 5x5](#) filter.
- **SHAVE_SYM_EXPORT** (**svuSSD5x5**)

5.208.1 Detailed Description

This filter performs sum of squared differences (SSD), the differences are squared and aggregated within a square window (5x5 in this case).

Output data type(s):

UInt8

Inputs:

- name: input1; datatypes: UInt8; kernels: 5x5
- name: input2; datatypes: UInt8; kernels: 5x5

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/sumOfSquaredDiff5x5/arch//shave/src/sumOfSquared-Diff5x5.asm

5.208.2 Function Documentation

SHAVE_SYM_EXPORT (**svuSSD5x5**)

void **SVU_SYM**() **svuSSD5x5** (**SippFilter** * fptr)

Shave function of the [Sum of Squared Differences 5x5](#) filter.

5.209 Sum of Squared Differences 7x7 (U8 to U32)

This filter performs sum of squared differences (SSD) to a U32 output array, the differences are squared and aggregated within a square window (7x7 in this case).

Functions

- void **SVU_SYM**() **svuSSD7x7U8ToU32** (**SippFilter** *fptr)
*Shave function of the **Sum of Squared Differences 7x7 (U8 to U32)** filter.*
- **SHAVE_SYM_EXPORT** (**svuSSD7x7U8ToU32**)

5.209.1 Detailed Description

This filter performs sum of squared differences (SSD) to a U32 output array, the differences are squared and aggregated within a square window (7x7 in this case).

Preserve:

numPlanes, imgSize

Output data type(s):

UInt32

Inputs:

- name: input1; datatypes: UInt8; kernels: 7x7
- name: input2; datatypes: UInt8; kernels: 7x7

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/sumOfSquaredDiff7x7U8ToU32/arch//shave/src/sumOf-SquaredDiff7x7U8ToU32.asm

5.209.2 Function Documentation

SHAVE_SYM_EXPORT (**svuSSD7x7U8ToU32**)

void **SVU_SYM**() **svuSSD7x7U8ToU32** (**SippFilter** * fptr)

Shave function of the **Sum of Squared Differences 7x7 (U8 to U32)** filter.

5.210 Sum of Squared Differences 7x7

This filter performs sum of squared differences (SSD), the differences are squared and aggregated within a square window (7x7 in this case).

Functions

- void **SVU_SYM()** **svuSsdPointLine7x7U8U32** (**SippFilter** *fptr)
*Shave function of the **Sum of Squared Differences 7x7** filter.*
- **SHAVE_SYM_EXPORT** (**svuSsdPointLine7x7U8U32**)

5.210.1 Detailed Description

This filter performs sum of squared differences (SSD), the differences are squared and aggregated within a square window (7x7 in this case).

Preserve:

numPlanes

Output data type(s):

UInt32

Inputs:

- name: input1; datatypes: UInt8; kernels: 7x7
- name: input2; datatypes: UInt8; kernels: 7x7

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/ssdPointLine7x7U8U32/arch//shave/src/ssdPointLine7x7-U8U32.asm

5.210.2 Function Documentation

SHAVE_SYM_EXPORT (**svuSsdPointLine7x7U8U32**)

void **SVU_SYM()** **svuSsdPointLine7x7U8U32** (**SippFilter** * fptr)

Shave function of the **Sum of Squared Differences 7x7** filter.

5.211 Threshold

This filter computes the output image based on a threshold value and a threshold type.

Data Structures

- struct `ThresholdParam`
Parameter structure of the `Threshold` filter.

Enumerations

- enum {
 `Thresh_To_Zero` = 0, `Thresh_To_Zero_Inv` = 1, `Thresh_To_Binary` = 2, `Thresh_To_Binary_Inv` = 3,
 `Thresh_Trunc` = 4 }

Functions

- void `SVU_SYM()` `svuThreshold` (`SippFilter` *fptr)
Shave function of the `Threshold` filter.
- `SHAVE_SYM_EXPORT` (`svuThreshold`)

5.211.1 Detailed Description

This filter computes the output image based on a threshold value and a threshold type.

Preserve:

numPlanes, imgSize

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 1x1

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/thresholdKernel/arch//shave/src/thresholdKernel.asm

5.211.2 Enumeration Type Documentation

anonymous enum

Enumerator

Thresh_To_Zero

Thresh_To_Zero_Inv

Thresh_To_Binary

Thresh_To_Binary_Inv

Thresh_Trunc

5.211.3 Function Documentation

SHAVE_SYM_EXPORT (**svuThreshold**)

void **SVU_SYM**() svuThreshold (**SippFilter** * fptr)

Shave function of the **Threshold** filter.

5.212 Threshold Binary Range

This filter sets output to 0xFF if pixel value is in specified range, otherwise output is 0.

Data Structures

- struct `ThresholdBinaryRangeParam`
Parameter structure of the [Threshold Binary Range](#) filter.

Functions

- void `SVU_SYM() svuThresholdBinaryRange (SippFilter *fptr)`
Shave function of the [Threshold Binary Range](#) filter.
- `SHAVE_SYM_EXPORT` (`svuThresholdBinaryRange`)

5.212.1 Detailed Description

This filter sets output to 0xFF if pixel value is in specified range, otherwise output is 0.

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 1x1

Path to external ASM file(s) used in the filter implementation:

`/components/kernelLib/MvCV/kernels/thresholdBinaryRange/arch//shave/src/thresholdBinary-Range.asm`

5.212.2 Function Documentation

`SHAVE_SYM_EXPORT (svuThresholdBinaryRange)`

void `SVU_SYM() svuThresholdBinaryRange (SippFilter * fptr)`

Shave function of the [Threshold Binary Range](#) filter.

5.213 Threshold Binary U8

This filter sets output to 0 if threshold value is less than input value and to 0xFF if threshold value is greater than input value.

Data Structures

- struct [ThresholdBinaryU8Param](#)
Parameter structure of the [Threshold Binary U8](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuThresholdBinaryU8](#) ([SippFilter](#) *fptr)
Shave function of the [Threshold Binary U8](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuThresholdBinaryU8](#))

5.213.1 Detailed Description

This filter sets output to 0 if threshold value is less than input value and to 0xFF if threshold value is greater than input value.

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 1x1

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/thresholdBinaryU8/arch//shave/src/thresholdBinaryU8.asm

5.213.2 Function Documentation

[SHAVE_SYM_EXPORT](#) ([svuThresholdBinaryU8](#))

void [SVU_SYM\(\)](#) [svuThresholdBinaryU8](#) ([SippFilter](#) * fptr)

Shave function of the [Threshold Binary U8](#) filter.

5.214 ThresholdFilter

This filter computes the output image based on a thresholdFilter value.

Data Structures

- struct **ThresholdFilterParam**
Parameter structure of the [ThresholdFilter](#) filter.

Functions

- void **SVU_SYM()** **svuThresholdFilter** (**SippFilter** *fptr)
Shave function of the [ThresholdFilter](#) filter.
- **SHAVE_SYM_EXPORT** (**svuThresholdFilter**)

5.214.1 Detailed Description

This filter computes the output image based on a thresholdFilter value.

Output data type(s):

UInt16

Preserve:

numPlanes

Inputs:

- datatypes: float; kernels: 1x1

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/thresholdFilter/arch//shave/src/thresholdFilter.asm

5.214.2 Function Documentation

SHAVE_SYM_EXPORT (**svuThresholdFilter**)

void **SVU_SYM()** **svuThresholdFilter** (**SippFilter** * fptr)

Shave function of the [ThresholdFilter](#) filter.

5.215 Undistort

This filter applies undistort using Brown's distortion model for known lens distortion coefficients.

Data Structures

- struct **UndistortBParam**
*Parameter structure of the **Undistort** filter.*

Functions

- void **SVU_SYM()** **svuUndistortBrown** (**SippFilter** *fptr)
*Shave function of the **Undistort** filter.*
- **SHAVE_SYM_EXPORT** (**svuUndistortBrown**)

5.215.1 Detailed Description

This filter applies undistort using Brown's distortion model for known lens distortion coefficients.

Output data type(s):

UInt8

Inputs:

- datatypes: UInt8; kernels: 128x2

5.215.2 Function Documentation

SHAVE_SYM_EXPORT (**svuUndistortBrown**)

void **SVU_SYM()** **svuUndistortBrown** (**SippFilter** * fptr)

Shave function of the **Undistort** filter.

5.216 White Balance Bayer GBRG

This filter applies white balance gains for BayerGBRG input.

Data Structures

- struct `WhiteBalanceBayerGBRGParam`
Parameter structure of the [White Balance Bayer GBRG](#) filter.

Functions

- void `SVU_SYM() svuWhiteBalanceBayerGBRG (SippFilter *fptr)`
Shave function of the [White Balance Bayer GBRG](#) filter.
- `SHAVE_SYM_EXPORT (svuWhiteBalanceBayerGBRG)`

5.216.1 Detailed Description

This filter applies white balance gains for BayerGBRG input.

Output data type(s):

UInt16

Inputs:

- datatypes: UInt16; kernels: 1x1

Path to external ASM file(s) used in the filter implementation:

`/components/kernelLib/MvCV/kernels/whiteBalanceBayerGBRG/arch//shave/src/whiteBalance-BayerGBRG.asm`

5.216.2 Function Documentation

`SHAVE_SYM_EXPORT (svuWhiteBalanceBayerGBRG)`

void `SVU_SYM() svuWhiteBalanceBayerGBRG (SippFilter * fptr)`

Shave function of the [White Balance Bayer GBRG](#) filter.

5.217 White Balance RGB

This filter applies white balance gains for RGB input.

Data Structures

- struct **WhiteBalanceRGBParam**
*Parameter structure of the **White Balance RGB** filter.*

Functions

- void **SVU_SYM()** **svuWhiteBalanceRGB** (**SippFilter** *fptr)
*Shave function of the **White Balance RGB** filter.*
- **SHAVE_SYM_EXPORT** (**svuWhiteBalanceRGB**)

5.217.1 Detailed Description

This filter applies white balance gains for RGB input.

Output data type(s):

UInt16

Inputs:

- datatypes: UInt16; kernels: 1x1

Path to external ASM file(s) used in the filter implementation:

/components/kernelLib/MvCV/kernels/whiteBalanceRGB/arch//shave/src/whiteBalanceRGB.asm

5.217.2 Function Documentation

SHAVE_SYM_EXPORT (**svuWhiteBalanceRGB**)

void **SVU_SYM()** **svuWhiteBalanceRGB** (**SippFilter** * fptr)

Shave function of the **White Balance RGB** filter.

5.218 XY Generator

This filter performs xyGenerator for Bicubic.

Data Structures

- struct **XYGenParam**
Parameter structure of the XY Generator filter.

Functions

- void **SVU_SYM()** **svuXYgen** (**SippFilter** *fptr)
Shave function of the XY Generator filter.
- **SHAVE_SYM_EXPORT** (**svuXYgen**)

5.218.1 Detailed Description

This filter performs xyGenerator for Bicubic.

Output data type(s):

UInt16

Inputs:

- datatypes: UInt16; kernels: 1x1

5.218.2 Function Documentation

SHAVE_SYM_EXPORT (**svuXYgen**)

void **SVU_SYM()** **svuXYgen** (**SippFilter** * fptr)

Shave function of the XY Generator filter.

5.219 DMA

DMA In/Out filter.

Data Structures

- struct [DmaParam](#)
Parameter structure of the [DMA](#) filter.

5.219.1 Detailed Description

DMA In/Out filter.

Output data type(s):

UInt8, UInt16, UInt32, UInt64, Int8, Int16, Int32, half, fp16, fp32

Filter function:

SIPP_DMA_ID

Inputs:

- datatypes: UInt8, UInt16, UInt32, UInt64, Int8, Int16, Int32, half, fp16, fp32; kernels: 1x1

Chapter 6

Data Structure Documentation

6.1 AccumulateWeightedParam Struct Reference

Parameter structure of the [Accumulate Weighted](#) filter.

```
#include <accumulateWeighted.h>
```

Data Fields

- float [alpha](#)
Weight of the input image must be a fp32 between 0 and 1.

6.1.1 Detailed Description

Parameter structure of the [Accumulate Weighted](#) filter.

6.1.2 Field Documentation

float [AccumulateWeightedParam::alpha](#)

Weight of the input image must be a fp32 between 0 and 1.

Referenced by `svuAccumulateWeighted()`.

6.2 ae_patch_stats Struct Reference

```
#include <sippTypes.h>
```

Data Fields

- uint32_t [count](#) [4]
- uint32_t [accum](#) [4]
- uint32_t [alt_accum](#) [4]

6.2.1 Field Documentation

uint32_t ae_patch_stats::accum[4]

uint32_t ae_patch_stats::alt_accum[4]

uint32_t ae_patch_stats::count[4]

6.3 AeAwbStatsCfg Struct Reference

```
#include <combDecimDemosaicAwbGainsStats.h>
```

Data Fields

- [UInt32 firstPatchX](#)
- [UInt32 firstPatchY](#)
- [UInt32 patchWidth](#)
- [UInt32 patchHeight](#)
- [UInt32 patchGapX](#)
- [UInt32 patchGapY](#)
- [UInt32 nPatchesX](#)
- [UInt32 nPatchesY](#)
- [UInt32 satThresh](#)

6.3.1 Field Documentation

UInt32 AeAwbStatsCfg::firstPatchX

UInt32 AeAwbStatsCfg::firstPatchY

UInt32 AeAwbStatsCfg::nPatchesX

Referenced by svuCombDecimDemosaicAwbGainsStats().

UInt32 AeAwbStatsCfg::nPatchesY

UInt32 AeAwbStatsCfg::patchGapX

UInt32 AeAwbStatsCfg::patchGapY

UInt32 AeAwbStatsCfg::patchHeight

UInt32 AeAwbStatsCfg::patchWidth

UInt32 AeAwbStatsCfg::satThresh

Referenced by svuCombDecimDemosaicAwbGainsStats().

6.4 AF_paxel_statistics Struct Reference

```
#include <sippTypes.h>
```

Data Fields

- int32_t UNDEFINED
- int32_t sum_all_green
- int32_t filter1_sum_max_green
- int32_t filter2_sum_max_green
- int32_t filter1_number_of_used_pixels_green
- int32_t filter1_sum_green
- int32_t filter2_number_of_used_pixels_green
- int32_t filter2_sum_green

6.4.1 Field Documentation

int32_t AF_paxel_statistics::filter1_number_of_used_pixels_green

int32_t AF_paxel_statistics::filter1_sum_green

int32_t AF_paxel_statistics::filter1_sum_max_green

int32_t AF_paxel_statistics::filter2_number_of_used_pixels_green

int32_t AF_paxel_statistics::filter2_sum_green

int32_t AF_paxel_statistics::filter2_sum_max_green

int32_t AF_paxel_statistics::sum_all_green

int32_t AF_paxel_statistics::UNDEFINED

6.5 Bilateral5x5Param Struct Reference

```
#include <bilateral5x5.h>
```

Data Fields

- UInt16 * sigma

6.5.1 Field Documentation

UInt16* Bilateral5x5Param::sigma

Referenced by svuBilateral5x5().

6.6 BoxFilter11x11Param Struct Reference

Parameter structure of the [Box Filter 11x11](#) filter.

```
#include <boxFilter11x11.h>
```

Data Fields

- [UInt32 normalize](#)
1 to normalize to kernel size, 0 otherwise

6.6.1 Detailed Description

Parameter structure of the [Box Filter 11x11](#) filter.

6.6.2 Field Documentation

UInt32 [BoxFilter11x11Param::normalize](#)

1 to normalize to kernel size, 0 otherwise

Referenced by [svuBoxFilter11x11\(\)](#).

6.7 BoxFilter13x13Param Struct Reference

Parameter structure of the [Box Filter 13x13](#) filter.

```
#include <boxFilter13x13.h>
```

Data Fields

- [UInt32 normalize](#)
1 to normalize to kernel size, 0 otherwise

6.7.1 Detailed Description

Parameter structure of the [Box Filter 13x13](#) filter.

6.7.2 Field Documentation

UInt32 [BoxFilter13x13Param::normalize](#)

1 to normalize to kernel size, 0 otherwise

Referenced by [svuBoxFilter13x13\(\)](#).

6.8 BoxFilter15x15Param Struct Reference

Parameter structure of the [Box Filter 15x15](#) filter.

```
#include <boxFilter15x15.h>
```

Data Fields

- [UInt32 normalize](#)
1 to normalize to kernel size, 0 otherwise

6.8.1 Detailed Description

Parameter structure of the [Box Filter 15x15](#) filter.

6.8.2 Field Documentation

UInt32 [BoxFilter15x15Param::normalize](#)

1 to normalize to kernel size, 0 otherwise

Referenced by [svuBoxFilter15x15\(\)](#).

6.9 BoxFilter3x3Param Struct Reference

Parameter structure of the [Box Filter 3x3](#) filter.

```
#include <boxFilter3x3.h>
```

Data Fields

- [UInt32 normalize](#)
1 to normalize to kernel size, 0 otherwise

6.9.1 Detailed Description

Parameter structure of the [Box Filter 3x3](#) filter.

6.9.2 Field Documentation

UInt32 [BoxFilter3x3Param::normalize](#)

1 to normalize to kernel size, 0 otherwise

Referenced by [svuBoxFilter3x3\(\)](#).

6.10 BoxFilter5x5Param Struct Reference

Parameter structure of the [Box Filter 5x5](#) filter.

```
#include <boxFilter5x5.h>
```

Data Fields

- [UInt32 normalize](#)
1 to normalize to kernel size, 0 otherwise

6.10.1 Detailed Description

Parameter structure of the [Box Filter 5x5](#) filter.

6.10.2 Field Documentation

UInt32 [BoxFilter5x5Param::normalize](#)

1 to normalize to kernel size, 0 otherwise

Referenced by [svuBoxFilter5x5\(\)](#).

6.11 BoxFilter7x7Param Struct Reference

Parameter structure of the [Box Filter 7x7](#) filter.

```
#include <boxFilter7x7.h>
```

Data Fields

- [UInt32 normalize](#)
1 to normalize to kernel size, 0 otherwise

6.11.1 Detailed Description

Parameter structure of the [Box Filter 7x7](#) filter.

6.11.2 Field Documentation

UInt32 [BoxFilter7x7Param::normalize](#)

1 to normalize to kernel size, 0 otherwise

Referenced by [svuBoxFilter7x7\(\)](#).

6.12 BoxFilter9x9Param Struct Reference

Parameter structure of the [Box Filter 9x9](#) filter.

```
#include <boxFilter9x9.h>
```

Data Fields

- [UInt32 normalize](#)
1 to normalize to kernel size, 0 otherwise

6.12.1 Detailed Description

Parameter structure of the [Box Filter 9x9](#) filter.

6.12.2 Field Documentation

UInt32 [BoxFilter9x9Param::normalize](#)

1 to normalize to kernel size, 0 otherwise

Referenced by [svuBoxFilter9x9\(\)](#).

6.13 BoxFilterParam Struct Reference

Parameter structure of the [Generic Box Filter](#) filter.

```
#include <boxFilter.h>
```

Data Fields

- [boxDataFmt dataFormat](#)
data format from enum boxDataFmt
- [UInt32 filterSizeH](#)
kernel width
- [UInt32 filterSizeV](#)
kernel height
- [UInt32 normalize](#)
1 to normalize to kernel size, 0 otherwise

6.13.1 Detailed Description

Parameter structure of the [Generic Box Filter](#) filter.

6.13.2 Field Documentation

boxDataFmt BoxFilterParam::dataFormat

data format from enum boxDataFmt

Referenced by svuBoxFilter().

UInt32 BoxFilterParam::filterSizeH

kernel width

Referenced by svuBoxFilter().

UInt32 BoxFilterParam::filterSizeV

kernel height

Referenced by svuBoxFilter().

UInt32 BoxFilterParam::normalize

1 to normalize to kernel size, 0 otherwise

Referenced by svuBoxFilter().

6.14 cannyEdgeDetectionParam Struct Reference

Parameter structure of the [Canny Edge Detection](#) filter.

```
#include <cannyEdgeDetection.h>
```

Data Fields

- **UInt32** threshold1
lower threshold - value between 0-255
- **UInt32** threshold2
upper threshold - value between 0-255

6.14.1 Detailed Description

Parameter structure of the [Canny Edge Detection](#) filter.

6.14.2 Field Documentation

UInt32 cannyEdgeDetectionParam::threshold1

lower threshold - value between 0-255

Referenced by svuCannyEdgeDetection().

UInt32 cannyEdgeDetectionParam::threshold2

upper threshold - value between 0-255

Referenced by svuCannyEdgeDetection().

6.15 CensusMatching32Param Struct Reference

Parameter structure of the **censusMatching32** filter.

```
#include <censusMatching32.h>
```

Data Fields

- **UInt32** flag
flag that enables right crossing (by default left crossing)

6.15.1 Detailed Description

Parameter structure of the **censusMatching32** filter.

6.15.2 Field Documentation

UInt32 CensusMatching32Param::flag

flag that enables right crossing (by default left crossing)

Referenced by svuCensusMatching32().

6.16 CensusMatching64Param Struct Reference

Parameter structure of the **censusMatching64** filter.

```
#include <censusMatching64.h>
```

Data Fields

- **UInt32** flag
flag that enables right crossing (by default left crossing)

6.16.1 Detailed Description

Parameter structure of the **censusMatching64** filter.

6.16.2 Field Documentation

UInt32 CensusMatching64Param::flag

flag that enables right crossing (by default left crossing)

Referenced by svuCensusMatching64().

6.17 CensusMatchingPyrParam Struct Reference

Parameter structure of the **censusMatchingPyr** filter.

```
#include <censusMatchingPyr.h>
```

Data Fields

- **UInt8** * **predicted**

6.17.1 Detailed Description

Parameter structure of the **censusMatchingPyr** filter.

6.17.2 Field Documentation

UInt8* CensusMatchingPyrParam::predicted

Referenced by svuCensusMatchingPyr().

6.18 ChannelExtractParam Struct Reference

Parameter structure of the **channelExtract** filter.

```
#include <channelExtract.h>
```

Data Fields

- **UInt32** **plane**
number 0 to extract plane R, 1 for extracting G, 2 for extracting B

6.18.1 Detailed Description

Parameter structure of the **channelExtract** filter.

6.18.2 Field Documentation

UInt32 ChannelExtractParam::plane

number 0 to extract plane R, 1 for extracting G, 2 for extracting B

Referenced by svuChannelExtract().

6.19 ChrDnsParam Struct Reference

Parameter structure of the chormadns filter.

```
#include <sippHwDefs_ma2x5x.h>
```

Data Fields

- **UInt32 frmDim**
Reserved field yes.
- **UInt32 cfg**
configuration bitfield(see SIPP_CHROMA_CFG_ADR)
- **UInt32 thr [2]**
thresholds(see SIPP_CHROMA_THRESH_ADR, SIPP_CHROMA_THRESH2_ADR)
- **UInt32 greyPt**
Sets Color to desaturate towards (see SIPP_CHROMA_GREY_POINT_ADR)
- **UInt32 chrCoefs**
Sets coefficinets of symmetric 3x3 pre filter (see SIPP_CHROMA_CHROMA_COEFFFS_ADR)

6.19.1 Detailed Description

Parameter structure of the chormadns filter.

6.19.2 Field Documentation

UInt32 ChrDnsParam::cfg

configuration bitfield(see SIPP_CHROMA_CFG_ADR)

UInt32 ChrDnsParam::chrCoefs

Sets coefficinets of symmetric 3x3 pre filter (see SIPP_CHROMA_CHROMA_COEFFFS_ADR)

UInt32 ChrDnsParam::frmDim

Reserved field yes.

UInt32 ChrDnsParam::greyPt

Sets Color to desaturate towards (see SIPP_CHROMA_GREY_POINT_ADR)

UInt32 ChrDnsParam::thr[2]

thresholds(see SIPP_CHROMA_THRESH_ADR, SIPP_CHROMA_THRESH2_ADR)

6.20 ChrGenParam Struct Reference

Parameter structure of the filter.

```
#include <genChroma.h>
```

Data Fields

- int [epsilon](#)
epsilon value. Range is [0, 255], but it's normally a very small non-zero

6.20.1 Detailed Description

Parameter structure of the filter.

6.20.2 Field Documentation

[int ChrGenParam::epsilon](#)

epsilon value. Range is [0, 255], but it's normally a very small non-zero

6.21 ChrGenSSParam Struct Reference

Parameter structure of the [Generate Chroma with subsampling](#) filter.

```
#include <genChromaSS.h>
```

Data Fields

- int [epsilon](#)
epsilon value. Range is [0, 255], but it's normally a very small non-zero
- float [scale](#) [3]
scaling factor. 3 element array: 0 for R channel, 1 for G channel, 2 for B channel

6.21.1 Detailed Description

Parameter structure of the [Generate Chroma with subsampling](#) filter.

6.21.2 Field Documentation

[int ChrGenSSParam::epsilon](#)

epsilon value. Range is [0, 255], but it's normally a very small non-zero

[float ChrGenSSParam::scale\[3\]](#)

scaling factor. 3 element array: 0 for R channel, 1 for G channel, 2 for B channel

6.22 ChromaBlkParam Struct Reference

Parameter structure of the **Chroma Block** filter.

```
#include <chromaBlock.h>
```

Data Fields

- float * **ccm**
pointer to 3x3 color correction matrix
- UInt8 * **rangeLut**
range look-up table

6.22.1 Detailed Description

Parameter structure of the **Chroma Block** filter.

6.22.2 Field Documentation

float* ChromaBlkParam::ccm

pointer to 3x3 color correction matrix

Referenced by svuChromaBlock().

UInt8* ChromaBlkParam::rangeLut

range look-up table

Referenced by svuChromaBlock().

6.23 cmxRegUsage Struct Reference

```
#include <sippTypesPrivate.h>
```

Data Fields

- u32 **cmxRegionIdx**
- u32 **usedCount**
- u8 **fullCmxSlice**

6.23.1 Field Documentation

u32 cmxRegUsage::cmxRegionIdx

Referenced by sippAllocCmxLineBuffers().

u8 cmxRegUsage::fullCmxSlice

Referenced by sippAllocCmxLineBuffers().

u32 cmxRegUsage::usedCount

Referenced by sippAllocCmxLineBuffers().

6.24 ColCombParam Struct Reference

Parameter structure of the colorcomb filter.

```
#include <sippHwDefs_ma2x5x.h>
```

Data Fields

- **UInt32** frmDim
Reserved field yes.
- **UInt32** cfg
configuration bit field(see SIPP_CC_CFG_ADR)
- **UInt32** krgb [2]
*coefficient array for luma+chroma recombination(see SIPP_CC_KRGB*_ADR (4.8))*
- **UInt32** ccm [5]
*Color adjustment matrix(see SIPP_CC_CCM*_ADR (6.10))*
- **UInt32** ccOffs
- void * threeDLut
- **UInt8** lutFormat

6.24.1 Detailed Description

Parameter structure of the colorcomb filter.

6.24.2 Field Documentation

UInt32 ColCombParam::ccm[5]

Color adjustment matrix(see SIPP_CC_CCM*_ADR (6.10))

Referenced by packColCombCCM().

UInt32 ColCombParam::ccOffs

UInt32 ColCombParam::cfg

configuration bit field(see SIPP_CC_CFG_ADR)

UInt32 ColCombParam::frmDim

Reserved field yes.

UInt32 ColCombParam::krb[2]

coefficient array for luma+chroma recombination(see SIPP_CC_KRGB*_ADR (4.8))

UInt8 ColCombParam::lutFormat

void* ColCombParam::threeDLut

6.25 CombDecimAwbGainsParam Struct Reference

```
#include <combDecimDemosaicAwbGains.h>
```

Data Fields

- unsigned short [gains](#) [3]
- unsigned int [bayerOrder](#)

6.25.1 Field Documentation

unsigned int CombDecimAwbGainsParam::bayerOrder

Referenced by svuCombDecimDemosaicAwbGains().

unsigned short CombDecimAwbGainsParam::gains[3]

Referenced by svuCombDecimDemosaicAwbGains().

6.26 CombDecimStatsGainsParam Struct Reference

```
#include <combDecimDemosaicAwbGainsStats.h>
```

Data Fields

- [AeAwbStatsCfg](#) * [satPixelsStats](#)
- **UInt32** [runNr](#)
- unsigned int * [statsOutput](#)
- unsigned short * [paxelsIntervalsVert](#)
- unsigned short * [paxelsIntervalsHz](#)
- unsigned int [crtPaxelLine](#)
- unsigned int [gains](#) [3]

6.26.1 Field Documentation

`unsigned int CombDecimStatsGainsParam::crtPaxelLine`

Referenced by `svuCombDecimDemosaicAwbGainsStats()`.

`unsigned int CombDecimStatsGainsParam::gains[3]`

Referenced by `svuCombDecimDemosaicAwbGainsStats()`.

`unsigned short* CombDecimStatsGainsParam::paxelsIntervalsHz`

Referenced by `svuCombDecimDemosaicAwbGainsStats()`.

`unsigned short* CombDecimStatsGainsParam::paxelsIntervalsVert`

Referenced by `svuCombDecimDemosaicAwbGainsStats()`.

`UInt32 CombDecimStatsGainsParam::runNr`

Referenced by `svuCombDecimDemosaicAwbGainsStats()`.

`AeAwbStatsCfg* CombDecimStatsGainsParam::satPixelsStats`

Referenced by `svuCombDecimDemosaicAwbGainsStats()`.

`unsigned int* CombDecimStatsGainsParam::statsOutput`

Referenced by `svuCombDecimDemosaicAwbGainsStats()`.

6.27 CommInfo Struct Reference

```
#include <sippTypes.h>
```

Data Fields

- `u32 sliceFirst`
- `u32 sliceLast`
- `u32 sliceSize`
- `u32 curFrame`
- `SippPipeline * pl`
- `u32 numShaves`

6.27.1 Field Documentation

u32 `CommInfo::curFrame`

Referenced by `sippDbgDumpAsmOffsets()`, and `sippGenericRuntimeProcessIters()`.

u32 `CommInfo::numShaves`

Referenced by `sippComputeChunkWidths()`, `sippCoreFinalisePipeline()`, `sippGenericStartUnits()`, `sippInitPipeline()`, and `sippValidatePipe()`.

SippPipeline* `CommInfo::pl`

Referenced by `sippInitDma()`, and `sippInitPipeline()`.

u32 `CommInfo::sliceFirst`

Referenced by `sippAddFilterToPipe()`, `sippAllocCmxLineBuffers()`, `sippComputeBufferProps()`, `sippComputeChunkWidths()`, `sippComputeChunkWidthsSW()`, `sippComputePaddingOffsets()`, `sippComputeSwOutCt()`, `sippCoreFinalisePipeline()`, `sippDbgDumpAsmOffsets()`, `sippDbgDumpFilterOuts()`, `sippDumpHtmlMap()`, `sippHorizontalPadding()`, `sippHWSessionAddActiveLists()`, `sippHWSessionCommand()`, `sippHWSessionRemoveActiveLists()`, `sippIbufSetup()`, `sippInitDma()`, `sippInitPipeline()`, `sippKickShaveM1PC()`, `sippKickSvus()`, `sippMapRegionMapAddrToSliceZero()`, `sippObufSetup()`, `sippSetupSvus()`, `sippStopSvus()`, `sippWaitShave()`, `svuExtAfStats()`, `svuExtStatsSatPixelsU32()`, `svuGenDnsRef()`, `svuGenDnsRefFp16()`, `svuHomography()`, `svuPadBayer5()`, `svuPadBayer5Frame()`, and `svuUndistortBrown()`.

u32 `CommInfo::sliceLast`

Referenced by `sippAllocCmxLineBuffers()`, `sippComputeBufferProps()`, `sippComputeChunkWidths()`, `sippComputeChunkWidthsSW()`, `sippComputePaddingOffsets()`, `sippComputeSwOutCt()`, `sippCoreFinalisePipeline()`, `sippDbgDumpAsmOffsets()`, `sippDumpHtmlMap()`, `sippHorizontalPadding()`, `sippInitDma()`, `sippInitPipeline()`, `sippKickShaveM1PC()`, `sippSetupSvus()`, `sippWaitShave()`, `svuExtAfStats()`, `svuExtStatsSatPixelsU32()`, `svuPadBayer5()`, and `svuPadBayer5Frame()`.

u32 `CommInfo::sliceSize`

Referenced by `sippComputeBufferProps()`, `sippComputePaddingOffsets()`, `sippComputeSwOutCt()`, `sippCoreFinalisePipeline()`, `sippDbgDumpAsmOffsets()`, `sippDumpHtmlMap()`, `sippInitPipeline()`, `sippMapRegionMapAddrToSliceZero()`, `sippSetupSvus()`, `sippUpdateDmaAddr()`, and `sippUpdateDmaAddrCQ()`.

6.28 ContrastParam Struct Reference

Parameter structure of the [Contrast](#) filter.

```
#include <contrast.h>
```

Data Fields

- float `idxLow`
index that sets the low end value for grey scale clipping
- float `scale`
scaling factor

6.28.1 Detailed Description

Parameter structure of the `Contrast` filter.

6.28.2 Field Documentation

float `ContrastParam::idxLow`

index that sets the low end value for grey scale clipping

Referenced by `svuContrast()`.

float `ContrastParam::scale`

scaling factor

Referenced by `svuContrast()`.

6.29 Conv11x11Param Struct Reference

Parameter structure of the `Convolution 11x11` filter.

```
#include <conv11x11.h>
```

Data Fields

- `UInt16 * cMat`
121 element array with fp16 values containing the 11x11 convolution matrix.

6.29.1 Detailed Description

Parameter structure of the `Convolution 11x11` filter.

6.29.2 Field Documentation

`UInt16*` `Conv11x11Param::cMat`

121 element array with fp16 values containing the 11x11 convolution matrix.

Default value:

Size of the array behind:

121

Referenced by `svuConv11x11()`.

6.30 Conv15x1Param Struct Reference

Parameter structure of the [Convolution 15x1](#) filter.

```
#include <conv15x1.h>
```

Data Fields

- [UInt16](#) * [cMat](#)
15 element array with fp16 values containing the 15x1 convolution matrix.

6.30.1 Detailed Description

Parameter structure of the [Convolution 15x1](#) filter.

6.30.2 Field Documentation

[UInt16](#)* [Conv15x1Param::cMat](#)

15 element array with fp16 values containing the 15x1 convolution matrix.

Default value:

Size of the array behind:

15

Referenced by `svuConv15x1()`.

6.31 Conv1x15Param Struct Reference

Parameter structure of the [Convolution 1x15](#) filter.

```
#include <conv1x15.h>
```

Data Fields

- [UInt16](#) * [cMat](#)
15 element array with fp16 values containing the 1x15 convolution matrix.

6.31.1 Detailed Description

Parameter structure of the [Convolution 1x15](#) filter.

6.31.2 Field Documentation

UInt16* Conv1x15Param::cMat

15 element array with fp16 values containing the 1x15 convolution matrix.

Default value:

Size of the array behind:

15

Referenced by `svuConv1x15()`.

6.32 Conv1x5Fp16ToFp16Param Struct Reference

Parameter structure of the [Convolution 1x5 Fp16ToFp16](#) filter.

```
#include <conv1x5Fp16ToFp16.h>
```

Data Fields

- **UInt16 *** cMat

5 element array with fp16 values containing the 1x5 convolution matrix.

6.32.1 Detailed Description

Parameter structure of the [Convolution 1x5 Fp16ToFp16](#) filter.

6.32.2 Field Documentation

UInt16* Conv1x5Fp16ToFp16Param::cMat

5 element array with fp16 values containing the 1x5 convolution matrix.

Default value:

Size of the array behind:

5

Referenced by `svuConv1x5Fp16ToFp16()`.

6.33 Conv1x5Param Struct Reference

Parameter structure of the [Convolution 1x5](#) filter.

```
#include <conv1x5.h>
```

Data Fields

- [UInt16](#) * [cMat](#)
5 element array with fp16 values containing the 1x5 convolution matrix.

6.33.1 Detailed Description

Parameter structure of the [Convolution 1x5](#) filter.

6.33.2 Field Documentation

[UInt16](#)* [Conv1x5Param::cMat](#)

5 element array with fp16 values containing the 1x5 convolution matrix.

Default value:

Size of the array behind:

5

Referenced by [svuConv1x5\(\)](#).

6.34 Conv1x7Fp16ToFp16Param Struct Reference

Parameter structure of the [Convolution 1x7 Fp16ToFp16](#) filter.

```
#include <conv1x7Fp16ToFp16.h>
```

Data Fields

- [UInt16](#) * [cMat](#)
7 element array with fp16 values containing the 1x7 convolution matrix.

6.34.1 Detailed Description

Parameter structure of the [Convolution 1x7 Fp16ToFp16](#) filter.

6.34.2 Field Documentation

UInt16* Conv1x7Fp16ToFp16Param::cMat

7 element array with fp16 values containing the 1x7 convolution matrix.

Default value:

Size of the array behind:

7

Referenced by svuConv1x7Fp16ToFp16().

6.35 Conv1x7Param Struct Reference

Parameter structure of the [Convolution 1x7](#) filter.

```
#include <conv1x7.h>
```

Data Fields

- **UInt16 *** cMat

7 element array with fp16 values containing the 1x7 convolution matrix.

6.35.1 Detailed Description

Parameter structure of the [Convolution 1x7](#) filter.

6.35.2 Field Documentation

UInt16* Conv1x7Param::cMat

7 element array with fp16 values containing the 1x7 convolution matrix.

Default value:

Size of the array behind:

7

Referenced by svuConv1x7().

6.36 Conv1x9Param Struct Reference

Parameter structure of the [Convolution 1x9](#) filter.

```
#include <conv1x9.h>
```

Data Fields

- **UInt16 * cMat**

9 element array with fp16 values containing the 1x9 convolution matrix.

6.36.1 Detailed Description

Parameter structure of the **Convolution 1x9** filter.

6.36.2 Field Documentation

UInt16* Conv1x9Param::cMat

9 element array with fp16 values containing the 1x9 convolution matrix.

Default value:

Size of the array behind:

9

Referenced by svuConv1x9().

6.37 Conv3x3Fp16ToFp16Param Struct Reference

Parameter structure of the **Convolution 3x3 Fp16ToFp16** filter.

```
#include <conv3x3Fp16ToFp16.h>
```

Data Fields

- **UInt16 * cMat**

9 element array with fp16 values containing the 3x3 convolution matrix.

6.37.1 Detailed Description

Parameter structure of the **Convolution 3x3 Fp16ToFp16** filter.

6.37.2 Field Documentation

UInt16* Conv3x3Fp16ToFp16Param::cMat

9 element array with fp16 values containing the 3x3 convolution matrix.

Default value:

0x2C00,0x3000,0x2C00,0x3000,0x3400,0x3000,0x2C00,0x3000,0x2C00

Size of the array behind:

9

Referenced by `svuConv3x3Fp16ToFp16()`.

6.38 Conv3x3Param Struct Reference

Parameter structure of the [Convolution 3x3](#) filter.

```
#include <conv3x3.h>
```

Data Fields

- [UInt16](#) * `cMat`
9 element array with fp16 values containing the 3x3 convolution matrix.

6.38.1 Detailed Description

Parameter structure of the [Convolution 3x3](#) filter.

6.38.2 Field Documentation

[UInt16](#)* `Conv3x3Param::cMat`

9 element array with fp16 values containing the 3x3 convolution matrix.

Default value:

```
0x2C00,0x3000,0x2C00,0x3000,0x3400,0x3000,0x2C00,0x3000,0x2C00
```

Size of the array behind:

9

Referenced by `svuConv3x3()`.

6.39 Conv5x1Fp16ToFp16Param Struct Reference

Parameter structure of the [Convolution 5x1 Fp16ToFp16](#) filter.

```
#include <conv5x1Fp16ToFp16.h>
```

Data Fields

- [UInt16](#) * `cMat`
5 element array with fp16 values containing the 5x1 convolution matrix.

6.39.1 Detailed Description

Parameter structure of the [Convolution 5x1 Fp16ToFp16](#) filter.

6.39.2 Field Documentation

UInt16* Conv5x1Fp16ToFp16Param::cMat

5 element array with fp16 values containing the 5x1 convolution matrix.

Default value:

Size of the array behind:

5

Referenced by `svuConv5x1Fp16ToFp16()`.

6.40 Conv5x1Param Struct Reference

Parameter structure of the [Convolution 5x1](#) filter.

```
#include <conv5x1.h>
```

Data Fields

- **UInt16 *** cMat

5 element array with fp16 values containing the 5x1 convolution matrix.

6.40.1 Detailed Description

Parameter structure of the [Convolution 5x1](#) filter.

6.40.2 Field Documentation

UInt16* Conv5x1Param::cMat

5 element array with fp16 values containing the 5x1 convolution matrix.

Default value:

Size of the array behind:

5

Referenced by `svuConv5x1()`.

6.41 Conv5x5Fp16ToFp16Param Struct Reference

Parameter structure of the [Convolution 5x5 Fp16ToFp16](#) filter.

```
#include <conv5x5Fp16ToFp16.h>
```

Data Fields

- [UInt16](#) * [cMat](#)
25 element array with fp16 values containing the 5x5 convolution matrix.

6.41.1 Detailed Description

Parameter structure of the [Convolution 5x5 Fp16ToFp16](#) filter.

6.41.2 Field Documentation

[UInt16](#)* [Conv5x5Fp16ToFp16Param::cMat](#)

25 element array with fp16 values containing the 5x5 convolution matrix.

Default value:

Size of the array behind:

25

Referenced by [svuConv5x5Fp16ToFp16\(\)](#).

6.42 Conv5x5Param Struct Reference

Parameter structure of the [Convolution 5x5](#) filter.

```
#include <conv5x5.h>
```

Data Fields

- [UInt16](#) * [cMat](#)
25 element array with fp16 values containing the 5x5 convolution matrix.

6.42.1 Detailed Description

Parameter structure of the [Convolution 5x5](#) filter.

6.42.2 Field Documentation

UInt16* Conv5x5Param::cMat

25 element array with fp16 values containing the 5x5 convolution matrix.

Default value:

Size of the array behind:

25

Referenced by svuConv5x5().

6.43 Conv7x1Fp16ToFp16Param Struct Reference

Parameter structure of the [Convolution 7x1 Fp16ToFp16](#) filter.

```
#include <conv7x1Fp16ToFp16.h>
```

Data Fields

- **UInt16 *** cMat
7 element array with fp16 values containing the 7x1 convolution matrix.

6.43.1 Detailed Description

Parameter structure of the [Convolution 7x1 Fp16ToFp16](#) filter.

6.43.2 Field Documentation

UInt16* Conv7x1Fp16ToFp16Param::cMat

7 element array with fp16 values containing the 7x1 convolution matrix.

Default value:

Size of the array behind:

7

Referenced by svuConv7x1Fp16ToFp16().

6.44 Conv7x1Param Struct Reference

Parameter structure of the [Convolution 7x1](#) filter.

```
#include <conv7x1.h>
```

Data Fields

- **UInt16 * cMat**

7 element array with fp16 values containing the 7x1 convolution matrix.

6.44.1 Detailed Description

Parameter structure of the **Convolution 7x1** filter.

6.44.2 Field Documentation

UInt16* Conv7x1Param::cMat

7 element array with fp16 values containing the 7x1 convolution matrix.

Default value:

Size of the array behind:

7

Referenced by `svuConv7x1()`.

6.45 Conv7x7Param Struct Reference

Parameter structure of the **Convolution 7x7** filter.

```
#include <conv7x7.h>
```

Data Fields

- **UInt16 * cMat**

49 element array with fp16 values containing the 7x7 convolution matrix.

6.45.1 Detailed Description

Parameter structure of the **Convolution 7x7** filter.

6.45.2 Field Documentation

UInt16* Conv7x7Param::cMat

49 element array with fp16 values containing the 7x7 convolution matrix.

Default value:

```
0x2532, 0x2532, 0x2532, 0x2532, 0x2532, 0x2532, 0x2532, 0x2532, 0x2532, 0x2532, 0x2532,
0x2532, 0x2532, 0x2532, 0x2532, 0x2532, 0x2532, 0x2532, 0x2532, 0x2532, 0x2532, 0x2532,
```

0x2532, 0x2532, 0x2532, 0x2532, 0x2532, 0x2532, 0x2532, 0x2532, 0x2532, 0x2532, 0x2532,
 0x2532, 0x2532, 0x2532, 0x2532, 0x2532, 0x2532, 0x2532, 0x2532, 0x2532, 0x2532, 0x2532,
 0x2532, 0x2532, 0x2532, 0x2532, 0x2532

Size of the array behind:

49

Referenced by `svuConv7x7()`.

6.46 Conv7x7ParamFp16ToFp16 Struct Reference

Parameter structure of the [Convolution 7x7 Fp16ToFp16](#) filter.

```
#include <conv7x7Fp16ToFp16.h>
```

Data Fields

- **UInt16 * cMat**
49 element array with fp16 values containing the 7x7 convolution matrix.

6.46.1 Detailed Description

Parameter structure of the [Convolution 7x7 Fp16ToFp16](#) filter.

6.46.2 Field Documentation

UInt16* `Conv7x7ParamFp16ToFp16::cMat`

49 element array with fp16 values containing the 7x7 convolution matrix.

Default value:

Size of the array behind:

49

Referenced by `svuConv7x7Fp16ToFp16()`.

6.47 Conv7x7ParamFp16ToU8 Struct Reference

Parameter structure of the [Convolution 7x7 Fp16ToU8](#) filter.

```
#include <conv7x7Fp16ToU8.h>
```

Data Fields

- **UInt16 * cMat**

49 element array with fp16 values containing the 7x7 convolution matrix.

6.47.1 Detailed Description

Parameter structure of the **Convolution 7x7 Fp16ToU8** filter.

6.47.2 Field Documentation

UInt16* Conv7x7ParamFp16ToU8::cMat

49 element array with fp16 values containing the 7x7 convolution matrix.

Default value:

Size of the array behind:

49

Referenced by svuConv7x7Fp16ToU8().

6.48 Conv9x1Param Struct Reference

Parameter structure of the **Convolution 9x1** filter.

```
#include <conv9x1.h>
```

Data Fields

- **UInt16 * cMat**

9 element array with fp16 values containing the 9x1 convolution matrix.

6.48.1 Detailed Description

Parameter structure of the **Convolution 9x1** filter.

6.48.2 Field Documentation

UInt16* Conv9x1Param::cMat

9 element array with fp16 values containing the 9x1 convolution matrix.

Default value:

Size of the array behind:

9

Referenced by `svuConv9x1()`.

6.49 Conv9x9Fp16ToFp16Param Struct Reference

Parameter structure of the [Convolution 9x9 Fp16ToFp16](#) filter.

```
#include <conv9x9Fp16ToFp16.h>
```

Data Fields

- [UInt16](#) * `cMat`
81 element array with fp16 values containing the 9x9 convolution matrix.

6.49.1 Detailed Description

Parameter structure of the [Convolution 9x9 Fp16ToFp16](#) filter.

6.49.2 Field Documentation

[UInt16](#)* `Conv9x9Fp16ToFp16Param::cMat`

81 element array with fp16 values containing the 9x9 convolution matrix.

Default value:

Size of the array behind:

81

Referenced by `svuConv9x9Fp16ToFp16()`.

6.50 Conv9x9Param Struct Reference

Parameter structure of the [Convolution 9x9](#) filter.

```
#include <conv9x9.h>
```

Data Fields

- [UInt16](#) * `cMat`
81 element array with fp16 values containing the 9x9 convolution matrix.

6.50.1 Detailed Description

Parameter structure of the [Convolution 9x9](#) filter.

6.50.2 Field Documentation

UInt16* Conv9x9Param::cMat

81 element array with fp16 values containing the 9x9 convolution matrix.

Default value:

```
0x2252, 0x2252, 0x2252, 0x2252, 0x2252, 0x2252, 0x2252, 0x2252, 0x2252, 0x2252, 0x2252,
0x2252, 0x2252, 0x2252, 0x2252, 0x2252, 0x2252, 0x2252, 0x2252, 0x2252, 0x2252, 0x2252,
0x2252, 0x2252, 0x2252, 0x2252, 0x2252, 0x2252, 0x2252, 0x2252, 0x2252, 0x2252, 0x2252,
0x2252, 0x2252, 0x2252, 0x2252, 0x2252, 0x2252, 0x2252, 0x2252, 0x2252, 0x2252, 0x2252,
0x2252, 0x2252, 0x2252, 0x2252, 0x2252, 0x2252, 0x2252, 0x2252, 0x2252, 0x2252, 0x2252,
0x2252, 0x2252, 0x2252, 0x2252, 0x2252, 0x2252, 0x2252, 0x2252, 0x2252, 0x2252, 0x2252,
0x2252, 0x2252, 0x2252, 0x2252, 0x2252, 0x2252, 0x2252, 0x2252, 0x2252, 0x2252, 0x2252,
0x2252, 0x2252, 0x2252, 0x2252
```

Size of the array behind:

81

Referenced by `svuConv9x9()`.

6.51 ConvGenericParam Struct Reference

Parameter structure of the [Generic Convolution](#) filter.

```
#include <convGeneric.h>
```

Data Fields

- **UInt16 * cMat**
Array with fp16 values containing the convolution matrix.
- **UInt32 filterSize**
u32 Kernel size.

6.51.1 Detailed Description

Parameter structure of the [Generic Convolution](#) filter.

6.51.2 Field Documentation

UInt16* ConvGenericParam::cMat

Array with fp16 values containing the convolution matrix.

Referenced by `svuConvGeneric()`.

UInt32 ConvGenericParam::filterSize

u32 Kernel size.

Referenced by svuConvGeneric().

6.52 ConvParam Struct Reference

Parameter structure of the convolution filter.

```
#include <sippHwDefs_ma2x5x.h>
```

Data Fields

- **UInt32** frmDim
Reserved field yes.
- **UInt32** cfg
configuration bit field(see SIPP_CONV_CFG_ADR)
- **UInt32** kernel [15]
*Convolution kernel coefficients(see SIPP_CONV_COEFF_*_ADR (fp16 values))*
- **UInt32** shadowKernel [15]
*Convolution kernel coefficients(see SIPP_CONV_COEFF_*_ADR (fp16 values))*

6.52.1 Detailed Description

Parameter structure of the convolution filter.

6.52.2 Field Documentation

UInt32 ConvParam::cfg

configuration bit field(see SIPP_CONV_CFG_ADR)

UInt32 ConvParam::frmDim

Reserved field yes.

UInt32 ConvParam::kernel[15]

Convolution kernel coefficients(see SIPP_CONV_COEFF_*_ADR (fp16 values))

Referenced by packConv3x3CCM(), and packConv5x5CCM().

UInt32 ConvParam::shadowKernel[15]

Convolution kernel coefficients(see SIPP_CONV_COEFF_*_ADR (fp16 values))

6.53 ConvSeparable11x11Fp16ToFp16Param Struct Reference

Parameter structure of the [Convolution Separable 11x11 Fp16ToFp16](#) filter.

```
#include <convSeparable11x11Fp16ToFp16.h>
```

Data Fields

- [UInt16](#) * [cMat](#)

Array with 6 fp32 values that represent the separable values for 11x11 matrix.

6.53.1 Detailed Description

Parameter structure of the [Convolution Separable 11x11 Fp16ToFp16](#) filter.

6.53.2 Field Documentation

[UInt16](#)* [ConvSeparable11x11Fp16ToFp16Param::cMat](#)

Array with 6 fp32 values that represent the separable values for 11x11 matrix.

Size of the array behind:

6

Referenced by [svuConvSeparable11x11Fp16ToFp16\(\)](#).

6.54 ConvSeparable11x11Param Struct Reference

Parameter structure of the [Convolution Separable 11x11](#) filter.

```
#include <convSeparable11x11.h>
```

Data Fields

- [UInt32](#) * [cMat](#)

Array with 6 fp32 values that represent the separable values for 11x11 matrix.

6.54.1 Detailed Description

Parameter structure of the [Convolution Separable 11x11](#) filter.

6.54.2 Field Documentation

[UInt32](#)* [ConvSeparable11x11Param::cMat](#)

Array with 6 fp32 values that represent the separable values for 11x11 matrix.

Size of the array behind:

6

Referenced by `svuConvSeparable11x11()`.

6.55 ConvSeparable3x3Fp16ToFp16Param Struct Reference

Parameter structure of the [Convolution Separable 3x3 Fp16ToFp16](#) filter.

```
#include <convSeparable3x3Fp16ToFp16.h>
```

Data Fields

- [UInt16 cMat](#) [2]
Array with 2 fp32 values that represent the separable values for 3x3 matrix.

6.55.1 Detailed Description

Parameter structure of the [Convolution Separable 3x3 Fp16ToFp16](#) filter.

6.55.2 Field Documentation

[UInt16](#) `ConvSeparable3x3Fp16ToFp16Param::cMat`[2]

Array with 2 fp32 values that represent the separable values for 3x3 matrix.

Referenced by `svuConvSeparable3x3Fp16ToFp16()`.

6.56 ConvSeparable3x3Param Struct Reference

Parameter structure of the [Convolution Separable 3x3](#) filter.

```
#include <convSeparable3x3.h>
```

Data Fields

- float [cMat](#) [2]
Array with 2 fp32 values that represent the separable values for 3x3 matrix.

6.56.1 Detailed Description

Parameter structure of the [Convolution Separable 3x3](#) filter.

6.56.2 Field Documentation

`float ConvSeparable3x3Param::cMat[2]`

Array with 2 fp32 values that represent the separable values for 3x3 matrix.

Referenced by `svuConvSeparable3x3()`.

6.57 ConvSeparable5x5Fp16ToFp16Param Struct Reference

Parameter structure of the [Convolution Separable 5x5 Fp16ToFp16](#) filter.

```
#include <convSeparable5x5Fp16ToFp16.h>
```

Data Fields

- `UInt16 * cMat`

Array with 3 fp32 values that represent the separable values for 5x5 matrix.

6.57.1 Detailed Description

Parameter structure of the [Convolution Separable 5x5 Fp16ToFp16](#) filter.

6.57.2 Field Documentation

`UInt16* ConvSeparable5x5Fp16ToFp16Param::cMat`

Array with 3 fp32 values that represent the separable values for 5x5 matrix.

Size of the array behind:

3

Referenced by `svuConvSeparable5x5Fp16ToFp16()`.

6.58 ConvSeparable5x5Param Struct Reference

Parameter structure of the [Convolution Separable 5x5](#) filter.

```
#include <convSeparable5x5.h>
```

Data Fields

- `float cMat [3]`

Array with 3 fp32 values that represent the separable values for 5x5 matrix.

6.58.1 Detailed Description

Parameter structure of the [Convolution Separable 5x5](#) filter.

6.58.2 Field Documentation

`float ConvSeparable5x5Param::cMat[3]`

Array with 3 fp32 values that represent the separable values for 5x5 matrix.

Referenced by `svuConvSeparable5x5()`.

6.59 ConvSeparable7x7Fp16ToFp16Param Struct Reference

Parameter structure of the [Convolution Separable 7x7 Fp16ToFp16](#) filter.

```
#include <convSeparable7x7Fp16ToFp16.h>
```

Data Fields

- `UInt16 * cMat`

Array with 4 fp32 values that represent the separable values for 7x7 matrix.

6.59.1 Detailed Description

Parameter structure of the [Convolution Separable 7x7 Fp16ToFp16](#) filter.

6.59.2 Field Documentation

`UInt16* ConvSeparable7x7Fp16ToFp16Param::cMat`

Array with 4 fp32 values that represent the separable values for 7x7 matrix.

Size of the array behind:

4

Referenced by `svuConvSeparable7x7Fp16ToFp16()`.

6.60 ConvSeparable7x7Param Struct Reference

Parameter structure of the [Convolution Separable 7x7](#) filter.

```
#include <convSeparable7x7.h>
```

Data Fields

- `float cMat [4]`

Array with 4 fp32 values that represent the separable values for 7x7 matrix.

6.60.1 Detailed Description

Parameter structure of the [Convolution Separable 7x7](#) filter.

6.60.2 Field Documentation

`float ConvSeparable7x7Param::cMat[4]`

Array with 4 fp32 values that represent the separable values for 7x7 matrix.

Referenced by `svuConvSeparable7x7()`.

6.61 ConvSeparable9x9Fp16ToFp16Param Struct Reference

Parameter structure of the [Convolution Separable 9x9 Fp16ToFp16](#) filter.

```
#include <convSeparable9x9Fp16ToFp16.h>
```

Data Fields

- `UInt16 * cMat`

Array with 5 fp32 values that represent the separable values for 9x9 matrix.

6.61.1 Detailed Description

Parameter structure of the [Convolution Separable 9x9 Fp16ToFp16](#) filter.

6.61.2 Field Documentation

`UInt16* ConvSeparable9x9Fp16ToFp16Param::cMat`

Array with 5 fp32 values that represent the separable values for 9x9 matrix.

Size of the array behind:

5

Referenced by `svuConvSeparable9x9Fp16ToFp16()`.

6.62 ConvSeparable9x9Param Struct Reference

Parameter structure of the [Convolution Separable 9x9](#) filter.

```
#include <convSeparable9x9.h>
```

Data Fields

- `float cMat [5]`

Array with 5 fp32 values that represent the separable values for 9x9 matrix.

6.62.1 Detailed Description

Parameter structure of the [Convolution Separable 9x9](#) filter.

6.62.2 Field Documentation

`float ConvSeparable9x9Param::cMat[5]`

Array with 5 fp32 values that represent the separable values for 9x9 matrix.

Referenced by `svuConvSeparable9x9()`.

6.63 CropParam Struct Reference

Parameter structure of the [Crop](#) filter.

```
#include <crop.h>
```

Data Fields

- [UInt32 st_Y](#)
Starting point on vertical crop.

6.63.1 Detailed Description

Parameter structure of the [Crop](#) filter.

6.63.2 Field Documentation

[UInt32](#) `CropParam::st_Y`

Starting point on vertical crop.

6.64 cvtColorChromaNV12Param Struct Reference

Parameter structure of the [RGB to Chroma NV12 conversion](#) filter.

```
#include <cvtColorRGBtoChromaNV12.h>
```

Data Fields

- `float * coefsMat`
3x3 conversion matrix
- `float * offset`
3 offset parameters

6.64.1 Detailed Description

Parameter structure of the [RGB to Chroma NV12 conversion](#) filter.

6.64.2 Field Documentation

`float* cvtColorChromaNV12Param::coefsMat`

3x3 conversion matrix

Referenced by `svuCvtColorRGBtoChromaNV12()`.

`float* cvtColorChromaNV12Param::offset`

3 offset parameters

Referenced by `svuCvtColorRGBtoChromaNV12()`.

6.65 CvtColorChromaYUVToNV12Param Struct Reference

Parameter structure of the [YUV to NV12 chroma conversion](#) filter.

```
#include <cvtColorChromaYUVToNV12.h>
```

Data Fields

- frameType [inputFrameType](#)
input frame format, supported: yuv420p, yuv444p
- [UInt8 needs2Parents](#)

6.65.1 Detailed Description

Parameter structure of the [YUV to NV12 chroma conversion](#) filter.

6.65.2 Field Documentation

`frameType CvtColorChromaYUVToNV12Param::inputFrameType`

input frame format, supported: yuv420p, yuv444p

Referenced by `svuCvtColorChromaYUVToNV12()`.

[UInt8](#) `CvtColorChromaYUVToNV12Param::needs2Parents`

Referenced by `svuCvtColorChromaYUVToNV12()`.

6.66 cvtColorLumaNV12Param Struct Reference

Parameter structure of the [RGB to Luma NV12 conversion](#) filter.

```
#include <cvtColorRGBtoLumaNV12.h>
```

Data Fields

- float * [coefsMat](#)
3x3 conversion matrix
- float * [offset](#)
3 offset parameters

6.66.1 Detailed Description

Parameter structure of the [RGB to Luma NV12 conversion](#) filter.

6.66.2 Field Documentation

[float* cvtColorLumaNV12Param::coefsMat](#)

3x3 conversion matrix

Referenced by `svuCvtColorRGBtoLumaNV12()`.

[float* cvtColorLumaNV12Param::offset](#)

3 offset parameters

Referenced by `svuCvtColorRGBtoLumaNV12()`.

6.67 DbYrParam Struct Reference

Parameter structure of the debayer filter.

```
#include <sippHwDefs_ma2x5x.h>
```

Data Fields

- [UInt32 frmDim](#)
Reserved field yes.
- [UInt32 cfg](#)
configuration bit field(see `SIPP_DBYR_CFG_ADR`)
- [UInt32 thresh](#)
thresholds (see `SIPP_DBYR_THRES_ADR`)
- [UInt32 dewormCfg](#)
De-worming bit field(see `SIPP_DBYR_DEWORM_ADR`)
- [UInt32 lumaWeight](#)
De-worming bit field(see `SIPP_DBYR_DEWORM_ADR`)

6.67.1 Detailed Description

Parameter structure of the debayer filter.

6.67.2 Field Documentation

UInt32 DbyrParam::cfg

configuration bit field(see SIPP_DBYR_CFG_ADR)

UInt32 DbyrParam::dewormCfg

De-worming bit field(see SIPP_DBYR_DEWORM_ADR)

UInt32 DbyrParam::frmDim

Reserved field yes.

UInt32 DbyrParam::lumaWeight

De-worming bit field(see SIPP_DBYR_DEWORM_ADR)

UInt32 DbyrParam::thresh

thresholds (see SIPP_DBYR_THRES_ADR)

Referenced by sippBufGetObufCtx().

6.68 Dilate3x3Param Struct Reference

Parameter structure of the [Dilate 3x3](#) filter.

```
#include <dilate3x3.h>
```

Data Fields

- [UInt8](#) * [dMat](#) [3]
3x3 dilate matrix.

6.68.1 Detailed Description

Parameter structure of the [Dilate 3x3](#) filter.

6.68.2 Field Documentation

UInt8* Dilate3x3Param::dMat[3]

3x3 dilate matrix.

Referenced by svuDilate3x3().

6.69 Dilate5x5Param Struct Reference

Parameter structure of the [Dilate 5x5](#) filter.

```
#include <dilate5x5.h>
```

Data Fields

- [UInt8](#) * [dMat](#) [5]
5x5 dilate matrix.

6.69.1 Detailed Description

Parameter structure of the [Dilate 5x5](#) filter.

6.69.2 Field Documentation

[UInt8](#)* [Dilate5x5Param::dMat](#)[5]

5x5 dilate matrix.

Referenced by [svuDilate5x5\(\)](#).

6.70 Dilate7x7Param Struct Reference

Parameter structure of the [Dilate 7x7](#) filter.

```
#include <dilate7x7.h>
```

Data Fields

- [UInt8](#) * [dMat](#) [7]
7x7 dilate matrix.

6.70.1 Detailed Description

Parameter structure of the [Dilate 7x7](#) filter.

6.70.2 Field Documentation

[UInt8](#)* [Dilate7x7Param::dMat](#)[7]

7x7 dilate matrix.

Referenced by [svuDilate7x7\(\)](#).

6.71 DilateGenericParam Struct Reference

Parameter structure of the [Generic Dilate](#) filter.

```
#include <dilateGeneric.h>
```

Data Fields

- [UInt32 * dMat](#)
Array with fp16 values containing the dilate matrix.
- [UInt32 kernelSize](#)
u32 Kernel size.

6.71.1 Detailed Description

Parameter structure of the [Generic Dilate](#) filter.

6.71.2 Field Documentation

[UInt32* DilateGenericParam::dMat](#)

Array with fp16 values containing the dilate matrix.

Referenced by `svuDilateGeneric()`.

[UInt32 DilateGenericParam::kernelSize](#)

u32 Kernel size.

Referenced by `svuDilateGeneric()`.

6.72 Disp2depthParam Struct Reference

Parameter structure of the [Threshold](#) filter.

```
#include <disp2depth.h>
```

Public Member Functions

- half disp2depthLUT[3040] [ALIGNED](#) (16)

Data Fields

- [bool flip_disp2depth](#)
- [bool _16bitmode](#)

6.72.1 Detailed Description

Parameter structure of the [Threshold](#) filter.

6.72.2 Member Function Documentation

half disp2depthLUT [3040] Disp2depthParam::ALIGNED (16)

6.72.3 Field Documentation

bool Disp2depthParam::_16bitmode

Referenced by svudisp2depth().

bool Disp2depthParam::flip_disp2depth

Referenced by svudisp2depth().

6.73 DmaDesc Struct Reference

```
#include <sippTypes.h>
```

Data Fields

- [u64 dscCtrlLinkAddr](#)
- [u64 dscDstSrcAddr](#)
- [u64 dscPlanesLen](#)
- [u64 dscSrcStrdWidth](#)
- [u64 dscDstStrdWidth](#)
- [u64 dscPlStrides](#)

6.73.1 Field Documentation

u64 DmaDesc::dscCtrlLinkAddr

Referenced by sippChainDmaDesc(), sippCoreFinalisePipeline(), and sippInitDma().

u64 DmaDesc::dscDstSrcAddr

Referenced by sippCoreFinalisePipeline(), sippUpdateDmaAddr(), and sippUpdateDmaAddrCQ().

u64 DmaDesc::dscDstStrdWidth

Referenced by sippCoreFinalisePipeline(), and sippInitDma().

u64 DmaDesc::dscPlanesLen

Referenced by sippCoreFinalisePipeline(), and sippInitDma().

u64 DmaDesc::dscPlStrides

Referenced by `sippCoreFinalisePipeline()`, and `sippInitDma()`.

u64 DmaDesc::dscSrcStrdWidth

Referenced by `sippCoreFinalisePipeline()`, and `sippInitDma()`.

6.74 DMAExtCfg Union Reference

```
#include <sippTypes.h>
```

Data Fields

- **u32 DMACfgPlaceholder**
- **DMAPartialCfg tPartialCfg**

6.74.1 Field Documentation

u32 DMAExtCfg::DMACfgPlaceholder

DMAPartialCfg DMAExtCfg::tPartialCfg

Referenced by `sippInitDma()`.

6.75 DmaParam Struct Reference

Parameter structure of the **DMA** filter.

```
#include <sippHwDefs.h>
```

Data Fields

- **DmaDesc dmaDsc**
Private member. Myriad2 DMA 2D-chunked descriptor, as in CMXDMA_controller.doc.
- **UInt32 ddrAddr**
User level params to customize transfer.
- **UInt32 dstPlS**
Private member.
- **UInt32 dstLnS**
Private member. Full line strides.
- **UInt32 srcLnS**
Private member. Full line strides.
- **DmaDesc * pLineDesList**
Private member. Myriad2 DMA 2D-chunked descriptor pointer used when multiple lines per iteration, as in CMXDMA_controller.doc.

- **eDmaMode dmaMode**
Allow dma filter to be configured for standard and smart modes.
- **DMAExtCfg extCfg**
dmaMode specific config information
- **bool bChunked**
bChunked specifies if buffer is chunked or not

6.75.1 Detailed Description

Parameter structure of the **DMA** filter.

6.75.2 Field Documentation

bool DmaParam::bChunked

bChunked specifies if buffer is chunked or not

UInt32 DmaParam::ddrAddr

User level params to customize transfer.

Referenced by sippUpdateDmaAddr(), and sippUpdateDmaAddrCQ().

DmaDesc DmaParam::dmaDsc

Private member. Myriad2 DMA 2D-chunked descriptor, as in CMXDMA_controller.doc.

Referenced by sippChainDmaDesc(), sippInitDma(), sippUpdateDmaAddr(), and sippUpdateDmaAddrCQ().

eDmaMode DmaParam::dmaMode

Allow dma filter to be configured for standard and smart modes.

Referenced by sippInitDma().

UInt32 DmaParam::dstLnS

Private member. Full line strides.

Referenced by sippInitDma(), sippUpdateDmaAddr(), and sippUpdateDmaAddrCQ().

UInt32 DmaParam::dstPls

Private member.

Referenced by sippInitDma().

DMAExtCfg DmaParam::extCfg

dmaMode specific config information

Referenced by sippInitDma().

DmaDesc* DmaParam::pLineDesList

Private member. Myriad2 DMA 2D-chunked descriptor pointer used when multiple lines per iteration, as in CMXDMA_controller.doc.

Referenced by sippChainDmaDesc(), sippInitDma(), sippUpdateDmaAddr(), and sippUpdateDmaAddrCQ().

UInt32 DmaParam::srcLnS

Private member. Full line strides.

Referenced by sippInitDma(), sippUpdateDmaAddr(), and sippUpdateDmaAddrCQ().

6.76 DMAPartialCfg Union Reference

```
#include <sippTypes.h>
```

Data Fields

- **u32** ddrLineStride

6.76.1 Field Documentation

u32 DMAPartialCfg::ddrLineStride

Referenced by sippInitDma().

6.77 DmaTaskList Struct Reference

Data Fields

- **SippPipeline** * taskPl [DMA_TASK_LIST_SZ]
- **u32** wPtr
- **u32** rPtr

6.77.1 Field Documentation

u32 DmaTaskList::rPtr

Referenced by sippCmxDmaDoneIrqHandler().

SippPipeline* DmaTaskList::taskPl[DMA_TASK_LIST_SZ]

Referenced by dmaKickSequenceConcurrent(), and sippCmxDmaDoneIrqHandler().

u32 DmaTaskList::wPtr

Referenced by dmaKickSequenceConcurrent().

6.78 DogLtmParam Struct Reference

```
#include <sippHwDefs_ma2x5x.h>
```

Data Fields

- **UInt32** frmDim
Reserved field yes.
- **UInt32** cfg
configuration bitfield
- **UInt8** * dogCoeffs11
- **UInt8** * dogCoeffs15
- **UInt8** dogStrength
- **UInt16** * ltmCurves

6.78.1 Field Documentation

UInt32 DogLtmParam::cfg

configuration bitfield

UInt8* DogLtmParam::dogCoeffs11

UInt8* DogLtmParam::dogCoeffs15

UInt8 DogLtmParam::dogStrength

UInt32 DogLtmParam::frmDim

Reserved field yes.

UInt16* DogLtmParam::ltmCurves

6.79 EdgeParam Struct Reference

Parameter structure of the edgeoperator filter.

```
#include <sippHwDefs_ma2x5x.h>
```

Data Fields

- **UInt32 frmDim**
Reserved field yes.
- **UInt32 cfg**
configuration bitfield(see SIPP_EDGE_OP_CFG_ADR)
- **UInt32 xCoeff**
Edge operator X coefficients(see SIPP_EDGE_OP_XCOEFF_ADR)
- **UInt32 yCoeff**
Edge operator Y coefficients(see SIPP_EDGE_OP_YCOEFF_ADR)

6.79.1 Detailed Description

Parameter structure of the edgeoperator filter.

6.79.2 Field Documentation

UInt32 EdgeParam::cfg

configuration bitfield(see SIPP_EDGE_OP_CFG_ADR)

UInt32 EdgeParam::frmDim

Reserved field yes.

UInt32 EdgeParam::xCoeff

Edge operator X coefficients(see SIPP_EDGE_OP_XCOEFF_ADR)

UInt32 EdgeParam::yCoeff

Edge operator Y coefficients(see SIPP_EDGE_OP_YCOEFF_ADR)

6.80 EqualizeHistParam Struct Reference

Parameter structure of the [Equalize Histogram](#) filter.

```
#include <equalizeHist.h>
```

Data Fields

- **UInt32 * cum_hist**
pointer to an input array that indicates the cumulative histogram of the image

6.80.1 Detailed Description

Parameter structure of the [Equalize Histogram](#) filter.

6.80.2 Field Documentation

UInt32* EqualizeHistParam::cum_hist

pointer to an input array that indicates the cumulative histogram of the image
Referenced by svuEqualizeHist().

6.81 Erode3x3Param Struct Reference

Parameter structure of the **Erode 3x3** filter.

```
#include <erode3x3.h>
```

Data Fields

- **UInt8** * **eMat** [3]
3x3 erode matrix.

6.81.1 Detailed Description

Parameter structure of the **Erode 3x3** filter.

6.81.2 Field Documentation

UInt8* Erode3x3Param::eMat[3]

3x3 erode matrix.

Referenced by svuErode3x3().

6.82 Erode5x5Param Struct Reference

Parameter structure of the **Erode 5x5** filter.

```
#include <erode5x5.h>
```

Data Fields

- **UInt8** * **eMat** [5]
5x5 erode matrix.

6.82.1 Detailed Description

Parameter structure of the **Erode 5x5** filter.

6.82.2 Field Documentation

UInt8* Erode5x5Param::eMat[5]

5x5 erode matrix.

Referenced by svuErode5x5().

6.83 Erode7x7Param Struct Reference

Parameter structure of the [Erode 7x7](#) filter.

```
#include <erode7x7.h>
```

Data Fields

- **UInt8 * eMat** [7]
7x7 erode matrix.

6.83.1 Detailed Description

Parameter structure of the [Erode 7x7](#) filter.

6.83.2 Field Documentation

UInt8* Erode7x7Param::eMat[7]

7x7 erode matrix.

Referenced by svuErode7x7().

6.84 ExtStatsSatPixelsU32Param Struct Reference

```
#include <extStatsSatPixelsU32.h>
```

Data Fields

- **UInt32 mask**
 - **UInt32 runNr**
 - struct {
 - UInt32 base**
 - UInt32 cfg**
 - UInt32 ls**
 - UInt32 ps**
 - UInt32 irqRate**
 - UInt32 fillCtrl**
 - UInt32 ctx**
- } in

- `UInt8 * outStatsBuffer`
- `UInt8 * outStatsBufferOutside`
- `UInt32 crtPaxel`
- `UInt32 nextVerticalStartPos`
- `UInt32 crtPosInPaxel`
- `UInt32 firstPatchX`
- `UInt32 firstPatchY`
- `UInt32 patchWidth`
- `UInt32 patchHeight`
- `UInt32 patchGapX`
- `UInt32 patchGapY`
- `UInt32 nPatchesX`
- `UInt32 nPatchesY`
- `UInt32 satThresh`
- `void(* statsSaturatePxl)(void *)`
- `UInt32 shaveNr`
- `UInt32 stackPointer`

6.84.1 Field Documentation

UInt32 ExtStatsSatPixelsU32Param::base

Referenced by `svuExtStatsSatPixelsU32()`.

UInt32 ExtStatsSatPixelsU32Param::cfg

Referenced by `svuExtStatsSatPixelsU32()`.

UInt32 ExtStatsSatPixelsU32Param::crtPaxel

UInt32 ExtStatsSatPixelsU32Param::crtPosInPaxel

UInt32 ExtStatsSatPixelsU32Param::ctx

UInt32 ExtStatsSatPixelsU32Param::fillCtrl

Referenced by `svuExtStatsSatPixelsU32()`.

UInt32 ExtStatsSatPixelsU32Param::firstPatchX

UInt32 ExtStatsSatPixelsU32Param::firstPatchY

`struct { ... } ExtStatsSatPixelsU32Param::in`

Referenced by `svuExtStatsSatPixelsU32()`.

UInt32 ExtStatsSatPixelsU32Param::irqRate

UInt32 ExtStatsSatPixelsU32Param::ls

Referenced by svuExtStatsSatPixelsU32().

UInt32 ExtStatsSatPixelsU32Param::mask

UInt32 ExtStatsSatPixelsU32Param::nextVerticalStartPos

UInt32 ExtStatsSatPixelsU32Param::nPatchesX

UInt32 ExtStatsSatPixelsU32Param::nPatchesY

UInt8* ExtStatsSatPixelsU32Param::outStatsBuffer

UInt8* ExtStatsSatPixelsU32Param::outStatsBufferOutside

UInt32 ExtStatsSatPixelsU32Param::patchGapX

UInt32 ExtStatsSatPixelsU32Param::patchGapY

UInt32 ExtStatsSatPixelsU32Param::patchHeight

UInt32 ExtStatsSatPixelsU32Param::patchWidth

UInt32 ExtStatsSatPixelsU32Param::ps

UInt32 ExtStatsSatPixelsU32Param::runNr

UInt32 ExtStatsSatPixelsU32Param::satThresh

UInt32 ExtStatsSatPixelsU32Param::shaveNr

Referenced by svuExtStatsSatPixelsU32().

UInt32 ExtStatsSatPixelsU32Param::stackPointer

Referenced by svuExtStatsSatPixelsU32().

void(* ExtStatsSatPixelsU32Param::statsSaturatePxl)(**void** *)

Referenced by svuExtStatsSatPixelsU32().

6.85 Fast9M2Param Struct Reference

Parameter structure of the [Fast9M2](#) filter.

```
#include <fast9M2.h>
```

Data Fields

- [UInt8 threshold](#)
threshold
- [UInt16 frameSliceWidth](#)
image width per slices used

6.85.1 Detailed Description

Parameter structure of the [Fast9M2](#) filter.

6.85.2 Field Documentation

[UInt16](#) Fast9M2Param::frameSliceWidth

image width per slices used

Referenced by `svuFast9M2()`.

[UInt8](#) Fast9M2Param::threshold

threshold

Referenced by `svuFast9M2()`.

6.86 Fast9ScoreCvParam Struct Reference

Parameter structure of the [Fast9ScoreCv](#) filter.

```
#include <fast9ScoreCv.h>
```

Data Fields

- [UInt8 threshold](#)
- [UInt16 frameSliceWidth](#)

6.86.1 Detailed Description

Parameter structure of the [Fast9ScoreCv](#) filter.

6.86.2 Field Documentation

[UInt16](#) Fast9ScoreCvParam::frameSliceWidth

Referenced by `svuFast9ScoreCv()`.

UInt8 Fast9ScoreCvParam::threshold

Referenced by svuFast9ScoreCv().

6.87 GenChrParam Struct Reference

Parameter structure of the edgeoperator filter.

```
#include <sippHwDefs_ma2x5x.h>
```

Data Fields

- **UInt32** frmDim
Reserved field yes.
- **UInt32** cfg
configuration bitfield
- **UInt32** yCoefs
???
- **UInt32** chrCoefs
???

6.87.1 Detailed Description

Parameter structure of the edgeoperator filter.

6.87.2 Field Documentation

UInt32 GenChrParam::cfg

configuration bitfield

Referenced by sippCheckOPipeConnectionGenChroma().

UInt32 GenChrParam::chrCoefs

???

UInt32 GenChrParam::frmDim

Reserved field yes.

UInt32 GenChrParam::yCoefs

???

6.88 GenLumaU8Fp16Param Struct Reference

Shave function of the [Generate Luma U8 to Fp16](#) filter.

```
#include <genLumaU8Fp16.h>
```

Data Fields

- [UInt16](#) * [coefs](#)
3 element array with fp16 values containing the 3 coefficients for genluma.

6.88.1 Detailed Description

Shave function of the [Generate Luma U8 to Fp16](#) filter.

6.88.2 Field Documentation

[UInt16](#)* [GenLumaU8Fp16Param::coefs](#)

3 element array with fp16 values containing the 3 coefficients for genluma.

Size of the array behind:

3

Referenced by [svuGenLumaU8Fp16\(\)](#).

6.89 GreyDesatParam Struct Reference

Parameter structure of the [greyDesat](#) filter.

```
#include <greyDesat.h>
```

Data Fields

- int [offset](#)
- int [slope](#)
- [Int32](#) [grey](#) [3]

6.89.1 Detailed Description

Parameter structure of the [greyDesat](#) filter.

6.89.2 Field Documentation

[Int32](#) [GreyDesatParam::grey](#)[3]

Referenced by [svuGreyDesat\(\)](#).

`int GreyDesatParam::offset`

Referenced by `svuGreyDesat()`.

`int GreyDesatParam::slope`

Referenced by `svuGreyDesat()`.

6.90 HammingDistanceParam Struct Reference

Parameter structure of the `hammingDistance` filter.

```
#include <hammingDistance.h>
```

Data Fields

- `int descriptor_size`

6.90.1 Detailed Description

Parameter structure of the `hammingDistance` filter.

6.90.2 Field Documentation

`int HammingDistanceParam::descriptor_size`

Referenced by `svuHammingDistance()`.

6.91 HarrisParam Struct Reference

Parameter structure of the `harriscorners` filter.

```
#include <sippHwDefs_ma2x5x.h>
```

Data Fields

- `UInt32 frmDim`
Reserved field yes.
- `UInt32 cfg`
configuration bit field(see `SIPP_HARRIS_CFG_ADR`)
- `float kValue`
value that changes the response of the edges(FP32)(see `SIPP_HARRIS_K_ADR`)

6.91.1 Detailed Description

Parameter structure of the `harriscorners` filter.

6.91.2 Field Documentation

UInt32 HarrisParam::cfg

configuration bit field(see SIPP_HARRIS_CFG_ADR)

UInt32 HarrisParam::frmDim

Reserved field yes.

float HarrisParam::kValue

value that changes the response of the edges(FP32)(see SIPP_HARRIS_K_ADR)

6.92 HarrisSwParam Struct Reference

Parameter structure of the [harrisResponse](#) filter.

```
#include <harrisResponse.h>
```

Data Fields

- float [k](#)

6.92.1 Detailed Description

Parameter structure of the [harrisResponse](#) filter.

6.92.2 Field Documentation

float HarrisSwParam::k

Referenced by `svuHarrisResponse()`.

6.93 HistogramParam Struct Reference

Parameter structure of the [Histogram](#) filter.

```
#include <histogram.h>
```

Data Fields

- **UInt32** * [hist](#)
array of values from histogram

6.93.1 Detailed Description

Parameter structure of the [Histogram](#) filter.

6.93.2 Field Documentation

UInt32* [HistogramParam::hist](#)

array of values from histogram

Referenced by [svuHistogram\(\)](#).

6.94 HistogramStatParam Struct Reference

Parameter structure of the [histogramStat](#) filter.

```
#include <histogramStat.h>
```

Data Fields

- **UInt32** [step](#)

6.94.1 Detailed Description

Parameter structure of the [histogramStat](#) filter.

6.94.2 Field Documentation

UInt32 [HistogramStatParam::step](#)

Referenced by [svuHistogramStat\(\)](#).

6.95 HomographyParam Struct Reference

Parameter structure of the [Homography](#) filter.

```
#include <homography.h>
```

Data Fields

- float * [homoMat3x3](#)
Pointer to float 3x3 Homography matrix

6.95.1 Detailed Description

Parameter structure of the [Homography](#) filter.

6.95.2 Field Documentation

`float* HomographyParam::homoMat3x3`

Pointer to float 3x3 Homography matrix

6.96 HorizPaddingOffS Struct Reference

```
#include <sippTypes.h>
```

Data Fields

- [s32 srcLeftO](#)
- [s32 srcRightO](#)
- [s32 dstLeftO](#)
- [s32 dstRightO](#)

6.96.1 Field Documentation

[s32 HorizPaddingOffS::dstLeftO](#)

Referenced by `sippComputePaddingOffsets()`, and `sippHorizontalPadding()`.

[s32 HorizPaddingOffS::dstRightO](#)

Referenced by `sippComputePaddingOffsets()`, and `sippHorizontalPadding()`.

[s32 HorizPaddingOffS::srcLeftO](#)

Referenced by `sippComputePaddingOffsets()`, and `sippHorizontalPadding()`.

[s32 HorizPaddingOffS::srcRightO](#)

Referenced by `sippComputePaddingOffsets()`, and `sippHorizontalPadding()`.

6.97 HPadInfoS Struct Reference

```
#include <sippTypes.h>
```

Data Fields

- [SippFilter * fptr](#)
- [u32 svuPadFunc](#)
- [HorizPaddingOff * padOffsets](#)
- [u32 CT1](#)
- [u32 oBufId](#)

6.97.1 Field Documentation

u32 HPadInfoS::CT1

Referenced by sippComputePaddingOffsets(), and sippHorizontalPadding().

SippFilter* HPadInfoS::fptr

Referenced by sippBuildLnBufs(), sippComputePaddingOffsets(), and sippHorizontalPadding().

u32 HPadInfoS::oBufId

Referenced by sippBuildLnBufs(), sippComputePaddingOffsets(), and sippHorizontalPadding().

HorizPaddingOff* HPadInfoS::padOffsets

Referenced by sippComputePaddingOffsets(), and sippHorizontalPadding().

u32 HPadInfoS::svuPadFunc

Referenced by sippBuildLnBufs().

6.98 InterpolatePixelBilinearParam Struct Reference

Parameter structure of the [interpolatePixelBilinear](#) filter.

```
#include <interpolatePixelBilinear.h>
```

Data Fields

- float [x](#)
- float [y](#)

6.98.1 Detailed Description

Parameter structure of the [interpolatePixelBilinear](#) filter.

6.98.2 Field Documentation

float InterpolatePixelBilinearParam::x

Referenced by svuInterpolatePixelBilinear().

float InterpolatePixelBilinearParam::y

Referenced by svuInterpolatePixelBilinear().

6.99 LocalTMParam Struct Reference

Parameter structure of the [localTM](#) filter.

```
#include <localTM.h>
```

Data Fields

- [UInt16](#) * [curves](#)

6.99.1 Detailed Description

Parameter structure of the [localTM](#) filter.

6.99.2 Field Documentation

[UInt16](#)* [LocalTMParam::curves](#)

Referenced by [svuLocalTM\(\)](#).

6.100 LowLvlCorrNPIParam Struct Reference

Parameter structure of the [Low Level Correction on Multiple Planes](#) filter.

```
#include <lowLvlCorrMultiplePlanes.h>
```

Data Fields

- [UInt8](#) [blackLevel](#)
black level
- float [alphaBadPixel](#)
level of correction

6.100.1 Detailed Description

Parameter structure of the [Low Level Correction on Multiple Planes](#) filter.

6.100.2 Field Documentation

float [LowLvlCorrNPIParam::alphaBadPixel](#)

level of correction

Referenced by [svulowLvlCorrMultiplePlanes\(\)](#).

UInt8 LowLvlCorrNPIParam::blackLevel

black level

Referenced by svulowLvlCorrMultiplePlanes().

6.101 LowLvlCorrParam Struct Reference

Parameter structure of the [Low Level Correction](#) filter.

```
#include <lowLvlCorr.h>
```

Data Fields

- **UInt8** blackLevel
black level
- float alphaBadPixel
level of correction

6.101.1 Detailed Description

Parameter structure of the [Low Level Correction](#) filter.

6.102 LscParam Struct Reference

Parameter structure of the lsc filter.

```
#include <sippHwDefs_ma2x5x.h>
```

Data Fields

- **UInt32** frmDim
Reserved field yes.
- **UInt32** fraction
Reserved field yes.
- **UInt32** gmDim
Reserved field yes.
- **UInt32** cfg
Reserved field yes.
- **UInt16** * gmBase
Gain Map base.
- **UInt32** gmWidth
Gain Map width.
- **UInt32** gmHeight
Gain Map height.
- **UInt32** dataFormat

- data format: Planar(0), Bayer(1)*
 - UInt32 dataWidth**
data width (8-16 bit)

6.102.1 Detailed Description

Parameter structure of the lsc filter.

6.102.2 Field Documentation

UInt32 LscParam::cfg

Reserved field yes.

UInt32 LscParam::dataFormat

data format: Planar(0), Bayer(1)

UInt32 LscParam::dataWidth

data width (8-16 bit)

UInt32 LscParam::fraction

Reserved field yes.

UInt32 LscParam::frmDim

Reserved field yes.

UInt16* LscParam::gmBase

Gain Map base.

UInt32 LscParam::gmDim

Reserved field yes.

UInt32 LscParam::gmHeight

Gain Map height.

UInt32 LscParam::gmWidth

Gain Map width.

6.103 Lut10to16Param Struct Reference

Parameter structure of the [LUT 10 to 16](#) filter.

```
#include <lut10to16.h>
```

Data Fields

- [UInt16 lutValue](#) [1024]

Look-up table of 256 elements; should have the same depth as the input line.

6.103.1 Detailed Description

Parameter structure of the [LUT 10 to 16](#) filter.

6.103.2 Field Documentation

[UInt16](#) Lut10to16Param::lutValue[1024]

Look-up table of 256 elements; should have the same depth as the input line.

Referenced by `svuLut10to16()`.

6.104 Lut10to8Param Struct Reference

Parameter structure of the [LUT 10 to 8](#) filter.

```
#include <lut10to8.h>
```

Data Fields

- [UInt8 lutValue](#) [1024]

Look-up table of 1024 elements; should have the same depth as the input line.

6.104.1 Detailed Description

Parameter structure of the [LUT 10 to 8](#) filter.

6.104.2 Field Documentation

[UInt8](#) Lut10to8Param::lutValue[1024]

Look-up table of 1024 elements; should have the same depth as the input line.

Referenced by `svuLut10to8()`.

6.105 Lut12to16Param Struct Reference

Parameter structure of the [LUT 12 to 16](#) filter.

```
#include <lut12to16.h>
```

Data Fields

- [UInt16 lutValue](#) [8193]

Look-up table of 8193 elements; should have the same depth as the input line.

6.105.1 Detailed Description

Parameter structure of the [LUT 12 to 16](#) filter.

6.105.2 Field Documentation

[UInt16](#) Lut12to16Param::lutValue[8193]

Look-up table of 8193 elements; should have the same depth as the input line.

Referenced by `svuLut12to16()`.

6.106 Lut12to8Param Struct Reference

Parameter structure of the [LUT 12 to 8](#) filter.

```
#include <lut12to8.h>
```

Data Fields

- [UInt8 lutValue](#) [4096]

Look-up table of 4096 elements; should have the same depth as the input line.

6.106.1 Detailed Description

Parameter structure of the [LUT 12 to 8](#) filter.

6.106.2 Field Documentation

[UInt8](#) Lut12to8Param::lutValue[4096]

Look-up table of 4096 elements; should have the same depth as the input line.

Referenced by `svuLut12to8()`.

6.107 Lut8to8Param Struct Reference

Parameter structure of the [LUT 8 to 8](#) filter.

```
#include <lut8to8.h>
```

Data Fields

- [UInt8 lutValue](#) [256]

Look-up table of 256 elements; should have the same depth as the input line.

6.107.1 Detailed Description

Parameter structure of the [LUT 8 to 8](#) filter.

6.107.2 Field Documentation

[UInt8 Lut8to8Param::lutValue](#)[256]

Look-up table of 256 elements; should have the same depth as the input line.

Referenced by `svuLut8to8()`.

6.108 LutParam Struct Reference

Parameter structure of the [lut](#) filter.

```
#include <sippHwDefs_ma2x5x.h>
```

Data Fields

- [UInt32 frmDim](#)
Reserved field yes.
- [UInt32 cfg](#)
configuration bitfield(see SIPP_LUT_CFG_ADR)
- [UInt32 sizeA](#)
see SIPP_LUT_SIZES7_0_ADR
- [UInt32 sizeB](#)
see SIPP_LUT_SIZES15_8_ADR
- `void * lut`
pointer to UInt8 or fp16 Lookup Table
- [UInt32 lutFormat](#)
Set format of LUT.
- [UInt16 mat](#) [9]
3x3 array of CSC matrix coeffs
- [UInt16 offset](#) [3]
Color conversion offsets in S(1,12) format.

6.108.1 Detailed Description

Parameter structure of the [lut](#) filter.

6.108.2 Field Documentation

UInt32 LutParam::cfg

configuration bitfield(see SIPP_LUT_CFG_ADR)

Referenced by `sippCheckOPipeConnectionLut()`.

UInt32 LutParam::frmDim

Reserved field yes.

void* LutParam::lut

pointer to UInt8 or fp16 Lookup Table

UInt32 LutParam::lutFormat

Set format of LUT.

UInt16 LutParam::mat[9]

3x3 array of CSC matrix coeffs

UInt16 LutParam::offset[3]

Color conversion offsets in S(1,12) format.

UInt32 LutParam::sizeA

see SIPP_LUT_SIZES7_0_ADR

UInt32 LutParam::sizeB

see SIPP_LUT_SIZES15_8_ADR

6.109 MaxTest3x3fp16Param Struct Reference

Parameter structure of the [maxTest3x3_fp16](#) filter.

```
#include <maxTest3x3_fp16.h>
```

Data Fields

- half * [inBufferCandidates](#)
- [UInt32](#) * [maxLocationsIn](#)
- [UInt32](#) [maxCountIn](#)

6.109.1 Detailed Description

Parameter structure of the [maxTest3x3_fp16](#) filter.

6.109.2 Field Documentation

[half*](#) [MaxTest3x3fp16Param::inBufferCandidates](#)

Referenced by [svuMaxTest3x3_fp16\(\)](#).

[UInt32](#) [MaxTest3x3fp16Param::maxCountIn](#)

Referenced by [svuMaxTest3x3_fp16\(\)](#).

[UInt32*](#) [MaxTest3x3fp16Param::maxLocationsIn](#)

Referenced by [svuMaxTest3x3_fp16\(\)](#).

6.110 MedParam Struct Reference

Parameter structure of the median filter.

```
#include <sippHwDefs_ma2x5x.h>
```

Data Fields

- [UInt32](#) [frmDim](#)
Reserved field yes.
- [UInt32](#) [cfg](#)
configuration bit field, see [SIPP_MED_CFG_ADR](#)
- [UInt32](#) [lumaAlpha](#)
configuration bit field, see [SIPP_MED_LUMA_ALPHA_ADR](#)

6.110.1 Detailed Description

Parameter structure of the median filter.

- name: input1; datatypes: [UInt8](#), [fp16](#); kernels: 1x0, 3x0

6.110.2 Field Documentation

UInt32 MedParam::cfg

configuration bit field, see SIPP_MED_CFG_ADR

Referenced by sippGenericScheduleSetBufConsModels().

UInt32 MedParam::frmDim

Reserved field yes.

UInt32 MedParam::lumaAlpha

configuration bit field, see SIPP_MED_LUMA_ALPHA_ADR

6.111 memRegDescriptor Struct Reference

```
#include <sippTypesPrivate.h>
```

Data Fields

- [u32 freeBase](#)
- [u32 sizeRemaining](#)

6.111.1 Field Documentation

u32 memRegDescriptor::freeBase

Referenced by sippAllocCmxLineBuffersOPipe().

u32 memRegDescriptor::sizeRemaining

Referenced by sippAllocCmxLineBuffersOPipe().

6.112 MinMaxPosParam Struct Reference

Parameter structure of the [Min/Max Value Position](#) filter.

```
#include <minMaxPos.h>
```

Data Fields

- [UInt8 * Mask](#)
mask filled with 1s and 0s which determines the image area to compute minimum and maximum

6.112.1 Detailed Description

Parameter structure of the [Min/Max Value Position](#) filter.

6.112.2 Field Documentation

UInt8* MinMaxPosParam::Mask

mask filled with 1s and 0s which determines the image area to compute minimum and maximum
Referenced by `svuMinMaxPos()`.

6.113 minMaxValParam Struct Reference

Parameter structure of the [Min/Max Value](#) filter.

```
#include <minMaxValue.h>
```

Data Fields

- **UInt8 minVal**
stores the minimum value on the line
- **UInt8 maxVal**
stores the maximum value on the line
- **UInt8 * maskAddr**
mask filled with 1s and 0s which determines the image area to compute minimum and maximum

6.113.1 Detailed Description

Parameter structure of the [Min/Max Value](#) filter.

6.113.2 Field Documentation

UInt8* minMaxValParam::maskAddr

mask filled with 1s and 0s which determines the image area to compute minimum and maximum
Referenced by `svuMinMaxValue()`.

UInt8 minMaxValParam::maxVal

stores the maximum value on the line
Referenced by `svuMinMaxValue()`.

UInt8 minMaxValParam::minVal

stores the minimum value on the line
Referenced by `svuMinMaxValue()`.

6.114 MinTest3x3fp16Param Struct Reference

Parameter structure of the `minTest3x3_fp16` filter.

```
#include <minTest3x3_fp16.h>
```

Data Fields

- `half * inBufferCandidates`
- `UInt32 * minLocationsIn`
- `UInt32 minCountIn`

6.114.1 Detailed Description

Parameter structure of the `minTest3x3_fp16` filter.

6.114.2 Field Documentation

`half* MinTest3x3fp16Param::inBufferCandidates`

Referenced by `svuMinTest3x3_fp16()`.

`UInt32 MinTest3x3fp16Param::minCountIn`

Referenced by `svuMinTest3x3_fp16()`.

`UInt32* MinTest3x3fp16Param::minLocationsIn`

Referenced by `svuMinTest3x3_fp16()`.

6.115 MipiRxLoopbackParam Struct Reference

```
#include <sippHwDefs_ma2x5x.h>
```

Data Fields

- `UInt32 rxID`
- `UInt8 * imgAddr`
- `UInt32 imgW`
- `UInt32 imgH`
- `UInt32 bpp`

6.115.1 Field Documentation

UInt32 MipiRxLoopbackParam::bpp

UInt8* MipiRxLoopbackParam::imgAddr

UInt32 MipiRxLoopbackParam::imgH

UInt32 MipiRxLoopbackParam::imgW

UInt32 MipiRxLoopbackParam::rxID

6.116 MipiRxParam Struct Reference

Parameter structure of the **MIPI Rx** filter.

```
#include <sippHwDefs_ma2x5x.h>
```

Data Fields

- **UInt32** frmDim
- **UInt32** cfg
*configuration bit field(see SIPP_MIPI_RX*_CFG_ADR)*
- **UInt32** winX [4]
Window x co-ordinate configuration.
- **UInt32** winY [4]
Window y co-ordinate configuration.
- **UInt32** sel01
Selection enables and least significant bit for windows 0 and 1.
- **UInt32** sel23
Selection enables and least significant bit for windows 2 and 3.
- **UInt32** selMask [4]
Selection mask.
- **UInt32** black01
Black levels for windows 0 and 1 or even lines of RAW Bayer data.
- **UInt32** black23
Black levels for windows 2 and 3 or odd lines of RAW Bayer data.
- **UInt32** vbp
Vertical back porch and private slice stride.

6.116.1 Detailed Description

Parameter structure of the **MIPI Rx** filter.

6.116.2 Field Documentation

UInt32 MipiRxParam::black01

Black levels for windows 0 and 1 or even lines of RAW Bayer data.
Referenced by sippLoadMipiRx().

UInt32 MipiRxParam::black23

Black levels for windows 2 and 3 or odd lines of RAW Bayer data.
Referenced by sippLoadMipiRx().

UInt32 MipiRxParam::cfg

configuration bit field(see SIPP_MIPI_RX*_CFG_ADR)
Referenced by sippLoadMipiRx().

UInt32 MipiRxParam::frmDim

Private:

yes

Referenced by sippInitMipiRx(), and sippLoadMipiRx().

UInt32 MipiRxParam::sel01

Selection enables and least significant bit for windows 0 and 1.
Referenced by sippLoadMipiRx().

UInt32 MipiRxParam::sel23

Selection enables and least significant bit for windows 2 and 3.
Referenced by sippLoadMipiRx().

UInt32 MipiRxParam::selMask[4]

Selection mask.
Referenced by sippLoadMipiRx().

UInt32 MipiRxParam::vbp

Vertical back porch and private slice stride.
Referenced by sippLoadMipiRx().

UInt32 MipiRxParam::winX[4]

Window x co-ordinate configuration.

Referenced by sippLoadMipiRx().

UInt32 MipiRxParam::winY[4]

Window y co-ordinate configuration.

Referenced by sippLoadMipiRx().

6.117 MipiTxLoopbackParam Struct Reference

```
#include <sippHwDefs_ma2x5x.h>
```

Data Fields

- **UInt32** txID
- **UInt8** * imgAddr
- **UInt32** imgW
- **UInt32** imgH
- **UInt32** bpp
- **UInt32** hbp
- **UInt32** hfp
- **UInt32** hsync
- **UInt32** vsync

6.117.1 Field Documentation

UInt32 MipiTxLoopbackParam::bpp

UInt32 MipiTxLoopbackParam::hbp

UInt32 MipiTxLoopbackParam::hfp

UInt32 MipiTxLoopbackParam::hsync

UInt8* MipiTxLoopbackParam::imgAddr

UInt32 MipiTxLoopbackParam::imgH

UInt32 MipiTxLoopbackParam::imgW

UInt32 MipiTxLoopbackParam::txID

UInt32 MipiTxLoopbackParam::vsync

6.118 MipiTxParam Struct Reference

Parameter structure of the [MIPI Rx](#) filter.

```
#include <sippHwDefs_ma2x5x.h>
```

Data Fields

- [UInt32 frmDim](#)
- [UInt32 cfg](#)
configuration bit field
- [UInt32 lineCompare](#)
Line count at which to generate line compare interrupt.
- [UInt32 vCompare](#)
Vertical interval in which to generate vertical interval interrupt.
- [UInt32 hSyncWidth](#)
Specifies the width, in PCLK clock periods, of the horizontal sync pulse (value programmed is HSW-1)
- [UInt32 hBackPorch](#)
Specifies the number of PCLK clocks from the end of the horizontal sync pulse to the start of horizontal active (value programmed is HBP so a back porch of 0 cycles can be set)
- [UInt32 hActiveWidth](#)
Specifies the number of PCLK clocks in the horizontal active section (value programmed is AVW-1)
- [UInt32 hFrontPorch](#)
Specifies the number of PCLK clocks from end of active video to the start of horizontal sync (value programmed is HFP)
- [UInt32 vSyncWidth](#)
Specifies the width in lines of the vertical sync pulse (value programmed is VSW-1).
- [UInt32 vBackPorch](#)
Specifies the number of lines from the end of the vertical sync pulse to the start of vertical active (value programmed is VBP)
- [UInt32 vActiveHeight](#)
Specifies the number of lines in the vertical active section (value programmed is AVH-1)
- [UInt32 vFrontPorch](#)
Specifies the number of lines from the end of active data to the start of vertical sync pulse (value programmed is VFP).
- [UInt32 vSyncStartOff](#)
Number of PCLKs from the start of the last horizontal sync pulse in the Vertical Front Porch to the start of the vertical sync pulse.
- [UInt32 vSyncEndOff](#)
Number of PCLKs from the end of the last horizontal sync pulse in the Vertical Sync Active to the end of the vertical sync pulse.

6.118.1 Detailed Description

Parameter structure of the [MIPI Rx](#) filter.

6.118.2 Field Documentation

UInt32 MipiTxParam::cfg

configuration bit field

Referenced by sippLoadMipiTx().

UInt32 MipiTxParam::frmDim

Private:

yes

Referenced by sippInitMipiTx(), and sippLoadMipiTx().

UInt32 MipiTxParam::hActiveWidth

Specifies the number of PCLK clocks in the horizontal active section (value programmed is AVW-1)

Referenced by sippLoadMipiTx().

UInt32 MipiTxParam::hBackPorch

Specifies the number of PCLK clocks from the end of the horizontal sync pulse to the start of horizontal active (value programmed is HBP so a back porch of 0 cycles can be set)

Referenced by sippLoadMipiTx().

UInt32 MipiTxParam::hFrontPorch

Specifies the number of PCLK clocks from end of active video to the start of horizontal sync (value programmed is HFP)

Referenced by sippLoadMipiTx().

UInt32 MipiTxParam::hSyncWidth

Specifies the width, in PCLK clock periods, of the horizontal sync pulse (value programmed is HSW-1)

Referenced by sippLoadMipiTx().

UInt32 MipiTxParam::lineCompare

Line count at which to generate line compare interrupt.

Referenced by sippLoadMipiTx().

UInt32 MipiTxParam::vActiveHeight

Specifies the number of lines in the vertical active section (value programmed is AVH-1)

Referenced by sippLoadMipiTx().

UInt32 MipiTxParam::vBackPorch

Specifies the number of lines from the end of the vertical sync pulse to the start of vertical active (value programmed is VBP)

Referenced by sippLoadMipiTx().

UInt32 MipiTxParam::vCompare

Vertical interval in which to generate vertical interval interrupt.

Referenced by sippLoadMipiTx().

UInt32 MipiTxParam::vFrontPorch

Specifies the number of lines from the end of active data to the start of vertical sync pulse (value programmed is VFP).

Referenced by sippLoadMipiTx().

UInt32 MipiTxParam::vSyncEndOff

Number of PCLKs from the end of the last horizontal sync pulse in the Vertical Sync Active to the end of the vertical sync pulse.

Referenced by sippLoadMipiTx().

UInt32 MipiTxParam::vSyncStartOff

Number of PCLKs from the start of the last horizontal sync pulse in the Vertical Front Porch to the start of the vertical sync pulse.

Referenced by sippLoadMipiTx().

UInt32 MipiTxParam::vSyncWidth

Specifies the width in lines of the vertical sync pulse (value programmed is VSW-1).

Referenced by sippLoadMipiTx().

6.119 MixMedianParam Struct Reference

Parameter structure of the [Mix Median](#) filter.

```
#include <mixMedian.h>
```

Data Fields

- float [offset](#)
offset
- float [slope](#)

slope

6.119.1 Detailed Description

Parameter structure of the **Mix Median** filter.

6.119.2 Field Documentation

`float MixMedianParam::offset`

offset

`float MixMedianParam::slope`

slope

6.120 MonoImbalanceParam Struct Reference

Parameter structure of the monoImbalance filter.

```
#include <monoImbalance.h>
```

Data Fields

- `int thrDark`
- `int thrBright`
- `int inputBits`

6.120.1 Detailed Description

Parameter structure of the monoImbalance filter.

6.120.2 Field Documentation

`int MonoImbalanceParam::inputBits`

Referenced by `svuMonoImbalance()`.

`int MonoImbalanceParam::thrBright`

Referenced by `svuMonoImbalance()`.

`int MonoImbalanceParam::thrDark`

Referenced by `svuMonoImbalance()`.

6.121 nonMax3x3Fp32Param Struct Reference

```
#include <nonMax3x3Fp32.h>
```

Data Fields

- [UInt16 * candPos](#)

6.121.1 Field Documentation

[UInt16*](#) [nonMax3x3Fp32Param::candPos](#)

Referenced by [svuNonMax3x3Fp32\(\)](#).

6.122 ParentInfoS Struct Reference

```
#include <sippTypes.h>
```

Data Fields

- [u32 nPlanes](#)
- [u32 outOff](#)

6.122.1 Field Documentation

[u32](#) [ParentInfoS::nPlanes](#)

Referenced by [sippAsmOptSetup\(\)](#).

[u32](#) [ParentInfoS::outOff](#)

Referenced by [sippAsmOptSetup\(\)](#).

6.123 PixelUnpackerMipi10bParam Struct Reference

Parameter structure of the [Pixel Unpacker Mipi 10b](#) filter.

```
#include <pixelUnpackerMipi10b.h>
```

Data Fields

- [UInt32 lineNo](#)
Line number
- [UInt32 coefs](#) [4]
coefficients for black correction

- **UInt32 noMipiRxWorkaround**

If image is from MA2100, need software workaround to unpack 10-bit data into 16-bit words. If not set, do black-level subtraction only.

6.123.1 Detailed Description

Parameter structure of the **Pixel Unpacker Mipi 10b** filter.

6.123.2 Field Documentation

UInt32 PixelUnpackerMipi10bParam::coefs[4]

coefficients for black correction

Referenced by svuPixelUnpackerMipi10b().

UInt32 PixelUnpackerMipi10bParam::lineNo

Line number

UInt32 PixelUnpackerMipi10bParam::noMipiRxWorkaround

If image is from MA2100, need software workaround to unpack 10-bit data into 16-bit words. If not set, do black-level subtraction only.

Referenced by svuPixelUnpackerMipi10b().

6.124 PixelUnpackerParam Struct Reference

Parameter structure of the **Pixel Unpacker** filter.

```
#include <pixelUnpacker.h>
```

Data Fields

- **UInt8 shift**

sf_desc number of bits for shifting the result to left

6.124.1 Detailed Description

Parameter structure of the **Pixel Unpacker** filter.

6.124.2 Field Documentation

UInt8 PixelUnpackerParam::shift

sf_desc number of bits for shifting the result to left

Referenced by svuPixelUnpacker().

6.125 PixelUnpackerWBParam Struct Reference

Parameter structure of the [Pixel Unpacker WB](#) filter.

```
#include <pixelUnpackerWB.h>
```

Data Fields

- [UInt8 shift](#)
sf_desc number of bits for shifting the result to left
- [UInt16 * awbCoef](#)

6.125.1 Detailed Description

Parameter structure of the [Pixel Unpacker WB](#) filter.

6.125.2 Field Documentation

[UInt16*](#) PixelUnpackerWBParam::awbCoef

sf_desc coefficients needed for white balance

Size of the array behind:

4

Referenced by [svuPixelUnpackerWB\(\)](#).

[UInt8](#) PixelUnpackerWBParam::shift

sf_desc number of bits for shifting the result to left

Referenced by [svuPixelUnpackerWB\(\)](#).

6.126 PolyFirParam Struct Reference

Parameter structure of the polyphasefir filter.

```
#include <sippHwDefs_ma2x5x.h>
```

Data Fields

- [UInt32 cfgReg](#)
Reserved field yes.
- [UInt32 kerSz](#)
Reserved field yes.
- [UInt32 frmDimPar](#)
Reserved field yes.

- **UInt32 frmDimFlt**
Reserved field yes.
- **PolyModes mode**
Reserved field yes.
- **PolyScalerType autoType**
Reserved field yes.
- **UInt32 clamp**
clamp enable
- **UInt32 horzD**
Horizontal Denominator factor.
- **UInt32 horzN**
Numerator factor
- **UInt32 vertD**
Vertical Denominator factor.
- **UInt32 vertN**
Vertical Numerator factor.
- **UInt8 * horzCoefs**
*pointer to horizontal filter coefficients(see SIPP_UPFIRDN_HCOEFF_*_ADR)*
- **UInt8 * vertCoefs**
*pointer to vertical filter coefficients(see SIPP_UPFIRDN_VCOEFF_*_ADR)*
- **PolyPlaneMode planeMode**
Enables override of filter plane mode for running multiple units on same stream.

6.126.1 Detailed Description

Parameter structure of the polyphasefir filter.

6.126.2 Field Documentation

PolyScalerType PolyFirParam::autoType

Reserved field yes.

UInt32 PolyFirParam::cfgReg

Reserved field yes.

UInt32 PolyFirParam::clamp

clamp enable

UInt32 PolyFirParam::frmDimFlt

Reserved field yes.

UInt32 PolyFirParam::frmDimPar

Reserved field yes.

UInt8* PolyFirParam::horzCoefs

pointer to horizontal filter coefficients(see SIPP_UPFIRDN_HCOEFF_*_ADR)

UInt32 PolyFirParam::horzD

Horizontal Denominator factor.

UInt32 PolyFirParam::horzN

Numerator factor

UInt32 PolyFirParam::kerSz

Reserved field yes.

PolyModes PolyFirParam::mode

Reserved field yes.

PolyPlaneMode PolyFirParam::planeMode

Enables override of filter plane mode for running multiple units on same stream.

UInt8* PolyFirParam::vertCoefs

pointer to vertical filter coefficients(see SIPP_UPFIRDN_VCOEFF_*_ADR)

UInt32 PolyFirParam::vertD

Vertical Denominator factor.

Referenced by askPolyFirResizer().

UInt32 PolyFirParam::vertN

Vertical Numerator factor.

Referenced by askPolyFirResizer().

6.127 positionKernelParam Struct Reference

Parameter structure of the [Pixel Position](#) filter.

```
#include <positionKernel.h>
```

Data Fields

- [UInt8 * maskAddr](#)
mask filled with 1s and 0s which determines the image area to find position
- [UInt8 pixelValue](#)
stores the pixel value to be searched
- [UInt32 pixelPosition](#)
stores the position occupied by the searched value within line
- [UInt8 status](#)
stores 0x11 if pixel value found, else 0x00

6.127.1 Detailed Description

Parameter structure of the [Pixel Position](#) filter.

6.127.2 Field Documentation

[UInt8* positionKernelParam::maskAddr](#)

mask filled with 1s and 0s which determines the image area to find position

Referenced by [svuPositionKernel\(\)](#).

[UInt32 positionKernelParam::pixelPosition](#)

stores the position occupied by the searched value within line

Referenced by [svuPositionKernel\(\)](#).

[UInt8 positionKernelParam::pixelValue](#)

stores the pixel value to be searched

Referenced by [svuPositionKernel\(\)](#).

[UInt8 positionKernelParam::status](#)

stores 0x11 if pixel value found, else 0x00

Referenced by [svuPositionKernel\(\)](#).

6.128 PpAf Struct Reference

Parameter structure of the [AF Stats](#) filter.

```
#include <extAfStats.h>
```

Data Fields

- struct {
 - UInt32 firstPatchX
 - UInt32 firstPatchY
 - UInt32 patchWidth
 - UInt32 patchHeight
 - UInt32 nPatchesX
 - UInt32 nPatchesY
 - UInt32 nSkipRows
 - Int32 initialSubtractionValue
 - Int32 f1Threshold
 - Int32 f2Threshold
 - Int32 f2Coeffs [11]
 - Int32 f1Coeffs [11]
- } afConfig
- UInt32 * outStatsBuffer
- UInt32 runNr [2]
- UInt32 posInPaxel [2]
- UInt32 paxelNr [2]
- Int32 a_f1f2 [4]
- Int32 b_f1f2 [4]
- UInt32 mask
- SippHwIOBuf in [2]
- UInt32 * outStatsBufferOutside
- void(* statsAf0)(void *)
- UInt32 shaveNr
- UInt32 stackPointer

6.128.1 Detailed Description

Parameter structure of the [AF Stats](#) filter.

6.128.2 Field Documentation

Int32 PpAf::a_f1f2[4]

struct { ... } PpAf::afConfig

Int32 PpAf::b_f1f2[4]

Int32 PpAf::f1Coeffs[11]

Int32 PpAf::f1Threshold

Int32 PpAf::f2Coeffs[11]

Int32 PpAf::f2Threshold

UInt32 PpAf::firstPatchX

UInt32 PpAf::firstPatchY

SippHwIOBuf PpAf::in[2]

Referenced by svuExtAfStats().

Int32 PpAf::initialSubtractionValue

UInt32 PpAf::mask

UInt32 PpAf::nPatchesX

UInt32 PpAf::nPatchesY

UInt32 PpAf::nSkipRows

UInt32* PpAf::outStatsBuffer

UInt32* PpAf::outStatsBufferOutside

UInt32 PpAf::patchHeight

UInt32 PpAf::patchWidth

UInt32 PpAf::paxelNr[2]

UInt32 PpAf::posInPaxel[2]

UInt32 PpAf::runNr[2]

UInt32 PpAf::shaveNr

Referenced by svuExtAfStats().

UInt32 PpAf::stackPointer

Referenced by svuExtAfStats().

void(* PpAf::statsAf0)(**void** *)

Referenced by svuExtAfStats().

6.129 PurpleFlareParam Struct Reference

Parameter structure of the [purpleFlare](#) filter.

```
#include <purpleFlare.h>
```

Data Fields

- [UInt8 strength](#)

6.129.1 Detailed Description

Parameter structure of the [purpleFlare](#) filter.

6.129.2 Field Documentation

UInt8 PurpleFlareParam::strength

6.130 RandNoiseFp16Param Struct Reference

Parameter structure of the [Random Noise \(high speed\)](#) filter.

```
#include <randNoiseFp16.h>
```

Data Fields

- float [strength](#)
noise strength [0,1]

6.130.1 Detailed Description

Parameter structure of the [Random Noise \(high speed\)](#) filter.

6.130.2 Field Documentation

float RandNoiseFp16Param::strength

noise strength [0,1]

Referenced by `svuGenNoiseFp16()`.

6.131 RandNoiseParam Struct Reference

Parameter structure of the [Random Noise](#) filter.

```
#include <randNoise.h>
```

Data Fields

- float [strength](#)
noise strength [0,1]

6.131.1 Detailed Description

Parameter structure of the [Random Noise](#) filter.

6.131.2 Field Documentation

float RandNoiseParam::strength

noise strength [0,1]

Referenced by `svuGenNoise()`.

6.132 RawParam Struct Reference

Parameter structure of the raw filter.

```
#include <sippHwDefs_ma2x5x.h>
```

Data Fields

- [UInt32 frmDim](#)
Reserved field yes.
- [UInt32 grgbPlat](#)
Maximum local green difference reduction(see SIPP_GRGB_PLATO_ADR)
- [UInt32 grgbDecay](#)
Decay control of local green difference reduction(see SIPP_GRGB_SLOPE_ADR)
- [UInt32 badPixCfg](#)
Filter aggressiveness control(see SIPP_BAD_PIXEL_CFG_ADR)
- [UInt32 cfg](#)
configuration bit field(see SIPP_RAW_CFG_ADR)
- [UInt32 gainSat](#) [4]
array containing 4 gain values
- [UInt32 * statsBase](#)
maps on O_BASE(SIPP_STATS_ID)
- [UInt32 statsPlanes](#)
see SIPP_RAW_STATS_PLANES_ADR
- [UInt32 statsFrmDim](#)
Reserved field yes.
- [UInt32 statsPatchCfg](#)
see SIPP_STATS_PATCH_CFG_ADR
- [UInt32 statsPatchStart](#)

- *see SIPP_STATS_PATCH_START_ADR*
- [UInt32 statsPatchSkip](#)
see SIPP_STATS_PATCH_SKIP_ADR
- [UInt32 statsThresh](#)
Thresholds for AE/AWB stats - see SIPP_RAW_STATS_THRESHOLDS_ADR.
- [Int32 afF1coefs](#) [11]
Array of coefficients for auto-focus stats filter 1.
- [Int32 afF2coefs](#) [11]
Array of coefficients for auto-focus stats filter 2.
- [UInt32 afMinThresh](#)
Minimum thresholds for per-patch auto-focus stats accumulation.
- [UInt32 afSubtract](#)
Initial subtraction value at start of IIR filter.
- [UInt32 afPatchCfg](#)
Accumulation patch configuration for auto-focus statistics.
- [UInt32 afPatchStart](#)
Start location of first (top-left) patch, and hence of the ROI, for auto-focus statistics.
- [UInt32 * afStatsBase](#)
Pointer to stats Base.
- [UInt32 * histLumaBase](#)
Pointer to Luma Histogram Base.
- [UInt32 * histRgbBase](#)
Pointer to RGB Histogram Base.

6.132.1 Detailed Description

Parameter structure of the raw filter.

6.132.2 Field Documentation

[Int32 RawParam::afF1coefs\[11\]](#)

Array of coefficients for auto-focus stats filter 1.

[Int32 RawParam::afF2coefs\[11\]](#)

Array of coefficients for auto-focus stats filter 2.

[UInt32 RawParam::afMinThresh](#)

Minimum thresholds for per-patch auto-focus stats accumulation.

[UInt32 RawParam::afPatchCfg](#)

Accumulation patch configuration for auto-focus statistics.

UInt32 RawParam::afPatchStart

Start location of first (top-left) patch, and hence of the ROI, for auto-focus statistics.

UInt32* RawParam::afStatsBase

Pointer to stats Base.

UInt32 RawParam::afSubtract

Initial subtraction value at start of IIR filter.

UInt32 RawParam::badPixCfg

Filter aggressiveness control(see SIPP_BAD_PIXEL_CFG_ADR)

UInt32 RawParam::cfg

configuration bit field(see SIPP_RAW_CFG_ADR)

UInt32 RawParam::frmDim

Reserved field yes.

UInt32 RawParam::gainSat[4]

array containing 4 gain values

UInt32 RawParam::grgbDecay

Decay control of local green difference reduction(see SIPP_GRGB_SLOPE_ADR)

UInt32 RawParam::grgbPlat

Maximum local green difference reduction(see SIPP_GRGB_PLATO_ADR)

UInt32* RawParam::histLumaBase

Pointer to Luma Histogram Base.

UInt32* RawParam::histRgbBase

Pointer to RGB Histogram Base.

UInt32* RawParam::statsBase

maps on [O_BASE\(SIPP_STATS_ID\)](#)

UInt32 RawParam::statsFrmDim

Reserved field yes.

UInt32 RawParam::statsPatchCfg

see [SIPP_STATS_PATCH_CFG_ADR](#)

UInt32 RawParam::statsPatchSkip

see [SIPP_STATS_PATCH_SKIP_ADR](#)

UInt32 RawParam::statsPatchStart

see [SIPP_STATS_PATCH_START_ADR](#)

UInt32 RawParam::statsPlanes

see [SIPP_RAW_STATS_PLANES_ADR](#)

UInt32 RawParam::statsThresh

Thresholds for AE/AWB stats - see [SIPP_RAW_STATS_THRESHOLDS_ADR](#).

6.133 SchedInfoS Struct Reference

```
#include <sippTypes.h>
```

Data Fields

- [u32 sippHwWaitMask](#)
- [u32 shaveMask](#)
- [u32 dmaMask](#)
- [u32 allMask](#)

6.133.1 Field Documentation

u32 SchedInfoS::allMask

Referenced by [sippDbgDumpFilterOuts\(\)](#), [sippDbgDumpSchedForVcsCArr\(\)](#), [sippGenericLinePrepare\(\)](#), [sippGenericSchedWr\(\)](#), [sippGenericStartUnits\(\)](#), and [sippGetFirstHwFiltIdx\(\)](#).

u32 SchedInfoS::dmaMask

Referenced by dmaKickSequence(), dmaKickSequenceCQ(), sippCheckIterComplete(), sippDbgDumpSchedForVcsCArr(), sippGenericRunNextIter(), sippGenericSchedWr(), sippGenericStartUnits(), sippGenericWaitUnits(), sippKickDma(), and sippKickDmaCQ().

u32 SchedInfoS::shaveMask

Referenced by SHAVE_MAIN(), sippCheckIterComplete(), sippDbgDumpAsmOffsets(), sippDbgDumpSchedForVcsCArr(), sippGenericSchedWr(), sippGenericStartUnits(), and sippGenericWaitUnits().

u32 SchedInfoS::sippHwWaitMask

Referenced by sippCheckIterComplete(), sippDbgDumpFilterOuts(), sippDbgDumpSchedForVcsCArr(), sippGenericBlockHWUnits2x5x(), sippGenericLinePrepare(), sippGenericSchedWr(), sippGenericStartHWUnits2x5x(), sippGenericStartUnits(), sippGenericWaitUnits(), and sippObflIncHandler().

6.134 Semaphore Class Reference

```
#include <wrapperSem.h>
```

Public Member Functions

- [Semaphore](#) (unsigned int InitialCount, unsigned int MaxCount)
Constructor.
- [~Semaphore](#) ()
Destructor.
- int [Post](#) ()
Member functions.
- int [Wait](#) ()
- [Semaphore](#) (unsigned int InitialCount, unsigned int MaxCount)
- [~Semaphore](#) ()
- int [Post](#) ()
- int [Wait](#) ()

Private Attributes

- sem_t [posix_sem](#)

6.134.1 Constructor & Destructor Documentation

[Semaphore::Semaphore](#) (unsigned int InitialCount, unsigned int MaxCount)

Constructor.

`Semaphore::~~Semaphore ()`

Destructor.

`Semaphore::Semaphore (unsigned int InitialCount, unsigned int MaxCount)`

`Semaphore::~~Semaphore ()`

6.134.2 Member Function Documentation

`int Semaphore::Post ()`

Member functions.

Referenced by `sippCheckIterComplete()`, `sippGenericRunNextIter()`, `sippKickShaveM1PC()`, and `sippPalTrace()`.

`int Semaphore::Post ()`

`int Semaphore::Wait ()`

`int Semaphore::Wait ()`

Referenced by `sippCheckIterComplete()`, and `sippPalTrace()`.

6.134.3 Field Documentation

`sem_t Semaphore::posix_sem [private]`

Referenced by `Post()`, `Semaphore()`, `Wait()`, and `~Semaphore()`.

6.135 SigmaParam Struct Reference

Parameter structure of the edgeoperator filter.

```
#include <sippHwDefs_ma2x5x.h>
```

Data Fields

- [UInt32 frmDim](#)
Reserved field yes.
- [UInt32 thresh](#) [2]
???
- [UInt32 cfg](#)
configuration bitfield(see SIPP_EDGE_OP_CFG_ADR)
- [UInt32 bayerPattern](#)
Bayer Pattern.
- [UInt32 blcGR](#)

- Black level Correction GR.*
 - [UInt32 blcR](#)
- Black level Correction R.*
 - [UInt32 blcB](#)
- Black level Correction B.*
 - [UInt32 blcGB](#)
- Black level Correction GB.*

6.135.1 Detailed Description

Parameter structure of the edgeoperator filter.

6.135.2 Field Documentation

[UInt32 SigmaParam::bayerPattern](#)

Bayer Pattern.

Referenced by `sippLoadSigma()`.

[UInt32 SigmaParam::blcB](#)

Black level Correction B.

Referenced by `sippLoadSigma()`.

[UInt32 SigmaParam::blcGB](#)

Black level Correction GB.

Referenced by `sippLoadSigma()`.

[UInt32 SigmaParam::blcGR](#)

Black level Correction GR.

Referenced by `sippLoadSigma()`.

[UInt32 SigmaParam::blcR](#)

Black level Correction R.

Referenced by `sippLoadSigma()`.

[UInt32 SigmaParam::cfg](#)

configuration bitfield(see `SIPP_EDGE_OP_CFG_ADR`)

Referenced by `sippLoadSigma()`.

UInt32 SigmaParam::frmDim

Reserved field yes.

Referenced by sippInitSigma(), and sippLoadSigma().

UInt32 SigmaParam::thresh[2]

???

Referenced by sippLoadSigma().

6.136 SIPP_ACCESS_SCHEDULER Struct Reference

```
#include <sippAccessSchedulerTypes.h>
```

Data Fields

- **SIPP_ACCESS_SCHEDULER_QU** SippQu
- **eSIPP_ACCESS_SCHEDULER_EVENT** ePendActions [SIPP_MAX_SUPPORTED_PIPELINES][SIPP_MAX_EVENTS_PER_PIPE]
- **SIPP_ACCESS_SCHEDULER_EVENT_DATA** pPendData [SIPP_MAX_SUPPORTED_PIPELINES][SIPP_MAX_EVENTS_PER_PIPE]
- **eSIPP_ACCESS_SCHEDULER_EVENT** eControlPipeAction
- **SIPP_ACCESS_SCHEDULER_EVENT_DATA** pControlPipePendData
- **volatile u32** uSWPendList
- **volatile u32** uHWPendList
- **bool** bWaitLongest [SIPP_MAX_SUPPORTED_PIPELINES]
- **u32** uHWWaitList [SIPP_MAX_SUPPORTED_PIPELINES]
- **u32** uPipeEventWrIndex [SIPP_MAX_SUPPORTED_PIPELINES]
- **u32** uPipeEventRdIndex [SIPP_MAX_SUPPORTED_PIPELINES]
- **u32** uPipeHWEventCount [SIPP_MAX_SUPPORTED_PIPELINES]

6.136.1 Field Documentation

bool SIPP_ACCESS_SCHEDULER::bWaitLongest[SIPP_MAX_SUPPORTED_PIPELINES]

eSIPP_ACCESS_SCHEDULER_EVENT SIPP_ACCESS_SCHEDULER::eControlPipeAction

Referenced by sippAccessSchedulerInit().

eSIPP_ACCESS_SCHEDULER_EVENT SIPP_ACCESS_SCHEDULER::ePendActions[SIPP_MAX_SUPPORTED_PIPELINES][SIPP_MAX_EVENTS_PER_PIPE]

SIPP_ACCESS_SCHEDULER_EVENT_DATA SIPP_ACCESS_SCHEDULER::pControlPipePendData

SIPP_ACCESS_SCHEDULER_EVENT_DATA SIPP_ACCESS_SCHEDULER::pPendData[SIPP_MAX_SUPPORTED_PIPELINES][SIPP_MAX_EVENTS_PER_PIPE]

SIPP_ACCESS_SCHEDULER_QU SIPP_ACCESS_SCHEDULER::SippQu

Referenced by sippAccessSchedulerInit(), and sippAccessSchedulerQuPush().

volatile **u32** SIPP_ACCESS_SCHEDULER::uHWPendList

Referenced by sippAccessScheduleCheckPending(), and sippAccessSchedulerInit().

u32 SIPP_ACCESS_SCHEDULER::uHWaitList[SIPP_MAX_SUPPORTED_PIPELINES]

u32 SIPP_ACCESS_SCHEDULER::uPipeEventRdIndex[SIPP_MAX_SUPPORTED_PIPELINES]

u32 SIPP_ACCESS_SCHEDULER::uPipeEventWrIndex[SIPP_MAX_SUPPORTED_PIPELINES]

u32 SIPP_ACCESS_SCHEDULER::uPipeHWEventCount[SIPP_MAX_SUPPORTED_PIPELINE-S]

volatile **u32** SIPP_ACCESS_SCHEDULER::uSWPendList

6.137 SIPP_ACCESS_SCHEDULER_QU Struct Reference

```
#include <sippAccessSchedulerTypes.h>
```

Data Fields

- **SIPP_ACCESS_SCHEDULER_QU_ENTRY** Qu [SIPP_ACCESS_SCHEDULER_QU_SIZE]
- **u32** uWrIdx
- **u32** uRdIdx
- **u32** uSize

6.137.1 Field Documentation

SIPP_ACCESS_SCHEDULER_QU_ENTRY SIPP_ACCESS_SCHEDULER_QU::Qu[SIPP_ACCESS_SCHEDULER_QU_SIZE]

Referenced by sippAccessSchedulerQuPush().

u32 SIPP_ACCESS_SCHEDULER_QU::uRdIdx

Referenced by sippAccessSchedulerInit().

u32 SIPP_ACCESS_SCHEDULER_QU::uSize

Referenced by sippAccessSchedulerQuPush().

u32 SIPP_ACCESS_SCHEDULER_QU::uWrIdx

Referenced by sippAccessSchedulerInit(), and sippAccessSchedulerQuPush().

6.138 SIPP_ACCESS_SCHEDULER_QU_ENTRY Struct Reference

```
#include <sippAccessSchedulerTypes.h>
```

Data Fields

- **eSIPP_ACCESS_SCHEDULER_EVENT** eEvent
- **SIPP_ACCESS_SCHEDULER_EVENT_DATA** pEventData
- **u32** uPipeIdx

6.138.1 Field Documentation

eSIPP_ACCESS_SCHEDULER_EVENT SIPP_ACCESS_SCHEDULER_QU_ENTRY::eEvent

Referenced by sippAccessSchedulerQuPush().

SIPP_ACCESS_SCHEDULER_EVENT_DATA SIPP_ACCESS_SCHEDULER_QU_ENTRY::p-EventData

Referenced by sippAccessSchedulerQuPush().

u32 SIPP_ACCESS_SCHEDULER_QU_ENTRY::uPipeIdx

Referenced by sippAccessSchedulerQuPush().

6.139 SIPP_HW_SESSION Struct Reference

```
#include <sippTypesPrivate.h>
```

Data Fields

- **eSIPP_HW_STATE** eState
- **u32** uHWFeatures
- **pSippPipeline** pSippPrevSvuPipe [SIPP_NUM_SVUS]
- **pSippPipeline** pSippCurrSvuPipe [SIPP_NUM_SVUS]
- **pSippPipeline** pSippLoadedHWPipe [SIPP_MAX_SUPPORTED_HW_PIPELINES]
- **pSippPipeline** pSippCurrHWPipe [SIPP_MAX_SUPPORTED_HW_PIPELINES]
- **u32** uHWFilterUsageBitMask [SIPP_HW_FILTER_MASK_SIZE]
- **u32** uNumCurrHwPipes
- **u32** useIntBar [SIPP_MAX_SUPPORTED_HW_PIPELINES]
- **u32** isrSetup
- **u32** noIntBarSwitchPend

6.139.1 Field Documentation

eSIPP_HW_STATE SIPP_HW_SESSION::eState

Referenced by sippHWSessionInit().

u32 SIPP_HW_SESSION::isrSetup

Referenced by sippIntBarrierSetup().

u32 SIPP_HW_SESSION::noIntBarSwitchPend

pSippPipeline SIPP_HW_SESSION::pSippCurrHWPipe[SIPP_MAX_SUPPORTED_HW_PIPELINES]

Referenced by sippHWSessionAddActiveLists(), sippHWSessionInit(), and sippHWSessionRemoveActiveLists().

pSippPipeline SIPP_HW_SESSION::pSippCurrSvuPipe[SIPP_NUM_SVUS]

Referenced by sippHWSessionAddActiveLists(), sippHWSessionInit(), sippHWSessionRemoveActiveLists(), and sippSvuDoneIrqHandler().

pSippPipeline SIPP_HW_SESSION::pSippLoadedHWPipe[SIPP_MAX_SUPPORTED_HW_PIPELINES]

Referenced by sippHWSessionAddActiveLists(), sippHWSessionRemoveLoadedPipe(), sippIbflDecHandler(), sippObflIncHandler(), and sippPipeSessionControl().

pSippPipeline SIPP_HW_SESSION::pSippPrevSvuPipe[SIPP_NUM_SVUS]

Referenced by sippHWSessionCommand(), and sippHWSessionRemoveActiveLists().

u32 SIPP_HW_SESSION::uHWFeatures

u32 SIPP_HW_SESSION::uHWFilterUsageBitMask[SIPP_HW_FILTER_MASK_SIZE]

Referenced by sippHWSessionAddActiveLists(), and sippHWSessionRemoveActiveLists().

u32 SIPP_HW_SESSION::uNumCurrHwPipes

Referenced by sippHWSessionAddActiveLists(), sippHWSessionInit(), sippHWSessionRemoveActiveLists(), and sippProcessFrame().

u32 SIPP_HW_SESSION::useIntBar[SIPP_MAX_SUPPORTED_HW_PIPELINES]

6.140 SIPP_PAL_QU Struct Reference

```
#include <sippPalTypes.h>
```

Data Fields

- uint16_t [elementsInQ](#)
- uint16_t [maxQElements](#)
- uint16_t [qFlags](#)
- mqd_t [rtemsQuId](#)
- mqd_t [rtemsRWQuId](#)

6.140.1 Field Documentation

uint16_t SIPP_PAL_QU::elementsInQ

Referenced by [sippPalQuCreate\(\)](#), [sippPalQuPost\(\)](#), and [sippPalQuReceive\(\)](#).

uint16_t SIPP_PAL_QU::maxQElements

Referenced by [sippPalQuCreate\(\)](#), and [sippPalQuPost\(\)](#).

uint16_t SIPP_PAL_QU::qFlags

Referenced by [sippPalQuCreate\(\)](#), [sippPalQuPost\(\)](#), and [sippPalQuReceive\(\)](#).

mqd_t SIPP_PAL_QU::rtemsQuId

Referenced by [sippPalQuCreate\(\)](#), [sippPalQuDestroy\(\)](#), [sippPalQuPost\(\)](#), and [sippPalQuReceive\(\)](#).

mqd_t SIPP_PAL_QU::rtemsRWQuId

Referenced by [sippPalQuAttach\(\)](#).

6.141 SIPP_PAL_THREAD Struct Reference

```
#include <sippPalTypes.h>
```

Data Fields

- uint32_t [args](#) [4]
- uint32_t [stackSize](#)
- uint8_t [priority](#)
- pthread_t [rtemsThread](#)

6.141.1 Field Documentation

`uint32_t SIPP_PAL_THREAD::args[4]`

Referenced by `sippPalThreadCreate()`.

`uint8_t SIPP_PAL_THREAD::priority`

Referenced by `sippPalThreadCreate()`.

`pthread_t SIPP_PAL_THREAD::rtemsThread`

Referenced by `sippPalThreadCreate()`, and `sippPalThreadTerminate()`.

`uint32_t SIPP_PAL_THREAD::stackSize`

Referenced by `sippPalThreadCreate()`.

6.142 SIPP_PIPELINE_FINALISED_DATA Struct Reference

```
#include <sippEvents.h>
```

Data Fields

- `u32 uDummy`

6.142.1 Field Documentation

`u32 SIPP_PIPELINE_FINALISED_DATA::uDummy`

6.143 SIPP_TRACE_FLAGS Struct Reference

```
#include <sippPalTypes.h>
```

Data Fields

- `u32 Flags [SIPP_TRACE_FLAGS_WORDS]`

6.143.1 Field Documentation

`u32 SIPP_TRACE_FLAGS::Flags`

6.144 SippCmxBufferMapS Struct Reference

```
#include <sippTypesPrivate.h>
```

Data Fields

- [u32 cmxSliceUsageBitMask](#)
- [u32 numCmxSlicesAvail](#)
- [u32 totalMem](#)
- [pSippMemRegionListNode pCmxSliceRegionList \[CMX_NSLICES\]](#)

6.144.1 Field Documentation

[u32 SippCmxBufferMapS::cmxSliceUsageBitMask](#)

Referenced by [sippAllocCmxLineBuffers\(\)](#), and [sippMemLBMatchRegionsToChunks\(\)](#).

[u32 SippCmxBufferMapS::numCmxSlicesAvail](#)

Referenced by [sippMemLBMatchRegionsToChunks\(\)](#).

[pSippMemRegionListNode SippCmxBufferMapS::pCmxSliceRegionList\[CMX_NSLICES\]](#)

Referenced by [sippAllocCmxLineBuffers\(\)](#), [sippAllocCmxLineBuffersOPipe\(\)](#), [sippMapRegionToCmx\(\)](#), [sippMemLBConsolidateRegions\(\)](#), [sippMemLBMatchRegionsToChunks\(\)](#), [sippMemLBRemoveNode\(\)](#), and [sippMemRegionAllocLineBuffer\(\)](#).

[u32 SippCmxBufferMapS::totalMem](#)

Referenced by [sippAssignCmxMemRegion\(\)](#).

6.145 SippFilterS Struct Reference

```
#include <sippTypes.h>
```

Data Fields

- [u32 exeNo](#)
- [u32 nPlanes \[SIPP_FILTER_MAX_OBUFS\]](#)
- [u32 nCons](#)
- [u32 outputH](#)
- [u32 linesPerIter](#)
- [u32 firstIterLines](#)
- [u32 firstRunNumLines](#)
- [u32 firstRunRollLines](#)
- [u32 outLineDeficit](#)
- [u32 schNo](#)
- [SippFilter * parents \[SIPP_FILTER_MAX_PARENTS\]](#)
- [u32 parentsKS \[SIPP_FILTER_MAX_PARENTS\]](#)
- [ParentInfo parInfo \[SIPP_FILTER_MAX_PARENTS\]](#)

- pSippPipeline pPipe
- FnSvuRun funcSvuRun
- void * params
- u32 flags
- u32 nParents
- u32 nLinesUsed [SIPP_FILTER_MAX_PARENTS]
- u32 id
- u32 firstOutSlc
- u32 nCtxLoads
- u32 unit
- const CommInfo * gi
- SippFilter * cons [SIPP_FILTER_MAX_CONSUMERS]
- u32 * dbLinesIn [SIPP_FILTER_MAX_PARENTS][2][SIPP_MAX_LINES_PER_ITER]
- u32 outputW
- u32 sliceWidth
- u32 sliceWidthLastSvu
- u32 linesThisIter [2]
- u32 hPadding [SIPP_FILTER_MAX_OBUFS]
- u32 bpp [SIPP_FILTER_MAX_OBUFS]
- s32 * outOff [SIPP_FILTER_MAX_OBUFS]
- u8 * dbLineOut [SIPP_FILTER_MAX_OBUFS][2][SIPP_MAX_LINES_PER_ITER]
- u8 * lnToPad [SIPP_FILTER_MAX_OBUFS][2][SIPP_MAX_LINES_PER_ITER]
- u32 lineStride [SIPP_FILTER_MAX_OBUFS]
- u32 planeStride [SIPP_FILTER_MAX_OBUFS]
- u8 * outputBuffer [SIPP_FILTER_MAX_OBUFS]
- u8 * outLinePtr [SIPP_FILTER_MAX_OBUFS][SIPP_MAX_LINES_PER_ITER]
- u32 outLineOffset [SIPP_FILTER_MAX_OBUFS]
- u8 * linePtrs [SIPP_FILTER_MAX_OBUFS]
- u8 ** linePtrs1stBase [SIPP_FILTER_MAX_OBUFS]
- u8 * linePtrs2ndBase [SIPP_FILTER_MAX_OBUFS]
- u8 * linePtrs3rdBase [SIPP_FILTER_MAX_OBUFS]
- u8 * linePtrs4thBase [SIPP_FILTER_MAX_OBUFS]
- u8 * linePtrs5thBase [SIPP_FILTER_MAX_OBUFS]
- u8 oBufAlloc [SIPP_FILTER_MAX_OBUFS]
- u32 nLines [SIPP_FILTER_MAX_OBUFS]
- u32 parentOBufIdx [SIPP_FILTER_MAX_IBUFS]
- SippHwBuf * iBuf [SIPP_FILTER_MAX_IBUFS]
- SippHwBuf * oBuf [SIPP_FILTER_MAX_OBUFS]
- SippSchEnt * sch
- pSippFilterSchedInfo pOpipesch
- psSchLineBuffer oBufs [SIPP_FILTER_MAX_OBUFS]
- psSchLineBuffer iBufs [SIPP_FILTER_MAX_IBUFS]
- SchedIBufUsageInfo * iBufCtx [SIPP_FILTER_MAX_IBUFS]
- u32 numOBufs
- u32 numIBufs
- u32 consIbufIdx [SIPP_FILTER_MAX_CONSUMERS]
- u16 * KSIterList [SIPP_FILTER_MAX_IBUFS]

- `SippPixelChunkPos * ptrChunkPos`
- `u32 errorStatus [SIPP_ERROR_MASK_SIZE]`
- `u32 bytesPerPix`
- `u8 linesPerIterShift`

6.145.1 Field Documentation

`u32 SippFilterS::bpp[SIPP_FILTER_MAX_OBUFS]`

Referenced by `sippAddFilterToPipe()`, `sippAllocCmxLineBuffers()`, `sippAllocCmxLineBuffersOPipe()`, `sippBuildLnBufs()`, `sippComputeBufferProps()`, `sippComputePaddingOffsets()`, `sippComputeSwOutCt()`, `sippDbgCreateDumpFiles()`, `sippDbgDumpAsmOffsets()`, `sippDbgDumpFilterOuts()`, `sippDbgDumpGraph()`, `sippFilterAddOBuf()`, `sippFilterSetBufBitsPP()`, `sippHorizontalPadding()`, `sippIbufSetup()`, `sippInitBufferLnPointers()`, `sippInitDma()`, `sippMemAllocChainChunk()`, `sippMemRegionAllocLineBuffer()`, and `sippOBufSetup()`.

`u32 SippFilterS::bytesPerPix`

Referenced by `sippAddFilterToPipe()`, `sippDbgDumpAsmOffsets()`, `sippFilterAddOBuf()`, and `sippFilterGetOutputBpp()`.

`SippFilter* SippFilterS::cons[SIPP_FILTER_MAX_CONSUMERS]`

Referenced by `adjustNodesRecursively()`, `sippAnalysePipe2x5x()`, `sippComputeChunkWidthsSW()`, `sippDbgDumpGraph()`, `sippFindConnectionsListRecursive()`, `sippGenericSchCreateSEFromFilter()`, `sippLinkFilter()`, `sippOSEProcessFilterIBufs()`, and `sippOSEProcessFilterOBufs()`.

`u32 SippFilterS::consIbufIdx[SIPP_FILTER_MAX_CONSUMERS]`

Referenced by `sippAnalysePipe2x5x()`, `sippGenericSchCreateSEFromFilter()`, `sippLinkFilter()`, `sippOSEProcessFilterIBufs()`, and `sippOSEProcessFilterOBufs()`.

`u8* SippFilterS::dbLineOut[SIPP_FILTER_MAX_OBUFS][2][SIPP_MAX_LINES_PER_ITER]`

Referenced by `getOutPtr()`, `getPlaneIoPtrs()`, `sippDbgDumpAsmOffsets()`, `sippUpdateDmaAddr()`, `sippUpdateDmaAddrCQ()`, and `svuCensusMin64()`.

`u32* SippFilterS::dbLinesIn[SIPP_FILTER_MAX_PARENTS][2][SIPP_MAX_LINES_PER_ITER]`

Referenced by `getIn3PlanePtr()`, `getInPtr()`, `getInPtrAbs()`, `getPlaneIoPtrs()`, `sippDbgDumpAsmOffsets()`, `sippUpdateDmaAddr()`, and `sippUpdateDmaAddrCQ()`.

`u32 SippFilterS::errorStatus[SIPP_ERROR_MASK_SIZE]`

Referenced by `adjustNodesRecursively()`, `sippComputeBufferProps()`, `sippFilterAddOBuf()`, `sippGenericSchAllocTempStorage()`, `sippInitDma()`, `sippLinkFilter()`, `sippLinkFilterSetOBuf()`, `sippOSEAddFilter()`, and `sippOSEProcessFilterIBufs()`.

u32 SippFilterS::exeNo

Referenced by getIn3PlanePtr(), getInPtr(), getInPtrAbs(), getOutPtr(), getPlaneIoPtrs(), sippDbgDumpAsmOffsets(), sippDbgDumpFilterOuts(), sippFilterGetLinesThisIter(), sippGenericRuntimeFrameReset(), sippGenericStartUnits(), sippGetChunkStartPixelPos(), sippHorizontalPadding(), sippUpdateDmaAddr(), svuCensusMin64(), svuCombDecimDemosaicAwbGains(), svuCombDecimDemosaicAwbGainsStats(), svuCvtColorRGBtoUV(), svuGenDnsRef(), svuGenDnsRefFp16(), svuHomography(), svuIntegralImageSqSumF32M2(), svuIntegralImageSqSumU32M2(), svuIntegralImageSumF32M2(), svuIntegralImageSumU16U32(), svuIntegralImageSumU32M2(), svuLocalTM(), svuPadBayer5(), svuPadBayer5Frame(), svuPixelUnpackerMipi10b(), svuPixelUnpackerWB(), svuScale2xBilinHV_025_075_Fp16ToFp16(), svuScale2xBilinHV_025_075_U16ToU16(), svuScale2xBilinHV_Fp16U8_phase025_075(), svuScale2xBilinHV_U8ToU8_phase025_075(), svuScl2xBilinHV(), svuScl2xLancHV(), svuScl2xLancV(), svuSclBilinArb(), svuUndistortBrown(), svuWhiteBalanceBayerGBRG(), and svuXYgen().

u32 SippFilterS::firstIterLines

Referenced by sippResetFilterVariables().

u32 SippFilterS::firstOutSlc

Referenced by sippAddFilterToPipe(), sippComputeBufferProps(), sippComputePaddingOffsets(), sippComputeSwOutCt(), sippDbgDumpAsmOffsets(), sippDumpHtmlMap(), sippHorizontalPadding(), sippIbufSetup(), sippObufSetup(), sippUpdateDmaAddr(), and sippUpdateDmaAddrCQ().

u32 SippFilterS::firstRunNumLines

Referenced by sippDbgDumpSchedForVcsCArr(), and sippResetFilterVariables().

u32 SippFilterS::firstRunRollLines

Referenced by sippGenericLinePrepare(), sippGenericRuntimeFrameReset(), and sippResetFilterVariables().

u32 SippFilterS::flags

Referenced by sippAddFilterToPipe(), sippAllocCmxLineBuffersOPipe(), sippComputeBufferProps(), sippComputeSwOutCt(), sippCoreReInitLineBuffers(), sippGenericLinePrepare(), sippGenericRuntimeFrameReset(), sippGenericSchCreateSEFromFilter(), sippGenericScheduleSetBufConsModels(), sippInitBufferLnPointers(), sippLinkFilter(), and sippOSEAddFilter().

FnSvuRun SippFilterS::funcSvuRun

Referenced by SHAVE_MAIN(), sippAddFilterToPipe(), and sippDbgDumpAsmOffsets().

const CommInfo* SippFilterS::gi

Referenced by sippAddFilterToPipe(), sippDbgDumpAsmOffsets(), sippIbufSetup(), sippInitDma(), sippObufSetup(), sippUpdateDmaAddr(), sippUpdateDmaAddrCQ(), svuExtAfStats(), svuExtStatsSat-PixelsU32(), svuGenDnsRef(), svuGenDnsRefFp16(), svuHomography(), svuPadBayer5(), svuPad-Bayer5Frame(), and svuUndistortBrown().

u32 SippFilterS::hPadding[SIPP_FILTER_MAX_OBUFS]

Referenced by sippBuildLnBufs(), sippComputeBufferProps(), sippComputePaddingOffsets(), sipp-DbgDumpAsmOffsets(), sippDbgDumpSchedForVcsCArr(), sippFilterAddOBuf(), sippHorizontal-Padding(), sippIbufSetup(), sippInitBufferLnPointers(), sippObufSetup(), and sippProcessSchedData().

SippHwBuf* SippFilterS::iBuf[SIPP_FILTER_MAX_IBUFS]

Referenced by sippGenericRuntimeFrameReset(), sippIbufSetup(), sippLoadMipiTx(), and sippLoad-Sigma().

SchedIBufUsageInfo* SippFilterS::iBufCtx[SIPP_FILTER_MAX_IBUFS]

Referenced by sippAnalysePipe2x5x(), sippCoreReInitLineBuffers(), sippCoreSetPaddingReqs(), sipp-GenericSchCreateSEFromFilter(), sippGenericScheduleSetBufConsModels(), sippLinkFilter(), sippOS-EProcessFilterIBufs(), sippSetBufLatenciesMipiTx(), and sippSetBufLatenciesSigma().

psSchLineBuffer SippFilterS::iBufs[SIPP_FILTER_MAX_IBUFS]

Referenced by sippAnalysePipe2x5x(), sippCoreSetPaddingReqs(), sippGenericSchCreateSEFrom-Filter(), sippLinkFilter(), sippLinkFilterSetOBuf(), sippOSEProcessFilterIBufs(), and sippOSEProcess-FilterOBufs().

u32 SippFilterS::id

Referenced by sippAddFilterToPipe(), sippDbgCreateDumpFiles(), sippDbgDumpFilterOuts(), sipp-DbgDumpGraph(), and sippDumpHtmlMap().

u16* SippFilterS::KSIterList[SIPP_FILTER_MAX_IBUFS]

Referenced by sippDbgDumpSchedForVcsCArr().

u8* SippFilterS::linePtrs[SIPP_FILTER_MAX_OBUFS]

Referenced by sippDbgShowBuffPtr(), sippGenericLinePrepare(), sippGenericRuntimeFrameReset(), and sippInitBufferLnPointers().

u8 SippFilterS::linePtrs1stBase[SIPP_FILTER_MAX_OBUFS]**

Referenced by sippGenericRuntimeFrameReset(), and sippInitBufferLnPointers().

u8* SippFilterS::linePtrs2ndBase[SIPP_FILTER_MAX_OBUFS]

Referenced by sippGenericLinePrepare(), and sippInitBufferLnPointers().

u8* SippFilterS::linePtrs3rdBase[SIPP_FILTER_MAX_OBUFS]

Referenced by sippGenericLinePrepare(), and sippInitBufferLnPointers().

u8* SippFilterS::linePtrs4thBase[SIPP_FILTER_MAX_OBUFS]

Referenced by sippGenericLinePrepare(), and sippInitBufferLnPointers().

u8* SippFilterS::linePtrs5thBase[SIPP_FILTER_MAX_OBUFS]

Referenced by sippGenericLinePrepare(), and sippInitBufferLnPointers().

u32 SippFilterS::linesPerIter

Referenced by sippAddFilterToPipe(), sippAnalysePipe2x5x(), sippChainDmaDesc(), sippCheckOPipeConnectionChrDns(), sippCheckOPipeConnectionColourComb(), sippCheckOPipeConnectionGenChroma(), sippCheckOPipeConnectionMedian(), sippDbgDumpAsmOffsets(), sippDbgDumpFilterOuts(), sippDbgDumpSchedForVcsCArr(), sippFilterGetLinesPerIter(), sippGenericSchedWr(), sippHorizontalPadding(), sippIncrementOutBufs(), sippInitBufferLnPointers(), sippInitDma(), sippObufSetup(), sippOSEAddFilter(), sippOSEProcessFilter(), sippOSEProcessFilterIBufs(), sippSetOBufLevelsMipiRx0(), sippSetOBufLevelsMipiRx1(), sippSetOBufLevelsMipiRx2(), sippSetOBufLevelsMipiRx3(), sippSetOBufLevelsSigma(), sippUpdateDmaAddr(), and sippUpdateDmaAddrCQ().

u8 SippFilterS::linesPerIterShift

u32 SippFilterS::linesThisIter[2]

Referenced by sippAddFilterToPipe(), sippDbgDumpAsmOffsets(), sippFilterGetLinesThisIter(), and sippGenericSchedPipeInit().

u32 SippFilterS::lineStride[SIPP_FILTER_MAX_OBUFS]

Referenced by sippAllocCmxLineBuffers(), sippAllocCmxLineBuffersOPipe(), sippComputeBufferProps(), sippIbufSetup(), sippInitBufferLnPointers(), sippMemAllocChainChunk(), sippMemRegionAllocLineBuffer(), sippObufSetup(), and svuHarrisResponse().

u8* SippFilterS::lnToPad[SIPP_FILTER_MAX_OBUFS][2][SIPP_MAX_LINES_PER_ITER]

Referenced by sippDbgDumpAsmOffsets(), sippGenericLinePrepare(), sippGenericRuntimeFrameReset(), and sippHorizontalPadding().

u32 SippFilterS::nCons

Referenced by `adjustNodesRecursively()`, `sippAnalysePipe2x5x()`, `sippComputeChunkWidths()`, `sippComputeChunkWidthsSW()`, `sippDbgDumpGraph()`, `sippFindConnectionsListRecursive()`, `sippGenericLinePrepare()`, `sippGenericSchCreateSEFromFilter()`, `sippInitDma()`, `sippLinkFilter()`, `sippOSEProcessFilterIBufs()`, `sippOSEProcessFilterOBufs()`, `sippUpdateDmaAddr()`, and `sippUpdateDmaAddrCQ()`.

u32 SippFilterS::nCtxLoads

u32 SippFilterS::nLines[SIPP_FILTER_MAX_OBUFS]

Referenced by `sippAllocCmxLineBuffers()`, `sippAllocCmxLineBuffersOPipe()`, `sippBuildLnBufs()`, `sippComputeBufferProps()`, `sippDbgDumpAsmOffsets()`, `sippDbgDumpFilterOuts()`, `sippDbgDumpGraph()`, `sippDbgDumpSchedForVcsCArr()`, `sippDbgShowBuffPtr()`, `sippDumpHtmlMap()`, `sippFilterAddOBuf()`, `sippFilterGetParentInputLines()`, `sippGenericLinePrepare()`, `sippGenericRuntimeFrameReset()`, `sippIbufSetup()`, `sippIncrementOutBufs()`, `sippInitBufferLnPointers()`, `sippMemAllocChainChunk()`, `sippMemRegionAllocLineBuffer()`, `sippObufSetup()`, `sippPipeSetNumLinesPerBuf()`, `sippProcessSchedData()`, `sippSetOBufLevelsMipiRx0()`, `sippSetOBufLevelsMipiRx1()`, `sippSetOBufLevelsMipiRx2()`, `sippSetOBufLevelsMipiRx3()`, `sippSetOBufLevelsSigma()`, and `sippUsingPrecompSched()`.

u32 SippFilterS::nLinesUsed[SIPP_FILTER_MAX_PARENTS]

Referenced by `getPlaneIoPtrs()`, `sippDbgDumpAsmOffsets()`, and `sippLinkFilter()`.

u32 SippFilterS::nParents

Referenced by `adjustNodesRecursively()`, `sippAnalysePipe2x5x()`, `sippAsmOptSetup()`, `sippCoreReInitLineBuffers()`, `sippDbgDumpSchedForVcsCArr()`, `sippFindConnectionsListRecursive()`, `sippGenericScheduleSetBufConsModels()`, `sippIdentifyOPipeSchedulingEntity()`, `sippIncrementOutBufs()`, `sippInitDma()`, `sippLinkFilter()`, `sippLinkFilterSetOBuf()`, `sippOSEProcessFilterIBufs()`, `sippUpdateDmaAddr()`, and `sippUpdateDmaAddrCQ()`.

u32 SippFilterS::nPlanes[SIPP_FILTER_MAX_OBUFS]

Referenced by `getIn3PlanePtr()`, `getInPtr()`, `getInPtrAbs()`, `getOutPtr()`, `getPlaneIoPtrs()`, `sippAddFilterToPipe()`, `sippAllocCmxLineBuffers()`, `sippAllocCmxLineBuffersOPipe()`, `sippAsmOptSetup()`, `sippBuildLnBufs()`, `sippComputePaddingOffsets()`, `sippComputeSwOutCt()`, `sippDbgCreateDumpFiles()`, `sippDbgDumpFilterOuts()`, `sippDbgDumpGraph()`, `sippFilterAddOBuf()`, `sippFilterGetNumOutPlanes()`, `sippHorizontalPadding()`, `sippIbufSetup()`, `sippInitDma()`, `sippLoadMipiRx()`, `sippLoadSigma()`, `sippMemAllocChainChunk()`, `sippMemRegionAllocLineBuffer()`, `sippObufSetup()`, and `svu-CensusMin64()`.

u32 SippFilterS::numIBufs

Referenced by `sippCoreSetPaddingReqs()`, `sippGenericSchCreateSEFromFilter()`, `sippLinkFilter()`, and `sippValidatePipe()`.

u32 SippFilterS::numOBufs

Referenced by sippAddFilterToPipe(), sippAllocCmxLineBuffers(), sippAllocCmxLineBuffersOPipe(), sippAnalysePipe2x5x(), sippBuildLnBufs(), sippComputeBufferProps(), sippComputeSwOutCt(), sippCoreReInitLineBuffers(), sippDbgDumpSchedForVcsCarr(), sippFilterAddOBuf(), sippGenericLinePrepare(), sippGenericRuntimeFrameReset(), sippGenericRuntimeLoadPipeline(), sippGenericSchCreateSEFromFilter(), sippGenericSchedWr(), sippIncrementOutBufs(), sippInitBufferLnPointers(), sippOSEProcessFilterIBufs(), sippProcessSchedData(), sippUsingPrecompSched(), and sippValidatePipe().

SippHwBuf* SippFilterS::oBuf[SIPP_FILTER_MAX_OBUFS]

Referenced by sippGenericRuntimeFrameReset(), sippLoadMipiRx(), sippLoadSigma(), and sippOBufSetup().

u8 SippFilterS::oBufAlloc[SIPP_FILTER_MAX_OBUFS]

Referenced by sippAllocCmxLineBuffers(), sippAllocCmxLineBuffersOPipe(), sippDumpHtmlMap(), sippFilterAddOBuf(), and sippMemRegionAllocLineBuffer().

psSchLineBuffer SippFilterS::oBufs[SIPP_FILTER_MAX_OBUFS]

Referenced by sippAddFilterToPipe(), sippAllocCmxLineBuffers(), sippAnalysePipe2x5x(), sippCheckOPipeConnectionColourComb(), sippCheckOPipeConnectionDoGLTM(), sippCheckOPipeConnectionMedian(), sippCheckOPipeConnectionPoly(), sippCoreReInitLineBuffers(), sippCreateFilter(), sippFilterAddOBuf(), sippGenericSchCreateSEFromFilter(), sippGenericSchedWr(), sippIbufSetup(), sippLinkFilter(), sippLinkFilterSetOBuf(), sippOBufSetup(), sippOSEProcessFilterIBufs(), sippPipeSetNumLinesPerBuf(), and sippProcessSchedData().

u32 SippFilterS::outLineDeficit

Referenced by sippInitBufferLnPointers().

u32 SippFilterS::outLineOffset[SIPP_FILTER_MAX_OBUFS]

Referenced by sippFilterAddOBuf().

u8* SippFilterS::outLinePtr[SIPP_FILTER_MAX_OBUFS][SIPP_MAX_LINES_PER_ITER]

Referenced by sippDbgDumpAsmOffsets(), sippDbgShowBuffPtr(), sippGenericLinePrepare(), and sippGenericRuntimeFrameReset().

s32* SippFilterS::outOff[SIPP_FILTER_MAX_OBUFS]

Referenced by getIn3PlanePtr(), getInPtr(), getInPtrAbs(), getOutPtr(), getPlaneIoPtrs(), sippAsmOptSetup(), sippComputeSwOutCt(), sippDbgDumpAsmOffsets(), sippDbgDumpFilterOuts(), sippFilterAddOBuf(), and svuCensusMin64().

u8* SippFilterS::outputBuffer[SIPP_FILTER_MAX_OBUFS]

Referenced by sippAllocCmxLineBuffers(), sippAllocCmxLineBuffersOPipe(), sippDumpHtmlMap(), sippFilterAddOBuf(), sippIbufSetup(), sippInitBufferLnPointers(), sippMemRegionAllocLineBuffer(), and sippObufSetup().

u32 SippFilterS::outputH

Referenced by sippAddFilterToPipe(), sippCoreReInitLineBuffers(), sippDbgCreateDumpFiles(), sippDbgDumpFilterOuts(), sippFilterAddOBuf(), sippFilterGetParentOutputHeight(), sippGenericLinePrepare(), sippGenericSchedWr(), sippInitBufferLnPointers(), sippInitDma(), sippInitMipiRx(), sippInitMipiTx(), sippInitSigma(), sippLinkFilter(), sippUpdateDmaAddr(), sippUpdateDmaAddrCQ(), svuCombDecimDemosaicAwbGains(), svuCombDecimDemosaicAwbGainsStats(), svuGenDnsRef(), svuGenDnsRefFp16(), svuHomography(), svuPadBayer5(), svuPadBayer5Frame(), svuScIbilinArb(), and svuUndistortBrown().

u32 SippFilterS::outputW

Referenced by adjustNodesRecursively(), sippAddFilterToPipe(), sippComputeBufferProps(), sippComputeChunkWidths(), sippComputeChunkWidthsSW(), sippDbgCreateDumpFiles(), sippDbgDumpFilterOuts(), sippFilterGetParentOutputWidth(), sippInitDma(), sippInitMipiRx(), sippInitMipiTx(), sippInitSigma(), svuGenDnsRef(), svuGenDnsRefFp16(), svuHomography(), svuPadBayer5(), svuPadBayer5Frame(), svuScIbilinArb(), svuUndistortBrown(), and upscale2xH().

void* SippFilterS::params

Referenced by sippBufGetObufCtx(), sippChainDmaDesc(), sippCheckOPipeConnectionGenChroma(), sippCheckOPipeConnectionLut(), sippCoreReInitLineBuffers(), sippCreateFilter(), sippGenericScheduleSetBufConsModels(), sippInitDma(), sippInitMipiRx(), sippInitMipiTx(), sippInitSigma(), sippLinkFilter(), sippLoadMipiRx(), sippLoadMipiTx(), sippLoadSigma(), sippUpdateDmaAddr(), sippUpdateDmaAddrCQ(), svuAccumulateWeighted(), svuBilateral5x5(), svuBoxFilter(), svuBoxFilter11x11(), svuBoxFilter13x13(), svuBoxFilter15x15(), svuBoxFilter3x3(), svuBoxFilter5x5(), svuBoxFilter7x7(), svuBoxFilter9x9(), svuCannyEdgeDetection(), svuCensusMatching32(), svuCensusMatching64(), svuCensusMatchingPyr(), svuChannelExtract(), svuChromaBlock(), svuCombDecimDemosaicAwbGains(), svuCombDecimDemosaicAwbGainsStats(), svuContrast(), svuConv11x11(), svuConv15x1(), svuConv1x15(), svuConv1x5(), svuConv1x5Fp16ToFp16(), svuConv1x7(), svuConv1x7Fp16ToFp16(), svuConv1x9(), svuConv3x3(), svuConv3x3Fp16ToFp16(), svuConv5x1(), svuConv5x1Fp16ToFp16(), svuConv5x5(), svuConv5x5Fp16ToFp16(), svuConv7x1(), svuConv7x1Fp16ToFp16(), svuConv7x7(), svuConv7x7Fp16ToFp16(), svuConv7x7Fp16ToU8(), svuConv9x1(), svuConv9x9(), svuConv9x9Fp16ToFp16(), svuConvGeneric(), svuConvSeparable11x11(), svuConvSeparable11x11Fp16ToFp16(), svuConvSeparable3x3(), svuConvSeparable3x3Fp16ToFp16(), svuConvSeparable5x5(), svuConvSeparable5x5Fp16ToFp16(), svuConvSeparable7x7(), svuConvSeparable7x7Fp16ToFp16(), svuConvSeparable9x9(), svuConvSeparable9x9Fp16ToFp16(), svuCvtColorChromaYUVToNV12(), svuCvtColorRGBtoChromaNV12(), svuCvtColorRGBtoLumaNV12(), svuDilate3x3(), svuDilate5x5(), svuDilate7x7(), svuDilateGeneric(), svudisp2depth(), svuEqualizeHist(), svuErode3x3(), svuErode5x5(), svuErode7x7(), svuExtAfStats(), svuExtStatsSatPixelsU32(), svuFast9M2(), svuFast9ScoreCv(), svuGenChroma(), svuGenChromaSS(), svuGenDnsRef(), svuGenDnsRefFp16(), svuGenLumaU8Fp16(), svuGenNoise(), svuGenNoiseFp16(), svuGreyDesat(), svuHammingDistance(), svuHarrisResponse(), svuHistogram(), svuHistogramStat(), svuHomography(),

svuInterpolatePixelBilinear(), svuLocalTM(), svuLowLvlCorr(), svuLowLvlCorrMultiplePlanes(), svuLut10to16(), svuLut10to8(), svuLut12to16(), svuLut12to8(), svuLut8to8(), svuLutP10BppU16inU8out(), svuMaxTest3x3_fp16(), svuMinMaxPos(), svuMinMaxValue(), svuMinTest3x3_fp16(), svuMixMedian(), svuMonoImbalance(), svuNonMax3x3Fp32(), svuPixelUnpacker(), svuPixelUnpackerMipi10b(), svuPixelUnpackerWB(), svuPositionKernel(), svuPurpleFlare(), svuStartBicubic(), svuStatsAwbSatPixels(), svuStatsAwbSatPixelsU32(), svuSubpixelFilter(), svuThreshold(), svuThresholdBinaryRange(), svuThresholdBinaryU8(), svuThresholdFilter(), svuUndistortBrown(), svuWhiteBalanceBayerGBRG(), svuWhiteBalanceRGB(), and svuXYgen().

u32 SippFilterS::parentOBufIdx[SIPP_FILTER_MAX_IBUFS]

Referenced by getIn3PlanePtr(), getInPtr(), getInPtrAbs(), getPlaneIoPtrs(), sippAsmOptSetup(), sippDbgDumpAsmOffsets(), sippFilterGetParentInputLines(), sippFilterGetParentPlaneStride(), sippIbufSetup(), sippLinkFilter(), sippLinkFilterSetOBuf(), and sippLoadSigma().

SippFilter* SippFilterS::parents[SIPP_FILTER_MAX_PARENTS]

Referenced by adjustNodesRecursively(), getIn3PlanePtr(), getInPtr(), getInPtrAbs(), getPlaneIoPtrs(), sippAsmOptSetup(), sippComputeChunkWidths(), sippDbgDumpAsmOffsets(), sippFilterGetParentInputLines(), sippFilterGetParentOutputHeight(), sippFilterGetParentOutputWidth(), sippFilterGetParentPlaneStride(), sippFilterGetParentSliceWidth(), sippFindConnectionsListRecursive(), sippGenericSchCreateSEFromFilter(), sippIbufSetup(), sippIdentifyOPipeSchedulingEntity(), sippInitDma(), sippInitMipiTx(), sippInitSigma(), sippLinkFilter(), sippLinkFilterSetOBuf(), sippLoadSigma(), sippOSEProcessFilterIBufs(), sippUpdateDmaAddr(), and sippUpdateDmaAddrCQ().

u32 SippFilterS::parentsKS[SIPP_FILTER_MAX_PARENTS]

Referenced by sippDbgDumpSchedForVcsCArr(), sippGenericSchCreateSEFromFilter(), sippIncrementOutBufs(), and sippOSEProcessFilterIBufs().

ParentInfo SippFilterS::parInfo[SIPP_FILTER_MAX_PARENTS]

Referenced by sippAsmOptSetup(), sippDbgDumpAsmOffsets(), and svuSc12xLancV().

u32 SippFilterS::planeStride[SIPP_FILTER_MAX_OBUFS]

Referenced by sippComputeBufferProps(), sippComputePaddingOffsets(), sippComputeSwOutCt(), sippDbgDumpAsmOffsets(), sippFilterAddOBuf(), sippFilterGetParentPlaneStride(), sippFilterGetPlaneStride(), sippIbufSetup(), sippInitDma(), sippOBufSetup(), and svuSc12xLancV().

pSippFilterSchedInfo SippFilterS::pOpipesch

pSippPipeline SippFilterS::pPipe

Referenced by sippAddFilterToPipe(), sippCheckOPipeConnectionChrDns(), sippCheckOPipeConnectionColourComb(), sippCheckOPipeConnectionDoGLTM(), sippCheckOPipeConnectionGenChroma(), sippCheckOPipeConnectionMedian(), sippCheckOPipeConnectionPoly(), sippFilterAddO-

Buf(), sippGenericSchCreateSEFromFilter(), sippIbufSetup(), sippInitDma(), sippLinkFilter(), sippObufSetup(), and sippOSEProcessFilterIBufs().

SippPixelChunkPos* SippFilterS::ptrChunkPos

Referenced by sippComputeBufferProps(), sippDbgDumpAsmOffsets(), sippGenericLinePrepare(), and sippGetChunkStartPixelPos().

SippSchEnt* SippFilterS::sch

Referenced by sippDbgDumpSchedForVcsCArr(), sippGenericSchCreateSEFromFilter(), sippGenericSchedAllocTempStorage(), sippGenericSchedWr(), sippOSEProcessFilterIBufs(), and sippOSEProcessFilterOBufs().

u32 SippFilterS::schNo

Referenced by sippGenericLinePrepare(), sippGenericRuntimeFrameReset(), and sippUpdateDmaAddrCQ().

u32 SippFilterS::sliceWidth

Referenced by adjustNodesRecursively(), sippComputeBufferProps(), sippComputeChunkWidths(), sippComputeChunkWidthsSW(), sippComputePaddingOffsets(), sippDbgDumpFilterOuts(), sippFilterGetParentSliceWidth(), sippFiltersResetSliceWidths(), sippIbufSetup(), sippInitDma(), sippObufSetup(), sippPrintSliceWidth(), subs05sync7(), svuAbsdiff(), svuAccumulateSquare(), svuAccumulateWeighted(), svuArithmeticAdd(), svuArithmeticAddmask(), svuArithmeticSub(), svuArithmeticSubFp16ToFp16(), svuArithmeticSubmask(), svuAvg(), svuBilateral5x5(), svuBitwiseAnd(), svubitwiseAndMask(), svuBitwiseNot(), svuBitwiseOr(), svuBitwiseOrMask(), svuBitwiseXor(), svuBitwiseXorMask(), svuBoxFilter(), svuBoxFilter11x11(), svuBoxFilter13x13(), svuBoxFilter15x15(), svuBoxFilter3x3(), svuBoxFilter5x5(), svuBoxFilter7x7(), svuBoxFilter9x9(), svuCannyEdgeDetection(), svuCensusMin16(), svuCensusMin64(), svuCensusMin65(), svuCensusMin7(), svuCensusTransform5x5(), svuChannelExtract(), svuChromaBlock(), svuCombDecimDemosaicAwbGains(), svuCombDecimDemosaicAwbGainsStats(), svuContrast(), svuConv11x11(), svuConv15x1(), svuConv1x15(), svuConv1x5(), svuConv1x5Fp16ToFp16(), svuConv1x7(), svuConv1x7Fp16ToFp16(), svuConv1x9(), svuConv3x3(), svuConv3x3Fp16ToFp16(), svuConv5x1(), svuConv5x1Fp16ToFp16(), svuConv5x5(), svuConv5x5Fp16ToFp16(), svuConv7x1(), svuConv7x1Fp16ToFp16(), svuConv7x7(), svuConv7x7Fp16ToFp16(), svuConv7x7Fp16ToU8(), svuConv9x1(), svuConv9x9(), svuConv9x9Fp16ToFp16(), svuConvert16bppTo8bpp(), svuConvertF16ToU8(), svuConvertFrom12BppTo8Bpp(), svuConvertPFp16ToU16(), svuConvertPU16Fp16(), svuConvertU8ToF16(), svuConvertYUV400ToYUV422(), svuConvGeneric(), svuConvSeparable11x11(), svuConvSeparable11x11Fp16ToFp16(), svuConvSeparable3x3(), svuConvSeparable3x3Fp16ToFp16(), svuConvSeparable5x5(), svuConvSeparable5x5Fp16ToFp16(), svuConvSeparable7x7(), svuConvSeparable7x7Fp16ToFp16(), svuConvSeparable9x9(), svuConvSeparable9x9Fp16ToFp16(), svuCopy(), svuCornerMinEigenVal(), svuCornerMinEigenValpatched(), svuCrop(), svuCropCvtPlaneMode(), svuCvtColorChromaYUVToNV12(), svucvtColorNV21toRGB(), svuCvtColorRGBfp16ToLumaU8(), svuCvtColorRGBfp16ToUV420U8(), svuCvtColorRGBtoChromaNV12(), svuCvtColorRGBtoLuma(), svuCvtColorRGBtoLumaNV12(), svuCvtColorRGBtoUV(), svuCvtColorRGBtoUV420(), svuCvtColorRGBToYUV422(), svuCvtColorYUV422ToRGB(), svuCvtColorYUVToRGB(), svuDilate3x3(), svuDilate5x5(), svuDilate7x7(), svuDilateGeneric(), svudisp2depth(), svuEqualizeHist(), svuErode3x3(), svuErode5x5(), svuErode7x7(), svuExtAfStats(), svu-

ExtStatsSatPixelsU32(), svuGauss(), svuGaussHx2(), svuGaussHx2_fp16(), svuGaussVx2(), svuGaussVx2_fp16(), svuGenChroma(), svuGenChromaSS(), svuGenDnsRef(), svuGenDnsRefFp16(), svuGenLuma(), svuGenLumaU8Fp16(), svuGenNoise(), svuGenNoiseFp16(), svuGreyDesat(), svuHammingDistance(), svuHarrisResponse(), svuHistogram(), svuHistogramStat(), svuHomography(), svuIntegralImageSqSumF32M2(), svuIntegralImageSqSumU32M2(), svuIntegralImageSumF32M2(), svuIntegralImageSumU16U32(), svuIntegralImageSumU32M2(), svuLaplacian3x3(), svuLaplacian5x5(), svuLaplacian5x5Fp16ToFp16(), svuLaplacian7x7(), svuLaplacian7x7Fp16ToFp16(), svuLocalMaxMin3x3_fp16(), svuLocalTM(), svuLowLvlCorr(), svuLowLvlCorrMultiplePlanes(), svuLumaBlur(), svuLut10to16(), svuLut10to8(), svuLut12to16(), svuLut12to8(), svuLut8to8(), svuLutP10BppU16inU8out(), svuMaxTest3x3_fp16(), svuMeanStdDev(), svuMinMaxPos(), svuMinMaxValue(), svuMinTest3x3_fp16(), svuMixMedian(), svuMonoImbalance(), svuNegative(), svuNonMax3x3Fp32(), svuNonMax3x3U8(), svuPadBayer5(), svuPadBayer5Frame(), svuPixelPacker10b(), svuPixelUnpacker(), svuPixelUnpackerMipi10b(), svuPixelUnpackerWB(), svuPositionKernel(), svuPurpleFlare(), svuPyrDown(), svuRgbyuv444(), svuSAD11x11(), svuSAD5x5(), svuScale05BilinHV_Fp16U8(), svuScale05BilinHVFp16(), svuScale2xBilinHV_025_075_Fp16ToFp16(), svuScale2xBilinHV_025_075_U16ToU16(), svuScale2xBilinHV_Fp16U8_phase025_075(), svuScale2xBilinHV_U8ToU8_phase025_075(), svuScharr_fp16(), svuScl05BilinHV(), svuScl05Lanc6(), svuScl2xBilinHV(), svuScl2xLancH(), svuScl2xLancV(), svuSclBilinArb(), svuSclLaplacian3x3Fp16ToFp16(), svuSobel(), svuSSD11x11(), svuSSD5x5(), svuSSD7x7U8ToU32(), svuSsdPointLine7x7U8U32(), svuSubpixelFilter(), svuThreshold(), svuThresholdBinaryRange(), svuThresholdBinaryU8(), svuThresholdFilter(), svuUndistortBrown(), svuWhiteBalanceBayerGBRG(), svuWhiteBalanceRGB(), svuXYgen(), and upscale2xV().

u32 SippFilterS::sliceWidthLastSvu

Referenced by sippComputeBufferProps(), sippComputePaddingOffsets(), svuExtAfStats(), and svuExtStatsSatPixelsU32().

u32 SippFilterS::unit

Referenced by adjustNodesRecursively(), sippAddFilterToPipe(), sippAllocCmxLineBuffers(), sippAllocCmxLineBuffersOPipe(), sippAnalysePipe2x5x(), sippBufGetObufCtx(), sippCheckOPipeConnectionChrDns(), sippCheckOPipeConnectionColourComb(), sippCheckOPipeConnectionDbyr(), sippCheckOPipeConnectionDoGLTM(), sippCheckOPipeConnectionGenChroma(), sippCheckOPipeConnectionLsc(), sippCheckOPipeConnectionLuma(), sippCheckOPipeConnectionLut(), sippCheckOPipeConnectionMedian(), sippCheckOPipeConnectionPoly(), sippCheckOPipeConnectionRaw(), sippCheckOPipeConnectionSharpen(), sippCheckOPipeConnectionSigma(), sippComputeChunkWidths(), sippComputeChunkWidthsSW(), sippCoreSetPaddingReqs(), sippDbgDumpFilterOuts(), sippFindConnectionsListRecursive(), sippGenericBlockHWUnits2x5x(), sippGenericLinePrepare(), sippGenericRuntimeLoadPipeline(), sippGenericSchCreateSEFromFilter(), sippGenericScheduleSetBufConsModels(), sippGenericSchedWr(), sippGetFirstHwFiltIdx(), sippIdentifyOPipeSchedulingEntity(), sippIniHwFilters(), sippLinkFilter(), sippLinkFilterSetObuf(), sippLoadMipiRx(), sippLoadMipiTx(), sippOSEProcessFilterIBufs(), sippOSEProcessFilterOBufs(), and sippValidatePipe().

6.146 SippFilterSchedInfoS Struct Reference

```
#include <sippSchTypes.h>
```

Data Fields

- [bool parentLatenciesKnown](#)
- [bool latenciesKnown](#)
- [u8 subSampleScale](#)
- [u8 nlinesUsedParent](#) [SIPP_SE_MAX_IBUFS]
- [u8 oBufLatency](#) [SIPP_FILTER_MAX_OBUFS]
- [bool hasBuffers](#)

6.146.1 Field Documentation

bool SippFilterSchedInfoS::hasBuffers

bool SippFilterSchedInfoS::latenciesKnown

u8 SippFilterSchedInfoS::nlinesUsedParent[SIPP_SE_MAX_IBUFS]

u8 SippFilterSchedInfoS::oBufLatency[SIPP_FILTER_MAX_OBUFS]

bool SippFilterSchedInfoS::parentLatenciesKnown

u8 SippFilterSchedInfoS::subSampleScale

6.147 SippGlobals Struct Reference

```
#include <sippTypesPrivate.h>
```

Data Fields

- [UnitInfo uInfo](#) [EXE_NUM]

6.147.1 Field Documentation

UnitInfo SippGlobals::uInfo[EXE_NUM]

Referenced by `sippAnalysePipe2x5x()`, `sippCoreUnitLoad()`, `sippCreateFilter()`, `sippFindConnections-ListRecursive()`, `sippGenericBlockHWUnits2x5x()`, `sippGenericRuntimeLoadPipeline()`, `sippIdentify-OPipeSchedulingEntity()`, `sippIniHwFilters()`, `sippOSEProcessFilterIBufs()`, `sippOSEProcessFilterO-Bufs()`, and `sippValidatePipe()`.

6.148 SippHeapCB Struct Reference

```
#include <sippTypesPrivate.h>
```

Data Fields

- [u32 used_size](#)

- `void * pNext`
- `u32 data []`

6.148.1 Field Documentation

`u32 SippHeapCB::data[]`

Referenced by `sippAlloc()`, and `sippFreeList()`.

`void* SippHeapCB::pNext`

Referenced by `sippAlloc()`, `sippFreeList()`, and `sippSizeList()`.

`u32 SippHeapCB::used_size`

Referenced by `sippFree()`, `sippHeapAlloc()`, `sippHeapCheck()`, `sippHeapDefrag()`, and `sippSizeList()`.

6.149 SippHwBufS Struct Reference

```
#include <sippTypesPrivate.h>
```

Data Fields

- `u32 base`
- `u32 cfg`
- `u32 ls`
- `u32 ps`
- `u32 irqRate`
- `u32 fillCtrl`
- `u32 ctx`

6.149.1 Field Documentation

`u32 SippHwBufS::base`

Referenced by `sippIbufSetup()`, and `sippObufSetup()`.

`u32 SippHwBufS::cfg`

Referenced by `sippIbufSetup()`, `sippLoadMipiRx()`, `sippLoadSigma()`, and `sippObufSetup()`.

`u32 SippHwBufS::ctx`

Referenced by `sippGenericRuntimeFrameReset()`, `sippIbufSetup()`, and `sippObufSetup()`.

u32 SippHwBufS::fillCtrl

u32 SippHwBufS::irqRate

Referenced by sippBufSetupIrqRate(), sippBufSetupIrqRateCQ(), sippIbufSetup(), and sippObufSetup().

u32 SippHwBufS::ls

Referenced by sippIbufSetup(), and sippObufSetup().

u32 SippHwBufS::ps

Referenced by sippIbufSetup(), and sippObufSetup().

6.150 SippHwIOBuf Struct Reference

```
#include <extAfStats.h>
```

Data Fields

- [UInt32 base](#)
- [UInt32 cfg](#)
- [UInt32 ls](#)
- [UInt32 ps](#)
- [UInt32 irqRate](#)
- [UInt32 fillCtrl](#)
- [UInt32 ctx](#)

6.150.1 Field Documentation

UInt32 SippHwIOBuf::base

Referenced by svuExtAfStats().

UInt32 SippHwIOBuf::cfg

Referenced by svuExtAfStats().

UInt32 SippHwIOBuf::ctx

UInt32 SippHwIOBuf::fillCtrl

Referenced by svuExtAfStats().

UInt32 SippHwIOBuf::irqRate

UInt32 SippHwIOBuf::ls

Referenced by svuExtAfStats().

UInt32 SippHwIOBuf::ps

6.151 SippManagedBufSchedInfo Struct Reference

```
#include <sippSchTypes.h>
```

Data Fields

- **u32** placeholder

6.151.1 Field Documentation

u32 SippManagedBufSchedInfo::placeholder

6.152 SippMemRegion Struct Reference

```
#include <sippTypes.h>
```

Data Fields

- **u32** regionOffset
- **u32** regionSize
- **u32** regionUsed

6.152.1 Field Documentation

u32 SippMemRegion::regionOffset

Referenced by sippGetNextMemRegion(), and sippMapRegionToCmx().

u32 SippMemRegion::regionSize

Referenced by sippAssignCmxMemRegion(), sippGetNextMemRegion(), and sippMapRegionToCmx().

u32 SippMemRegion::regionUsed

Referenced by sippGetNextMemRegion(), and sippMapRegionToCmx().

6.153 SippMemRegionListNode Struct Reference

```
#include <sippTypesPrivate.h>
```

Data Fields

- void * [pNext](#)
- void * [pNextChunkReg](#)
- u8 [chainStart](#)
- u8 [chainLinked](#)
- u32 [regionAddr](#)
- u32 [slice0Addr](#)
- u32 [regionSize](#)
- u32 [regionUsed](#)
- u32 * [regionUsedPtr](#)

6.153.1 Field Documentation

u8 SippMemRegionListNode::chainLinked

Referenced by [sippConfirmChunkChain\(\)](#), [sippMapRegionToCmx\(\)](#), and [sippMemLBMatchRegionsToChunks\(\)](#).

u8 SippMemRegionListNode::chainStart

Referenced by [sippConfirmChunkChain\(\)](#), [sippMapRegionToCmx\(\)](#), and [sippMemRegionAllocLineBuffer\(\)](#).

void* SippMemRegionListNode::pNext

Referenced by [sippAllocCmxLineBuffers\(\)](#), [sippAllocCmxLineBuffersOPipe\(\)](#), [sippMapRegionToCmx\(\)](#), [sippMemLBConsolidateRegions\(\)](#), [sippMemLBMatchRegionsToChunks\(\)](#), [sippMemLBRemoveNode\(\)](#), and [sippMemRegionAllocLineBuffer\(\)](#).

void* SippMemRegionListNode::pNextChunkReg

Referenced by [sippMapRegionToCmx\(\)](#), [sippMemAllocChainChunk\(\)](#), and [sippMemLBMatchRegionsToChunks\(\)](#).

u32 SippMemRegionListNode::regionAddr

Referenced by [sippAllocCmxLineBuffersOPipe\(\)](#), [sippMapRegionToCmx\(\)](#), [sippMemLBConsolidateRegions\(\)](#), and [sippMemLBMatchRegionsToChunks\(\)](#).

u32 SippMemRegionListNode::regionSize

Referenced by [sippAllocCmxLineBuffersOPipe\(\)](#), [sippMapRegionToCmx\(\)](#), [sippMemAllocChainChunk\(\)](#), [sippMemLBConsolidateRegions\(\)](#), and [sippMemRegionAllocLineBuffer\(\)](#).

u32 SippMemRegionListNode::regionUsed

Referenced by sippAllocCmxLineBuffers(), sippMapRegionToCmx(), sippMemAllocChainChunk(), and sippMemRegionAllocLineBuffer().

u32* SippMemRegionListNode::regionUsedPtr

Referenced by sippAllocCmxLineBuffers(), and sippMapRegionToCmx().

u32 SippMemRegionListNode::slice0Addr

Referenced by sippMapRegionToCmx(), sippMemAllocChainChunk(), and sippMemRegionAllocLineBuffer().

6.154 sippOpipeBufInfo Struct Reference

```
#include <sippTypesPrivate.h>
```

Data Fields

- [SippFilter](#) * ptrFilt
- [u32](#) oBufIdx

6.154.1 Field Documentation

u32 sippOpipeBufInfo::oBufIdx

Referenced by sippAllocCmxLineBuffersOPipe().

[SippFilter](#)* sippOpipeBufInfo::ptrFilt

Referenced by sippAllocCmxLineBuffersOPipe().

6.155 SippOseS Struct Reference

```
#include <sippTypesPrivate.h>
```

Data Fields

- [SippFilter](#) * filts [SIPP_MAX_FILTS_OSE]
- [psSchLineBuffer](#) oBufs [SIPP_SE_MAX_OBUFS]
- [psSchLineBuffer](#) iBufs [SIPP_SE_MAX_IBUFS]
- [SchedIBufUsageInfo](#) * iBufCtx [SIPP_SE_MAX_IBUFS]
- [u32](#) numOBufs
- [u32](#) numIBufs

- `u32 linesPerIter`
- `u32 * ptrFiltKS [SIPP_SE_MAX_IBUFS]`
- `SippSchEnt ** consSE [SIPP_MAX_FILTS_OSE]`
- `u32 consIbufIdx [SIPP_SE_MAX_CONSUMERS]`
- `u32 numConsumers`
- `u32 numFiltS`
- `u32 OSEFiltCfg`
- `u32 flags`
- `SippSchEnt * sch`

6.155.1 Field Documentation

`u32 SippOseS::consIbufIdx[SIPP_SE_MAX_CONSUMERS]`

`SippSchEnt** SippOseS::consSE[SIPP_MAX_FILTS_OSE]`

`SippFilter* SippOseS::filtS[SIPP_MAX_FILTS_OSE]`

`u32 SippOseS::flags`

`SchedIBufUsageInfo* SippOseS::iBufCtx[SIPP_SE_MAX_IBUFS]`

`psSchLineBuffer SippOseS::iBufs[SIPP_SE_MAX_IBUFS]`

`u32 SippOseS::linesPerIter`

`u32 SippOseS::numConsumers`

`u32 SippOseS::numFiltS`

`u32 SippOseS::numIBufs`

`u32 SippOseS::numOBufs`

`psSchLineBuffer SippOseS::oBufs[SIPP_SE_MAX_OBUFS]`

`u32 SippOseS::OSEFiltCfg`

`u32* SippOseS::ptrFiltKS[SIPP_SE_MAX_IBUFS]`

`SippSchEnt* SippOseS::sch`

6.156 SippPipelineS Struct Reference

```
#include <sippTypes.h>
```

Data Fields

- `CommInfo gi`
- `u32 oldRunMask`

- u32 canRunMask
- u32 nFilters
- u32 nPadFilters
- u32 svuWinRegs [4]
- u8 * mbinImg
- u32 nIter
- u32 linesPerIter
- u32 svuCmd
- u32 svuSyncMtx [2]
- u32 svuSyncSem
- u32 svuSyncMtxAddr
- u32 svuSyncMtxParity
- SippFilter * filters [SIPP_MAX_FILTERS_PER_PIPELINE]
- HPadInfo padList [SIPP_MAX_FILTERS_PER_PIPELINE]
- SippFilter * filtersHw [SIPP_MAX_FILTERS_PER_PIPELINE]
- u32 nFiltersHw
- SippFilter * filtersSvu [SIPP_MAX_FILTERS_PER_PIPELINE]
- u32 nFiltersSvu
- SippFilter * filtersDMA [SIPP_MAX_DMA_FILTERS_PER_PIPELINE]
- u32 nFiltersDMA
- SippFilter * firstRunFilts [SIPP_MAX_FILTERS_PER_PIPELINE]
- u32 nFirstRunFilts
- s32 hwSippFirst [EXE_NUM]
- u32 hwSippFltCnt [EXE_NUM]
- u32 shadowSelect
- u32 hwSippCtxSwMask
- SchedInfo * schedInfo
- u32 schedInfoEntries
- u32 * schedInfoCtx
- u32 schedInfoCtxSz
- u32 schedConsumeMask
- u32 sliceSz
- int dbgLevel
- u32 iteration
- u32 flags
- u32 svuStack
- u32 startDelta
- u32 ibflIncDelta
- u32 firstUseMask
- u32 firstIbflUseMask
- u32 firstUseInProg
- u32 nxtExeNo [SIPP_MAX_FILTERS_PER_PIPELINE]
- u32 multiHwCtx
- u8 * ddrCmxBackupAdr
- u32 ddrCmxBackupLen
- u64 dmaTaskList
- DmaDesc dmaCmxPush

- DmaDesc dmaCmxPop
- u64 pushCmd
- u64 popCmd
- sippEventCallback_t pfCallback
- u32 uHWFilterUsageBitMask [SIPP_HW_FILTER_MASK_SIZE]
- u32 uHWFilterIBufUsageMask
- u32 uHWFilterOBufUsageMask
- u8 useCmxRegMap
- u32 numMemRegions
- pSippCmxBufferMap pCmxMap
- u8 cmxMapResult
- SippSchEnt * pSE [SIPP_MAX_SE_PER_PIPE]
- u32 numSE
- u32 HWPipeID
- u32 itersLeft
- u32 endIter
- u32 isrFlags
- u32 useSyncRuntime
- u32 errorStatus [SIPP_ERROR_MASK_SIZE]
- u32 allDoneMask
- u32 setIterMask
- sippSchedFunc pfnSippSchedule
- sippSchedSetBufConsModels pfnSippScheSetBufConsModels
- sippRuntimeClaimHWResourceFunc pfnSippRuntimeClaimHWResource
- sippRuntimeFunc pfnSippRuntime
- sippRunIterDoneFunc pfnSippRunIterDone
- sippRunNextIterFunc pfnSippRunNextIter
- sippRunFrameReset pfnSippRunFrameReset
- sSippCdmaQu tCmxDmaQu [0x2]
- sSippCMDQu tCMDUpdateQu
- sSippCMDQu tCMDStartQu
- u32 runAddSchedCheck
- tSippMCB tHeapMCB
- u8 * lineMemPoolBase

6.156.1 Field Documentation

u32 SippPipelineS::allDoneMask

Referenced by sippCheckIterComplete().

u32 SippPipelineS::canRunMask

Referenced by sippGenericLinePrepare(), and sippGenericRuntimeProcessIters().

u8 SippPipelineS::cmxMapResult

Referenced by sippAllocCmxMemRegion().

int SippPipelineS::dbgLevel

Referenced by sippAllocCmxLineBuffersOPipe(), sippCoreFinalisePipeline(), sippCoreReschedulePipeline(), sippDbgCreateDumpFiles(), sippDbgDumpFilterOuts(), sippDbgLevel(), sippGenericLinePrepare(), sippGenericRunIterDone(), sippGenericWaitUnits(), sippIncrementOutBufs(), and sippInitPipeline().

u8* SippPipelineS::ddrCmxBackupAdr

Referenced by sippCoreFinalisePipeline().

u32 SippPipelineS::ddrCmxBackupLen

Referenced by sippCoreFinalisePipeline().

DmaDesc SippPipelineS::dmaCmxPop

Referenced by sippCoreFinalisePipeline().

DmaDesc SippPipelineS::dmaCmxPush

Referenced by sippCoreFinalisePipeline().

u64 SippPipelineS::dmaTaskList

Referenced by dmaKickSequence(), dmaKickSequenceCQ(), and sippInitDma().

u32 SippPipelineS::endIter

Referenced by sippGenericRunIterDone(), and sippGenericRuntimeProcessIters().

u32 SippPipelineS::errorStatus[SIPP_ERROR_MASK_SIZE]

Referenced by sippAddFilterToPipe(), sippCoreFinalisePipeline(), sippCoreReschedulePipeline(), sippCQInit(), sippCreateFilter(), sippDmaCQInit(), sippEventNotify(), sippGenericAllocRuntimeSched(), sippGenericSchedAllocTempStorage(), sippGenericSchedule(), sippIdentifyOPipeSchedulingEntity(), sippIssueCommand(), sippKickDmaCQ(), sippPipeGetErrorStatus(), sippPipeSessionControl(), and sippProcessFrame().

SippFilter* SippPipelineS::filters[SIPP_MAX_FILTERS_PER_PIPELINE]

Referenced by sippAllocCmxLineBuffers(), sippAllocCmxLineBuffersOPipe(), sippAnalysePipe2x5x(), sippBuildLnBufs(), sippComputeBufferProps(), sippComputeChunkWidths(), sippComputeChunkWidthsSW(), sippComputeSwOutCt(), sippCoreReInitLineBuffers(), sippCoreSetPaddingReqs(), sippCoreUnitLoad(), sippCreateFilter(), sippDbgCreateDumpFiles(), sippDbgDumpAsmOffsets(), sippDbgDumpFilterOuts(), sippDbgDumpGraph(), sippDbgDumpSchedForVcsCarr(), sippDumpHtmlMap(), sippFiltersResetSliceWidths(), sippGenericBlockHWUnits2x5x(), sippGenericLinePrepare(), sippGenericRuntimeFrameReset(), sippGenericRuntimeLoadPipeline(), sippGenericSchedAllocTempStorage(), sippGenericSchedInitTempStorage(), sippGenericSchedPipeInit(), sippGenericScheduleSetBufConsModels(), sippGenericSchedWr(), sippGenericStartUnits(), sippGenericUpdateExecNums(), sippGetFirstHwFiltIdx(), sippIdentifyOPipeSchedulingEntity(), sippIncrementOutBufs(), sippIniHwFilters(), sippInitBufferLnPointers(), sippPrintSliceWidth(), sippProcessSchedData(), sippResetFilterVariables(), sippUsingPrecompSched(), and sippValidatePipe().

SippFilter* SippPipelineS::filtersDMA[SIPP_MAX_DMA_FILTERS_PER_PIPELINE]

Referenced by sippAddFilterToPipe(), sippChainDmaDesc(), sippCoreReInitLineBuffers(), sippGenericSchedWr(), sippInitDma(), sippKickDma(), and sippKickDmaCQ().

SippFilter* SippPipelineS::filtersHw[SIPP_MAX_FILTERS_PER_PIPELINE]

SippFilter* SippPipelineS::filtersSvu[SIPP_MAX_FILTERS_PER_PIPELINE]

Referenced by SHAVE_MAIN(), sippAddFilterToPipe(), sippAsmOptSetup(), sippDbgDumpAsmOffsets(), and sippGenericSchedWr().

u32 SippPipelineS::firstIbflUseMask

Referenced by sippProcessFrame(), and sippProcessFrameNB().

SippFilter* SippPipelineS::firstRunFilts[SIPP_MAX_FILTERS_PER_PIPELINE]

u32 SippPipelineS::firstUseInProg

u32 SippPipelineS::firstUseMask

Referenced by sippProcessFrame(), and sippProcessFrameNB().

u32 SippPipelineS::flags

Referenced by dmaKickSequenceConcurrent(), sippBuildLnBufs(), sippCheckIterComplete(), sippCmxDmaDoneIrqHandler(), sippComputeSliceLayout(), sippCoreFinalisePipeline(), sippCoreReschedulePipeline(), sippCQInit(), sippFinalizePipeline(), sippGenericLinePrepare(), sippGenericStartUnits(), sippGenericWaitUnits(), sippIdentifyOPipeSchedulingEntity(), sippProcessFrame(), sippProcessFrameNB(), and sippSetupSvus().

CommInfo SippPipelineS::gi

Referenced by sippAddFilterToPipe(), sippAllocCmxLineBuffers(), sippComputeBufferProps(), sippComputeChunkWidths(), sippComputeChunkWidthsSW(), sippComputePaddingOffsets(), sippComputeSwOutCt(), sippCoreFinalisePipeline(), sippDbgDumpAsmOffsets(), sippDbgDumpFilterOuts(), sippDumpHtmlMap(), sippGenericRuntimeProcessIters(), sippGenericStartUnits(), sippHorizontalPadding(), sippHWSessionAddActiveLists(), sippHWSessionCommand(), sippHWSessionRemoveActiveLists(), sippInitDma(), sippInitPipeline(), sippKickShaveM1PC(), sippKickSvus(), sippMapRegionMapAddrToSliceZero(), sippSetupSvus(), sippStopSvus(), sippValidatePipe(), and sippWaitShave().

u32 SippPipelineS::HWPipeID

Referenced by sippGenericRuntimeLoadPipeline(), sippGenericStartHWUnits2x5x(), and sippHWSessionAddActiveLists().

u32 SippPipelineS::hwSippCtxSwMask

s32 SippPipelineS::hwSippFirst[EXE_NUM]

Referenced by sippCoreUnitLoad(), sippGetFirstHwFiltIdx(), and sippInitPipeline().

u32 SippPipelineS::hwSippFltCnt[EXE_NUM]

Referenced by sippAddFilterToPipe(), sippCoreHwInitialLoad(), sippGetFirstHwFiltIdx(), and sippValidatePipe().

u32 SippPipelineS::ibflIncDelta

Referenced by sippAnalysePipe2x5x().

u32 SippPipelineS::isrFlags

Referenced by sippCheckIterComplete(), and sippGenericRunIterDone().

u32 SippPipelineS::iteration

Referenced by dmaKickSequence(), SHAVE_MAIN(), sippCheckIterComplete(), sippCmxDmaDoneIrqHandler(), sippDbgDumpAsmOffsets(), sippDbgDumpFilterOuts(), sippGenericBlockHWUnits2x5x(), sippGenericRunIterDone(), sippGenericRunNextIter(), sippGenericRuntimeHWProcessIters(), sippGenericRuntimeProcessIters(), sippGenericStartHWUnits2x5x(), sippGenericStartUnits(), sippGenericUpdateHWUnits2x5x(), sippGenericWaitUnits(), sippKickDma(), sippObflIncHandler(), sippProcessFrame(), sippProcessFrameNB(), sippRunDmaCQDrain(), and sippSvuDoneIrqHandler().

u32 SippPipelineS::itersLeft

Referenced by sippGenericRuntimeProcessIters().

u8* SippPipelineS::lineMemPoolBase

Referenced by sippComputeSliceLayout(), and sippCoreReschedulePipeline().

u32 SippPipelineS::linesPerIter

Referenced by sippAnalysePipe2x5x(), sippCQInit(), sippDbgDumpSchedForVcsCArr(), sippDmaCQInit(), sippInitPipeline(), and sippPipeSetLinesPerIter().

u8* SippPipelineS::mbinImg

Referenced by sippComputeSliceLayout(), sippCoreFinalisePipeline(), sippInitPipeline(), and sippSetupSvus().

u32 SippPipelineS::multiHwCtx

Referenced by sippGetCtxOrder().

u32 SippPipelineS::nFilters

Referenced by sippAddFilterToPipe(), sippAllocCmxLineBuffers(), sippAllocCmxLineBuffersOPipe(), sippAnalysePipe2x5x(), sippBuildLnBufs(), sippComputeBufferProps(), sippComputeChunkWidths(), sippComputeChunkWidthsSW(), sippComputeSwOutCt(), sippCoreReInitLineBuffers(), sippCoreSetPaddingReqs(), sippCQInit(), sippCreateFilter(), sippDbgCreateDumpFiles(), sippDbgDumpFilterOuts(), sippDbgDumpGraph(), sippDbgDumpSchedForVcsCArr(), sippDumpHtmlMap(), sippFiltersResetSliceWidths(), sippGenericAllocRuntimeSched(), sippGenericBlockHWUnits2x5x(), sippGenericLinePrepare(), sippGenericRuntimeFrameReset(), sippGenericRuntimeLoadPipeline(), sippGenericSchedAllocTempStorage(), sippGenericSchedInitTempStorage(), sippGenericSchedPipeInit(), sippGenericScheduleSetBufConsModels(), sippGenericSchedWr(), sippGenericStartUnits(), sippGenericUpdateExecNums(), sippGetFirstHwFiltIdx(), sippIdentifyOPipeSchedulingEntity(), sippIncrementOutBufs(), sippIniHwFilters(), sippInitBufferLnPointers(), sippPrintSliceWidth(), sippProcessSchedData(), sippResetFilterVariables(), sippUsingPrecompSched(), and sippValidatePipe().

u32 SippPipelineS::nFiltersDMA

Referenced by sippAddFilterToPipe(), sippChainDmaDesc(), sippCoreReInitLineBuffers(), sippCQInit(), sippDmaCQInit(), sippGenericSchedWr(), sippInitDma(), sippKickDma(), and sippKickDmaCQ().

u32 SippPipelineS::nFiltersHw

u32 SippPipelineS::nFiltersSvu

Referenced by SHAVE_MAIN(), sippAddFilterToPipe(), sippAsmOptSetup(), sippCoreFinalisePipeline(), sippCQInit(), sippDbgDumpAsmOffsets(), sippGenericSchedWr(), sippStopSvus(), and sippValidatePipe().

u32 SippPipelineS::nFirstRunFiltS

u32 SippPipelineS::nIter

Referenced by sippCheckIterComplete(), sippDbgDumpSchedForVcsCArr(), sippGenericLinePrepare(), sippGenericRunIterDone(), sippGenericRuntimeProcessIters(), sippGenericSchedPipeInit(), sippGenericSchedule(), sippGenericSchedWr(), sippGetFirstHwFiltIdx(), sippProcessFrame(), and sippProcessFrameNB().

u32 SippPipelineS::nPadFilters

Referenced by sippBuildLnBufs(), sippComputePaddingOffsets(), sippCoreReschedulePipeline(), sippDbgDumpAsmOffsets(), and sippHorizontalPadding().

u32 SippPipelineS::numMemRegions

Referenced by sippGetNextMemRegion().

u32 SippPipelineS::numSE

Referenced by sippCoreReschedulePipeline(), sippGenericAllocRuntimeSched(), and sippGenericSchedAllocTempStorage().

u32 SippPipelineS::nxtExeNo[SIPP_MAX_FILTERS_PER_PIPELINE]

Referenced by sippGenericStartUnits(), and sippGenericUpdateExecNums().

u32 SippPipelineS::oldRunMask

Referenced by sippGenericLinePrepare().

HPadInfo SippPipelineS::padList[SIPP_MAX_FILTERS_PER_PIPELINE]

Referenced by sippBuildLnBufs(), sippComputePaddingOffsets(), sippDbgDumpAsmOffsets(), and sippHorizontalPadding().

pSippCmxBufferMap SippPipelineS::pCmxMap

Referenced by sippAllocCmxLineBuffers(), sippAllocCmxLineBuffersOPipe(), sippAssignCmxMemRegion(), sippCoreFinalisePipeline(), sippInitPipeline(), and sippMapRegionToCmx().

sippEventCallback_t SippPipelineS::pfCallback

Referenced by sippEventNotify(), and sippRegisterEventCallback().

sippRunFrameReset SippPipelineS::pfnSippRunFrameReset

sippRunIterDoneFunc SippPipelineS::pfnSippRunIterDone

Referenced by sippCheckIterComplete().

sippRunNextIterFunc SippPipelineS::pfnSippRunNextIter

Referenced by sippCheckIterComplete().

sippRuntimeFunc SippPipelineS::pfnSippRuntime

Referenced by sippAnalysePipe2x5x(), sippCheckOPipeConnectionChrDns(), sippCheckOPipeConnectionColourComb(), sippCheckOPipeConnectionDoGLTM(), sippCheckOPipeConnectionGenChroma(), sippCheckOPipeConnectionMedian(), sippCheckOPipeConnectionPoly(), sippCoreFinalisePipeline(), sippCoreUnitLoad(), sippGetFirstHwFiltIdx(), sippHWSessionCommand(), sippIbufSetup(), sippObufSetup(), and sippOSEProcessFilterIBufs().

sippRuntimeClaimHWResourceFunc SippPipelineS::pfnSippRuntimeClaimHWResource

sippSchedFunc SippPipelineS::pfnSippSchedule

Referenced by sippCoreFinalisePipeline(), sippCoreReschedulePipeline(), sippIbflDecHandler(), and sippObflIncHandler().

sippSchedSetBufConsModels SippPipelineS::pfnSippScheSetBufConsModels

u64 SippPipelineS::popCmd

Referenced by sippCoreFinalisePipeline().

SippSchEnt* SippPipelineS::pSE[SIPP_MAX_SE_PER_PIPE]

Referenced by sippGenericAllocRuntimeSched(), and sippGenericSchedAllocTempStorage().

u64 SippPipelineS::pushCmd

Referenced by sippCoreFinalisePipeline().

u32 SippPipelineS::runAddSchedCheck

Referenced by sippGenericColorCombChromaCheck(), sippGenericDMACheck(), and sippGenericSchCreateSEFromFilter().

u32 SippPipelineS::schedConsumeMask

Referenced by sippGenericSchedWr().

SchedInfo* SippPipelineS::schedInfo

Referenced by dmaKickSequence(), dmaKickSequenceCQ(), SHAVE_MAIN(), sippCheckIterComplete(), sippDbgDumpAsmOffsets(), sippDbgDumpFilterOuts(), sippDbgDumpSchedForVcsC-
Arr(), sippGenericAllocRuntimeSched(), sippGenericBlockHWUnits2x5x(), sippGenericLinePrepare(),
sippGenericRunNextIter(), sippGenericSchedWr(), sippGenericStartHWUnits2x5x(), sippGenericStart-
Units(), sippGenericUpdateHWUnits2x5x(), sippGenericWaitUnits(), sippGetFirstHwFiltIdx(), sipp-
KickDma(), sippKickDmaCQ(), and sippObflIncHandler().

u32* SippPipelineS::schedInfoCtx

u32 SippPipelineS::schedInfoCtxSz

u32 SippPipelineS::schedInfoEntries

Referenced by sippCoreReschedulePipeline(), and sippGenericAllocRuntimeSched().

u32 SippPipelineS::setIterMask

Referenced by sippCheckIterComplete(), and sippInitPipeline().

u32 SippPipelineS::shadowSelect

u32 SippPipelineS::sliceSz

Referenced by sippCoreFinalisePipeline(), sippInitDma(), and sippInitPipeline().

u32 SippPipelineS::startDelta

Referenced by sippAnalysePipe2x5x().

u32 SippPipelineS::svuCmd

Referenced by SHAVE_MAIN(), sippDbgDumpAsmOffsets(), sippHorizontalPadding(), sippInit-
Pipeline(), and sippKickShaveM1PC().

u32 SippPipelineS::svuStack

Referenced by sippComputeSliceLayout(), sippKickShaveM1PC(), and sippKickSvus().

u32 SippPipelineS::svuSyncMtx[2]

Referenced by SHAVE_MAIN(), sippDbgDumpAsmOffsets(), sippGenericStartUnits(), sippInit-
Pipeline(), sippInitSyncMutexes(), and sippStopSvus().

u32 SippPipelineS::svuSyncMtxAddr

Referenced by sippCoreFinalisePipeline(), sippGenericStartUnits(), and sippStopSvus().

u32 SippPipelineS::svuSyncMtxParity

Referenced by sippGenericStartUnits(), and sippInitSyncMutexes().

u32 SippPipelineS::svuSyncSem

Referenced by sippDbgDumpAsmOffsets(), sippGenericStartUnits(), and sippGenericWaitUnits().

u32 SippPipelineS::svuWinRegs[4]

Referenced by sippComputeSliceLayout(), sippCoreFinalisePipeline(), and sippSetupSvus().

sSippCMDQu SippPipelineS::tCMDStartQu

Referenced by sippCQInit().

sSippCMDQu SippPipelineS::tCMDUpdateQu

Referenced by sippCQInit().

sSippCdmaQu SippPipelineS::tCmxDmaQu[0x2]

Referenced by sippDmaCQInit(), sippKickDmaCQ(), and sippRunDmaCQDrain().

tSippMCB SippPipelineS::tHeapMCB

Referenced by sippAllocCmxLineBuffers(), sippAllocCmxLineBuffersOPipe(), sippAssignCmxMemRegion(), sippComputeBufferProps(), sippComputePaddingOffsets(), sippComputeSliceLayout(), sippComputeSwOutCt(), sippCoreFinalisePipeline(), sippCoreReschedulePipeline(), sippCQInit(), sippCreateFilter(), sippDmaCQInit(), sippFilterAddOBuf(), sippFreePipeResource(), sippGenericAllocRuntimeSched(), sippGenericSchedAllocTempStorage(), sippGenericSchedule(), sippIbufSetup(), sippInitDma(), sippInitPipeline(), sippLinkFilter(), sippMapRegionToCmx(), sippObufSetup(), and sippOSECreate().

u32 SippPipelineS::uHWFilterIBufUsageMask

Referenced by sippHWSessionRemoveActiveLists(), and sippValidatePipe().

u32 SippPipelineS::uHWFilterOBufUsageMask

Referenced by sippHWSessionRemoveActiveLists(), and sippValidatePipe().

u32 SippPipelineS::uHWFilterUsageBitMask[SIPP_HW_FILTER_MASK_SIZE]

Referenced by sippHWSessionAddActiveLists(), sippHWSessionRemoveActiveLists(), and sippValidatePipe().

u8 SippPipelineS::useCmxRegMap

Referenced by sippAllocCmxMemRegion().

u32 SippPipelineS::useSyncRuntime

Referenced by sippGenericRuntimeClaimHWResource(), sippGenericRuntimeLoadPipeline(), sippGenericRuntimeProcessIters(), sippGenericStartHWUnits2x5x(), sippInitDma(), sippKickDma(), sippKickSvus(), sippObflIncHandler(), sippProcessFrame(), sippProcessFrameNB(), and sippRunDmaCQ().

6.157 SippPixelChunkPos Struct Reference

```
#include <sippTypes.h>
```

Data Fields

- [UInt32 YPos](#) [2]
- [UInt32 XPos](#) [12]

6.157.1 Field Documentation

UInt32 SippPixelChunkPos::XPos[12]

Referenced by sippComputeBufferProps(), and sippGetChunkStartPixelPos().

UInt32 SippPixelChunkPos::YPos[2]

Referenced by sippGenericLinePrepare(), and sippGetChunkStartPixelPos().

6.158 SippSchEntS Struct Reference

```
#include <sippSchTypes.h>
```

Data Fields

- [psSchLineBuffer oBufs](#) [SIPP_SE_MAX_OBUFS]
- [psSchLineBuffer iBufs](#) [SIPP_SE_MAX_IBUFS]
- [SchedIBufUsageInfo * iBufCtx](#) [SIPP_SE_MAX_IBUFS]
- [u32 numOBufs](#)
- [u32 numIBufs](#)
- [SippSchEnt * cons](#) [SIPP_SE_MAX_CONSUMERS]
- [u32 consIbufIdx](#) [SIPP_SE_MAX_CONSUMERS]
- [u32 numConsumers](#)
- [u32 maxLineRequired](#) [SIPP_SE_MAX_IBUFS]
- [u32 minLineRequired](#) [SIPP_SE_MAX_IBUFS]

- `u8 runPadCheck [SIPP_SE_MAX_IBUFS]`
- `s32 parentKS [SIPP_SE_MAX_IBUFS]`
- `u32 * ptrFiltKS [SIPP_SE_MAX_IBUFS]`
- `s32 parentKSMin [SIPP_SE_MAX_IBUFS]`
- `RunStatus canRunP`
- `RunStatus canRunC`
- `RunStatus canConsume`
- `s32 dbgJustRoll`
- `s32 numLineRuns`
- `u32 maxOutputH`
- `u32 linesPerIter`
- `u8 recordParentKS [SIPP_SE_MAX_IBUFS]`
- `s16 * currKSOOffset [SIPP_SE_MAX_IBUFS]`
- `u16 * KSIterList [SIPP_SE_MAX_IBUFS]`
- `SchedAddCheck pfRunAdditionalCheck`
- `void * pRunAdditionalParam`
- `u8 pipeSEId`

6.158.1 Field Documentation

`RunStatus SippSchEntS::canConsume`

Referenced by `sippGenericSchCreateSEFromFilter()`, `sippGenericSchCreateSEFromOSE()`, and `sippGenericSchedWr()`.

`RunStatus SippSchEntS::canRunC`

Referenced by `sippGenericDbgPrintRunnable()`, `sippGenericDMACheck()`, `sippGenericSchCreateSEFromFilter()`, `sippGenericSchCreateSEFromOSE()`, and `sippGenericSchedWr()`.

`RunStatus SippSchEntS::canRunP`

Referenced by `sippGenericColorCombChromaCheck()`, `sippGenericDbgPrintRunnable()`, `sippGenericDMACheck()`, `sippGenericSchCreateSEFromFilter()`, `sippGenericSchCreateSEFromOSE()`, and `sippGenericSchedWr()`.

`SippSchEnt* SippSchEntS::cons[SIPP_SE_MAX_CONSUMERS]`

Referenced by `sippGenericDMACheck()`, `sippGenericSchCreateSEFromFilter()`, and `sippGenericSchCreateSEFromOSE()`.

`u32 SippSchEntS::consIbufIdx[SIPP_SE_MAX_CONSUMERS]`

Referenced by `sippGenericDMACheck()`, `sippGenericSchCreateSEFromFilter()`, `sippGenericSchCreateSEFromOSE()`, and `sippOSEProcessFilterOBufs()`.

s16* SippSchEntS::currKsoffset[SIPP_SE_MAX_IBUFS]

s32 SippSchEntS::dbgJustRoll

Referenced by sippGenericCheckSERunParents(), and sippGenericDbgPrintRunnable().

SchedIBufUsageInfo* SippSchEntS::iBufCtx[SIPP_SE_MAX_IBUFS]

Referenced by askChromaGenDownsizer(), askCrop(), askCropLatency(), askHwColorCombChroma(), askHwMedLumaLatency(), askPolyFirResizer(), askRegular(), askRegularLatency(), askResizer(), askResizerLatency(), sippGenericCheckSERunParents(), sippGenericDMACheck(), sippGenericRecordParentKS(), sippGenericSchCreateSEFromFilter(), and sippGenericSchCreateSEFromOSE().

psSchLineBuffer SippSchEntS::iBufs[SIPP_SE_MAX_IBUFS]

Referenced by askChromaGenDownsizer(), askCrop(), askCropLatency(), askHwColorCombChroma(), askHwMedLumaLatency(), askPolyFirResizer(), askRegular(), askRegularLatency(), askResizer(), askResizerLatency(), sippGenericCheckSERunParents(), sippGenericRecordParentKS(), sippGenericSchCreateSEFromFilter(), and sippGenericSchCreateSEFromOSE().

u16* SippSchEntS::KSIterList[SIPP_SE_MAX_IBUFS]

u32 SippSchEntS::linesPerIter

u32 SippSchEntS::maxLineRequired[SIPP_SE_MAX_IBUFS]

Referenced by askChromaGenDownsizer(), askCrop(), askCropLatency(), askHwColorCombChroma(), askHwMedLumaLatency(), askPolyFirResizer(), askRegular(), askRegularLatency(), askResizer(), askResizerLatency(), sippGenericCheckSERunParents(), and sippGenericRecordParentKS().

u32 SippSchEntS::maxOutputH

Referenced by sippDbgDumpSchedForVcsCarr(), sippGenericAllocRuntimeSched(), sippGenericCheckSERunParents(), sippGenericSchCreateSEFromFilter(), sippGenericSchCreateSEFromOSE(), and sippGenericSchedWr().

u32 SippSchEntS::minLineRequired[SIPP_SE_MAX_IBUFS]

Referenced by askChromaGenDownsizer(), askCrop(), askCropLatency(), askHwColorCombChroma(), askHwMedLumaLatency(), askPolyFirResizer(), askRegular(), askRegularLatency(), askResizer(), askResizerLatency(), and sippGenericCheckSERunParents().

u32 SippSchEntS::numConsumers

Referenced by sippGenericDMACheck(), sippGenericSchCreateSEFromFilter(), and sippGenericSchCreateSEFromOSE().

u32 SippSchEntS::numIBufs

Referenced by sippGenericCheckSERunParents(), sippGenericColorCombChromaCheck(), sippGenericDbgPrintRunnable(), sippGenericDbgShowBufferReq(), sippGenericSchCreateSEFromFilter(), and sippGenericSchCreateSEFromOSE().

s32 SippSchEntS::numLineRuns

Referenced by sippGenericCheckSERunParents(), sippGenericSchCreateSEFromFilter(), sippGenericSchCreateSEFromOSE(), and sippGenericSchedWr().

u32 SippSchEntS::numOBufs

Referenced by sippGenericDbgShowBufferReq(), sippGenericSchCreateSEFromFilter(), and sippGenericSchCreateSEFromOSE().

psSchLineBuffer SippSchEntS::oBufs[SIPP_SE_MAX_OBUFS]

Referenced by sippGenericDMACheck(), sippGenericSchCreateSEFromFilter(), and sippGenericSchCreateSEFromOSE().

s32 SippSchEntS::parentKS[SIPP_SE_MAX_IBUFS]

Referenced by sippGenericCheckSERunParents(), sippGenericColorCombChromaCheck(), sippGenericDbgPrintRunnable(), sippGenericDMACheck(), sippGenericRecordParentKS(), sippGenericSchCreateSEFromFilter(), and sippGenericSchCreateSEFromOSE().

s32 SippSchEntS::parentKSMIn[SIPP_SE_MAX_IBUFS]

Referenced by sippGenericColorCombChromaCheck(), sippGenericRecordParentKS(), sippGenericSchCreateSEFromFilter(), and sippGenericSchCreateSEFromOSE().

SchedAddCheck SippSchEntS::pfRunAdditionalCheck

Referenced by sippGenericSchCreateSEFromFilter(), and sippGenericSchCreateSEFromOSE().

u8 SippSchEntS::pipeSEId

Referenced by sippGenericColorCombChromaCheck(), sippGenericDMACheck(), sippGenericSchCreateSEFromFilter(), sippGenericSchCreateSEFromOSE(), and sippGenericSchedAllocTempStorage().

void* SippSchEntS::pRunAdditionalParam

Referenced by sippGenericSchCreateSEFromFilter(), and sippGenericSchCreateSEFromOSE().

u32* SippSchEntS::ptrFiltKS[SIPP_SE_MAX_IBUFS]

Referenced by sippGenericSchCreateSEFromFilter(), and sippGenericSchCreateSEFromOSE().

u8 SippSchEntS::recordParentKS[SIPP_SE_MAX_IBUFS]

Referenced by sippGenericCheckSERunParents(), sippGenericColorCombChromaCheck(), sippGenericRecordParentKS(), sippGenericSchCreateSEFromFilter(), and sippGenericSchCreateSEFromOSE().

u8 SippSchEntS::runPadCheck[SIPP_SE_MAX_IBUFS]

Referenced by askChromaGenDownsizer(), askPolyFirResizer(), askResizer(), askResizerLatency(), sippGenericCheckSERunParents(), and sippGenericSchCreateSEFromFilter().

6.159 SippVPhysMapS Struct Reference

```
#include <sippTypesPrivate.h>
```

Data Fields

- **ptSippPhysicalPool** physPoolMap [vPoolLast]

6.159.1 Field Documentation

ptSippPhysicalPool SippVPhysMapS::physPoolMap[vPoolLast]

Referenced by sippChooseMemPool(), sippInitLnMemPool(), sippInitLnMemPoolSlices(), sippInitPhysicalPoolPipe(), sippInitSchedPool(), sippMemAlloc(), sippMemCheck(), sippMemFindMaxLnMemPoolFree(), sippMemFree(), sippMemFreeList(), and sippMemInitVirtPhysMaps().

6.160 sSchedIBufUsageInfoS Struct Reference

```
#include <sippTypesPrivate.h>
```

Data Fields

- **u32** consumptionLatency
- **u32** runFullSearch
- **SchedFuncAsk** funcAsk
- **u32** lastAskLineNo
- **u8** numLinesUsed
- **u8** hKerSz
- **u8** iPadLines
- **u32** outputHeight
- **void *** params
- **u8** KSDelta

6.160.1 Field Documentation

u32 sSchedIBufUsageInfoS::consumptionLatency

Referenced by askCropLatency(), askHwMedLumaLatency(), askRegular(), askRegularLatency(), askResizerLatency(), sippLinkFilter(), sippOSEProcessFilterIBufs(), sippSetBufLatenciesMipiTx(), and sippSetBufLatenciesSigma().

SchedFuncAsk sSchedIBufUsageInfoS::funcAsk

Referenced by sippGenericCheckSERunParents(), sippGenericSchCreateSEFromFilter(), and sippGenericScheduleSetBufConsModels().

u8 sSchedIBufUsageInfoS::hKerSz

Referenced by sippCoreSetPaddingReqs(), and sippLinkFilter().

u8 sSchedIBufUsageInfoS::iPadLines

Referenced by sippLinkFilter().

u8 sSchedIBufUsageInfoS::KSDelta

Referenced by sippLinkFilter().

u32 sSchedIBufUsageInfoS::lastAskLineNo

Referenced by askChromaGenDownsizer(), askCrop(), askCropLatency(), askHwColorCombChroma(), askHwMedLumaLatency(), askPolyFirResizer(), askRegular(), askRegularLatency(), askResizer(), askResizerLatency(), sippCoreReInitLineBuffers(), and sippLinkFilter().

u8 sSchedIBufUsageInfoS::numLinesUsed

Referenced by askChromaGenDownsizer(), askHwMedLumaLatency(), askPolyFirResizer(), askRegular(), askRegularLatency(), askResizer(), askResizerLatency(), sippAnalysePipe2x5x(), sippGenericCheckSERunParents(), sippGenericDMACheck(), sippGenericScheduleSetBufConsModels(), sippLinkFilter(), and sippOSEProcessFilterIBufs().

u32 sSchedIBufUsageInfoS::outputHeight

Referenced by askChromaGenDownsizer(), askResizer(), askResizerLatency(), and sippLinkFilter().

void* sSchedIBufUsageInfoS::params

Referenced by askCrop(), askCropLatency(), askPolyFirResizer(), and sippLinkFilter().

u32 sSchIBufUsageInfoS::runFullSearch

Referenced by sippCoreReInitLineBuffers(), sippGenericCheckSERunParents(), sippGenericRecordParentKS(), and sippLinkFilter().

6.161 sSchLineBufferS Struct Reference

```
#include <sippSchTypes.h>
```

Data Fields

- u8 latency
- u16 internalFillLevel
- s32 mostRecentLine
- u32 filtOutHeight
- u32 bottomLineReplication
- u32 numLines
- u8 allocReq
- u8 manageReq
- u8 numSWConsumers
- u8 hPadding
- u32 hwOutputBufId
- u32 hwInputBufId

6.161.1 Field Documentation

u8 sSchLineBufferS::allocReq

Referenced by sippAnalysePipe2x5x(), sippCheckOPipeConnectionColourComb(), sippCheckOPipeConnectionDoGLTM(), sippCheckOPipeConnectionMedian(), sippCheckOPipeConnectionPoly(), sippGenericSchedWr(), and sippProcessSchedData().

u32 sSchLineBufferS::bottomLineReplication

Referenced by sippCoreReInitLineBuffers(), sippFilterAddOBuf(), sippGenericGetBufferEntry(), and sippGenericSearchBuffer().

u32 sSchLineBufferS::filtOutHeight

Referenced by askChromaGenDownsizer(), askCrop(), askCropLatency(), askHwColorCombChroma(), askHwMedLumaLatency(), askPolyFirResizer(), askRegular(), askRegularLatency(), askResizer(), askResizerLatency(), sippAddFilterToPipe(), sippCoreReInitLineBuffers(), sippFilterAddOBuf(), sippGenericSchCreateSEFromFilter(), and sippGenericSchCreateSEFromOSE().

u8 sSchLineBufferS::hPadding

Referenced by sippCoreSetPaddingReqs(), and sippProcessSchedData().

u32 sSchLineBufferS::hwInputBufId

Referenced by sippAnalysePipe2x5x(), sippFilterAddOBuf(), and sippGenericSchedWr().

u32 sSchLineBufferS::hwOutputBufId

Referenced by sippAnalysePipe2x5x(), sippFilterAddOBuf(), and sippGenericSchedWr().

u16 sSchLineBufferS::internalFillLevel

Referenced by sippCoreReInitLineBuffers().

u8 sSchLineBufferS::latency

Referenced by sippGenericSchCreateSEFromFilter(), sippGenericSchedWr(), and sippOSEProcessFilterIBufs().

u8 sSchLineBufferS::manageReq

Referenced by sippAnalysePipe2x5x(), sippCheckOPipeConnectionColourComb(), sippCheckOPipeConnectionDoGLTM(), sippCheckOPipeConnectionMedian(), sippCheckOPipeConnectionPoly(), sippIbufSetup(), and sippOBufSetup().

s32 sSchLineBufferS::mostRecentLine

Referenced by sippAddFilterToPipe(), sippCoreReInitLineBuffers(), sippFilterAddOBuf(), sippGenericDMACheck(), sippGenericGetBufferEntry(), and sippGenericSearchBuffer().

u32 sSchLineBufferS::numLines

Referenced by sippAddFilterToPipe(), sippFilterAddOBuf(), sippPipeSetNumLinesPerBuf(), and sippProcessSchedData().

u8 sSchLineBufferS::numSWConsumers

Referenced by sippAllocCmxLineBuffers(), sippFilterAddOBuf(), sippLinkFilter(), and sippLinkFilterSetOBuf().

6.162 sSippCdmaQuEntryS Struct Reference

```
#include <sippTypes.h>
```

Data Fields

- **u64 addr**
- **u64 value**

6.162.1 Field Documentation

u64 `sSippCdmaQuEntryS::addr`

Referenced by `sippRunDmaCQDrain()`.

u64 `sSippCdmaQuEntryS::value`

Referenced by `sippRunDmaCQDrain()`.

6.163 sSippCdmaQuS Struct Reference

```
#include <sippTypes.h>
```

Data Fields

- **sSippCdmaQuEntry** * `entry`
- **u32** `numEntries`
- **u32** `size`

6.163.1 Field Documentation

sSippCdmaQuEntry* `sSippCdmaQuS::entry`

Referenced by `sippDmaCQInit()`, `sippKickDmaCQ()`, and `sippRunDmaCQDrain()`.

u32 `sSippCdmaQuS::numEntries`

Referenced by `sippDmaCQInit()`, `sippKickDmaCQ()`, and `sippRunDmaCQDrain()`.

u32 `sSippCdmaQuS::size`

Referenced by `sippDmaCQInit()`, and `sippKickDmaCQ()`.

6.164 sSippCMDQuEntryS Struct Reference

```
#include <sippTypes.h>
```

Data Fields

- **u32** `addr`
- **u32** `value`

6.164.1 Field Documentation

u32 sSippCMDQuEntryS::addr

u32 sSippCMDQuEntryS::value

6.165 sSippCMDQuS Struct Reference

```
#include <sippTypes.h>
```

Data Fields

- **sSippCMDQuEntry** * quEntry
- **u32** quNum
- **u32** quSize

6.165.1 Field Documentation

sSippCMDQuEntry* sSippCMDQuS::quEntry

Referenced by sippCQInit().

u32 sSippCMDQuS::quNum

Referenced by sippCQInit().

u32 sSippCMDQuS::quSize

Referenced by sippCQInit().

6.166 StartBicubicParam Struct Reference

```
#include <startBicubic.h>
```

Data Fields

- **UInt32** width
- **UInt32** height
- **UInt32** firstShave
- **UInt8** * input
- **UInt8** * output
- **UInt32** bpp
- **tBicubicReg cmd** [12]
- **UInt32** lastShave
- **UInt32** shaveCount

6.166.1 Field Documentation

UInt32 StartBicubicParam::bpp

Referenced by configureBicubicHWblock().

tBicubicReg StartBicubicParam::cmd[12]

Referenced by configureBicubicHWblock().

UInt32 StartBicubicParam::firstShave

Referenced by configureBicubicHWblock().

UInt32 StartBicubicParam::height

Referenced by configureBicubicHWblock().

UInt8* StartBicubicParam::input

Referenced by configureBicubicHWblock().

UInt32 StartBicubicParam::lastShave

Referenced by configureBicubicHWblock().

UInt8* StartBicubicParam::output

Referenced by configureBicubicHWblock().

UInt32 StartBicubicParam::shaveCount

Referenced by configureBicubicHWblock().

UInt32 StartBicubicParam::width

Referenced by configureBicubicHWblock().

6.167 StatsAwbSatPixelsParam Struct Reference

```
#include <statsAwbSatPixels.h>
```

Data Fields

- struct {
 - UInt32 firstPatchX
 - UInt32 firstPatchY
 - UInt32 patchWidth
 - UInt32 patchHeight
 - UInt32 patchGapX
 - UInt32 patchGapY
 - UInt32 nPatchesX
 - UInt32 nPatchesY
 - UInt32 satThresh
 } AeAwbStatsCfg
- UInt32 runNr
- UInt32 crtPaxel
- UInt8 * statsOutput
- UInt32 NextVerticalStartPos
- UInt32 crtPosInPaxel

6.167.1 Field Documentation

struct { ... } StatsAwbSatPixelsParam::AeAwbStatsCfg

Referenced by svuStatsAwbSatPixels().

UInt32 StatsAwbSatPixelsParam::crtPaxel

Referenced by svuStatsAwbSatPixels().

UInt32 StatsAwbSatPixelsParam::crtPosInPaxel

Referenced by svuStatsAwbSatPixels().

UInt32 StatsAwbSatPixelsParam::firstPatchX

Referenced by svuStatsAwbSatPixels().

UInt32 StatsAwbSatPixelsParam::firstPatchY

UInt32 StatsAwbSatPixelsParam::NextVerticalStartPos

Referenced by svuStatsAwbSatPixels().

UInt32 StatsAwbSatPixelsParam::nPatchesX

Referenced by svuStatsAwbSatPixels().

UInt32 StatsAwbSatPixelsParam::nPatchesY

Referenced by svuStatsAwbSatPixels().

UInt32 StatsAwbSatPixelsParam::patchGapX

Referenced by svuStatsAwbSatPixels().

UInt32 StatsAwbSatPixelsParam::patchGapY

Referenced by svuStatsAwbSatPixels().

UInt32 StatsAwbSatPixelsParam::patchHeight

Referenced by svuStatsAwbSatPixels().

UInt32 StatsAwbSatPixelsParam::patchWidth

Referenced by svuStatsAwbSatPixels().

UInt32 StatsAwbSatPixelsParam::runNr

Referenced by svuStatsAwbSatPixels().

UInt32 StatsAwbSatPixelsParam::satThresh

Referenced by svuStatsAwbSatPixels().

UInt8* StatsAwbSatPixelsParam::statsOutput

Referenced by svuStatsAwbSatPixels().

6.168 StatsAwbSatPixelsParamU32 Struct Reference

```
#include <statsAwbSatPixelsU32.h>
```

Data Fields

- struct {
 - UInt32 firstPatchX
 - UInt32 firstPatchY
 - UInt32 patchWidth
 - UInt32 patchHeight
 - UInt32 patchGapX
 - UInt32 patchGapY
 - UInt32 nPatchesX

```

    UInt32 nPatchesY
    UInt32 satThresh
} AeAwbStatsCfg

```

- UInt32 runNr
- UInt32 crtPaxel
- UInt32 * statsOutput
- UInt32 NextVerticalStartPos
- UInt32 crtPosInPaxel

6.168.1 Field Documentation

```
struct { ... } StatsAwbSatPixelsParamU32::AeAwbStatsCfg
```

Referenced by svuStatsAwbSatPixelsU32().

UInt32 StatsAwbSatPixelsParamU32::crtPaxel

Referenced by svuStatsAwbSatPixelsU32().

UInt32 StatsAwbSatPixelsParamU32::crtPosInPaxel

Referenced by svuStatsAwbSatPixelsU32().

UInt32 StatsAwbSatPixelsParamU32::firstPatchX

Referenced by svuStatsAwbSatPixelsU32().

UInt32 StatsAwbSatPixelsParamU32::firstPatchY

UInt32 StatsAwbSatPixelsParamU32::NextVerticalStartPos

Referenced by svuStatsAwbSatPixelsU32().

UInt32 StatsAwbSatPixelsParamU32::nPatchesX

Referenced by svuStatsAwbSatPixelsU32().

UInt32 StatsAwbSatPixelsParamU32::nPatchesY

Referenced by svuStatsAwbSatPixelsU32().

UInt32 StatsAwbSatPixelsParamU32::patchGapX

Referenced by svuStatsAwbSatPixelsU32().

UInt32 StatsAwbSatPixelsParamU32::patchGapY

Referenced by svuStatsAwbSatPixelsU32().

UInt32 StatsAwbSatPixelsParamU32::patchHeight

Referenced by svuStatsAwbSatPixelsU32().

UInt32 StatsAwbSatPixelsParamU32::patchWidth

Referenced by svuStatsAwbSatPixelsU32().

UInt32 StatsAwbSatPixelsParamU32::runNr

Referenced by svuStatsAwbSatPixelsU32().

UInt32 StatsAwbSatPixelsParamU32::satThresh

Referenced by svuStatsAwbSatPixelsU32().

UInt32* StatsAwbSatPixelsParamU32::statsOutput

Referenced by svuStatsAwbSatPixelsU32().

6.169 SubpixelFilterParam Struct Reference

Parameter structure of the [Threshold](#) filter.

```
#include <subpixelFilter.h>
```

Data Fields

- uint8_t [numDisp](#)
- uint8_t [numFractionalBits](#)
- uint8_t * [LUT](#)

6.169.1 Detailed Description

Parameter structure of the [Threshold](#) filter.

6.169.2 Field Documentation

uint8_t* SubpixelFilterParam::LUT

Referenced by svuSubpixelFilter().

`uint8_t SubpixelFilterParam::numDisp`

Referenced by `svuSubpixelFilter()`.

`uint8_t SubpixelFilterParam::numFractionalBits`

Referenced by `svuSubpixelFilter()`.

6.170 ThresholdBinaryRangeParam Struct Reference

Parameter structure of the [Threshold Binary Range](#) filter.

```
#include <thresholdBinaryRange.h>
```

Data Fields

- [UInt8 lowerValue](#)
lower value
- [UInt8 upperValue](#)
upper value

6.170.1 Detailed Description

Parameter structure of the [Threshold Binary Range](#) filter.

6.170.2 Field Documentation

UInt8 [ThresholdBinaryRangeParam::lowerValue](#)

lower value

Referenced by `svuThresholdBinaryRange()`.

UInt8 [ThresholdBinaryRangeParam::upperValue](#)

upper value

Referenced by `svuThresholdBinaryRange()`.

6.171 ThresholdBinaryU8Param Struct Reference

Parameter structure of the [Threshold Binary U8](#) filter.

```
#include <thresholdBinaryU8.h>
```

Data Fields

- [UInt8 threshold](#)
threshold value

6.171.1 Detailed Description

Parameter structure of the [Threshold Binary U8](#) filter.

6.171.2 Field Documentation

[UInt8](#) ThresholdBinaryU8Param::threshold

threshold value

Referenced by `svuThresholdBinaryU8()`.

6.172 ThresholdFilterParam Struct Reference

Parameter structure of the [ThresholdFilter](#) filter.

```
#include <thresholdFilter.h>
```

Data Fields

- float [threshold](#)
- [UInt32](#) posOffset

6.172.1 Detailed Description

Parameter structure of the [ThresholdFilter](#) filter.

6.172.2 Field Documentation

[UInt32](#) ThresholdFilterParam::posOffset

Referenced by `svuThresholdFilter()`.

[float](#) ThresholdFilterParam::threshold

Referenced by `svuThresholdFilter()`.

6.173 ThresholdParam Struct Reference

Parameter structure of the [Threshold](#) filter.

```
#include <threshold.h>
```

Data Fields

- [UInt8 thresholdValue](#)
threshold value
- [UInt32 threshType](#)

6.173.1 Detailed Description

Parameter structure of the [Threshold](#) filter.

6.173.2 Field Documentation

[UInt8 ThresholdParam::thresholdValue](#)

threshold value

Referenced by `svuThreshold()`.

[UInt32 ThresholdParam::threshType](#)

one of the 5 available threshold types:

- `Thresh_To_Zero`: values below threshold are zeroed
- `Thresh_To_Zero_Inv`: opposite of `Thresh_To_Zero`
- `Thresh_To_Binary`: values below threshold are zeroed and all others are saturated to pixel max value
- `Thresh_To_Binary_Inv`: opposite of `Thresh_To_Binary`
- `Thresh_Trunc`: values above threshold are given threshold value
- default mode: `Thresh_Trunc`

Referenced by `svuThreshold()`.

6.174 tMLPIStructCtrl Struct Reference

```
#include <sippTypesPrivate.h>
```

Data Fields

- [SippFilter * firstRunFilt](#) [`SIPP_MAX_FILTERS_PER_PIPELINE`]
- [u32 nFirstRunFilt](#)
- [u32 firstUseInProg](#)

6.174.1 Field Documentation

SippFilter* tMLPISstartCQCtrl::firstRunFiltS[SIPP_MAX_FILTERS_PER_PIPELINE]

u32 tMLPISstartCQCtrl::firstUseInProg

u32 tMLPISstartCQCtrl::nFirstRunFiltS

6.175 TripleConv3x3 Struct Reference

```
#include <tripleConv3x3.h>
```

Data Fields

- **SippFilter*** c1
- **SippFilter*** c2
- **SippFilter*** c3

6.175.1 Field Documentation

SippFilter* TripleConv3x3::c1

Referenced by createTripleConv3x3().

SippFilter* TripleConv3x3::c2

Referenced by createTripleConv3x3().

SippFilter* TripleConv3x3::c3

Referenced by createTripleConv3x3().

6.176 tRTStats Struct Reference

```
#include <sippTypes.h>
```

Data Fields

- **u64*** iterTime
- **u32** DMAWaitCycles
- **u32** HWWaitCycles
- **u32** ShaveWaitCycles
- **u32** ScheduleCycles

6.176.1 Field Documentation

u32 tRTStats::DMAWaitCycles

u32 tRTStats::HWWaitCycles

u64* tRTStats::iterTime

u32 tRTStats::ScheduleCycles

u32 tRTStats::ShaveWaitCycles

6.177 tSippFramework Struct Reference

```
#include <sippTypesPrivate.h>
```

Data Fields

- **bool** bInit
- **ptSippPipelineSuper** pPipelines [SIPP_MAX_SUPPORTED_PIPELINES]
- **u32** dynIrqSipp0
- **u32** dynIrqSipp1
- **u32** dynIrqSipp2

6.177.1 Field Documentation

bool tSippFramework::bInit

Referenced by sippCreatePipeline(), sippInitialize(), and sippTerm().

u32 tSippFramework::dynIrqSipp0

Referenced by sippDynRouteIrq(), sippHWInit(), and sippIsrSetup().

u32 tSippFramework::dynIrqSipp1

Referenced by sippDynRouteIrq(), sippGenericRuntimeClaimHWResource(), sippHWInit(), and sippIsrSetup().

u32 tSippFramework::dynIrqSipp2

Referenced by sippDynRouteIrq(), sippHWInit(), and sippIsrSetup().

ptSippPipelineSuper tSippFramework::pPipelines[SIPP_MAX_SUPPORTED_PIPELINES]

Referenced by sippCreatePipeline(), sippDeletePipeline(), and sippPipeSessionControl().

6.178 tSippMCB Struct Reference

```
#include <sippTypes.h>
```

Data Fields

- void * [pVPoolListStart](#) [[vPoolLast](#)]
- void * [pVPoolListEnd](#) [[vPoolLast](#)]
- [u32](#) [numVPools](#)
- [u32](#) [pipeIdx](#)
- [pSippVPhysMap](#) [pVirtPhysMap](#)

6.178.1 Field Documentation

[u32](#) [tSippMCB::numVPools](#)

Referenced by [sippFreePipeResource\(\)](#), [sippInitLnMemPoolSlices\(\)](#), and [sippInitPipeline\(\)](#).

[u32](#) [tSippMCB::pipeIdx](#)

Referenced by [sippChooseMemPool\(\)](#), [sippInitPhysicalPoolPipe\(\)](#), and [sippInitPipeline\(\)](#).

[pSippVPhysMap](#) [tSippMCB::pVirtPhysMap](#)

Referenced by [sippInitLnMemPool\(\)](#), [sippInitLnMemPoolSlices\(\)](#), [sippInitPhysicalPoolPipe\(\)](#), [sippInitSchedPool\(\)](#), [sippMemAlloc\(\)](#), [sippMemCheck\(\)](#), [sippMemFindMaxLnMemPoolFree\(\)](#), [sippMemFree\(\)](#), and [sippMemFreeList\(\)](#).

[void*](#) [tSippMCB::pVPoolListEnd](#)[[vPoolLast](#)]

Referenced by [sippMemAlloc\(\)](#).

[void*](#) [tSippMCB::pVPoolListStart](#)[[vPoolLast](#)]

Referenced by [sippMemAlloc\(\)](#), and [sippMemFreeList\(\)](#).

6.179 tSippPhysicalPool Struct Reference

```
#include <sippTypesPrivate.h>
```

Data Fields

- const char * [name](#)
- [u8](#) * [start](#)
- [u8](#) * [pos](#)
- [u8](#) * [end](#)
- [tsSippHeap](#) * [pHeap](#)

6.179.1 Field Documentation

u8* tSippPhysicalPool::end

Referenced by sippInitLnMemPool(), sippInitLnMemPoolSlices(), sippInitSchedPool(), sippMemAlloc(), sippMemCheck(), and sippMemFindMaxLnMemPoolFree().

const char* tSippPhysicalPool::name

tsSippHeap* tSippPhysicalPool::pHeap

Referenced by sippInitLnMemPool(), sippInitLnMemPoolSlices(), sippInitSchedPool(), sippMemAlloc(), sippMemCheck(), sippMemFree(), and sippMemFreeList().

u8* tSippPhysicalPool::pos

Referenced by sippInitLnMemPool(), sippInitLnMemPoolSlices(), sippInitSchedPool(), sippMemAlloc(), sippMemCheck(), and sippMemFindMaxLnMemPoolFree().

u8* tSippPhysicalPool::start

Referenced by sippInitLnMemPool(), sippInitLnMemPoolSlices(), and sippInitSchedPool().

6.180 tSippPipelineSuper Struct Reference

```
#include <sippTypesPrivate.h>
```

Data Fields

- [SippPipeline tPublicPipe](#)
- [u32 uPipeIdx](#)
- [eSippPipeState eState](#)
- [bool bSVUOnly](#)
- [u32 uHWPipeID](#)

6.180.1 Field Documentation

bool tSippPipelineSuper::bSVUOnly

eSippPipeState tSippPipelineSuper::eState

Referenced by sippCreatePipeline(), sippElaboratePipeline(), sippInitPipeline(), sippPipeSessionControl(), and sippRescheduleRequest().

SippPipeline tSippPipelineSuper::tPublicPipe

Referenced by sippEventNotify().

u32 tsSippPipelineSuper::uHWPipeID

Referenced by sippPipeSessionControl().

u32 tsSippPipelineSuper::uPipeIdx

Referenced by sippCmxDmaDoneIrqHandler(), sippCreatePipeline(), sippDbgDumpSchedForVcsC-
Arr(), sippEventNotify(), sippInitPipeline(), sippIssueCommand(), sippObflIncHandler(), and sippSvu-
DoneIrqHandler().

6.181 tsSippHeap Struct Reference

```
#include <sippTypesPrivate.h>
```

Data Fields

- **u8** * sippHeap_start
- **u8** * sippHeap_end
- **u32** sippHeap_size
- **u8** assigned
- **u8** freeCounter
- **u8** * freePtr

6.181.1 Field Documentation

u8 tsSippHeap::assigned

Referenced by sippHeapCreate().

u8 tsSippHeap::freeCounter

Referenced by sippAlloc(), sippFree(), and sippHeapInit().

u8* tsSippHeap::freePtr

Referenced by sippFree(), sippHeapDefrag(), and sippHeapInit().

u8* tsSippHeap::sippHeap_end

Referenced by sippFree(), sippHeapAlloc(), sippHeapCheck(), sippHeapDefrag(), and sippHeapInit().

u32 tsSippHeap::sippHeap_size

Referenced by sippHeapInit().

`u8* tsSippHeap::sippHeap_start`

Referenced by `sippFree()`, `sippHeapAlloc()`, `sippHeapCheck()`, `sippHeapDefrag()`, and `sippHeapInit()`.

6.182 UndistortBParam Struct Reference

Parameter structure of the `Undistort` filter.

```
#include <undistortBrown.h>
```

Data Fields

- `int cx`
Distortion center
- `int cy`
- `float p1`
Tangential distortion coefficients
- `float p2`
- `float k1`
Radial distortion coefficients
- `float k2`

6.182.1 Detailed Description

Parameter structure of the `Undistort` filter.

6.182.2 Field Documentation

`int UndistortBParam::cx`

Distortion center

Referenced by `svuUndistortBrown()`.

`int UndistortBParam::cy`

Referenced by `svuUndistortBrown()`.

`float UndistortBParam::k1`

Radial distortion coefficients

Referenced by `svuUndistortBrown()`.

`float UndistortBParam::k2`

Referenced by `svuUndistortBrown()`.

`float UndistortBParam::p1`

Tangential distortion coefficients

Referenced by `svuUndistortBrown()`.

`float UndistortBParam::p2`

Referenced by `svuUndistortBrown()`.

6.183 UnitInfo Struct Reference

```
#include <sippTypesPrivate.h>
```

Data Fields

- [FnHwFltInit hwFnInit](#)
- [FnHwFltLoad hwFnLoad](#)
- [u32 paramSz](#)

6.183.1 Field Documentation

FnHwFltInit `UnitInfo::hwFnInit`

Referenced by `sippIniHwFilters()`.

FnHwFltLoad `UnitInfo::hwFnLoad`

Referenced by `sippCoreUnitLoad()`.

u32 `UnitInfo::paramSz`

Referenced by `sippCreateFilter()`.

6.184 UsmParam Struct Reference

Parameter structure of the sharpen filter.

```
#include <sippHwDefs_ma2x5x.h>
```

Data Fields

- [UInt32 frmDim](#)
Reserved field yes.
- [UInt32 cfg](#)
configuration bit field(see `SIPP_SHARPEN_CFG_ADR`)

- [UInt32 strength](#)
Sharpen Strength(see SIPP_SHARPEN_STREN_ADR)
- [UInt32 clip](#)
Mix factor for mixing constrained sharpened pixel with unconstrained sharpened pixel(see SIPP_SHARPEN_CLIP_ADR)
- [UInt32 limit](#)
Undershoot limit control(see SIPP_SHARPEN_LIMIT_ADR)
- [UInt32 rgnStop01](#)
Range Stop value 0 and 1(see SIPP_SHARPEN_RANGETOP_1_0_ADR)
- [UInt32 rgnStop23](#)
Range Stop value 2 and 3(see SIPP_SHARPEN_RANGETOP_3_2_ADR)
- [UInt32 coef01](#)
Filter Coefficient 0 and 1(see SIPP_SHARPEN_GAUSSIAN_1_0_ADR)
- [UInt32 coef23](#)
Filter Coefficient 2 and 3(see SIPP_SHARPEN_GAUSSIAN_1_0_ADR)

6.184.1 Detailed Description

Parameter structure of the sharpen filter.

6.184.2 Field Documentation

[UInt32 UsmParam::cfg](#)

configuration bit field(see SIPP_SHARPEN_CFG_ADR)

[UInt32 UsmParam::clip](#)

Mix factor for mixing constrained sharpened pixel with unconstrained sharpened pixel(see SIPP_SHARPEN_CLIP_ADR)

[UInt32 UsmParam::coef01](#)

Filter Coefficient 0 and 1(see SIPP_SHARPEN_GAUSSIAN_1_0_ADR)

[UInt32 UsmParam::coef23](#)

Filter Coefficient 2 and 3(see SIPP_SHARPEN_GAUSSIAN_1_0_ADR)

[UInt32 UsmParam::frmDim](#)

Reserved field yes.

[UInt32 UsmParam::limit](#)

Undershoot limit control(see SIPP_SHARPEN_LIMIT_ADR)

UInt32 UsmParam::rgnStop01

Range Stop value 0 and 1(see SIPP_SHARPEN_RANGETOP_1_0_ADR)

UInt32 UsmParam::rgnStop23

Range Stop value 2 and 3(see SIPP_SHARPEN_RANGETOP_3_2_ADR)

UInt32 UsmParam::strength

Sharpen Strength(see SIPP_SHARPEN_STREN_ADR)

6.185 WhiteBalanceBayerGBRGParam Struct Reference

Parameter structure of the [White Balance Bayer GBRG](#) filter.

```
#include <whiteBalanceBayerGBRG.h>
```

Data Fields

- [UInt16 * awbCoef](#)
Auto White Balance gains coefficients
- [UInt16 clamp](#) [1]
Clamp value

6.185.1 Detailed Description

Parameter structure of the [White Balance Bayer GBRG](#) filter.

6.185.2 Field Documentation

UInt16* WhiteBalanceBayerGBRGParam::awbCoef

Auto White Balance gains coefficients

Referenced by `svuWhiteBalanceBayerGBRG()`.

UInt16 WhiteBalanceBayerGBRGParam::clamp[1]

Clamp value

Referenced by `svuWhiteBalanceBayerGBRG()`.

6.186 WhiteBalanceRGBParam Struct Reference

Parameter structure of the [White Balance RGB](#) filter.

```
#include <whiteBalanceRGB.h>
```

Data Fields

- **UInt16 * awbCoef**
Auto White Balance gains coefficients
- **UInt16 clamp [1]**
Clamp value

6.186.1 Detailed Description

Parameter structure of the **White Balance RGB** filter.

6.186.2 Field Documentation

UInt16* WhiteBalanceRGBParam::awbCoef

Auto White Balance gains coefficients

Referenced by `svuWhiteBalanceRGB()`.

UInt16 WhiteBalanceRGBParam::clamp[1]

Clamp value

Referenced by `svuWhiteBalanceRGB()`.

6.187 XYGenParam Struct Reference

Parameter structure of the **XY Generator** filter.

```
#include <xyGen.h>
```

Data Fields

- **UInt32 * coefMat**
pointer to the warp matrix
- **UInt32 firstShave**
first shave number
- **UInt32 lastShave**
last shave number

6.187.1 Detailed Description

Parameter structure of the **XY Generator** filter.

6.187.2 Field Documentation

UInt32* XYGenParam::coefMat

pointer to the warp matrix

Referenced by svuXYgen().

UInt32 XYGenParam::firstShave

first shave number

Referenced by svuXYgen().

UInt32 XYGenParam::lastShave

last shave number

Referenced by svuXYgen().

6.188 YDnsParam Struct Reference

Parameter structure of the ydns filter.

```
#include <sippHwDefs_ma2x5x.h>
```

Data Fields

- [UInt32 frmDim](#)
Reserved field yes.
- [UInt32 cfg](#)
configuration bit field(see SIPP_LUMA_CFG_ADR)
- [UInt32 gaussLut](#) [4]
Pointer to a 32-entry LUT(see SIPP_LUMA_LUT)
- [UInt32 f2](#)
F2 4x4 2-bit LUT entries(see SIPP_LUMA_F2LUT_ADR)
- [UInt32 gammaLut](#) [5]
LUT entries for applying Gamma to reference image.
- [UInt32 distCfg](#)
Pointer to (Cosine 4th law) look-up table.
- [UInt32 distOffsets](#)
Distance-based (Cosine 4th law) look-up table X and Y tile offsets.
- [UInt32 fullFrmDim](#)
Luma denoise filter full frame dimensions in pixels.

6.188.1 Detailed Description

Parameter structure of the ydns filter.

6.188.2 Field Documentation

UInt32 YDnsParam::cfg

configuration bit field(see SIPP_LUMA_CFG_ADR)

UInt32 YDnsParam::distCfg

Pointer to (Cosine 4th law) look-up table.

UInt32 YDnsParam::distOffsets

Distance-based (Cosine 4th law) look-up table X and Y tile offsets.

UInt32 YDnsParam::f2

F2 4x4 2-bit LUT entries(see SIPP_LUMA_F2LUT_ADR)

UInt32 YDnsParam::frmDim

Reserved field yes.

UInt32 YDnsParam::fullFrmDim

Luma denoise filter full frame dimensions in pixels.

UInt32 YDnsParam::gammaLut[5]

LUT entries for applying Gamma to reference image.

UInt32 YDnsParam::gaussLut[4]

Pointer to a 32-entry LUT(see SIPP_LUMA_LUT)

Referenced by packLumaDnsGaussLut().

6.189 YDnsRefFp16Param Struct Reference

Parameter structure of the [Generate Reference for Luma Denoise\(fp16 input\)](#) filter.

```
#include <genDnsRefFp16.h>
```

Data Fields

- **UInt8 * lutGamma**

8-bit 256-entry table which can be used to apply a gamma-like adjustment to the denoise reference image.

- `UInt8 * lutDist`
8-bit 256-entry table which can be used to attenuate the denoise reference image as a function of distance from the centre pixel.
- `int shift`
Used in conjunction with `lutDist` to attenuate the image as a function of distance from the centre pixel.
- `int xOffset`

6.189.1 Detailed Description

Parameter structure of the [Generate Reference for Luma Denoise\(fp16 input\)](#) filter.

6.189.2 Field Documentation

`UInt8* YDnsRefFp16Param::lutDist`

8-bit 256-entry table which can be used to attenuate the denoise reference image as a function of distance from the centre pixel.

Referenced by `genDnsRefFp16()`, and `svuGenDnsRefFp16()`.

`UInt8* YDnsRefFp16Param::lutGamma`

8-bit 256-entry table which can be used to apply a gamma-like adjustment to the denoise reference image.

Referenced by `genDnsRefFp16()`, and `svuGenDnsRefFp16()`.

`int YDnsRefFp16Param::shift`

Used in conjunction with `lutDist` to attenuate the image as a function of distance from the centre pixel.

Referenced by `genDnsRefFp16()`.

`int YDnsRefFp16Param::xOffset`

Referenced by `svuGenDnsRefFp16()`.

6.190 YDnsRefLut10bppParam Struct Reference

Parameter structure of the [LUT 16 to 8](#) filter.

```
#include <lutP10BppU16inU8out.h>
```

Data Fields

- `UInt8 * lut`
pointer to the look-up table

6.190.1 Detailed Description

Parameter structure of the [LUT 16 to 8](#) filter.

6.190.2 Field Documentation

UInt8* `YDnsRefLut10bppParam::lut`

pointer to the look-up table

Referenced by `svuLutP10BppU16inU8out()`.

6.191 YDnsRefParam Struct Reference

Parameter structure of the [Generate Reference for Luma Denoise](#) filter.

```
#include <genDnsRef.h>
```

Data Fields

- **UInt8 *** `lutGamma`
8-bit 256-entry table which can be used to apply a gamma-like adjustment to the denoise reference image.
- **UInt8 *** `lutDist`
8-bit 256-entry table which can be used to attenuate the denoise reference image as a function of distance from the centre pixel.
- **int** `shift`
Used in conjunction with `lutDist` to attenuate the image as a function of distance from the centre pixel.

6.191.1 Detailed Description

Parameter structure of the [Generate Reference for Luma Denoise](#) filter.

6.191.2 Field Documentation

UInt8* `YDnsRefParam::lutDist`

8-bit 256-entry table which can be used to attenuate the denoise reference image as a function of distance from the centre pixel.

Referenced by `genDnsRef()`, and `svuGenDnsRef()`.

UInt8* `YDnsRefParam::lutGamma`

8-bit 256-entry table which can be used to apply a gamma-like adjustment to the denoise reference image.

Referenced by `genDnsRef()`, and `svuGenDnsRef()`.

`int YDnsRefParam::shift`

Used in conjunction with `lutDist` to attenuate the image as a function of distance from the centre pixel.

Referenced by `genDnsRef()`.

Chapter 7

File Documentation

7.1 absdiff.h File Reference

This file contains the declaration of the [Absolute difference](#) SIPP filter API.

```
#include <sipp.h>
```

Functions

- void [SVU_SYM\(\)](#) [svuAbsdiff](#) ([SippFilter](#) *fptr)
Shave function of the [Absolute difference](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuAbsdiff](#))

7.1.1 Detailed Description

This file contains the declaration of the [Absolute difference](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see:
common/license.txt

7.2 accumulateSquare.h File Reference

This file contains the declaration of the [Accumulate Square](#) SIPP filter API.

```
#include <sipp.h>
```

Functions

- void [SVU_SYM\(\)](#) [svuAccumulateSquare](#) ([SippFilter](#) *fptr)
Shave function of the [Accumulate Square](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuAccumulateSquare](#))

7.2.1 Detailed Description

This file contains the declaration of the [Accumulate Square](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.3 accumulateWeighted.h File Reference

This file contains the declaration of the [Accumulate Weighted](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [AccumulateWeightedParam](#)
Parameter structure of the [Accumulate Weighted](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuAccumulateWeighted](#) ([SippFilter](#) *fptr)
Shave function of the [Accumulate Weighted](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuAccumulateWeighted](#))

7.3.1 Detailed Description

This file contains the declaration of the [Accumulate Weighted](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.4 arithmeticAdd.h File Reference

This file contains the declaration of the [Arithmetic addition](#) SIPP filter API.

```
#include <sipp.h>
```

Functions

- void [SVU_SYM\(\)](#) [svuArithmeticAdd](#) ([SippFilter](#) *fptr)
Shave function of the [Arithmetic addition](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuArithmeticAdd](#))

7.4.1 Detailed Description

This file contains the declaration of the [Arithmetic addition](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.5 arithmeticAddmask.h File Reference

This file contains the declaration of the [Arithmetic addition with mask](#) SIPP filter API.

```
#include <sipp.h>
```

Functions

- void [SVU_SYM\(\)](#) [svuArithmeticAddmask](#) ([SippFilter](#) *fptr)
Shave function of the [Arithmetic addition with mask](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuArithmeticAddmask](#))

7.5.1 Detailed Description

This file contains the declaration of the [Arithmetic addition with mask](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.6 arithmeticSub.h File Reference

This file contains the declaration of the [Arithmetic subtraction](#) SIPP filter API.

```
#include <sipp.h>
```

Functions

- void [SVU_SYM\(\)](#) [svuArithmeticSub](#) ([SippFilter](#) *fptr)
Shave function of the [Arithmetic subtraction](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuArithmeticSub](#))

7.6.1 Detailed Description

This file contains the declaration of the [Arithmetic subtraction](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.7 arithmeticSubFp16ToFp16.h File Reference

This file contains the declaration of the [Arithmetic subtraction fp16](#) SIPP filter API.

```
#include <sipp.h>
```

Functions

- void [SVU_SYM\(\)](#) [svuArithmeticSubFp16ToFp16](#) ([SippFilter](#) *fptr)
Shave function of the [Arithmetic subtraction fp16](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuArithmeticSubFp16ToFp16](#))

7.7.1 Detailed Description

This file contains the declaration of the [Arithmetic subtraction fp16](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.8 arithmeticSubmask.h File Reference

This file contains the declaration of the [Arithmetic subtraction with mask](#) SIPP filter API.

```
#include <sipp.h>
```

Functions

- void [SVU_SYM\(\)](#) [svuArithmeticSubmask](#) ([SippFilter](#) *fptr)
Shave function of the [Arithmetic subtraction with mask](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuArithmeticSubmask](#))

7.8.1 Detailed Description

This file contains the declaration of the [Arithmetic subtraction with mask](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.9 avg.h File Reference

This file contains the declaration of the [Average](#) SIPP filter API.

```
#include <sipp.h>
```

Functions

- void [SVU_SYM\(\)](#) [svuAvg](#) ([SippFilter](#) *fptr)
Shave function of the [Average](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuAvg](#))

7.9.1 Detailed Description

This file contains the declaration of the [Average](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see:
common/license.txt

7.10 bilateral5x5.h File Reference

This file contains the declaration of the [bilateral5x5](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [Bilateral5x5Param](#)

Functions

- void [SVU_SYM\(\)](#) [svuBilateral5x5](#) ([SippFilter](#) *fptr)
Shave function of the [bilateral5x5](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuBilateral5x5](#))

7.10.1 Detailed Description

This file contains the declaration of the [bilateral5x5](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see:
common/license.txt

7.11 bitwiseAnd.h File Reference

This file contains the declaration of the [Bitwise And](#) SIPP filter API.

```
#include <sipp.h>
```

Functions

- void [SVU_SYM\(\)](#) [svuBitwiseAnd](#) (SippFilter *fptr)
Shave function of the [Bitwise And](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuBitwiseAnd](#))

7.11.1 Detailed Description

This file contains the declaration of the [Bitwise And](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.12 bitwiseAndMask.h File Reference

This file contains the declaration of the [Bitwise And with mask](#) SIPP filter API.

```
#include <sipp.h>
```

Functions

- void [SVU_SYM\(\)](#) [svubitwiseAndMask](#) (SippFilter *fptr)
Shave function of the [Bitwise And with mask](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svubitwiseAndMask](#))

7.12.1 Detailed Description

This file contains the declaration of the [Bitwise And with mask](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.13 bitwiseNot.h File Reference

This file contains the declaration of the [Bitwise Not](#) SIPP filter API.

```
#include <sipp.h>
```

Functions

- void **SVU_SYM()** **svuBitwiseNot** (**SippFilter** *fptr)
*Shave function of the **Bitwise Not** filter.*
- **SHAVE_SYM_EXPORT** (**svuBitwiseNot**)

7.13.1 Detailed Description

This file contains the declaration of the **Bitwise Not** SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.14 bitwiseOr.h File Reference

This file contains the declaration of the **Bitwise Or** SIPP filter API.

```
#include <sipp.h>
```

Functions

- void **SVU_SYM()** **svuBitwiseOr** (**SippFilter** *fptr)
*Shave function of the **Bitwise Or** filter.*
- **SHAVE_SYM_EXPORT** (**svuBitwiseOr**)

7.14.1 Detailed Description

This file contains the declaration of the **Bitwise Or** SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.15 bitwiseOrMask.h File Reference

This file contains the declaration of the **Bitwise Or with mask** SIPP filter API.

```
#include <sipp.h>
```

Functions

- void **SVU_SYM()** **svuBitwiseOrMask** (**SippFilter** *fptr)
*Shave function of the **Bitwise Or with mask** filter.*
- **SHAVE_SYM_EXPORT** (**svuBitwiseOrMask**)

7.15.1 Detailed Description

This file contains the declaration of the [Bitwise Or with mask](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.16 bitwiseXor.h File Reference

This file contains the declaration of the [Bitwise Xor](#) SIPP filter API.

```
#include <sipp.h>
```

Functions

- void [SVU_SYM\(\)](#) [svuBitwiseXor](#) (SippFilter *fptr)
Shave function of the [Bitwise Xor](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuBitwiseXor](#))

7.16.1 Detailed Description

This file contains the declaration of the [Bitwise Xor](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.17 bitwiseXorMask.h File Reference

This file contains the declaration of the [Bitwise Xor with mask](#) SIPP filter API.

```
#include <sipp.h>
```

Functions

- void [SVU_SYM\(\)](#) [svuBitwiseXorMask](#) (SippFilter *fptr)
Shave function of the [Bitwise And with mask](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuBitwiseXorMask](#))

7.17.1 Detailed Description

This file contains the declaration of the [Bitwise Xor with mask](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.18 boxFilter.h File Reference

This file contains the declaration of the [Generic Box Filter](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [BoxFilterParam](#)
Parameter structure of the [Generic Box Filter](#) filter.

Macros

- #define [BOX_MAX_V_SZ](#) 16

Enumerations

- enum [boxDataFmt](#) {
 [FMT_U8](#), [FMT_U16](#), [FMT_U32](#), [FMT_F16](#),
 [FMT_F32](#) }

Functions

- void [SVU_SYM\(\)](#) [svuBoxFilter](#) ([SippFilter](#) *fptr)
Shave function of the [Generic Box Filter](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuBoxFilter](#))

7.18.1 Detailed Description

This file contains the declaration of the [Generic Box Filter](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.18.2 Macro Definition Documentation

```
#define BOX\_MAX\_V\_SZ 16
```

Referenced by [svuBoxFilter\(\)](#).

7.19 boxFilter11x11.h File Reference

This file contains the declaration of the [Box Filter 11x11](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [BoxFilter11x11Param](#)
Parameter structure of the [Box Filter 11x11](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuBoxFilter11x11](#) ([SippFilter](#) *fptr)
Shave function of the [Box Filter 11x11](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuBoxFilter11x11](#))

7.19.1 Detailed Description

This file contains the declaration of the [Box Filter 11x11](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.20 boxFilter13x13.h File Reference

This file contains the declaration of the [Box Filter 13x13](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [BoxFilter13x13Param](#)
Parameter structure of the [Box Filter 13x13](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuBoxFilter13x13](#) ([SippFilter](#) *fptr)
Shave function of the [Box Filter 13x13](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuBoxFilter13x13](#))

7.20.1 Detailed Description

This file contains the declaration of the [Box Filter 13x13](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.21 [boxFilter15x15.h](#) File Reference

This file contains the declaration of the [Box Filter 15x15](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [BoxFilter15x15Param](#)
Parameter structure of the [Box Filter 15x15](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuBoxFilter15x15](#) ([SippFilter](#) *fptr)
Shave function of the [Box Filter 15x15](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuBoxFilter15x15](#))

7.21.1 Detailed Description

This file contains the declaration of the [Box Filter 15x15](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.22 [boxFilter3x3.h](#) File Reference

This file contains the declaration of the [Box Filter 3x3](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [BoxFilter3x3Param](#)
Parameter structure of the [Box Filter 3x3](#) filter.

Functions

- void **SVU_SYM()** **svuBoxFilter3x3** (**SippFilter** *fptr)
*Shave function of the **Box Filter 3x3** filter.*
- **SHAVE_SYM_EXPORT** (**svuBoxFilter3x3**)

7.22.1 Detailed Description

This file contains the declaration of the **Box Filter 3x3** SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.23 boxFilter5x5.h File Reference

This file contains the declaration of the **Box Filter 5x5** SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct **BoxFilter5x5Param**
*Parameter structure of the **Box Filter 5x5** filter.*

Functions

- void **SVU_SYM()** **svuBoxFilter5x5** (**SippFilter** *fptr)
*Shave function of the **Box Filter 5x5** filter.*
- **SHAVE_SYM_EXPORT** (**svuBoxFilter5x5**)

7.23.1 Detailed Description

This file contains the declaration of the **Box Filter 5x5** SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.24 boxFilter7x7.h File Reference

This file contains the declaration of the **Box Filter 7x7** SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [BoxFilter7x7Param](#)
Parameter structure of the [Box Filter 7x7](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuBoxFilter7x7](#) ([SippFilter](#) *fptr)
Shave function of the [Box Filter 7x7](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuBoxFilter7x7](#))

7.24.1 Detailed Description

This file contains the declaration of the [Box Filter 7x7](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.25 [boxFilter9x9.h](#) File Reference

This file contains the declaration of the [Box Filter 9x9](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [BoxFilter9x9Param](#)
Parameter structure of the [Box Filter 9x9](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuBoxFilter9x9](#) ([SippFilter](#) *fptr)
Shave function of the [Box Filter 9x9](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuBoxFilter9x9](#))

7.25.1 Detailed Description

This file contains the declaration of the [Box Filter 9x9](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.26 cannyEdgeDetection.h File Reference

This file contains the declaration of the [Canny Edge Detection](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [cannyEdgeDetectionParam](#)
Parameter structure of the [Canny Edge Detection](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuCannyEdgeDetection](#) ([SippFilter](#) *fptr)
Shave function of the [Canny Edge Detection](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuCannyEdgeDetection](#))

7.26.1 Detailed Description

This file contains the declaration of the [Canny Edge Detection](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.27 censusMatching16.h File Reference

This file contains the declaration of the [censusMatching16](#) SIPP filter API.

```
#include <sipp.h>
```

Functions

- void [SVU_SYM\(\)](#) [svuCensusMatching16](#) ([SippFilter](#) *fptr)
Parameter structure of the [censusMatching16](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuCensusMatching16](#))

7.27.1 Detailed Description

This file contains the declaration of the [censusMatching16](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.28 censusMatching32.h File Reference

This file contains the declaration of the [censusMatching32](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [CensusMatching32Param](#)
Parameter structure of the [censusMatching32](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuCensusMatching32](#) ([SippFilter](#) *fptr)
Shave function of the [censusMatching32](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuCensusMatching32](#))

7.28.1 Detailed Description

This file contains the declaration of the [censusMatching32](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.29 censusMatching64.h File Reference

This file contains the declaration of the [censusMatching64](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [CensusMatching64Param](#)
Parameter structure of the [censusMatching64](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuCensusMatching64](#) ([SippFilter](#) *fptr)
Shave function of the [censusMatching64](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuCensusMatching64](#))

7.29.1 Detailed Description

This file contains the declaration of the `censusMatching64` SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: `common/license.txt`

7.30 `censusMatching65.h` File Reference

This file contains the declaration of the `censusMatching65` SIPP filter API.

```
#include <sipp.h>
```

Functions

- void `SVU_SYM()` `svuCensusMatching65` (`SippFilter *fptr`)
Parameter structure of the `censusMatching65` filter.
- `SHAVE_SYM_EXPORT` (`svuCensusMatching65`)

7.30.1 Detailed Description

This file contains the declaration of the `censusMatching65` SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: `common/license.txt`

7.31 `censusMatchingPyr.h` File Reference

This file contains the declaration of the `censusMatchingPyr` SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct `CensusMatchingPyrParam`
Parameter structure of the `censusMatchingPyr` filter.

Functions

- void `SVU_SYM()` `svuCensusMatchingPyr` (`SippFilter *fptr`)
Shave function of the `censusMatchingPyr` filter.
- `SHAVE_SYM_EXPORT` (`svuCensusMatchingPyr`)

7.31.1 Detailed Description

This file contains the declaration of the [censusMatchingPyr](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.32 censusMin16.h File Reference

This file contains the declaration of the [censusMin16](#) SIPP filter API.

```
#include <sipp.h>
```

Functions

- void [SVU_SYM\(\)](#) [svuCensusMin16](#) ([SippFilter](#) *fptr)
Parameter structure of the [censusMin16](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuCensusMin16](#))

7.32.1 Detailed Description

This file contains the declaration of the [censusMin16](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.33 censusMin64.h File Reference

This file contains the declaration of the [censusMin64](#) SIPP filter API.

```
#include <sipp.h>
```

Functions

- void [SVU_SYM\(\)](#) [svuCensusMin64](#) ([SippFilter](#) *fptr)
Parameter structure of the [censusMin64](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuCensusMin64](#))

7.33.1 Detailed Description

This file contains the declaration of the [censusMin64](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.34 censusMin65.h File Reference

This file contains the declaration of the [censusMin65](#) SIPP filter API.

```
#include <sipp.h>
```

Functions

- void [SVU_SYM\(\)](#) [svuCensusMin65](#) ([SippFilter](#) *fptr)
Parameter structure of the [censusMin65](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuCensusMin65](#))

7.34.1 Detailed Description

This file contains the declaration of the [censusMin65](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.35 censusMin7.h File Reference

This file contains the declaration of the [censusMin7](#) SIPP filter API.

```
#include <sipp.h>
```

Functions

- void [SVU_SYM\(\)](#) [svuCensusMin7](#) ([SippFilter](#) *fptr)
Parameter structure of the [censusMin7](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuCensusMin7](#))

7.35.1 Detailed Description

This file contains the declaration of the [censusMin7](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.36 censusTransform5x5.h File Reference

This file contains the declaration of the [CensusTransform5x5](#) SIPP filter API.

```
#include <sipp.h>
```

Functions

- void [SVU_SYM\(\)](#) [svuCensusTransform5x5](#) ([SippFilter](#) *fptr)
Parameter structure of the [CensusTransform5x5](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuCensusTransform5x5](#))

7.36.1 Detailed Description

This file contains the declaration of the [CensusTransform5x5](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.37 channelExtract.h File Reference

This file contains the declaration of the [channelExtract](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [ChannelExtractParam](#)
Parameter structure of the [channelExtract](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuChannelExtract](#) ([SippFilter](#) *fptr)
Shave function of the [channelExtract](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuChannelExtract](#))

7.37.1 Detailed Description

This file contains the declaration of the [channelExtract](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.38 chromaBlock.h File Reference

This file contains the declaration of the [Chroma Block](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [ChromaBlkParam](#)
Parameter structure of the [Chroma Block](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuChromaBlock](#) ([SippFilter](#) *fptr)
Shave function of the [Chroma Block](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuChromaBlock](#))

7.38.1 Detailed Description

This file contains the declaration of the [Chroma Block](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.39 combDecimDemosaicAwbGains.h File Reference

```
#include <sipp.h>
```

Data Structures

- struct [CombDecimAwbGainsParam](#)

Enumerations

- enum [BayerFormat](#) { [BAYER_FORMAT_GRBG](#) = 0, [BAYER_FORMAT_RGGB](#) = 1, [BAYER_FORMAT_GBRG](#) = 2, [BAYER_FORMAT_BGGR](#) = 3 }

Functions

- void [SVU_SYM\(\)](#) [svuCombDecimDemosaicAwbGains](#) ([SippFilter](#) *fptr)
- [SHAVE_SYM_EXPORT](#) ([svuCombDecimDemosaicAwbGains](#))

7.39.1 Enumeration Type Documentation

enum **BayerFormat**

Enumerator

BAYER_FORMAT_GRBG
BAYER_FORMAT_RGGB
BAYER_FORMAT_GBRG
BAYER_FORMAT_BGGR

7.39.2 Function Documentation

SHAVE_SYM_EXPORT (**svuCombDecimDemosaicAwbGains**)

void **SVU_SYM**() **svuCombDecimDemosaicAwbGains** (**SippFilter** * fptr)

7.40 combDecimDemosaicAwbGainsStats.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/combDecimDemosaicAwbGainsStats/combDecimDemosaic-
AwbGainsStats.h>
```

Macros

- #define **MAX_STATS_SIZE** (68 * 4)

Functions

- void **combDecimDemosaicFinal** (unsigned char **output, **UInt32** *outStatSat, unsigned short **iline, unsigned int width, unsigned int gains[3], **UInt32** satThresh, **UInt16** *indexListHz, **UInt32** clearBuff)
- void **svuCombDecimDemosaicAwbGainsStats** (**SippFilter** *fptr)

Variables

- **UInt32** emptyBuf [**MAX_STATS_SIZE**]
- **UInt32** DisablePaxelSumMacro = 0xFFFFFFFF
- **UInt16** * hzInterval = (**UInt16***)&DisablePaxelSumMacro

7.40.1 Macro Definition Documentation

```
#define MAX_STATS_SIZE (68 * 4)
```

7.40.2 Function Documentation

```
void combDecimDemosaicFinal ( unsigned char ** output, UInt32 * outStatSat, unsigned short **
iline, unsigned int width, unsigned int gains[3], UInt32 satThresh, UInt16 * indexListHz, UInt32
clearBuff )
```

Referenced by svuCombDecimDemosaicAwbGainsStats().

```
void svuCombDecimDemosaicAwbGainsStats ( SippFilter * fptr )
```

7.40.3 Variable Documentation

```
UInt32 DisablePaxelSumMacro = 0xFFFFFFFF
```

Referenced by svuCombDecimDemosaicAwbGainsStats().

```
UInt32 emptyBuf[MAX_STATS_SIZE]
```

Referenced by combDecimDemosaicFinal().

```
UInt16* hzInterval = (UInt16*)&DisablePaxelSumMacro
```

7.41 combDecimDemosaicAwbGainsStats.h File Reference

```
#include <sipp.h>
```

Data Structures

- struct [AeAwbStatsCfg](#)
- struct [CombDecimStatsGainsParam](#)

Functions

- void [SVU_SYM\(\)](#) svuCombDecimDemosaicAwbGainsStats (SippFilter *fptr)
- [SHAVE_SYM_EXPORT](#) (svuCombDecimDemosaicAwbGainsStats)

7.41.1 Function Documentation

```
SHAVE_SYM_EXPORT ( svuCombDecimDemosaicAwbGainsStats )
```

```
void SVU_SYM() svuCombDecimDemosaicAwbGainsStats ( SippFilter * fptr )
```

7.42 contrast.h File Reference

This file contains the declaration of the [Contrast](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [ContrastParam](#)
Parameter structure of the [Contrast](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuContrast](#) ([SippFilter](#) *fptr)
Shave function of the [Contrast](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuContrast](#))

7.42.1 Detailed Description

This file contains the declaration of the [Contrast](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.43 [conv11x11.h](#) File Reference

This file contains the declaration of the [Convolution 11x11](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [Conv11x11Param](#)
Parameter structure of the [Convolution 11x11](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuConv11x11](#) ([SippFilter](#) *fptr)
Shave function of the [Convolution 11x11](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuConv11x11](#))

7.43.1 Detailed Description

This file contains the declaration of the [Convolution 11x11](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.44 conv15x1.h File Reference

This file contains the declaration of the [Convolution 15x1](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [Conv15x1Param](#)
Parameter structure of the [Convolution 15x1](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuConv15x1](#) ([SippFilter](#) *fptr)
Shave function of the [Convolution 15x1](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuConv15x1](#))

7.44.1 Detailed Description

This file contains the declaration of the [Convolution 15x1](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.45 conv1x15.h File Reference

This file contains the declaration of the [Convolution 1x15](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [Conv1x15Param](#)
Parameter structure of the [Convolution 1x15](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuConv1x15](#) ([SippFilter](#) *fptr)
Shave function of the [Convolution 1x15](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuConv1x15](#))

7.45.1 Detailed Description

This file contains the declaration of the [Convolution 1x15](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.46 conv1x5.h File Reference

This file contains the declaration of the [Convolution 1x5](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [Conv1x5Param](#)
Parameter structure of the [Convolution 1x5](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuConv1x5](#) ([SippFilter](#) *fptr)
Shave function of the [Convolution 1x5](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuConv1x5](#))

7.46.1 Detailed Description

This file contains the declaration of the [Convolution 1x5](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.47 conv1x5Fp16ToFp16.h File Reference

This file contains the declaration of the [Convolution 1x5 Fp16ToFp16](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [Conv1x5Fp16ToFp16Param](#)
Parameter structure of the [Convolution 1x5 Fp16ToFp16](#) filter.

Functions

- void **SVU_SYM()** **svuConv1x5Fp16ToFp16** (SippFilter *fptr)
*Shave function of the **Convolution 1x5 Fp16ToFp16** filter.*
- **SHAVE_SYM_EXPORT** (svuConv1x5Fp16ToFp16)

7.47.1 Detailed Description

This file contains the declaration of the **Convolution 1x5 Fp16ToFp16** SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.48 conv1x7.h File Reference

This file contains the declaration of the **Convolution 1x7** SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct **Conv1x7Param**
*Parameter structure of the **Convolution 1x7** filter.*

Functions

- void **SVU_SYM()** **svuConv1x7** (SippFilter *fptr)
*Shave function of the **Convolution 1x7** filter.*
- **SHAVE_SYM_EXPORT** (svuConv1x7)

7.48.1 Detailed Description

This file contains the declaration of the **Convolution 1x7** SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.49 conv1x7Fp16ToFp16.h File Reference

This file contains the declaration of the **Convolution 1x7 Fp16ToFp16** SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [Conv1x7Fp16ToFp16Param](#)
Parameter structure of the [Convolution 1x7 Fp16ToFp16](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuConv1x7Fp16ToFp16](#) ([SippFilter](#) *fptr)
Shave function of the [Convolution 1x7 Fp16ToFp16](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuConv1x7Fp16ToFp16](#))

7.49.1 Detailed Description

This file contains the declaration of the [Convolution 1x7 Fp16ToFp16](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.50 conv1x9.h File Reference

This file contains the declaration of the [Convolution 1x9](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [Conv1x9Param](#)
Parameter structure of the [Convolution 1x9](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuConv1x9](#) ([SippFilter](#) *fptr)
Shave function of the [Convolution 1x9](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuConv1x9](#))

7.50.1 Detailed Description

This file contains the declaration of the [Convolution 1x9](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.51 conv3x3.h File Reference

This file contains the declaration of the [Convolution 3x3](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [Conv3x3Param](#)
Parameter structure of the [Convolution 3x3](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuConv3x3](#) ([SippFilter](#) *fptr)
Shave function of the [Convolution 3x3](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuConv3x3](#))

7.51.1 Detailed Description

This file contains the declaration of the [Convolution 3x3](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.52 conv3x3Fp16ToFp16.h File Reference

This file contains the declaration of the [Convolution 3x3 Fp16ToFp16](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [Conv3x3Fp16ToFp16Param](#)
Parameter structure of the [Convolution 3x3 Fp16ToFp16](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuConv3x3Fp16ToFp16](#) ([SippFilter](#) *fptr)
Shave function of the [Convolution 3x3 Fp16ToFp16](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuConv3x3Fp16ToFp16](#))

7.52.1 Detailed Description

This file contains the declaration of the [Convolution 3x3 Fp16ToFp16](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.53 conv5x1.h File Reference

This file contains the declaration of the [Convolution 5x1](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [Conv5x1Param](#)
Parameter structure of the [Convolution 5x1](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuConv5x1](#) ([SippFilter](#) *fptr)
Shave function of the [Convolution 5x1](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuConv5x1](#))

7.53.1 Detailed Description

This file contains the declaration of the [Convolution 5x1](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.54 conv5x1Fp16ToFp16.h File Reference

This file contains the declaration of the [Convolution 5x1 Fp16ToFp16](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [Conv5x1Fp16ToFp16Param](#)
Parameter structure of the [Convolution 5x1 Fp16ToFp16](#) filter.

Functions

- void **SVU_SYM()** **svuConv5x1Fp16ToFp16** (SippFilter *fptr)
*Shave function of the **Convolution 5x1 Fp16ToFp16** filter.*
- **SHAVE_SYM_EXPORT** (svuConv5x1Fp16ToFp16)

7.54.1 Detailed Description

This file contains the declaration of the **Convolution 5x1 Fp16ToFp16** SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.55 conv5x5.h File Reference

This file contains the declaration of the **Convolution 5x5** SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct **Conv5x5Param**
*Parameter structure of the **Convolution 5x5** filter.*

Functions

- void **SVU_SYM()** **svuConv5x5** (SippFilter *fptr)
*Shave function of the **Convolution 5x5** filter.*
- **SHAVE_SYM_EXPORT** (svuConv5x5)

7.55.1 Detailed Description

This file contains the declaration of the **Convolution 5x5** SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.56 conv5x5Fp16ToFp16.h File Reference

This file contains the declaration of the **Convolution 5x5 Fp16ToFp16** SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [Conv5x5Fp16ToFp16Param](#)
Parameter structure of the [Convolution 5x5 Fp16ToFp16](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuConv5x5Fp16ToFp16](#) ([SippFilter](#) *fptr)
Shave function of the [Convolution 5x5 Fp16ToFp16](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuConv5x5Fp16ToFp16](#))

7.56.1 Detailed Description

This file contains the declaration of the [Convolution 5x5 Fp16ToFp16](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.57 conv7x1.h File Reference

This file contains the declaration of the [Convolution 7x1](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [Conv7x1Param](#)
Parameter structure of the [Convolution 7x1](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuConv7x1](#) ([SippFilter](#) *fptr)
Shave function of the [Convolution 7x1](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuConv7x1](#))

7.57.1 Detailed Description

This file contains the declaration of the [Convolution 7x1](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.58 conv7x1Fp16ToFp16.h File Reference

This file contains the declaration of the [Convolution 7x1 Fp16ToFp16](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [Conv7x1Fp16ToFp16Param](#)
Parameter structure of the [Convolution 7x1 Fp16ToFp16](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuConv7x1Fp16ToFp16](#) ([SippFilter](#) *fptr)
Shave function of the [Convolution 7x1 Fp16ToFp16](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuConv7x1Fp16ToFp16](#))

7.58.1 Detailed Description

This file contains the declaration of the [Convolution 7x1 Fp16ToFp16](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.59 conv7x7.h File Reference

This file contains the declaration of the [Convolution 7x7](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [Conv7x7Param](#)
Parameter structure of the [Convolution 7x7](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuConv7x7](#) ([SippFilter](#) *fptr)
Shave function of the [Convolution 7x7](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuConv7x7](#))

7.59.1 Detailed Description

This file contains the declaration of the [Convolution 7x7](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.60 conv7x7Fp16ToFp16.h File Reference

This file contains the declaration of the [Convolution 7x7 Fp16ToFp16](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [Conv7x7ParamFp16ToFp16](#)
Parameter structure of the [Convolution 7x7 Fp16ToFp16](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuConv7x7Fp16ToFp16](#) ([SippFilter](#) *fptr)
Shave function of the [Convolution 7x7 Fp16ToFp16](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuConv7x7Fp16ToFp16](#))

7.60.1 Detailed Description

This file contains the declaration of the [Convolution 7x7 Fp16ToFp16](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.61 conv7x7Fp16ToU8.h File Reference

This file contains the declaration of the [Convolution 7x7 Fp16ToU8](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [Conv7x7ParamFp16ToU8](#)
Parameter structure of the [Convolution 7x7 Fp16ToU8](#) filter.

Functions

- void **SVU_SYM()** **svuConv7x7Fp16ToU8** (SippFilter *fptr)
*Shave function of the **Convolution 7x7 Fp16ToU8** filter.*
- **SHAVE_SYM_EXPORT** (svuConv7x7Fp16ToU8)

7.61.1 Detailed Description

This file contains the declaration of the **Convolution 7x7 Fp16ToU8** SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.62 conv9x1.h File Reference

This file contains the declaration of the **Convolution 9x1** SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct **Conv9x1Param**
*Parameter structure of the **Convolution 9x1** filter.*

Functions

- void **SVU_SYM()** **svuConv9x1** (SippFilter *fptr)
*Shave function of the **Convolution 9x1** filter.*
- **SHAVE_SYM_EXPORT** (svuConv9x1)

7.62.1 Detailed Description

This file contains the declaration of the **Convolution 9x1** SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.63 conv9x9.h File Reference

This file contains the declaration of the **Convolution 9x9** SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [Conv9x9Param](#)
Parameter structure of the [Convolution 9x9](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuConv9x9](#) ([SippFilter](#) *fptr)
Shave function of the [Convolution 9x9](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuConv9x9](#))

7.63.1 Detailed Description

This file contains the declaration of the [Convolution 9x9](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.64 [conv9x9Fp16ToFp16.h](#) File Reference

This file contains the declaration of the [Convolution 9x9 Fp16ToFp16](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [Conv9x9Fp16ToFp16Param](#)
Parameter structure of the [Convolution 9x9 Fp16ToFp16](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuConv9x9Fp16ToFp16](#) ([SippFilter](#) *fptr)
Shave function of the [Convolution 9x9 Fp16ToFp16](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuConv9x9Fp16ToFp16](#))

7.64.1 Detailed Description

This file contains the declaration of the [Convolution 9x9 Fp16ToFp16](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.65 convert16bppTo8bpp.h File Reference

This file contains the declaration of the [Convert 16bpp To 8bpp](#) SIPP filter API.

```
#include <sipp.h>
```

Functions

- void [SVU_SYM\(\)](#) [svuConvert16bppTo8bpp](#) ([SippFilter](#) *fptr)
Shave function of the [Convert 16bpp To 8bpp](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuConvert16bppTo8bpp](#))

7.65.1 Detailed Description

This file contains the declaration of the [Convert 16bpp To 8bpp](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see:
common/license.txt

7.66 convertF16ToU8.h File Reference

This file contains the declaration of the [Convert F16 To U8](#) SIPP filter API.

```
#include <sipp.h>
```

Functions

- void [SVU_SYM\(\)](#) [svuConvertF16ToU8](#) ([SippFilter](#) *fptr)
Shave function of the [Convert F16 To U8](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuConvertF16ToU8](#))

7.66.1 Detailed Description

This file contains the declaration of the [Convert F16 To U8](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see:
common/license.txt

7.67 convertFrom12BppTo8Bpp.h File Reference

This file contains the declaration of the [12Bpp to 8Bpp conversion](#) SIPP filter API.

```
#include <sipp.h>
```

Functions

- void **SVU_SYM()** **svuConvertFrom12BppTo8Bpp** (SippFilter *fptr)
Shave function of the 12Bpp to 8Bpp conversion filter.
- **SHAVE_SYM_EXPORT** (svuConvertFrom12BppTo8Bpp)

7.67.1 Detailed Description

This file contains the declaration of the **12Bpp to 8Bpp conversion** SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.68 convertPFp16U16.h File Reference

This file contains the declaration of the **Convert Fp16 to U16** SIPP filter API.

```
#include <sipp.h>
```

Functions

- void **SVU_SYM()** **svuConvertPFp16U16** (SippFilter *fptr)
Shave function of the Convert Fp16 to U16 filter.
- **SHAVE_SYM_EXPORT** (svuConvertPFp16U16)

7.68.1 Detailed Description

This file contains the declaration of the **Convert Fp16 to U16** SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.69 convertPU16Fp16.h File Reference

This file contains the declaration of the **Convert U16 to Fp16** SIPP filter API.

```
#include <sipp.h>
```

Functions

- void **SVU_SYM()** **svuConvertPU16Fp16** (SippFilter *fptr)
Shave function of the Convert U16 to Fp16 filter.
- **SHAVE_SYM_EXPORT** (svuConvertPU16Fp16)

7.69.1 Detailed Description

This file contains the declaration of the [Convert U16 to Fp16](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.70 [convertU8ToF16.h](#) File Reference

This file contains the declaration of the [Convert U8 To F16](#) SIPP filter API.

```
#include <sipp.h>
```

Functions

- void [SVU_SYM\(\)](#) [svuConvertU8ToF16](#) ([SippFilter](#) *fptr)
Shave function of the [Convert U8 To F16](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuConvertU8ToF16](#))

7.70.1 Detailed Description

This file contains the declaration of the [Convert U8 To F16](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.71 [convertYUV400ToYUV422.h](#) File Reference

This file contains the declaration of the [YUV400 to YUV422 conversion](#) SIPP filter API.

```
#include <sipp.h>
```

Functions

- void [SVU_SYM\(\)](#) [svuConvertYUV400ToYUV422](#) ([SippFilter](#) *fptr)
Shave function of the [YUV400 to YUV422 conversion](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuConvertYUV400ToYUV422](#))

7.71.1 Detailed Description

This file contains the declaration of the [YUV400 to YUV422 conversion](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.72 convGeneric.h File Reference

This file contains the declaration of the [Generic Convolution](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [ConvGenericParam](#)
Parameter structure of the [Generic Convolution](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuConvGeneric](#) ([SippFilter](#) *fptr)
Shave function of the [Generic Convolution](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuConvGeneric](#))

7.72.1 Detailed Description

This file contains the declaration of the [Generic Convolution](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.73 convSeparable11x11.h File Reference

This file contains the declaration of the [Convolution Separable 11x11](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [ConvSeparable11x11Param](#)
Parameter structure of the [Convolution Separable 11x11](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuConvSeparable11x11](#) ([SippFilter](#) *fptr)
Shave function of the [Convolution Separable 11x11](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuConvSeparable11x11](#))

7.73.1 Detailed Description

This file contains the declaration of the [Convolution Separable 11x11 SIPP](#) filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.74 convSeparable11x11Fp16ToFp16.h File Reference

This file contains the declaration of the [Convolution Separable 11x11 Fp16ToFp16 SIPP](#) filter API.

```
#include <sipp.h>
```

Data Structures

- struct [ConvSeparable11x11Fp16ToFp16Param](#)
Parameter structure of the [Convolution Separable 11x11 Fp16ToFp16](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuConvSeparable11x11Fp16ToFp16](#) ([SippFilter](#) *fptr)
Shave function of the [Convolution Separable 11x11 Fp16ToFp16](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuConvSeparable11x11Fp16ToFp16](#))

7.74.1 Detailed Description

This file contains the declaration of the [Convolution Separable 11x11 Fp16ToFp16 SIPP](#) filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.75 convSeparable3x3.h File Reference

This file contains the declaration of the [Convolution Separable 3x3 SIPP](#) filter API.

```
#include <sipp.h>
```

Data Structures

- struct [ConvSeparable3x3Param](#)
Parameter structure of the [Convolution Separable 3x3](#) filter.

Functions

- void **SVU_SYM()** **svuConvSeparable3x3** (SippFilter *fptr)
*Shave function of the **Convolution Separable 3x3** filter.*
- **SHAVE_SYM_EXPORT** (svuConvSeparable3x3)

7.75.1 Detailed Description

This file contains the declaration of the **Convolution Separable 3x3** SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.76 convSeparable3x3Fp16ToFp16.h File Reference

This file contains the declaration of the **Convolution Separable 3x3 Fp16ToFp16** SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct **ConvSeparable3x3Fp16ToFp16Param**
*Parameter structure of the **Convolution Separable 3x3 Fp16ToFp16** filter.*

Functions

- void **SVU_SYM()** **svuConvSeparable3x3Fp16ToFp16** (SippFilter *fptr)
*Shave function of the **Convolution Separable 3x3 Fp16ToFp16** filter.*
- **SHAVE_SYM_EXPORT** (svuConvSeparable3x3Fp16ToFp16)

7.76.1 Detailed Description

This file contains the declaration of the **Convolution Separable 3x3 Fp16ToFp16** SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.77 convSeparable5x5.h File Reference

This file contains the declaration of the **Convolution Separable 5x5** SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [ConvSeparable5x5Param](#)
Parameter structure of the [Convolution Separable 5x5](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuConvSeparable5x5](#) ([SippFilter](#) *fptr)
Shave function of the [Convolution Separable 5x5](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuConvSeparable5x5](#))

7.77.1 Detailed Description

This file contains the declaration of the [Convolution Separable 5x5](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.78 [convSeparable5x5Fp16ToFp16.h](#) File Reference

This file contains the declaration of the [Convolution Separable 5x5 Fp16ToFp16](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [ConvSeparable5x5Fp16ToFp16Param](#)
Parameter structure of the [Convolution Separable 5x5 Fp16ToFp16](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuConvSeparable5x5Fp16ToFp16](#) ([SippFilter](#) *fptr)
Shave function of the [Convolution Separable 5x5 Fp16ToFp16](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuConvSeparable5x5Fp16ToFp16](#))

7.78.1 Detailed Description

This file contains the declaration of the [Convolution Separable 5x5 Fp16ToFp16](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.79 convSeparable7x7.h File Reference

This file contains the declaration of the [Convolution Separable 7x7](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [ConvSeparable7x7Param](#)
Parameter structure of the [Convolution Separable 7x7](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuConvSeparable7x7](#) ([SippFilter](#) *fptr)
Shave function of the [Convolution Separable 7x7](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuConvSeparable7x7](#))

7.79.1 Detailed Description

This file contains the declaration of the [Convolution Separable 7x7](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.80 convSeparable7x7Fp16ToFp16.h File Reference

This file contains the declaration of the [Convolution Separable 7x7 Fp16ToFp16](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [ConvSeparable7x7Fp16ToFp16Param](#)
Parameter structure of the [Convolution Separable 7x7 Fp16ToFp16](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuConvSeparable7x7Fp16ToFp16](#) ([SippFilter](#) *fptr)
Shave function of the [Convolution Separable 7x7 Fp16ToFp16](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuConvSeparable7x7Fp16ToFp16](#))

7.80.1 Detailed Description

This file contains the declaration of the [Convolution Separable 7x7 Fp16ToFp16](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.81 convSeparable9x9.h File Reference

This file contains the declaration of the [Convolution Separable 9x9](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [ConvSeparable9x9Param](#)
Parameter structure of the [Convolution Separable 9x9](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuConvSeparable9x9](#) ([SippFilter](#) *fptr)
Shave function of the [Convolution Separable 9x9](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuConvSeparable9x9](#))

7.81.1 Detailed Description

This file contains the declaration of the [Convolution Separable 9x9](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.82 convSeparable9x9Fp16ToFp16.h File Reference

This file contains the declaration of the [Convolution Separable 9x9 Fp16ToFp16](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [ConvSeparable9x9Fp16ToFp16Param](#)
Parameter structure of the [Convolution Separable 9x9 Fp16ToFp16](#) filter.

Functions

- void **SVU_SYM()** **svuConvSeparable9x9Fp16ToFp16** (SippFilter *fptr)
*Shave function of the **Convolution Separable 9x9 Fp16ToFp16** filter.*
- **SHAVE_SYM_EXPORT** (svuConvSeparable9x9Fp16ToFp16)

7.82.1 Detailed Description

This file contains the declaration of the **Convolution Separable 9x9 Fp16ToFp16** SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.83 convYuv444.h File Reference

This file contains the declaration of the **Convert to YUV444** SIPP filter API.

```
#include <sipp.h>
```

Functions

- void **SVU_SYM()** **svuRgbYuv444** (SippFilter *fptr)
*Shave function of the **Convert to YUV444** filter.*
- **SHAVE_SYM_EXPORT** (svuRgbYuv444)

7.83.1 Detailed Description

This file contains the declaration of the **Convert to YUV444** SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.84 copy.h File Reference

This file contains the declaration of the **Copy** SIPP filter API.

```
#include <sipp.h>
```

Functions

- void **SVU_SYM()** **svuCopy** (SippFilter *fptr)
*Shave function of the **Copy** filter.*
- **SHAVE_SYM_EXPORT** (svuCopy)

7.84.1 Detailed Description

This file contains the declaration of the [Copy](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.85 cornerMinEigenVal.h File Reference

This file contains the declaration of the [Corner Min Eigenvalue](#) SIPP filter API.

```
#include <sipp.h>
```

Functions

- void [SVU_SYM\(\)](#) [svuCornerMinEigenVal](#) ([SippFilter](#) *fptr)
Shave function of the [Corner Min Eigenvalue](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuCornerMinEigenVal](#))

7.85.1 Detailed Description

This file contains the declaration of the [Corner Min Eigenvalue](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.86 cornerMinEigenValpatched.h File Reference

This file contains the declaration of the [Corner Min Eigenvalue Patched](#) SIPP filter API.

```
#include <sipp.h>
```

Functions

- void [SVU_SYM\(\)](#) [svuCornerMinEigenValpatched](#) ([SippFilter](#) *fptr)
Shave function of the [Corner Min Eigenvalue Patched](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuCornerMinEigenValpatched](#))

7.86.1 Detailed Description

This file contains the declaration of the [Corner Min Eigenvalue Patched](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.87 crop.h File Reference

This file contains the declaration of the [Crop](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [CropParam](#)
Parameter structure of the [Crop](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuCrop](#) ([SippFilter](#) *fptr)
Shave function of the [Crop](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuCrop](#))

7.87.1 Detailed Description

This file contains the declaration of the [Crop](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.88 cropCvtPlaneMode.h File Reference

This file contains the declaration of the [CropCvtPlaneMode](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [CropParam](#)
Parameter structure of the [Crop](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuCropCvtPlaneMode](#) ([SippFilter](#) *fptr)
Shave function of the [Crop](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuCropCvtPlaneMode](#))

7.88.1 Detailed Description

This file contains the declaration of the CropCvtPlaneMode SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.89 cvtColorChromaYUVToNV12.h File Reference

This file contains the declaration of the [YUV to NV12 chroma conversion](#) SIPP filter API.

```
#include <sipp.h>
#include <swcFrameTypes.h>
```

Data Structures

- struct [CvtColorChromaYUVToNV12Param](#)
Parameter structure of the YUV to NV12 chroma conversion filter.

Functions

- void [SVU_SYM\(\)](#) [svuCvtColorChromaYUVToNV12](#) ([SippFilter](#) *fptr)
Shave function of the YUV to NV12 chroma conversion filter.
- [SHAVE_SYM_EXPORT](#) ([svuCvtColorChromaYUVToNV12](#))

7.89.1 Detailed Description

This file contains the declaration of the [YUV to NV12 chroma conversion](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2017, all rights reserved. For License Warranty see: common/license.txt

7.89.2 Function Documentation

[SHAVE_SYM_EXPORT](#) ([svuCvtColorChromaYUVToNV12](#))

void [SVU_SYM\(\)](#) [svuCvtColorChromaYUVToNV12](#) ([SippFilter](#) * fptr)

Shave function of the [YUV to NV12 chroma conversion](#) filter.

7.90 cvtColorNV21toRGB.h File Reference

This file contains the declaration of the [NV21 to RGB conversion](#) SIPP filter API.

```
#include <sipp.h>
```

Functions

- void [SVU_SYM\(\)](#) [svucvtColorNV21toRGB](#) ([SippFilter](#) *fptr)
Shave function of the [NV21 to RGB conversion](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svucvtColorNV21toRGB](#))

7.90.1 Detailed Description

This file contains the declaration of the [NV21 to RGB conversion](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.91 cvtColorRGBfp16ToLumaU8.h File Reference

This file contains the declaration of the [RGB\(fp16\) to Luma\(u8\) conversion](#) SIPP filter API.

```
#include <sipp.h>
```

Functions

- void [SVU_SYM\(\)](#) [svuCvtColorRGBfp16ToLumaU8](#) ([SippFilter](#) *fptr)
Shave function of the [RGB\(fp16\) to Luma\(u8\) conversion](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuCvtColorRGBfp16ToLumaU8](#))

7.91.1 Detailed Description

This file contains the declaration of the [RGB\(fp16\) to Luma\(u8\) conversion](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.92 cvtColorRGBfp16ToUV420U8.h File Reference

This file contains the declaration of the [RGB\(fp16\) to UV420\(u8\) conversion](#) SIPP filter API.

```
#include <sipp.h>
```

Functions

- void **SVU_SYM()** **svuCvtColorRGBfp16ToUV420U8** (SippFilter *fptr)
*Shave function of the **RGB(fp16) to UV420(u8) conversion** filter.*
- **SHAVE_SYM_EXPORT** (svuCvtColorRGBfp16ToUV420U8)

7.92.1 Detailed Description

This file contains the declaration of the **RGB(fp16) to UV420(u8) conversion** SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.93 cvtColorRGBtoChromaNV12.h File Reference

This file contains the declaration of the **RGB to Chroma NV12 conversion** SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct **cvtColorChromaNV12Param**
*Parameter structure of the **RGB to Chroma NV12 conversion** filter.*

Functions

- void **SVU_SYM()** **svuCvtColorRGBtoChromaNV12** (SippFilter *fptr)
*Shave function of the **RGB to Chroma NV12 conversion** filter.*
- **SHAVE_SYM_EXPORT** (svuCvtColorRGBtoChromaNV12)

7.93.1 Detailed Description

This file contains the declaration of the **RGB to Chroma NV12 conversion** SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.94 cvtColorRGBtoLuma.h File Reference

This file contains the declaration of the **RGB to Luma conversion** SIPP filter API.

```
#include <sipp.h>
```

Functions

- void **SVU_SYM()** **svuCvtColorRGBtoLuma** (**SippFilter** *fptr)
*Shave function of the **RGB to Luma conversion** filter.*
- **SHAVE_SYM_EXPORT** (**svuCvtColorRGBtoLuma**)

7.94.1 Detailed Description

This file contains the declaration of the **RGB to Luma conversion** SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.95 cvtColorRGBtoLumaNV12.h File Reference

This file contains the declaration of the **RGB to Luma NV12 conversion** SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct **cvtColorLumaNV12Param**
*Parameter structure of the **RGB to Luma NV12 conversion** filter.*

Functions

- void **SVU_SYM()** **svuCvtColorRGBtoLumaNV12** (**SippFilter** *fptr)
*Shave function of the **RGB to Luma NV12 conversion** filter.*
- **SHAVE_SYM_EXPORT** (**svuCvtColorRGBtoLumaNV12**)

7.95.1 Detailed Description

This file contains the declaration of the **RGB to Luma NV12 conversion** SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.96 cvtColorRGBtoUV.h File Reference

This file contains the declaration of the **RGB to UV conversion** SIPP filter API.

```
#include <sipp.h>
```


Functions

- void **SVU_SYM()** **svuCvtColorRGBtoUV** (SippFilter *fptr)
*Shave function of the **RGB to UV conversion** filter.*
- **SHAVE_SYM_EXPORT** (svuCvtColorRGBtoUV)

7.96.1 Detailed Description

This file contains the declaration of the **RGB to UV conversion** SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.97 cvtColorRGBtoUV420.h File Reference

This file contains the declaration of the **RGB to UV420 conversion** SIPP filter API.

```
#include <sipp.h>
```

Functions

- void **SVU_SYM()** **svuCvtColorRGBtoUV420** (SippFilter *fptr)
*Shave function of the **RGB to UV420 conversion** filter.*
- **SHAVE_SYM_EXPORT** (svuCvtColorRGBtoUV420)

7.97.1 Detailed Description

This file contains the declaration of the **RGB to UV420 conversion** SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.98 cvtColorRGBToYUV422.h File Reference

This file contains the declaration of the **RGB to YUV422 conversion** SIPP filter API.

```
#include <sipp.h>
```

Functions

- void **SVU_SYM()** **svuCvtColorRGBToYUV422** (SippFilter *fptr)
*Shave function of the **RGB to YUV422 conversion** filter.*
- **SHAVE_SYM_EXPORT** (svuCvtColorRGBToYUV422)

7.98.1 Detailed Description

This file contains the declaration of the [RGB to YUV422 conversion](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.99 cvtColorYUV422ToRGB.h File Reference

This file contains the declaration of the [YUV422 to RGB conversion](#) SIPP filter API.

```
#include <sipp.h>
```

Functions

- void [SVU_SYM\(\)](#) [svuCvtColorYUV422ToRGB](#) ([SippFilter](#) *fptr)
Shave function of the [cvtColorYUV422ToRGB](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuCvtColorYUV422ToRGB](#))

7.99.1 Detailed Description

This file contains the declaration of the [YUV422 to RGB conversion](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.100 cvtColorYUVToRGB.h File Reference

This file contains the declaration of the [YUV to RGB conversion](#) SIPP filter API.

```
#include <sipp.h>
```

Functions

- void [SVU_SYM\(\)](#) [svuCvtColorYUVToRGB](#) ([SippFilter](#) *fptr)
Shave function of the [YUV to RGB conversion](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuCvtColorYUVToRGB](#))

7.100.1 Detailed Description

This file contains the declaration of the [YUV to RGB conversion](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.101 dilate3x3.h File Reference

This file contains the declaration of the [Dilate 3x3](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [Dilate3x3Param](#)
Parameter structure of the [Dilate 3x3](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuDilate3x3](#) ([SippFilter](#) *fptr)
Shave function of the [Dilate 3x3](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuDilate3x3](#))

7.101.1 Detailed Description

This file contains the declaration of the [Dilate 3x3](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.102 dilate5x5.h File Reference

This file contains the declaration of the [Dilate 5x5](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [Dilate5x5Param](#)
Parameter structure of the [Dilate 5x5](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuDilate5x5](#) ([SippFilter](#) *fptr)
Shave function of the [Dilate 5x5](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuDilate5x5](#))

7.102.1 Detailed Description

This file contains the declaration of the [Dilate 5x5](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.103 dilate7x7.h File Reference

This file contains the declaration of the [Dilate 7x7](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [Dilate7x7Param](#)
Parameter structure of the [Dilate 7x7](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuDilate7x7](#) ([SippFilter](#) *fptr)
Shave function of the [Dilate 7x7](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuDilate7x7](#))

7.103.1 Detailed Description

This file contains the declaration of the [Dilate 7x7](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.104 dilateGeneric.h File Reference

This file contains the declaration of the [Generic Dilate](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [DilateGenericParam](#)
Parameter structure of the [Generic Dilate](#) filter.

Functions

- void **SVU_SYM()** **svuDilateGeneric** (**SippFilter** *fptr)
*Shave function of the **Generic Dilate** filter.*
- **SHAVE_SYM_EXPORT** (**svuDilateGeneric**)

7.104.1 Detailed Description

This file contains the declaration of the **Generic Dilate** SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.105 disp2depth.h File Reference

This file contains the declaration of the **Threshold** SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct **Disp2depthParam**
*Parameter structure of the **Threshold** filter.*

Functions

- void **SVU_SYM()** **svudisp2depth** (**SippFilter** *fptr)
- **SHAVE_SYM_EXPORT** (**svudisp2depth**)

7.105.1 Detailed Description

This file contains the declaration of the **Threshold** SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.105.2 Function Documentation

SHAVE_SYM_EXPORT (**svudisp2depth**)

void **SVU_SYM()** **svudisp2depth** (**SippFilter** * fptr)

7.106 disp2depth_exec.c File Reference

```
#include "VectorTypes.h"
```

Functions

- void [flip_dsp2depth_explic_vect](#) (uint8_t *input, half8 *output, half *LUT, uint32_t width)
disp2depth kernel
- void [dsp2depth_explic_vect](#) (uint8_t *input, half8 *output, half *LUT, uint32_t width)
- void [flip_mvcevDisp2depth](#) (uint8_t *input, half8 *output, half *LUT, uint32_t width)
- void [flip_mvcevDisp2depth16bit](#) (uint16_t *input, half8 *output, half *LUT, uint32_t width)
- void [mvcevDisp2depth](#) (uint8_t *input, half8 *output, half *LUT, uint32_t width)
- void [mvcevDisp2depth16bit](#) (uint16_t *input, half8 *output, half *LUT, uint32_t width)

7.106.1 Function Documentation

```
void dsp2depth_explic_vect ( uint8_t * input, half8 * output, half * LUT, uint32_t width )
```

```
void flip_dsp2depth_explic_vect ( uint8_t * input, half8 * output, half * LUT, uint32_t width )
```

disp2depth kernel

```
void flip_mvcevDisp2depth ( uint8_t * input, half8 * output, half * LUT, uint32_t width )
```

Referenced by svudisp2depth().

```
void flip_mvcevDisp2depth16bit ( uint16_t * input, half8 * output, half * LUT, uint32_t width )
```

Referenced by svudisp2depth().

```
void mvcevDisp2depth ( uint8_t * input, half8 * output, half * LUT, uint32_t width )
```

Referenced by svudisp2depth().

```
void mvcevDisp2depth16bit ( uint16_t * input, half8 * output, half * LUT, uint32_t width )
```

Referenced by svudisp2depth().

7.107 disp2depth_exec.h File Reference

Functions

- void [flip_dsp2depth_explic_vect](#) (uint8_t *input, half8 *output, half *LUT, uint32_t width)
- void [dsp2depth_explic_vect](#) (uint8_t *input, half8 *output, half *LUT, uint32_t width)
- void [flip_mvcevDisp2depth](#) (uint8_t *input, half8 *output, half *LUT, uint32_t width)

- void [mvcvDisp2depth](#) (uint8_t *input, half8 *output, half *LUT, uint32_t width)
- void [flip_mvcvDisp2depth16bit](#) (uint16_t *input, half8 *output, half *LUT, uint32_t width)
- void [mvcvDisp2depth16bit](#) (uint16_t *input, half8 *output, half *LUT, uint32_t width)

7.107.1 Function Documentation

[void dsp2depth_explic_vect](#) (uint8_t * input, half8 * output, half * LUT, uint32_t width)

[void flip_dsp2depth_explic_vect](#) (uint8_t * input, half8 * output, half * LUT, uint32_t width)

[void flip_mvcvDisp2depth](#) (uint8_t * input, half8 * output, half * LUT, uint32_t width)

Referenced by [svudisp2depth](#)().

[void flip_mvcvDisp2depth16bit](#) (uint16_t * input, half8 * output, half * LUT, uint32_t width)

Referenced by [svudisp2depth](#)().

[void mvcvDisp2depth](#) (uint8_t * input, half8 * output, half * LUT, uint32_t width)

Referenced by [svudisp2depth](#)().

[void mvcvDisp2depth16bit](#) (uint16_t * input, half8 * output, half * LUT, uint32_t width)

Referenced by [svudisp2depth](#)().

7.108 [equalizeHist.h](#) File Reference

This file contains the declaration of the [Equalize Histogram](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [EqualizeHistParam](#)
Parameter structure of the [Equalize Histogram](#) filter.

Functions

- void [SVU_SYM](#)() [svuEqualizeHist](#) (SippFilter *fptr)
Shave function of the [Equalize Histogram](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuEqualizeHist](#))

7.108.1 Detailed Description

This file contains the declaration of the [Equalize Histogram](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.109 erode3x3.h File Reference

This file contains the declaration of the [Erode 3x3](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [Erode3x3Param](#)
Parameter structure of the [Erode 3x3](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuErode3x3](#) ([SippFilter](#) *fptr)
Shave function of the [Erode 3x3](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuErode3x3](#))

7.109.1 Detailed Description

This file contains the declaration of the [Erode 3x3](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.110 erode5x5.h File Reference

This file contains the declaration of the [Erode 5x5](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [Erode5x5Param](#)
Parameter structure of the [Erode 5x5](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuErode5x5](#) ([SippFilter](#) *fptr)
Shave function of the [Erode 5x5](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuErode5x5](#))

7.110.1 Detailed Description

This file contains the declaration of the [Erode 5x5](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.111 [erode7x7.h](#) File Reference

This file contains the declaration of the [Erode 7x7](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [Erode7x7Param](#)
Parameter structure of the [Erode 7x7](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuErode7x7](#) ([SippFilter](#) *fptr)
Shave function of the [Erode 7x7](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuErode7x7](#))

7.111.1 Detailed Description

This file contains the declaration of the [Erode 7x7](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.112 [extAfStats.h](#) File Reference

This file contains the declaration of the [AF Stats](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [SippHwIOBuf](#)
- struct [PpAf](#)

Parameter structure of the [AF Stats](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuExtAfStats](#) ([SippFilter](#) *fptr)
Shave function of the [AF Stats](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuExtAfStats](#))

7.112.1 Detailed Description

This file contains the declaration of the [AF Stats](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.113 extStatsSatPixelsU32.h File Reference

```
#include <sipp.h>
```

Data Structures

- struct [ExtStatsSatPixelsU32Param](#)

Functions

- void [SVU_SYM\(\)](#) [svuExtStatsSatPixelsU32](#) ([SippFilter](#) *fptr)
- [SHAVE_SYM_EXPORT](#) ([svuExtStatsSatPixelsU32](#))

7.113.1 Function Documentation

[SHAVE_SYM_EXPORT](#) ([svuExtStatsSatPixelsU32](#))

void [SVU_SYM\(\)](#) [svuExtStatsSatPixelsU32](#) ([SippFilter](#) * fptr)

7.114 fast9M2.h File Reference

This file contains the declaration of the [Fast9M2](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [Fast9M2Param](#)
Parameter structure of the [Fast9M2](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuFast9M2](#) ([SippFilter](#) *fptr)
Shave function of the [Fast9M2](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuFast9M2](#))

7.114.1 Detailed Description

This file contains the declaration of the [Fast9M2](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.115 [fast9ScoreCv.h](#) File Reference

This file contains the declaration of the [Fast9ScoreCv](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [Fast9ScoreCvParam](#)
Parameter structure of the [Fast9ScoreCv](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuFast9ScoreCv](#) ([SippFilter](#) *fptr)
Shave function of the [Fast9ScoreCv](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuFast9ScoreCv](#))

7.115.1 Detailed Description

This file contains the declaration of the [Fast9ScoreCv](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.116 gauss.h File Reference

This file contains the declaration of the [Gauss Blur](#) SIPP filter API.

```
#include <sipp.h>
```

Functions

- void [SVU_SYM\(\)](#) [svuGauss](#) ([SippFilter](#) *fptr)
Shave function of the [Gauss Blur](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuGauss](#))

7.116.1 Detailed Description

This file contains the declaration of the [Gauss Blur](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see:
common/license.txt

7.117 gaussHx2.h File Reference

This file contains the declaration of the [GaussHx2](#) SIPP filter API.

```
#include <sipp.h>
```

Functions

- void [SVU_SYM\(\)](#) [svuGaussHx2](#) ([SippFilter](#) *fptr)
Shave function of the [GaussHx2](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuGaussHx2](#))

7.117.1 Detailed Description

This file contains the declaration of the [GaussHx2](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see:
common/license.txt

7.118 gaussHx2_fp16.h File Reference

This file contains the declaration of the [GaussHx2_fp16](#) SIPP filter API.

```
#include <sipp.h>
```

Functions

- void **SVU_SYM()** **svuGaussHx2_fp16** (**SippFilter** *fptr)
*Shave function of the **GaussHx2_fp16** filter.*
- **SHAVE_SYM_EXPORT** (**svuGaussHx2_fp16**)

7.118.1 Detailed Description

This file contains the declaration of the **GaussHx2_fp16** SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.119 gaussVx2.h File Reference

This file contains the declaration of the **GaussVx2** SIPP filter API.

```
#include <sipp.h>
```

Functions

- void **SVU_SYM()** **svuGaussVx2** (**SippFilter** *fptr)
*Shave function of the **GaussVx2** filter.*
- **SHAVE_SYM_EXPORT** (**svuGaussVx2**)

7.119.1 Detailed Description

This file contains the declaration of the **GaussVx2** SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.120 gaussVx2_fp16.h File Reference

This file contains the declaration of the **GaussVx2_fp16** SIPP filter API.

```
#include <sipp.h>
```

Functions

- void **SVU_SYM()** **svuGaussVx2_fp16** (**SippFilter** *fptr)
*Shave function of the **GaussVx2_fp16** filter.*
- **SHAVE_SYM_EXPORT** (**svuGaussVx2_fp16**)

7.120.1 Detailed Description

This file contains the declaration of the [GaussVx2_fp16](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.121 genChroma.h File Reference

This file contains the declaration of the [Generate Chroma](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [ChrGenParam](#)
Parameter structure of the filter.

Functions

- void [SVU_SYM\(\)](#) [svuGenChroma](#) ([SippFilter](#) *fptr)
Shave function of the [Generate Chroma](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuGenChroma](#))

7.121.1 Detailed Description

This file contains the declaration of the [Generate Chroma](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.122 genChromaSS.h File Reference

This file contains the declaration of the [Generate Chroma with subsampling](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [ChrGenSSParam](#)
Parameter structure of the [Generate Chroma with subsampling](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuGenChromaSS](#) ([SippFilter](#) *fptr)
Shave function of the [Generate Chroma with subsampling](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuGenChromaSS](#))

7.122.1 Detailed Description

This file contains the declaration of the [Generate Chroma with subsampling](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.123 [genDnsRef.h](#) File Reference

This file contains the declaration of the [Generate Reference for Luma Denoise](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [YDnsRefParam](#)
Parameter structure of the [Generate Reference for Luma Denoise](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuGenDnsRef](#) ([SippFilter](#) *fptr)
Shave function of the [Generate Reference for Luma Denoise](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuGenDnsRef](#))

7.123.1 Detailed Description

This file contains the declaration of the [Generate Reference for Luma Denoise](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.124 [genDnsRefFp16.h](#) File Reference

This file contains the declaration of the [Generate Reference for Luma Denoise\(fp16 input\)](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [YDnsRefFp16Param](#)

Parameter structure of the [Generate Reference for Luma Denoise\(fp16 input\)](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuGenDnsRefFp16](#) (SippFilter *fptr)
Shave function of the [Generate Reference for Luma Denoise\(fp16 input\)](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuGenDnsRefFp16](#))

7.124.1 Detailed Description

This file contains the declaration of the [Generate Reference for Luma Denoise\(fp16 input\)](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.125 [genLuma.h](#) File Reference

This file contains the declaration of the [Luma Blur](#) SIPP filter API.

```
#include <sipp.h>
```

Functions

- void [SVU_SYM\(\)](#) [svuGenLuma](#) (SippFilter *fptr)
Shave function of the [Luma Blur](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuGenLuma](#))

7.125.1 Detailed Description

This file contains the declaration of the [Luma Blur](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.126 [genLumaU8Fp16.h](#) File Reference

This file contains the declaration of the [Generate Luma U8 to Fp16](#) SIPP filter API.

```
#include <sipp.h>
```


Data Structures

- struct [GenLumaU8Fp16Param](#)
Shave function of the [Generate Luma U8 to Fp16](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuGenLumaU8Fp16](#) ([SippFilter](#) *fptr)
- [SHAVE_SYM_EXPORT](#) ([svuGenLumaU8Fp16](#))

7.126.1 Detailed Description

This file contains the declaration of the [Generate Luma U8 to Fp16](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.127 [greyDesat.h](#) File Reference

This file contains the declaration of the [greyDesat](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [GreyDesatParam](#)
Parameter structure of the [greyDesat](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuGreyDesat](#) ([SippFilter](#) *fptr)
Shave function of the [greyDesat](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuGreyDesat](#))

7.127.1 Detailed Description

This file contains the declaration of the [greyDesat](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.128 hammingDistance.h File Reference

This file contains the declaration of the censusTransform7x7 SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [HammingDistanceParam](#)
Parameter structure of the [hammingDistance](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuHammingDistance](#) ([SippFilter](#) *fptr)
Shave function of the [hammingDistance](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuHammingDistance](#))

7.128.1 Detailed Description

This file contains the declaration of the censusTransform7x7 SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.129 harrisResponse.h File Reference

This file contains the declaration of the [harrisResponse](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [HarrisSwParam](#)
Parameter structure of the [harrisResponse](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuHarrisResponse](#) ([SippFilter](#) *fptr)
Shave function of the [harrisResponse](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuHarrisResponse](#))

7.129.1 Detailed Description

This file contains the declaration of the [harrisResponse](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.130 histogram.h File Reference

This file contains the declaration of the [Histogram](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [HistogramParam](#)
Parameter structure of the [Histogram](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuHistogram](#) ([SippFilter](#) *fptr)
Shave function of the [Histogram](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuHistogram](#))

7.130.1 Detailed Description

This file contains the declaration of the [Histogram](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.131 histogramStat.h File Reference

This file contains the declaration of the [histogramStat](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [HistogramStatParam](#)
Parameter structure of the [histogramStat](#) filter.

Functions

- void **SVU_SYM()** **svuHistogramStat** (SippFilter *fptr)
*Shave function of the **histogramStat** filter.*
- **SHAVE_SYM_EXPORT** (svuHistogramStat)

7.131.1 Detailed Description

This file contains the declaration of the **histogramStat** SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.132 homography.h File Reference

This file contains the declaration of the **Homography** SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct **HomographyParam**
*Parameter structure of the **Homography** filter.*

Functions

- void **SVU_SYM()** **svuHomography** (SippFilter *fptr)
*Shave function of the **Homography** filter.*
- **SHAVE_SYM_EXPORT** (svuHomography)

7.132.1 Detailed Description

This file contains the declaration of the **Homography** SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.133 integralImageSqSumF32M2.h File Reference

This file contains the declaration of the **Integral Image Square Sum(f32)** SIPP filter API.

```
#include <sipp.h>
```

Functions

- void **SVU_SYM()** **svuIntegralImageSqSumF32M2** (**SippFilter** *fptr)
*Shave function of the **Integral Image Square Sum(f32)** filter.*
- **SHAVE_SYM_EXPORT** (**svuIntegralImageSqSumF32M2**)

7.133.1 Detailed Description

This file contains the declaration of the **Integral Image Square Sum(f32)** SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.134 integralImageSqSumU32M2.h File Reference

This file contains the declaration of the **Integral Image Square Sum(U32)** SIPP filter API.

```
#include <sipp.h>
```

Functions

- void **SVU_SYM()** **svuIntegralImageSqSumU32M2** (**SippFilter** *fptr)
*Shave function of the **Integral Image Square Sum(U32)** filter.*
- **SHAVE_SYM_EXPORT** (**svuIntegralImageSqSumU32M2**)

7.134.1 Detailed Description

This file contains the declaration of the **Integral Image Square Sum(U32)** SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.135 integralImageSumF32M2.h File Reference

This file contains the declaration of the **Integral Image Sum(f32)** SIPP filter API.

```
#include <sipp.h>
```

Functions

- void **SVU_SYM()** **svuIntegralImageSumF32M2** (**SippFilter** *fptr)
*Shave function of the **Integral Image Sum(f32)** filter.*
- **SHAVE_SYM_EXPORT** (**svuIntegralImageSumF32M2**)

7.135.1 Detailed Description

This file contains the declaration of the [Integral Image Sum\(f32\)](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.136 integralImageSumU16U32.h File Reference

This file contains the declaration of the [Integral Image Sum\(U16toU32\)](#) SIPP filter API.

```
#include <sipp.h>
```

Functions

- void [SVU_SYM\(\)](#) [svuIntegralImageSumU16U32](#) ([SippFilter](#) *fptr)
Shave function of the [Integral Image Sum\(U16toU32\)](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuIntegralImageSumU16U32](#))

7.136.1 Detailed Description

This file contains the declaration of the [Integral Image Sum\(U16toU32\)](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.137 integralImageSumU32M2.h File Reference

This file contains the declaration of the [Integral Image Sum\(U32\)](#) SIPP filter API.

```
#include <sipp.h>
```

Functions

- void [SVU_SYM\(\)](#) [svuIntegralImageSumU32M2](#) ([SippFilter](#) *fptr)
Shave function of the [Integral Image Sum\(U32\)](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuIntegralImageSumU32M2](#))

7.137.1 Detailed Description

This file contains the declaration of the [Integral Image Sum\(U32\)](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.138 interpolatePixelBilinear.h File Reference

This file contains the declaration of the [interpolatePixelBilinear](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [InterpolatePixelBilinearParam](#)
Parameter structure of the [interpolatePixelBilinear](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuInterpolatePixelBilinear](#) ([SippFilter](#) *fptr)
Shave function of the [interpolatePixelBilinear](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuInterpolatePixelBilinear](#))

7.138.1 Detailed Description

This file contains the declaration of the [interpolatePixelBilinear](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.139 laplacian3x3.h File Reference

This file contains the declaration of the [Laplacian 3x3](#) SIPP filter API.

```
#include <sipp.h>
```

Functions

- void [SVU_SYM\(\)](#) [svuLaplacian3x3](#) ([SippFilter](#) *fptr)
Shave function of the [Laplacian 3x3](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuLaplacian3x3](#))

7.139.1 Detailed Description

This file contains the declaration of the [Laplacian 3x3](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.140 [laplacian5x5.h](#) File Reference

This file contains the declaration of the [Laplacian 5x5](#) SIPP filter API.

```
#include <sipp.h>
```

Functions

- void [SVU_SYM\(\)](#) [svuLaplacian5x5](#) ([SippFilter](#) *fptr)
Shave function of the [Laplacian 5x5](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuLaplacian5x5](#))

7.140.1 Detailed Description

This file contains the declaration of the [Laplacian 5x5](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.141 [laplacian5x5Fp16ToFp16.h](#) File Reference

This file contains the declaration of the [Laplacian 5x5 Fp16 To Fp16](#) SIPP filter API.

```
#include <sipp.h>
```

Functions

- void [SVU_SYM\(\)](#) [svuLaplacian5x5Fp16ToFp16](#) ([SippFilter](#) *fptr)
Shave function of the [Laplacian 5x5 Fp16 To Fp16](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuLaplacian5x5Fp16ToFp16](#))

7.141.1 Detailed Description

This file contains the declaration of the [Laplacian 5x5 Fp16 To Fp16](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.142 `laplacian7x7.h` File Reference

This file contains the declaration of the [Laplacian 7x7](#) SIPP filter API.

```
#include <sipp.h>
```

Functions

- void [SVU_SYM\(\)](#) [svuLaplacian7x7](#) ([SippFilter](#) *fptr)
Shave function of the [Laplacian 7x7](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuLaplacian7x7](#))

7.142.1 Detailed Description

This file contains the declaration of the [Laplacian 7x7](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.143 `laplacian7x7Fp16ToFp16.h` File Reference

This file contains the declaration of the [Laplacian 7x7 Fp16 To Fp16](#) SIPP filter API.

```
#include <sipp.h>
```

Functions

- void [SVU_SYM\(\)](#) [svuLaplacian7x7Fp16ToFp16](#) ([SippFilter](#) *fptr)
Shave function of the [Laplacian 7x7 Fp16 To Fp16](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuLaplacian7x7Fp16ToFp16](#))

7.143.1 Detailed Description

This file contains the declaration of the [Laplacian 7x7 Fp16 To Fp16](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.144 localMaxMin3x3_fp16.h File Reference

This file contains the declaration of the [localMaxMin3x3_fp16](#) SIPP filter API.

```
#include <sipp.h>
```

Functions

- void [SVU_SYM\(\)](#) [svuLocalMaxMin3x3_fp16](#) ([SippFilter](#) *fptr)
Shave function of the [localMaxMin3x3_fp16](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuLocalMaxMin3x3_fp16](#))

7.144.1 Detailed Description

This file contains the declaration of the [localMaxMin3x3_fp16](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.145 localTM.h File Reference

This file contains the declaration of the [localTM](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [LocalTMParam](#)
Parameter structure of the [localTM](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuLocalTM](#) ([SippFilter](#) *fptr)
Shave function of the [localTM](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuLocalTM](#))

7.145.1 Detailed Description

This file contains the declaration of the [localTM](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.146 lowLvlCorr.h File Reference

This file contains the declaration of the [Low Level Correction](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [LowLvlCorrParam](#)
Parameter structure of the [Low Level Correction](#) filter.

Functions

- void [SVU_SYM](#)() [svuLowLvlCorr](#) ([SippFilter](#) *fptr)
Shave function of the [Low Level Correction on Multiple Planes](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuLowLvlCorr](#))

7.146.1 Detailed Description

This file contains the declaration of the [Low Level Correction](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.147 lowLvlCorrMultiplePlanes.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/lowLvlCorrMultiplePlanes/lowLvlCorrMultiplePlanes.-
h>
```

Macros

- #define [SUB_SATURATE](#)(x, y) (x < y ? 0 : (x - y))

Functions

- void [svulowLvlCorrMultiplePlanes](#) ([SippFilter](#) *fptr)
Shave function of the [Low Level Correction on Multiple Planes](#) filter.

7.147.1 Macro Definition Documentation

```
#define SUB_SATURATE( x, y ) (x < y ? 0 : (x - y))
```

Referenced by `svulowLvlCorrMultiplePlanes()`.

7.148 lowLvlCorrMultiplePlanes.h File Reference

This file contains the declaration of the [Low Level Correction on Multiple Planes](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [LowLvlCorrNPIParam](#)
Parameter structure of the [Low Level Correction on Multiple Planes](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svulowLvlCorrMultiplePlanes](#) ([SippFilter](#) *fptr)
Shave function of the [Low Level Correction on Multiple Planes](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svulowLvlCorrMultiplePlanes](#))

7.148.1 Detailed Description

This file contains the declaration of the [Low Level Correction on Multiple Planes](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.149 lumaBlur.h File Reference

This file contains the declaration of the [Luma Blur](#) SIPP filter API.

```
#include <sipp.h>
```

Functions

- void [SVU_SYM\(\)](#) [svuLumaBlur](#) ([SippFilter](#) *fptr)
Shave function of the [Luma Blur](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuLumaBlur](#))

7.149.1 Detailed Description

This file contains the declaration of the [Luma Blur](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.150 lut10to16.h File Reference

This file contains the declaration of the [LUT 10 to 16](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [Lut10to16Param](#)
Parameter structure of the [LUT 10 to 16](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuLut10to16](#) ([SippFilter](#) *fptr)
Shave function of the [LUT 10 to 16](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuLut10to16](#))

7.150.1 Detailed Description

This file contains the declaration of the [LUT 10 to 16](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.151 lut10to8.h File Reference

This file contains the declaration of the [LUT 10 to 8](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [Lut10to8Param](#)
Parameter structure of the [LUT 10 to 8](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuLut10to8](#) ([SippFilter](#) *fptr)
Shave function of the [LUT 10 to 8](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuLut10to8](#))

7.151.1 Detailed Description

This file contains the declaration of the [LUT 10 to 8](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.152 lut12to16.h File Reference

This file contains the declaration of the [LUT 12 to 16](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [Lut12to16Param](#)
Parameter structure of the [LUT 12 to 16](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuLut12to16](#) ([SippFilter](#) *fptr)
Shave function of the [LUT 12 to 16](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuLut12to16](#))

7.152.1 Detailed Description

This file contains the declaration of the [LUT 12 to 16](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.153 lut12to8.h File Reference

This file contains the declaration of the [LUT 12 to 8](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [Lut12to8Param](#)
Parameter structure of the [LUT 12 to 8](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuLut12to8](#) ([SippFilter](#) *fptr)
Shave function of the [LUT 12 to 8](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuLut12to8](#))

7.153.1 Detailed Description

This file contains the declaration of the [LUT 12 to 8](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.154 lut8to8.h File Reference

This file contains the declaration of the [LUT 8 to 8](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [Lut8to8Param](#)
Parameter structure of the [LUT 8 to 8](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuLut8to8](#) ([SippFilter](#) *fptr)
Shave function of the [LUT 8 to 8](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuLut8to8](#))

7.154.1 Detailed Description

This file contains the declaration of the [LUT 8 to 8](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.155 lutP10BppU16inU8out.h File Reference

This file contains the declaration of the [LUT 16 to 8](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [YDnsRefLut10bppParam](#)
Parameter structure of the [LUT 16 to 8](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuLutP10BppU16inU8out](#) ([SippFilter](#) *fptr)
Shave function of the [LUT 16 to 8](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuLutP10BppU16inU8out](#))

7.155.1 Detailed Description

This file contains the declaration of the [LUT 16 to 8](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.156 maxTest3x3_fp16.h File Reference

This file contains the declaration of the [maxTest3x3_fp16](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [MaxTest3x3fp16Param](#)
Parameter structure of the [maxTest3x3_fp16](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuMaxTest3x3_fp16](#) ([SippFilter](#) *fptr)
Shave function of the [maxTest3x3_fp16](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuMaxTest3x3_fp16](#))

7.156.1 Detailed Description

This file contains the declaration of the [maxTest3x3_fp16](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.157 MDKdox-Sipp-intro.txt File Reference

7.158 meanStdDev.h File Reference

This file contains the declaration of the [meanStdDev](#) SIPP filter API.

```
#include <sipp.h>
```

Functions

- void [SVU_SYM\(\)](#) [svuMeanStdDev](#) ([SippFilter](#) *fptr)
Parameter structure of the [meanStdDev](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuMeanStdDev](#))

7.158.1 Detailed Description

This file contains the declaration of the [meanStdDev](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.159 minMaxPos.h File Reference

This file contains the declaration of the [Min/Max Value Position](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [MinMaxPosParam](#)
Parameter structure of the [Min/Max Value Position](#) filter.

Functions

- void **SVU_SYM()** **svuMinMaxPos** (SippFilter *fptr)
*Shave function of the **Min/Max Value Position** filter.*
- **SHAVE_SYM_EXPORT** (svuMinMaxPos)

7.159.1 Detailed Description

This file contains the declaration of the **Min/Max Value Position** SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.160 minMaxValue.h File Reference

This file contains the declaration of the **Min/Max Value** SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct **minMaxValParam**
*Parameter structure of the **Min/Max Value** filter.*

Functions

- void **SVU_SYM()** **svuMinMaxValue** (SippFilter *fptr)
*Shave function of the **Min/Max Value** filter.*
- **SHAVE_SYM_EXPORT** (svuMinMaxValue)

7.160.1 Detailed Description

This file contains the declaration of the **Min/Max Value** SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.161 minTest3x3_fp16.h File Reference

This file contains the declaration of the **minTest3x3_fp16** SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [MinTest3x3fp16Param](#)
Parameter structure of the [minTest3x3_fp16](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuMinTest3x3_fp16](#) ([SippFilter](#) *fptr)
Shave function of the [minTest3x3_fp16](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuMinTest3x3_fp16](#))

7.161.1 Detailed Description

This file contains the declaration of the [minTest3x3_fp16](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.162 [mixMedian.h](#) File Reference

This file contains the declaration of the [Mix Median](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [MixMedianParam](#)
Parameter structure of the [Mix Median](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuMixMedian](#) ([SippFilter](#) *fptr)
Shave function of the [Mix Median](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuMixMedian](#))

7.162.1 Detailed Description

This file contains the declaration of the [Mix Median](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.163 monoImbalance.h File Reference

This file contains the declaration of the [MonoImbalance](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [MonoImbalanceParam](#)
Parameter structure of the monoImbalance filter.

Functions

- void [SVU_SYM\(\)](#) [svuMonoImbalance](#) ([SippFilter](#) *fptr)
Shave function of the [MonoImbalance](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuMonoImbalance](#))

7.163.1 Detailed Description

This file contains the declaration of the [MonoImbalance](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.163.2 Function Documentation

[SHAVE_SYM_EXPORT](#) ([svuMonoImbalance](#))

void [SVU_SYM\(\)](#) [svuMonoImbalance](#) ([SippFilter](#) * fptr)

Shave function of the [MonoImbalance](#) filter.

7.164 myriad2SippDefs.inc File Reference

7.165 negative.h File Reference

This file contains the declaration of the [Negative](#) SIPP filter API.

```
#include <sipp.h>
```

Functions

- void [SVU_SYM\(\)](#) [svuNegative](#) ([SippFilter](#) *fptr)
Shave function of the [Negative](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuNegative](#))

7.165.1 Detailed Description

This file contains the declaration of the [Negative](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.166 nonMax3x3Fp32.h File Reference

This file contains the declaration of the [nonMaxFp32](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [nonMax3x3Fp32Param](#)

Functions

- void [SVU_SYM\(\)](#) [svuNonMax3x3Fp32](#) ([SippFilter](#) *fptr)
Shave function of the [RGB to Luma](#) conversion filter.
- [SHAVE_SYM_EXPORT](#) ([svuNonMax3x3Fp32](#))

7.166.1 Detailed Description

This file contains the declaration of the [nonMaxFp32](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.167 nonMax3x3U8.h File Reference

This file contains the declaration of the [nonMax3x3U8](#) SIPP filter API.

```
#include <sipp.h>
```

Functions

- void [SVU_SYM\(\)](#) [svuNonMax3x3U8](#) ([SippFilter](#) *fptr)
Shave function of the [nonMax3x3U8](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuNonMax3x3U8](#))

7.167.1 Detailed Description

This file contains the declaration of the `nonMax3x3U8` SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: `common/license.txt`

7.168 `padBayer5.h` File Reference

```
#include <sipp.h>
#include <sippInternal.h>
```

Functions

- void `SVU_SYM()` `svuPadBayer5` (`SippFilter *fptr`)
- `SHAVE_SYM_EXPORT` (`svuPadBayer5`)
- void `clampInLines` (`SippFilter *fptr`)

7.168.1 Detailed Description

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: `common/license.txt`

7.168.2 Function Documentation

void `clampInLines` (`SippFilter * fptr`)

`SHAVE_SYM_EXPORT` (`svuPadBayer5`)

void `SVU_SYM()` `svuPadBayer5` (`SippFilter * fptr`)

7.169 `padBayer5Frame.h` File Reference

```
#include <sipp.h>
#include <sippInternal.h>
```

Functions

- void `clampInLines` (`SippFilter *fptr`)
- void `SVU_SYM()` `svuPadBayer5Frame` (`SippFilter *fptr`)
Shave function of the `Threshold` filter.
- `SHAVE_SYM_EXPORT` (`svuPadBayer5Frame`)

7.169.1 Detailed Description

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.170 pixelPacker10b.h File Reference

This file contains the declaration of the [Pixel packer](#) SIPP filter API.

```
#include <sipp.h>
```

Functions

- void [SVU_SYM\(\)](#) [svuPixelPacker10b](#) ([SippFilter](#) *fptr)
Shave function of the [Pixel packer](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuPixelPacker10b](#))

7.170.1 Detailed Description

This file contains the declaration of the [Pixel packer](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.171 pixelUnpacker.h File Reference

This file contains the declaration of the [Pixel Unpacker](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [PixelUnpackerParam](#)
Parameter structure of the [Pixel Unpacker](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuPixelUnpacker](#) ([SippFilter](#) *fptr)
Shave function of the [Pixel Unpacker](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuPixelUnpacker](#))

7.171.1 Detailed Description

This file contains the declaration of the [Pixel Unpacker](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.172 pixelUnpackerMipi10b.h File Reference

This file contains the declaration of the [Pixel Unpacker Mipi 10b](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [PixelUnpackerMipi10bParam](#)
Parameter structure of the [Pixel Unpacker Mipi 10b](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuPixelUnpackerMipi10b](#) ([SippFilter](#) *fptr)
Shave function of the [Pixel Unpacker Mipi 10b](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuPixelUnpackerMipi10b](#))

7.172.1 Detailed Description

This file contains the declaration of the [Pixel Unpacker Mipi 10b](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.173 pixelUnpackerWB.h File Reference

This file contains the declaration of the [Pixel Unpacker WB](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [PixelUnpackerWBParam](#)
Parameter structure of the [Pixel Unpacker WB](#) filter.

Functions

- void **SVU_SYM()** **svuPixelUnpackerWB** (**SippFilter** *fptr)
*Shave function of the **Pixel Unpacker WB** filter.*
- **SHAVE_SYM_EXPORT** (**svuPixelUnpackerWB**)

7.173.1 Detailed Description

This file contains the declaration of the **Pixel Unpacker WB** SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.174 positionKernel.h File Reference

This file contains the declaration of the **Pixel Position** SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct **positionKernelParam**
*Parameter structure of the **Pixel Position** filter.*

Functions

- void **SVU_SYM()** **svuPositionKernel** (**SippFilter** *fptr)
*Shave function of the **Pixel Position** filter.*
- **SHAVE_SYM_EXPORT** (**svuPositionKernel**)

7.174.1 Detailed Description

This file contains the declaration of the **Pixel Position** SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.175 purpleFlare.h File Reference

This file contains the declaration of the **purpleFlare** SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [PurpleFlareParam](#)
Parameter structure of the [purpleFlare](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuPurpleFlare](#) ([SippFilter](#) *fptr)
Shave function of the [purpleFlare](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuPurpleFlare](#))

7.175.1 Detailed Description

This file contains the declaration of the [purpleFlare](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.176 pyrDown.h File Reference

This file contains the declaration of the [Pyramid Downscale](#) SIPP filter API.

```
#include <sipp.h>
```

Functions

- void [SVU_SYM\(\)](#) [svuPyrDown](#) ([SippFilter](#) *fptr)
Shave function of the [Pyramid Downscale](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuPyrDown](#))

7.176.1 Detailed Description

This file contains the declaration of the [Pyramid Downscale](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.177 randNoise.h File Reference

This file contains the declaration of the [Random Noise](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [RandNoiseParam](#)
Parameter structure of the [Random Noise](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuGenNoise](#) ([SippFilter](#) *fptr)
Shave function of the [Random Noise](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuGenNoise](#))

7.177.1 Detailed Description

This file contains the declaration of the [Random Noise](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.178 randNoiseFp16.h File Reference

This file contains the declaration of the [Random Noise \(high speed\)](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [RandNoiseFp16Param](#)
Parameter structure of the [Random Noise \(high speed\)](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuGenNoiseFp16](#) ([SippFilter](#) *fptr)
Shave function of the [Random Noise \(high speed\)](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuGenNoiseFp16](#))

7.178.1 Detailed Description

This file contains the declaration of the [Random Noise \(high speed\)](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.179 sad11x11.h File Reference

This file contains the declaration of the [Sum of Absolute Differences 11x11](#) SIPP filter API.

```
#include <sipp.h>
```

Functions

- void [SVU_SYM\(\)](#) [svuSAD11x11](#) ([SippFilter](#) *fptr)
Shave function of the [Sum of Absolute Differences 11x11](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuSAD11x11](#))

7.179.1 Detailed Description

This file contains the declaration of the [Sum of Absolute Differences 11x11](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.180 sad5x5.h File Reference

This file contains the declaration of the [Sum of Absolute Differences 5x5](#) SIPP filter API.

```
#include <sipp.h>
```

Functions

- void [SVU_SYM\(\)](#) [svuSAD5x5](#) ([SippFilter](#) *fptr)
Shave function of the [Sum of Absolute Differences 5x5](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuSAD5x5](#))

7.180.1 Detailed Description

This file contains the declaration of the [Sum of Absolute Differences 5x5](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.181 scale05BilinHV.h File Reference

This file contains the declaration of the [scale05bilinHV](#) SIPP filter API.

```
#include <sipp.h>
```

Functions

- void **SVU_SYM()** **svuScl05BilinHV** (SippFilter *fptr)
Shave function of the [Downscale by 2](#) filter.
- **SHAVE_SYM_EXPORT** (svuScl05BilinHV)

7.181.1 Detailed Description

This file contains the declaration of the scale05bilinHV SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.182 scale05BilinHV_Fp16U8.h File Reference

This file contains the declaration of the [Downscale by 2 \(fp16/u8\)](#) SIPP filter API.

```
#include <sipp.h>
```

Functions

- void **SVU_SYM()** **svuScale05BilinHV_Fp16U8** (SippFilter *fptr)
Shave function of the [Downscale by 2 \(fp16/u8\)](#) filter.
- **SHAVE_SYM_EXPORT** (svuScale05BilinHV_Fp16U8)

7.182.1 Detailed Description

This file contains the declaration of the [Downscale by 2 \(fp16/u8\)](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.183 scale05BilinHVFp16.h File Reference

This file contains the declaration of the [Downscale by 2 \(fp16/fp16\)](#) SIPP filter API.

```
#include <sipp.h>
```

Functions

- void **SVU_SYM()** **svuScale05BilinHVFp16** (SippFilter *fptr)
Shave function of the [Downscale by 2 \(fp16/fp16\)](#) filter.
- **SHAVE_SYM_EXPORT** (svuScale05BilinHVFp16)

7.183.1 Detailed Description

This file contains the declaration of the [Downscale by 2 \(fp16/fp16\)](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.184 scale05Lanc6HV.h File Reference

This file contains the declaration of the [Lanczos Downscale by 2 \(6 taps\)](#) SIPP filter API.

```
#include <sipp.h>
```

Functions

- void [SVU_SYM\(\)](#) [svuScl05Lanc6](#) ([SippFilter](#) *fptr)
Shave function of the [Lanczos Downscale by 2 \(6 taps\)](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuScl05Lanc6](#))

7.184.1 Detailed Description

This file contains the declaration of the [Lanczos Downscale by 2 \(6 taps\)](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.185 scale05Lanc7HV.h File Reference

This file contains the declaration of the [Lanczos Downscale by 2 \(7 taps\)](#) SIPP filter API.

```
#include <sipp.h>
```

Functions

- void [SVU_SYM\(\)](#) [svuScl05Lanc7](#) ([SippFilter](#) *fptr)
Shave function of the [Lanczos Downscale by 2 \(7 taps\)](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuScl05Lanc7](#))

7.185.1 Detailed Description

This file contains the declaration of the [Lanczos Downscale by 2 \(7 taps\)](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.186 scale2xBilinHV.h File Reference

This file contains the declaration of the [Upscale by 2](#) SIPP filter API.

```
#include <sipp.h>
```

Functions

- void [SVU_SYM\(\)](#) [svuScl2xBilinHV](#) (SippFilter *fptr)
Shave function of the [Upscale by 2](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuScl2xBilinHV](#))

7.186.1 Detailed Description

This file contains the declaration of the [Upscale by 2](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.187 scale2xBilinHV_025_075_Fp16ToFp16.h File Reference

This file contains the declaration of the [Upscale by 2 with phases 0.25 and 0.75 fp16 to fp16](#) SIPP filter API.

```
#include <sipp.h>
```

Functions

- void [SVU_SYM\(\)](#) [svuScale2xBilinHV_025_075_Fp16ToFp16](#) (SippFilter *fptr)
Shave function of the [Upscale by 2 with phases 0.25 and 0.75 fp16 to fp16](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuScale2xBilinHV_025_075_Fp16ToFp16](#))

7.187.1 Detailed Description

This file contains the declaration of the [Upscale by 2 with phases 0.25 and 0.75 fp16 to fp16](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.188 scale2xBilinHV_025_075_U16ToU16.h File Reference

This file contains the declaration of the [Upscale by 2 with phases 0.25 and 0.75 u16 to u16](#) SIPP filter API.

```
#include <sipp.h>
```

Functions

- void [SVU_SYM\(\)](#) [svuScale2xBilinHV_025_075_U16ToU16](#) (SippFilter *fptr)
Shave function of the [Upscale by 2 with phases 0.25 and 0.75 u16 to u16](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuScale2xBilinHV_025_075_U16ToU16](#))

7.188.1 Detailed Description

This file contains the declaration of the [Upscale by 2 with phases 0.25 and 0.75 u16 to u16](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.189 scale2xBilinHV_Fp16U8_phase025_075.h File Reference

This file contains the declaration of the [Upscale by 2 with phases 0.25 and 0.75 fp16 to u8](#) SIPP filter API.

```
#include <sipp.h>
```

Functions

- void [SVU_SYM\(\)](#) [svuScale2xBilinHV_Fp16U8_phase025_075](#) (SippFilter *fptr)
Shave function of the [Upscale by 2 with phases 0.25 and 0.75 fp16 to u8](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuScale2xBilinHV_Fp16U8_phase025_075](#))

7.189.1 Detailed Description

This file contains the declaration of the [Upscale by 2 with phases 0.25 and 0.75 fp16 to u8](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.190 scale2xBilinHV_U8ToU8_phase025_075.h File Reference

This file contains the declaration of the [Upscale by 2 with phases 0.25 and 0.75 u8 to u8](#) SIPP filter API.

```
#include <sipp.h>
```

Functions

- void [SVU_SYM\(\)](#) [svuScale2xBilinHV_U8ToU8_phase025_075](#) (SippFilter *fptr)
Shave function of the [Upscale by 2 with phases 0.25 and 0.75 u8 to u8](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuScale2xBilinHV_U8ToU8_phase025_075](#))

7.190.1 Detailed Description

This file contains the declaration of the [Upscale by 2 with phases 0.25 and 0.75 u8 to u8](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.191 scale2xLancH.h File Reference

This file contains the declaration of the [Lanczos Horizontal Upscale by 2](#) SIPP filter API.

```
#include <sipp.h>
```

Functions

- void [SVU_SYM\(\)](#) [svuScl2xLancH](#) (SippFilter *fptr)
Shave function of the [Lanczos Horizontal Upscale by 2](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuScl2xLancH](#))

7.191.1 Detailed Description

This file contains the declaration of the [Lanczos Horizontal Upscale by 2](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.192 scale2xLancHV.h File Reference

This file contains the declaration of the [Lanczos Upscale by 2](#) SIPP filter API.

```
#include <sipp.h>
```

Functions

- void **SVU_SYM()** **svuScl2xLancHV** (SippFilter *fptr)
Shave function of the [Lanczos Upscale by 2](#) filter.
- **SHAVE_SYM_EXPORT** (svuScl2xLancHV)

7.192.1 Detailed Description

This file contains the declaration of the [Lanczos Upscale by 2](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.193 scale2xLancV.h File Reference

This file contains the declaration of the [Lanczos Vertical Upscale by 2](#) SIPP filter API.

```
#include <sipp.h>
```

Functions

- void **SVU_SYM()** **svuScl2xLancV** (SippFilter *fptr)
Shave function of the [Lanczos Vertical Upscale by 2](#) filter.
- **SHAVE_SYM_EXPORT** (svuScl2xLancV)

7.193.1 Detailed Description

This file contains the declaration of the [Lanczos Vertical Upscale by 2](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.194 scaleBilinArb.h File Reference

This file contains the declaration of the [Arbitrary Downscale](#) SIPP filter API.

```
#include <sipp.h>
```

Functions

- void **SVU_SYM()** **svuSclBilinArb** (SippFilter *fptr)
Shave function of the [Arbitrary Downscale](#) filter.
- **SHAVE_SYM_EXPORT** (svuSclBilinArb)

7.194.1 Detailed Description

This file contains the declaration of the [Arbitrary Downscale](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.195 scharr_fp16.h File Reference

This file contains the declaration of the [scharr_fp16](#) SIPP filter API.

```
#include <sipp.h>
```

Functions

- void [SVU_SYM\(\)](#) [svuScharr_fp16](#) ([SippFilter](#) *fptr)
Shave function of the [scharr_fp16](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuScharr_fp16](#))

7.195.1 Detailed Description

This file contains the declaration of the [scharr_fp16](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.196 sipp.h File Reference

SIPP engine.

7.196.1 Detailed Description

SIPP engine.

Copyright

All code copyright Movidius Ltd 2015, all rights reserved. For License Warranty see: [common/license.txt](#)

7.197 sipp_ma2x5x.h File Reference

SIPP engine.

```
#include <sippBaseTypes.h>
#include <sippCfg.h>
#include <sippEvents.h>
#include "DrvSippDefines.h"
#include <sippHwIds.h>
#include <sippPlatform.h>
#include <sippHwBitfieldDefs.h>
#include <sippTypes.h>
#include <sippShaveSym.h>
#include <sippHwDefs.h>
```

Macros

- #define [SIPP_REQ_SW_VIEW](#) (1<<2)
- #define [SIPP_FLAG_DO_H_PADDING](#) (1<<3)
- #define [SIPP_RESIZE](#) (1<<4)
- #define [SIPP_CROP](#) (1<<5)
- #define [SIPP_SHIFT_PLANES](#) (1<<6)
- #define [SIPP_PROVIDE_CHUNK_POS](#) (1<<7)
- #define [PLF_REQUIRES_SW_PADDING](#) (1<<0)
- #define [PLF_UNIQUE_SVU_CODE_SECT](#) (1<<1)
- #define [PLF_IS_FINALIZED](#) (1<<2)
- #define [PLF_MAP_SVU_CODE_IN_DDR](#) (1<<3)
- #define [PLF_RUNS_ITER_GROUPS](#) (1<<4)
- #define [PLF_DISABLE_OPIPE_CONS](#) (1<<5)
- #define [PLF_PROVIDE_RT_STATS](#) (1<<6)
- #define [PLF_ENABLE_SW_QU_USE](#) (1<<7)
- #define [PLF_CONSIDER_OPIPE_RT](#) (1<<8)
- #define [N_PL\(x\)](#) (x)
- #define [BPP\(x\)](#) (x)
- #define [SZ\(x\)](#) sizeof(x)
- #define [SIPP_AUTO](#) (-1)

Functions

- void [sippSetInitMask](#) (u32 mask)
- bool [sippInitialize](#) (void)
- void [sippTerm](#) (void)
- [SippPipeline](#) * [sippCreatePipeline](#) (u32 sliceFirst, u32 sliceLast, u8 *pmBinImg)
- void [sippFinalizePipeline](#) ([SippPipeline](#) *pl)
- [SippFilter](#) * [sippCreateFilter](#) ([SippPipeline](#) *pPipe, u32 flags, u32 out_W, u32 out_H, u32 num_pl, u32 bpp, u32 paramSz, FnSvuRun funcSvuRun, const char *name)
- void [sippFilterSetBufBitsPP](#) ([pSippFilter](#) pFilter, u32 oBufIdx, u32 bitsPerPixel)
- void [sippPipeSetNumLinesPerBuf](#) ([pSippFilter](#) pFilter, u32 oBufIdx, u32 numLines)
- void [sippLinkFilter](#) ([SippFilter](#) *f, [SippFilter](#) *par, u32 vKerSz, u32 hKerSz)
- void [sippLinkFilterSetOBuf](#) ([SippFilter](#) *pFilter, [SippFilter](#) *pParent, u32 parentOBufIdx)
- void [sippFilterAddOBuf](#) ([pSippFilter](#) pFilter, u32 numPlanes, u32 bpp)

- void `sippDeletePipeline` (`SippPipeline` *pPipe)
- void `sippRegisterEventCallback` (`SippPipeline` *pPipe, `sippEventCallback_t` pfCallback)
- void `sippFrameworkRun` (void)
- `s32` `sippAllocCmxMemRegion` (`SippPipeline` *pipe, `SippMemRegion` *memRegList)
- void `sippMemStatus` (void)
- `u32` `sippMemCheck` (`ptSippMCB` pSippMCB, `SippVirtualPool` vPool)
- void `sippChooseMemPool` (`ptSippMCB` pSippMCB, `SippVirtualPool` vPool, `u32` physPoolIdx)
- void `sippProcessFrame` (`SippPipeline` *pl)
- void `sippProcessFrameNB` (`SippPipeline` *pl)
- void `sippProcessIters` (`SippPipeline` *pl, `u32` numIters)
- void `sippProcessItersNB` (`SippPipeline` *pl, `u32` numIters)
- void `sippReschedulePipeline` (`SippPipeline` *pPipe)
- void `sippPipeSetLinesPerIter` (`pSippPipeline` pPipe, `u32` linesPerIter)
- void `sippRdFileU8` (`u8` *buff, int count, const char *fName)
- void `sippWrFileU8` (`u8` *buff, int count, const char *fName)
- void `sippRdFileU8toF16` (half *buff, int count, const char *fName)
- void `sippWrFileF16toU8` (half *buff, int count, const char *fName)
- void `sippTestCrcCheck` (const void *pStart, `u32` lengthBytes, `u32` expectedCrc)
- void `sippErrorSetFatal` (`u32` errCode)
- void `sippError` (`u32` *errStatusMask, `u32` errCode)
- `u32` `sippGetLastError` (void)
- `u32` `sippGetErrorHistory` (`u32` *ptrErrList)
- `u32` `sippPipeGetErrorStatus` (`SippPipeline` *pPipe)
- void `sippDbgLevel` (`SippPipeline` *pl, int level)
- void `sippDbgCompareU8` (`u8` *refA, `u8` *refB, int len)
- void `sippDbgCompareU16` (`u16` *refA, `u16` *refB, int len)
- void `sippDbgCompareU32` (`u32` *refA, `u32` *refB, int len)
- void `sippDbgCompareDeltaU8` (`u8` *refA, `u8` *refB, int len, `u8` delta)
- void `sippSetSliceSize` (`u32` size)
- void `sippUtilComputeFp16Lut` (half(*formula)(half input), half *outLut, `u32` lutSize)
- void `sippUtilComputeFp16LutChannelMode` (half(*formula)(half input), half *outLut, `u32` lutSize)
- void `sippUtilPrintFp16Lut` (half *fp16Lut, `u32` lutSize, const char *fName)
- void `sharpenSigmaToCoefficients` (float sigma, `u16` *coeffs)
- void `lumaGenLut` (float strength, `u8` *lut, int *bitpos)
- `u32` `sippHwChromaDnsCfgReg` (`u32` numHorizPass, `u32` refInputEn, `u32` diffLimit, `u32` forceHorizWeight, `u32` forceVertWeight, `u32` threePlaneModeEn)
- `u32` `sippHwConvolutionCfgReg` (`u32` kernelSz, `u32` oputClamp, `u32` oputAbsVal, `u32` oputSquare, `u32` enAccum, `u32` disableFilter, `u32` accumThresh)
- `u32` `sippHwHarrisCornerCfgReg` (`u32` uKernelSize, `u32` uOutputDeterminant)
- `u32` `sippHwLutCfgReg` (`u32` interpMode, `u32` channelMode, `u32` integerWidth, `u32` numLUTSPerChannel, `u32` numActiveChannels, `u32` enLUTLoad, `u32` apbAccessEn)
- `u32` `sippHwMedianCfgReg` (`s32` nThres, `u32` uKernelSize, `u32` uOSel)
- `u32` `sippHwUnsharpCfgReg` (`u32` uThres, `u32` uKernelSize, `u32` uClampOutput, `u32` uMode, `u32` uOutputDeltas)
- `u32` `getInPtr` (`SippFilter` *fptr, `u32` parent, `u32` iterNo, `u32` lineNo, `u32` planeNo)
- `u32` `getOutPtr` (`SippFilter` *fptr, `u32` iterNo, `u32` planeNo)

- `u32 getPlaneIoPtrs (SippFilter *fptr, u32 parNo, u32 iterNo, u32 planeNo, void *inPtrs)`
- `void getIn3PlanePtr (SippFilter *fptr, u32 parent, u32 iterNo, u32 lineNo, void *out3Ptr)`
- `UInt32 sippFilterGetNumOutPlanes (SippFilter *fptr, UInt32 outBufferIdx)`
- `UInt32 sippFilterGetOutputBpp (SippFilter *fptr)`
- `UInt32 sippFilterGetParentInputLines (SippFilter *fptr, UInt32 parentIdx)`
- `UInt32 sippFilterGetParentPlaneStride (SippFilter *fptr, UInt32 parentIdx)`
- `UInt32 sippFilterGetPlaneStride (SippFilter *fptr, UInt32 outBufferIdx)`
- `UInt32 sippFilterGetLinesThisIter (SippFilter *fptr)`
- `UInt32 sippFilterGetLinesPerIter (SippFilter *fptr)`
- `UInt32 sippFilterGetParentSliceWidth (SippFilter *fptr, UInt32 parentIdx)`
- `UInt32 sippFilterGetParentOutputWidth (SippFilter *fptr, UInt32 parentIdx)`
- `UInt32 sippFilterGetParentOutputHeight (SippFilter *fptr, UInt32 parentIdx)`
- `void sippStopSvus (SippPipeline *pl)`
- `void sippDynRouteIrq (u32 irqSipp0_dynamic, u32 irqSipp1_dynamic, u32 irqSipp2_dynamic)`

7.197.1 Detailed Description

SIPP engine.

Copyright

All code copyright Movidius Ltd 2015, all rights reserved. For License Warranty see: common/license.txt

7.197.2 Macro Definition Documentation

```
#define BPP( x ) (x)
```

```
#define N_PL( x ) (x)
```

Referenced by `createTripleConv3x3()`.

```
#define PLF_CONSIDER_OPIPE_RT (1<<8)
```

Referenced by `sippCoreReschedulePipeline()`.

```
#define PLF_DISABLE_OPIPE_CONS (1<<5)
```

Referenced by `sippIdentifyOPipeSchedulingEntity()`.

```
#define PLF_ENABLE_SW_QU_USE (1<<7)
```

Referenced by `dmaKickSequenceConcurrent()`, `sippCmxDmaDoneIrqHandler()`, `sippCoreFinalisePipeline()`, `sippCQInit()`, `sippGenericLinePrepare()`, and `sippGenericStartUnits()`.

```
#define PLF_IS_FINALIZED (1<<2)
```

Referenced by sippCoreFinalisePipeline(), sippFinalizePipeline(), sippProcessFrame(), and sippProcessFrameNB().

```
#define PLF_MAP_SVU_CODE_IN_DDR (1<<3)
```

Referenced by sippComputeSliceLayout(), and sippSetupSvus().

```
#define PLF_PROVIDE_RT_STATS (1<<6)
```

Referenced by sippCoreFinalisePipeline().

```
#define PLF_REQUIRES_SW_PADDING (1<<0)
```

Referenced by sippBuildLnBufs(), sippCheckIterComplete(), sippGenericStartUnits(), and sippGenericWaitUnits().

```
#define PLF_RUNS_ITER_GROUPS (1<<4)
```

Referenced by sippCoreFinalisePipeline().

```
#define PLF_UNIQUE_SVU_CODE_SECT (1<<1)
```

```
#define SIPP_AUTO (-1)
```

```
#define SIPP_CROP (1<<5)
```

Referenced by sippGenericSchCreateSEFromFilter(), and sippGenericScheduleSetBufConsModels().

```
#define SIPP_FLAG_DO_H_PADDING (1<<3)
```

```
#define SIPP_PROVIDE_CHUNK_POS (1<<7)
```

Referenced by sippComputeBufferProps().

```
#define SIPP_REQ_SW_VIEW (1<<2)
```

Referenced by sippAddFilterToPipe(), sippComputeSwOutCt(), sippGenericLinePrepare(), sippGenericRuntimeFrameReset(), sippInitBufferLnPointers(), and sippLinkFilter().

```
#define SIPP_RESIZE (1<<4)
```

Referenced by sippCoreReInitLineBuffers(), sippGenericSchCreateSEFromOSE(), sippGenericScheduleSetBufConsModels(), and sippLinkFilter().

```
#define SIPP_SHIFT_PLANES (1<<6)
```

Referenced by sippAllocCmxLineBuffersOPipe(), and sippComputeBufferProps().

```
#define SZ( x ) sizeof(x)
```

Referenced by createTripleConv3x3().

7.197.3 Function Documentation

```
void getIn3PlanePtr ( SippFilter * fptr, u32 parent, u32 iterNo, u32 lineNo, void * out3Ptr )
```

Referenced by svuCvtColorRGBtoChromaNV12(), svuCvtColorRGBtoLumaNV12(), svuGenChromaSS(), svuGenLumaU8Fp16(), and svuMixMedian().

```
u32 getInPtr ( SippFilter * fptr, u32 parent, u32 iterNo, u32 lineNo, u32 planeNo )
```

Referenced by svuAbsdiff(), svuAccumulateSquare(), svuAccumulateWeighted(), svuArithmeticAdd(), svuArithmeticAddmask(), svuArithmeticSub(), svuArithmeticSubFp16ToFp16(), svuArithmeticSubmask(), svuAvg(), svuBilateral5x5(), svuBitwiseAnd(), svubitwiseAndMask(), svuBitwiseNot(), svuBitwiseOr(), svuBitwiseOrMask(), svuBitwiseXor(), svuBitwiseXorMask(), svuBoxFilter(), svuCannyEdgeDetection(), svuCensusMatching16(), svuCensusMatching32(), svuCensusMatching64(), svuCensusMatching65(), svuCensusMatchingPyr(), svuCensusMin16(), svuCensusMin64(), svuCensusMin65(), svuCensusMin7(), svuCensusTransform5x5(), svuChannelExtract(), svuChromaBlock(), svuCombDecimDemosaicAwbGains(), svuCombDecimDemosaicAwbGainsStats(), svuContrast(), svuConv11x11(), svuConv1x5Fp16ToFp16(), svuConv1x7(), svuConv1x7Fp16ToFp16(), svuConv1x9(), svuConv5x1Fp16ToFp16(), svuConv7x1(), svuConv7x1Fp16ToFp16(), svuConv7x7Fp16ToU8(), svuConv9x1(), svuConv9x9(), svuConvert16bppTo8bpp(), svuConvertF16ToU8(), svuConvertFrom12BppTo8Bpp(), svuConvertPFp16U16(), svuConvertPU16Fp16(), svuConvertU8ToF16(), svuConvertYUV400ToYUV422(), svuConvGeneric(), svuConvSeparable11x11(), svuConvSeparable11x11Fp16ToFp16(), svuConvSeparable3x3(), svuConvSeparable3x3Fp16ToFp16(), svuConvSeparable5x5(), svuConvSeparable9x9(), svuConvSeparable9x9Fp16ToFp16(), svuCopy(), svuCornerMinEigenVal(), svuCornerMinEigenValpatched(), svuCrop(), svuCropCvtPlaneMode(), svuCvtColorChromaYUVtoNV12(), svucvtColorNV21toRGB(), svuCvtColorRGBfp16ToLumaU8(), svuCvtColorRGBfp16ToUV420U8(), svuCvtColorRGBtoLuma(), svuCvtColorRGBtoUV(), svuCvtColorRGBtoUV420(), svuCvtColorRGBtoYUV422(), svuCvtColorYUV422ToRGB(), svuCvtColorYUVtoRGB(), svuDilate3x3(), svuDilate5x5(), svuDilate7x7(), svuDilateGeneric(), svudisp2depth(), svuEqualizeHist(), svuErode3x3(), svuErode5x5(), svuErode7x7(), svuExtAfStats(), svuExtStatsSatPixelsU32(), svuFast9M2(), svuFast9ScoreCv(), svuGauss(), svuGaussHx2(), svuGaussHx2_fp16(), svuGaussVx2(), svuGaussVx2_fp16(), svuGenChroma(), svuGenChromaSS(), svuGenDnsRef(), svuGenDnsRefFp16(), svuGenLuma(), svuGenNoise(), svuGenNoiseFp16(), svuGreyDesat(), svuHammingDistance(), svuHarrisResponse(), svuHistogram(), svuHistogramStat(), svuIntegralImageSqSumF32M2(), svuIntegralImageSqSumU32M2(), svuIntegralImageSumF32M2(), svuIntegralImageSumU16U32(), svuIntegralImageSumU32M2(), svuInterpolatePixelBilinear(), svuLaplacian3x3(), svuLaplacian5x5(), svuLaplacian5x5Fp16ToFp16(), svuLaplacian7x7(), svuLaplacian7x7Fp16ToFp16(), svuLocalMaxMin3x3_fp16(), svuLocalTM(), svuLowLvlCorr(), svuLumaBlur(), svuLut10to16(), svuLut10to8(), svuLut12to16(), svuLut12to8(), svuLut8to8(), svuLutP10BppU16inU8out(), svuMaxTest3x3_fp16(), svuMeanStdDev(), svuMinMaxPos(), svuMinMaxValue(), svuMinTest3x3_fp16(), svuMixMedian(), svuMonoImbalance(), svuNegative(), svuNonMax3x3Fp32(), svuNonMax3x3U8(),

svuPixelPacker10b(), svuPixelUnpacker(), svuPixelUnpackerMipi10b(), svuPixelUnpackerWB(), svuPositionKernel(), svuPurpleFlare(), svuPyrDown(), svuRgbYuv444(), svuSAD11x11(), svuSAD5x5(), svuScale05BilinHV_Fp16U8(), svuScale05BilinHVFp16(), svuScharr_fp16(), svuScl05Lanc7(), svuScl2xLancHV(), svuScl2xLancV(), svuSclBilinArb(), svusLaplacian3x3Fp16ToFp16(), svuSobel(), svuSSD11x11(), svuSSD5x5(), svuSSD7x7U8ToU32(), svuSsdPointLine7x7U8U32(), svuStartBicubic(), svuStatsAwbSatPixels(), svuStatsAwbSatPixelsU32(), svuSubpixelFilter(), svuThreshold(), svuThresholdBinaryRange(), svuThresholdBinaryU8(), svuThresholdFilter(), svuWhiteBalanceBayerGBRG(), and svuWhiteBalanceRGB().

u32 getOutPtr (SippFilter * fptr, u32 iterNo, u32 planeNo)

Referenced by svuAbsdiff(), svuAccumulateSquare(), svuAccumulateWeighted(), svuArithmeticAdd(), svuArithmeticAddmask(), svuArithmeticSub(), svuArithmeticSubFp16ToFp16(), svuArithmeticSubmask(), svuAvg(), svuBilateral5x5(), svuBitwiseAnd(), svubitwiseAndMask(), svuBitwiseNot(), svuBitwiseOr(), svuBitwiseOrMask(), svuBitwiseXor(), svuBitwiseXorMask(), svuBoxFilter(), svuCannyEdgeDetection(), svuCensusMatching16(), svuCensusMatching32(), svuCensusMatching64(), svuCensusMatching65(), svuCensusMatchingPyr(), svuCensusMin16(), svuCensusMin64(), svuCensusMin65(), svuCensusMin7(), svuCensusTransform5x5(), svuChannelExtract(), svuChromaBlock(), svuCombDecimDemosaicAwbGains(), svuCombDecimDemosaicAwbGainsStats(), svuContrast(), svuConv11x11(), svuConv1x5Fp16ToFp16(), svuConv1x7(), svuConv1x7Fp16ToFp16(), svuConv1x9(), svuConv5x1Fp16ToFp16(), svuConv7x1(), svuConv7x1Fp16ToFp16(), svuConvert16bppTo8bpp(), svuConvertF16ToU8(), svuConvertFrom12BppTo8Bpp(), svuConvertPFp16-U16(), svuConvertPU16Fp16(), svuConvertU8ToF16(), svuConvertYUV400ToYUV422(), svuConvGeneric(), svuConvSeparable11x11(), svuConvSeparable11x11Fp16ToFp16(), svuConvSeparable3x3(), svuConvSeparable3x3Fp16ToFp16(), svuConvSeparable9x9(), svuConvSeparable9x9Fp16ToFp16(), svuCopy(), svuCornerMinEigenVal(), svuCornerMinEigenValpatched(), svuCrop(), svuCropCvtPlaneMode(), svuCvtColorChromaYUVToNV12(), svucvtColorNV21toRGB(), svuCvtColorRGBfp16ToLumaU8(), svuCvtColorRGBfp16ToUV420U8(), svuCvtColorRGBtoChromaNV12(), svuCvtColorRGBtoLuma(), svuCvtColorRGBtoLumaNV12(), svuCvtColorRGBtoUV(), svuCvtColorRGBtoUV420(), svuCvtColorRGBtoYUV422(), svuCvtColorYUV422ToRGB(), svuCvtColorYUVToRGB(), svuDilate3x3(), svuDilate5x5(), svuDilate7x7(), svuDilateGeneric(), svudisp2depth(), svuEqualizeHist(), svuErode3x3(), svuErode5x5(), svuErode7x7(), svuFast9M2(), svuFast9ScoreCv(), svuGauss(), svuGaussHx2(), svuGaussHx2_fp16(), svuGaussVx2(), svuGaussVx2_fp16(), svuGenChroma(), svuGenChromaSS(), svuGenDnsRef(), svuGenDnsRefFp16(), svuGenLuma(), svuGenLumaU8Fp16(), svuGenNoise(), svuGenNoiseFp16(), svuGreyDesat(), svuHammingDistance(), svuHarrisResponse(), svuHistogramStat(), svuHomography(), svuIntegralImageSqSumF32M2(), svuIntegralImageSqSum-U32M2(), svuIntegralImageSumF32M2(), svuIntegralImageSumU16U32(), svuIntegralImageSum-U32M2(), svuInterpolatePixelBilinear(), svuLaplacian3x3(), svuLaplacian5x5(), svuLaplacian5x5-Fp16ToFp16(), svuLaplacian7x7(), svuLaplacian7x7Fp16ToFp16(), svuLocalMaxMin3x3_fp16(), svuLocalTM(), svuLowLvlCorr(), svuLumaBlur(), svuLut10to16(), svuLut10to8(), svuLut12to16(), svuLut12to8(), svuLut8to8(), svuLutP10BppU16inU8out(), svuMaxTest3x3_fp16(), svuMeanStdDev(), svuMinMaxPos(), svuMinMaxValue(), svuMinTest3x3_fp16(), svuMixMedian(), svuMonoImbalance(), svuNegative(), svuNonMax3x3Fp32(), svuNonMax3x3U8(), svuPixelPacker10b(), svuPixelUnpacker(), svuPixelUnpackerMipi10b(), svuPixelUnpackerWB(), svuPositionKernel(), svuPurpleFlare(), svuPyrDown(), svuRgbYuv444(), svuSAD11x11(), svuSAD5x5(), svuScale05BilinHV_Fp16U8(), svuScale05BilinHVFp16(), svuScharr_fp16(), svuScl05Lanc7(), svuScl2xLancHV(), svuScl2xLancV(), svuSclBilinArb(), svusLaplacian3x3Fp16ToFp16(), svuSobel(), svuSSD11x11(), svuSSD5x5(), svuSSD7x7U8ToU32(), svuSsdPointLine7x7U8U32(), svuSubpixelFilter(), svuThreshold(), svuThresholdBinaryRange(), svuThresholdBinaryU8(), svuThresholdFilter(), svuUndistortBrown(), svuWhiteBalanceBayerGBRG(), svuWhiteBalanceRGB(), and svuXYgen().

u32 getPlaneIoPtrs (**SippFilter** * fptr, **u32** parNo, **u32** iterNo, **u32** planeNo, void * inPtrs)

Referenced by svuBoxFilter(), svuBoxFilter11x11(), svuBoxFilter13x13(), svuBoxFilter15x15(), svuBoxFilter3x3(), svuBoxFilter5x5(), svuBoxFilter7x7(), svuBoxFilter9x9(), svuConv15x1(), svuConv1x15(), svuConv1x5(), svuConv3x3(), svuConv3x3Fp16ToFp16(), svuConv5x1(), svuConv5x5(), svuConv5x5Fp16ToFp16(), svuConv7x7(), svuConv7x7Fp16ToFp16(), svuConv7x7Fp16ToU8(), svuConv9x1(), svuConv9x9(), svuConv9x9Fp16ToFp16(), svuConvSeparable5x5(), svuConvSeparable5x5Fp16ToFp16(), svuConvSeparable7x7(), svuConvSeparable7x7Fp16ToFp16(), svuGreyDesat(), svulowLvlCorrMultiplePlanes(), svuLumaBlur(), svuPadBayer5(), svuPadBayer5Frame(), svuScale2xBilinHV_025_075_Fp16ToFp16(), svuScale2xBilinHV_025_075_U16ToU16(), svuScale2xBilinHV_Fp16U8_phase025_075(), svuScale2xBilinHV_U8ToU8_phase025_075(), svuScl05BilinHV(), svuScl05Lanc6(), svuScl2xBilinHV(), and svuScl2xLancH().

void lumaGenLut (float strength, **u8** * lut, int * bitpos)

void sharpenSigmaToCoefficients (float sigma, **u16** * coeffs)

s32 sippAllocCmxMemRegion (**SippPipeline** * pipe, **SippMemRegion** * memRegList)

void sippChooseMemPool (**ptSippMCB** pSippMCB, **SippVirtualPool** vPool, **u32** physPoolIdx)

SippFilter* sippCreateFilter (**SippPipeline** * pPipe, **u32** flags, **u32** out_W, **u32** out_H, **u32** num_pl, **u32** bpp, **u32** paramSz, **FnSvuRun** funcSvuRun, const char * name)

Referenced by createTripleConv3x3().

SippPipeline* sippCreatePipeline (**u32** sliceFirst, **u32** sliceLast, **u8** * pmBinImg)

void sippDbgCompareDeltaU8 (**u8** * refA, **u8** * refB, int len, **u8** delta)

void sippDbgCompareU16 (**u16** * refA, **u16** * refB, int len)

void sippDbgCompareU32 (**u32** * refA, **u32** * refB, int len)

void sippDbgCompareU8 (**u8** * refA, **u8** * refB, int len)

void sippDbgLevel (**SippPipeline** * pl, int level)

void sippDeletePipeline (**SippPipeline** * pPipe)

void sippDynRouteIrq (**u32** irqSipp0_dynamic, **u32** irqSipp1_dynamic, **u32** irqSipp2_dynamic)

void sippError (**u32** * errStatusMask, **u32** errCode)

Referenced by adjustNodesRecursively(), dmaKickSequenceConcurrent(), sippAddFilterToPipe(), sippAssert(), sippComputeBufferProps(), sippCoreFinalisePipeline(), sippCoreReschedulePipeline(), sippCQInit(), sippCreateFilter(), sippDmaCQInit(), sippEventNotify(), sippFilterAddOBuf(), sippGenericAllocRuntimeSched(), sippGenericRecordParentKS(), sippGenericSchedAllocTempStorage(), sippGenericSchedule(), sippIdentifyOPipeSchedulingEntity(), sippInitDma(), sippInitLnMemPoolSlices(), sippInitPhysicalPoolGlobal(), sippInitPhysicalPoolPipe(), sippIssueCommand(), sippKickDmaCQ(),

sippLinkFilter(), sippLinkFilterSetOBuf(), sippOSEAddFilter(), sippOSECreate(), sippOSEProcessFilterIBufs(), sippPipeSessionControl(), and sippProcessFrame().

`void sippErrorSetFatal (u32 errCode)`

Referenced by sippErrorInit().

`void sippFilterAddOBuf (pSippFilter pFilter, u32 numPlanes, u32 bpp)`

`UInt32 sippFilterGetLinesPerIter (SippFilter * fptr)`

Referenced by svuCombDecimDemosaicAwbGains(), svuCombDecimDemosaicAwbGainsStats(), svuGenDnsRef(), svuGenDnsRefFp16(), svuHomography(), svuIntegralImageSqSumF32M2(), svuIntegralImageSqSumU32M2(), svuIntegralImageSumF32M2(), svuIntegralImageSumU16U32(), svuIntegralImageSumU32M2(), svuLocalTM(), svuPadBayer5(), svuPadBayer5Frame(), svuPixelUnpackerMipi10b(), svuPixelUnpackerWB(), svuScale2xBilinHV_025_075_Fp16ToFp16(), svuScale2xBilinHV_025_075_U16ToU16(), svuScale2xBilinHV_Fp16U8_phase025_075(), svuScale2xBilinHV_U8ToU8_phase025_075(), svuScl2xBilinHV(), svuScl2xLancHV(), svuScl2xLancV(), svuSclBilinArb(), svuUndistortBrown(), svuWhiteBalanceBayerGBRG(), and svuXYgen().

`UInt32 sippFilterGetLinesThisIter (SippFilter * fptr)`

Referenced by svuAbsdiff(), svuAccumulateSquare(), svuAccumulateWeighted(), svuArithmeticAdd(), svuArithmeticAddmask(), svuArithmeticSub(), svuArithmeticSubFp16ToFp16(), svuArithmeticSubmask(), svuAvg(), svuBilateral5x5(), svuBitwiseAnd(), svubitwiseAndMask(), svuBitwiseNot(), svuBitwiseOr(), svuBitwiseOrMask(), svuBitwiseXor(), svuBitwiseXorMask(), svuBoxFilter(), svuBoxFilter11x11(), svuBoxFilter13x13(), svuBoxFilter15x15(), svuBoxFilter3x3(), svuBoxFilter5x5(), svuBoxFilter7x7(), svuBoxFilter9x9(), svuCannyEdgeDetection(), svuCensusMatching16(), svuCensusMatching32(), svuCensusMatching64(), svuCensusMatching65(), svuCensusMatchingPyr(), svuCensusMin16(), svuCensusMin64(), svuCensusMin65(), svuCensusMin7(), svuCensusTransform5x5(), svuChannelExtract(), svuChromaBlock(), svuCombDecimDemosaicAwbGains(), svuCombDecimDemosaicAwbGainsStats(), svuContrast(), svuConv11x11(), svuConv15x1(), svuConv1x15(), svuConv1x5(), svuConv1x5Fp16ToFp16(), svuConv1x7(), svuConv1x7Fp16ToFp16(), svuConv1x9(), svuConv3x3(), svuConv3x3Fp16ToFp16(), svuConv5x1(), svuConv5x1Fp16ToFp16(), svuConv5x5(), svuConv5x5Fp16ToFp16(), svuConv7x1(), svuConv7x1Fp16ToFp16(), svuConv7x7(), svuConv7x7Fp16ToFp16(), svuConv7x7Fp16ToU8(), svuConv9x1(), svuConv9x9(), svuConv9x9Fp16ToFp16(), svuConvert16bppTo8bpp(), svuConvertF16ToU8(), svuConvertFrom12BppTo8Bpp(), svuConvertPFp16U16(), svuConvertPU16Fp16(), svuConvertU8ToF16(), svuConvertYUV400ToYUV422(), svuConvGeneric(), svuConvSeparable11x11(), svuConvSeparable11x11Fp16ToFp16(), svuConvSeparable3x3(), svuConvSeparable3x3Fp16ToFp16(), svuConvSeparable5x5(), svuConvSeparable5x5Fp16ToFp16(), svuConvSeparable7x7(), svuConvSeparable7x7Fp16ToFp16(), svuConvSeparable9x9(), svuConvSeparable9x9Fp16ToFp16(), svuCopy(), svuCornerMinEigenVal(), svuCornerMinEigenValpatched(), svuCrop(), svuCropCvtPlaneMode(), svuCvtColorChromaYUVToNV12(), svucvtColorNV21toRGB(), svuCvtColorRGBfp16ToLumaU8(), svuCvtColorRGBfp16ToUV420U8(), svuCvtColorRGBtoChromaNV12(), svuCvtColorRGBtoLuma(), svuCvtColorRGBtoLumaNV12(), svuCvtColorRGBtoUV(), svuCvtColorRGBtoUV420(), svuCvtColorRGBtoYUV422(), svuCvtColorYUV422ToRGB(), svuCvtColorYUVToRGB(), svuDilate3x3(), svuDilate5x5(), svuDilate7x7(), svuDilateGeneric(), svudisp2depth(), svuEqualizeHist(), svuErode3x3(), svuErode5x5(), svuErode7x7(), svuFast9M2(), svuFast9ScoreCv(), svuGauss(), svuGaussHx2(), svuGaussHx2_fp16(), svuGaussVx2(), svuGaussVx2_fp16(), svuGenChroma(), svuGenChromaSS(), svuGenDnsRef(), svuGenDnsRefFp16(), svu-

GenLuma(), svuGenLumaU8Fp16(), svuGenNoise(), svuGenNoiseFp16(), svuGreyDesat(), svuHammingDistance(), svuHarrisResponse(), svuHistogram(), svuHistogramStat(), svuHomography(), svuIntegralImageSqSumF32M2(), svuIntegralImageSqSumU32M2(), svuIntegralImageSumF32M2(), svuIntegralImageSumU16U32(), svuIntegralImageSumU32M2(), svuInterpolatePixelBilinear(), svuLaplacian3x3(), svuLaplacian5x5(), svuLaplacian5x5Fp16ToFp16(), svuLaplacian7x7(), svuLaplacian7x7Fp16ToFp16(), svuLocalMaxMin3x3_fp16(), svuLocalTM(), svuLowLvlCorr(), svuLowLvlCorrMultiplePlanes(), svuLumaBlur(), svuLut10to16(), svuLut10to8(), svuLut12to16(), svuLut12to8(), svuLut8to8(), svuLutP10BppU16inU8out(), svuMaxTest3x3_fp16(), svuMeanStdDev(), svuMinMaxPos(), svuMinMaxValue(), svuMinTest3x3_fp16(), svuMixMedian(), svuMonoImbalance(), svuNegative(), svuNonMax3x3Fp32(), svuNonMax3x3U8(), svuPadBayer5(), svuPadBayer5Frame(), svuPixelPacker10b(), svuPixelUnpacker(), svuPixelUnpackerMipi10b(), svuPixelUnpackerWB(), svuPositionKernel(), svuPurpleFlare(), svuPyrDown(), svuRgbYuv444(), svuSAD11x11(), svuSAD5x5(), svuScale05BilinHV_Fp16U8(), svuScale05BilinHVFp16(), svuScale2xBilinHV_025_075_Fp16ToFp16(), svuScale2xBilinHV_025_075_U16ToU16(), svuScale2xBilinHV_Fp16U8_phase025_075(), svuScale2xBilinHV_U8ToU8_phase025_075(), svuScharr_fp16(), svuScl05BilinHV(), svuScl05Lanc6(), svuScl05Lanc7(), svuScl2xBilinHV(), svuScl2xLancH(), svuScl2xLancHV(), svuScl2xLancV(), svuSclBilinArb(), svuSclLaplacian3x3Fp16ToFp16(), svuSobel(), svuSSD11x11(), svuSSD5x5(), svuSSD7x7U8ToU32(), svuSsdPointLine7x7U8U32(), svuStatsAwbSatPixels(), svuStatsAwbSatPixels-U32(), svuSubpixelFilter(), svuThreshold(), svuThresholdBinaryRange(), svuThresholdBinaryU8(), svuThresholdFilter(), svuUndistortBrown(), svuWhiteBalanceBayerGBRG(), svuWhiteBalanceRGB(), and svuXYgen().

UInt32 sippFilterGetNumOutPlanes (SippFilter * fptr, UInt32 outBufferIdx)

Referenced by subs05sync7(), svuBoxFilter(), svuBoxFilter11x11(), svuBoxFilter13x13(), svuBoxFilter15x15(), svuBoxFilter3x3(), svuBoxFilter5x5(), svuBoxFilter7x7(), svuBoxFilter9x9(), svuConv11x11(), svuConv15x1(), svuConv1x15(), svuConv1x5(), svuConv1x7(), svuConv1x9(), svuConv3x3(), svuConv3x3Fp16ToFp16(), svuConv5x1(), svuConv5x5(), svuConv5x5Fp16ToFp16(), svuConv7x7(), svuConv7x7Fp16ToFp16(), svuConv7x7Fp16ToU8(), svuConv9x1(), svuConv9x9(), svuConv9x9Fp16ToFp16(), svuConvertYUV400ToYUV422(), svuConvSeparable3x3(), svuConvSeparable3x3Fp16ToFp16(), svuConvSeparable5x5(), svuConvSeparable5x5Fp16ToFp16(), svuConvSeparable7x7(), svuConvSeparable7x7Fp16ToFp16(), svuCrop(), svuIntegralImageSqSumF32M2(), svuIntegralImageSqSumU32M2(), svuIntegralImageSumF32M2(), svuIntegralImageSumU16U32(), svuIntegralImageSumU32M2(), svuLowLvlCorrMultiplePlanes(), svuLumaBlur(), svuScale05BilinHV_Fp16U8(), svuScale2xBilinHV_025_075_Fp16ToFp16(), svuScale2xBilinHV_025_075_U16ToU16(), svuScale2xBilinHV_Fp16U8_phase025_075(), svuScale2xBilinHV_U8ToU8_phase025_075(), svuScl05BilinHV(), svuScl05Lanc6(), svuScl2xBilinHV(), svuScl2xLancH(), svuScl2xLancV(), svuSclBilinArb(), upscale2xH(), and upscale2xV().

UInt32 sippFilterGetOutputBpp (SippFilter * fptr)

Referenced by svuCrop(), svuCropCvtPlaneMode(), and svuScl05BilinHV().

UInt32 sippFilterGetParentInputLines (SippFilter * fptr, UInt32 parentIdx)

Referenced by svuHomography(), and svuUndistortBrown().

UInt32 sippFilterGetParentOutputHeight (**SippFilter** * fptr, **UInt32** parentIdx)

Referenced by svuScIbilinArb().

UInt32 sippFilterGetParentOutputWidth (**SippFilter** * fptr, **UInt32** parentIdx)

Referenced by svuScIbilinArb().

UInt32 sippFilterGetParentPlaneStride (**SippFilter** * fptr, **UInt32** parentIdx)

Referenced by subs05sync7(), svuScIbilinArb(), and upscale2xV().

UInt32 sippFilterGetParentSliceWidth (**SippFilter** * fptr, **UInt32** parentIdx)

Referenced by svuCensusMatching16(), svuCensusMatching32(), svuCensusMatching64(), svuCensusMatching65(), and svuCensusMatchingPyr().

UInt32 sippFilterGetPlaneStride (**SippFilter** * fptr, **UInt32** outBufferIdx)

Referenced by subs05sync7(), svuScIbilinArb(), upscale2xH(), and upscale2xV().

void sippFilterSetBufBitsPP (**pSippFilter** pFilter, **u32** oBufIdx, **u32** bitsPerPixel)

void sippFinalizePipeline (**SippPipeline** * pl)

Referenced by sippAllocCmxMemRegion().

void sippFrameworkRun (void)

u32 sippGetErrorHistory (**u32** * ptrErrList)

u32 sippGetLastError (void)

u32 sippHwChromaDnsCfgReg (**u32** numHorizPass, **u32** refInputEn, **u32** diffLimit, **u32** forceHorizWeight, **u32** forceVertWeight, **u32** threePlaneModeEn)

u32 sippHwConvolutionCfgReg (**u32** kernelSz, **u32** oputClamp, **u32** oputAbsVal, **u32** oputSquare, **u32** enAccum, **u32** disableFilter, **u32** accumThresh)

u32 sippHwHarrisCornerCfgReg (**u32** uKernelSize, **u32** uOutputDeterminant)

u32 sippHwLutCfgReg (**u32** interpMode, **u32** channelMode, **u32** integerWidth, **u32** numLUTSPerChannel, **u32** numActiveChannels, **u32** enLUTLoad, **u32** apbAccessEn)

u32 sippHwMedianCfgReg (**s32** nThres, **u32** uKernelSize, **u32** uOSel)

u32 sippHwUnsharpCfgReg (**u32** uThres, **u32** uKernelSize, **u32** uClampOutput, **u32** uMode, **u32** uOutputDeltas)

bool sippInitialize (void)

Referenced by sippPlatformInit().

void sippLinkFilter (**SippFilter** * f, **SippFilter** * par, **u32** vKerSz, **u32** hKerSz)

Referenced by createTripleConv3x3().

void sippLinkFilterSetOBuf (**SippFilter** * pFilter, **SippFilter** * pParent, **u32** parentOBufIdx)

u32 sippMemCheck (**ptSippMCB** pSippMCB, **SippVirtualPool** vPool)

Referenced by sippMemAlloc().

void sippMemStatus (void)

Referenced by sippAllocCmxLineBuffersOPipe().

u32 sippPipeGetErrorStatus (**SippPipeline** * pPipe)

void sippPipeSetLinesPerIter (**pSippPipeline** pPipe, **u32** linesPerIter)

void sippPipeSetNumLinesPerBuf (**pSippFilter** pFilter, **u32** oBufIdx, **u32** numLines)

void sippProcessFrame (**SippPipeline** * pl)

void sippProcessFrameNB (**SippPipeline** * pl)

void sippProcessIters (**SippPipeline** * pl, **u32** numIters)

void sippProcessItersNB (**SippPipeline** * pl, **u32** numIters)

void sippRdFileU8 (**u8** * buff, int count, const char * fName)

void sippRdFileU8toF16 (half * buff, int count, const char * fName)

void sippRegisterEventCallback (**SippPipeline** * pPipe, **sippEventCallback_t** pfCallback)

void sippReschedulePipeline (**SippPipeline** * pPipe)

void sippSetInitMask (**u32** mask)

Referenced by sippPlatformInit().

void sippSetSliceSize (**u32** size)

void sippStopSvus (**SippPipeline** * pl)

Referenced by sippHWSessionRemoveActiveLists().


```
void sippTerm ( void )

void sippTestCrcCheck ( const void * pStart, u32 lengthBytes, u32 expectedCrc )

void sippUtilComputeFp16Lut ( half(*)(half input) formula, half * outLut, u32 lutSize )

void sippUtilComputeFp16LutChannelMode ( half(*)(half input) formula, half * outLut, u32 lutSize
)

void sippUtilPrintFp16Lut ( half * fp16Lut, u32 lutSize, const char * fName )

void sippWrFileF16toU8 ( half * buff, int count, const char * fName )

void sippWrFileU8 ( u8 * buff, int count, const char * fName )
```

7.198 sippAccessScheduler.c File Reference

SIPP framework API Platform(s) supported : MA2x5x.

```
#include <sipp.h>
#include <sippInternal.h>
```

Macros

- #define [UPDATE_PEND_LISTS](#)(uPipeIdx)

Functions

- [bool](#) [sippAccessSchedulerInit](#) ()
- [bool](#) [sippAccessSchedulerQuPush](#) ([u32](#) uPipeIdx, [eSIPP_ACCESS_SCHEDULER_EVENT](#) e-Event, [SIPP_ACCESS_SCHEDULER_EVENT_DATA](#) pData)
- [void](#) [sippAccessSchedulerControl](#) ()
- [u32](#) [sippAccessScheduleCheckPending](#) ()

Variables

- [pSIPP_HW_SESSION](#) pgSippHW
- [tSippFramework](#) gSippFramework

7.198.1 Detailed Description

SIPP framework API Platform(s) supported : MA2x5x.

Copyright

All code copyright Movidius Ltd 2015, all rights reserved. For License Warranty see: common/license.txt

7.198.2 Macro Definition Documentation

#define UPDATE_PEND_LISTS(uPipeIdx)

Value:

```
{ \
    uPipeEventWrIndex[uPipeIdx] == \
    uPipeEventRdIndex[uPipeIdx])\
    { \
        CLR_SW_PEND_LIST(uPipeIdx); \
        CLR_HW_PEND_LIST(uPipeIdx); \
    } \
    else \
    { \
        if ( gSippAccessScheduler.
ePendActions[uPipeIdx][gSippAccessScheduler.uPipeEventRdIndex[uPipeIdx]] &
SIPP_HW_EVENT_FLAG ) \
        { \
            CLR_SW_PEND_LIST(uPipeIdx); \
            SET_HW_PEND_LIST(uPipeIdx); \
        } \
        else \
        { \
            SET_SW_PEND_LIST(uPipeIdx); \
            CLR_HW_PEND_LIST(uPipeIdx); \
        } \
    } \
}
```

7.198.3 Function Documentation

u32 sippAccessScheduleCheckPending (void)

Referenced by sippProcessFrame().

void sippAccessSchedulerControl (void)

Referenced by sippFrameworkRun(), sippIssueCommand(), and sippProcessFrame().

bool sippAccessSchedulerInit (void)

Referenced by sippSWInit().

bool sippAccessSchedulerQuPush (**u32** uPipeIdx, **eSIPP_ACCESS_SCHEDULER_EVENT** eEvent, **SIPP_ACCESS_SCHEDULER_EVENT_DATA** pData)

Referenced by sippEventNotify(), and sippIssueCommand().

7.198.4 Variable Documentation

tSippFramework gSippFramework

pSIPP_HW_SESSION pgSippHW

7.199 sippAccessScheduler.h File Reference

SIPP framework API Platform(s) supported : MA2x5x.

Functions

- `bool sippAccessSchedulerInit (void)`
- `bool sippAccessSchedulerQuPush (u32 uPipeIdx, eSIPP_ACCESS_SCHEDULER_EVENT e-Event, SIPP_ACCESS_SCHEDULER_EVENT_DATA pData)`
- `void sippAccessSchedulerControl (void)`
- `u32 sippAccessScheduleCheckPending (void)`

7.199.1 Detailed Description

SIPP framework API Platform(s) supported : MA2x5x.

Copyright

All code copyright Movidius Ltd 2015, all rights reserved. For License Warranty see: common/license.txt

7.199.2 Function Documentation

u32 sippAccessScheduleCheckPending (void)

Referenced by `sippProcessFrame()`.

void sippAccessSchedulerControl (void)

Referenced by `sippFrameworkRun()`, `sippIssueCommand()`, and `sippProcessFrame()`.

bool sippAccessSchedulerInit (void)

Referenced by `sippSWInit()`.

bool sippAccessSchedulerQuPush (u32 uPipeIdx, eSIPP_ACCESS_SCHEDULER_EVENT eEvent, SIPP_ACCESS_SCHEDULER_EVENT_DATA pData)

Referenced by `sippEventNotify()`, and `sippIssueCommand()`.

7.200 sippAccessSchedulerTypes.h File Reference

SIPP framework API Platform(s) supported : MA2x5x.

```
#include "sippCfg.h"
#include <sippPalTypes.h>
```

Data Structures

- struct [SIPP_ACCESS_SCHEDULER_QU_ENTRY](#)
- struct [SIPP_ACCESS_SCHEDULER_QU](#)
- struct [SIPP_ACCESS_SCHEDULER](#)

Macros

- #define [SIPP_SW_CMD_PENDING](#) gSippAccessScheduler.uSWPendList
- #define [SET_SW_PEND_LIST](#)(uPipeIdx) gSippAccessScheduler.uSWPendList |= (0x1 << uPipeIdx)
- #define [CLR_SW_PEND_LIST](#)(uPipeIdx) gSippAccessScheduler.uSWPendList &= (~(0x1 << uPipeIdx))
- #define [SET_HW_PEND_LIST](#)(uPipeIdx) gSippAccessScheduler.uHWPendList |= (0x1 << uPipeIdx)
- #define [CLR_HW_PEND_LIST](#)(uPipeIdx) gSippAccessScheduler.uHWPendList &= (~(0x1 << uPipeIdx))
- #define [CHK_HW_PEND_LIST](#)(uPipeIdx) gSippAccessScheduler.uHWPendList & (0x1 << uPipeIdx)
- #define [CLR_HW_WAIT_IVAL](#)(uPipeIdx) gSippAccessScheduler.uHWWaitList[uPipeIdx] = 0
- #define [SIPP_HW_EVENT_FLAG](#) 0x10000000

Typedefs

- typedef void * [SIPP_ACCESS_SCHEDULER_EVENT_DATA](#)

Enumerations

- enum [eSIPP_ACCESS_SCHEDULER_EVENT](#) {
[eSIPP_ACCESS_SCHEDULER_CMD_RESET](#) = 0x0, [eSIPP_ACCESS_SCHEDULER_CMD_FINALISE_PIPE](#), [eSIPP_ACCESS_SCHEDULER_CMD_RESCHEDULE_PIPE](#), [eSIPP_ACCESS_SCHEDULER_CMD_TEARDOWN_PIPE](#),
[eSIPP_ACCESS_SCHEDULER_CMD_HWINIT](#) = 0x10000000, [eSIPP_ACCESS_SCHEDULER_CMD_PROCESS_ITERS](#), [eSIPP_ACCESS_SCHEDULER_PIPE_UPDATE_STATUS](#) = 0x80000000, [eSIPP_ACCESS_SCHEDULER_NULL](#) }

7.200.1 Detailed Description

SIPP framework API Platform(s) supported : MA2x5x.

Copyright

All code copyright Movidius Ltd 2015, all rights reserved. For License Warranty see: common/license.txt

7.200.2 Macro Definition Documentation

```
#define CHK_HW_PEND_LIST( uPipeIdx ) gSippAccessScheduler.uHWPendList & ( 0x1 << uPipeIdx)
```

```
#define CLR_HW_PEND_LIST( uPipeIdx ) gSippAccessScheduler.uHWPendList &= ~( 0x1 << uPipeIdx)
```

```
#define CLR_HW_WAIT_IVAL( uPipeIdx ) gSippAccessScheduler.uHWWaitList[uPipeIdx] = 0
```

Referenced by sippAccessSchedulerInit().

```
#define CLR_SW_PEND_LIST( uPipeIdx ) gSippAccessScheduler.uSWPendList &= ~( 0x1 << uPipeIdx)
```

```
#define SET_HW_PEND_LIST( uPipeIdx ) gSippAccessScheduler.uHWPendList |= ( 0x1 << uPipeIdx)
```

```
#define SET_SW_PEND_LIST( uPipeIdx ) gSippAccessScheduler.uSWPendList |= ( 0x1 << uPipeIdx)
```

```
#define SIPP_HW_EVENT_FLAG 0x10000000
```

```
#define SIPP_SW_CMD_PENDING gSippAccessScheduler.uSWPendList
```

Referenced by sippAccessSchedulerControl().

7.200.3 Typedef Documentation

```
typedef void* SIPP_ACCESS_SCHEDULER_EVENT_DATA
```

7.200.4 Enumeration Type Documentation

```
enum eSIPP_ACCESS_SCHEDULER_EVENT
```

Enumerator

```
eSIPP_ACCESS_SCHEDULER_CMD_RESET
eSIPP_ACCESS_SCHEDULER_CMD_FINALISE_PIPE
eSIPP_ACCESS_SCHEDULER_CMD_RESCHEDULE_PIPE
eSIPP_ACCESS_SCHEDULER_CMD_TEARDOWN_PIPE
eSIPP_ACCESS_SCHEDULER_CMD_HWINIT
eSIPP_ACCESS_SCHEDULER_CMD_PROCESS_ITERS
eSIPP_ACCESS_SCHEDULER_PIPE_UPDATE_STATUS
eSIPP_ACCESS_SCHEDULER_NULL
```

7.201 sippAnalysePipema2x5x.c File Reference

Examine a pipeline for validity & features pertinent to implementation Platform(s) supported : MA2x5x.

```
#include <sipp.h>
#include <sippInternal.h>
```

Functions

- void [sippAnalysePipe2x5x](#) (pSippPipeline pPipe)

7.201.1 Detailed Description

Examine a pipeline for validity & features pertinent to implementation Platform(s) supported : MA2x5x.

Copyright

All code copyright Movidius Ltd 2015, all rights reserved. For License Warranty see: common/license.txt

7.201.2 Function Documentation

void [sippAnalysePipe2x5x](#) ([pSippPipeline](#) pPipe)

Referenced by [sippAnalysePipe\(\)](#).

7.202 sippApi.c File Reference

SIPP framework API Platform(s) supported : MA2x5x.

```
#include <sipp.h>
#include <sippInternal.h>
```

Macros

- #define [SIPP_CHECK_VALID_PIPE](#)(pPipe, bValid)

Functions

- [bool](#) [sippInitialize](#) (void)
- void [sippTerm](#) (void)
- [SippPipeline](#) * [sippCreatePipeline](#) (u32 sliceFirst, u32 sliceLast, u8 *pmBinImg)
- void [sippDeletePipeline](#) ([SippPipeline](#) *pPipe)
- void [sippRegisterEventCallback](#) ([SippPipeline](#) *pPipe, [sippEventCallback_t](#) pfCallback)
- [SippFilter](#) * [sippCreateFilter](#) ([pSippPipeline](#) pPipe, u32 flags, u32 out_W, u32 out_H, u32 num_pl, u32 bpp, u32 paramSz, [FnSvuRun](#) funcSvuRun, const char *name)
- void [sippFilterSetBufBitsPP](#) ([pSippFilter](#) pFilter, u32 oBufIdx, u32 bitsPerPixel)
- void [sippLinkFilter](#) ([pSippFilter](#) pFilter, [pSippFilter](#) pParent, u32 vKerSz, u32 hKerSz)
- void [sippLinkFilterSetOBuf](#) ([pSippFilter](#) pFilter, [pSippFilter](#) pParent, u32 parentOBufIdx)
- void [sippFilterAddOBuf](#) ([pSippFilter](#) pFilter, u32 numPlanes, u32 bpp)

- void `sippReschedulePipeline` (`pSippPipeline` pPipe)
- void `sippFinalizePipeline` (`pSippPipeline` pPipe)
- void `sippProcessFrame` (`pSippPipeline` pPipe)
- void `sippProcessFrameNB` (`pSippPipeline` pPipe)
- void `sippFrameworkRun` ()
- void `sippPipeSetLinesPerIter` (`pSippPipeline` pPipe, `u32` linesPerIter)
- void `sippPipeSetNumLinesPerBuf` (`pSippFilter` pFilter, `u32` oBufIdx, `u32` numLines)
- void `sippDynRouteIrq` (`u32` dynamic_irqSipp0, `u32` dynamic_irqSipp1, `u32` dynamic_irqSipp2)

Variables

- `tSippFramework` gSippFramework
- `u32` gSippInitMask = 0xFFFFFFFF
- `u32` gSippSliceSz = SIPP_DEFAULT_SLICE_SIZE
- `pSIPP_HW_SESSION` pgSippHW

7.202.1 Detailed Description

SIPP framework API Platform(s) supported : MA2x5x.

Copyright

All code copyright Movidius Ltd 2015, all rights reserved. For License Warranty see: common/license.txt

7.202.2 Macro Definition Documentation

```
#define SIPP_CHECK_VALID_PIPE( pPipe, bValid )
```

Value:

```
{ \
    u32 uIdx; \
    for (uIdx = 0; uIdx < SIPP_MAX_SUPPORTED_PIPELINES; uIdx++) \
    { \
        if (gSippFramework.pPipelines[uIdx] == ( \
            ptSippPipelineSuper)pPipe) \
        { \
            bValid = TRUE; \
            break; \
        } \
    } \
}
```

Referenced by `sippCreateFilter()`, `sippFinalizePipeline()`, `sippPipeSetLinesPerIter()`, `sippProcessFrame()`, `sippProcessFrameNB()`, `sippRegisterEventCallback()`, and `sippReschedulePipeline()`.

7.202.3 Function Documentation

SippFilter* `sippCreateFilter` (`pSippPipeline` pPipe, `u32` flags, `u32` out_W, `u32` out_H, `u32` num_pl, `u32` bpp, `u32` paramSz, `FnSvuRun` funcSvuRun, `const char *` name)

```

SippPipeline* sippCreatePipeline ( u32 sliceFirst, u32 sliceLast, u8 * pmBinImg )

void sippDeletePipeline ( SippPipeline * pPipe )

void sippDynRouteIrq ( u32 dynamic_irqSipp0, u32 dynamic_irqSipp1, u32 dynamic_irqSipp2 )

void sippFilterAddOBuf ( pSippFilter pFilter, u32 numPlanes, u32 bpp )

void sippFilterSetBufBitsPP ( pSippFilter pFilter, u32 oBufIdx, u32 bitsPerPixel )

void sippFinalizePipeline ( pSippPipeline pPipe )

void sippFrameworkRun ( void )

bool sippInitialize ( void )

```

Referenced by sippPlatformInit().

```

void sippLinkFilter ( pSippFilter pFilter, pSippFilter pParent, u32 vKerSz, u32 hKerSz )

void sippLinkFilterSetOBuf ( pSippFilter pFilter, pSippFilter pParent, u32 parentOBufIdx )

void sippPipeSetLinesPerIter ( pSippPipeline pPipe, u32 linesPerIter )

void sippPipeSetNumLinesPerBuf ( pSippFilter pFilter, u32 oBufIdx, u32 numLines )

void sippProcessFrame ( pSippPipeline pPipe )

void sippProcessFrameNB ( pSippPipeline pPipe )

void sippRegisterEventCallback ( SippPipeline * pPipe, sippEventCallback_t pfCallback )

void sippReschedulePipeline ( pSippPipeline pPipe )

void sippTerm ( void )

```

7.202.4 Variable Documentation

tSippFramework gSippFramework

u32 gSippInitMask = 0xFFFFFFFF

Referenced by sippCmxDmaInit(), and sippSetInitMask().

u32 gSippSliceSz = **SIPP_DEFAULT_SLICE_SIZE**

Referenced by sippCmxDmaInit(), sippCreatePipeline(), sippInitLnMemPool(), sippInitLnMemPoolSlices(), sippInitPhysicalPoolGlobal(), sippInitSchedPool(), sippMemFindMaxLnMemPoolFree(), and sippSetSliceSize().

pSIPP_HW_SESSION pgSippHW

7.203 sippApiInternal.c File Reference

```
#include <sipp.h>
#include <sippInternal.h>
```

Functions

- void **sippHWInit** (void)
- bool **sippSWInit** (void)
- void **sippInitPipeline** (ptSippPipelineSuper ptSPipe, u32 sliceFirst, u32 sliceLast, u32 sliceSize, u8 *mbinImg)
- void **sippAddFilterToPipe** (pSippPipeline pPipe, pSippFilter pFilter, u32 flags, u32 out_W, u32 out_H, u32 num_pl, u32 bpp, FnSvuRun funcSvuRun, const char *name)
- eSIPP_STATUS **sippElaboratePipeline** (pSippPipeline pPipe)
- eSIPP_STATUS **sippTermInternal** (void)
- eSIPP_STATUS **sippFreePipeResource** (pSippPipeline pPipe)
- eSIPP_STATUS **sippRescheduleRequest** (pSippPipeline pPipe)
- eSIPP_STATUS **sippRunItersRequest** (pSippPipeline pPipe, u32 uNumIters)
- eSIPP_STATUS **sippResetFilterVariables** (pSippPipeline pPipe)

Variables

- tSippFramework gSippFramework

7.203.1 Function Documentation

void sippAddFilterToPipe (**pSippPipeline** pPipe, **pSippFilter** pFilter, **u32** flags, **u32** out_W, **u32** out_H, **u32** num_pl, **u32** bpp, **FnSvuRun** funcSvuRun, const char * name)

Referenced by **sippCreateFilter**().

eSIPP_STATUS sippElaboratePipeline (**pSippPipeline** pPipe)

Referenced by **sippFinalizePipeline**(), **sippProcessFrame**(), and **sippProcessFrameNB**().

eSIPP_STATUS sippFreePipeResource (**pSippPipeline** pPipe)

Referenced by **sippDeletePipeline**().

void sippHWInit (void)

Referenced by **sippInitialize**().

```
void sippInitPipeline ( ptSippPipelineSuper ptSPipe, u32 sliceFirst, u32 sliceLast, u32 sliceSize,
u8 * mbinImg )
```

Referenced by sippCreatePipeline().

```
eSIPP_STATUS sippRescheduleRequest ( pSippPipeline pPipe )
```

Referenced by sippReschedulePipeline().

```
eSIPP_STATUS sippResetFilterVariables ( pSippPipeline pPipe )
```

Referenced by sippProcessFrame(), and sippProcessFrameNB().

```
eSIPP_STATUS sippRunItersRequest ( pSippPipeline pPipe, u32 uNumIters )
```

Referenced by sippProcessFrame(), and sippProcessFrameNB().

```
bool sippSWInit ( void )
```

Referenced by sippInitialize().

```
eSIPP_STATUS sippTermInternal ( void )
```

Referenced by sippTerm().

7.203.2 Variable Documentation

```
tSippFramework gSippFramework
```

7.204 sippBaseTypes.h File Reference

```
#include <stdint.h>
```

Macros

- #define **true** 1
- #define **false** 0

Typedefs

- typedef uint8_t **u8**
- typedef int8_t **s8**
- typedef uint16_t **u16**
- typedef int16_t **s16**
- typedef uint32_t **u32**

- typedef int32_t **s32**
- typedef uint64_t **u64**
- typedef int64_t **s64**
- typedef float **fp32**
- typedef **u8** bool
- typedef uint64_t **UInt64**
- typedef uint32_t **UInt32**
- typedef uint16_t **UInt16**
- typedef uint8_t **UInt8**
- typedef int32_t **Int32**
- typedef int16_t **Int16**
- typedef int8_t **Int8**

7.204.1 Macro Definition Documentation

```
#define false 0
```

```
#define true 1
```

7.204.2 Typedef Documentation

```
typedef u8 bool
```

```
typedef float fp32
```

```
typedef int16_t Int16
```

```
typedef int32_t Int32
```

```
typedef int8_t Int8
```

```
typedef int16_t s16
```

```
typedef int32_t s32
```

```
typedef int64_t s64
```

```
typedef int8_t s8
```

```
typedef uint16_t u16
```

```
typedef uint32_t u32
```

```
typedef uint64_t u64
```

```
typedef uint8_t u8
```

```
typedef uint16_t UInt16
```

```
typedef uint32_t UInt32
```

```
typedef uint64_t UInt64
```

```
typedef uint8_t UInt8
```

7.205 sippCfg.h File Reference

```
#include <sippPlatform.h>
```

Macros

- #define SIPP_MAX_SUPPORTED_PIPELINES 8 /* Should we read this in from a cfg struct and then in SWInit */
- #define SIPP_MAX_FILTERS_PER_PIPELINE 32
- #define SIPP_MAX_BUFFERS_PER_PIPELINE (SIPP_MAX_FILTERS_PER_PIPELINE + (SIPP_MAX_FILTERS_PER_PIPELINE >> 1))
- #define SIPP_FILTER_MAX_PARENTS 4
- #define SIPP_FILTER_MAX_CONSUMERS 6
- #define SIPP_MAX_DMA_FILTERS_PER_PIPELINE 32
- #define SIPP_NUM_SVUS 12
- #define SIPP_MAX_SE_PER_PIPE SIPP_MAX_FILTERS_PER_PIPELINE
- #define SIPP_MAX_LINES_PER_ITER 16
- #define SIPP_CDMA_INT_NO 12
- #define SIPP_CDMA_AGENT_NO 0
- #define SIPP_OPIPE_CDMA_AGENT_NO 3
- #define SIPP_NUM_DESCS_PER_CDMA 4
- #define SIPP_MAX_SUPPORTED_HW_PIPELINES SIPP_NUM_IRQ_BARRIERS
- #define SIPP_DEFAULT_SLICE_SIZE (128 << 0xA)
- #define SIPP_CMX_POOL_SZ (192*1024)
- #define SIPP_DDR_POOL_SZ (4*1024*1024)
- #define BUFF_HUGE_SZ 128
- #define SIPP_ERROR_HISTORY_SIZE 16
- #define SIPP_ACCESS_SCHEDULER_QU_SIZE (SIPP_MAX_SUPPORTED_PIPELINES * 8)
- #define SIPP_MAX_EVENTS_PER_PIPE 0x8
- #define SIPP_PAL_MESSAGE_SIZE 0x4
- #define SIPP_THREAD_STACK_SIZE 0x2000
- #define SIPP_THREAD_NAME "/SIPP"
- #define SIPP_THREAD_PRIORITY 225
- #define SIPP_ACCESS_SCHEDULER_QU_NAME "/SIPPQU"
- #define DYNAMIC_IRQ_SIPP_0 IRQ_DYNAMIC_6
- #define DYNAMIC_IRQ_SIPP_1 IRQ_DYNAMIC_7
- #define DYNAMIC_IRQ_SIPP_2 IRQ_DYNAMIC_8

7.205.1 Macro Definition Documentation

```
#define BUFF_HUGE_SZ 128
```

Referenced by sippAddFilterToPipe(), and sippFilterAddOBuf().

```
#define DYNAMIC_IRQ_SIPP_0 IRQ_DYNAMIC_6
```

Referenced by sippHWInit().

```
#define DYNAMIC_IRQ_SIPP_1 IRQ_DYNAMIC_7
```

Referenced by sippHWInit().

```
#define DYNAMIC_IRQ_SIPP_2 IRQ_DYNAMIC_8
```

Referenced by sippHWInit().

```
#define SIPP_ACCESS_SCHEDULER_QU_NAME "/SIPPQU"
```

Referenced by sippAccessSchedulerInit().

```
#define SIPP_ACCESS_SCHEDULER_QU_SIZE ( SIPP_MAX_SUPPORTED_PIPELINES * 8 )
```

Referenced by sippAccessSchedulerInit(), and sippAccessSchedulerQuPush().

```
#define SIPP_CDMA_AGENT_NO 0
```

Referenced by dmaKickSequence(), dmaKickSequenceCQ(), and sippCmxDmaInit().

```
#define SIPP_CDMA_INT_NO 12
```

Referenced by dmaKickSequence(), dmaKickSequenceCQ(), sippCmxDmaDoneIrqHandler(), sippCmxDmaInitAsync(), sippCoreFinalisePipeline(), sippGenericRuntimeClaimHWResource(), sippInitDma(), and topLevelCmxDmaIrqHandler().

```
#define SIPP_CMx_POOL_SZ (192*1024)
```

```
#define SIPP_DDR_POOL_SZ (4*1024*1024)
```

```
#define SIPP_DEFAULT_SLICE_SIZE (128 << 0xA)
```

```
#define SIPP_ERROR_HISTORY_SIZE 16
```

Referenced by sippError(), sippGetErrorHistory(), and sippGetLastError().

```
#define SIPP_FILTER_MAX_CONSUMERS 6
```

Referenced by sippLinkFilter().

```
#define SIPP_FILTER_MAX_PARENTS 4
```

Referenced by sippLinkFilter().

```
#define SIPP_MAX_BUFFERS_PER_PIPELINE (SIPP_MAX_FILTERS_PER_PIPELINE +
(SIPP_MAX_FILTERS_PER_PIPELINE>>1))
```

Referenced by sippAllocCmxLineBuffers().

```
#define SIPP_MAX_DMA_FILTERS_PER_PIPELINE 32
```

Referenced by sippAddFilterToPipe().

```
#define SIPP_MAX_EVENTS_PER_PIPE 0x8
```

Referenced by sippEventNotify().

```
#define SIPP_MAX_FILTERS_PER_PIPELINE 32
```

Referenced by sippAllocCmxLineBuffersOPipe(), sippComputeChunkWidths(), sippCreateFilter(), and sippIdentifyOPipeSchedulingEntity().

```
#define SIPP_MAX_LINES_PER_ITER 16
```

```
#define SIPP_MAX_SE_PER_PIPE SIPP_MAX_FILTERS_PER_PIPELINE
```

```
#define SIPP_MAX_SUPPORTED_HW_PIPELINES SIPP_NUM_IRQ_BARRIERS
```

Referenced by sippHWSessionInit(), sippHWSessionRemoveActiveLists(), sippHWSessionRemoveLoadedPipe(), sippIbflDecHandler(), sippObflIncHandler(), and sippPipeSessionControl().

```
#define SIPP_MAX_SUPPORTED_PIPELINES 8 /* Should we read this in from a cfg struct and then
in SWInit */
```

Referenced by sippAccessSchedulerInit(), sippCallbackInit(), sippCreatePipeline(), sippDeletePipeline(), and sippMemInitVirtPhysMaps().

```
#define SIPP_NUM_DESCS_PER_CDMA 4
```

```
#define SIPP_NUM_SVUS 12
```

Referenced by sippHWSessionInit(), sippHWSessionRemoveActiveLists(), and sippInitPipeline().

```
#define SIPP_OPIPE_CDMA_AGENT_NO 3
```

```
#define SIPP_PAL_MESSAGE_SIZE 0x4
```

Referenced by sippPalQuCreate(), sippPalQuPost(), and sippPalQuReceive().

```
#define SIPP_THREAD_NAME "/SIPP"

#define SIPP_THREAD_PRIORITY 225

#define SIPP_THREAD_STACK_SIZE 0x2000
```

7.206 sippCmxDmaIf.c File Reference

SIPP cmx dma interface - abstracted to this file for ease of switching cmx dma interfacing mechanisms - potentially to use generic driver.

```
#include <sipp.h>
#include <sippInternal.h>
```

Data Structures

- struct [DmaTaskList](#)

Macros

- #define [SIPP_DMA_CQ_ADD_WRITE](#)(quEntry, quNum, address, value64)
TODO - sippDataSectAction uses cmx dma - can we tailor a function??
- #define [DMA_TASK_LIST_SZ](#) 16

Functions

- void [sippRunDmaCQDrain](#) (SippPipeline *pl)
- void [sippCmxDmaInit](#) ()
- void [dmaKickSequenceConcurrent](#) (SippPipeline *pl)
- u32 [sippInitDma](#) (SippFilter *fptr)
- void [sippUpdateDmaAddr](#) (SippFilter *fptr)
- void [dmaKickSequence](#) (SippPipeline *pl)
- void [sippKickDma](#) (SippPipeline *pl)
- u32 [sippUpdateDmaAddrCQ](#) (sSippCdmaQuEntry *pQuEntry, SippFilter *fptr)
- u32 [dmaKickSequenceCQ](#) (sSippCdmaQuEntry *pQuEntry, SippPipeline *pl, u32 iteration)
- void [sippKickDmaCQ](#) (SippPipeline *pl, u32 iteration)
- void [sippRunDmaCQ](#) (SippPipeline *pl)
- u32 [sippDmaCQInit](#) (SippPipeline *pPipe)
- u32 [sippWaitDma](#) ()
- void [sippChainDmaDesc](#) (SippPipeline *pl)
- void [sippCmxDmaDoneIrqHandler](#) ()
- void [topLevelCmxDmaIrqHandler](#) (u32 irqSource)
- void [sippCmxDmaInitAsync](#) ()

Variables

- volatile `u32 dmaIdle = 1`
- `DmaTaskList dmaTaskList`
- `u32 gSippSliceSz`
- `u32 gSippInitMask`

7.206.1 Detailed Description

SIPP cmx dma interface - abstracted to this file for ease of switching cmx dma interfacing mechanisms - potentially to use generic driver.

Copyright

All code copyright Movidius Ltd 2015, all rights reserved. For License Warranty see: common/license.txt

7.206.2 Macro Definition Documentation

```
#define DMA_TASK_LIST_SZ 16
```

```
#define SIPP_DMA_CQ_ADD_WRITE( quEntry, quNum, address, value64 )
```

Value:

```
{ \
    quEntry[quNum].addr = address; \
    quEntry[quNum].value = value64; \
    quNum++; \
}
```

TODO - sippDataSectAction uses cmx dma - can we tailor a function??

Referenced by dmaKickSequenceCQ(), and sippUpdateDmaAddrCQ().

7.206.3 Function Documentation

```
void dmaKickSequence ( SippPipeline * pl )
```

Referenced by dmaKickSequenceConcurrent(), sippCmxDmaDoneIrqHandler(), and sippKickDma().

```
void dmaKickSequenceConcurrent ( SippPipeline * pl )
```

Referenced by sippKickDma(), and sippRunDmaCQ().

```
u32 dmaKickSequenceCQ ( sSippCdmaQuEntry * pQuEntry, SippPipeline * pl, u32 iteration )
```

Referenced by sippKickDmaCQ().

`void sippChainDmaDesc (SippPipeline * pl)`

Referenced by `sippCoreFinalisePipeline()`, and `sippCoreReschedulePipeline()`.

`void sippCmxDmaDoneIrqHandler (void)`

Referenced by `topLevelCmxDmaIrqHandler()`.

`void sippCmxDmaInit (void)`

Referenced by `sippHWSessionInit()`.

`void sippCmxDmaInitAsync (void)`

Referenced by `sippInitDma()`.

u32 `sippDmaCQInit (SippPipeline * pPipe)`

Referenced by `sippCoreFinalisePipeline()`.

u32 `sippInitDma (SippFilter * fptr)`

`void sippKickDma (SippPipeline * pl)`

Referenced by `sippGenericStartUnits()`.

`void sippKickDmaCQ (SippPipeline * pl, u32 iteration)`

Referenced by `sippGenericLinePrepare()`.

`void sippRunDmaCQ (SippPipeline * pl)`

Referenced by `sippGenericStartUnits()`.

`void sippRunDmaCQDrain (SippPipeline * pl)`

Referenced by `dmaKickSequenceConcurrent()`, `sippCmxDmaDoneIrqHandler()`, and `sippRunDmaCQ()`.

`void sippUpdateDmaAddr (SippFilter * fptr)`

Referenced by `sippKickDma()`.

u32 `sippUpdateDmaAddrCQ (sSippCdmaQuEntry * pQuEntry, SippFilter * fptr)`

Referenced by `sippKickDmaCQ()`.

u32 sippWaitDma (void)

Referenced by sippGenericWaitUnits(), and sippKickDma().

void topLevelCmxDmaIrqHandler (**u32** irqSource)

Referenced by sippCmxDmaInit().

7.206.4 Variable Documentation

volatile **u32** dmaIdle = 1

Referenced by dmaKickSequenceConcurrent(), and sippCmxDmaDoneIrqHandler().

DmaTaskList dmaTaskList

u32 gSippInitMask

u32 gSippSliceSz

7.207 sippCoreApi.c File Reference

SIPP engine.

```
#include <sipp.h>
#include <sippInternal.h>
```

Functions

- int **SVU_SYM()** **SHAVE_MAIN** (void)
- void **sippSetSliceSize** (**u32** size)
- void **sippSetInitMask** (**u32** mask)
- void **sippCoreSetPaddingReqs** (**pSippPipeline** pPipe)
- void **sippProcessSchedData** (**pSippPipeline** pPipe)
- **eSIPP_STATUS** **sippCoreFinalisePipeline** (**pSippPipeline** pPipe)
- **eSIPP_STATUS** **sippCoreResourceInit** ()
- void **sippCoreReInitLineBuffers** (**pSippPipeline** pPipe)
- **eSIPP_STATUS** **sippCoreReschedulePipeline** (**pSippPipeline** pPipe)

Variables

- **u32** gSippErrCode
- **SippFilter** * gFakeFptr
- **u32** gSippSliceSz
- **u32** gSippInitMask

7.207.1 Detailed Description

SIPP engine.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.207.2 Function Documentation

`int SVU_SYM() SHAVE_MAIN (void)`

`eSIPP_STATUS sippCoreFinalisePipeline (pSippPipeline pPipe)`

Referenced by sippPipeSessionControl().

`void sippCoreReInitLineBuffers (pSippPipeline pPipe)`

Referenced by sippCoreReschedulePipeline().

`eSIPP_STATUS sippCoreReschedulePipeline (pSippPipeline pPipe)`

Referenced by sippPipeSessionControl().

`eSIPP_STATUS sippCoreResourceInit (void)`

Referenced by sippPipeSessionControlInit().

`void sippCoreSetPaddingReqs (pSippPipeline pPipe)`

Referenced by sippCoreFinalisePipeline().

`void sippProcessSchedData (pSippPipeline pPipe)`

Referenced by sippCoreFinalisePipeline(), and sippCoreReschedulePipeline().

`void sippSetInitMask (u32 mask)`

Referenced by sippPlatformInit().

`void sippSetSliceSize (u32 size)`

7.207.3 Variable Documentation

`SippFilter* gFakeFptr`

u32 gSippErrCode

Referenced by sippError(), sippGetErrorHistory(), and sippGetLastError().

u32 gSippInitMask

Referenced by sippCmxDmaInit(), and sippSetInitMask().

u32 gSippSliceSz

7.208 sippCoreGlobals.c File Reference

SIPP engine.

```
#include <sipp.h>
#include <sippInternal.h>
```

7.208.1 Detailed Description

SIPP engine.

Copyright

All code copyright Movidius Ltd 2015, all rights reserved. For License Warranty see: common/license.txt

7.209 sippCoreHw.c File Reference

```
#include <sipp.h>
#include <sippInternal.h>
```

Functions

- void [sippCoreUnitLoad](#) (SippPipeline *pPipe, **u32** uID)
- void [sippCoreHwInitialLoad](#) (SippPipeline *pPipe)
- void [sippCoreHwInitialSave](#) (SippPipeline *pPipe)

7.209.1 Function Documentation

void [sippCoreHwInitialLoad](#) (**SippPipeline** * pPipe)

void [sippCoreHwInitialSave](#) (**SippPipeline** * pPipe)

void [sippCoreUnitLoad](#) (**SippPipeline** * pPipe, **u32** uID)

Referenced by sippCoreHwInitialLoad().

7.210 sippCoreUtils.c File Reference

SIPP engine.

```
#include <sipp.h>
#include <sippInternal.h>
```

Functions

- int [sippUsingPrecompSched](#) (SippPipeline *pl)
- void [adjustNodesRecursively](#) (SippFilter *fptr)
- void [sippFiltersResetSliceWidths](#) (SippPipeline *pl)
- u32 [sippComputeChunkWidthsSW](#) (SippPipeline *pl)
- void [sippComputeChunkWidths](#) (SippPipeline *pl)
- void [sippComputeBufferProps](#) (SippPipeline *pl)
- void [sippInitBufferLnPointers](#) (SippPipeline *pl)
- void [sippIncrementOutBufs](#) (SippPipeline *pPipe)
- u32 [sippBuildLnBufs](#) (SippPipeline *pl)
- void [sippComputePaddingOffsets](#) (SippPipeline *pl)
- void [sippGetFirstHwFiltIdx](#) (SippPipeline *pl)
- u32 [sippIniHwFilters](#) (SippPipeline *pl)
- void [sippComputeSwOutCt](#) (SippPipeline *pl)
- void [sippInitSyncMutexes](#) (SippPipeline *pl)
- void [sippAsmOptSetup](#) (SippPipeline *pl)
- void [sippComputeSliceLayout](#) (SippPipeline *pl)
- void [sippGetCtxOrder](#) (SippPipeline *pl)
- float [sippGetCoord2](#) (s32 in, float factor, float centreIn, float centreOut)
- void [sippCQInit](#) (SippPipeline *pPipe)

Variables

- u32 [gSippSliceSz](#)

7.210.1 Detailed Description

SIPP engine.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.210.2 Function Documentation

[void adjustNodesRecursively](#) ([SippFilter](#) * fptr)

Referenced by [sippComputeChunkWidthsSW](#)().

`void sippAsmOptSetup (SippPipeline * pl)`

Referenced by `sippCoreFinalisePipeline()`, and `sippCoreReschedulePipeline()`.

`u32 sippBuildLnBufs (SippPipeline * pl)`

Referenced by `sippCoreFinalisePipeline()`, and `sippCoreReschedulePipeline()`.

`void sippComputeBufferProps (SippPipeline * pl)`

Referenced by `sippCoreReschedulePipeline()`.

`void sippComputeChunkWidths (SippPipeline * pl)`

Referenced by `sippComputeBufferProps()`.

`u32 sippComputeChunkWidthsSW (SippPipeline * pl)`

Referenced by `sippComputeChunkWidths()`.

`void sippComputePaddingOffsets (SippPipeline * pl)`

Referenced by `sippCoreFinalisePipeline()`.

`void sippComputeSliceLayout (SippPipeline * pl)`

Referenced by `sippCoreFinalisePipeline()`.

`void sippComputeSwOutCt (SippPipeline * pl)`

Referenced by `sippCoreFinalisePipeline()`, and `sippCoreReschedulePipeline()`.

`void sippCQInit (SippPipeline * pPipe)`

Referenced by `sippCoreFinalisePipeline()`.

`void sippFiltersResetSliceWidths (SippPipeline * pl)`

Referenced by `sippComputeChunkWidthsSW()`.

`float sippGetCoord2 (s32 in, float factor, float centreIn, float centreOut)`

Referenced by `askResizer()`, and `askResizerLatency()`.

```
void sippGetCtxOrder ( SippPipeline * pl )
```

Referenced by sippCoreFinalisePipeline().

```
void sippGetFirstHwFiltIdx ( SippPipeline * pl )
```

Referenced by sippCoreFinalisePipeline().

```
void sippIncrementOutBufs ( SippPipeline * pPipe )
```

Referenced by sippBuildLnBufs().

```
u32 sippIniHwFilters ( SippPipeline * pl )
```

Referenced by sippCoreFinalisePipeline(), and sippCoreReschedulePipeline().

```
void sippInitBufferLnPointers ( SippPipeline * pl )
```

Referenced by sippCoreReschedulePipeline().

```
void sippInitSyncMutexes ( SippPipeline * pl )
```

```
int sippUsingPrecompSched ( SippPipeline * pl )
```

Referenced by sippCoreFinalisePipeline().

7.210.3 Variable Documentation

```
u32 gSippSliceSz
```

7.211 sippDbg.c File Reference

SIPP engine.

```
#include <sipp.h>
#include <sippInternal.h>
```

Functions

- void [sippDbgCompareU8](#) (**u8** *refA, **u8** *refB, int len)
- void [sippDbgCompareDeltaU8](#) (**u8** *refA, **u8** *refB, int len, **u8** delta)
- void [sippDbgCompareU16](#) (**u16** *refA, **u16** *refB, int len)
- void [sippDbgCompareU32](#) (**u32** *refA, **u32** *refB, int len)
- void [sippDbgDumpGraph](#) (**SippPipeline** *pl, const char *fname)
- void [sippDbgDumpRunMask](#) (**u32** mask, int iteration, int dbgDump)
- void [sippDbgDumpSchedForVcsCArr](#) (**SippPipeline** *pl)

- void `sippDbgDumpAsmOffsets` (`SippPipeline` *pl)
- void `sippDbgShowBuffPtr` (`SippFilter` *fptr, const char *buff_name)
- void `sippRdFileU8` (`u8` *buff, int count, const char *fName)
- void `sippWrFileU8` (`u8` *buff, int count, const char *fName)
- void `sippRdFileU8toF16` (half *buff, int count, const char *fName)
- void `sippWrFileF16toU8` (half *buff, int count, const char *fName)
- void `sippDbgCreateDumpFiles` (`SippPipeline` *pl)
- void `sippDbgDumpFilterOuts` (`SippPipeline` *pl)
- void `sippDbgLevel` (`SippPipeline` *pl, int dbgLevel)
- void `sippDumpHtmlMap` (`SippPipeline` *pl)
- void `sippPrintSliceWidth` (`SippPipeline` *pl)
- void `sippTestCrcCheck` (const void *pStart, `u32` lengthBytes, `u32` expectedCrc)

Variables

- `u8` * `sippCmxBase`

7.211.1 Detailed Description

SIPP engine.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.211.2 Function Documentation

void `sippDbgCompareDeltaU8` (`u8` * refA, `u8` * refB, int len, `u8` delta)

void `sippDbgCompareU16` (`u16` * refA, `u16` * refB, int len)

void `sippDbgCompareU32` (`u32` * refA, `u32` * refB, int len)

void `sippDbgCompareU8` (`u8` * refA, `u8` * refB, int len)

void `sippDbgCreateDumpFiles` (`SippPipeline` * pl)

Referenced by `sippGenericRuntimeFrameReset()`.

void `sippDbgDumpAsmOffsets` (`SippPipeline` * pl)

Referenced by `sippCoreFinalisePipeline()`, and `sippCoreReschedulePipeline()`.

void `sippDbgDumpFilterOuts` (`SippPipeline` * pl)

Referenced by `sippGenericRunIterDone()`, and `sippGenericWaitUnits()`.

```
void sippDbgDumpGraph ( SippPipeline * pl, const char * fname )
```

Referenced by sippCoreFinalisePipeline().

```
void sippDbgDumpRunMask ( u32 mask, int iteration, int dbgDump )
```

Referenced by sippGenericLinePrepare().

```
void sippDbgDumpSchedForVcsCArr ( SippPipeline * pl )
```

Referenced by sippCoreFinalisePipeline(), and sippCoreReschedulePipeline().

```
void sippDbgLevel ( SippPipeline * pl, int dbgLevel )
```

```
void sippDbgShowBuffPtr ( SippFilter * fptr, const char * buff_name )
```

```
void sippDumpHtmlMap ( SippPipeline * pl )
```

Referenced by sippCoreFinalisePipeline().

```
void sippPrintSliceWidth ( SippPipeline * pl )
```

```
void sippRdFileU8 ( u8 * buff, int count, const char * fName )
```

```
void sippRdFileU8toF16 ( half * buff, int count, const char * fName )
```

```
void sippTestCrcCheck ( const void * pStart, u32 lengthBytes, u32 expectedCrc )
```

```
void sippWrFileF16toU8 ( half * buff, int count, const char * fName )
```

```
void sippWrFileU8 ( u8 * buff, int count, const char * fName )
```

7.211.3 Variable Documentation

u8* sippCmxBase

Referenced by sippComputeSliceLayout(), sippDumpHtmlMap(), sippInitLnMemPool(), sippInitLnMemPoolSlices(), sippInitPhysicalPoolGlobal(), sippInitSchedPool(), sippMapRegionMapAddrToSliceZero(), sippMapRegionToCmx(), and sippMemLBMatchRegionsToChunks().

7.212 sippDefines.h File Reference

Macros

- #define **SIPP_CONTROL_PIPE_IDX** 0x1F
- #define **SIPP_INVALID_HW_PIPE_ID** 0xFF
- #define **SIPP_INVALID_PARENT_KS** 0xBAD01
- #define **CONCAT3**(A, B, C) A ## B ## C

- #define `I_BASE(X) CONCAT3(SIPP_IBUF, X, _BASE_ADR)`
- #define `I_CFG(X) CONCAT3(SIPP_IBUF, X, _CFG_ADR)`
- #define `I_LS(X) CONCAT3(SIPP_IBUF, X, _LS_ADR)`
- #define `I_PS(X) CONCAT3(SIPP_IBUF, X, _PS_ADR)`
- #define `I_FC(X) CONCAT3(SIPP_IBUF, X, _FC_ADR)`
- #define `O_BASE(X) CONCAT3(SIPP_OBUF, X, _BASE_ADR)`
- #define `O_CFG(X) CONCAT3(SIPP_OBUF, X, _CFG_ADR)`
- #define `O_LS(X) CONCAT3(SIPP_OBUF, X, _LS_ADR)`
- #define `O_FC(X) CONCAT3(SIPP_OBUF, X, _FC_ADR)`
- #define `I_CTX(X) CONCAT3(SIPP_ICTX, X, _ADR)`
- #define `O_CTX(X) CONCAT3(SIPP_OCTX, X, _ADR)`
- #define `I_SHADOW_BASE(X) CONCAT3(SIPP_IBUF, X, _BASE_SHADOW_ADR)`
- #define `I_SHADOW_CFG(X) CONCAT3(SIPP_IBUF, X, _CFG_SHADOW_ADR)`
- #define `I_SHADOW_LS(X) CONCAT3(SIPP_IBUF, X, _LS_SHADOW_ADR)`
- #define `I_SHADOW_PS(X) CONCAT3(SIPP_IBUF, X, _PS_SHADOW_ADR)`
- #define `O_SHADOW_BASE(X) CONCAT3(SIPP_OBUF, X, _BASE_SHADOW_ADR)`
- #define `PROG_IO_BUFF(target, src)`
- #define `SIPP_BUFF_IRQ_RATE_MASK 0x0000000F`
- #define `SIPP_LINE_BUFFER_ALIGNMENT 0x7`
- #define `CMD_H_PAD 0x01`
- #define `CMD_RUN 0x02`
- #define `CMD_EXIT 0x04`
- #define `PFL_SIPP_DONE (0x1 << 0)`
- #define `PFL_DMA_DONE (0x1 << 1)`
- #define `PFL_SVU_DONE (0x1 << 2)`
- #define `PFL_SIPP_EOF_DONE (0x1 << 3)`
- #define `SIPP_ITER_STAT_STARTED_POS 0x0`
- #define `SIPP_ITER_STAT_COMPLETE_POS 0x1`
- #define `SIPP_ITER_STAT_HW_CYCLE_POS 0x2`
- #define `SIPP_ITER_STAT_CDMA_CYCLE_POS 0x3`
- #define `SIPP_ITER_STAT_SW_CYCLE_POS 0x4`
- #define `SIPP_ITER_STAT_ASYNC_START_POS 0x0`
- #define `SIPP_ITER_STAT_ASYNC_HW_POS 0x1`
- #define `SIPP_ITER_STAT_ASYNC_SW_POS 0x2`
- #define `SIPP_ITER_STAT_ASYNC_CDMA_POS 0x3`
- #define `SIPP_ITER_STAT_ASYNC_FW_POS 0x4`
- #define `SIPP_RT_PER_ITER_STATS_SIZE 5`

7.212.1 Macro Definition Documentation

#define `CMD_EXIT 0x04`

Referenced by `SHAVE_MAIN()`, and `sippKickShaveM1PC()`.

#define `CMD_H_PAD 0x01`

Referenced by `sippHorizontalPadding()`, and `sippKickShaveM1PC()`.


```
#define CMD_RUN 0x02
```

Referenced by SHAVE_MAIN(), and sippKickShaveM1PC().

```
#define CONCAT3( A, B, C ) A ## B ## C
```

```
#define I_BASE( X ) CONCAT3(SIPP_IBUF, X, _BASE_ADR)
```

Referenced by sippLoadMipiTx(), and sippLoadSigma().

```
#define I_CFG( X ) CONCAT3(SIPP_IBUF, X, _CFG_ADR)
```

```
#define I_CTX( X ) CONCAT3(SIPP_ICTX, X, _ADR)
```

```
#define I_FC( X ) CONCAT3(SIPP_IBUF, X, _FC_ADR)
```

```
#define I_LS( X ) CONCAT3(SIPP_IBUF, X, _LS_ADR)
```

```
#define I_PS( X ) CONCAT3(SIPP_IBUF, X, _PS_ADR)
```

```
#define I_SHADOW_BASE( X ) CONCAT3(SIPP_IBUF, X, _BASE_SHADOW_ADR)
```

```
#define I_SHADOW_CFG( X ) CONCAT3(SIPP_IBUF, X, _CFG_SHADOW_ADR)
```

```
#define I_SHADOW_LS( X ) CONCAT3(SIPP_IBUF, X, _LS_SHADOW_ADR)
```

```
#define I_SHADOW_PS( X ) CONCAT3(SIPP_IBUF, X, _PS_SHADOW_ADR)
```

```
#define O_BASE( X ) CONCAT3(SIPP_OBUF, X, _BASE_ADR)
```

Referenced by sippLoadMipiRx(), and sippLoadSigma().

```
#define O_CFG( X ) CONCAT3(SIPP_OBUF, X, _CFG_ADR)
```

```
#define O_CTX( X ) CONCAT3(SIPP_OCTX, X, _ADR)
```

```
#define O_FC( X ) CONCAT3(SIPP_OBUF, X, _FC_ADR)
```

Referenced by sippSetOBufLevelsMipiRx0(), sippSetOBufLevelsMipiRx1(), sippSetOBufLevelsMipiRx2(), sippSetOBufLevelsMipiRx3(), and sippSetOBufLevelsSigma().

```
#define O_LS( X ) CONCAT3(SIPP_OBUF, X, _LS_ADR)
```

```
#define O_SHADOW_BASE( X ) CONCAT3(SIPP_OBUF, X, _BASE_SHADOW_ADR)
```

```
#define PFL_DMA_DONE (0x1 << 1)
```

Referenced by sippCheckIterComplete(), and sippCmxDmaDoneIrqHandler().

```
#define PFL_SIPP_DONE (0x1 << 0)
```

Referenced by sippCheckIterComplete(), and sippObflIncHandler().

```
#define PFL_SIPP_EOF_DONE (0x1 << 3)
```

Referenced by sippCheckIterComplete().

```
#define PFL_SVU_DONE (0x1 << 2)
```

Referenced by sippCheckIterComplete(), and sippSvuDoneIrqHandler().

```
#define PROG_IO_BUFF( target, src )
```

Value:

```
SET_REG_WORD((u32)&target->base,      src->base); \
SET_REG_WORD((u32)&target->cfg,      src->cfg); \
SET_REG_WORD((u32)&target->ls,      src->ls); \
SET_REG_WORD((u32)&target->ps,      src->ps); \
SET_REG_WORD((u32)&target->irqRate, src->irqRate);
```

Referenced by sippLoadMipiRx(), sippLoadMipiTx(), and sippLoadSigma().

```
#define SIPP_BUFF_IRQ_RATE_MASK 0x0000000F
```

Referenced by sippBufSetupIrqRate(), and sippBufSetupIrqRateCQ().

```
#define SIPP_CONTROL_PIPE_IDX 0x1F
```

Referenced by sippIssueCommand(), and sippPipeSessionControl().

```
#define SIPP_INVALID_HW_PIPE_ID 0xFF
```

Referenced by sippHWSessionAddActiveLists(), sippHWSessionCommand(), and sippHWSession-RemoveActiveLists().

```
#define SIPP_INVALID_PARENT_KS 0xBAD01
```

Referenced by sippOSEProcessFilterIBufs().

```
#define SIPP_ITER_STAT_ASYNC_CDMA_POS 0x3
```

Referenced by sippCmxDmaDoneIrqHandler().

```
#define SIPP_ITER_STAT_ASYNC_FW_POS 0x4
```

Referenced by sippGenericRunNextIter().

```
#define SIPP_ITER_STAT_ASYNC_HW_POS 0x1
```

Referenced by sippObflIncHandler().

```
#define SIPP_ITER_STAT_ASYNC_START_POS 0x0
```

Referenced by sippGenericRunNextIter().

```
#define SIPP_ITER_STAT_ASYNC_SW_POS 0x2
```

Referenced by sippSvuDoneIrqHandler().

```
#define SIPP_ITER_STAT_CDMA_CYCLE_POS 0x3
```

Referenced by sippGenericWaitUnits().

```
#define SIPP_ITER_STAT_COMPLETE_POS 0x1
```

Referenced by sippGenericRuntimeHWProcessIters().

```
#define SIPP_ITER_STAT_HW_CYCLE_POS 0x2
```

Referenced by sippGenericWaitUnits().

```
#define SIPP_ITER_STAT_STARTED_POS 0x0
```

Referenced by sippGenericRuntimeHWProcessIters().

```
#define SIPP_ITER_STAT_SW_CYCLE_POS 0x4
```

Referenced by sippGenericWaitUnits().

```
#define SIPP_LINE_BUFFER_ALIGNMENT 0x7
```

Referenced by adjustNodesRecursively(), and sippComputeChunkWidthsSW().

```
#define SIPP_RT_PER_ITER_STATS_SIZE 5
```

Referenced by sippCmxDmaDoneIrqHandler(), sippGenericRunNextIter(), sippGenericRuntimeHWProcessIters(), sippGenericSchedule(), sippGenericWaitUnits(), sippObflIncHandler(), and sippSvuDoneIrqHandler().

7.213 sippDriverCmxDmaIf.c File Reference

SIPP cmx dma driver interface.

```
#include <sipp.h>
#include <sippInternal.h>
#include <DrvCmxDma.h>
```

7.213.1 Detailed Description

SIPP cmx dma driver interface.

Copyright

All code copyright Movidius Ltd 2015, all rights reserved. For License Warranty see: common/license.txt

7.214 sippError.c File Reference

SIPP framework Error management With a concurrent pipeline API, the minimum aim is that an error on one pipeline does NOT effect other pipelines also running at that time.

```
#include <sipp.h>
#include <sippInternal.h>
```

Functions

- void [sippErrorInit](#) ()
- void [sippError](#) (u32 *errStatusMask, u32 errCode)
- void [sippAssert](#) (u32 condition, u32 errCode)
- void [sippErrorSetFatal](#) (u32 errCode)
- u32 [sippGetLastError](#) ()
- u32 [sippPipeGetErrorStatus](#) (SippPipeline *pPipe)
- u32 [sippGetErrorHistory](#) (u32 *ptrErrList)

Variables

- u32 [gSippErrCode](#) [SIPP_ERROR_HISTORY_SIZE]
- u32 [gSippErrWrIdx](#)
- u32 [gSippErrRdIdx](#)
- u32 [gSippFatalErrList](#) [SIPP_ERROR_MASK_SIZE]

7.214.1 Detailed Description

SIPP framework Error management With a concurrent pipeline API, the minimum aim is that an error on one pipeline does NOT effect other pipelines also running at that time.

Copyright

All code copyright Movidius Ltd 2015, all rights reserved. For License Warranty see: common/license.txt

Platform(s) supported : MA2150

7.214.2 Function Documentation

`void sippAssert (u32 condition, u32 errCode)`

Referenced by svuBoxFilter().

`void sippError (u32 * errStatusMask, u32 errCode)`

Referenced by adjustNodesRecursively(), dmaKickSequenceConcurrent(), sippAddFilterToPipe(), sippAssert(), sippComputeBufferProps(), sippCoreFinalisePipeline(), sippCoreReschedulePipeline(), sippCQInit(), sippCreateFilter(), sippDmaCQInit(), sippEventNotify(), sippFilterAddOBuf(), sippGenericAllocRuntimeSched(), sippGenericRecordParentKS(), sippGenericSchedAllocTempStorage(), sippGenericSchedule(), sippIdentifyOPipeSchedulingEntity(), sippInitDma(), sippInitLnMemPoolSlices(), sippInitPhysicalPoolGlobal(), sippInitPhysicalPoolPipe(), sippIssueCommand(), sippKickDmaCQ(), sippLinkFilter(), sippLinkFilterSetOBuf(), sippOSEAddFilter(), sippOSECreate(), sippOSEProcessFilterIBufs(), sippPipeSessionControl(), and sippProcessFrame().

`void sippErrorInit (void)`

Referenced by sippSWInit().

`void sippErrorSetFatal (u32 errCode)`

Referenced by sippErrorInit().

`u32 sippGetErrorHistory (u32 * ptrErrList)`

`u32 sippGetLastError (void)`

`u32 sippPipeGetErrorStatus (SippPipeline * pPipe)`

7.214.3 Variable Documentation

`u32 gSippErrCode[SIPP_ERROR_HISTORY_SIZE]`

Referenced by sippError(), sippGetErrorHistory(), and sippGetLastError().

`u32 gSippErrRdIdx`

Referenced by sippErrorInit(), and sippGetErrorHistory().

u32 gSippErrWrIdx

Referenced by sippError(), sippErrorInit(), sippGetErrorHistory(), and sippGetLastError().

u32 gSippFatalErrList[SIPP_ERROR_MASK_SIZE]

Referenced by sippError(), and sippErrorSetFatal().

7.215 sippEvents.h File Reference

```
#include <sippBaseTypes.h>
```

Data Structures

- struct **SIPP_PIPELINE_FINALISED_DATA**

Typedefs

- typedef void **SIPP_PIPELINE_EVENT_DATA**
- typedef struct **SIPP_PIPELINE_FINALISED_DATA * pSIPP_PIPELINE_FINALISED_DATA**

Enumerations

- enum **eSIPP_PIPELINE_EVENT** {
eSIPP_PIPELINE_FINALISED = 0x0, **eSIPP_PIPELINE_RESCHEDULED**, **eSIPP_PIPELINE_FRAME_DONE**, **eSIPP_PIPELINE_ITERS_DONE**,
eSIPP_PIPELINE_SYNC_OP_DONE, **eSIPP_PIPELINE_STARTED** }

7.215.1 Typedef Documentation

typedef struct **SIPP_PIPELINE_FINALISED_DATA * pSIPP_PIPELINE_FINALISED_DATA**

typedef void **SIPP_PIPELINE_EVENT_DATA**

7.215.2 Enumeration Type Documentation

enum **eSIPP_PIPELINE_EVENT**

Enumerator

eSIPP_PIPELINE_FINALISED
eSIPP_PIPELINE_RESCHEDULED
eSIPP_PIPELINE_FRAME_DONE
eSIPP_PIPELINE_ITERS_DONE
eSIPP_PIPELINE_SYNC_OP_DONE
eSIPP_PIPELINE_STARTED

7.216 sippFastExeUpd.h File Reference

SIPP core : fast update of filters execution numbers.

7.216.1 Detailed Description

SIPP core : fast update of filters execution numbers.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.217 sippFilterAccesors.c File Reference

```
#include <sipp.h>
#include <sippInternal.h>
```

Functions

- [u32 sippFilterGetNumOutPlanes \(SippFilter *fptr, u32 outBufferIdx\)](#)
- [u32 sippFilterGetOutputBpp \(SippFilter *fptr\)](#)
- [u32 sippFilterGetParentInputLines \(SippFilter *fptr, u32 parentIdx\)](#)
- [u32 sippFilterGetParentPlaneStride \(SippFilter *fptr, u32 parentIdx\)](#)
- [u32 sippFilterGetPlaneStride \(SippFilter *fptr, u32 outBufferIdx\)](#)
- [u32 sippFilterGetLinesThisIter \(SippFilter *fptr\)](#)
- [u32 sippFilterGetLinesPerIter \(SippFilter *fptr\)](#)
- [u32 sippFilterGetParentSliceWidth \(SippFilter *fptr, u32 parentIdx\)](#)
- [u32 sippFilterGetParentOutputWidth \(SippFilter *fptr, u32 parentIdx\)](#)
- [u32 sippFilterGetParentOutputHeight \(SippFilter *fptr, u32 parentIdx\)](#)

7.217.1 Function Documentation

u32 sippFilterGetLinesPerIter (SippFilter * fptr)

Referenced by `svuCombDecimDemosaicAwbGains()`, `svuCombDecimDemosaicAwbGainsStats()`, `svuGenDnsRef()`, `svuGenDnsRefFp16()`, `svuHomography()`, `svuIntegralImageSqSumF32M2()`, `svuIntegralImageSqSumU32M2()`, `svuIntegralImageSumF32M2()`, `svuIntegralImageSumU16U32()`, `svuIntegralImageSumU32M2()`, `svuLocalTM()`, `svuPadBayer5()`, `svuPadBayer5Frame()`, `svuPixelUnpackerMipi10b()`, `svuPixelUnpackerWB()`, `svuScale2xBilinHV_025_075_Fp16ToFp16()`, `svuScale2xBilinHV_025_075_U16ToU16()`, `svuScale2xBilinHV_Fp16U8_phase025_075()`, `svuScale2xBilinHV_U8ToU8_phase025_075()`, `svuScl2xBilinHV()`, `svuScl2xLancHV()`, `svuScl2xLancV()`, `svuSclBilinArb()`, `svuUndistortBrown()`, `svuWhiteBalanceBayerGBRG()`, and `svuXYgen()`.

u32 sippFilterGetLinesThisIter (SippFilter * fptr)

Referenced by svuAbsdiff(), svuAccumulateSquare(), svuAccumulateWeighted(), svuArithmeticAdd(), svuArithmeticAddmask(), svuArithmeticSub(), svuArithmeticSubFp16ToFp16(), svuArithmeticSubmask(), svuAvg(), svuBilateral5x5(), svuBitwiseAnd(), svubitwiseAndMask(), svuBitwiseNot(), svuBitwiseOr(), svuBitwiseOrMask(), svuBitwiseXor(), svuBitwiseXorMask(), svuBoxFilter(), svuBoxFilter11x11(), svuBoxFilter13x13(), svuBoxFilter15x15(), svuBoxFilter3x3(), svuBoxFilter5x5(), svuBoxFilter7x7(), svuBoxFilter9x9(), svuCannyEdgeDetection(), svuCensusMatching16(), svuCensusMatching32(), svuCensusMatching64(), svuCensusMatching65(), svuCensusMatchingPyr(), svuCensusMin16(), svuCensusMin64(), svuCensusMin65(), svuCensusMin7(), svuCensusTransform5x5(), svuChannelExtract(), svuChromaBlock(), svuCombDecimDemosaicAwbGains(), svuCombDecimDemosaicAwbGainsStats(), svuContrast(), svuConv11x11(), svuConv15x1(), svuConv1x15(), svuConv1x5(), svuConv1x5Fp16ToFp16(), svuConv1x7(), svuConv1x7Fp16ToFp16(), svuConv1x9(), svuConv3x3(), svuConv3x3Fp16ToFp16(), svuConv5x1(), svuConv5x1Fp16ToFp16(), svuConv5x5(), svuConv5x5Fp16ToFp16(), svuConv7x1(), svuConv7x1Fp16ToFp16(), svuConv7x7(), svuConv7x7Fp16ToFp16(), svuConv7x7Fp16ToU8(), svuConv9x1(), svuConv9x9(), svuConv9x9Fp16ToFp16(), svuConvert16bppTo8bpp(), svuConvertF16ToU8(), svuConvertFrom12BppTo8Bpp(), svuConvertPFp16-U16(), svuConvertPU16Fp16(), svuConvertU8ToF16(), svuConvertYUV400ToYUV422(), svuConvGeneric(), svuConvSeparable11x11(), svuConvSeparable11x11Fp16ToFp16(), svuConvSeparable3x3(), svuConvSeparable3x3Fp16ToFp16(), svuConvSeparable5x5(), svuConvSeparable5x5Fp16ToFp16(), svuConvSeparable7x7(), svuConvSeparable7x7Fp16ToFp16(), svuConvSeparable9x9(), svuConvSeparable9x9Fp16ToFp16(), svuCopy(), svuCornerMinEigenVal(), svuCornerMinEigenValpatched(), svuCrop(), svuCropCvtPlaneMode(), svuCvtColorChromaYUVToNV12(), svucvtColorNV21toRGB(), svuCvtColorRGBfp16ToLumaU8(), svuCvtColorRGBfp16ToUV420U8(), svuCvtColorRGBtoChromaNV12(), svuCvtColorRGBtoLuma(), svuCvtColorRGBtoLumaNV12(), svuCvtColorRGBtoUV(), svuCvtColorRGBtoUV420(), svuCvtColorRGBtoYUV422(), svuCvtColorYUV422ToRGB(), svuCvtColorYUVToRGB(), svuDilate3x3(), svuDilate5x5(), svuDilate7x7(), svuDilateGeneric(), svudisp2depth(), svuEqualizeHist(), svuErode3x3(), svuErode5x5(), svuErode7x7(), svuFast9M2(), svuFast9ScoreCv(), svuGauss(), svuGaussHx2(), svuGaussHx2_fp16(), svuGaussVx2(), svuGaussVx2_fp16(), svuGenChroma(), svuGenChromaSS(), svuGenDnsRef(), svuGenDnsRefFp16(), svuGenLuma(), svuGenLumaU8Fp16(), svuGenNoise(), svuGenNoiseFp16(), svuGreyDesat(), svuHammingDistance(), svuHarrisResponse(), svuHistogram(), svuHistogramStat(), svuHomography(), svuIntegralImageSqSumF32M2(), svuIntegralImageSqSumU32M2(), svuIntegralImageSumF32M2(), svuIntegralImageSumU16U32(), svuIntegralImageSumU32M2(), svuInterpolatePixelBilinear(), svuLaplacian3x3(), svuLaplacian5x5(), svuLaplacian5x5Fp16ToFp16(), svuLaplacian7x7(), svuLaplacian7x7Fp16ToFp16(), svuLocalMaxMin3x3_fp16(), svuLocalTM(), svuLowLvlCorr(), svulowLvlCorrMultiplePlanes(), svuLumaBlur(), svuLut10to16(), svuLut10to8(), svuLut12to16(), svuLut12to8(), svuLut8to8(), svuLutP10BppU16inU8out(), svuMaxTest3x3_fp16(), svuMeanStdDev(), svuMinMaxPos(), svuMinMaxValue(), svuMinTest3x3_fp16(), svuMixMedian(), svuMonoImbalance(), svuNegative(), svuNonMax3x3Fp32(), svuNonMax3x3U8(), svuPadBayer5(), svuPadBayer5Frame(), svuPixelPacker10b(), svuPixelUnpacker(), svuPixelUnpackerMipi10b(), svuPixelUnpackerWB(), svuPositionKernel(), svuPurpleFlare(), svuPyrDown(), svuRgbYuv444(), svuSAD11x11(), svuSAD5x5(), svuScale05BilinHV_Fp16U8(), svuScale05BilinHVFp16(), svuScale2xBilinHV_025_075_Fp16ToFp16(), svuScale2xBilinHV_025_075_U16ToU16(), svuScale2xBilinHV_Fp16U8_phase025_075(), svuScale2xBilinHV_U8ToU8_phase025_075(), svuScharr_fp16(), svuScl05BilinHV(), svuScl05Lanc6(), svuScl05Lanc7(), svuScl2xBilinHV(), svuScl2xLancH(), svuScl2xLancHV(), svuScl2xLancV(), svuSclBilinArb(), svusLaplacian3x3Fp16ToFp16(), svuSobel(), svuSSD11x11(), svuSSD5x5(), svuSSD7x7U8ToU32(), svuSsdPointLine7x7U8U32(), svuStatsAwbSatPixels(), svuStatsAwbSatPixels-U32(), svuSubpixelFilter(), svuThreshold(), svuThresholdBinaryRange(), svuThresholdBinaryU8(), svuThresholdFilter(), svuUndistortBrown(), svuWhiteBalanceBayerGBRG(), svuWhiteBalanceRGB(), and svuXYgen().

u32 sippFilterGetNumOutPlanes (SippFilter * fptr, u32 outBufferIdx)

Referenced by subs05sync7(), svuBoxFilter(), svuBoxFilter11x11(), svuBoxFilter13x13(), svuBoxFilter15x15(), svuBoxFilter3x3(), svuBoxFilter5x5(), svuBoxFilter7x7(), svuBoxFilter9x9(), svuConv11x11(), svuConv15x1(), svuConv1x15(), svuConv1x5(), svuConv1x7(), svuConv1x9(), svuConv3x3(), svuConv3x3Fp16ToFp16(), svuConv5x1(), svuConv5x5(), svuConv5x5Fp16ToFp16(), svuConv7x7(), svuConv7x7Fp16ToFp16(), svuConv7x7Fp16ToU8(), svuConv9x1(), svuConv9x9(), svuConv9x9Fp16ToFp16(), svuConvertYUV400ToYUV422(), svuConvSeparable3x3(), svuConvSeparable3x3Fp16ToFp16(), svuConvSeparable5x5(), svuConvSeparable5x5Fp16ToFp16(), svuConvSeparable7x7(), svuConvSeparable7x7Fp16ToFp16(), svuCrop(), svuIntegralImageSqSumF32M2(), svuIntegralImageSqSumU32M2(), svuIntegralImageSumF32M2(), svuIntegralImageSumU16U32(), svuIntegralImageSumU32M2(), svulowLvlCorrMultiplePlanes(), svuLumaBlur(), svuScale05BilinHV_Fp16U8(), svuScale2xBilinHV_025_075_Fp16ToFp16(), svuScale2xBilinHV_025_075_U16ToU16(), svuScale2xBilinHV_Fp16U8_phase025_075(), svuScale2xBilinHV_U8ToU8_phase025_075(), svuScI05BilinHV(), svuScI05Lanc6(), svuScI2xBilinHV(), svuScI2xLancH(), svuScI2xLancV(), svuScIBilinArb(), upscale2xH(), and upscale2xV().

u32 sippFilterGetOutputBpp (SippFilter * fptr)

Referenced by svuCrop(), svuCropCvtPlaneMode(), and svuScI05BilinHV().

u32 sippFilterGetParentInputLines (SippFilter * fptr, u32 parentIdx)

Referenced by svuHomography(), and svuUndistortBrown().

u32 sippFilterGetParentOutputHeight (SippFilter * fptr, u32 parentIdx)

Referenced by svuScIBilinArb().

u32 sippFilterGetParentOutputWidth (SippFilter * fptr, u32 parentIdx)

Referenced by svuScIBilinArb().

u32 sippFilterGetParentPlaneStride (SippFilter * fptr, u32 parentIdx)

Referenced by subs05sync7(), svuScIBilinArb(), and upscale2xV().

u32 sippFilterGetParentSliceWidth (SippFilter * fptr, u32 parentIdx)

Referenced by svuCensusMatching16(), svuCensusMatching32(), svuCensusMatching64(), svuCensusMatching65(), and svuCensusMatchingPyr().

u32 sippFilterGetPlaneStride (SippFilter * fptr, u32 outBufferIdx)

Referenced by subs05sync7(), svuScIBilinArb(), upscale2xH(), and upscale2xV().

7.218 sippGenericRuntime.c File Reference

SIPP engine.

```
#include <sipp.h>
#include <sippInternal.h>
```

Functions

- void [sippGenericStartHWUnits2x5x](#) ([SippPipeline](#) *pPipe)
- void [sippGenericLinePrepare](#) ([SippPipeline](#) *pl, int iteration)
- void [sippGenericRuntimeFrameReset](#) ([SippPipeline](#) *pl)
- void [sippGenericStartUnits](#) ([SippPipeline](#) *pl)
- void [sippGenericUpdateExecNums](#) ([SippPipeline](#) *pl)
- void [sippGenericWaitUnits](#) ([SippPipeline](#) *pPipe)
- void [sippGenericRuntimeLoadPipeline](#) ([SippPipeline](#) *pPipe)
- void [sippGenericRuntimeProcessIters](#) ([SippPipeline](#) *pPipe, [u32](#) numIters)
- void [sippGenericRuntimeClaimHWResource](#) ([pSippPipeline](#) pPipe)
- void [sippGenericRuntime](#) ([pSippPipeline](#) pPipe, [eSIPP_ACCESS_SCHEDULER_EVENT](#) e-Event, [SIPP_ACCESS_SCHEDULER_EVENT_DATA](#) pData)
- [u32](#) [sippGenericRunIterDone](#) ([SippPipeline](#) *pPipe)
- void [sippGenericRunNextIter](#) ([pSippPipeline](#) pPipe)

Variables

- [tSippFramework](#) gSippFramework
- [u32](#) [sippGlobalOBFLIncStatus](#)

7.218.1 Detailed Description

SIPP engine.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.218.2 Function Documentation

[void sippGenericLinePrepare \(\[SippPipeline\]\(#\) * pl, int iteration \)](#)

Referenced by [sippGenericRunNextIter\(\)](#), [sippGenericRuntimeHWProcessIters\(\)](#), and [sippGenericRuntimeProcessIters\(\)](#).

[u32 sippGenericRunIterDone \(\[SippPipeline\]\(#\) * pPipe \)](#)

[void sippGenericRunNextIter \(\[pSippPipeline\]\(#\) pPipe \)](#)

Referenced by [sippGenericRuntimeProcessIters\(\)](#).

```
void sippGenericRuntime ( pSippPipeline pPipe, eSIPP_ACCESS_SCHEDULER_EVENT eEvent,
SIPP_ACCESS_SCHEDULER_EVENT_DATA pData )
```

```
void sippGenericRuntimeClaimHWResource ( pSippPipeline pPipe )
```

```
void sippGenericRuntimeFrameReset ( SippPipeline * pl )
```

```
void sippGenericRuntimeLoadPipeline ( SippPipeline * pPipe )
```

Referenced by sippGenericRuntimeProcessIters().

```
void sippGenericRuntimeProcessIters ( SippPipeline * pPipe, u32 numIters )
```

Referenced by sippGenericRuntime().

```
void sippGenericStartHWUnits2x5x ( SippPipeline * pPipe )
```

Referenced by sippGenericStartUnits().

```
void sippGenericStartUnits ( SippPipeline * pl )
```

Referenced by sippGenericRunNextIter(), and sippGenericRuntimeHWProcessIters().

```
void sippGenericUpdateExecNums ( SippPipeline * pl )
```

Referenced by sippGenericRunIterDone(), and sippGenericWaitUnits().

```
void sippGenericWaitUnits ( SippPipeline * pPipe )
```

Referenced by sippGenericRuntimeHWProcessIters().

7.218.3 Variable Documentation

tSippFramework gSippFramework

u32 sippGlobalOBFLIncStatus

Referenced by sippGenericRuntimeLoadPipeline(), sippHWSessionRemoveActiveLists(), sippIntBarrierSetup(), and sippObflIncHandler().

7.219 sippGenericRuntimeema2x5x.c File Reference

SIPP engine.

```
#include <sipp.h>
#include <sippInternal.h>
```

Functions

- void `sippGenericWaitUnits` (`SippPipeline *pl`)
- void `sippGenericStartHWUnits2x5x` (`SippPipeline *pPipe`)
- void `sippGenericBlockHWUnits2x5x` (`SippPipeline *pPipe`)
- void `sippGenericUpdateHWUnits2x5x` (`SippPipeline *pPipe`)
- void `sippGenericRuntimeHWProcessIters` (`pSippPipeline pPipe`, `u32 numIters`)

7.219.1 Detailed Description

SIPP engine.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: `common/license.txt`

7.219.2 Function Documentation

`void sippGenericBlockHWUnits2x5x (SippPipeline * pPipe)`

`void sippGenericRuntimeHWProcessIters (pSippPipeline pPipe, u32 numIters)`

Referenced by `sippGenericRuntimeProcessIters()`.

`void sippGenericStartHWUnits2x5x (SippPipeline * pPipe)`

Referenced by `sippGenericStartUnits()`.

`void sippGenericUpdateHWUnits2x5x (SippPipeline * pPipe)`

Referenced by `sippGenericRunIterDone()`, and `sippGenericRuntimeHWProcessIters()`.

`void sippGenericWaitUnits (SippPipeline * pl)`

Referenced by `sippGenericRuntimeHWProcessIters()`.

7.220 sippGenericSchApi.c File Reference

```
#include <sipp.h>
#include <sippInternal.h>
```

Functions

- `s32 sippGenericSearchBuffer` (`psSchLineBuffer psLineBuf`, `u32 lineRequired`, `s32 linePos`, `u32 fullSearch`)

- void sippGenericRecordParentKS (SippSchEnt *pSchEnt, u32 iBufIdx)
- RunStatus sippGenericCheckSERunParents (SippSchEnt *pSchEnt)
- s32 sippGenericGetBufferEntry (psSchLineBuffer psLineBuf, s32 linePos)
- void sippGenericSchedPipeInit (pSippPipeline pipeLine)
- u32 sippGenericAllocRuntimeSched (pSippPipeline pipeLine)
- void sippGenericSchCreateSEFromFilter (SippFilter *pFilter)
- void sippGenericSchCreateSEFromOSE (pSippOse pOSE)
- u32 sippGenericSchedAllocTempStorage (pSippPipeline pipeLine)
- u32 sippGenericSchedInitTempStorage (pSippPipeline pipeLine)
- u32 sippGenericSchedule (pSippPipeline pipeLine, bool allocMem, bool reschedPipe)
- void sippGenericScheduleSetBufConsModels (pSippPipeline pipeLine)

7.220.1 Function Documentation

u32 sippGenericAllocRuntimeSched (**pSippPipeline** pipeLine)

Referenced by sippGenericSchedule().

RunStatus sippGenericCheckSERunParents (**SippSchEnt** * pSchEnt)

s32 sippGenericGetBufferEntry (**psSchLineBuffer** psLineBuf, **s32** linePos)

void sippGenericRecordParentKS (**SippSchEnt** * pSchEnt, **u32** iBufIdx)

Referenced by sippGenericCheckSERunParents().

void sippGenericSchCreateSEFromFilter (**SippFilter** * pFilter)

Referenced by sippGenericSchedInitTempStorage().

void sippGenericSchCreateSEFromOSE (pSippOse pOSE)

Referenced by sippGenericSchedInitTempStorage().

u32 sippGenericSchedAllocTempStorage (**pSippPipeline** pipeLine)

Referenced by sippGenericSchedule().

u32 sippGenericSchedInitTempStorage (**pSippPipeline** pipeLine)

Referenced by sippGenericSchedule().

void sippGenericSchedPipeInit (**pSippPipeline** pipeLine)

Referenced by sippGenericSchedule().

```
u32 sippGenericSchedule ( pSippPipeline pipeLine, bool allocMem, bool reschedPipe )
```

```
void sippGenericScheduleSetBufConsModels ( pSippPipeline pipeLine )
```

```
s32 sippGenericSearchBuffer ( psSchLineBuffer psLineBuf, u32 lineRequired, s32 linePos, u32 fullSearch )
```

Referenced by sippGenericCheckSERunParents(), and sippGenericRecordParentKS().

7.221 sippGenericSchDebug.c File Reference

```
#include <sipp.h>
#include <sippInternal.h>
```

Functions

- void **sippGenericDbgPrintRunnable** (**SippSchEnt** *SEs[], **u32** nSE, **u32** iteration)
- void **sippGenericDbgShowBufferReq** (**SippSchEnt** *SEs[], **u32** nSE)

7.221.1 Function Documentation

```
void sippGenericDbgPrintRunnable ( SippSchEnt * SEs[], u32 nSE, u32 iteration )
```

```
void sippGenericDbgShowBufferReq ( SippSchEnt * SEs[], u32 nSE )
```

7.222 sippGenericSchReq.c File Reference

SIPP engine.

```
#include <sipp.h>
#include <sippInternal.h>
```

Functions

- **u8** **askRegular** (**SippSchEnt** *pSE, **u32** iBufId, **s32** iterationNum, **u32** linesPerIter)
- **u8** **askRegularLatency** (**SippSchEnt** *pSE, **u32** iBufId, **s32** iterationNum, **u32** linesPerIter)
- **u8** **askCrop** (**SippSchEnt** *pSE, **u32** iBufId, **s32** iterationNum, **u32** linesPerIter)
- **u8** **askCropLatency** (**SippSchEnt** *pSE, **u32** iBufId, **s32** iterationNum, **u32** linesPerIter)
- **u8** **askHwColorCombChroma** (**SippSchEnt** *pSE, **u32** iBufId, **s32** iterationNum, **u32** linesPerIter)
- **u8** **askResizer** (**SippSchEnt** *pSE, **u32** iBufId, **s32** iterationNum, **u32** linesPerIter)
- **u8** **askPolyFirResizer** (**SippSchEnt** *pSE, **u32** iBufId, **s32** iterationNum, **u32** linesPerIter)
- **u8** **askChromaGenDownsizer** (**SippSchEnt** *pSE, **u32** iBufId, **s32** iterationNum, **u32** linesPerIter)
- **u8** **askResizerLatency** (**SippSchEnt** *pSE, **u32** iBufId, **s32** iterationNum, **u32** linesPerIter)
- **u8** **askHwMedLumaLatency** (**SippSchEnt** *pSE, **u32** iBufId, **s32** iterationNum, **u32** linesPerIter)
- void **sippGenericDMACheck** (void *pPipe, void *pSE, void *pParam)
- void **sippGenericColorCombChromaCheck** (void *pPipe, void *pSE, void *pParam)

7.222.1 Detailed Description

SIPP engine.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.222.2 Function Documentation

u8 askChromaGenDownsizer (**SippSchEnt** * pSE, **u32** iBufId, **s32** iterationNum, **u32** linesPerIter)

Referenced by sippGenericScheduleSetBufConsModels().

u8 askCrop (**SippSchEnt** * pSE, **u32** iBufId, **s32** iterationNum, **u32** linesPerIter)

Referenced by sippGenericScheduleSetBufConsModels().

u8 askCropLatency (**SippSchEnt** * pSE, **u32** iBufId, **s32** iterationNum, **u32** linesPerIter)

u8 askHwColorCombChroma (**SippSchEnt** * pSE, **u32** iBufId, **s32** iterationNum, **u32** linesPerIter)

Referenced by sippGenericSchCreateSEFromFilter(), sippGenericSchCreateSEFromOSE(), and sippGenericScheduleSetBufConsModels().

u8 askHwMedLumaLatency (**SippSchEnt** * pSE, **u32** iBufId, **s32** iterationNum, **u32** linesPerIter)

Referenced by sippGenericScheduleSetBufConsModels().

u8 askPolyFirResizer (**SippSchEnt** * pSE, **u32** iBufId, **s32** iterationNum, **u32** linesPerIter)

Referenced by sippGenericScheduleSetBufConsModels().

u8 askRegular (**SippSchEnt** * pSE, **u32** iBufId, **s32** iterationNum, **u32** linesPerIter)

Referenced by sippGenericScheduleSetBufConsModels().

u8 askRegularLatency (**SippSchEnt** * pSE, **u32** iBufId, **s32** iterationNum, **u32** linesPerIter)

u8 askResizer (**SippSchEnt** * pSE, **u32** iBufId, **s32** iterationNum, **u32** linesPerIter)

Referenced by sippGenericScheduleSetBufConsModels().

u8 askResizerLatency (**SippSchEnt** * pSE, **u32** iBufId, **s32** iterationNum, **u32** linesPerIter)

```
void sippGenericColorCombChromaCheck ( void * pPipe, void * pSE, void * pParam )
```

Referenced by sippGenericSchCreateSEFromFilter(), and sippGenericSchCreateSEFromOSE().

```
void sippGenericDMACheck ( void * pPipe, void * pSE, void * pParam )
```

Referenced by sippGenericSchCreateSEFromFilter().

7.223 sippGenericSchWrite.c File Reference

SIPP engine.

```
#include <sipp.h>
#include <sippInternal.h>
```

Functions

- void [sippGenericSchedWr](#) ([SippPipeline](#) *pl, [u32](#) iteration)

7.223.1 Detailed Description

SIPP engine.

Copyright

All code copyright Movidius Ltd 2015, all rights reserved. For License Warranty see: common/license.txt

7.223.2 Function Documentation

```
void sippGenericSchedWr ( SippPipeline * pl, u32 iteration )
```

7.224 sippHeap.c File Reference

```
#include <sipp.h>
#include <sippInternal.h>
```

Macros

- #define [SIPP_CFG_MAX_NUM_HEAPS](#) 0x2
- #define [MCB_MAGIC](#) 0x42434D4C
- #define [SIPP_HEAP_DEFRAG_COUNT](#) 0x5

Functions

- void `sippHeapInit` (`tsSippHeap` *pSippHeap, `u8` *start_address, `u32` size)
- `SippHeapCB` * `sippHeapAlloc` (`tsSippHeap` *pSippHeap, `u32` size)
- void `sippHeapDefrag` (`tsSippHeap` *pSippHeap)
- `tsSippHeap` * `sippHeapCreate` (`u8` *sippHeap_start, `u32` sippHeap_size)
- void * `sippAlloc` (`tsSippHeap` *pSippHeap, void *pPrev, `u32` size)
- void `sippFree` (`tsSippHeap` *pSippHeap, void *pPtr)
- void `sippFreeList` (`tsSippHeap` *pSippHeap, void *pStartPtr)
- `u32` `sippSizeList` (`tsSippHeap` *pSippHeap, void *pStartPtr)
- `u32` `sippHeapCheck` (`tsSippHeap` *pSippHeap)

Variables

- `tsSippHeap` `sippHeaps` [`SIPP_CFG_MAX_NUM_HEAPS`]

7.224.1 Detailed Description

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.224.2 Macro Definition Documentation

```
#define MCB_MAGIC 0x42434D4C
```

Referenced by `sippHeapAlloc()`, `sippHeapCheck()`, and `sippHeapInit()`.

```
#define SIPP_CFG_MAX_NUM_HEAPS 0x2
```

Referenced by `sippHeapCreate()`.

```
#define SIPP_HEAP_DEFRAG_COUNT 0x5
```

Referenced by `sippFree()`.

7.224.3 Function Documentation

```
void* sippAlloc ( tsSippHeap * pSippHeap, void * pPrev, u32 size )
```

Referenced by `sippMemAlloc()`.

```
void sippFree ( tsSippHeap * pSippHeap, void * pPtr )
```

Referenced by `sippFreeList()`.

```
void sippFreeList ( tsSippHeap * pSippHeap, void * pStartPtr )
```

Referenced by sippMemFree(), and sippMemFreeList().

```
SippHeapCB* sippHeapAlloc ( tsSippHeap * pSippHeap, u32 size )
```

Referenced by sippAlloc().

```
u32 sippHeapCheck ( tsSippHeap * pSippHeap )
```

Referenced by sippMemCheck(), and sippMemStatus().

```
tsSippHeap* sippHeapCreate ( u8 * sippHeap_start, u32 sippHeap_size )
```

Referenced by sippInitPhysicalPoolGlobal().

```
void sippHeapDefrag ( tsSippHeap * pSippHeap )
```

Referenced by sippAlloc(), and sippFree().

```
void sippHeapInit ( tsSippHeap * pSippHeap, u8 * start_address, u32 size )
```

Referenced by sippHeapCreate().

```
u32 sippSizeList ( tsSippHeap * pSippHeap, void * pStartPtr )
```

7.224.4 Variable Documentation

```
tsSippHeap sippHeaps[SIPP_CFG_MAX_NUM_HEAPS]
```

7.225 sippHpad.c File Reference

SIPP engine.

```
#include <sipp.h>
#include <sippInternal.h>
```

Functions

- `int scGetShaveNumber ()`
- `void memsetBpp (u8 *i_dest, u8 *i_source, u32 i_padding, u32 i_bpp)`
- `void sippHorizontalPadding (SippPipeline *pl)`

Variables

- `SippPipeline * sipp_pl`
- `u32 dbg_svu_no`

7.225.1 Detailed Description

SIPP engine.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.225.2 Function Documentation

`void memsetBpp (u8 * i_dest, u8 * i_source, u32 i_padding, u32 i_bpp)`

Referenced by `sippHorizontalPadding()`.

`int scGetShaveNumber ()`

`void sippHorizontalPadding (SippPipeline * pl)`

Referenced by `SHAVE_MAIN()`.

7.225.3 Variable Documentation

`u32 dbg_svu_no`

Referenced by `scGetShaveNumber()`, and `sippKickShaveM1PC()`.

`SippPipeline* sipp_pl`

Referenced by `sippKickSvus()`.

7.226 sippHwBitfieldDefs.h File Reference

```
#include "sippPlatform.h"
```

Macros

- #define `CONV_CFG_KERNEL_SIZE_MASK` 0x7
- #define `CONV_CFG_KERNEL_SIZE_SHIFT` 0x0
- #define `CONV_CFG_OUTPUT_CLAMP_EN_MASK` 0x1
- #define `CONV_CFG_OUTPUT_CLAMP_EN_SHIFT` 0x3
- #define `CONV_CFG_OUTPUT_ABS_VAL_MASK` 0x1
- #define `CONV_CFG_OUTPUT_ABS_VAL_SHIFT` 0x4
- #define `CONV_CFG_OUTPUT_SQUARE_EN_MASK` 0x1
- #define `CONV_CFG_OUTPUT_SQUARE_EN_SHIFT` 0x5
- #define `CONV_CFG_ACCUMULATION_EN_MASK` 0x1

- #define CONV_CFG_ACCUMULATION_EN_SHIFT 0x6
- #define CONV_CFG_FILTER_DISABLE_MASK 0x1
- #define CONV_CFG_FILTER_DISABLE_SHIFT 0x7
- #define CONV_CFG_ACCUM_THRESHOLD_MASK 0xFFFF
- #define CONV_CFG_ACCUM_THRESHOLD_SHIFT 0x8
- #define CONV_CFG_EVENODD_PIX_MASK 0x1
- #define CONV_CFG_EVENODD_PIX_SHIFT 0x18
- #define CONV_CFG_EVENODD_LINE_MASK 0x1
- #define CONV_CFG_EVENODD_LINE_SHIFT 0x19
- #define CONV_KERNEL_SIZE(val) ((val & CONV_CFG_KERNEL_SIZE_MASK) << CONV_CFG_KERNEL_SIZE_SHIFT)
- #define CONV_OUTPUT_CLAMP(val) ((val & CONV_CFG_OUTPUT_CLAMP_EN_MASK) << CONV_CFG_OUTPUT_CLAMP_EN_SHIFT)
- #define CONV_OUTPUT_ABS_VAL(val) ((val & CONV_CFG_OUTPUT_ABS_VAL_MASK) << CONV_CFG_OUTPUT_ABS_VAL_SHIFT)
- #define CONV_OUTPUT_SQUARE(val) ((val & CONV_CFG_OUTPUT_SQUARE_EN_MASK) << CONV_CFG_OUTPUT_SQUARE_EN_SHIFT)
- #define CONV_ACCUMULATION_EN(val) ((val & CONV_CFG_ACCUMULATION_EN_MASK) << CONV_CFG_ACCUMULATION_EN_SHIFT)
- #define CONV_FILTER_DISABLE(val) ((val & CONV_CFG_FILTER_DISABLE_MASK) << CONV_CFG_FILTER_DISABLE_SHIFT)
- #define CONV_ACCUM_THRESHOLD(val) ((val & CONV_CFG_ACCUM_THRESHOLD_MASK) << CONV_CFG_ACCUM_THRESHOLD_SHIFT)
- #define CONV_EVENODD_PIX(val) ((val & CONV_CFG_EVENODD_PIX_MASK) << CONV_CFG_EVENODD_PIX_SHIFT)
- #define CONV_EVENODD_LINE(val) ((val & CONV_CFG_EVENODD_LINE_MASK) << CONV_CFG_EVENODD_LINE_SHIFT)
- #define CONV_KSZ_3x3 CONV_KERNEL_SIZE(0x3)
- #define CONV_KSZ_5x5 CONV_KERNEL_SIZE(0x5)
- #define CONV_CFG(kernSize, outClamp, outAbsVal, outSquare, accEn, filterDis, accTresh, evenOddPix, evenOddLine)
- #define DEBAYER_CFG_BAYER_ORDER_MASK 0x3
- #define DEBAYER_CFG_BAYER_ORDER_SHIFT 0x0
- #define DEBAYER_CFG_LUMA_ONLY_MASK 0x1
- #define DEBAYER_CFG_LUMA_ONLY_SHIFT 0x2
- #define DEBAYER_CFG_FORCE_RB_ZERO_MASK 0x1
- #define DEBAYER_CFG_FORCE_RB_ZERO_SHIFT 0x3
- #define DEBAYER_CFG_IP_DATAWIDTH_M1_MASK 0xF
- #define DEBAYER_CFG_IP_DATAWIDTH_M1_SHIFT 0x4
- #define DEBAYER_CFG_OP_DATAWIDTH_M1_MASK 0xF
- #define DEBAYER_CFG_OP_DATAWIDTH_M1_SHIFT 0x8
- #define DEBAYER_CFG_OUTPLANE_ORDER_MASK 0x7
- #define DEBAYER_CFG_OUTPLANE_ORDER_SHIFT 0xC
- #define DEBAYER_CFG_NUM_OUTPLANES_MASK 0x3
- #define DEBAYER_CFG_NUM_OUTPLANES_SHIFT 0xF
- #define DEBAYER_CFG_GRAD_MULTIPLIER_MASK 0xFF
- #define DEBAYER_CFG_GRAD_MULTIPLIER_SHIFT 0x18

- #define DEBAYER_BAYER_ORDER(val) ((val & DEBAYER_CFG_BAYER_ORDER_MASK) << DEBAYER_CFG_BAYER_ORDER_SHIFT)
- #define DEBAYER_LUMA_ONLY(val) ((val & DEBAYER_CFG_LUMA_ONLY_MASK) << DEBAYER_CFG_LUMA_ONLY_SHIFT)
- #define DEBAYER_FORCE_RB_ZERO(val) ((val & DEBAYER_CFG_FORCE_RB_ZERO_MASK) << DEBAYER_CFG_FORCE_RB_ZERO_SHIFT)
- #define DEBAYER_IP_DATAWIDTH(val) ((val & DEBAYER_CFG_IP_DATAWIDTH_M1_MASK) << DEBAYER_CFG_IP_DATAWIDTH_M1_SHIFT)
- #define DEBAYER_OP_DATAWIDTH(val) ((val & DEBAYER_CFG_OP_DATAWIDTH_M1_MASK) << DEBAYER_CFG_OP_DATAWIDTH_M1_SHIFT)
- #define DEBAYER_OUTPLANE_ORDER(val) ((val & DEBAYER_CFG_OUTPLANE_ORDER_MASK) << DEBAYER_CFG_OUTPLANE_ORDER_SHIFT)
- #define DEBAYER_NUM_OUTPLANES(val) ((val & DEBAYER_CFG_NUM_OUTPLANES_MASK) << DEBAYER_CFG_NUM_OUTPLANES_SHIFT)
- #define DEBAYER_GRAD_MULTIPLIER(val) ((val & DEBAYER_CFG_GRAD_MULTIPLIER_MASK) << DEBAYER_CFG_GRAD_MULTIPLIER_SHIFT)
- #define DEBAYER_CFG(bayerOrder, lumaOnly, forceRBZero, inDataWidth, outDataWidth, outplaneOrder, numOutplanes, gradMultiplier)
- #define BAYER_ORDER_GRBG DEBAYER_BAYER_ORDER(0x0)
- #define BAYER_ORDER_RGGG DEBAYER_BAYER_ORDER(0x1)
- #define BAYER_ORDER_GBRG DEBAYER_BAYER_ORDER(0x2)
- #define BAYER_ORDER_BGGR DEBAYER_BAYER_ORDER(0x3)
- #define OUTPLANE_ORDER_RGB DEBAYER_OUTPLANE_ORDER(0x0)
- #define OUTPLANE_ORDER_BGR DEBAYER_OUTPLANE_ORDER(0x1)
- #define OUTPLANE_ORDER_RBG DEBAYER_OUTPLANE_ORDER(0x2)
- #define OUTPLANE_ORDER_BRG DEBAYER_OUTPLANE_ORDER(0x3)
- #define OUTPLANE_ORDER_GRB DEBAYER_OUTPLANE_ORDER(0x4)
- #define OUTPLANE_ORDER_GBR DEBAYER_OUTPLANE_ORDER(0x5)
- #define DEBAYER_THRESHOLD_ABS_THRESH1_MASK 0x1FFF
- #define DEBAYER_THRESHOLD_ABS_THRESH1_SHIFT 0x0
- #define DEBAYER_THRESHOLD_ABS_THRESH2_MASK 0xFF
- #define DEBAYER_THRESHOLD_ABS_THRESH2_SHIFT 0xD
- #define DEBAYER_THRESHOLD_RGB_EN_MASK 0x1
- #define DEBAYER_THRESHOLD_RGB_EN_SHIFT 0x19
- #define DEBAYER_THRESHOLD_LUMA_EN_MASK 0x1
- #define DEBAYER_THRESHOLD_LUMA_EN_SHIFT 0x1A
- #define DEBAYER_THRESHOLD_LUMA_WC_EN_MASK 0x1
- #define DEBAYER_THRESHOLD_LUMA_WC_EN_SHIFT 0x1B
- #define DEBAYER_THRESHOLD_PREVIEW_EN_MASK 0x1
- #define DEBAYER_THRESHOLD_PREVIEW_EN_SHIFT 0x1C
- #define DEBAYER_ABS_THRESH1(val) ((val & DEBAYER_THRESHOLD_ABS_THRESH1_MASK) << DEBAYER_THRESHOLD_ABS_THRESH1_SHIFT)
- #define DEBAYER_ABS_THRESH2(val) ((val & DEBAYER_THRESHOLD_ABS_THRESH2_MASK) << DEBAYER_THRESHOLD_ABS_THRESH2_SHIFT)
- #define DEBAYER_RGB_EN(val) ((val & DEBAYER_THRESHOLD_RGB_EN_MASK) << DEBAYER_THRESHOLD_RGB_EN_SHIFT)
- #define DEBAYER_LUMA_EN(val) ((val & DEBAYER_THRESHOLD_LUMA_EN_MASK) << DEBAYER_THRESHOLD_LUMA_EN_SHIFT)

- #define DEBAYER_LUMA_WC_EN(val) ((val & DEBAYER_THRESHOLD_LUMA_WC_EN_MASK) << DEBAYER_THRESHOLD_LUMA_WC_EN_SHIFT)
- #define DEBAYER_PREVIEW_EN(val) ((val & DEBAYER_THRESHOLD_PREVIEW_EN_MASK) << DEBAYER_THRESHOLD_PREVIEW_EN_SHIFT)
- #define DEBAYER_DEWORM_OFFSET_MASK 0xFFFF
- #define DEBAYER_DEWORM_OFFSET_SHIFT 0x10
- #define DEBAYER_DEWORM_SLOPE_MASK 0xFFFF
- #define DEBAYER_DEWORM_SLOPE_SHIFT 0x0
- #define DEBAYER_OFFSET(val) ((val & DEBAYER_DEWORM_OFFSET_MASK) << DEBAYER_DEWORM_OFFSET_SHIFT)
- #define DEBAYER_SLOPE(val) ((val & DEBAYER_DEWORM_SLOPE_MASK) << DEBAYER_DEWORM_SLOPE_SHIFT)
- #define DEBAYER_THRESH(absThresh1, absThresh2, rgbEn, lumaEn, lumaWc, prewEn)
- #define DEBAYER_DEWORM(offset, slope)
- #define DEBAYER_LUMA_RED_COEF_MASK 0xFF
- #define DEBAYER_LUMA_RED_COEF_SHIFT 0x10
- #define DEBAYER_LUMA_BLUE_COEF_MASK 0xFF
- #define DEBAYER_LUMA_BLUE_COEF_SHIFT 0x00
- #define DEBAYER_LUMA_GREEN_COEF_MASK 0xFF
- #define DEBAYER_LUMA_GREEN_COEF_SHIFT 0x8
- #define DEBAYER_LUMA_RED_COEF(val) ((val & DEBAYER_LUMA_RED_COEF_MASK) << DEBAYER_LUMA_RED_COEF_SHIFT)
- #define DEBAYER_LUMA_BLUE_COEF(val) ((val & DEBAYER_LUMA_BLUE_COEF_MASK) << DEBAYER_LUMA_BLUE_COEF_SHIFT)
- #define DEBAYER_LUMA_GREEN_COEF(val) ((val & DEBAYER_LUMA_GREEN_COEF_MASK) << DEBAYER_LUMA_GREEN_COEF_SHIFT)
- #define DEBAYER_LUMA_COEFS_CFG(red, green, blue)
- #define CC_CFG_FORCE_LUMA_MASK 0x1
- #define CC_CFG_FORCE_LUMA_SHIFT 0x0
- #define CC_CFG_CHROMA_SUB_SAMP_MASK 0x3
- #define CC_CFG_CHROMA_SUB_SAMP_SHIFT 0x1
- #define CC_CFG_3DLUT_BYPASS_MASK 0x1
- #define CC_CFG_3DLUT_BYPASS_SHIFT 0x3
- #define CC_CFG_3DLUT_LOAD_ENABLE_MASK 0x1
- #define CC_CFG_3DLUT_LOAD_ENABLE_SHIFT 0x4
- #define CC_CFG_ENABLE_U12_OUT_MASK 0x1
- #define CC_CFG_ENABLE_U12_OUT_SHIFT 0x5
- #define CC_CFG_NUM_PLANES_MINUS_1_MASK 0x3
- #define CC_CFG_NUM_PLANES_MINUS_1_SHIFT 0x18
- #define CC_CFG_FORCE_LUMA(val) ((val & CC_CFG_FORCE_LUMA_MASK) << CC_CFG_FORCE_LUMA_SHIFT)
- #define CC_CFG_CHROMA_SUB_SAMP(val) ((val & CC_CFG_CHROMA_SUB_SAMP_MASK) << CC_CFG_CHROMA_SUB_SAMP_SHIFT)
- #define CC_CFG_3DLUT_BYPASS(val) ((val & CC_CFG_3DLUT_BYPASS_MASK) << CC_CFG_3DLUT_BYPASS_SHIFT)
- #define CC_CFG_3DLUT_LOAD_ENABLE(val) ((val & CC_CFG_3DLUT_LOAD_ENABLE_MASK) << CC_CFG_3DLUT_LOAD_ENABLE_SHIFT)

- #define CC_CFG_ENABLE_U12_OUT(val) ((val & CC_CFG_ENABLE_U12_OUT_MASK) << CC_CFG_ENABLE_U12_OUT_SHIFT)
- #define CC_CFG_NUM_PLANES_MINUS_1(val) ((val & CC_CFG_NUM_PLANES_MINUS_1_MASK) << CC_CFG_NUM_PLANES_MINUS_1_SHIFT)
- #define CC_CFG(forceLuma1, chrSubSamp, threeDLutBypass, threeDLutLoadEn, u12OutEn, numPlanes)
- #define HARRIS_CORNER_CFG_OUTPUT_DETERMINANT_MASK 0x1
- #define HARRIS_CORNER_CFG_OUTPUT_DETERMINANT_SHIFT 0x4
- #define HARRIS_CORNER_CFG_KERNEL_SIZE_MASK 0xF
- #define HARRIS_CORNER_CFG_KERNEL_SIZE_SHIFT 0x0
- #define HARRIS_CORNER_CFG_EXP_SUBTRAHEND_MASK 0xFF
- #define HARRIS_CORNER_CFG_EXP_SUBTRAHEND_SHIFT 0x8
- #define HARRIS_CORNER_CFG_OUTPUT_DETERMINANT(val) ((val & HARRIS_CORNER_CFG_OUTPUT_DETERMINANT_MASK) << HARRIS_CORNER_CFG_OUTPUT_DETERMINANT_SHIFT)
- #define HARRIS_CORNER_CFG_KERNEL_SIZE(val) ((val & HARRIS_CORNER_CFG_KERNEL_SIZE_MASK) << HARRIS_CORNER_CFG_KERNEL_SIZE_SHIFT)
- #define HARRIS_CORNER_CFG_EXP_SUBTRAHEND(val) ((val & HARRIS_CORNER_CFG_EXP_SUBTRAHEND_MASK) << HARRIS_CORNER_CFG_EXP_SUBTRAHEND_SHIFT)
- #define HARRIS_CORNER_CFG(kns, outDet, expSubtrahend)
- #define HARRIS_CORNER_KSZ_5 0x5
- #define HARRIS_CORNER_KSZ_7 0x7
- #define HARRIS_CORNER_KSZ_9 0x9
- #define SIPP_LSC_FRM_DIM_HEIGHT_MASK 0xFFFF
- #define SIPP_LSC_FRM_DIM_HEIGHT_SHIFT 0x10
- #define SIPP_LSC_FRM_DIM_WIDTH_MASK 0xFFFF
- #define SIPP_LSC_FRM_DIM_WIDTH_SHIFT 0x0
- #define SIPP_LSC_FRM_DIM_HEIGHT_CFG(val) ((val & SIPP_LSC_FRM_DIM_HEIGHT_MASK) << SIPP_LSC_FRM_DIM_HEIGHT_SHIFT)
- #define SIPP_LSC_FRM_DIM_WIDTH_CFG(val) ((val & SIPP_LSC_FRM_DIM_WIDTH_MASK) << SIPP_LSC_FRM_DIM_WIDTH_SHIFT)
- #define SIPP_LSC_FRM_DIM_CFG(height, width)
- #define SIPP_LSC_GM_DIM_HEIGHT_MASK 0x3FF
- #define SIPP_LSC_GM_DIM_HEIGHT_SHIFT 0x10
- #define SIPP_LSC_GM_DIM_WIDTH_MASK 0x3FF
- #define SIPP_LSC_GM_DIM_WIDTH_SHIFT 0x0
- #define SIPP_LSC_GM_DIM_HEIGHT_CFG(val) ((val & SIPP_LSC_GM_DIM_HEIGHT_MASK) << SIPP_LSC_GM_DIM_HEIGHT_SHIFT)
- #define SIPP_LSC_GM_DIM_WIDTH_CFG(val) ((val & SIPP_LSC_GM_DIM_WIDTH_MASK) << SIPP_LSC_GM_DIM_WIDTH_SHIFT)
- #define SIPP_LSC_GM_DIM_CFG(height, width)
- #define SIPP_LSC_CFG_FORMAT_MASK 0x1
- #define SIPP_LSC_CFG_FORMAT_SHIFT 0x0
- #define SIPP_LSC_CFG_FORMAT(val) ((val & SIPP_LSC_CFG_FORMAT_MASK) << SIPP_LSC_CFG_FORMAT_SHIFT)
- #define SIPP_LSC_CFG_DATA_WIDTH_MASK 0xF
- #define SIPP_LSC_CFG_DATA_WIDTH_SHIFT 0x4

- #define SIPP_LSC_CFG_DATA_WIDTH(val) ((val & SIPP_LSC_CFG_DATA_WIDTH_MASK) << SIPP_LSC_CFG_DATA_WIDTH_SHIFT)
- #define MED_CFG_KERNEL_SIZE_MASK 0x7
- #define MED_CFG_KERNEL_SIZE_SHIFT 0x0
- #define MED_CFG_OUTPUT_SELECT_MASK 0x3F
- #define MED_CFG_OUTPUT_SELECT_SHIFT 0x8
- #define MED_CFG_THRESHOLD_MASK 0x1FF
- #define MED_CFG_THRESHOLD_SHIFT 0x10
- #define MED_CFG_ALPHA_BLEND_EN_MASK 0x1
- #define MED_CFG_ALPHA_BLEND_EN_SHIFT 0x1C
- #define MED_CFG_LUM_SUBSAMP_EN_MASK 0x1
- #define MED_CFG_LUM_SUBSAMP_EN_SHIFT 0x1D
- #define MED_CFG_GO_FAST_MASK 0x1
- #define MED_CFG_GO_FAST_SHIFT 0x1E
- #define MED_CFG_KERNEL_SIZE(val) ((val & MED_CFG_KERNEL_SIZE_MASK) << MED_CFG_KERNEL_SIZE_SHIFT)
- #define MED_CFG_OUTPUT_SELECT(val) ((val & MED_CFG_OUTPUT_SELECT_MASK) << MED_CFG_OUTPUT_SELECT_SHIFT)
- #define MED_CFG_THRESHOLD(val) ((val & MED_CFG_THRESHOLD_MASK) << MED_CFG_THRESHOLD_SHIFT)
- #define MED_CFG_ALPHA_BLEND_EN(val) ((val & MED_CFG_ALPHA_BLEND_EN_MASK) << MED_CFG_ALPHA_BLEND_EN_SHIFT)
- #define MED_CFG_LUM_SUBSAMP_EN(val) ((val & MED_CFG_LUM_SUBSAMP_EN_MASK) << MED_CFG_LUM_SUBSAMP_EN_SHIFT)
- #define MED_CFG_GO_FAST(val) ((val & MED_CFG_GO_FAST_MASK) << MED_CFG_GO_FAST_SHIFT)
- #define MED_CFG(ks, outSelect, thresh, alphaBlend, lumaSubSamp, goFast)
- #define MED_LUMA_ALPHA_SLOPE_MASK 0xFF
- #define MED_LUMA_ALPHA_SLOPE_SHIFT 0x8
- #define MED_LUMA_ALPHA_OFFSET_MASK 0xFF
- #define MED_LUMA_ALPHA_OFFSET_SHIFT 0x0
- #define MED_LUMA_ALPHA_SLOPE(val) ((val & MED_LUMA_ALPHA_SLOPE_MASK) << MED_LUMA_ALPHA_SLOPE_SHIFT)
- #define MED_LUMA_ALPHA_OFFSET(val) ((val & MED_LUMA_ALPHA_OFFSET_MASK) << MED_LUMA_ALPHA_OFFSET_SHIFT)
- #define MED_LUMA_ALPHA(slope, offset)
- #define EDGE_OPERATOR_MAGNITUDE_SCALE_FACTOR_MASK 0xFFFF
- #define EDGE_OPERATOR_MAGNITUDE_SCALE_FACTOR_SHIFT 0x10
- #define EDGE_OPERATOR_THETA_OVX_MASK 0x1
- #define EDGE_OPERATOR_THETA_OVX_SHIFT 0x7
- #define EDGE_OPERATOR_THETA_MODE_MASK 0x3
- #define EDGE_OPERATOR_THETA_MODE_SHIFT 0x5
- #define EDGE_OPERATOR_OUTPUT_MODE_MASK 0x7
- #define EDGE_OPERATOR_OUTPUT_MODE_SHIFT 0x2
- #define EDGE_OPERATOR_INPUT_MODE_MASK 0x2
- #define EDGE_OPERATOR_INPUT_MODE_SHIFT 0x0
- #define EDGE_OPERATOR_BUFFER_HEIGHT_MASK 0xFFFF
- #define EDGE_OPERATOR_BUFFER_HEIGHT_SHIFT 0x10

- #define EDGE_OPERATOR_BUFFER_WIDTH_MASK 0xFFFF
- #define EDGE_OPERATOR_BUFFER_WIDTH_SHIFT 0x0
- #define EDGE_OPERATOR_MAG_SCALE_FACTOR(val) ((val & EDGE_OPERATOR_MAGNITUDE_SCALE_FACTOR_MASK) << EDGE_OPERATOR_MAGNITUDE_SCALE_FACTOR_SHIFT)
- #define EDGE_OPERATOR_THETA_OVX(val) ((val & EDGE_OPERATOR_THETA_OVX_MASK) << EDGE_OPERATOR_THETA_OVX_SHIFT)
- #define EDGE_OPERATOR_THETA_MODE(val) ((val & EDGE_OPERATOR_THETA_MODE_MASK) << EDGE_OPERATOR_THETA_MODE_SHIFT)
- #define EDGE_OPERATOR_OUTPUT_MODE(val) ((val & EDGE_OPERATOR_OUTPUT_MODE_MASK) << EDGE_OPERATOR_OUTPUT_MODE_SHIFT)
- #define EDGE_OPERATOR_INPUT_MODE(val) ((val & EDGE_OPERATOR_INPUT_MODE_MASK) << EDGE_OPERATOR_INPUT_MODE_SHIFT)
- #define SIPP_HW_EDGE_OPERATOR_BUFFER_HEIGHT(val) ((val & EDGE_OPERATOR_BUFFER_HEIGHT_MASK) << EDGE_OPERATOR_BUFFER_HEIGHT_SHIFT)
- #define SIPP_HW_EDGE_OPERATOR_BUFFER_WIDTH(val) ((val & EDGE_OPERATOR_BUFFER_WIDTH_MASK) << EDGE_OPERATOR_BUFFER_WIDTH_SHIFT)
- #define EDGE_OPERATOR_CFG(inMode, outMode, thetaMode, magScaleFactor, thetaScaleFactor)
- #define EDGE_OPERATOR_XCOEFF_F_MASK 0x1F
- #define EDGE_OPERATOR_XCOEFF_F_SHIFT 0x19
- #define EDGE_OPERATOR_XCOEFF_E_MASK 0x1F
- #define EDGE_OPERATOR_XCOEFF_E_SHIFT 0x14
- #define EDGE_OPERATOR_XCOEFF_D_MASK 0x1F
- #define EDGE_OPERATOR_XCOEFF_D_SHIFT 0xF
- #define EDGE_OPERATOR_XCOEFF_B_MASK 0x1F
- #define EDGE_OPERATOR_XCOEFF_B_SHIFT 0x5
- #define EDGE_OPERATOR_XCOEFF_A_MASK 0x1F
- #define EDGE_OPERATOR_XCOEFF_A_SHIFT 0x0
- #define EDGE_OPERATOR_XCOEFF_C_MASK 0x1F
- #define EDGE_OPERATOR_XCOEFF_C_SHIFT 0xA
- #define EDGE_OPERATOR_XCOEFF_A(val) ((val & EDGE_OPERATOR_XCOEFF_A_MASK) << EDGE_OPERATOR_XCOEFF_A_SHIFT)
- #define EDGE_OPERATOR_XCOEFF_B(val) ((val & EDGE_OPERATOR_XCOEFF_B_MASK) << EDGE_OPERATOR_XCOEFF_B_SHIFT)
- #define EDGE_OPERATOR_XCOEFF_C(val) ((val & EDGE_OPERATOR_XCOEFF_C_MASK) << EDGE_OPERATOR_XCOEFF_C_SHIFT)
- #define EDGE_OPERATOR_XCOEFF_D(val) ((val & EDGE_OPERATOR_XCOEFF_D_MASK) << EDGE_OPERATOR_XCOEFF_D_SHIFT)
- #define EDGE_OPERATOR_XCOEFF_E(val) ((val & EDGE_OPERATOR_XCOEFF_E_MASK) << EDGE_OPERATOR_XCOEFF_E_SHIFT)
- #define EDGE_OPERATOR_XCOEFF_F(val) ((val & EDGE_OPERATOR_XCOEFF_F_MASK) << EDGE_OPERATOR_XCOEFF_F_SHIFT)
- #define EDGE_OPERATOR_YCOEF_A_MASK 0x1F
- #define EDGE_OPERATOR_YCOEF_A_SHIFT 0x0
- #define EDGE_OPERATOR_YCOEF_B_MASK 0x1F
- #define EDGE_OPERATOR_YCOEF_B_SHIFT 0x5
- #define EDGE_OPERATOR_YCOEF_C_MASK 0x1F

- #define EDGE_OPERATOR_YCOEF_C_SHIFT 0xA
- #define EDGE_OPERATOR_YCOEF_D_MASK 0x1F
- #define EDGE_OPERATOR_YCOEF_D_SHIFT 0xF
- #define EDGE_OPERATOR_YCOEF_E_MASK 0x1F
- #define EDGE_OPERATOR_YCOEF_E_SHIFT 0x14
- #define EDGE_OPERATOR_YCOEF_F_MASK 0x1F
- #define EDGE_OPERATOR_YCOEF_F_SHIFT 0x19
- #define EDGE_OPERATOR_YCOEF_A(val) ((val & EDGE_OPERATOR_YCOEF_A_MASK) << EDGE_OPERATOR_YCOEF_A_SHIFT)
- #define EDGE_OPERATOR_YCOEF_B(val) ((val & EDGE_OPERATOR_YCOEF_B_MASK) << EDGE_OPERATOR_YCOEF_B_SHIFT)
- #define EDGE_OPERATOR_YCOEF_C(val) ((val & EDGE_OPERATOR_YCOEF_C_MASK) << EDGE_OPERATOR_YCOEF_C_SHIFT)
- #define EDGE_OPERATOR_YCOEF_D(val) ((val & EDGE_OPERATOR_YCOEF_D_MASK) << EDGE_OPERATOR_YCOEF_D_SHIFT)
- #define EDGE_OPERATOR_YCOEF_E(val) ((val & EDGE_OPERATOR_YCOEF_E_MASK) << EDGE_OPERATOR_YCOEF_E_SHIFT)
- #define EDGE_OPERATOR_YCOEF_F(val) ((val & EDGE_OPERATOR_YCOEF_F_MASK) << EDGE_OPERATOR_YCOEF_F_SHIFT)
- #define EDGE_OPERATOR_XCOEFS(a, b, c, d, e, f)
- #define EDGE_OPERATOR_YCOEFS(a, b, c, d, e, f)
- #define CHROMA_DNS_HORZ_ENABLE_MASK 0x7
- #define CHROMA_DNS_HORZ_ENABLE_SHIFT 0x0
- #define CHROMA_DNS_LIMIT_MASK 0xFF
- #define CHROMA_DNS_LIMIT_SHIFT 0x4
- #define CHROMA_DNS_FORCE_WEIGHTS_HORIZONTAL_MASK 0x1
- #define CHROMA_DNS_FORCE_WEIGHTS_HORIZONTAL_SHIFT 0xC
- #define CHROMA_DNS_FORCE_WEIGHTS_VERTICAL_MASK 0x1
- #define CHROMA_DNS_FORCE_WEIGHTS_VERTICAL_SHIFT 0xD
- #define CHROMA_DNS_PLANES_MODE_MASK 0x3
- #define CHROMA_DNS_PLANES_MODE_SHIFT 0xE
- #define CHROMA_DNS_DESAT_OFFSET_MASK 0xFF
- #define CHROMA_DNS_DESAT_OFFSET_SHIFT 0x10
- #define CHROMA_DNS_DESAT_SLOPE_MASK 0xFF
- #define CHROMA_DNS_DESAT_SLOPE_SHIFT 0x18
- #define CHROMA_DNS_HORZ_ENABLE(val) ((val & CHROMA_DNS_HORZ_ENABLE_MASK) << CHROMA_DNS_HORZ_ENABLE_SHIFT)
- #define CHROMA_DNS_LIMIT(val) ((val & CHROMA_DNS_LIMIT_MASK) << CHROMA_DNS_LIMIT_SHIFT)
- #define CHROMA_DNS_FORCE_WEIGHTS_HORIZONTAL(val) ((val & CHROMA_DNS_FORCE_WEIGHTS_HORIZONTAL_MASK) << CHROMA_DNS_FORCE_WEIGHTS_HORIZONTAL_SHIFT)
- #define CHROMA_DNS_FORCE_WEIGHTS_VERTICAL(val) ((val & CHROMA_DNS_FORCE_WEIGHTS_VERTICAL_MASK) << CHROMA_DNS_FORCE_WEIGHTS_VERTICAL_SHIFT)
- #define CHROMA_DNS_PLANES_MODE(val) ((val & CHROMA_DNS_PLANES_MODE_MASK) << CHROMA_DNS_PLANES_MODE_SHIFT)

- #define CHROMA_DNS_DESAT_OFFSET(val) ((val & CHROMA_DNS_DESAT_OFFSET_MASK) << CHROMA_DNS_DESAT_OFFSET_SHIFT)
- #define CHROMA_DNS_DESAT_SLOPE(val) ((val & CHROMA_DNS_DESAT_SLOPE_MASK) << CHROMA_DNS_DESAT_SLOPE_SHIFT)
- #define CHROMA_DNS_CFG(horzEnable, limit, forceWgthsHor, forceWgthsVer, planesMode, desatOffset, desatSlope)
- #define CHROMA_DNS_COEFFS_CENTRE_MASK 0xFF
- #define CHROMA_DNS_COEFFS_CENTRE_SHIFT 0x0
- #define CHROMA_DNS_COEFFS_CENTRE_EDGE_MASK 0xFF
- #define CHROMA_DNS_COEFFS_CENTRE_EDGE_SHIFT 0x8
- #define CHROMA_DNS_COEFFS_CORNER_MASK 0xFF
- #define CHROMA_DNS_COEFFS_CORNER_SHIFT 0x10
- #define CHROMA_DNS_COEFFS_CENTRE_SET(val) ((val & CHROMA_DNS_COEFFS_CENTRE_MASK) << CHROMA_DNS_COEFFS_CENTRE_SHIFT)
- #define CHROMA_DNS_COEFFS_CENTRE_EDGE_SET(val) ((val & CHROMA_DNS_COEFFS_CENTRE_EDGE_MASK) << CHROMA_DNS_COEFFS_CENTRE_EDGE_SHIFT)
- #define CHROMA_DNS_COEFFS_CORNER_SET(val) ((val & CHROMA_DNS_COEFFS_CORNER_MASK) << CHROMA_DNS_COEFFS_CORNER_SHIFT)
- #define CHROMA_DNS_COEFFS_SET(centre, centreEdge, corner)
- #define CHORMA_GREY_PT_CB_MASK 0xFF
- #define CHORMA_GREY_PT_CB_SHIFT 0x0
- #define CHORMA_GREY_PT_CG_MASK 0xFF
- #define CHORMA_GREY_PT_CG_SHIFT 0x8
- #define CHORMA_GREY_PT_CR_MASK 0xFF
- #define CHORMA_GREY_PT_CR_SHIFT 0x10
- #define CHORMA_GREY_PT_PASSTHRU_MASK 0xFF
- #define CHORMA_GREY_PT_PASSTHRU_SHIFT 0x1F
- #define CHORMA_GREY_PT_CB_SET(val) ((val & CHORMA_GREY_PT_CB_MASK) << CHORMA_GREY_PT_CB_SHIFT)
- #define CHORMA_GREY_PT_CG_SET(val) ((val & CHORMA_GREY_PT_CG_MASK) << CHORMA_GREY_PT_CG_SHIFT)
- #define CHORMA_GREY_PT_CR_SET(val) ((val & CHORMA_GREY_PT_CR_MASK) << CHORMA_GREY_PT_CR_SHIFT)
- #define CHORMA_GREY_PT_PASSTHRU_SET(val) ((val & CHORMA_GREY_PT_PASSTHRU_MASK) << CHORMA_GREY_PT_PASSTHRU_SHIFT)
- #define CHORMA_GREY_PT_SET(cb, cg, cr, passthru)
- #define CHROMA_DNS_HOR_T1_MASK 0xFF
- #define CHROMA_DNS_HOR_T1_SHIFT 0x0
- #define CHROMA_DNS_HOR_T2_MASK 0xFF
- #define CHROMA_DNS_HOR_T2_SHIFT 0x8
- #define CHROMA_DNS_VER_T1_MASK 0xFF
- #define CHROMA_DNS_VER_T1_SHIFT 0x10
- #define CHROMA_DNS_VER_T2_MASK 0xFF
- #define CHROMA_DNS_VER_T2_SHIFT 0x18
- #define CHROMA_DNS_HOR_T3_MASK 0xFF
- #define CHROMA_DNS_HOR_T3_SHIFT 0x0
- #define CHROMA_DNS_VER_T3_MASK 0xFF
- #define CHROMA_DNS_VER_T3_SHIFT 0x10

- #define CHROMA_DNS_HOR_T1(val) ((val & CHROMA_DNS_HOR_T1_MASK) << CHROMA_DNS_HOR_T1_SHIFT)
- #define CHROMA_DNS_HOR_T2(val) ((val & CHROMA_DNS_HOR_T2_MASK) << CHROMA_DNS_HOR_T2_SHIFT)
- #define CHROMA_DNS_VER_T1(val) ((val & CHROMA_DNS_VER_T1_MASK) << CHROMA_DNS_VER_T1_SHIFT)
- #define CHROMA_DNS_VER_T2(val) ((val & CHROMA_DNS_VER_T2_MASK) << CHROMA_DNS_VER_T2_SHIFT)
- #define CHROMA_DNS_HOR_T3(val) ((val & CHROMA_DNS_HOR_T3_MASK) << CHROMA_DNS_HOR_T3_SHIFT)
- #define CHROMA_DNS_VER_T3(val) ((val & CHROMA_DNS_VER_T3_MASK) << CHROMA_DNS_VER_T3_SHIFT)
- #define CHROMA_DNS_CFG_THRESH12(horzThr1, horzThr2, vertThr1, vertThr2)
- #define CHROMA_DNS_CFG_THRESH3(horzThr3, vertThr3)
- #define LUMA_DNS_CFG_C4_LUT_LOAD_EN_MASK 0x1
- #define LUMA_DNS_CFG_C4_LUT_LOAD_EN_SHIFT 0x18
- #define LUMA_DNS_CFG_BITSHIFT_MASK 0x1F
- #define LUMA_DNS_CFG_BITSHIFT_SHIFT 0x10
- #define LUMA_DNS_CFG_ALPHA_MASK 0xFF
- #define LUMA_DNS_CFG_ALPHA_SHIFT 0x8
- #define LUMA_DNS_CFG_BITPOS_MASK 0xF
- #define LUMA_DNS_CFG_BITPOS_SHIFT 0x0
- #define LUMA_DNS_CFG_C4_LUT_LOAD_EN(val) ((val & LUMA_DNS_CFG_C4_LUT_LOAD_EN_MASK) << LUMA_DNS_CFG_C4_LUT_LOAD_EN_SHIFT)
- #define LUMA_DNS_CFG_BITSHIFT(val) ((val & LUMA_DNS_CFG_BITSHIFT_MASK) << LUMA_DNS_CFG_BITSHIFT_SHIFT)
- #define LUMA_DNS_CFG_ALPHA(val) ((val & LUMA_DNS_CFG_ALPHA_MASK) << LUMA_DNS_CFG_ALPHA_SHIFT)
- #define LUMA_DNS_CFG_BITPOS(val) ((val & LUMA_DNS_CFG_BITPOS_MASK) << LUMA_DNS_CFG_BITPOS_SHIFT)
- #define LUMA_DNS_CFG(loadEn, bitShift, cfgAlpha, cfgBitpos)
- #define LUT_CFG_COLOR_CONVERSION_EN_MASK 0x1
- #define LUT_CFG_COLOR_CONVERSION_EN_SHIFT 0x10
- #define LUT_CFG_APB_ACCESS_EN_MASK 0x1
- #define LUT_CFG_APB_ACCESS_EN_SHIFT 0xF
- #define LUT_CFG_LUT_LOAD_EN_MASK 0x1
- #define LUT_CFG_LUT_LOAD_EN_SHIFT 0xE
- #define LUT_CFG_NUM_CHANNELS_MASK 0x3
- #define LUT_CFG_NUM_CHANNELS_SHIFT 0xC
- #define LUT_CFG_NUM_LUTS_MASK 0xF
- #define LUT_CFG_NUM_LUTS_SHIFT 0x8
- #define LUT_CFG_INTEGER_MODE_WIDTH_MASK 0x1F
- #define LUT_CFG_INTEGER_MODE_WIDTH_SHIFT 0x3
- #define LUT_CFG_CHANNEL_MODE_MASK 0x1
- #define LUT_CFG_CHANNEL_MODE_SHIFT 0x1
- #define LUT_CFG_INTERPOLATE_FP16_MASK 0x1
- #define LUT_CFG_INTERPOLATE_FP16_SHIFT 0x0

- #define LUT_CFG_COLOR_CONVERSION_EN(val) ((val & LUT_CFG_COLOR_CONVERSION_EN_MASK) << LUT_CFG_COLOR_CONVERSION_EN_SHIFT)
- #define LUT_CFG_APB_ACCESS_EN(val) ((val & LUT_CFG_APB_ACCESS_EN_MASK) << LUT_CFG_APB_ACCESS_EN_SHIFT)
- #define LUT_CFG_LUT_LOAD_EN(val) ((val & LUT_CFG_LUT_LOAD_EN_MASK) << LUT_CFG_LUT_LOAD_EN_SHIFT)
- #define LUT_CFG_NUM_CHANNELS(val) ((val & LUT_CFG_NUM_CHANNELS_MASK) << LUT_CFG_NUM_CHANNELS_SHIFT)
- #define LUT_CFG_NUM_LUTS(val) ((val & LUT_CFG_NUM_LUTS_MASK) << LUT_CFG_NUM_LUTS_SHIFT)
- #define LUT_CFG_INTEGER_MODE_WIDTH(val) ((val & LUT_CFG_INTEGER_MODE_WIDTH_MASK) << LUT_CFG_INTEGER_MODE_WIDTH_SHIFT)
- #define LUT_CFG_CHANNEL_MODE(val) ((val & LUT_CFG_CHANNEL_MODE_MASK) << LUT_CFG_CHANNEL_MODE_SHIFT)
- #define LUT_CFG_INTERPOLATE_FP16(val) ((val & LUT_CFG_INTERPOLATE_FP16_MASK) << LUT_CFG_INTERPOLATE_FP16_SHIFT)
- #define LUT_CFG(colorConv, apbAccesEN, loadEn, numChanel, numLuts, intModew, channelMode, interpoalateFp16)
- #define LUT_REGION0_SIZE_INDEX_MASK 0xF
- #define LUT_REGION0_SIZE_INDEX_SHIFT 0x0
- #define LUT_REGION1_SIZE_INDEX_MASK 0xF
- #define LUT_REGION1_SIZE_INDEX_SHIFT 0x4
- #define LUT_REGION2_SIZE_INDEX_MASK 0xF
- #define LUT_REGION2_SIZE_INDEX_SHIFT 0x8
- #define LUT_REGION3_SIZE_INDEX_MASK 0xF
- #define LUT_REGION3_SIZE_INDEX_SHIFT 0xC
- #define LUT_REGION4_SIZE_INDEX_MASK 0xF
- #define LUT_REGION4_SIZE_INDEX_SHIFT 0x10
- #define LUT_REGION5_SIZE_INDEX_MASK 0xF
- #define LUT_REGION5_SIZE_INDEX_SHIFT 0x14
- #define LUT_REGION6_SIZE_INDEX_MASK 0xF
- #define LUT_REGION6_SIZE_INDEX_SHIFT 0x18
- #define LUT_REGION7_SIZE_INDEX_MASK 0xF
- #define LUT_REGION7_SIZE_INDEX_SHIFT 0x1C
- #define LUT_REGION0_SIZE_INDEX(val) ((val & LUT_REGION0_SIZE_INDEX_MASK) << LUT_REGION0_SIZE_INDEX_SHIFT)
- #define LUT_REGION1_SIZE_INDEX(val) ((val & LUT_REGION1_SIZE_INDEX_MASK) << LUT_REGION1_SIZE_INDEX_SHIFT)
- #define LUT_REGION2_SIZE_INDEX(val) ((val & LUT_REGION2_SIZE_INDEX_MASK) << LUT_REGION2_SIZE_INDEX_SHIFT)
- #define LUT_REGION3_SIZE_INDEX(val) ((val & LUT_REGION3_SIZE_INDEX_MASK) << LUT_REGION3_SIZE_INDEX_SHIFT)
- #define LUT_REGION4_SIZE_INDEX(val) ((val & LUT_REGION4_SIZE_INDEX_MASK) << LUT_REGION4_SIZE_INDEX_SHIFT)
- #define LUT_REGION5_SIZE_INDEX(val) ((val & LUT_REGION5_SIZE_INDEX_MASK) << LUT_REGION5_SIZE_INDEX_SHIFT)
- #define LUT_REGION6_SIZE_INDEX(val) ((val & LUT_REGION6_SIZE_INDEX_MASK) << LUT_REGION6_SIZE_INDEX_SHIFT)

- #define LUT_REGION7_SIZE_INDEX(val) ((val & LUT_REGION7_SIZE_INDEX_MASK) << LUT_REGION7_SIZE_INDEX_SHIFT)
- #define LUT_REGION8_SIZE_INDEX_MASK 0xF
- #define LUT_REGION8_SIZE_INDEX_SHIFT 0x0
- #define LUT_REGION9_SIZE_INDEX_MASK 0xF
- #define LUT_REGION9_SIZE_INDEX_SHIFT 0x4
- #define LUT_REGION10_SIZE_INDEX_MASK 0xF
- #define LUT_REGION10_SIZE_INDEX_SHIFT 0x8
- #define LUT_REGION11_SIZE_INDEX_MASK 0xF
- #define LUT_REGION11_SIZE_INDEX_SHIFT 0xC
- #define LUT_REGION12_SIZE_INDEX_MASK 0xF
- #define LUT_REGION12_SIZE_INDEX_SHIFT 0x10
- #define LUT_REGION13_SIZE_INDEX_MASK 0xF
- #define LUT_REGION13_SIZE_INDEX_SHIFT 0x14
- #define LUT_REGION14_SIZE_INDEX_MASK 0xF
- #define LUT_REGION14_SIZE_INDEX_SHIFT 0x18
- #define LUT_REGION15_SIZE_INDEX_MASK 0xF
- #define LUT_REGION15_SIZE_INDEX_SHIFT 0x1C
- #define LUT_REGION8_SIZE_INDEX(val) ((val & LUT_REGION8_SIZE_INDEX_MASK) << LUT_REGION8_SIZE_INDEX_SHIFT)
- #define LUT_REGION9_SIZE_INDEX(val) ((val & LUT_REGION9_SIZE_INDEX_MASK) << LUT_REGION9_SIZE_INDEX_SHIFT)
- #define LUT_REGION10_SIZE_INDEX(val) ((val & LUT_REGION10_SIZE_INDEX_MASK) << LUT_REGION10_SIZE_INDEX_SHIFT)
- #define LUT_REGION11_SIZE_INDEX(val) ((val & LUT_REGION11_SIZE_INDEX_MASK) << LUT_REGION11_SIZE_INDEX_SHIFT)
- #define LUT_REGION12_SIZE_INDEX(val) ((val & LUT_REGION12_SIZE_INDEX_MASK) << LUT_REGION12_SIZE_INDEX_SHIFT)
- #define LUT_REGION13_SIZE_INDEX(val) ((val & LUT_REGION13_SIZE_INDEX_MASK) << LUT_REGION13_SIZE_INDEX_SHIFT)
- #define LUT_REGION14_SIZE_INDEX(val) ((val & LUT_REGION14_SIZE_INDEX_MASK) << LUT_REGION14_SIZE_INDEX_SHIFT)
- #define LUT_REGION15_SIZE_INDEX(val) ((val & LUT_REGION15_SIZE_INDEX_MASK) << LUT_REGION15_SIZE_INDEX_SHIFT)
- #define LUT_REGION_0_TO_7_SIZE(szR0, szR1, szR2, szR3, szR4, szR5, szR6, szR7)
- #define LUT_REGION_8_TO_15_SIZE(szR8, szR9, szR10, szR11, szR12, szR13, szR14, szR15)
- #define RAW_CFG_SDC_EN_MASK 0x1
- #define RAW_CFG_SDC_EN_SHIFT 0x1B
- #define RAW_CFG_RGB_HIST_EN_MASK 0x1
- #define RAW_CFG_RGB_HIST_EN_SHIFT 0x18
- #define RAW_CFG_BADP_THRES_MASK 0xFF
- #define RAW_CFG_BADP_THRES_SHIFT 0x10
- #define RAW_CFG_AF_STATS_EN_MASK 0x1
- #define RAW_CFG_AF_STATS_EN_SHIFT 0xD
- #define RAW_CFG_GAIN_MODE_MASK 0x1
- #define RAW_CFG_GAIN_MODE_SHIFT 0xC
- #define RAW_CFG_DATA_WIDTH_MASK 0xF

- #define RAW_CFG_DATA_WIDTH_SHIFT 0x8
- #define RAW_CFG_LUMA_HIST_EN_MASK 0x1
- #define RAW_CFG_LUMA_HIST_EN_SHIFT 0x7
- #define RAW_CFG_AWB_STATS_EN_MASK 0x1
- #define RAW_CFG_AWB_STATS_EN_SHIFT 0x6
- #define RAW_CFG_GREEN_FIX_ONLY_MASK 0x1
- #define RAW_CFG_GREEN_FIX_ONLY_SHIFT 0x5
- #define RAW_CFG_BAD_PIXEL_FIX_EN_MASK 0x1
- #define RAW_CFG_BAD_PIXEL_FIX_EN_SHIFT 0x4
- #define RAW_CFG_GRGB_IMB_EN_MASK 0x1
- #define RAW_CFG_GRGB_IMB_EN_SHIFT 0x3
- #define RAW_CFG_BAYER_PATTERN_MASK 0x3
- #define RAW_CFG_BAYER_PATTERN_SHIFT 0x1
- #define RAW_CFG_FORMAT_MASK 0x1
- #define RAW_CFG_FORMAT_SHIFT 0x0
- #define RAW_CFG_SDC_EN(val) ((val & RAW_CFG_SDC_EN_MASK) << RAW_CFG_SDC_EN_SHIFT)
- #define RAW_CFG_RGB_HIST_EN(val) ((val & RAW_CFG_RGB_HIST_EN_MASK) << RAW_CFG_RGB_HIST_EN_SHIFT)
- #define RAW_CFG_BADP_THRES(val) ((val & RAW_CFG_BADP_THRES_MASK) << RAW_CFG_BADP_THRES_SHIFT)
- #define RAW_CFG_AF_STATS_EN(val) ((val & RAW_CFG_AF_STATS_EN_MASK) << RAW_CFG_AF_STATS_EN_SHIFT)
- #define RAW_CFG_GAIN_MODE(val) ((val & RAW_CFG_GAIN_MODE_MASK) << RAW_CFG_GAIN_MODE_SHIFT)
- #define RAW_CFG_DATA_WIDTH(val) ((val & RAW_CFG_DATA_WIDTH_MASK) << RAW_CFG_DATA_WIDTH_SHIFT)
- #define RAW_CFG_LUMA_HIST_EN(val) ((val & RAW_CFG_LUMA_HIST_EN_MASK) << RAW_CFG_LUMA_HIST_EN_SHIFT)
- #define RAW_CFG_AWB_STATS_EN(val) ((val & RAW_CFG_AWB_STATS_EN_MASK) << RAW_CFG_AWB_STATS_EN_SHIFT)
- #define RAW_CFG_GREEN_FIX_ONLY(val) ((val & RAW_CFG_GREEN_FIX_ONLY_MASK) << RAW_CFG_GREEN_FIX_ONLY_SHIFT)
- #define RAW_CFG_BAD_PIXEL_FIX_EN(val) ((val & RAW_CFG_BAD_PIXEL_FIX_EN_MASK) << RAW_CFG_BAD_PIXEL_FIX_EN_SHIFT)
- #define RAW_CFG_GRGB_IMB_EN(val) ((val & RAW_CFG_GRGB_IMB_EN_MASK) << RAW_CFG_GRGB_IMB_EN_SHIFT)
- #define RAW_CFG_BAYER_PATTERN(val) ((val & RAW_CFG_BAYER_PATTERN_MASK) << RAW_CFG_BAYER_PATTERN_SHIFT)
- #define RAW_CFG_FORMAT(val) ((val & RAW_CFG_FORMAT_MASK) << RAW_CFG_FORMAT_SHIFT)
- #define RAW_CFG(format, bayerPatern, grgbImbEn, greenFixOnly, awbStatsEn, rgbHistEn, dataWidth, badPThresh, sdcEn, afStatsEn, gainMode, lumaHist, badPixelFixEn)
- #define RAW_BAD_PIXEL_CFG_NOISE_LEVEL_MASK 0xFFFF
- #define RAW_BAD_PIXEL_CFG_NOISE_LEVEL_SHIFT 0x10
- #define RAW_BAD_PIXEL_CFG_ALPHA_G_HOT_MASK 0xF
- #define RAW_BAD_PIXEL_CFG_ALPHA_G_HOT_SHIFT 0xC
- #define RAW_BAD_PIXEL_CFG_APLHA_G_COLD_MASK 0xF

- #define RAW_BAD_PIXEL_CFG_APLHA_G_COLD_SHIFT 0x8
- #define RAW_BAD_PIXEL_CFG_ALPHA_RB_HOT_MASK 0xF
- #define RAW_BAD_PIXEL_CFG_ALPHA_RB_HOT_SHIFT 0x4
- #define RAW_BAD_PIXEL_CFG_ALPHA_RB_COLD_MASK 0xF
- #define RAW_BAD_PIXEL_CFG_ALPHA_RB_COLD_SHIFT 0x0
- #define RAW_BAD_PIXEL_CFG_NOISE_LEVEL(val) ((val & RAW_BAD_PIXEL_CFG_NOISE_LEVEL_MASK) << RAW_BAD_PIXEL_CFG_NOISE_LEVEL_SHIFT)
- #define RAW_BAD_PIXEL_CFG_ALPHA_G_HOT(val) ((val & RAW_BAD_PIXEL_CFG_ALPHA_G_HOT_MASK) << RAW_BAD_PIXEL_CFG_ALPHA_G_HOT_SHIFT)
- #define RAW_BAD_PIXEL_CFG_APLHA_G_COLD(val) ((val & RAW_BAD_PIXEL_CFG_APLHA_G_COLD_MASK) << RAW_BAD_PIXEL_CFG_APLHA_G_COLD_SHIFT)
- #define RAW_BAD_PIXEL_CFG_ALPHA_RB_HOT(val) ((val & RAW_BAD_PIXEL_CFG_ALPHA_RB_HOT_MASK) << RAW_BAD_PIXEL_CFG_ALPHA_RB_HOT_SHIFT)
- #define RAW_BAD_PIXEL_CFG_ALPHA_RB_COLD(val) ((val & RAW_BAD_PIXEL_CFG_ALPHA_RB_COLD_MASK) << RAW_BAD_PIXEL_CFG_ALPHA_RB_COLD_SHIFT)
- #define RAW_BAD_PIX_CFG(noiseLevel, alphaGHot, alphaGCold, alphaRbHot, alphaRbCold)
- #define RAW_GRGB_PLATO_DARK_MASK 0x3FFF
- #define RAW_GRGB_PLATO_DARK_SHIFT 0x0
- #define RAW_GRGB_PLATO_BRIGHT_MASK 0x3FFF
- #define RAW_GRGB_PLATO_BRIGHT_SHIFT 0x10
- #define RAW_GRGB_PLATO_DARK(val) ((val & RAW_GRGB_PLATO_DARK_MASK) << RAW_GRGB_PLATO_DARK_SHIFT)
- #define RAW_GRGB_PLATO_BRIGHT(val) ((val & RAW_GRGB_PLATO_BRIGHT_MASK) << RAW_GRGB_PLATO_BRIGHT_SHIFT)
- #define RAW_GRGB_PLATO(dark, bright)
- #define RAW_GRGB_SLOPE_BRIGHT_MASK 0x3FFF
- #define RAW_GRGB_SLOPE_BRIGHT_SHIFT 0x10
- #define RAW_GRGB_SLOPE_DARK_MASK 0x3FFF
- #define RAW_GRGB_SLOPE_DARK_SHIFT 0x0
- #define RAW_GRGB_SLOPE_BRIGHT(val) ((val & RAW_GRGB_SLOPE_BRIGHT_MASK) << RAW_GRGB_SLOPE_BRIGHT_SHIFT)
- #define RAW_GRGB_SLOPE_DARK(val) ((val & RAW_GRGB_SLOPE_DARK_MASK) << RAW_GRGB_SLOPE_DARK_SHIFT)
- #define RAW_GRGB_DECAY(slopeBright, slopeDark)
- #define RAW_PATCH_CFG_NUM_HORIZ_PATCHES_MASK 0x3F
- #define RAW_PATCH_CFG_NUM_HORIZ_PATCHES_SHIFT 0x0
- #define RAW_PATCH_CFG_NUM_VERT_PATCHES_MASK 0x3F
- #define RAW_PATCH_CFG_NUM_VERT_PATCHES_SHIFT 0x8
- #define RAW_PATCH_CFG_PATCH_WIDTH_MASK 0xFF
- #define RAW_PATCH_CFG_PATCH_WIDTH_SHIFT 0x10
- #define RAW_PATCH_CFG_PATCH_HEIGHT_MASK 0xFF
- #define RAW_PATCH_CFG_PATCH_HEIGHT_SHIFT 0x18
- #define RAW_PATCH_CFG_NUM_HORIZ_PATCHES(val) ((val & RAW_PATCH_CFG_NUM_HORIZ_PATCHES_MASK) << RAW_PATCH_CFG_NUM_HORIZ_PATCHES_SHIFT)
- #define RAW_PATCH_CFG_NUM_VERT_PATCHES(val) ((val & RAW_PATCH_CFG_NUM_VERT_PATCHES_MASK) << RAW_PATCH_CFG_NUM_VERT_PATCHES_SHIFT)
- #define RAW_PATCH_CFG_PATCH_WIDTH(val) ((val & RAW_PATCH_CFG_PATCH_WIDTH_MASK) << RAW_PATCH_CFG_PATCH_WIDTH_SHIFT)

- #define RAW_PATCH_CFG_PATCH_HEIGHT(val) ((val & RAW_PATCH_CFG_PATCH_HEIGHT_MASK) << RAW_PATCH_CFG_PATCH_HEIGHT_SHIFT)
- #define RAW_PATCH_CFG(numHorizPatches, numVertPatches, patchWidth, patchHeight)
- #define RAW_PATCH_START_X_POS_MASK 0xFFFF
- #define RAW_PATCH_START_X_POS_SHIFT 0x0
- #define RAW_PATCH_START_Y_POS_MASK 0xFFFF
- #define RAW_PATCH_START_Y_POS_SHIFT 0x10
- #define RAW_PATCH_START_X_POS(val) ((val & RAW_PATCH_START_X_POS_MASK) << RAW_PATCH_START_X_POS_SHIFT)
- #define RAW_PATCH_START_Y_POS(val) ((val & RAW_PATCH_START_Y_POS_MASK) << RAW_PATCH_START_Y_POS_SHIFT)
- #define RAW_PATCH_START(xCoord, yCoord)
- #define RAW_STATS_PLANES_PLANE0_MASK 0xF
- #define RAW_STATS_PLANES_PLANE0_SHIFT 0x0
- #define RAW_STATS_PLANES_PLANE1_MASK 0xF
- #define RAW_STATS_PLANES_PLANE1_SHIFT 0x4
- #define RAW_STATS_PLANES_PLANE2_MASK 0xF
- #define RAW_STATS_PLANES_PLANE2_SHIFT 0x8
- #define RAW_STATS_PLANES_PLANE3_MASK 0xF
- #define RAW_STATS_PLANES_PLANE3_SHIFT 0xC
- #define RAW_STATS_PLANES_HGRAM_PLANE_MASK 0xF
- #define RAW_STATS_PLANES_HGRAM_PLANE_SHIFT 0x10
- #define RAW_STATS_PLANES_ACTIVE_PLANES_MASK 0x3
- #define RAW_STATS_PLANES_ACTIVE_PLANES_SHIFT 0x14
- #define RAW_STATS_PLANES_PLANE0(val) ((val & RAW_STATS_PLANES_PLANE0_MASK) << RAW_STATS_PLANES_PLANE0_SHIFT)
- #define RAW_STATS_PLANES_PLANE1(val) ((val & RAW_STATS_PLANES_PLANE1_MASK) << RAW_STATS_PLANES_PLANE1_SHIFT)
- #define RAW_STATS_PLANES_PLANE2(val) ((val & RAW_STATS_PLANES_PLANE2_MASK) << RAW_STATS_PLANES_PLANE2_SHIFT)
- #define RAW_STATS_PLANES_PLANE3(val) ((val & RAW_STATS_PLANES_PLANE3_MASK) << RAW_STATS_PLANES_PLANE3_SHIFT)
- #define RAW_STATS_PLANES_HGRAM_PLANE(val) ((val & RAW_STATS_PLANES_HGRAM_PLANE_MASK) << RAW_STATS_PLANES_HGRAM_PLANE_SHIFT)
- #define RAW_STATS_PLANES_ACTIVE_PLANES(val) (((val ? val-1 : val) & RAW_STATS_PLANES_ACTIVE_PLANES_MASK) << RAW_STATS_PLANES_ACTIVE_PLANES_SHIFT)
- #define RAW_STATS_PLANES(plane0, plane1, plane2, plane3, histPlane, activePlanes)
- #define POLY_FRM_HEIGHT_IN_MASK 0xFFFF
- #define POLY_FRM_HEIGHT_IN_SHIFT 0x10
- #define POLY_FRM_WIDTH_IN_MASK 0xFFFF
- #define POLY_FRM_WIDTH_IN_SHIFT 0x0
- #define POLY_FRM_HEIGHT_IN(val) ((val & POLY_FRM_HEIGHT_IN_MASK) << POLY_FRM_HEIGHT_IN_SHIFT)
- #define POLY_FRM_WIDTH_IN(val) ((val & POLY_FRM_WIDTH_IN_MASK) << POLY_FRM_WIDTH_IN_SHIFT)
- #define POLY_CFG_DIM_IN
- #define POLY_FRM_HEIGHT_OUT_MASK 0xFFFF

- #define POLY_FRM_HEIGHT_OUT_SHIFT 0x10
- #define POLY_FRM_WIDTH_OUT_MASK 0xFFFF
- #define POLY_FRM_WIDTH_OUT_SHIFT 0x0
- #define POLY_FRM_HEIGHT_OUT(val) ((val & POLY_FRM_HEIGHT_OUT_MASK) << POLY_FRM_HEIGHT_OUT_SHIFT)
- #define POLY_FRM_WIDTH_OUT(val) ((val & POLY_FRM_WIDTH_OUT_MASK) << POLY_FRM_WIDTH_OUT_SHIFT)
- #define POLY_CFG_DIM_OUT
- #define POLY_CFG_UPFIRDN_VSCALE_FACTOR_NUM_MASK 0x1F
- #define POLY_CFG_UPFIRDN_VSCALE_FACTOR_NUM_SHIFT 0x16
- #define POLY_CFG_UPFIRDN_VSCALE_FACTOR_DENOM_MASK 0x3F
- #define POLY_CFG_UPFIRDN_VSCALE_FACTOR_DENOM_SHIFT 0x10
- #define POLY_CFG_UPFIRDN_HSCALE_FACTOR_NUM_MASK 0x7FFF
- #define POLY_CFG_UPFIRDN_HSCALE_FACTOR_NUM_SHIFT 0xA
- #define POLY_CFG_UPFIRDN_HSCALE_FACTOR_DENOM_MASK 0x3F
- #define POLY_CFG_UPFIRDN_HSCALE_FACTOR_DENOM_SHIFT 0x4
- #define POLY_CFG_UPFIRDN_OUTPUT_CLAMP_MASK 0x1
- #define POLY_CFG_UPFIRDN_OUTPUT_CLAMP_SHIFT 0x3
- #define POLY_CFG_KERNEL_SIZE_MASK 0x7
- #define POLY_CFG_KERNEL_SIZE_SHIFT 0x0
- #define SIPP_HW_POLY_FIR_CLAMP_DISABLE 0x0
- #define SIPP_HW_POLY_FIR_CLAMP_ENABLE 0x1
- #define SIPP_UPFIRDN_ID 15
- #define POLY_CFG_UPFIRDN_VSCALE_FACTOR_NUM(val) ((val & POLY_CFG_UPFIRDN_VSCALE_FACTOR_NUM_MASK) << POLY_CFG_UPFIRDN_VSCALE_FACTOR_NUM_SHIFT)
- #define POLY_CFG_UPFIRDN_VSCALE_FACTOR_DENOM(val) ((val & POLY_CFG_UPFIRDN_VSCALE_FACTOR_DENOM_MASK) << POLY_CFG_UPFIRDN_VSCALE_FACTOR_DENOM_SHIFT)
- #define POLY_CFG_UPFIRDN_HSCALE_FACTOR_NUM(val) ((val & POLY_CFG_UPFIRDN_HSCALE_FACTOR_NUM_MASK) << POLY_CFG_UPFIRDN_HSCALE_FACTOR_NUM_SHIFT)
- #define POLY_CFG_UPFIRDN_HSCALE_FACTOR_DENOM(val) ((val & POLY_CFG_UPFIRDN_HSCALE_FACTOR_DENOM_MASK) << POLY_CFG_UPFIRDN_HSCALE_FACTOR_DENOM_SHIFT)
- #define POLY_CFG_UPFIRDN_OUTPUT_CLAMP(val) ((val & POLY_CFG_UPFIRDN_OUTPUT_CLAMP_MASK) << POLY_CFG_UPFIRDN_OUTPUT_CLAMP_SHIFT)
- #define POLY_CFG_KERNEL_SIZE(val) ((val & POLY_CFG_KERNEL_SIZE_MASK) << POLY_CFG_KERNEL_SIZE_SHIFT)
- #define POLY_CFG
- #define POLY_VERTICAL_COEFF0_MASK 0xFF
- #define POLY_VERTICAL_COEFF0_SHIFT 0x0
- #define POLY_VERTICAL_COEFF1_MASK 0xFF
- #define POLY_VERTICAL_COEFF1_SHIFT 0x8
- #define POLY_VERTICAL_COEFF2_MASK 0xFF
- #define POLY_VERTICAL_COEFF2_SHIFT 0x10
- #define POLY_VERTICAL_COEFF3_MASK 0xFF
- #define POLY_VERTICAL_COEFF3_SHIFT 0x18

- #define POLY_VERTICAL_COEFF4_MASK 0xFF
- #define POLY_VERTICAL_COEFF4_SHIFT 0x0
- #define POLY_VERTICAL_COEFF5_MASK 0xFF
- #define POLY_VERTICAL_COEFF5_SHIFT 0x10
- #define POLY_VERTICAL_COEFF6_MASK 0xFF
- #define POLY_VERTICAL_COEFF6_SHIFT 0x10
- #define POLY_VERTICAL_COEFF0 (val & POLY_VERTICAL_COEFF0_MASK) << POLY_VERTICAL_COEFF0_SHIFT)
- #define POLY_VERTICAL_COEFF1 (val & POLY_VERTICAL_COEFF1_MASK) << POLY_VERTICAL_COEFF1_SHIFT)
- #define POLY_VERTICAL_COEFF2 (val & POLY_VERTICAL_COEFF2_MASK) << POLY_VERTICAL_COEFF2_SHIFT)
- #define POLY_VERTICAL_COEFF3 (val & POLY_VERTICAL_COEFF3_MASK) << POLY_VERTICAL_COEFF3_SHIFT)
- #define POLY_CFG_VERTICAL_COEFFS_0_TO_3
- #define POLY_CFG_VERTICAL_COEFFS_4_TO_6
- #define POLY_HORIZONTAL_COEFF0_MASK 0xFF
- #define POLY_HORIZONTAL_COEFF0_SHIFT 0x0
- #define POLY_HORIZONTAL_COEFF1_MASK 0xFF
- #define POLY_HORIZONTAL_COEFF1_SHIFT 0x8
- #define POLY_HORIZONTAL_COEFF2_MASK 0xFF
- #define POLY_HORIZONTAL_COEFF2_SHIFT 0x10
- #define POLY_HORIZONTAL_COEFF3_MASK 0xFF
- #define POLY_HORIZONTAL_COEFF3_SHIFT 0x18
- #define POLY_HORIZONTAL_COEFF4_MASK 0xFF
- #define POLY_HORIZONTAL_COEFF4_SHIFT 0x0
- #define POLY_HORIZONTAL_COEFF5_MASK 0xFF
- #define POLY_HORIZONTAL_COEFF5_SHIFT 0x10
- #define POLY_HORIZONTAL_COEFF6_MASK 0xFF
- #define POLY_HORIZONTAL_COEFF6_SHIFT 0x10
- #define POLY_HORIZONTAL_COEFF0 (val & POLY_HORIZONTAL_COEFF0_MASK) << POLY_HORIZONTAL_COEFF0_SHIFT)
- #define POLY_HORIZONTAL_COEFF1 (val & POLY_HORIZONTAL_COEFF1_MASK) << POLY_HORIZONTAL_COEFF1_SHIFT)
- #define POLY_HORIZONTAL_COEFF2 (val & POLY_HORIZONTAL_COEFF2_MASK) << POLY_HORIZONTAL_COEFF2_SHIFT)
- #define POLY_HORIZONTAL_COEFF3 (val & POLY_HORIZONTAL_COEFF3_MASK) << POLY_HORIZONTAL_COEFF3_SHIFT)
- #define POLY_CFG_HORIZONTAL_COEFFS_0_TO_3
- #define POLY_CFG_HORIZONTAL_COEFFS_4_TO_6
- #define MIPI_RX_OUT_FRM_HEIGHT_MASK 0xFFFF
- #define MIPI_RX_OUT_FRM_HEIGHT_SHIFT 0x10
- #define MIPI_RX_OUT_FRM_WIDTH_MASK 0xFFFF
- #define MIPI_RX_OUT_FRM_WIDTH_SHIFT 0x0
- #define MIPI_RX_OUT_FRM_HEIGHT(val) ((MIPI_RX_OUT_FRM_HEIGHT_MASK) << MIPI_RX_OUT_FRM_HEIGHT_SHIFT)
- #define MIPI_RX_OUT_FRM_WIDTH(val) ((MIPI_RX_OUT_FRM_WIDTH_MASK) << MIPI_RX_OUT_FRM_WIDTH_SHIFT)

- #define MIPI_RX_CFG_OUT_FRM_DIM
- #define MIPI_RX_CFG_INPUT_BIT_DEPTH_MASK 0xF
- #define MIPI_RX_CFG_INPUT_BIT_DEPTH_SHIFT 0x1C
- #define MIPI_RX_CFG_PROMOTE_MASK 0x1
- #define MIPI_RX_CFG_PROMOTE_SHIFT 0x1A
- #define MIPI_RX_CFG_USE_PRIVATE_LCS_MASK 0x1
- #define MIPI_RX_CFG_USE_PRIVATE_SHIFT 0x19
- #define MIPI_RX_CFG_PACK_BUFFER_MASK 0x1
- #define MIPI_RX_CFG_PACK_BUFFER_SHIFT 0x13
- #define MIPI_RX_CFG_BAYER_MODE_MASK 0x1
- #define MIPI_RX_CFG_BAYER_MODE_SHIFT 0x11
- #define MIPI_RX_FMT_CONV_RS_MASK 0x1F
- #define MIPI_RX_FMT_CONV_RS_SHIFT 0xC
- #define MIPI_RX_FMT_CONV_EN_MASK 0xF
- #define MIPI_RX_FMT_CONV_EN_SHIFT 0x8
- #define MIPI_RX_CFG_INPUT_BIT_DEPTH(val) ((val & MIPI_RX_CFG_INPUT_BIT_DEPTH_MASK) << MIPI_RX_CFG_INPUT_BIT_DEPTH_SHIFT)
- #define MIPI_RX_CFG_PROMOTE(val) ((val & MIPI_RX_CFG_PROMOTE_MASK) << MIPI_RX_CFG_PROMOTE_SHIFT)
- #define MIPI_RX_CFG_USE_PRIVATE_LCS(val) ((val & MIPI_RX_CFG_USE_PRIVATE_LCS_MASK) << MIPI_RX_CFG_USE_PRIVATE_LCS_SHIFT)
- #define MIPI_RX_CFG_PACK_BUFFER(val) ((val & MIPI_RX_CFG_PACK_BUFFER_MASK) << MIPI_RX_CFG_PACK_BUFFER_SHIFT)
- #define MIPI_RX_CFG_BAYER_MODE(val) ((val & MIPI_RX_CFG_BAYER_MODE_MASK) << MIPI_RX_CFG_BAYER_MODE_SHIFT)
- #define MIPI_RX_FMT_CONV_RS(val) ((val & MIPI_RX_FMT_CONV_RS_MASK) << MIPI_RX_FMT_CONV_RS_SHIFT)
- #define MIPI_RX_FMT_CONV_EN(val) ((val & MIPI_RX_FMT_CONV_EN_MASK) << MIPI_RX_FMT_CONV_EN_SHIFT)
- #define MIPI_RX_CFG(inBitDepth, promote, usePrivateLcs, packBuffer, bayerMode, convRs, convEn)
- #define MIPI_RX_WINDOW0_HORIZONTAL_WIDTH_CFG_MASK 0xFFFF
- #define MIPI_RX_WINDOW0_HORIZONTAL_WIDTH_CFG_SHIFT 0x10
- #define MIPI_RX_WINDOW0_VERTICAL_HEIGHT_CFG_MASK 0xFFFF
- #define MIPI_RX_WINDOW0_VERTICAL_HEIGHT_CFG_SHIFT 0x10
- #define MIPI_RX_WINDOW0_HORIZONTAL_START_CFG_MASK 0xFFFF
- #define MIPI_RX_WINDOW0_HORIZONTAL_START_CFG_SHIFT 0x0
- #define MIPI_RX_WINDOW0_VERTICAL_START_CFG_MASK 0xFFFF
- #define MIPI_RX_WINDOW0_VERTICAL_START_CFG_SHIFT 0x0
- #define MIPI_RX_WINDOW0_HORIZONTAL_WIDTH_CFG(val) ((val & MIPI_RX_WINDOW0_HORIZONTAL_WIDTH_CFG_MASK) << MIPI_RX_WINDOW0_HORIZONTAL_WIDTH_CFG_SHIFT)
- #define MIPI_RX_WINDOW0_VERTICAL_HEIGHT_CFG(val) ((val & MIPI_RX_WINDOW0_VERTICAL_HEIGHT_CFG_MASK) << MIPI_RX_WINDOW0_VERTICAL_HEIGHT_CFG_SHIFT)
- #define MIPI_RX_WINDOW0_HORIZONTAL_START_CFG(val) ((val & MIPI_RX_WINDOW0_HORIZONTAL_START_CFG_MASK) << MIPI_RX_WINDOW0_HORIZONTAL_START_CFG_SHIFT)

- #define MIPI_RX_WINDOW0_VERTICAL_START_CFG(val) ((val & MIPI_RX_WINDOW0_VERTICAL_START_CFG_MASK) << MIPI_RX_WINDOW0_VERTICAL_START_CFG_SHIFT)
- #define MIPI_RX_WINDOW1_HORIZONTAL_WIDTH_CFG_MASK 0xFFFF
- #define MIPI_RX_WINDOW1_HORIZONTAL_WIDTH_CFG_SHIFT 0x10
- #define MIPI_RX_WINDOW1_VERTICAL_HEIGHT_CFG_MASK 0xFFFF
- #define MIPI_RX_WINDOW1_VERTICAL_HEIGHT_CFG_SHIFT 0x10
- #define MIPI_RX_WINDOW1_HORIZONTAL_START_CFG_MASK 0xFFFF
- #define MIPI_RX_WINDOW1_HORIZONTAL_START_CFG_SHIFT 0x0
- #define MIPI_RX_WINDOW1_VERTICAL_START_CFG_MASK 0xFFFF
- #define MIPI_RX_WINDOW1_VERTICAL_START_CFG_SHIFT 0x0
- #define MIPI_RX_WINDOW1_HORIZONTAL_WIDTH_CFG(val) ((val & MIPI_RX_WINDOW1_HORIZONTAL_WIDTH_CFG_MASK) << MIPI_RX_WINDOW1_HORIZONTAL_WIDTH_CFG_SHIFT)
- #define MIPI_RX_WINDOW1_VERTICAL_WIDTH_CFG(val) ((val & MIPI_RX_WINDOW1_VERTICAL_WIDTH_CFG_MASK) << MIPI_RX_WINDOW1_VERTICAL_WIDTH_CFG_SHIFT)
- #define MIPI_RX_WINDOW1_HORIZONTAL_START_CFG(val) ((val & MIPI_RX_WINDOW1_HORIZONTAL_START_CFG_MASK) << MIPI_RX_WINDOW1_HORIZONTAL_START_CFG_SHIFT)
- #define MIPI_RX_WINDOW1_VERTICAL_START_CFG(val) ((val & MIPI_RX_WINDOW1_VERTICAL_START_CFG_MASK) << MIPI_RX_WINDOW1_VERTICAL_START_CFG_SHIFT)
- #define MIPI_RX_WINDOW2_HORIZONTAL_WIDTH_CFG_MASK 0xFFFF
- #define MIPI_RX_WINDOW2_HORIZONTAL_WIDTH_CFG_SHIFT 0x10
- #define MIPI_RX_WINDOW2_VERTICAL_HEIGHT_CFG_MASK 0xFFFF
- #define MIPI_RX_WINDOW2_VERTICAL_HEIGHT_CFG_SHIFT 0x10
- #define MIPI_RX_WINDOW2_HORIZONTAL_START_CFG_MASK 0xFFFF
- #define MIPI_RX_WINDOW2_HORIZONTAL_START_CFG_SHIFT 0x0
- #define MIPI_RX_WINDOW2_VERTICAL_START_CFG_MASK 0xFFFF
- #define MIPI_RX_WINDOW2_VERTICAL_START_CFG_SHIFT 0x0
- #define MIPI_RX_WINDOW2_HORIZONTAL_WIDTH_CFG(val) ((val & MIPI_RX_WINDOW2_HORIZONTAL_WIDTH_CFG_MASK) << MIPI_RX_WINDOW2_HORIZONTAL_WIDTH_CFG_SHIFT)
- #define MIPI_RX_WINDOW2_VERTICAL_WIDTH_CFG(val) ((val & MIPI_RX_WINDOW2_VERTICAL_WIDTH_CFG_MASK) << MIPI_RX_WINDOW2_VERTICAL_WIDTH_CFG_SHIFT)
- #define MIPI_RX_WINDOW2_HORIZONTAL_START_CFG(val) ((val & MIPI_RX_WINDOW2_HORIZONTAL_START_CFG_MASK) << MIPI_RX_WINDOW2_HORIZONTAL_START_CFG_SHIFT)
- #define MIPI_RX_WINDOW2_VERTICAL_START_CFG(val) ((val & MIPI_RX_WINDOW2_VERTICAL_START_CFG_MASK) << MIPI_RX_WINDOW2_VERTICAL_START_CFG_SHIFT)
- #define MIPI_RX_WINDOW3_HORIZONTAL_WIDTH_CFG_MASK 0xFFFF
- #define MIPI_RX_WINDOW3_HORIZONTAL_WIDTH_CFG_SHIFT 0x10
- #define MIPI_RX_WINDOW3_VERTICAL_HEIGHT_CFG_MASK 0xFFFF
- #define MIPI_RX_WINDOW3_VERTICAL_HEIGHT_CFG_SHIFT 0x10
- #define MIPI_RX_WINDOW3_HORIZONTAL_START_CFG_MASK 0xFFFF

- #define MIPI_RX_WINDOW3_HORIZONTAL_START_CFG_SHIFT 0x0
- #define MIPI_RX_WINDOW3_VERTICAL_START_CFG_MASK 0xFFFF
- #define MIPI_RX_WINDOW3_VERTICAL_START_CFG_SHIFT 0x0
- #define MIPI_RX_WINDOW3_HORIZONTAL_WIDTH_CFG(val) ((val & MIPI_RX_WINDOW3_HORIZONTAL_WIDTH_CFG_MASK) << MIPI_RX_WINDOW3_HORIZONTAL_WIDTH_CFG_SHIFT)
- #define MIPI_RX_WINDOW3_VERTICAL_WIDTH_CFG(val) ((val & MIPI_RX_WINDOW3_VERTICAL_WIDTH_CFG_MASK) << MIPI_RX_WINDOW3_VERTICAL_WIDTH_CFG_SHIFT)
- #define MIPI_RX_WINDOW3_HORIZONTAL_START_CFG(val) ((val & MIPI_RX_WINDOW3_HORIZONTAL_START_CFG_MASK) << MIPI_RX_WINDOW3_HORIZONTAL_START_CFG_SHIFT)
- #define MIPI_RX_WINDOW3_VERTICAL_START_CFG(val) ((val & MIPI_RX_WINDOW3_VERTICAL_START_CFG_MASK) << MIPI_RX_WINDOW3_VERTICAL_START_CFG_SHIFT)
- #define MIPI_RX_WINDOW0_SELECTION_ENABLE_MASK 0xF
- #define MIPI_RX_WINDOW0_SELECTION_ENABLE_SHIFT 0x8
- #define MIPI_RX_WINDOW0_LEAST_SIGNIFICANT_BIT_MASK 0x1F
- #define MIPI_RX_WINDOW0_LEAST_SIGNIFICANT_BIT_SHIFT 0x0
- #define MIPI_RX_WINDOW1_SELECTION_ENABLE_MASK 0xF
- #define MIPI_RX_WINDOW1_SELECTION_ENABLE_SHIFT 0x18
- #define MIPI_RX_WINDOW1_LEAST_SIGNIFICANT_BIT_MASK 0x1F
- #define MIPI_RX_WINDOW1_LEAST_SIGNIFICANT_BIT_SHIFT 0xF
- #define MIPI_RX_WINDOW2_SELECTION_ENABLE_MASK 0xF
- #define MIPI_RX_WINDOW2_SELECTION_ENABLE_SHIFT 0x8
- #define MIPI_RX_WINDOW2_LEAST_SIGNIFICANT_BIT_MASK 0x1F
- #define MIPI_RX_WINDOW2_LEAST_SIGNIFICANT_BIT_SHIFT 0x0
- #define MIPI_RX_WINDOW3_SELECTION_ENABLE_MASK 0xF
- #define MIPI_RX_WINDOW3_SELECTION_ENABLE_SHIFT 0x18
- #define MIPI_RX_WINDOW3_LEAST_SIGNIFICANT_BIT_MASK 0x1F
- #define MIPI_RX_WINDOW3_LEAST_SIGNIFICANT_BIT_SHIFT 0xF
- #define MIPI_TX_FRM_HEIGHT_MASK 0xFFFF
- #define MIPI_TX_FRM_HEIGHT_SHIFT 0x10
- #define MIPI_TX_FRM_WIDTH_MASK 0xFFFF
- #define MIPI_TX_FRM_WIDTH_SHIFT 0x0
- #define MIPI_TX_FRM_HEIGHT(val) ((val & MIPI_TX_FRM_HEIGHT_MASK) << MIPI_TX_FRM_HEIGHT_SHIFT)
- #define MIPI_TX_FRM_WIDTH(val) ((val & MIPI_TX_FRM_WIDTH_MASK) << MIPI_TX_FRM_WIDTH_SHIFT)
- #define MIPI_TX_FRM_CFG
- #define MIPI_TX_CFG_FRONTPORCH_MASK 0x1
- #define MIPI_TX_CFG_FRONTPORCH_SHIFT 0x9
- #define MIPI_TX_CFG_BACKPORCH_MASK 0x1
- #define MIPI_TX_CFG_BACKPORCH_SHIFT 0x8
- #define MIPI_TX_INT_SEL_MASK 0x3
- #define MIPI_TX_INT_SEL_SHIFT 0x6
- #define MIPI_TX_SYNC_IDLE_MASK 0x1
- #define MIPI_TX_SYNC_IDLE_SHIFT 0x3

- #define MIPI_TX_ONESHOT_MODE_MASK 0x1
- #define MIPI_TX_ONESHOT_MODE_SHIFT 0x2
- #define MIPI_TX_FIRST_FIELD_MASK 0x1
- #define MIPI_TX_FIRST_FIELD_SHIFT 0x1
- #define MIPI_TX_SCAN_MODE_MASK 0x1
- #define MIPI_TX_SCAN_MODE_SHIFT 0x0
- #define MIPI_TX_CFG_FRONTPORCH(val) ((val & MIPI_TX_CFG_FRONTPORCH_MASK) << MIPI_TX_CFG_FRONTPORCH_SHIFT)
- #define MIPI_TX_CFG_BACKPORCH(val) ((val & MIPI_TX_CFG_BACKPORCH_MASK) << MIPI_TX_CFG_BACKPORCH+SHIFT)
- #define MIPI_TX_INT_SEL(val) ((val & MIPI_TX_INT_SEL_MASK) << MIPI_TX_INT_SEL_SHIFT)
- #define MIPI_TX_SYNC_IDLE(val) ((val & MIPI_TX_SYNC_IDLE_MASK) << MIPI_TX_SYNC_IDLE_SHIFT)
- #define MIPI_TX_ONESHOT(val) ((val & MIPI_TX_ONESHOT_MASK) << MIPI_TX_ONESHOT_SHIFT)
- #define MIPI_TX_FIRST_FIELD(val) ((val & MIPI_TX_FIRST_FIELD_MASK) << MIPI_TX_FIRST_FIELD_SHIFT)
- #define MIPI_TX_SCAN_MODE(val) ((val & MIPI_TX_SCAN_MODE_MASK) << MIPI_TX_SCAN_MODE_SHIFT)
- #define SIGMA_DNS_FRM_HEIGHT_MASK 0xFFFF
- #define SIGMA_DNS_FRM_HEIGHT_SHIFT 0x10
- #define SIGMA_DNS_FRM_HEIGHT(val) ((val & SIGMA_DNS_FRM_HEIGHT_MASK) << SIGMA_DNS_FRM_HEIGHT_SHIFT)
- #define SIGMA_DNS_FRM_WIDTH(val) ((val & SIGMA_DNS_FRM_WIDTH_MASK) << SIGMA_DNS_FRM_WIDTH_SHIFT)
- #define SIGMA_DNS_CFG_FORMAT_MASK 0x1
- #define SIGMA_DNS_CFG_FORMAT_SHIFT 0x0
- #define SIGMA_DNS_CFG_PASSTHRU_EN_MASK 0x1
- #define SIGMA_DNS_CFG_PASSTHRU_EN_SHIFT 0x1
- #define SIGMA_DNS_CFG_DATA_WIDTH_MASK 0xF
- #define SIGMA_DNS_CFG_DATA_WIDTH_SHIFT 0x4
- #define SIGMA_DNS_CFG_NF_MASK 0x3FFF
- #define SIGMA_DNS_CFG_NF_SHIFT 0x8
- #define SIGMA_DNS_CFG_NF(val) ((val & SIGMA_DNS_CFG_NF_MASK) << SIGMA_DNS_CFG_NF_SHIFT)
- #define SIGMA_DNS_CFG_DATA_WIDTH(val) ((val & SIGMA_DNS_CFG_DATA_WIDTH_MASK) << SIGMA_DNS_CFG_DATA_WIDTH_SHIFT)
- #define SIGMA_DNS_CFG_PASSTHRU_EN(val) ((val & SIGMA_DNS_CFG_PASSTHRU_EN_MASK) << SIGMA_DNS_CFG_PASSTHRU_EN_SHIFT)
- #define SIGMA_DNS_CFG_FORMAT(val) ((val & SIGMA_DNS_CFG_FORMAT_MASK) << SIGMA_DNS_CFG_FORMAT_SHIFT)
- #define SIGMA_DNS_CFG(nf, dataWidth, passthr, format)
- #define SIGMA_DNS_TRESH_T2P1_MASK 0xFF
- #define SIGMA_DNS_TRESH_T2P1_SHIFT 0x18
- #define SIGMA_DNS_TRESH_T1P1_MASK 0xFF
- #define SIGMA_DNS_TRESH_T1P1_SHIFT 0x10
- #define SIGMA_DNS_TRESH_T2P0_MASK 0xFF

- #define SIGMA_DNS_TRESH_T2P0_SHIFT 0x8
- #define SIGMA_DNS_TRESH_T1P0_MASK 0xFF
- #define SIGMA_DNS_TRESH_T1P0_SHIFT 0x0
- #define SIGMA_DNS_TRESH_T2P3_MASK 0xFF
- #define SIGMA_DNS_TRESH_T2P3_SHIFT 0x18
- #define SIGMA_DNS_TRESH_T1P3_MASK 0xFF
- #define SIGMA_DNS_TRESH_T1P3_SHIFT 0x10
- #define SIGMA_DNS_TRESH_T2P2_MASK 0xFF
- #define SIGMA_DNS_TRESH_T2P2_SHIFT 0x8
- #define SIGMA_DNS_TRESH_T1P2_MASK 0xFF
- #define SIGMA_DNS_TRESH_T1P2_SHIFT 0x0
- #define SIGMA_DNS_TRESH_T2P1(val) ((val & SIGMA_DNS_TRESH_T2P1_MASK) << SIGMA_DNS_TRESH_T2P1_SHIFT)
- #define SIGMA_DNS_TRESH_T1P1(val) ((val & SIGMA_DNS_TRESH_T1P1_MASK) << SIGMA_DNS_TRESH_T1P1_SHIFT)
- #define SIGMA_DNS_TRESH_T2P0(val) ((val & SIGMA_DNS_TRESH_T2P0_MASK) << SIGMA_DNS_TRESH_T2P0_SHIFT)
- #define SIGMA_DNS_TRESH_T2P3(val) ((val & SIGMA_DNS_TRESH_T2P3_MASK) << SIGMA_DNS_TRESH_T2P3_SHIFT)
- #define SIGMA_DNS_TRESH_T1P3(val) ((val & SIGMA_DNS_TRESH_T1P3_MASK) << SIGMA_DNS_TRESH_T1P3_SHIFT)
- #define SIGMA_DNS_TRESH_T2P2(val) ((val & SIGMA_DNS_TRESH_T2P2_MASK) << SIGMA_DNS_TRESH_T2P2_SHIFT)
- #define SIGMA_DNS_TRESH_T1P2(val) ((val & SIGMA_DNS_TRESH_T1P2_MASK) << SIGMA_DNS_TRESH_T1P2_SHIFT)
- #define SIGMA_DNS_TRESH_T1P0(val) ((val & SIGMA_DNS_TRESH_T1P0_MASK) << SIGMA_DNS_TRESH_T1P0_SHIFT)
- #define SIGMA_DNS_TRESH_CFG(t1, t2, t3, t4, t5, t6, t7, t8)
- #define DOGL_CFG_MODE_MASK 0x3
- #define DOGL_CFG_MODE_SHIFT 0x0
- #define DOGL_CFG_DOG_THRESHOLD_MASK 0xFF
- #define DOGL_CFG_DOG_THRESHOLD_SHIFT 0x2
- #define DOGL_CFG_OUTPUT_CLAMP_MASK 0x1
- #define DOGL_CFG_OUTPUT_CLAMP_SHIFT 0xA
- #define DOGL_CFG_LLB_DSMODE_MASK 0x3
- #define DOGL_CFG_LLB_DSMODE_SHIFT 0xC
- #define DOGL_CFG_BACK_GEN_THRESH_MASK 0xFF
- #define DOGL_CFG_BACK_GEN_THRESH_SHIFT 0xE
- #define DOGL_CFG_NUM_PLANES_MASK 0xF
- #define DOGL_CFG_NUM_PLANES_SHIFT 0x16
- #define DOGL_CFG_KERNEL_HEIGHT_MASK 0xF
- #define DOGL_CFG_KERNEL_HEIGHT_SHIFT 0x1A
- #define DOGL_CFG_KERNEL_HEIGHT(val) ((val & DOGL_CFG_KERNEL_HEIGHT_MASK) << DOGL_CFG_KERNEL_HEIGHT_SHIFT)
- #define DOGL_CFG_NUM_PLANES(val) ((val & DOGL_CFG_NUM_PLANES_MASK) << DOGL_CFG_NUM_PLANES_SHIFT)
- #define DOGL_CFG_BACK_GEN_THRESH(val) ((val & DOGL_CFG_BACK_GEN_THRESH_MASK) << DOGL_CFG_BACK_GEN_THRESH_SHIFT)

- #define DOGL_CFG_LLB_DSMODE(val) ((val & DOGL_CFG_LLB_DSMODE_MASK) << DOGL_CFG_LLB_DSMODE_SHIFT)
- #define DOGL_CFG_OUTPUT_CLAMP(val) ((val & DOGL_CFG_OUTPUT_CLAMP_MASK) << DOGL_CFG_OUTPUT_CLAMP_SHIFT)
- #define DOGL_CFG_DOG_THRESHOLD(val) ((val & DOGL_CFG_DOG_THRESHOLD_MASK) << DOGL_CFG_DOG_THRESHOLD_SHIFT)
- #define DOGL_CFG_MODE(val) ((val & DOGL_CFG_MODE_MASK) << DOGL_CFG_MODE_SHIFT)
- #define DOGL_MODE_DOG_ONLY 0x0
- #define DOGL_MODE_LTM_ONLY 0x1
- #define DOGL_MODE_DOG_DENOISE 0x2
- #define DOGL_MODE_DOG_LTM 0x3
- #define DOGL_CFG_SET(mode, thresh, outClamp, llbDSMode, backGenThresh, numPlanes, kernelHeight)
- #define SHARPEN_CFG_KERNEL_SIZE_MASK 0x7
- #define SHARPEN_CFG_KERNEL_SIZE_SHIFT 0x0
- #define SHARPEN_CFG_CLAMP_MASK 0x1
- #define SHARPEN_CFG_CLAMP_SHIFT 0x3
- #define SHARPEN_CFG_MODE_MASK 0x1
- #define SHARPEN_CFG_MODE_SHIFT 0x4
- #define SHARPEN_CFG_OPUT_DELTAS_ONLY_MASK 0x1
- #define SHARPEN_CFG_OPUT_DELTAS_ONLY_SHIFT 0x5
- #define SHARPEN_CFG_MIN_THRESHOLD_MASK 0xFFFF
- #define SHARPEN_CFG_MIN_THRESHOLD_SHIFT 0x10
- #define SHARPEN_CFG_KERNEL_SIZE(val) ((val & SHARPEN_CFG_KERNEL_SIZE_MASK) << SHARPEN_CFG_KERNEL_SIZE_SHIFT)
- #define SHARPEN_CFG_CLAMP(val) ((val & SHARPEN_CFG_CLAMP_MASK) << SHARPEN_CFG_CLAMP_SHIFT)
- #define SHARPEN_CFG_MODE(val) ((val & SHARPEN_CFG_MODE_MASK) << SHARPEN_CFG_MODE_SHIFT)
- #define SHARPEN_CFG_OPUT_DELTAS_ONLY(val) ((val & SHARPEN_CFG_OPUT_DELTAS_ONLY_MASK) << SHARPEN_CFG_OPUT_DELTAS_ONLY_SHIFT)
- #define SHARPEN_CFG_MIN_THRESHOLD(val) ((val & SHARPEN_CFG_MIN_THRESHOLD_MASK) << SHARPEN_CFG_MIN_THRESHOLD_SHIFT)
- #define SHARPEN_CFG(thresh, ksz, clamp, mode, opDeltas)
- #define SHARPEN_COEF0_MASK 0xFFFF
- #define SHARPEN_COEF0_SHIFT 0x0
- #define SHARPEN_COEF1_MASK 0xFFFF
- #define SHARPEN_COEF1_SHIFT 0x10
- #define SHARPEN_COEF2_MASK 0xFFFF
- #define SHARPEN_COEF2_SHIFT 0x0
- #define SHARPEN_COEF3_MASK 0xFFFF
- #define SHARPEN_COEF3_SHIFT 0x10
- #define SHARPEN_COEF0_CFG(val) ((val & SHARPEN_COEF0_MASK) << SHARPEN_COEF0_SHIFT)
- #define SHARPEN_COEF1_CFG(val) ((val & SHARPEN_COEF1_MASK) << SHARPEN_COEF1_SHIFT)

- #define SHARPEN_COEF2_CFG(val) ((val & SHARPEN_COEF2_MASK) << SHARPEN_COEF2_SHIFT)
- #define SHARPEN_COEF3_CFG(val) ((val & SHARPEN_COEF3_MASK) << SHARPEN_COEF3_SHIFT)
- #define SHARPEN_COEFS01_CFG(coef0, coef1)
- #define SHARPEN_COEFS23_CFG(coef2, coef3)
- #define SHARPEN_RANGESTOP0_MASK 0xFFFF
- #define SHARPEN_RANGESTOP0_SHIFT 0x0
- #define SHARPEN_RANGESTOP1_MASK 0xFFFF
- #define SHARPEN_RANGESTOP1_SHIFT 0x10
- #define SHARPEN_RANGESTOP2_MASK 0xFFFF
- #define SHARPEN_RANGESTOP2_SHIFT 0x0
- #define SHARPEN_RANGESTOP3_MASK 0xFFFF
- #define SHARPEN_RANGESTOP3_SHIFT 0x10
- #define SHARPEN_RANGESTOP0_CFG(val) ((val & SHARPEN_RANGESTOP0_MASK) << SHARPEN_RANGESTOP0_SHIFT)
- #define SHARPEN_RANGESTOP1_CFG(val) ((val & SHARPEN_RANGESTOP1_MASK) << SHARPEN_RANGESTOP1_SHIFT)
- #define SHARPEN_RANGESTOP2_CFG(val) ((val & SHARPEN_RANGESTOP2_MASK) << SHARPEN_RANGESTOP2_SHIFT)
- #define SHARPEN_RANGESTOP3_CFG(val) ((val & SHARPEN_RANGESTOP3_MASK) << SHARPEN_RANGESTOP3_SHIFT)
- #define SHARPEN_RANGESTOP01_CFG(rangeStop0, rangeStop1)
- #define SHARPEN_RANGESTOP23_CFG(rangeStop2, rangeStop3)
- #define SHARPEN_LIMITS_UNDERSHOOT_MASK 0xFFFF
- #define SHARPEN_LIMITS_UNDERSHOOT_SHIFT 0x0
- #define SHARPEN_LIMITS_OVERSHOOT_MASK 0xFFFF
- #define SHARPEN_LIMITS_OVERSHOOT_SHIFT 0x10
- #define SHARPEN_LIMITS_UNDERSHOOT_CFG(val) ((val & SHARPEN_LIMITS_UNDERSHOOT_MASK) << SHARPEN_LIMITS_UNDERSHOOT_SHIFT)
- #define SHARPEN_LIMITS_OVERSHOOT_CFG(val) ((val & SHARPEN_LIMITS_OVERSHOOT_MASK) << SHARPEN_LIMITS_OVERSHOOT_SHIFT)
- #define SHARPEN_LIMITS_CFG(underShoot, overShoot)
- #define SHARPEN_STRENGTH_POSITIVE_MASK 0xFFFF
- #define SHARPEN_STRENGTH_POSITIVE_SHIFT 0x10
- #define SHARPEN_STRENGTH_NEGATIVE_MASK 0xFFFF
- #define SHARPEN_STRENGTH_NEGATIVE_SHIFT 0x0
- #define SHARPEN_STRENGTH_POSITIVE_CFG(val) ((val & SHARPEN_STRENGTH_POSITIVE_MASK) << SHARPEN_STRENGTH_POSITIVE_SHIFT)
- #define SHARPEN_STRENGTH_NEGATIVE_CFG(val) ((val & SHARPEN_STRENGTH_NEGATIVE_MASK) << SHARPEN_STRENGTH_NEGATIVE_SHIFT)
- #define SHARPEN_STRENGTH_CFG(pos, neg)
- #define CGEN_CFG_PURP_FLARE_STR_MASK 0xFF
- #define CGEN_CFG_PURP_FLARE_STR_SHIFT 0x0
- #define CGEN_CFG_DESAT_OFFSET_MASK 0xFF
- #define CGEN_CFG_DESAT_OFFSET_SHIFT 0x8
- #define CGEN_CFG_DESAT_SLOPE_MASK 0xFF
- #define CGEN_CFG_DESAT_SLOPE_SHIFT 0x10

- #define CGEN_CFG_DATA_WIDTH_MASK 0xF
- #define CGEN_CFG_DATA_WIDTH_SHIFT 0x18
- #define CGEN_CFG_BYPASS_MASK 0x1
- #define CGEN_CFG_BYPASS_SHIFT 0x1C
- #define CGEN_CFG_LLB_DSMODE_MASK 0x3
- #define CGEN_CFG_LLB_DSMODE_SHIFT 0x1D
- #define CGEN_CFG_PURP_FLARE_STR(val) ((val & CGEN_CFG_PURP_FLARE_STR_MASK) << CGEN_CFG_PURP_FLARE_STR_SHIFT)
- #define CGEN_CFG_DESAT_OFFSET(val) ((val & CGEN_CFG_DESAT_OFFSET_MASK) << CGEN_CFG_DESAT_OFFSET_SHIFT)
- #define CGEN_CFG_DESAT_SLOPE(val) ((val & CGEN_CFG_DESAT_SLOPE_MASK) << CGEN_CFG_DESAT_SLOPE_SHIFT)
- #define CGEN_CFG_DATA_WIDTH(val) (((val-1) & CGEN_CFG_DATA_WIDTH_MASK) << CGEN_CFG_DATA_WIDTH_SHIFT)
- #define CGEN_CFG_BYPASS(val) ((val & CGEN_CFG_BYPASS_MASK) << CGEN_CFG_BYPASS_SHIFT)
- #define CGEN_CFG_LLB_DSMODE(val) ((val & CGEN_CFG_LLB_DSMODE_MASK) << CGEN_CFG_LLB_DSMODE_SHIFT)
- #define CGEN_CFG_SET(purpFlareStr, desatOffset, desatSlope, dataWidth, bypass, llbDSMode)
- #define CGEN_LUMA_COEFF_RED_MASK 0xFF
- #define CGEN_LUMA_COEFF_RED_SHIFT 0x0
- #define CGEN_LUMA_COEFF_GREEN_MASK 0xFF
- #define CGEN_LUMA_COEFF_GREEN_SHIFT 0x8
- #define CGEN_LUMA_COEFF_BLUE_MASK 0xFF
- #define CGEN_LUMA_COEFF_BLUE_SHIFT 0x10
- #define CGEN_LUMA_COEFF_RED(val) ((val & CGEN_LUMA_COEFF_RED_MASK) << CGEN_LUMA_COEFF_RED_SHIFT)
- #define CGEN_LUMA_COEFF_GREEN(val) ((val & CGEN_LUMA_COEFF_GREEN_MASK) << CGEN_LUMA_COEFF_GREEN_SHIFT)
- #define CGEN_LUMA_COEFF_BLUE(val) ((val & CGEN_LUMA_COEFF_BLUE_MASK) << CGEN_LUMA_COEFF_BLUE_SHIFT)
- #define CGEN_LUMA_COEFF_SET(red, green, blue)
- #define CGEN_CHROMA_COEFF_EPSILON_MASK 0xFF
- #define CGEN_CHROMA_COEFF_EPSILON_SHIFT 0x0
- #define CGEN_CHROMA_COEFF_KR_MASK 0xFF
- #define CGEN_CHROMA_COEFF_KR_SHIFT 0x8
- #define CGEN_CHROMA_COEFF_KG_MASK 0xFF
- #define CGEN_CHROMA_COEFF_KG_SHIFT 0x10
- #define CGEN_CHROMA_COEFF_KB_MASK 0xFF
- #define CGEN_CHROMA_COEFF_KB_SHIFT 0x18
- #define CGEN_CHROMA_COEFF_EPSILON(val) ((val & CGEN_CHROMA_COEFF_EPSILON_MASK) << CGEN_CHROMA_COEFF_EPSILON_SHIFT)
- #define CGEN_CHROMA_COEFF_RED(val) ((val & CGEN_CHROMA_COEFF_KR_MASK) << CGEN_CHROMA_COEFF_KR_SHIFT)
- #define CGEN_CHROMA_COEFF_GREEN(val) ((val & CGEN_CHROMA_COEFF_KG_MASK) << CGEN_CHROMA_COEFF_KG_SHIFT)
- #define CGEN_CHROMA_COEFF_BLUE(val) ((val & CGEN_CHROMA_COEFF_KB_MASK) << CGEN_CHROMA_COEFF_KB_SHIFT)
- #define CGEN_CHROMA_COEFF_SET(epsilon, kr, kg, kb)

Enumerations

- enum eRawInputFmt { eRawFormatPlanar = 0, eRawFormatBayer }
- enum eBayerOrder { eBayerOrderGRBG = 0, eBayerOrderRGGB, eBayerOrderGBRG, eBayerOrderBGGR }

7.226.1 Macro Definition Documentation

```
#define BAYER_ORDER_BGGR DEBAYER_BAYER_ORDER(0x3)
```

```
#define BAYER_ORDER_GBRG DEBAYER_BAYER_ORDER(0x2)
```

```
#define BAYER_ORDER_GRBG DEBAYER_BAYER_ORDER(0x0)
```

```
#define BAYER_ORDER_RGGB DEBAYER_BAYER_ORDER(0x1)
```

```
#define CC_CFG( forceLuma1, chrSubSamp, threeDLutBypass, threeDLutLoadEn, u12OutEn, numPlanes )
```

Value:

```
CC_CFG_FORCE_LUMA (forceLuma1)  |\
    CC_CFG_CHROMA_SUB_SAMP (chrSubSamp)  |\
    CC_CFG_3DLUT_BYPASS (threeDLutBypass)  |\
    CC_CFG_3DLUT_LOAD_ENABLE (threeDLutLoadEn)  |\
    CC_CFG_ENABLE_U12_OUT (u12OutEn)  |\
    CC_CFG_NUM_PLANES_MINUS_1 ((numPlanes-1))

#define CC_CFG_3DLUT_BYPASS( val ) ((val & CC_CFG_3DLUT_BYPASS_MASK) << CC_CFG_3DLUT_BYPASS_SHIFT)

#define CC_CFG_3DLUT_BYPASS_MASK 0x1

#define CC_CFG_3DLUT_BYPASS_SHIFT 0x3

#define CC_CFG_3DLUT_LOAD_ENABLE( val ) ((val & CC_CFG_3DLUT_LOAD_ENABLE_MASK) << CC_CFG_3DLUT_LOAD_ENABLE_SHIFT)

#define CC_CFG_3DLUT_LOAD_ENABLE_MASK 0x1

#define CC_CFG_3DLUT_LOAD_ENABLE_SHIFT 0x4

#define CC_CFG_CHROMA_SUB_SAMP( val ) ((val & CC_CFG_CHROMA_SUB_SAMP_MASK) << CC_CFG_CHROMA_SUB_SAMP_SHIFT)

#define CC_CFG_CHROMA_SUB_SAMP_MASK 0x3
```

```

#define CC_CFG_CHROMA_SUB_SAMP_SHIFT 0x1

#define CC_CFG_ENABLE_U12_OUT( val ) ((val & CC_CFG_ENABLE_U12_OUT_MASK)
<< CC_CFG_ENABLE_U12_OUT_SHIFT)

#define CC_CFG_ENABLE_U12_OUT_MASK 0x1

#define CC_CFG_ENABLE_U12_OUT_SHIFT 0x5

#define CC_CFG_FORCE_LUMA( val ) ((val & CC_CFG_FORCE_LUMA_MASK ) <<
CC_CFG_FORCE_LUMA_SHIFT )

#define CC_CFG_FORCE_LUMA_MASK 0x1

#define CC_CFG_FORCE_LUMA_SHIFT 0x0

#define CC_CFG_NUM_PLANES_MINUS_1( val ) ((val & CC_CFG_NUM_PLANES_MINUS_-
1_MASK) << CC_CFG_NUM_PLANES_MINUS_1_SHIFT)

#define CC_CFG_NUM_PLANES_MINUS_1_MASK 0x3

#define CC_CFG_NUM_PLANES_MINUS_1_SHIFT 0x18

#define CGEN_CFG_BYPASS( val ) (( val & CGEN_CFG_BYPASS_MASK ) <<
CGEN_CFG_BYPASS_SHIFT )

#define CGEN_CFG_BYPASS_MASK 0x1

#define CGEN_CFG_BYPASS_SHIFT 0x1C

#define CGEN_CFG_DATA_WIDTH( val ) (((val-1) & CGEN_CFG_DATA_WIDTH_MASK )
<< CGEN_CFG_DATA_WIDTH_SHIFT )

#define CGEN_CFG_DATA_WIDTH_MASK 0xF

#define CGEN_CFG_DATA_WIDTH_SHIFT 0x18

#define CGEN_CFG_DESAT_OFFSET( val ) (( val & CGEN_CFG_DESAT_OFFSET_MASK )
<< CGEN_CFG_DESAT_OFFSET_SHIFT )

#define CGEN_CFG_DESAT_OFFSET_MASK 0xFF

#define CGEN_CFG_DESAT_OFFSET_SHIFT 0x8

#define CGEN_CFG_DESAT_SLOPE( val ) (( val & CGEN_CFG_DESAT_SLOPE_MASK ) <<
CGEN_CFG_DESAT_SLOPE_SHIFT )

#define CGEN_CFG_DESAT_SLOPE_MASK 0xFF

#define CGEN_CFG_DESAT_SLOPE_SHIFT 0x10

```

```
#define CGEN_CFG_LLB_DSMODE( val ) (( val & CGEN_CFG_LLB_DSMODE_MASK ) <<
CGEN_CFG_LLB_DSMODE_SHIFT )
```

```
#define CGEN_CFG_LLB_DSMODE_MASK 0x3
```

Referenced by sippCheckOPipeConnectionGenChroma().

```
#define CGEN_CFG_LLB_DSMODE_SHIFT 0x1D
```

Referenced by sippCheckOPipeConnectionGenChroma().

```
#define CGEN_CFG_PURP_FLARE_STR( val ) (( val & CGEN_CFG_PU-
RP_FLARE_STR_MASK ) << CGEN_CFG_PURP_FLARE_STR_SHIFT
)
```

```
#define CGEN_CFG_PURP_FLARE_STR_MASK 0xFF
```

```
#define CGEN_CFG_PURP_FLARE_STR_SHIFT 0x0
```

```
#define CGEN_CFG_SET( purpFlareStr, desatOffset, desatSlope, dataWidth, bypass,
llbDSMode )
```

Value:

```
CGEN_CFG_PURP_FLARE_STR(purpFlareStr) |\
    (desatOffset) |\
    desatSlope) |\
    dataWidth) |\
    bypass) |\
    llbDSMode) CGEN_CFG_DESAT_OFFSET
CGEN_CFG_DESAT_SLOPE (
CGEN_CFG_DATA_WIDTH (
CGEN_CFG_BYPASS (
CGEN_CFG_LLB_DSMODE (
```

```
#define CGEN_CHROMA_COEFF_BLUE( val ) (( val & CGEN_CHROMA_COEFF_KB_MASK
) << CGEN_CHROMA_COEFF_KB_SHIFT )
```

```
#define CGEN_CHROMA_COEFF_EPSILON( val ) (( val & CGEN_CHROMA-
_COEFF_EPSILON_MASK ) << CGEN_CHROMA_COEFF_EPSILON_SHIFT
)
```

```
#define CGEN_CHROMA_COEFF_EPSILON_MASK 0xFF
```

```
#define CGEN_CHROMA_COEFF_EPSILON_SHIFT 0x0
```

```
#define CGEN_CHROMA_COEFF_GREEN( val ) (( val & CGEN_CHR-
OMA_COEFF_KG_MASK ) << CGEN_CHROMA_COEFF_KG_SHIFT
)
```

```
#define CGEN_CHROMA_COEFF_KB_MASK 0xFF
```

```
#define CGEN_CHROMA_COEFF_KB_SHIFT 0x18

#define CGEN_CHROMA_COEFF_KG_MASK 0xFF

#define CGEN_CHROMA_COEFF_KG_SHIFT 0x10

#define CGEN_CHROMA_COEFF_KR_MASK 0xFF

#define CGEN_CHROMA_COEFF_KR_SHIFT 0x8

#define CGEN_CHROMA_COEFF_RED( val ) (( val & CGEN_CHROMA_COEFF_KR_MASK ) << CGEN_CHROMA_COEFF_KR_SHIFT )

#define CGEN_CHROMA_COEFF_SET( epsilon, kr, kg, kb )
```

Value:

```
CGEN_CHROMA_COEFF_EPSILON(epsilon) |\
                                         CGEN_CHROMA_COEFF_RED(kr) |\
                                         CGEN_CHROMA_COEFF_GREEN(kg) |\
                                         CGEN_CHROMA_COEFF_BLUE(kb)

#define CGEN_LUMA_COEFF_BLUE( val ) (( val & CGEN_LUMA_COEFF_BLUE_MASK ) << CGEN_LUMA_COEFF_BLUE_SHIFT )

#define CGEN_LUMA_COEFF_BLUE_MASK 0xFF

#define CGEN_LUMA_COEFF_BLUE_SHIFT 0x10

#define CGEN_LUMA_COEFF_GREEN( val ) (( val & CGEN_LUMA_COEFF_GREEN_MASK ) << CGEN_LUMA_COEFF_GREEN_SHIFT )

#define CGEN_LUMA_COEFF_GREEN_MASK 0xFF

#define CGEN_LUMA_COEFF_GREEN_SHIFT 0x8

#define CGEN_LUMA_COEFF_RED( val ) (( val & CGEN_LUMA_COEFF_RED_MASK ) << CGEN_LUMA_COEFF_RED_SHIFT )

#define CGEN_LUMA_COEFF_RED_MASK 0xFF

#define CGEN_LUMA_COEFF_RED_SHIFT 0x0

#define CGEN_LUMA_COEFF_SET( red, green, blue )
```

Value:

```
CGEN_LUMA_COEFF_RED(red) |\
                                         CGEN_LUMA_COEFF_GREEN(green) |\
                                         CGEN_LUMA_COEFF_BLUE(blue)
```

```
#define CHORMA_GREY_PT_CB_MASK 0xFF

#define CHORMA_GREY_PT_CB_SET( val ) ((val & CHORMA_GREY_PT_CB_MASK) <<
CHORMA_GREY_PT_CB_SHIFT)

#define CHORMA_GREY_PT_CB_SHIFT 0x0

#define CHORMA_GREY_PT_CG_MASK 0xFF

#define CHORMA_GREY_PT_CG_SET( val ) ((val & CHORMA_GREY_PT_CG_MASK) <<
CHORMA_GREY_PT_CG_SHIFT)

#define CHORMA_GREY_PT_CG_SHIFT 0x8

#define CHORMA_GREY_PT_CR_MASK 0xFF

#define CHORMA_GREY_PT_CR_SET( val ) ((val & CHORMA_GREY_PT_CR_MASK) <<
CHORMA_GREY_PT_CR_SHIFT)

#define CHORMA_GREY_PT_CR_SHIFT 0x10

#define CHORMA_GREY_PT_PASSTHRU_MASK 0xFF

#define CHORMA_GREY_PT_PASSTHRU_SET( val ) ((val & CHORMA_GREY_PT_PASSTH-
RU_MASK) << CHORMA_GREY_PT_PASSTHRU_SHIFT)

#define CHORMA_GREY_PT_PASSTHRU_SHIFT 0x1F

#define CHORMA_GREY_PT_SET( cb, cg, cr, passthru )
```

Value:

```
CHORMA_GREY_PT_CB_SET(cb) |\
                                CHORMA_GREY_PT_CG_SET(cg) |\
                                CHORMA_GREY_PT_CR_SET(cr) |\
                                CHORMA_GREY_PT_PASSTHRU_SET(passthru)
```

```
#define CHROMA_DNS_CFG( horzEnable, limit, forceWgthsHor, forceWgthsVer, planesMode,
desatOffset, desatSlope )
```

Value:

```
CHROMA_DNS_HORZ_ENABLE(horzEnable) |\
                                CHROMA_DNS_LIMIT(limit) |\
                                CHROMA_DNS_FORCE_WEIGHTS_HORIZONTAL(forceWgthsHor) |\
                                CHROMA_DNS_FORCE_WEIGHTS_VERTICAL(forceWgthsVer) |\
                                CHROMA_DNS_PLANES_MODE(planesMode) |\
                                CHROMA_DNS_DESAT_OFFSET(desatOffset) |\
                                CHROMA_DNS_DESAT_SLOPE(desatSlope)
```



```
#define CHROMA_DNS_CFG_THRESH12( horzThr1, horzThr2, vertThr1, vertThr2 )
```

Value:

```
CHROMA_DNS_VER_T2 (vertThr2) +\
CHROMA_DNS_HOR_T2 (horzThr2) +\
CHROMA_DNS_HOR_T1 (horzThr1) +\
CHROMA_DNS_VER_T1 (vertThr1)
```

```
#define CHROMA_DNS_CFG_THRESH3( horzThr3, vertThr3 )
```

Value:

```
CHROMA_DNS_HOR_T3 (horzThr3) +\
CHROMA_DNS_VER_T3 (vertThr3)
```

```
#define CHROMA_DNS_COEFFS_CENTRE_EDGE_MASK 0xFF
```

```
#define CHROMA_DNS_COEFFS_CENTRE_EDGE_SET( val ) ((val & CHROMA_DNS_CO-  
EFFS_CENTRE_EDGE_MASK) << CHROMA_DNS_COEFFS_CENTRE_EDGE_SHIFT  
)
```

```
#define CHROMA_DNS_COEFFS_CENTRE_EDGE_SHIFT 0x8
```

```
#define CHROMA_DNS_COEFFS_CENTRE_MASK 0xFF
```

```
#define CHROMA_DNS_COEFFS_CENTRE_SET( val ) ((val & CHROMA_DNS-  
S_COEFFS_CENTRE_MASK) << CHROMA_DNS_COEFFS_CENTRE_SHIFT  
)
```

```
#define CHROMA_DNS_COEFFS_CENTRE_SHIFT 0x0
```

```
#define CHROMA_DNS_COEFFS_CORNER_MASK 0xFF
```

```
#define CHROMA_DNS_COEFFS_CORNER_SET( val ) ((val & CHROMA_DNS-  
_COEFFS_CORNER_MASK) << CHROMA_DNS_COEFFS_CORNER_SHIFT  
)
```

```
#define CHROMA_DNS_COEFFS_CORNER_SHIFT 0x10
```

```
#define CHROMA_DNS_COEFFS_SET( centre, centreEdge, corner )
```

Value:

```
CHROMA_DNS_COEFFS_CENTRE_SET(centre) |\  
CHROMA_DNS_COEFFS_CENTRE_EDGE_SET(centreEdge) |\  
CHROMA_DNS_COEFFS_CORNER_SET(corner)
```

```

#define CHROMA_DNS_DESAT_OFFSET( val ) ((val & CHROMA_DNS-  

_DESAT_OFFSET_MASK) << CHROMA_DNS_DESAT_OFFSET_SHIFT  

)

#define CHROMA_DNS_DESAT_OFFSET_MASK 0xFF

#define CHROMA_DNS_DESAT_OFFSET_SHIFT 0x10

#define CHROMA_DNS_DESAT_SLOPE( val ) ((val & CHROMA_DN-  

S_DESAT_SLOPE_MASK) << CHROMA_DNS_DESAT_SLOPE_SHIFT  

)

#define CHROMA_DNS_DESAT_SLOPE_MASK 0xFF

#define CHROMA_DNS_DESAT_SLOPE_SHIFT 0x18

#define CHROMA_DNS_FORCE_WEIGHTS_HORIZONTAL( val ) ((val  

& CHROMA_DNS_FORCE_WEIGHTS_HORIZONTAL_MASK) <<  

CHROMA_DNS_FORCE_WEIGHTS_HORIZONTAL_SHIFT)

#define CHROMA_DNS_FORCE_WEIGHTS_HORIZONTAL_MASK 0x1

#define CHROMA_DNS_FORCE_WEIGHTS_HORIZONTAL_SHIFT 0xC

#define CHROMA_DNS_FORCE_WEIGHTS_VERTICAL( val ) ((val &  

CHROMA_DNS_FORCE_WEIGHTS_VERTICAL_MASK) << CHROMA_DNS_FORCE_W-  

EIGHTS_VERTICAL_SHIFT)

#define CHROMA_DNS_FORCE_WEIGHTS_VERTICAL_MASK 0x1

#define CHROMA_DNS_FORCE_WEIGHTS_VERTICAL_SHIFT 0xD

#define CHROMA_DNS_HOR_T1( val ) (( val & CHROMA_DNS_HOR_T1_MASK) <<  

CHROMA_DNS_HOR_T1_SHIFT)

#define CHROMA_DNS_HOR_T1_MASK 0xFF

#define CHROMA_DNS_HOR_T1_SHIFT 0x0

#define CHROMA_DNS_HOR_T2( val ) (( val & CHROMA_DNS_HOR_T2_MASK) <<  

CHROMA_DNS_HOR_T2_SHIFT)

#define CHROMA_DNS_HOR_T2_MASK 0xFF

#define CHROMA_DNS_HOR_T2_SHIFT 0x8

#define CHROMA_DNS_HOR_T3( val ) (( val & CHROMA_DNS_HOR_T3_MASK) <<  

CHROMA_DNS_HOR_T3_SHIFT)

#define CHROMA_DNS_HOR_T3_MASK 0xFF

```

```

#define CHROMA_DNS_HOR_T3_SHIFT 0x0

#define CHROMA_DNS_HORZ_ENABLE( val ) ((val & CHROMA_DNS-  

_HORZ_ENABLE_MASK) << CHROMA_DNS_HORZ_ENABLE_SHIFT  

)

#define CHROMA_DNS_HORZ_ENABLE_MASK 0x7

#define CHROMA_DNS_HORZ_ENABLE_SHIFT 0x0

#define CHROMA_DNS_LIMIT( val ) ((val & CHROMA_DNS_LIMIT_MASK) <<  

CHROMA_DNS_LIMIT_SHIFT)

#define CHROMA_DNS_LIMIT_MASK 0xFF

#define CHROMA_DNS_LIMIT_SHIFT 0x4

#define CHROMA_DNS_PLANES_MODE( val ) ((val & CHROMA_DNS-  

_PLANES_MODE_MASK) << CHROMA_DNS_PLANES_MODE_SHIFT  

)

#define CHROMA_DNS_PLANES_MODE_MASK 0x3

#define CHROMA_DNS_PLANES_MODE_SHIFT 0xE

#define CHROMA_DNS_VER_T1( val ) (( val & CHROMA_DNS_VER_T1_MASK) <<  

CHROMA_DNS_VER_T1_SHIFT)

#define CHROMA_DNS_VER_T1_MASK 0xFF

#define CHROMA_DNS_VER_T1_SHIFT 0x10

#define CHROMA_DNS_VER_T2( val ) (( val & CHROMA_DNS_VER_T2_MASK) <<  

CHROMA_DNS_VER_T2_SHIFT)

#define CHROMA_DNS_VER_T2_MASK 0xFF

#define CHROMA_DNS_VER_T2_SHIFT 0x18

#define CHROMA_DNS_VER_T3( val ) (( val & CHROMA_DNS_VER_T3_MASK) <<  

CHROMA_DNS_VER_T3_SHIFT)

#define CHROMA_DNS_VER_T3_MASK 0xFF

#define CHROMA_DNS_VER_T3_SHIFT 0x10

#define CONV_ACCUM_THRESHOLD( val ) ((val & CONV_CFG_ACCUM_THRESHOLD_M-  

ASK) << CONV_CFG_ACCUM_THRESHOLD_SHIFT)

#define CONV_ACCUMULATION_EN( val ) ((val & CONV_CFG_ACCUMULATION_EN_M-  

ASK) << CONV_CFG_ACCUMULATION_EN_SHIFT)

```

```
#define CONV_CFG( kernSize, outClamp, outAbsVal, outSquare, accEn, filterDis, accTresh,
evenOddPix, evenOddLine )
```

Value:

```
CONV_KERNEL_SIZE(kernSize) |\
```

```
CONV_OUTPUT_CLAMP(outClamp) |\
```

```
CONV_OUTPUT_ABS_VAL(outAbsVal) |\
```

```
CONV_OUTPUT_SQUARE(outSquare) |\
```

```
CONV_ACCUMULATION_EN(accEn) |\
```

```
CONV_FILTER_DISABLE(filterDis) |\
```

```
CONV_ACCUM_THRESHOLD(accTresh) |\
```

```
CONV_EVENODD_PIX(evenOddPix) |\
```

```
CONV_EVENODD_LINE(evenOddLine)
```

```
#define CONV_CFG_ACCUM_THRESHOLD_MASK 0xFFFF
```

```
#define CONV_CFG_ACCUM_THRESHOLD_SHIFT 0x8
```

```
#define CONV_CFG_ACCUMULATION_EN_MASK 0x1
```

```
#define CONV_CFG_ACCUMULATION_EN_SHIFT 0x6
```

```
#define CONV_CFG_EVENODD_LINE_MASK 0x1
```

```
#define CONV_CFG_EVENODD_LINE_SHIFT 0x19
```

```
#define CONV_CFG_EVENODD_PIX_MASK 0x1
```

```
#define CONV_CFG_EVENODD_PIX_SHIFT 0x18
```

```
#define CONV_CFG_FILTER_DISABLE_MASK 0x1
```

```
#define CONV_CFG_FILTER_DISABLE_SHIFT 0x7
```

```
#define CONV_CFG_KERNEL_SIZE_MASK 0x7
```

```
#define CONV_CFG_KERNEL_SIZE_SHIFT 0x0
```

```
#define CONV_CFG_OUTPUT_ABS_VAL_MASK 0x1
```

```
#define CONV_CFG_OUTPUT_ABS_VAL_SHIFT 0x4
```

```
#define CONV_CFG_OUTPUT_CLAMP_EN_MASK 0x1
```

```
#define CONV_CFG_OUTPUT_CLAMP_EN_SHIFT 0x3
```

```
#define CONV_CFG_OUTPUT_SQUARE_EN_MASK 0x1

#define CONV_CFG_OUTPUT_SQUARE_EN_SHIFT 0x5

#define CONV_EVENODD_LINE( val ) ((val & CONV_CFG_EVENODD_LINE_MASK) <<
CONV_CFG_EVENODD_LINE_SHIFT)

#define CONV_EVENODD_PIX( val ) ((val & CONV_CFG_EVENODD_PIX_MASK) <<
CONV_CFG_EVENODD_PIX_SHIFT)

#define CONV_FILTER_DISABLE( val ) ((val & CONV_CFG_FILTER_DISABLE_MASK) <<
CONV_CFG_FILTER_DISABLE_SHIFT)

#define CONV_KERNEL_SIZE( val ) ((val & CONV_CFG_KERNEL_SIZE_MASK) <<
CONV_CFG_KERNEL_SIZE_SHIFT)

#define CONV_KSZ_3x3 CONV_KERNEL_SIZE(0x3)

#define CONV_KSZ_5x5 CONV_KERNEL_SIZE(0x5)

#define CONV_OUTPUT_ABS_VAL( val ) ((val & CONV_CFG_OUTPUT_ABS_VAL_MASK)
<< CONV_CFG_OUTPUT_ABS_VAL_SHIFT)

#define CONV_OUTPUT_CLAMP( val ) ((val & CONV_CFG_OUTPUT_CLAMP_EN_MASK)
<< CONV_CFG_OUTPUT_CLAMP_EN_SHIFT)

#define CONV_OUTPUT_SQUARE( val ) ((val & CONV_CFG_OUTPUT_SQUARE_EN_MAS-
K) << CONV_CFG_OUTPUT_SQUARE_EN_SHIFT)

#define DEBAYER_ABS_THRESH1( val ) ((val & DEBAYER_THRESHOLD_ABS_THRESH1-
_MASK) << DEBAYER_THRESHOLD_ABS_THRESH1_SHIFT)

#define DEBAYER_ABS_THRESH2( val ) ((val & DEBAYER_THRESHOLD_ABS_THRESH2-
_MASK) << DEBAYER_THRESHOLD_ABS_THRESH2_SHIFT)

#define DEBAYER_BAYER_ORDER( val ) ((val & DEBAYER_CFG_BAYER_ORDER_MASK)
<< DEBAYER_CFG_BAYER_ORDER_SHIFT)

#define DEBAYER_CFG( bayerOrder, lumaOnly, forceRBZero, inDataWidth, outDataWidth,
outplaneOrder, numOutplanes, gradMultiplier )
```

Value:

```
DEBAYER_BAYER_ORDER(bayerOrder) +\
    DEBAYER_LUMA_ONLY(lumaOnly) +\
    DEBAYER_FORCE_RB_ZERO(forceRBZero) +\
    DEBAYER_IP_DATAWIDTH((inDataWidth ? inDataWidth - 0x1 : inDataWidth)) +
    \
    DEBAYER_OP_DATAWIDTH((outDataWidth ? outDataWidth - 0x1 : outDataWidth)
    ) +\
    DEBAYER_OUTPLANE_ORDER(outplaneOrder) +\
    DEBAYER_NUM_OUTPLANES((numOutplanes ? numOutplanes - 0x1 : numOutplanes
    )) +\
    DEBAYER_GRAD_MULTIPLIER(gradMultiplier)
```

```
#define DEBAYER_CFG_BAYER_ORDER_MASK 0x3

#define DEBAYER_CFG_BAYER_ORDER_SHIFT 0x0

#define DEBAYER_CFG_FORCE_RB_ZERO_MASK 0x1

#define DEBAYER_CFG_FORCE_RB_ZERO_SHIFT 0x3

#define DEBAYER_CFG_GRAD_MULTIPLIER_MASK 0xFF

#define DEBAYER_CFG_GRAD_MULTIPLIER_SHIFT 0x18

#define DEBAYER_CFG_IP_DATAWIDTH_M1_MASK 0xF

#define DEBAYER_CFG_IP_DATAWIDTH_M1_SHIFT 0x4

#define DEBAYER_CFG_LUMA_ONLY_MASK 0x1

#define DEBAYER_CFG_LUMA_ONLY_SHIFT 0x2

#define DEBAYER_CFG_NUM_OUTPLANES_MASK 0x3

#define DEBAYER_CFG_NUM_OUTPLANES_SHIFT 0xF

#define DEBAYER_CFG_OP_DATAWIDTH_M1_MASK 0xF

#define DEBAYER_CFG_OP_DATAWIDTH_M1_SHIFT 0x8

#define DEBAYER_CFG_OUTPLANE_ORDER_MASK 0x7

#define DEBAYER_CFG_OUTPLANE_ORDER_SHIFT 0xC

#define DEBAYER_DEWORM( offset, slope )
```

Value:

```
DEBAYER_OFFSET( offset ) | \
                                DEBAYER_SLOPE( slope )
```

```
#define DEBAYER_DEWORM_OFFSET_MASK 0xFFFF

#define DEBAYER_DEWORM_OFFSET_SHIFT 0x10

#define DEBAYER_DEWORM_SLOPE_MASK 0xFFFF

#define DEBAYER_DEWORM_SLOPE_SHIFT 0x0

#define DEBAYER_FORCE_RB_ZERO( val ) ((val & DEBAYER_CFG_FORCE_RB_ZERO_M-
ASK) << DEBAYER_CFG_FORCE_RB_ZERO_SHIFT)
```

```
#define DEBAYER_GRAD_MULTIPLIER( val ) ((val & DEBAYER_CFG_GRAD_MULTIPLIER_MASK) << DEBAYER_CFG_GRAD_MULTIPLIER_SHIFT)

#define DEBAYER_IP_DATAWIDTH( val ) ((val & DEBAYER_CFG_IP_DATAWIDTH_M1_MASK) << DEBAYER_CFG_IP_DATAWIDTH_M1_SHIFT)

#define DEBAYER_LUMA_BLUE_COEF( val ) ((val & DEBAYER_LUMA_BLUE_COEF_MASK) << DEBAYER_LUMA_BLUE_COEF_SHIFT)

#define DEBAYER_LUMA_BLUE_COEF_MASK 0xFF

#define DEBAYER_LUMA_BLUE_COEF_SHIFT 0x00

#define DEBAYER_LUMA_COEFS_CFG( red, green, blue )
```

Value:

```
DEBAYER_LUMA_RED_COEF (red) |\
DEBAYER_LUMA_GREEN_COEF (green) |\
DEBAYER_LUMA_BLUE_COEF (blue)

#define DEBAYER_LUMA_EN( val ) ((val & DEBAYER_THRESHOLD_LUMA_EN_MASK) << DEBAYER_THRESHOLD_LUMA_EN_SHIFT)

#define DEBAYER_LUMA_GREEN_COEF( val ) ((val & DEBAYER_LUMA_GREEN_COEF_MASK) << DEBAYER_LUMA_GREEN_COEF_SHIFT)

#define DEBAYER_LUMA_GREEN_COEF_MASK 0xFF

#define DEBAYER_LUMA_GREEN_COEF_SHIFT 0x8

#define DEBAYER_LUMA_ONLY( val ) ((val & DEBAYER_CFG_LUMA_ONLY_MASK) << DEBAYER_CFG_LUMA_ONLY_SHIFT)

#define DEBAYER_LUMA_RED_COEF( val ) ((val & DEBAYER_LUMA_RED_COEF_MASK) << DEBAYER_LUMA_RED_COEF_SHIFT)

#define DEBAYER_LUMA_RED_COEF_MASK 0xFF

#define DEBAYER_LUMA_RED_COEF_SHIFT 0x10

#define DEBAYER_LUMA_WC_EN( val ) ((val & DEBAYER_THRESHOLD_LUMA_WC_EN_MASK) << DEBAYER_THRESHOLD_LUMA_WC_EN_SHIFT)

#define DEBAYER_NUM_OUTPLANES( val ) ((val & DEBAYER_CFG_NUM_OUTPLANES_MASK) << DEBAYER_CFG_NUM_OUTPLANES_SHIFT)

#define DEBAYER_OFFSET( val ) ((val & DEBAYER_DEWORM_OFFSET_MASK) << DEBAYER_DEWORM_OFFSET_SHIFT)
```

```
#define DEBAYER_OP_DATAWIDTH( val ) ((val & DEBAYER_CFG_OP_DATAWIDTH_M1-  
_MASK) << DEBAYER_CFG_OP_DATAWIDTH_M1_SHIFT)
```

```
#define DEBAYER_OUTPLANE_ORDER( val ) ((val & DEBAYER_CFG_OUTPLANE_ORDE-  
R_MASK) << DEBAYER_CFG_OUTPLANE_ORDER_SHIFT)
```

```
#define DEBAYER_PREVIEW_EN( val ) ((val & DEBAYER_THRESHOLD_PREVIEW_EN_-  
MASK) << DEBAYER_THRESHOLD_PREVIEW_EN_SHIFT)
```

```
#define DEBAYER_RGB_EN( val ) ((val & DEBAYER_THRESHOLD_RGB_EN_MASK) <<  
DEBAYER_THRESHOLD_RGB_EN_SHIFT)
```

Referenced by `sippBufGetObufCtx()`.

```
#define DEBAYER_SLOPE( val ) ((val & DEBAYER_DEWORM_SLOPE_MASK) <<  
DEBAYER_DEWORM_SLOPE_SHIFT)
```

```
#define DEBAYER_THRESH( absThresh1, absThresh2, rgbEn, lumaEn, lumaWc, prewEn )
```

Value:

```
DEBAYER_ABS_THRESH1(absThresh1) | \
    \
    DEBAYER_ABS_THRESH2(absThresh2) |
    DEBAYER_RGB_EN(rgbEn) | \
    DEBAYER_LUMA_EN(lumaEn) | \
    DEBAYER_LUMA_WC_EN(lumaWc) | \
    DEBAYER_PREVIEW_EN(prewEn)
```

```
#define DEBAYER_THRESHOLD_ABS_THRESH1_MASK 0x1FFF
```

```
#define DEBAYER_THRESHOLD_ABS_THRESH1_SHIFT 0x0
```

```
#define DEBAYER_THRESHOLD_ABS_THRESH2_MASK 0xFFFF
```

```
#define DEBAYER_THRESHOLD_ABS_THRESH2_SHIFT 0xD
```

```
#define DEBAYER_THRESHOLD_LUMA_EN_MASK 0x1
```

```
#define DEBAYER_THRESHOLD_LUMA_EN_SHIFT 0x1A
```

```
#define DEBAYER_THRESHOLD_LUMA_WC_EN_MASK 0x1
```

```
#define DEBAYER_THRESHOLD_LUMA_WC_EN_SHIFT 0x1B
```

```
#define DEBAYER_THRESHOLD_PREVIEW_EN_MASK 0x1
```

```
#define DEBAYER_THRESHOLD_PREVIEW_EN_SHIFT 0x1C
```

```
#define DEBAYER_THRESHOLD_RGB_EN_MASK 0x1
```

```
#define DEBAYER_THRESHOLD_RGB_EN_SHIFT 0x19
```



```

#define DOGL_CFG_BACK_GEN_THRESH( val ) (( val & DOGL_CFG_BACK_GEN_THRESH_MASK ) << DOGL_CFG_BACK_GEN_THRESH_SHIFT )

#define DOGL_CFG_BACK_GEN_THRESH_MASK 0xFF

#define DOGL_CFG_BACK_GEN_THRESH_SHIFT 0xE

#define DOGL_CFG_DOG_THRESHOLD( val ) (( val & DOGL_CFG_DOG_THRESHOLD_MASK ) << DOGL_CFG_DOG_THRESHOLD_SHIFT )

#define DOGL_CFG_DOG_THRESHOLD_MASK 0xFF

#define DOGL_CFG_DOG_THRESHOLD_SHIFT 0x2

#define DOGL_CFG_KERNEL_HEIGHT( val ) (( val & DOGL_CFG_KERNEL_HEIGHT_MASK ) << DOGL_CFG_KERNEL_HEIGHT_SHIFT )

#define DOGL_CFG_KERNEL_HEIGHT_MASK 0xF

#define DOGL_CFG_KERNEL_HEIGHT_SHIFT 0x1A

#define DOGL_CFG_LLB_DSMODE( val ) (( val & DOGL_CFG_LLB_DSMODE_MASK ) << DOGL_CFG_LLB_DSMODE_SHIFT )

#define DOGL_CFG_LLB_DSMODE_MASK 0x3

#define DOGL_CFG_LLB_DSMODE_SHIFT 0xC

#define DOGL_CFG_MODE( val ) (( val & DOGL_CFG_MODE_MASK ) << DOGL_CFG_MODE_SHIFT )

#define DOGL_CFG_MODE_MASK 0x3

#define DOGL_CFG_MODE_SHIFT 0x0

#define DOGL_CFG_NUM_PLANES( val ) (( val & DOGL_CFG_NUM_PLANES_MASK ) << DOGL_CFG_NUM_PLANES_SHIFT )

#define DOGL_CFG_NUM_PLANES_MASK 0xF

#define DOGL_CFG_NUM_PLANES_SHIFT 0x16

#define DOGL_CFG_OUTPUT_CLAMP( val ) (( val & DOGL_CFG_OUTPUT_CLAMP_MASK ) << DOGL_CFG_OUTPUT_CLAMP_SHIFT )

#define DOGL_CFG_OUTPUT_CLAMP_MASK 0x1

#define DOGL_CFG_OUTPUT_CLAMP_SHIFT 0xA

```

```
#define DOGL_CFG_SET( mode, thresh, outClamp, llbDSMode, backGenThresh, numPlanes,
kernelHeight )
```

Value:

```
DOGL_CFG_MODE (mode)  |\
    DOGL_CFG_DOG_THRESHOLD (thresh)  |\
    DOGL_CFG_OUTPUT_CLAMP (outClamp)  |\
    DOGL_CFG_LLBD_SMODE (llbDSMode)  |\
    DOGL_CFG_BACK_GEN_THRESH (backGenThresh)  |\
    DOGL_CFG_NUM_PLANES (numPlanes)  |\
    DOGL_CFG_KERNEL_HEIGHT (kernelHeight)

#define DOGL_MODE_DOG_DENOISE 0x2

#define DOGL_MODE_DOG_LTM 0x3

#define DOGL_MODE_DOG_ONLY 0x0

#define DOGL_MODE_LTM_ONLY 0x1

#define EDGE_OPERATOR_BUFFER_HEIGHT_MASK 0xFFFF

#define EDGE_OPERATOR_BUFFER_HEIGHT_SHIFT 0x10

#define EDGE_OPERATOR_BUFFER_WIDTH_MASK 0xFFFF

#define EDGE_OPERATOR_BUFFER_WIDTH_SHIFT 0x0

#define EDGE_OPERATOR_CFG( inMode, outMode, thetaMode, magScaleFactor,
thetaScaleFactor )
```

Value:

```
EDGE_OPERATOR_INPUT_MODE (inMode)  |\
    EDGE_OPERATOR_OUTPUT_MODE (outMode)  |\
    EDGE_OPERATOR_THETA_MODE (thetaMode)  |\
    EDGE_OPERATOR_MAG_SCALE_FACTOR (magScaleFactor)  |\
    EDGE_OPERATOR_THETA_OVX (thetaScaleFactor)

#define EDGE_OPERATOR_INPUT_MODE( val ) ((val & EDGE_OPERATOR_INPUT_MODE_MASK) << EDGE_OPERATOR_INPUT_MODE_SHIFT
)

#define EDGE_OPERATOR_INPUT_MODE_MASK 0x2
```

```

#define EDGE_OPERATOR_INPUT_MODE_SHIFT 0x0

#define EDGE_OPERATOR_MAG_SCALE_FACTOR( val ) ((val &
EDGE_OPERATOR_MAGNITUDE_SCALE_FACTOR_MASK) <<
EDGE_OPERATOR_MAGNITUDE_SCALE_FACTOR_SHIFT)

#define EDGE_OPERATOR_MAGNITUDE_SCALE_FACTOR_MASK 0xFFFF

#define EDGE_OPERATOR_MAGNITUDE_SCALE_FACTOR_SHIFT 0x10

#define EDGE_OPERATOR_OUTPUT_MODE( val ) ((val & EDGE_OPERATOR_OUTPUT_M-
ODE_MASK) << EDGE_OPERATOR_OUTPUT_MODE_SHIFT)

#define EDGE_OPERATOR_OUTPUT_MODE_MASK 0x7

#define EDGE_OPERATOR_OUTPUT_MODE_SHIFT 0x2

#define EDGE_OPERATOR_THETA_MODE( val ) ((val & EDGE_OPERATOR-
R_THETA_MODE_MASK) << EDGE_OPERATOR_THETA_MODE_SHIFT
)

#define EDGE_OPERATOR_THETA_MODE_MASK 0x3

#define EDGE_OPERATOR_THETA_MODE_SHIFT 0x5

#define EDGE_OPERATOR_THETA_OVX( val ) ((val & EDGE_OPERATOR-
OR_THETA_OVX_MASK) << EDGE_OPERATOR_THETA_OVX_SHIFT
)

#define EDGE_OPERATOR_THETA_OVX_MASK 0x1

#define EDGE_OPERATOR_THETA_OVX_SHIFT 0x7

#define EDGE_OPERATOR_XCOEFF_A( val ) ((val & EDGE_OPERATOR_XCOEFF_A_MAS-
K) << EDGE_OPERATOR_XCOEFF_A_SHIFT)

#define EDGE_OPERATOR_XCOEFF_A_MASK 0x1F

#define EDGE_OPERATOR_XCOEFF_A_SHIFT 0x0

#define EDGE_OPERATOR_XCOEFF_B( val ) ((val & EDGE_OPERATOR_XCOEFF_B_MAS-
K) << EDGE_OPERATOR_XCOEFF_B_SHIFT)

#define EDGE_OPERATOR_XCOEFF_B_MASK 0x1F

#define EDGE_OPERATOR_XCOEFF_B_SHIFT 0x5

#define EDGE_OPERATOR_XCOEFF_C( val ) ((val & EDGE_OPERATOR_XCOEFF_C_MAS-
K) << EDGE_OPERATOR_XCOEFF_C_SHIFT)

#define EDGE_OPERATOR_XCOEFF_C_MASK 0x1F

```

```
#define EDGE_OPERATOR_XCOEFF_C_SHIFT 0xA

#define EDGE_OPERATOR_XCOEFF_D( val ) ((val & EDGE_OPERATOR_XCOEFF_D_MASK) << EDGE_OPERATOR_XCOEFF_D_SHIFT)

#define EDGE_OPERATOR_XCOEFF_D_MASK 0x1F

#define EDGE_OPERATOR_XCOEFF_D_SHIFT 0xF

#define EDGE_OPERATOR_XCOEFF_E( val ) ((val & EDGE_OPERATOR_XCOEFF_E_MASK) << EDGE_OPERATOR_XCOEFF_E_SHIFT)

#define EDGE_OPERATOR_XCOEFF_E_MASK 0x1F

#define EDGE_OPERATOR_XCOEFF_E_SHIFT 0x14

#define EDGE_OPERATOR_XCOEFF_F( val ) ((val & EDGE_OPERATOR_XCOEFF_F_MASK) << EDGE_OPERATOR_XCOEFF_F_SHIFT)

#define EDGE_OPERATOR_XCOEFF_F_MASK 0x1F

#define EDGE_OPERATOR_XCOEFF_F_SHIFT 0x19

#define EDGE_OPERATOR_XCOEFS( a, b, c, d, e, f )
```

Value:

```
EDGE_OPERATOR_XCOEFF_A(a) | \
                                EDGE_OPERATOR_XCOEFF_B(b) | \
                                EDGE_OPERATOR_XCOEFF_C(c) | \
                                EDGE_OPERATOR_XCOEFF_D(d) | \
                                EDGE_OPERATOR_XCOEFF_E(e) | \
                                EDGE_OPERATOR_XCOEFF_F(f) \
```

```
#define EDGE_OPERATOR_YCOEF_A( val ) ((val & EDGE_OPERATOR_YCOEF_A_MASK) << EDGE_OPERATOR_YCOEF_A_SHIFT)

#define EDGE_OPERATOR_YCOEF_A_MASK 0x1F

#define EDGE_OPERATOR_YCOEF_A_SHIFT 0x0

#define EDGE_OPERATOR_YCOEF_B( val ) ((val & EDGE_OPERATOR_YCOEF_B_MASK) << EDGE_OPERATOR_YCOEF_B_SHIFT)

#define EDGE_OPERATOR_YCOEF_B_MASK 0x1F

#define EDGE_OPERATOR_YCOEF_B_SHIFT 0x5

#define EDGE_OPERATOR_YCOEF_C( val ) ((val & EDGE_OPERATOR_YCOEF_C_MASK) << EDGE_OPERATOR_YCOEF_C_SHIFT)

#define EDGE_OPERATOR_YCOEF_C_MASK 0x1F
```

```
#define EDGE_OPERATOR_YCOEF_C_SHIFT 0xA

#define EDGE_OPERATOR_YCOEF_D( val ) ((val & EDGE_OPERATOR_YCOEF_D_MASK)
<< EDGE_OPERATOR_YCOEF_D_SHIFT)

#define EDGE_OPERATOR_YCOEF_D_MASK 0x1F

#define EDGE_OPERATOR_YCOEF_D_SHIFT 0xF

#define EDGE_OPERATOR_YCOEF_E( val ) ((val & EDGE_OPERATOR_YCOEF_E_MASK)
<< EDGE_OPERATOR_YCOEF_E_SHIFT)

#define EDGE_OPERATOR_YCOEF_E_MASK 0x1F

#define EDGE_OPERATOR_YCOEF_E_SHIFT 0x14

#define EDGE_OPERATOR_YCOEF_F( val ) ((val & EDGE_OPERATOR_YCOEF_F_MASK)
<< EDGE_OPERATOR_YCOEF_F_SHIFT)

#define EDGE_OPERATOR_YCOEF_F_MASK 0x1F

#define EDGE_OPERATOR_YCOEF_F_SHIFT 0x19

#define EDGE_OPERATOR_YCOEFS( a, b, c, d, e, f )
```

Value:

```
EDGE_OPERATOR_YCOEF_A(a) |\
                                EDGE_OPERATOR_YCOEF_B(b) |\
                                EDGE_OPERATOR_YCOEF_C(c) |\
                                EDGE_OPERATOR_YCOEF_D(d) |\
                                EDGE_OPERATOR_YCOEF_E(e) |\
                                EDGE_OPERATOR_YCOEF_F(f) \
```

```
#define HARISS_CORNER_CFG_EXP_SUBTRAHEND( val ) ((val & HARISS_CORNER_CFG-
_EXP_SUBTRAHEND_MASK) << HARRIS_CORNER_CFG_EXP_SUBTRAHEND_SHIFT)

#define HARISS_CORNER_CFG_EXP_SUBTRAHEND_MASK 0xFF

#define HARRIS_CORNER_CFG( kns, outDet, expSubtrahend )
```

Value:

```
HARRIS_CORNER_CFG_KERNEL_SIZE(kns) |\
                                HARRIS_CORNER_CFG_OUTPUT_DETERMINANT(outDet) |\
                                HARISS_CORNER_CFG_EXP_SUBTRAHEND(expSubtrahend)
```

```
#define HARRIS_CORNER_CFG_EXP_SUBTRAHEND_SHIFT 0x8

#define HARRIS_CORNER_CFG_KERNEL_SIZE( val ) ((val & HARRIS_CORNER_CFG_KE-
RNEL_SIZE_MASK) << HARRIS_CORNER_CFG_KERNEL_SIZE_SHIFT)
```

```
#define HARRIS_CORNER_CFG_KERNEL_SIZE_MASK 0xF

#define HARRIS_CORNER_CFG_KERNEL_SIZE_SHIFT 0x0

#define HARRIS_CORNER_CFG_OUTPUT_DETERMINANT( val ) ((val
& HARRIS_CORNER_CFG_OUTPUT_DETERMINANT_MASK) <<
HARRIS_CORNER_CFG_OUTPUT_DETERMINANT_SHIFT)

#define HARRIS_CORNER_CFG_OUTPUT_DETERMINANT_MASK 0x1

#define HARRIS_CORNER_CFG_OUTPUT_DETERMINANT_SHIFT 0x4

#define HARRIS_CORNER_KSZ_5 0x5

#define HARRIS_CORNER_KSZ_7 0x7

#define HARRIS_CORNER_KSZ_9 0x9

#define LUMA_DNS_CFG( loadEn, bitShift, cfgAlpha, cfgBitpos )
```

Value:

```
LUMA_DNS_CFG_C4_LUT_LOAD_EN( loadEn ) |\
LUMA_DNS_CFG_BITSHIFT( bitShift ) |\
LUMA_DNS_CFG_ALPHA( cfgAlpha ) |\
LUMA_DNS_CFG_BITPOS( cfgBitpos)

#define LUMA_DNS_CFG_ALPHA( val ) (( val & LUMA_DNS_CFG_ALPHA_MASK ) <<
LUMA_DNS_CFG_ALPHA_SHIFT )

#define LUMA_DNS_CFG_ALPHA_MASK 0xFF

#define LUMA_DNS_CFG_ALPHA_SHIFT 0x8

#define LUMA_DNS_CFG_BITPOS( val ) (( val & LUMA_DNS_CFG_BITPOS_MASK ) <<
LUMA_DNS_CFG_BITPOS_SHIFT )

#define LUMA_DNS_CFG_BITPOS_MASK 0xF

#define LUMA_DNS_CFG_BITPOS_SHIFT 0x0

#define LUMA_DNS_CFG_BITSHIFT( val ) (( val & LUMA_DNS_CFG_BITSHIFT_MASK )
<< LUMA_DNS_CFG_BITSHIFT_SHIFT )

#define LUMA_DNS_CFG_BITSHIFT_MASK 0x1F

#define LUMA_DNS_CFG_BITSHIFT_SHIFT 0x10

#define LUMA_DNS_CFG_C4_LUT_LOAD_EN( val ) (( val & LUMA_DNS_CFG-
_C4_LUT_LOAD_EN_MASK ) << LUMA_DNS_CFG_C4_LUT_LOAD_EN_SHIFT
)
```

```
#define LUMA_DNS_CFG_C4_LUT_LOAD_EN_MASK 0x1

#define LUMA_DNS_CFG_C4_LUT_LOAD_EN_SHIFT 0x18

#define LUT_CFG( colorConv, apbAccesEN, loadEn, numChanel, numLuts, intModew,
channelMode, interpolateFp16 )
```

Value:

```
LUT_CFG_COLOR_CONVERSION_EN( colorConv ) |\
    LUT_CFG_APB_ACCESS_EN( apbAccesEN ) |\
    LUT_CFG_LUT_LOAD_EN( loadEn ) |\
    LUT_CFG_NUM_CHANNELS( numChanel ) |\
    LUT_CFG_NUM_LUTS( numLuts ) |\
    LUT_CFG_INTEGER_MODE_WIDTH( intModew ) |\
    LUT_CFG_CHANNEL_MODE( channelMode ) |\
    LUT_CFG_INTERPOLATE_FP16( interpolateFp16 )

#define LUT_CFG_APB_ACCESS_EN( val ) (( val & LUT_CFG_APB_ACCESS_EN_MASK )
<< LUT_CFG_APB_ACCESS_EN_SHIFT )

#define LUT_CFG_APB_ACCESS_EN_MASK 0x1

#define LUT_CFG_APB_ACCESS_EN_SHIFT 0xF

#define LUT_CFG_CHANNEL_MODE( val ) (( val & LUT_CFG_CHANNEL_MODE_MASK )
<< LUT_CFG_CHANNEL_MODE_SHIFT )

#define LUT_CFG_CHANNEL_MODE_MASK 0x1
```

Referenced by sippCheckOPipeConnectionLut().

```
#define LUT_CFG_CHANNEL_MODE_SHIFT 0x1
```

Referenced by sippCheckOPipeConnectionLut().

```
#define LUT_CFG_COLOR_CONVERSION_EN( val ) (( val & LUT_CFG_COLOR-
_CONVERSION_EN_MASK ) << LUT_CFG_COLOR_CONVERSION_EN_SHIFT
)

#define LUT_CFG_COLOR_CONVERSION_EN_MASK 0x1

#define LUT_CFG_COLOR_CONVERSION_EN_SHIFT 0x10

#define LUT_CFG_INTEGER_MODE_WIDTH( val ) (( val & LUT_CFG_INTEGE-
R_MODE_WIDTH_MASK ) << LUT_CFG_INTEGER_MODE_WIDTH_SHIFT
)
```

```

#define LUT_CFG_INTEGER_MODE_WIDTH_MASK 0x1F

#define LUT_CFG_INTEGER_MODE_WIDTH_SHIFT 0x3

#define LUT_CFG_INTERPOLATE_FP16( val ) (( val & LUT_CFG_INT-
ERPOLATE_FP16_MASK ) << LUT_CFG_INTERPOLATE_FP16_SHIFT
)

#define LUT_CFG_INTERPOLATE_FP16_MASK 0x1

#define LUT_CFG_INTERPOLATE_FP16_SHIFT 0x0

#define LUT_CFG_LUT_LOAD_EN( val ) (( val & LUT_CFG_LUT_LOAD_EN_MASK ) <<
LUT_CFG_LUT_LOAD_EN_SHIFT )

#define LUT_CFG_LUT_LOAD_EN_MASK 0x1

#define LUT_CFG_LUT_LOAD_EN_SHIFT 0xE

#define LUT_CFG_NUM_CHANNELS( val ) (( val & LUT_CFG_NUM_CHANNELS_MASK )
<< LUT_CFG_NUM_CHANNELS_SHIFT )

#define LUT_CFG_NUM_CHANNELS_MASK 0x3

#define LUT_CFG_NUM_CHANNELS_SHIFT 0xC

#define LUT_CFG_NUM_LUTS( val ) (( val & LUT_CFG_NUM_LUTS_MASK ) <<
LUT_CFG_NUM_LUTS_SHIFT )

#define LUT_CFG_NUM_LUTS_MASK 0xF

#define LUT_CFG_NUM_LUTS_SHIFT 0x8

#define LUT_REGION0_SIZE_INDEX( val ) ((val & LUT_REGION0_SIZE_INDEX_MASK)
<< LUT_REGION0_SIZE_INDEX_SHIFT)

#define LUT_REGION0_SIZE_INDEX_MASK 0xF

#define LUT_REGION0_SIZE_INDEX_SHIFT 0x0

#define LUT_REGION10_SIZE_INDEX( val ) ((val & LUT_REGION10_SIZE_INDEX_MASK)
<< LUT_REGION10_SIZE_INDEX_SHIFT)

#define LUT_REGION10_SIZE_INDEX_MASK 0xF

#define LUT_REGION10_SIZE_INDEX_SHIFT 0x8

#define LUT_REGION11_SIZE_INDEX( val ) ((val & LUT_REGION11_SIZE_INDEX_MASK)
<< LUT_REGION11_SIZE_INDEX_SHIFT)

#define LUT_REGION11_SIZE_INDEX_MASK 0xF

```



```
#define LUT_REGION11_SIZE_INDEX_SHIFT 0xC

#define LUT_REGION12_SIZE_INDEX( val ) ((val & LUT_REGION12_SIZE_INDEX_MASK)
<< LUT_REGION12_SIZE_INDEX_SHIFT)

#define LUT_REGION12_SIZE_INDEX_MASK 0xF

#define LUT_REGION12_SIZE_INDEX_SHIFT 0x10

#define LUT_REGION13_SIZE_INDEX( val ) ((val & LUT_REGION13_SIZE_INDEX_MASK)
<< LUT_REGION13_SIZE_INDEX_SHIFT)

#define LUT_REGION13_SIZE_INDEX_MASK 0xF

#define LUT_REGION13_SIZE_INDEX_SHIFT 0x14

#define LUT_REGION14_SIZE_INDEX( val ) ((val & LUT_REGION14_SIZE_INDEX_MASK)
<< LUT_REGION14_SIZE_INDEX_SHIFT)

#define LUT_REGION14_SIZE_INDEX_MASK 0xF

#define LUT_REGION14_SIZE_INDEX_SHIFT 0x18

#define LUT_REGION15_SIZE_INDEX( val ) ((val & LUT_REGION15_SIZE_INDEX_MASK)
<< LUT_REGION15_SIZE_INDEX_SHIFT)

#define LUT_REGION15_SIZE_INDEX_MASK 0xF

#define LUT_REGION15_SIZE_INDEX_SHIFT 0x1C

#define LUT_REGION1_SIZE_INDEX( val ) ((val & LUT_REGION1_SIZE_INDEX_MASK)
<< LUT_REGION1_SIZE_INDEX_SHIFT)

#define LUT_REGION1_SIZE_INDEX_MASK 0xF

#define LUT_REGION1_SIZE_INDEX_SHIFT 0x4

#define LUT_REGION2_SIZE_INDEX( val ) ((val & LUT_REGION2_SIZE_INDEX_MASK)
<< LUT_REGION2_SIZE_INDEX_SHIFT)

#define LUT_REGION2_SIZE_INDEX_MASK 0xF

#define LUT_REGION2_SIZE_INDEX_SHIFT 0x8

#define LUT_REGION3_SIZE_INDEX( val ) ((val & LUT_REGION3_SIZE_INDEX_MASK)
<< LUT_REGION3_SIZE_INDEX_SHIFT)

#define LUT_REGION3_SIZE_INDEX_MASK 0xF

#define LUT_REGION3_SIZE_INDEX_SHIFT 0xC
```

```
#define LUT_REGION4_SIZE_INDEX( val ) ((val & LUT_REGION4_SIZE_INDEX_MASK)
<< LUT_REGION4_SIZE_INDEX_SHIFT)

#define LUT_REGION4_SIZE_INDEX_MASK 0xF

#define LUT_REGION4_SIZE_INDEX_SHIFT 0x10

#define LUT_REGION5_SIZE_INDEX( val ) ((val & LUT_REGION5_SIZE_INDEX_MASK)
<< LUT_REGION5_SIZE_INDEX_SHIFT)

#define LUT_REGION5_SIZE_INDEX_MASK 0xF

#define LUT_REGION5_SIZE_INDEX_SHIFT 0x14

#define LUT_REGION6_SIZE_INDEX( val ) ((val & LUT_REGION6_SIZE_INDEX_MASK)
<< LUT_REGION6_SIZE_INDEX_SHIFT)

#define LUT_REGION6_SIZE_INDEX_MASK 0xF

#define LUT_REGION6_SIZE_INDEX_SHIFT 0x18

#define LUT_REGION7_SIZE_INDEX( val ) ((val & LUT_REGION7_SIZE_INDEX_MASK)
<< LUT_REGION7_SIZE_INDEX_SHIFT)

#define LUT_REGION7_SIZE_INDEX_MASK 0xF

#define LUT_REGION7_SIZE_INDEX_SHIFT 0x1C

#define LUT_REGION8_SIZE_INDEX( val ) ((val & LUT_REGION8_SIZE_INDEX_MASK)
<< LUT_REGION8_SIZE_INDEX_SHIFT)

#define LUT_REGION8_SIZE_INDEX_MASK 0xF

#define LUT_REGION8_SIZE_INDEX_SHIFT 0x0

#define LUT_REGION9_SIZE_INDEX( val ) ((val & LUT_REGION9_SIZE_INDEX_MASK)
<< LUT_REGION9_SIZE_INDEX_SHIFT)

#define LUT_REGION9_SIZE_INDEX_MASK 0xF

#define LUT_REGION9_SIZE_INDEX_SHIFT 0x4

#define LUT_REGION_0_TO_7_SIZE( szR0, szR1, szR2, szR3, szR4, szR5, szR6, szR7 )
```

Value:

```
LUT_REGION0_SIZE_INDEX (szR0) | \
LUT_REGION1_SIZE_INDEX (szR1) | \
LUT_REGION2_SIZE_INDEX (szR2) | \
LUT_REGION3_SIZE_INDEX (szR3) | \
LUT_REGION4_SIZE_INDEX (szR4) | \
LUT_REGION5_SIZE_INDEX (szR5) | \
LUT_REGION6_SIZE_INDEX (szR6) | \
LUT_REGION7_SIZE_INDEX (szR7)
```

```
#define LUT_REGION_8_TO_15_SIZE( szR8, szR9, szR10, szR11, szR12, szR13, szR14,
szR15 )
```

Value:

```
LUT_REGION8_SIZE_INDEX (szR8) |\
LUT_REGION9_SIZE_INDEX (
szR9) |\
LUT_REGION10_SIZE_INDEX (
szR10) |\
LUT_REGION11_SIZE_INDEX (
szR11) |\
LUT_REGION12_SIZE_INDEX (
szR12) |\
LUT_REGION13_SIZE_INDEX (
szR13) |\
LUT_REGION14_SIZE_INDEX (
szR14) |\
LUT_REGION15_SIZE_INDEX (
szR15)
```

```
#define MED_CFG( ks, outSelect, thresh, alphaBlend, lumaSubSamp, goFast )
```

Value:

```
MED_CFG_KERNEL_SIZE (ks) |\
MED_CFG_OUTPUT_SELECT (outSelect) |\
MED_CFG_THRESHOLD (thresh) |\
MED_CFG_ALPHA_BLEND_EN (alphaBlend) |\
MED_CFG_LUM_SUBSAMP_EN (lumaSubSamp) |\
MED_CFG_GO_FAST (goFast)
```

```
#define MED_CFG_ALPHA_BLEND_EN( val ) ((val & MED_CFG_ALPHA_BLEND_EN_MA-
SK) << MED_CFG_ALPHA_BLEND_EN_SHIFT)
```

```
#define MED_CFG_ALPHA_BLEND_EN_MASK 0x1
```

```
#define MED_CFG_ALPHA_BLEND_EN_SHIFT 0x1C
```

```
#define MED_CFG_GO_FAST( val ) ((val & MED_CFG_GO_FAST_MASK) <<
MED_CFG_GO_FAST_SHIFT)
```

```
#define MED_CFG_GO_FAST_MASK 0x1
```

```
#define MED_CFG_GO_FAST_SHIFT 0x1E
```

```
#define MED_CFG_KERNEL_SIZE( val ) ((val & MED_CFG_KERNEL_SIZE_MASK) <<
MED_CFG_KERNEL_SIZE_SHIFT)
```

```
#define MED_CFG_KERNEL_SIZE_MASK 0x7
```

```
#define MED_CFG_KERNEL_SIZE_SHIFT 0x0
```

```
#define MED_CFG_LUM_SUBSAMP_EN( val ) ((val & MED_CFG_LUM_SUBSAMP_EN_MA-
SK) << MED_CFG_LUM_SUBSAMP_EN_SHIFT)
```

```
#define MED_CFG_LUM_SUBSAMP_EN_MASK 0x1

#define MED_CFG_LUM_SUBSAMP_EN_SHIFT 0x1D

#define MED_CFG_OUTPUT_SELECT( val ) ((val & MED_CFG_OUTPUT_SELECT_MASK)
<< MED_CFG_OUTPUT_SELECT_SHIFT)

#define MED_CFG_OUTPUT_SELECT_MASK 0x3F

#define MED_CFG_OUTPUT_SELECT_SHIFT 0x8

#define MED_CFG_THRESHOLD( val ) ((val & MED_CFG_THRESHOLD_MASK) <<
MED_CFG_THRESHOLD_SHIFT)

#define MED_CFG_THRESHOLD_MASK 0x1FF

#define MED_CFG_THRESHOLD_SHIFT 0x10

#define MED_LUMA_ALPHA( slope, offset )
```

Value:

```
MED_LUMA_ALPHA_SLOPE(slope) |\
MED_LUMA_ALPHA_OFFSET(offset)

#define MED_LUMA_ALPHA_OFFSET( val ) ((val & MED_LUMA_ALPHA_OFFSET_MASK)
<< MED_LUMA_ALPHA_OFFSET_SHIFT)

#define MED_LUMA_ALPHA_OFFSET_MASK 0xFF

#define MED_LUMA_ALPHA_OFFSET_SHIFT 0x0

#define MED_LUMA_ALPHA_SLOPE( val ) ((val & MED_LUMA_ALPHA_SLOPE_MASK)
<< MED_LUMA_ALPHA_SLOPE_SHIFT)

#define MED_LUMA_ALPHA_SLOPE_MASK 0xFF

#define MED_LUMA_ALPHA_SLOPE_SHIFT 0x8

#define MIPI_RX_CFG( inBitDepth, promote, usePrivateLcs, packBuffer, bayerMode, convRs,
convEn )
```

Value:

```
MIPI_RX_CFG_INPUT_BIT_DEPTH(inBitDepth) |\
MIPI_RX_CFG_PROMOTE(promote) |\
MIPI_RX_CFG_USE_PRIVATE_LCS(usePrivateLcs) |\
MIPI_RX_CFG_PACK_BUFFER(packBuffer) |\ MIPI_RX_CFG_BAYER_MODE(bayerMode) |\
MIPI_RX_FMT_CONV_RS(convRs) |\
MIPI_RX_FMT_CONV_EN(convEn)
```

```
#define MIPI_RX_CFG_BAYER_MODE( val ) (( val & MIPI_RX_CFG_BAYER_MODE_MASK)
<< MIPI_RX_CFG_BAYER_MODE_MASK_SHIFT )
```

```
#define MIPI_RX_CFG_BAYER_MODE_MASK 0x1
```

```
#define MIPI_RX_CFG_BAYER_MODE_SHIFT 0x11
```

```
#define MIPI_RX_CFG_INPUT_BIT_DEPTH( val ) (( val & MIPI_RX_CFG_I-
NPUT_BIT_DEPTH_MASK) << MIPI_RX_CFG_INPUT_BIT_DEPTH_SHIFT
)
```

```
#define MIPI_RX_CFG_INPUT_BIT_DEPTH_MASK 0xF
```

```
#define MIPI_RX_CFG_INPUT_BIT_DEPTH_SHIFT 0x1C
```

```
#define MIPI_RX_CFG_OUT_FRM_DIM
```

Value:

```
(width, height) MIPI_RX_OUT_FRM_WIDTH (width) | \
MIPI_RX_OUT_FRM_HEIGHT (height)
```

```
#define MIPI_RX_CFG_PACK_BUFFER( val ) (( val & MIPI_RX_CFG_P-
ACK_BUFFER_MASK) << MIPI_RX_CFG_PACK_BUFFER_MASK_SHIFT
)
```

```
#define MIPI_RX_CFG_PACK_BUFFER_MASK 0x1
```

```
#define MIPI_RX_CFG_PACK_BUFFER_SHIFT 0x13
```

```
#define MIPI_RX_CFG_PROMOTE( val ) (( val & MIPI_RX_CFG_PROMOTE_MASK) <<
MIPI_RX_CFG_PROMOTE_SHIFT )
```

```
#define MIPI_RX_CFG_PROMOTE_MASK 0x1
```

```
#define MIPI_RX_CFG_PROMOTE_SHIFT 0x1A
```

```
#define MIPI_RX_CFG_USE_PRIVATE_LCS( val ) (( val & MIPI_RX_CFG_-
USE_PRIVATE_LCS_MASK) << MIPI_RX_CFG_USE_PRIVATE_LCS_SHIFT
)
```

```
#define MIPI_RX_CFG_USE_PRIVATE_LCS_MASK 0x1
```

```
#define MIPI_RX_CFG_USE_PRIVATE_SHIFT 0x19
```

```
#define MIPI_RX_FMT_CONV_EN( val ) (( val & MIPI_RX_FMT_CONV_EN_MASK) <<
MIPI_RX_FMT_CONV_EN_SHIFT )
```

```
#define MIPI_RX_FMT_CONV_EN_MASK 0xF
```

```
#define MIPI_RX_FMT_CONV_EN_SHIFT 0x8
```

```

#define MIPI_RX_FMT_CONV_RS( val ) (( val & MIPI_RX_FMT_CONV_RS_MASK) <<
MIPI_RX_FMT_CONV_RS_SHIFT)

#define MIPI_RX_FMT_CONV_RS_MASK 0x1F

#define MIPI_RX_FMT_CONV_RS_SHIFT 0xC

#define MIPI_RX_OUT_FRM_HEIGHT( val ) ((MIPI_RX_OUT_FRM_HEIGHT_MASK) <<
MIPI_RX_OUT_FRM_HEIGHT_SHIFT)

#define MIPI_RX_OUT_FRM_HEIGHT_MASK 0xFFFF

#define MIPI_RX_OUT_FRM_HEIGHT_SHIFT 0x10

#define MIPI_RX_OUT_FRM_WIDTH( val ) ((MIPI_RX_OUT_FRM_WIDTH_MASK) <<
MIPI_RX_OUT_FRM_WIDTH_SHIFT)

#define MIPI_RX_OUT_FRM_WIDTH_MASK 0xFFFF

#define MIPI_RX_OUT_FRM_WIDTH_SHIFT 0x0

#define MIPI_RX_WINDOW0_HORIZONTAL_START_CFG( val ) ((
val & MIPI_RX_WINDOW0_HORIZONTAL_START_CFG_MASK) <<
MIPI_RX_WINDOW0_HORIZONTAL_START_CFG_SHIFT)

#define MIPI_RX_WINDOW0_HORIZONTAL_START_CFG_MASK 0xFFFF

#define MIPI_RX_WINDOW0_HORIZONTAL_START_CFG_SHIFT 0x0

#define MIPI_RX_WINDOW0_HORIZONTAL_WIDTH_CFG( val ) ((
val & MIPI_RX_WINDOW0_HORIZONTAL_WIDTH_CFG_MASK) <<
MIPI_RX_WINDOW0_HORIZONTAL_WIDTH_CFG_SHIFT)

#define MIPI_RX_WINDOW0_HORIZONTAL_WIDTH_CFG_MASK 0xFFFF

#define MIPI_RX_WINDOW0_HORIZONTAL_WIDTH_CFG_SHIFT 0x10

#define MIPI_RX_WINDOW0_LEAST_SEGNIFICANT_BIT_MASK 0x1F

#define MIPI_RX_WINDOW0_LEAST_SEGNIFICANT_BIT_SHIFT 0x0

#define MIPI_RX_WINDOW0_SELECTION_ENABLE_MASK 0xF

#define MIPI_RX_WINDOW0_SELECTION_ENABLE_SHIFT 0x8

#define MIPI_RX_WINDOW0_VERTICAL_HEIGHT_CFG_MASK 0xFFFF

#define MIPI_RX_WINDOW0_VERTICAL_HEIGHT_CFG_SHIFT 0x10

```

```

#define MIPI_RX_WINDOW0_VERTICAL_START_CFG( val ) (( val &
MIPI_RX_WINDOW0_VERTICAL_START_CFG_MASK) << MIPI_RX_WINDOW0_VERT-
ICAL_START_CFG_SHIFT )

#define MIPI_RX_WINDOW0_VERTICAL_START_CFG_MASK 0xFFFF

#define MIPI_RX_WINDOW0_VERTICAL_START_CFG_SHIFT 0x0

#define MIPI_RX_WINDOW0_VERTICAL_WIDTH_CFG( val ) (( val & MIPI_RX_WINDOW0_-
VERTICAL_WIDTH_CFG_MASK ) << MIPI_RX_WINDOW0_VERTICAL_WIDTH_CFG_SHIFT
)

#define MIPI_RX_WINDOW1_HORIZONTAL_START_CFG( val ) ((
val & MIPI_RX_WINDOW1_HORIZONTAL_START_CFG_MASK) <<
MIPI_RX_WINDOW1_HORIZONTAL_START_CFG_SHIFT)

#define MIPI_RX_WINDOW1_HORIZONTAL_START_CFG_MASK 0xFFFF

#define MIPI_RX_WINDOW1_HORIZONTAL_START_CFG_SHIFT 0x0

#define MIPI_RX_WINDOW1_HORIZONTAL_WIDTH_CFG( val ) ((
val & MIPI_RX_WINDOW1_HORIZONTAL_WIDTH_CFG_MASK) <<
MIPI_RX_WINDOW1_HORIZONTAL_WIDTH_CFG_SHIFT)

#define MIPI_RX_WINDOW1_HORIZONTAL_WIDTH_CFG_MASK 0xFFFF

#define MIPI_RX_WINDOW1_HORIZONTAL_WIDTH_CFG_SHIFT 0x10

#define MIPI_RX_WINDOW1_LEAST_SIGNIFICANT_BIT_MASK 0x1F

#define MIPI_RX_WINDOW1_LEAST_SIGNIFICANT_BIT_SHIFT 0xF

#define MIPI_RX_WINDOW1_SELECTION_ENABLE_MASK 0xF

#define MIPI_RX_WINDOW1_SELECTION_ENABLE_SHIFT 0x18

#define MIPI_RX_WINDOW1_VERTICAL_HEIGHT_CFG_MASK 0xFFFF

#define MIPI_RX_WINDOW1_VERTICAL_HEIGHT_CFG_SHIFT 0x10

#define MIPI_RX_WINDOW1_VERTICAL_START_CFG( val ) (( val &
MIPI_RX_WINDOW1_VERTICAL_START_CFG_MASK) << MIPI_RX_WINDOW1_VERT-
ICAL_START_CFG_SHIFT )

#define MIPI_RX_WINDOW1_VERTICAL_START_CFG_MASK 0xFFFF

#define MIPI_RX_WINDOW1_VERTICAL_START_CFG_SHIFT 0x0

#define MIPI_RX_WINDOW1_VERTICAL_WIDTH_CFG( val ) (( val & MIPI_RX_WINDOW1_-
VERTICAL_WIDTH_CFG_MASK ) << MIPI_RX_WINDOW1_VERTICAL_WIDTH_CFG_SHIFT
)

```

```

#define MIPI_RX_WINDOW2_HORIZONTAL_START_CFG( val ) ((
val & MIPI_RX_WINDOW2_HORIZONTAL_START_CFG_MASK) <<
MIPI_RX_WINDOW2_HORIZONTAL_START_CFG_SHIFT)

#define MIPI_RX_WINDOW2_HORIZONTAL_START_CFG_MASK 0xFFFF

#define MIPI_RX_WINDOW2_HORIZONTAL_START_CFG_SHIFT 0x0

#define MIPI_RX_WINDOW2_HORIZONTAL_WIDTH_CFG( val ) ((
val & MIPI_RX_WINDOW2_HORIZONTAL_WIDTH_CFG_MASK) <<
MIPI_RX_WINDOW2_HORIZONTAL_WIDTH_CFG_SHIFT)

#define MIPI_RX_WINDOW2_HORIZONTAL_WIDTH_CFG_MASK 0xFFFF

#define MIPI_RX_WINDOW2_HORIZONTAL_WIDTH_CFG_SHIFT 0x10

#define MIPI_RX_WINDOW2_LEAST_SEGNIFICANT_BIT_MASK 0x1F

#define MIPI_RX_WINDOW2_LEAST_SEGNIFICANT_BIT_SHIFT 0x0

#define MIPI_RX_WINDOW2_SELECTION_ENABLE_MASK 0xF

#define MIPI_RX_WINDOW2_SELECTION_ENABLE_SHIFT 0x8

#define MIPI_RX_WINDOW2_VERTICAL_HEIGHT_CFG_MASK 0xFFFF

#define MIPI_RX_WINDOW2_VERTICAL_HEIGHT_CFG_SHIFT 0x10

#define MIPI_RX_WINDOW2_VERTICAL_START_CFG( val ) (( val &
MIPI_RX_WINDOW2_VERTICAL_START_CFG_MASK) << MIPI_RX_WINDOW2_VERT-
ICAL_START_CFG_SHIFT )

#define MIPI_RX_WINDOW2_VERTICAL_START_CFG_MASK 0xFFFF

#define MIPI_RX_WINDOW2_VERTICAL_START_CFG_SHIFT 0x0

#define MIPI_RX_WINDOW2_VERTICAL_WIDTH_CFG( val ) (( val & MIPI_RX_WINDOW2_-
VERTICAL_WIDTH_CFG_MASK ) << MIPI_RX_WINDOW2_VERTICAL_WIDTH_CFG_SHIFT
)

#define MIPI_RX_WINDOW3_HORIZONTAL_START_CFG( val ) ((
val & MIPI_RX_WINDOW3_HORIZONTAL_START_CFG_MASK) <<
MIPI_RX_WINDOW3_HORIZONTAL_START_CFG_SHIFT)

#define MIPI_RX_WINDOW3_HORIZONTAL_START_CFG_MASK 0xFFFF

#define MIPI_RX_WINDOW3_HORIZONTAL_START_CFG_SHIFT 0x0

#define MIPI_RX_WINDOW3_HORIZONTAL_WIDTH_CFG( val ) ((
val & MIPI_RX_WINDOW3_HORIZONTAL_WIDTH_CFG_MASK) <<
MIPI_RX_WINDOW3_HORIZONTAL_WIDTH_CFG_SHIFT)

```



```
#define MIPI_RX_WINDOW3_HORIZONTAL_WIDTH_CFG_MASK 0xFFFF

#define MIPI_RX_WINDOW3_HORIZONTAL_WIDTH_CFG_SHIFT 0x10

#define MIPI_RX_WINDOW3_LEAST_SIGNIFICANT_BIT_MASK 0x1F

#define MIPI_RX_WINDOW3_LEAST_SIGNIFICANT_BIT_SHIFT 0xF

#define MIPI_RX_WINDOW3_SELECTION_ENABLE_MASK 0xF

#define MIPI_RX_WINDOW3_SELECTION_ENABLE_SHIFT 0x18

#define MIPI_RX_WINDOW3_VERTICAL_HEIGHT_CFG_MASK 0xFFFF

#define MIPI_RX_WINDOW3_VERTICAL_HEIGHT_CFG_SHIFT 0x10

#define MIPI_RX_WINDOW3_VERTICAL_START_CFG( val ) (( val &
MIPI_RX_WINDOW3_VERTICAL_START_CFG_MASK) << MIPI_RX_WINDOW3_VERT-
ICAL_START_CFG_SHIFT )

#define MIPI_RX_WINDOW3_VERTICAL_START_CFG_MASK 0xFFFF

#define MIPI_RX_WINDOW3_VERTICAL_START_CFG_SHIFT 0x0

#define MIPI_RX_WINDOW3_VERTICAL_WIDTH_CFG( val ) (( val & MIPI_RX_WINDOW3_-
VERTICAL_WIDTH_CFG_MASK ) << MIPI_RX_WINDOW3_VERTICAL_WIDTH_CFG_SHIFT
)

#define MIPI_TX_CFG_BACKPORCH( val ) (( val & MIPI_TX_CFG_BACKPORCH_MASK )
<< MIPI_TX_CFG_BACKPORCH+_SHIFT )

#define MIPI_TX_CFG_BACKPORCH_MASK 0x1

#define MIPI_TX_CFG_BACKPORCH_SHIFT 0x8

#define MIPI_TX_CFG_FRONTPORCH( val ) (( val & MIPI_TX_CFG_FRONTPORCH_MASK)
<< MIPI_TX_CFG_FRONTPORCH_SHIFT )

#define MIPI_TX_CFG_FRONTPORCH_MASK 0x1

#define MIPI_TX_CFG_FRONTPORCH_SHIFT 0x9

#define MIPI_TX_FIRST_FIELD( val ) (( val & MIPI_TX_FIRST_FIELD_MASK ) <<
MIPI_TX_FIRST_FIELD_SHIFT )

#define MIPI_TX_FIRST_FIELD_MASK 0x1

#define MIPI_TX_FIRST_FIELD_SHIFT 0x1

#define MIPI_TX_FRM_CFG
```

Value:

```
(width, height) MIPI_TX_FRM_HEIGHT (height) | \
MIPI_TX_FRM_WIDTH (width)

#define MIPI_TX_FRM_HEIGHT( val ) (( val & MIPI_TX_FRM_HEIGHT_MASK) <<
MIPI_TX_FRM_HEIGHT_SHIFT )

#define MIPI_TX_FRM_HEIGHT_MASK 0xFFFF

#define MIPI_TX_FRM_HEIGHT_SHIFT 0x10

#define MIPI_TX_FRM_WIDTH( val ) (( val & MIPI_TX_FRM_WIDTH_MASK) <<
MIPI_TX_FRM_WIDTH_SHIFT )

#define MIPI_TX_FRM_WIDTH_MASK 0xFFFF

#define MIPI_TX_FRM_WIDTH_SHIFT 0x0

#define MIPI_TX_INT_SEL( val ) (( val & MIPI_TX_INT_SEL_MASK ) <<
MIPI_TX_INT_SEL_SHIFT )

#define MIPI_TX_INT_SEL_MASK 0x3

#define MIPI_TX_INT_SEL_SHIFT 0x6

#define MIPI_TX_ONESHOT( val ) (( val & MIPI_TX_ONESHOT_MASK ) <<
MIPI_TX_ONESHOT_SHIFT )

#define MIPI_TX_ONESHOT_MODE_MASK 0x1

#define MIPI_TX_ONESHOT_MODE_SHIFT 0x2

#define MIPI_TX_SCAN_MODE( val ) (( val & MIPI_TX_SCAN_MODE_MASK ) <<
MIPI_TX_SCAN_MODE_SHIFT )

#define MIPI_TX_SCAN_MODE_MASK 0x1

#define MIPI_TX_SCAN_MODE_SHIFT 0x0

#define MIPI_TX_SYNC_IDLE( val ) (( val & MIPI_TX_SYNC_IDLE_MASK ) <<
MIPI_TX_SYNC_IDLE_SHIFT )

#define MIPI_TX_SYNC_IDLE_MASK 0x1

#define MIPI_TX_SYNC_IDLE_SHIFT 0x3

#define OUTPLANE_ORDER_BGR DEBAYER_OUTPLANE_ORDER(0x1)

#define OUTPLANE_ORDER_BRG DEBAYER_OUTPLANE_ORDER(0x3)

#define OUTPLANE_ORDER_GBR DEBAYER_OUTPLANE_ORDER(0x5)
```

```
#define OUTPLANE_ORDER_GRB DEBAYER_OUTPLANE_ORDER(0x4)
```

```
#define OUTPLANE_ORDER_RGB DEBAYER_OUTPLANE_ORDER(0x2)
```

```
#define OUTPLANE_ORDER_RGB DEBAYER_OUTPLANE_ORDER(0x0)
```

```
#define POLY_CFG
```

Value:

```
(vscaleFactorNum, vscaleFactorDnom, hscaleFactorNum, hscaleFactorDnom, outClamp, kernelSize)
    POLY_CFG_UPFIRDN_VSCALE_FACTOR_NUM (vscaleFactorNum) |\
        POLY_CFG_UPFIRDN_VSCALE_FACTOR_DENOM (vscaleFactorDnom) |\
        POLY_CFG_UPFIRDN_HSCALE_FACTOR_NUM (hscaleFactorNum) |\
        POLY_CFG_UPFIRDN_HSCALE_FACTOR_DENOM (hscaleFactorDnom) |\
        POLY_CFG_UPFIRDN_OUTPUT_CLAMP (outClamp) |\
        POLY_CFG_KERNEL_SIZE (kernelSize)
```

```
#define POLY_CFG_DIM_IN
```

Value:

```
(width, height) POLY_FRM_HEIGHT_IN (height) |\
    POLY_FRM_WIDTH_IN (width)
```

```
#define POLY_CFG_DIM_OUT
```

Value:

```
(width, height) POLY_FRM_HEIGHT_OUT (height) |\
    POLY_FRM_WIDTH_OUT (width)
```

```
#define POLY_CFG_HORIZONTAL_COEFFS_0_TO_3
```

Value:

```
(val0, val1, val2, val3) POLY_HORIZONTAL_COEFF0 (val0) |\
    POLY_HORIZONTAL_COEFF1 (val1) |\
    POLY_HORIZONTAL_COEFF2 (val2) |\
    POLY_HORIZONTAL_COEFF3 (val3)
```

```
#define POLY_CFG_HORIZONTAL_COEFFS_4_TO_6
```

Value:

```
(val4, val5, val6) POLY_VERTICAL_COEFF4 (val4) |\
    POLY_HORIZONTAL_COEFF5 (val5) |\
    POLY_HORIZONTAL_COEFF6 (val6)
```

```

#define POLY_CFG_KERNEL_SIZE( val ) (( val & POLY_CFG_KERNEL_SIZE_MASK ) <<
POLY_CFG_KERNEL_SIZE_SHIFT )

#define POLY_CFG_KERNEL_SIZE_MASK 0x7

#define POLY_CFG_KERNEL_SIZE_SHIFT 0x0

#define POLY_CFG_UPFIRDN_HSCALE_FACTOR_DENOM( val ) (( val
& POLY_CFG_UPFIRDN_HSCALE_FACTOR_DENOM_MASK ) <<
POLY_CFG_UPFIRDN_HSCALE_FACTOR_DENOM_SHIFT )

#define POLY_CFG_UPFIRDN_HSCALE_FACTOR_DENOM_MASK 0x3F

#define POLY_CFG_UPFIRDN_HSCALE_FACTOR_DENOM_SHIFT 0x4

#define POLY_CFG_UPFIRDN_HSCALE_FACTOR_NUM( val ) (( val &
POLY_CFG_UPFIRDN_HSCALE_FACTOR_NUM_MASK ) << POLY_CFG_UPFIRDN_HS-
CALE_FACTOR_NUM_SHIFT )

#define POLY_CFG_UPFIRDN_HSCALE_FACTOR_NUM_MASK 0x7FFF

#define POLY_CFG_UPFIRDN_HSCALE_FACTOR_NUM_SHIFT 0xA

#define POLY_CFG_UPFIRDN_OUTPUT_CLAMP( val ) (( val & POLY_CFG_UPFIRD-
N_OUTPUT_CLAMP_MASK ) << POLY_CFG_UPFIRDN_OUTPUT_CLAMP_SHIFT
)

#define POLY_CFG_UPFIRDN_OUTPUT_CLAMP_MASK 0x1

#define POLY_CFG_UPFIRDN_OUTPUT_CLAMP_SHIFT 0x3

#define POLY_CFG_UPFIRDN_VSCALE_FACTOR_DENOM( val ) ((val
& POLY_CFG_UPFIRDN_VSCALE_FACTOR_DENOM_MASK ) <<
POLY_CFG_UPFIRDN_VSCALE_FACTOR_DENOM_SHIFT )

#define POLY_CFG_UPFIRDN_VSCALE_FACTOR_DENOM_MASK 0x3F

#define POLY_CFG_UPFIRDN_VSCALE_FACTOR_DENOM_SHIFT 0x10

#define POLY_CFG_UPFIRDN_VSCALE_FACTOR_NUM( val ) ((val &
POLY_CFG_UPFIRDN_VSCALE_FACTOR_NUM_MASK ) << POLY_CFG_UPFIRDN_VS-
CALE_FACTOR_NUM_SHIFT )

#define POLY_CFG_UPFIRDN_VSCALE_FACTOR_NUM_MASK 0x1F

#define POLY_CFG_UPFIRDN_VSCALE_FACTOR_NUM_SHIFT 0x16

#define POLY_CFG_VERTICAL_COEFFS_0_TO_3

```

Value:

```

(val0, val1, val2, val3) POLY_VERTICAL_COEFF0 (val0) | \
POLY_VERTICAL_COEFF1 (val1) | \

```

```
POLY_VERTICAL_COEFF2 (val2) |\
POLY_VERTICAL_COEFF3 (val3)
```

```
#define POLY_CFG_VERTICAL_COEFFS_4_TO_6
```

Value:

```
(val4, val5, val6) POLY_VERTICAL_COEFF4 (val4) |\
POLY_VERTICAL_COEFF5 (val5) |\
POLY_VERTICAL_COEFF6 (val6)
```

```
#define POLY_FRM_HEIGHT_IN( val ) (( val & POLY_FRM_HEIGHT_IN_MASK ) <<
POLY_FRM_HEIGHT_IN_SHIFT)
```

```
#define POLY_FRM_HEIGHT_IN_MASK 0xFFFF
```

```
#define POLY_FRM_HEIGHT_IN_SHIFT 0x10
```

```
#define POLY_FRM_HEIGHT_OUT( val ) (( val & POLY_FRM_HEIGHT_OUT_MASK ) <<
POLY_FRM_HEIGHT_OUT_SHIFT)
```

```
#define POLY_FRM_HEIGHT_OUT_MASK 0xFFFF
```

```
#define POLY_FRM_HEIGHT_OUT_SHIFT 0x10
```

```
#define POLY_FRM_WIDTH_IN( val ) (( val & POLY_FRM_WIDTH_IN_MASK ) <<
POLY_FRM_WIDTH_IN_SHIFT)
```

```
#define POLY_FRM_WIDTH_IN_MASK 0xFFFF
```

```
#define POLY_FRM_WIDTH_IN_SHIFT 0x0
```

```
#define POLY_FRM_WIDTH_OUT( val ) (( val & POLY_FRM_WIDTH_OUT_MASK ) <<
POLY_FRM_WIDTH_OUT_SHIFT)
```

```
#define POLY_FRM_WIDTH_OUT_MASK 0xFFFF
```

```
#define POLY_FRM_WIDTH_OUT_SHIFT 0x0
```

```
#define POLY_HORIZONTAL_COEFF0 ( val & POLY_HORIZONTAL_COEFF0_MASK ) <<
POLY_HORIZONTAL_COEFF0_SHIFT)
```

```
#define POLY_HORIZONTAL_COEFF0_MASK 0xFF
```

```
#define POLY_HORIZONTAL_COEFF0_SHIFT 0x0
```

```
#define POLY_HORIZONTAL_COEFF1 ( val & POLY_HORIZONTAL_COEFF1_MASK ) <<
POLY_HORIZONTAL_COEFF1_SHIFT)
```

```
#define POLY_HORIZONTAL_COEFF1_MASK 0xFF
```

```
#define POLY_HORIZONTAL_COEFF1_SHIFT 0x8

#define POLY_HORIZONTAL_COEFF2 ( val & POLY_HORIZONTAL_COEFF2_MASK) <<
POLY_HORIZONTAL_COEFF2_SHIFT)

#define POLY_HORIZONTAL_COEFF2_MASK 0xFF

#define POLY_HORIZONTAL_COEFF2_SHIFT 0x10

#define POLY_HORIZONTAL_COEFF3 ( val & POLY_HORIZONTAL_COEFF3_MASK) <<
POLY_HORIZONTAL_COEFF3_SHIFT)

#define POLY_HORIZONTAL_COEFF3_MASK 0xFF

#define POLY_HORIZONTAL_COEFF3_SHIFT 0x18

#define POLY_HORIZONTAL_COEFF4_MASK 0xFF

#define POLY_HORIZONTAL_COEFF4_SHIFT 0x0

#define POLY_HORIZONTAL_COEFF5_MASK 0xFF

#define POLY_HORIZONTAL_COEFF5_SHIFT 0x10

#define POLY_HORIZONTAL_COEFF6_MASK 0xFF

#define POLY_HORIZONTAL_COEFF6_SHIFT 0x10

#define POLY_VERTICAL_COEFF0 ( val & POLY_VERTICAL_COEFF0_MASK) <<
POLY_VERTICAL_COEFF0_SHIFT)

#define POLY_VERTICAL_COEFF0_MASK 0xFF

#define POLY_VERTICAL_COEFF0_SHIFT 0x0

#define POLY_VERTICAL_COEFF1 ( val & POLY_VERTICAL_COEFF1_MASK) <<
POLY_VERTICAL_COEFF1_SHIFT)

#define POLY_VERTICAL_COEFF1_MASK 0xFF

#define POLY_VERTICAL_COEFF1_SHIFT 0x8

#define POLY_VERTICAL_COEFF2 ( val & POLY_VERTICAL_COEFF2_MASK) <<
POLY_VERTICAL_COEFF2_SHIFT)

#define POLY_VERTICAL_COEFF2_MASK 0xFF

#define POLY_VERTICAL_COEFF2_SHIFT 0x10

#define POLY_VERTICAL_COEFF3 ( val & POLY_VERTICAL_COEFF3_MASK) <<
POLY_VERTICAL_COEFF3_SHIFT)
```

```
#define POLY_VERTICAL_COEFF3_MASK 0xFF

#define POLY_VERTICAL_COEFF3_SHIFT 0x18

#define POLY_VERTICAL_COEFF4_MASK 0xFF

#define POLY_VERTICAL_COEFF4_SHIFT 0x0

#define POLY_VERTICAL_COEFF5_MASK 0xFF

#define POLY_VERTICAL_COEFF5_SHIFT 0x10

#define POLY_VERTICAL_COEFF6_MASK 0xFF

#define POLY_VERTICAL_COEFF6_SHIFT 0x10

#define RAW_BAD_PIX_CFG( noiseLevel, alphaGHot, alphaGCold, alphaRbHot, alphaRbCold
)
```

Value:

```
RAW_BAD_PIXEL_CFG_NOISE_LEVEL(noiseLevel) |\
    RAW_BAD_PIXEL_CFG_ALPHA_G_HOT(alphaGHot) |\
    RAW_BAD_PIXEL_CFG_ALPHA_G_COLD(alphaGCold) |\
    RAW_BAD_PIXEL_CFG_ALPHA_RB_HOT(alphaRbHot) |\
    RAW_BAD_PIXEL_CFG_ALPHA_RB_COLD(alphaRbCold)

#define RAW_BAD_PIXEL_CFG_ALPHA_G_HOT( val ) (( val & RAW_BAD_PIXEL_CFG_
FG_ALPHA_G_HOT_MASK) << RAW_BAD_PIXEL_CFG_ALPHA_G_HOT_SHIFT
)

#define RAW_BAD_PIXEL_CFG_ALPHA_G_HOT_MASK 0xF

#define RAW_BAD_PIXEL_CFG_ALPHA_G_HOT_SHIFT 0xC

#define RAW_BAD_PIXEL_CFG_ALPHA_RB_COLD( val ) (( val & RAW_BAD_PIXEL_CFG_
ALPHA_RB_COLD_MASK) << RAW_BAD_PIXEL_CFG_ALPHA_RB_COLD_SHIFT)

#define RAW_BAD_PIXEL_CFG_ALPHA_RB_COLD_MASK 0xF

#define RAW_BAD_PIXEL_CFG_ALPHA_RB_COLD_SHIFT 0x0

#define RAW_BAD_PIXEL_CFG_ALPHA_RB_HOT( val ) (( val & RAW_BAD_PIXEL_CFG_
ALPHA_RB_HOT_MASK) << RAW_BAD_PIXEL_CFG_ALPHA_RB_HOT_SHIFT)

#define RAW_BAD_PIXEL_CFG_ALPHA_RB_HOT_MASK 0xF

#define RAW_BAD_PIXEL_CFG_ALPHA_RB_HOT_SHIFT 0x4
```

```
#define RAW_BAD_PIXEL_CFG_APLHA_G_COLD( val ) (( val & RAW_BAD_PIXEL_CFG_-
APLHA_G_COLD_MASK) << RAW_BAD_PIXEL_CFG_APLHA_G_COLD_SHIFT)

#define RAW_BAD_PIXEL_CFG_APLHA_G_COLD_MASK 0xF

#define RAW_BAD_PIXEL_CFG_APLHA_G_COLD_SHIFT 0x8

#define RAW_BAD_PIXEL_CFG_NOISE_LEVEL( val ) (( val & RAW_BAD_PIXEL_-
CFG_NOISE_LEVEL_MASK) << RAW_BAD_PIXEL_CFG_NOISE_LEVEL_SHIFT
)

#define RAW_BAD_PIXEL_CFG_NOISE_LEVEL_MASK 0xFFFF

#define RAW_BAD_PIXEL_CFG_NOISE_LEVEL_SHIFT 0x10

#define RAW_CFG( format, bayerPatern, grgbImbEn, greenFixOnly, awbStatsEn, rgbHistEn,
dataWidth, badPThresh, sdcEn, afStatsEn, gainMode, lumaHist, badPixelFixEn )
```

Value:

<pre>RAW_CFG_FORMAT(format) \ bayerPatern) \ \ greenFixOnly) \ awbStatsEn) \ \ \ \ \ \ badPixelFixEn)</pre>	<pre>RAW_CFG_BAYER_PATTERN (RAW_CFG_GRGB_IMB_EN (grgbImbEn) RAW_CFG_GREEN_FIX_ONLY (RAW_CFG_AWB_STATS_EN (RAW_CFG_RGB_HIST_EN (rgbHistEn) RAW_CFG_DATA_WIDTH (dataWidth) RAW_CFG_BADP_THRES (badPThresh) RAW_CFG_SDC_EN(sdcEn) \ RAW_CFG_AF_STATS_EN (afStatsEn) RAW_CFG_GAIN_MODE (gainMode) \ RAW_CFG_LUMA_HIST_EN (lumaHist) RAW_CFG_BAD_PIXEL_FIX_EN (</pre>
---	---

```
#define RAW_CFG_AF_STATS_EN( val ) (( val & RAW_CFG_AF_STATS_EN_MASK) <<
RAW_CFG_AF_STATS_EN_SHIFT)

#define RAW_CFG_AF_STATS_EN_MASK 0x1

#define RAW_CFG_AF_STATS_EN_SHIFT 0xD

#define RAW_CFG_AWB_STATS_EN( val ) (( val & RAW_CFG_AWB_STATS_EN_MASK)
<< RAW_CFG_AWB_STATS_EN_SHIFT)

#define RAW_CFG_AWB_STATS_EN_MASK 0x1

#define RAW_CFG_AWB_STATS_EN_SHIFT 0x6
```



```

#define RAW_CFG_BAD_PIXEL_FIX_EN( val ) (( val & RAW_CFG_BAD_PIXEL_FIX_EN-  

MASK) << RAW_CFG_BAD_PIXEL_FIX_EN_SHIFT)

#define RAW_CFG_BAD_PIXEL_FIX_EN_MASK 0x1

#define RAW_CFG_BAD_PIXEL_FIX_EN_SHIFT 0x4

#define RAW_CFG_BADP_THRES( val ) (( val & RAW_CFG_BADP_THRES_MASK) <<  

RAW_CFG_BADP_THRES_SHIFT)

#define RAW_CFG_BADP_THRES_MASK 0xFF

#define RAW_CFG_BADP_THRES_SHIFT 0x10

#define RAW_CFG_BAYER_PATTERN( val ) (( val & RAW_CFG_BAYER_PATTERN_MASK)  

<< RAW_CFG_BAYER_PATTERN_SHIFT)

#define RAW_CFG_BAYER_PATTERN_MASK 0x3

#define RAW_CFG_BAYER_PATTERN_SHIFT 0x1

#define RAW_CFG_DATA_WIDTH( val ) (( val & RAW_CFG_DATA_WIDTH_MASK) <<  

RAW_CFG_DATA_WIDTH_SHIFT)

#define RAW_CFG_DATA_WIDTH_MASK 0xF

#define RAW_CFG_DATA_WIDTH_SHIFT 0x8

#define RAW_CFG_FORMAT( val ) (( val & RAW_CFG_FORMAT_MASK) <<  

RAW_CFG_FORMAT_SHIFT)

#define RAW_CFG_FORMAT_MASK 0x1

#define RAW_CFG_FORMAT_SHIFT 0x0

#define RAW_CFG_GAIN_MODE( val ) (( val & RAW_CFG_GAIN_MODE_MASK) <<  

RAW_CFG_GAIN_MODE_SHIFT)

#define RAW_CFG_GAIN_MODE_MASK 0x1

#define RAW_CFG_GAIN_MODE_SHIFT 0xC

#define RAW_CFG_GREEN_FIX_ONLY( val ) (( val & RAW_CFG_GREEN_FIX_ONLY_MA-  

SK) << RAW_CFG_GREEN_FIX_ONLY_SHIFT)

#define RAW_CFG_GREEN_FIX_ONLY_MASK 0x1

#define RAW_CFG_GREEN_FIX_ONLY_SHIFT 0x5

#define RAW_CFG_GRGB_IMB_EN( val ) (( val & RAW_CFG_GRGB_IMB_EN_MASK) <<  

RAW_CFG_GRGB_IMB_EN_SHIFT)

```

```
#define RAW_CFG_GRGB_IMB_EN_MASK 0x1

#define RAW_CFG_GRGB_IMB_EN_SHIFT 0x3

#define RAW_CFG_LUMA_HIST_EN( val ) (( val & RAW_CFG_LUMA_HIST_EN_MASK) << RAW_CFG_LUMA_HIST_EN_SHIFT)

#define RAW_CFG_LUMA_HIST_EN_MASK 0x1

#define RAW_CFG_LUMA_HIST_EN_SHIFT 0x7

#define RAW_CFG_RGB_HIST_EN( val ) (( val & RAW_CFG_RGB_HIST_EN_MASK) << RAW_CFG_RGB_HIST_EN_SHIFT)

#define RAW_CFG_RGB_HIST_EN_MASK 0x1

#define RAW_CFG_RGB_HIST_EN_SHIFT 0x18

#define RAW_CFG_SDC_EN( val ) (( val & RAW_CFG_SDC_EN_MASK) << RAW_CFG_SDC_EN_SHIFT)

#define RAW_CFG_SDC_EN_MASK 0x1

#define RAW_CFG_SDC_EN_SHIFT 0x1B

#define RAW_GRGB_DECAY( slopeBright, slopeDark )
```

Value:

```
RAW_GRGB_SLOPE_BRIGHT (slopeBright) + \
    RAW_GRGB_SLOPE_DARK (slopeDark)
```

```
#define RAW_GRGB_PLATO( dark, bright )
```

Value:

```
RAW_GRGB_PLATO_DARK(dark) + \
    RAW_GRGB_PLATO_BRIGHT(bright)
```

```
#define RAW_GRGB_PLATO_BRIGHT( val ) ((val & RAW_GRGB_PLATO_BRIGHT_MASK) << RAW_GRGB_PLATO_BRIGHT_SHIFT)
```

```
#define RAW_GRGB_PLATO_BRIGHT_MASK 0x3FFF
```

```
#define RAW_GRGB_PLATO_BRIGHT_SHIFT 0x10
```

```
#define RAW_GRGB_PLATO_DARK( val ) ((val & RAW_GRGB_PLATO_DARK_MASK) << RAW_GRGB_PLATO_DARK_SHIFT)
```

```
#define RAW_GRGB_PLATO_DARK_MASK 0x3FFF
```

```
#define RAW_GRGB_PLATO_DARK_SHIFT 0x0

#define RAW_GRGB_SLOPE_BRIGHT( val ) ((val & RAW_GRGB_SLOPE_BRIGHT_MASK)
<< RAW_GRGB_SLOPE_BRIGHT_SHIFT )

#define RAW_GRGB_SLOPE_BRIGHT_MASK 0x3FFF

#define RAW_GRGB_SLOPE_BRIGHT_SHIFT 0x10

#define RAW_GRGB_SLOPE_DARK( val ) (( val & RAW_GRGB_SLOPE_DARK_MASK) <<
RAW_GRGB_SLOPE_DARK_SHIFT )

#define RAW_GRGB_SLOPE_DARK_MASK 0x3FFF

#define RAW_GRGB_SLOPE_DARK_SHIFT 0x0

#define RAW_PATCH_CFG( numHorizPatches, numVertPatches, patchWidth, patchHeight )
```

Value:

```
RAW_PATCH_CFG_NUM_HORIZ_PATCHES((numHorizPatches ? numHorizPatches - 0x1 :
0x0)) +\
numVertPatches - 0x1 : 0x0)) +\
: 0x0)) +\
x1 : 0x0))
RAW_PATCH_CFG_NUM_VERT_PATCHES((numVertPatches ?
RAW_PATCH_CFG_PATCH_WIDTH((patchWidth ? patchWidth - 0x1
RAW_PATCH_CFG_PATCH_HEIGHT((patchHeight ? patchHeight - 0
```

```
#define RAW_PATCH_CFG_NUM_HORIZ_PATCHES( val ) ((val & RAW_PATCH_CFG_NUM-
M_HORIZ_PATCHES_MASK) << RAW_PATCH_CFG_NUM_HORIZ_PATCHES_SHIFT)

#define RAW_PATCH_CFG_NUM_HORIZ_PATCHES_MASK 0x3F

#define RAW_PATCH_CFG_NUM_HORIZ_PATCHES_SHIFT 0x0

#define RAW_PATCH_CFG_NUM_VERT_PATCHES( val ) ((val & RAW_PATCH_CFG_NUM-
_VERT_PATCHES_MASK) << RAW_PATCH_CFG_NUM_VERT_PATCHES_SHIFT)

#define RAW_PATCH_CFG_NUM_VERT_PATCHES_MASK 0x3F

#define RAW_PATCH_CFG_NUM_VERT_PATCHES_SHIFT 0x8

#define RAW_PATCH_CFG_PATCH_HEIGHT( val ) ((val & RAW_PATCH_CFG_PATCH_HE-
IGHT_MASK) << RAW_PATCH_CFG_PATCH_HEIGHT_SHIFT)

#define RAW_PATCH_CFG_PATCH_HEIGHT_MASK 0xFF

#define RAW_PATCH_CFG_PATCH_HEIGHT_SHIFT 0x18

#define RAW_PATCH_CFG_PATCH_WIDTH( val ) ((val & RAW_PATCH_CFG_PATCH_WI-
DTH_MASK) << RAW_PATCH_CFG_PATCH_WIDTH_SHIFT)
```

```
#define RAW_PATCH_CFG_PATCH_WIDTH_MASK 0xFF
```

```
#define RAW_PATCH_CFG_PATCH_WIDTH_SHIFT 0x10
```

```
#define RAW_PATCH_START( xCoord, yCoord )
```

Value:

```
RAW_PATCH_START_X_POS(xCoord) + \
    RAW_PATCH_START_Y_POS(yCoord)
```

```
#define RAW_PATCH_START_X_POS( val ) ((val & RAW_PATCH_START_X_POS_MASK)
<< RAW_PATCH_START_X_POS_SHIFT)
```

```
#define RAW_PATCH_START_X_POS_MASK 0xFFFF
```

```
#define RAW_PATCH_START_X_POS_SHIFT 0x0
```

```
#define RAW_PATCH_START_Y_POS( val ) ((val & RAW_PATCH_START_Y_POS_MASK)
<< RAW_PATCH_START_Y_POS_SHIFT)
```

```
#define RAW_PATCH_START_Y_POS_MASK 0xFFFF
```

```
#define RAW_PATCH_START_Y_POS_SHIFT 0x10
```

```
#define RAW_STATS_PLANES( plane0, plane1, plane2, plane3, histPlane, activePlanes )
```

Value:

```
RAW_STATS_PLANES_PLANE0(plane0) + \
    plane1) + \
    plane2) + \
    plane3) + \
    (histPlane) + \
    RAW_STATS_PLANES_ACTIVE_PLANES(activePlanes)
```

```
#define RAW_STATS_PLANES_ACTIVE_PLANES( val ) (((val ? val-1 : val) & RAW_STATS_P-
LANES_ACTIVE_PLANES_MASK) << RAW_STATS_PLANES_ACTIVE_PLANES_SHIFT)
```

```
#define RAW_STATS_PLANES_ACTIVE_PLANES_MASK 0x3
```

```
#define RAW_STATS_PLANES_ACTIVE_PLANES_SHIFT 0x14
```

```
#define RAW_STATS_PLANES_HGRAM_PLANE( val ) ((val & RAW_STATS_PLANES_HGR-
AM_PLANE_MASK) << RAW_STATS_PLANES_HGRAM_PLANE_SHIFT)
```

```
#define RAW_STATS_PLANES_HGRAM_PLANE_MASK 0xF
```

```
#define RAW_STATS_PLANES_HGRAM_PLANE_SHIFT 0x10

#define RAW_STATS_PLANES_PLANE0( val ) ((val & RAW_STATS_PLANES_PLANE0_MA-
SK) << RAW_STATS_PLANES_PLANE0_SHIFT)

#define RAW_STATS_PLANES_PLANE0_MASK 0xF

#define RAW_STATS_PLANES_PLANE0_SHIFT 0x0

#define RAW_STATS_PLANES_PLANE1( val ) ((val & RAW_STATS_PLANES_PLANE1_MA-
SK) << RAW_STATS_PLANES_PLANE1_SHIFT)

#define RAW_STATS_PLANES_PLANE1_MASK 0xF

#define RAW_STATS_PLANES_PLANE1_SHIFT 0x4

#define RAW_STATS_PLANES_PLANE2( val ) ((val & RAW_STATS_PLANES_PLANE2_MA-
SK) << RAW_STATS_PLANES_PLANE2_SHIFT)

#define RAW_STATS_PLANES_PLANE2_MASK 0xF

#define RAW_STATS_PLANES_PLANE2_SHIFT 0x8

#define RAW_STATS_PLANES_PLANE3( val ) ((val & RAW_STATS_PLANES_PLANE3_MA-
SK) << RAW_STATS_PLANES_PLANE3_SHIFT)

#define RAW_STATS_PLANES_PLANE3_MASK 0xF

#define RAW_STATS_PLANES_PLANE3_SHIFT 0xC

#define SHARPEN_CFG( thresh, ksz, clamp, mode, opDeltas )
```

Value:

```
SHARPEN_CFG_MIN_THRESHOLD (thresh) |\
SHARPEN_CFG_KERNEL_SIZE (ksz) |\
SHARPEN_CFG_CLAMP (clamp) |\
SHARPEN_CFG_MODE (mode) |\
SHARPEN_CFG_OPUT_DELTAS_ONLY (opDeltas)
```

```
#define SHARPEN_CFG_CLAMP( val ) ((val & SHARPEN_CFG_CLAMP_MASK) <<
SHARPEN_CFG_CLAMP_SHIFT)

#define SHARPEN_CFG_CLAMP_MASK 0x1

#define SHARPEN_CFG_CLAMP_SHIFT 0x3

#define SHARPEN_CFG_KERNEL_SIZE( val ) ((val & SHARPEN_CFG_KERNEL_SIZE_MA-
SK) << SHARPEN_CFG_KERNEL_SIZE_SHIFT)

#define SHARPEN_CFG_KERNEL_SIZE_MASK 0x7
```

```
#define SHARPEN_CFG_KERNEL_SIZE_SHIFT 0x0

#define SHARPEN_CFG_MIN_THRESHOLD( val ) ((val & SHARPEN_CFG_MIN_THRESHOLD_MASK) << SHARPEN_CFG_MIN_THRESHOLD_SHIFT)

#define SHARPEN_CFG_MIN_THRESHOLD_MASK 0xFFFF

#define SHARPEN_CFG_MIN_THRESHOLD_SHIFT 0x10

#define SHARPEN_CFG_MODE( val ) ((val & SHARPEN_CFG_MODE_MASK) << SHARPEN_CFG_MODE_SHIFT)

#define SHARPEN_CFG_MODE_MASK 0x1

#define SHARPEN_CFG_MODE_SHIFT 0x4

#define SHARPEN_CFG_OPUT_DELTAS_ONLY( val ) ((val & SHARPEN_CFG_OPUT_DELTAS_ONLY_MASK) << SHARPEN_CFG_OPUT_DELTAS_ONLY_SHIFT)

#define SHARPEN_CFG_OPUT_DELTAS_ONLY_MASK 0x1

#define SHARPEN_CFG_OPUT_DELTAS_ONLY_SHIFT 0x5

#define SHARPEN_COEF0_CFG( val ) ((val & SHARPEN_COEF0_MASK) << SHARPEN_COEF0_SHIFT)

#define SHARPEN_COEF0_MASK 0xFFFF

#define SHARPEN_COEF0_SHIFT 0x0

#define SHARPEN_COEF1_CFG( val ) ((val & SHARPEN_COEF1_MASK) << SHARPEN_COEF1_SHIFT)

#define SHARPEN_COEF1_MASK 0xFFFF

#define SHARPEN_COEF1_SHIFT 0x10

#define SHARPEN_COEF2_CFG( val ) ((val & SHARPEN_COEF2_MASK) << SHARPEN_COEF2_SHIFT)

#define SHARPEN_COEF2_MASK 0xFFFF

#define SHARPEN_COEF2_SHIFT 0x0

#define SHARPEN_COEF3_CFG( val ) ((val & SHARPEN_COEF3_MASK) << SHARPEN_COEF3_SHIFT)

#define SHARPEN_COEF3_MASK 0xFFFF

#define SHARPEN_COEF3_SHIFT 0x10
```

```
#define SHARPEN_COEFS01_CFG( coef0, coef1 )
```

Value:

```
SHARPEN_COEF0_CFG(coef0) |\
SHARPEN_COEF1_CFG(coef1)
```

```
#define SHARPEN_COEFS23_CFG( coef2, coef3 )
```

Value:

```
SHARPEN_COEF2_CFG(coef2) |\
SHARPEN_COEF3_CFG(coef3)
```

```
#define SHARPEN_LIMITS_CFG( underShoot, overShoot )
```

Value:

```
SHARPEN_LIMITS_UNDERSHOOT_CFG(underShoot) |\
SHARPEN_LIMITS_OVERSHOOT_CFG(overShoot)
```

```
#define SHARPEN_LIMITS_OVERSHOOT_CFG( val ) ((val & SHARPEN_LIMITS_OVERSHOOT_MASK) << SHARPEN_LIMITS_OVERSHOOT_SHIFT)
```

```
#define SHARPEN_LIMITS_OVERSHOOT_MASK 0xFFFF
```

```
#define SHARPEN_LIMITS_OVERSHOOT_SHIFT 0x10
```

```
#define SHARPEN_LIMITS_UNDERSHOOT_CFG( val ) ((val & SHARPEN_LIMITS_UNDERSHOOT_MASK) << SHARPEN_LIMITS_UNDERSHOOT_SHIFT)
```

```
#define SHARPEN_LIMITS_UNDERSHOOT_MASK 0xFFFF
```

```
#define SHARPEN_LIMITS_UNDERSHOOT_SHIFT 0x0
```

```
#define SHARPEN_RANGESTOP0_CFG( rangeStop0, rangeStop1 )
```

Value:

```
SHARPEN_RANGESTOP0_CFG(rangeStop0) |\
SHARPEN_RANGESTOP1_CFG(rangeStop1)
```

```
#define SHARPEN_RANGESTOP0_CFG( val ) ((val & SHARPEN_RANGESTOP0_MASK) << SHARPEN_RANGESTOP0_SHIFT)
```

```
#define SHARPEN_RANGESTOP0_MASK 0xFFFF
```

```
#define SHARPEN_RANGESTOP0_SHIFT 0x0
```

```
#define SHARPEN_RANGESTOP1_CFG( val ) ((val & SHARPEN_RANGESTOP1_MASK) <<  
SHARPEN_RANGESTOP1_SHIFT)
```

```
#define SHARPEN_RANGESTOP1_MASK 0xFFFF
```

```
#define SHARPEN_RANGESTOP1_SHIFT 0x10
```

```
#define SHARPEN_RANGESTOP23_CFG( rangeStop2, rangeStop3 )
```

Value:

```
SHARPEN_RANGESTOP2_CFG (rangeStop2) | \
SHARPEN_RANGESTOP3_CFG (rangeStop3)
```

```
#define SHARPEN_RANGESTOP2_CFG( val ) ((val & SHARPEN_RANGESTOP2_MASK) <<  
SHARPEN_RANGESTOP2_SHIFT)
```

```
#define SHARPEN_RANGESTOP2_MASK 0xFFFF
```

```
#define SHARPEN_RANGESTOP2_SHIFT 0x0
```

```
#define SHARPEN_RANGESTOP3_CFG( val ) ((val & SHARPEN_RANGESTOP3_MASK) <<  
SHARPEN_RANGESTOP3_SHIFT)
```

```
#define SHARPEN_RANGESTOP3_MASK 0xFFFF
```

```
#define SHARPEN_RANGESTOP3_SHIFT 0x10
```

```
#define SHARPEN_STRENGTH_CFG( pos, neg )
```

Value:

```
SHARPEN_STRENGTH_POSITIVE_CFG (pos) | \
SHARPEN_STRENGTH_NEGATIVE_CFG (neg)
```

```
#define SHARPEN_STRENGTH_NEGATIVE_CFG( val ) ((val & SHARPEN_STRENGTH_NE-  
GATIVE_MASK) << SHARPEN_STRENGTH_NEGATIVE_SHIFT)
```

```
#define SHARPEN_STRENGTH_NEGATIVE_MASK 0xFFFF
```

```
#define SHARPEN_STRENGTH_NEGATIVE_SHIFT 0x0
```

```
#define SHARPEN_STRENGTH_POSITIVE_CFG( val ) ((val & SHARPEN_STRENGTH_POSI-  
TIVE_MASK) << SHARPEN_STRENGTH_POSITIVE_SHIFT)
```

```
#define SHARPEN_STRENGTH_POSITIVE_MASK 0xFFFF
```

```
#define SHARPEN_STRENGTH_POSITIVE_SHIFT 0x10
```



```
#define SIGMA_DNS_CFG( nf, dataWidth, passthr, format )
```

Value:

```
SIGMA_DNS_CFG_NF( nf ) | \
                                SIGMA_DNS_CFG_DATA_WIDTH( dataWidth ) | \
                                SIGMA_DNS_CFG_PASSTHRU_EN( passthr ) | \
                                SIGMA_DNS_CFG_FORMAT( format )
```

```
#define SIGMA_DNS_CFG_DATA_WIDTH( val ) (( val & SIGMA_DNS_CFG_DATA_WIDTH_MASK ) << SIGMA_DNS_CFG_DATA_WIDTH_SHIFT )
```

```
#define SIGMA_DNS_CFG_DATA_WIDTH_MASK 0xF
```

```
#define SIGMA_DNS_CFG_DATA_WIDTH_SHIFT 0x4
```

```
#define SIGMA_DNS_CFG_FORMAT( val ) (( val & SIGMA_DNS_CFG_FORMAT_MASK ) << SIGMA_DNS_CFG_FORMAT_SHIFT )
```

```
#define SIGMA_DNS_CFG_FORMAT_MASK 0x1
```

Referenced by sippLoadSigma().

```
#define SIGMA_DNS_CFG_FORMAT_SHIFT 0x0
```

Referenced by sippLoadSigma().

```
#define SIGMA_DNS_CFG_NF( val ) (( val & SIGMA_DNS_CFG_NF_MASK ) << SIGMA_DNS_CFG_NF_SHIFT )
```

```
#define SIGMA_DNS_CFG_NF_MASK 0x3FFF
```

```
#define SIGMA_DNS_CFG_NF_SHIFT 0x8
```

```
#define SIGMA_DNS_CFG_PASSTHRU_EN( val ) (( val & SIGMA_DNS_CFG_PASSTHRU_EN_MASK ) << SIGMA_DNS_CFG_PASSTHRU_EN_SHIFT )
```

```
#define SIGMA_DNS_CFG_PASSTHRU_EN_MASK 0x1
```

```
#define SIGMA_DNS_CFG_PASSTHRU_EN_SHIFT 0x1
```

```
#define SIGMA_DNS_FRM_HEIGHT( val ) (( val & SIGMA_DNS_FRM_HEIGHT_MASK ) << SIGMA_DNS_FRM_HEIGHT_SHIFT )
```

```
#define SIGMA_DNS_FRM_HEIGHT_MASK 0xFFFF
```

```
#define SIGMA_DNS_FRM_HEIGHT_SHIFT 0x10
```

```
#define SIGMA_DNS_FRM_WIDTH( val ) (( val & SIGMA_DNS_FRM_WIDTH_MASK) <<
SIGMA_DNS_FRM_WIDTH_SHIFT )
```

```
#define SIGMA_DNS_TRESH_CFG( t1, t2, t3, t4, t5, t6, t7, t8 )
```

Value:

```
SIGMA_DNS_TRESH_T1P0 (t1) |\
```

```
SIGMA_DNS_TRESH_T1P1 (t2) |\
SIGMA_DNS_TRESH_T2P0 (t3) |\
SIGMA_DNS_TRESH_T2P3 (t4) |\
SIGMA_DNS_TRESH_T1P3 (t5) |\
SIGMA_DNS_TRESH_T2P2 (t6) |\
SIGMA_DNS_TRESH_T1P2 (t7) |\
SIGMA_DNS_TRESH_T1P0 (t8)
```

```
#define SIGMA_DNS_TRESH_T1P0( val ) (( val & SIGMA_DNS_TRESH_T1P0_MASK) <<
SIGMA_DNS_TRESH_T1P0_SHIFT )
```

```
#define SIGMA_DNS_TRESH_T1P0_MASK 0xFF
```

```
#define SIGMA_DNS_TRESH_T1P0_SHIFT 0x0
```

```
#define SIGMA_DNS_TRESH_T1P1( val ) (( val & SIGMA_DNS_TRESH_T1P1_MASK) <<
SIGMA_DNS_TRESH_T1P1_SHIFT )
```

```
#define SIGMA_DNS_TRESH_T1P1_MASK 0xFF
```

```
#define SIGMA_DNS_TRESH_T1P1_SHIFT 0x10
```

```
#define SIGMA_DNS_TRESH_T1P2( val ) (( val & SIGMA_DNS_TRESH_T1P2_MASK) <<
SIGMA_DNS_TRESH_T1P2_SHIFT )
```

```
#define SIGMA_DNS_TRESH_T1P2_MASK 0xFF
```

```
#define SIGMA_DNS_TRESH_T1P2_SHIFT 0x0
```

```
#define SIGMA_DNS_TRESH_T1P3( val ) (( val & SIGMA_DNS_TRESH_T1P3_MASK) <<
SIGMA_DNS_TRESH_T1P3_SHIFT )
```

```
#define SIGMA_DNS_TRESH_T1P3_MASK 0xFF
```

```
#define SIGMA_DNS_TRESH_T1P3_SHIFT 0x10
```

```
#define SIGMA_DNS_TRESH_T2P0( val ) (( val & SIGMA_DNS_TRESH_T2P0_MASK) <<
SIGMA_DNS_TRESH_T2P0_SHIFT )
```

```
#define SIGMA_DNS_TRESH_T2P0_MASK 0xFF
```

```
#define SIGMA_DNS_TRESH_T2P0_SHIFT 0x8
```

```
#define SIGMA_DNS_TRESH_T2P1( val ) (( val & SIGMA_DNS_TRESH_T2P1_MASK) <<
SIGMA_DNS_TRESH_T2P1_SHIFT )
```

```
#define SIGMA_DNS_TRESH_T2P1_MASK 0xFF

#define SIGMA_DNS_TRESH_T2P1_SHIFT 0x18

#define SIGMA_DNS_TRESH_T2P2( val ) (( val & SIGMA_DNS_TRESH_T2P2_MASK) <<
SIGMA_DNS_TRESH_T2P2_SHIFT )

#define SIGMA_DNS_TRESH_T2P2_MASK 0xFF

#define SIGMA_DNS_TRESH_T2P2_SHIFT 0x8

#define SIGMA_DNS_TRESH_T2P3( val ) (( val & SIGMA_DNS_TRESH_T2P3_MASK) <<
SIGMA_DNS_TRESH_T2P3_SHIFT )

#define SIGMA_DNS_TRESH_T2P3_MASK 0xFF

#define SIGMA_DNS_TRESH_T2P3_SHIFT 0x18

#define SIPP_HW_EDGE_OPERATOR_BUFFER__HEIGHT( val ) ((val & EDGE_OPERATOR_-
BUFFER_HEIGHT_MASK) <<EDGE_OPERATOR_BUFFER_HEIGHT_shift)

#define SIPP_HW_EDGE_OPERATOR_BUFFER__WIDTH( val ) ((val & EDGE_OPERATOR_-
BUFFER_WIDTH_MASK) <<EDGE_OPERATOR_BUFFER_WIDTH_SHIFT)

#define SIPP_HW_POLY_FIR_CLAMP_DISABLE 0x0

#define SIPP_HW_POLY_FIR_CLAMP_ENABLE 0x1

#define SIPP_LSC_CFG_DATA_WIDTH( val ) ((val & SIPP_LSC_CFG_DATA_WIDTH_MASK)
<< SIPP_LSC_CFG_DATA_WIDTH_SHIFT)

#define SIPP_LSC_CFG_DATA_WIDTH_MASK 0xF

#define SIPP_LSC_CFG_DATA_WIDTH_SHIFT 0x4

#define SIPP_LSC_CFG_FORMAT( val ) ((val & SIPP_LSC_CFG_FORMAT_MASK ) <<
SIPP_LSC_CFG_FORMAT_SHIFT)

#define SIPP_LSC_CFG_FORMAT_MASK 0x1

#define SIPP_LSC_CFG_FORMAT_SHIFT 0x0

#define SIPP_LSC_FRM_DIM_CFG( height, width )
```

Value:

```
SIPP_LSC_FRM_DIM_HEIGHT_CFG(height)|\
SIPP_LSC_FRM_DIM_HEIGHT_CFG(width)
```

```
#define SIPP_LSC_FRM_DIM_HEIGHT_CFG( val ) ((val & SIPP_LSC_FRM_DIM_HEIGHT_-
MASK) << SIPP_LSC_FRM_DIM_HEIGHT_SHIFT)
```

```
#define SIPP_LSC_FRM_DIM_HEIGHT_MASK 0xFFFF

#define SIPP_LSC_FRM_DIM_HEIGHT_SHIFT 0x10

#define SIPP_LSC_FRM_DIM_WIDTH_CFG( val ) ((val & SIPP_LSC_FRM_DIM_WIDTH_MASK) << SIPP_LSC_FRM_DIM_WIDTH_SHIFT)

#define SIPP_LSC_FRM_DIM_WIDTH_MASK 0xFFFF

#define SIPP_LSC_FRM_DIM_WIDTH_SHIFT 0x0

#define SIPP_LSC_GM_DIM_CFG( height, width )
```

Value:

```
SIPP_LSC_GM_DIM_HEIGHT_CFG(height) | \
                                SIPP_LSC_GM_DIM_WIDTH_CFG(width)

#define SIPP_LSC_GM_DIM_HEIGHT_CFG( val ) ((val & SIPP_LSC_GM_DIM_HEIGHT_MASK) << SIPP_LSC_GM_DIM_HEIGHT_SHIFT)

#define SIPP_LSC_GM_DIM_HEIGHT_MASK 0x3FF

#define SIPP_LSC_GM_DIM_HEIGHT_SHIFT 0x10

#define SIPP_LSC_GM_DIM_WIDTH_CFG( val ) ((val & SIPP_LSC_GM_DIM_WIDTH_MASK) << SIPP_LSC_GM_DIM_WIDTH_SHIFT)

#define SIPP_LSC_GM_DIM_WIDTH_MASK 0x3FF

#define SIPP_LSC_GM_DIM_WIDTH_SHIFT 0x0

#define SIPP_UPFIRDN_ID 15
```

Referenced by sippAddFilterToPipe().

7.226.2 Enumeration Type Documentation

enum **eBayerOrder**

Enumerator

eBayerOrderGRBG
eBayerOrderRGGB
eBayerOrderGBRG
eBayerOrderBGGR

enum **eRawInputFmt**

Enumerator

eRawFormatPlanar

eRawFormatBayer

7.227 sippHwChromaDns.c File Reference

SIPP engine.

```
#include <sipp.h>
#include <sippInternal.h>
```

7.227.1 Detailed Description

SIPP engine.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.228 sippHwColComb.c File Reference

SIPP engine.

```
#include <sipp.h>
#include <sippInternal.h>
```

7.228.1 Detailed Description

SIPP engine.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.229 sippHwCommon.c File Reference

SIPP engine.

```
#include <sipp.h>
#include <sippInternal.h>
```

Macros

- #define `SIPP_CQ_ADD_WRITE`(qu, address, val)

Functions

- `u32 sippIbufSetup` (`SippFilter *fptr`, `u32 parNo`)
- `u32 sippObufSetup` (`SippFilter *fptr`, `u32 oBufIdx`)
- `u32 sippBufSetupIrqRate` (`SippHwBuf *buf`, `u32 oBuf`, `u32 bufId`, `u32 numLinesPerIrq`)
- `u32 sippBufSetupIrqRateCQ` (`SippHwBuf *buf`, `u32 oBuf`, `u32 bufId`, `u32 numLinesPerIrq`, `psSippCMDQu pQu`)
- `u32 sippBufGetObufCtx` (`SippFilter *fptr`, `u32 oBufIdx`)

7.229.1 Detailed Description

SIPP engine.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: `common/license.txt`

7.229.2 Macro Definition Documentation

```
#define SIPP_CQ_ADD_WRITE( qu, address, val )
```

Value:

```
{ \
    qu->quEntry[qu->quNum].addr = address; \
    qu->quEntry[qu->quNum].value = val; \
    qu->quNum++; \
}
```

Referenced by `sippBufSetupIrqRateCQ()`.

7.229.3 Function Documentation

`u32 sippBufGetObufCtx` (`SippFilter * fptr`, `u32 oBufIdx`)

`u32 sippBufSetupIrqRate` (`SippHwBuf * buf`, `u32 oBuf`, `u32 bufId`, `u32 numLinesPerIrq`)

`u32 sippBufSetupIrqRateCQ` (`SippHwBuf * buf`, `u32 oBuf`, `u32 bufId`, `u32 numLinesPerIrq`, `psSippCMDQu pQu`)

`u32 sippIbufSetup` (`SippFilter * fptr`, `u32 parNo`)

Referenced by `sippInitMipiTx()`, and `sippInitSigma()`.

u32 sippObufSetup (**SippFilter** * fptr, **u32** oBufIdx)

Referenced by sippInitMipiRx(), and sippInitSigma().

7.230 sippHwCommon_ma2x5x.h File Reference

```
#include <stdint.h>
```

Macros

- #define REV1_DEF 0
- #define REV2_DEF 1
- #define SIPP_INCDEC_BIT 30
- #define SIPP_START_BIT 30
- #define SIPP_CTXUP_BIT 31
- #define SIPP_INCDEC_BIT_MASK (1 << SIPP_INCDEC_BIT)
- #define SIPP_START_BIT_MASK (1 << SIPP_START_BIT)
- #define SIPP_CTXUP_BIT_MASK (1 << SIPP_CTXUP_BIT)
- #define SIPP_CBL_OFFSET 16
- #define SIPP_IMGDIM_SIZE 16
- #define SIPP_IMGDIM_MASK 0xffff
- #define SIPP_KL_MASK 0xf
- #define SIPP_NL_MASK 0x3ff
- #define SIPP_SC_MASK 0x1
- #define SIPP_OF_MASK 0x7
- #define SIPP_NP_MASK 0xf
- #define SIPP_FO_MASK 0x7
- #define SIPP_LS_MASK 0x1ffffff
- #define SIPP_PS_MASK 0x1ffffff
- #define SIPP_IR_MASK 0xf
- #define SIPP_IC_MASK 0x3
- #define SIPP_SS_MASK 0xf
- #define SIPP_CS_MASK 0xffff8
- #define SIPP_SB_MASK 0x1
- #define SIPP_NL_OFFSET 0
- #define SIPP_SC_OFFSET 10
- #define SIPP_SL_OFFSET 12
- #define SIPP_OF_OFFSET 12
- #define SIPP_NP_OFFSET 24
- #define SIPP_FO_OFFSET 28
- #define SIPP_IR_OFFSET 0
- #define SIPP_IC_OFFSET 4
- #define SIPP_SS_OFFSET 8
- #define SIPP_CS_OFFSET 16
- #define CONCAT5(A,B, C, D,E) A ## B ## C ## D ## E

- #define CONCAT7(A, B, C, D, E, F, G) A ## B ## C ## D ## E ## F ## G
- #define ENABLED 1
- #define DISABLED 0
- #define ACTIVE ENABLED
- #define DEFAULT 0
- #define SHADOW 1
- #define SIGMA_KERNEL_SIZE 5
- #define RAW_KERNEL_SIZE 5
- #define HIST_KERNEL_SIZE 3
- #define LSC_KERNEL_SIZE 1
- #define DBYR_KERNEL_SIZE 11
- #define DBYR_PPM_KERNEL_SIZE 3
- #define CHROMA_V_KERNEL_SIZE 21
- #define CHROMA_H0_KERNEL_SIZE 23
- #define CHROMA_H1_KERNEL_SIZE 17
- #define CHROMA_H2_KERNEL_SIZE 13
- #define CHROMA_REF_KERNEL_SIZE 21
- #define LUMA_KERNEL_SIZE 7
- #define LUMA_REF_KERNEL_SIZE 11
- #define MED_KERNEL_SIZE 7
- #define MED_LUMA_KERNEL_SIZE 1
- #define SHARPEN_KERNEL_SIZE 7
- #define LUT_KERNEL_SIZE 1
- #define UPFIRDN_KERNEL_SIZE 7
- #define EDGE_OP_KERNEL_SIZE 3
- #define CC_LUMA_KERNEL_SIZE 1
- #define CC_CHROMA_KERNEL_SIZE 5
- #define MIPI_RX_KERNEL_SIZE 1
- #define MIPI_TX_KERNEL_SIZE 1
- #define SIPP_MIPI_RX_INSTANCES 4
- #define SIPP_MIPI_TX_INSTANCES 2
- #define SIPP_RAW_NLBRC 2
- #define SIPP_LSC_NLBRC 2
- #define SIPP_DBYR_NLBRC 2
- #define SIPP_CHROMA_NLBRC 3
- #define SIPP_LUMA_NLBRC 4
- #define SIPP_SHARPEN_NLBRC 2
- #define SIPP_UPFIRDN_NLBRC 1
- #define SIPP_MED_NLBRC 1
- #define SIPP_LUT_NLBRC 1
- #define SIPP_EDGE_OP_NLBRC 1
- #define SIPP_CONV_NLBRC 2
- #define SIPP_HARRIS_NLBRC 1
- #define SIPP_CC_NLBRC 4
- #define SIPP_MIPI_RX_NLBRC 0
- #define SIPP_MIPI_TX_NLBRC 1
- #define SIPP_NLBWC 1

- #define DEF_SLICE_SIZE 128*1024
- #define CMX_NSLICES 16
- #define MAX_PLANES 16
- #define AMC_WIDTH 64
- #define PLANAR 0
- #define BAYER 1
- #define GRBG 0
- #define RGGB 1
- #define GBRG 2
- #define BGGR 3
- #define P_RGB 0
- #define P_BGR 1
- #define P_RBG 2
- #define P_BRG 3
- #define P_GRB 4
- #define P_GBR 5
- #define NORMAL_MODE 0
- #define PRE_FP16_GRAD 1
- #define PRE_U8_GRAD 2
- #define SCALED_MAGN_16BIT 0
- #define SCALED_MAGN_8BIT 1
- #define MAGN_ORIENT_16BIT 2
- #define ORIENT_8BIT 3
- #define SCALED_GRADIENTS_16BIT 4
- #define SCALED_GRADIENTS_32BIT 5
- #define NORMAL_THETA 0
- #define X_AXIS_REFL 1
- #define XY_AXIS_REFL 2
- #define STATS_AE_BASE_ADR O_BASE(SIPP_STATS_ID)
- #define STATS_AF_BASE_ADR O_BASE(SIPP_AF_STATS_ID)
- #define STATS_HIST_LUMA_BASE_ADR O_BASE(SIPP_LUMA_HIST_ID)
- #define STATS_HIST_RGB_BASE_ADR O_BASE(SIPP_RGB_HIST_ID)
- #define SIPP_DOG_NLBRC 1
- #define DOG_INNER_KERNEL_SIZE 11
- #define DOGLTM_KERNEL_SIZE 15
- #define MULTI_UP_KERNEL_SIZE (((DOGLTM_KERNEL_SIZE + 1) >> 1) + 1)
- #define DOG_SUB_KERNEL_SIZE 1
- #define SIPP_GCHR_NLBRC 3
- #define RSZBHF_KERNEL_SIZE 2
- #define PFLARE_KERNEL_SIZE 3
- #define DARK_DESAT_KERNEL_SIZE 1
- #define GEN_CHROMA_KERNEL_SIZE 7
- #define CHRGAUSS_KERNEL_SIZE 3
- #define GREYDESAT_KERNEL_SIZE 1
- #define SIPP_DOGL_NLBRC 1
- #define BGGEN_KERNEL_SIZE 11
- #define RSZBTWO_KERNEL_SIZE 2
- #define LTMAPPLY_KERNEL_SIZE 1
- #define SIPP_SIGMA_NLBRC 1

Enumerations

- enum `Revision` { `REV1` = `REV1_DEF`, `REV2` = `REV2_DEF` }
- enum `AddressType` {
`OTHER`, `CMX_ADDRESS`, `CMX_MIRRORED_ADDRESS`, `DDR_ADDRESS`,
`DDR_MIRRORED_ADDRESS` }

7.230.1 Macro Definition Documentation

```
#define ACTIVE_ENABLED
```

```
#define AMC_WIDTH 64
```

```
#define BAYER 1
```

```
#define BGEN_KERNEL_SIZE 11
```

```
#define BGGR 3
```

Referenced by `sippUtilOrderPixels()`.

```
#define CC_CHROMA_KERNEL_SIZE 5
```

```
#define CC_LUMA_KERNEL_SIZE 1
```

```
#define CHRGauss_KERNEL_SIZE 3
```

```
#define CHROMA_H0_KERNEL_SIZE 23
```

```
#define CHROMA_H1_KERNEL_SIZE 17
```

```
#define CHROMA_H2_KERNEL_SIZE 13
```

```
#define CHROMA_REF_KERNEL_SIZE 21
```

```
#define CHROMA_V_KERNEL_SIZE 21
```

```
#define CMX_NSLICES 16
```

Referenced by `sippAllocCmxLineBuffers()`, `sippAllocCmxLineBuffersOPipe()`, `sippMemLBConsolidateRegions()`, and `sippMemLBMatchRegionsToChunks()`.

```
#define CONCAT5( A, B, C, D, E ) A ## B ## C ## D ## E
```

```
#define CONCAT7( A, B, C, D, E, F, G ) A ## B ## C ## D ## E ## F ## G
```

```
#define DARK_DESAT_KERNEL_SIZE 1
```

```
#define DBYR_KERNEL_SIZE 11
```

```
#define DBYR_PPM_KERNEL_SIZE 3

#define DEF_SLICE_SIZE 128*1024

#define DEFAULT 0

#define DISABLED 0

#define DOG_INNER_KERNEL_SIZE 11

#define DOG_SUB_KERNEL_SIZE 1

#define DOGLTM_KERNEL_SIZE 15

#define EDGE_OP_KERNEL_SIZE 3

#define ENABLED 1

#define GBRG 2
```

Referenced by sippUtilOrderPixels().

```
#define GEN_CHROMA_KERNEL_SIZE 7

#define GRBG 0
```

Referenced by sippUtilOrderPixels().

```
#define GREYDESAT_KERNEL_SIZE 1

#define HIST_KERNEL_SIZE 3

#define LSC_KERNEL_SIZE 1

#define LTMAPPLY_KERNEL_SIZE 1

#define LUMA_KERNEL_SIZE 7

#define LUMA_REF_KERNEL_SIZE 11

#define LUT_KERNEL_SIZE 1

#define MAGN_ORIENT_16BIT 2

#define MAX_PLANES 16

#define MED_KERNEL_SIZE 7

#define MED_LUMA_KERNEL_SIZE 1

#define MIPI_RX_KERNEL_SIZE 1
```

```
#define MIPI_TX_KERNEL_SIZE 1

#define MULTI_UP_KERNEL_SIZE (((DOGLTM_KERNEL_SIZE + 1) >> 1) + 1)

#define NORMAL_MODE 0

#define NORMAL_THETA 0

#define ORIENT_8BIT 3

#define P_BGR 1

#define P_BRG 3

#define P_GBR 5

#define P_GRB 4

#define P_RBG 2

#define P_RGB 0

#define PFLARE_KERNEL_SIZE 3

#define PLANAR 0

#define PRE_FP16_GRAD 1

#define PRE_U8_GRAD 2

#define RAW_KERNEL_SIZE 5

#define REV1_DEF 0

#define REV2_DEF 1

#define RGGB 1
```

Referenced by sippUtilOrderPixels().

```
#define RSZBHF_KERNEL_SIZE 2

#define RSZBTWO_KERNEL_SIZE 2

#define SCALED_GRADIENTS_16BIT 4

#define SCALED_GRADIENTS_32BIT 5

#define SCALED_MAGN_16BIT 0

#define SCALED_MAGN_8BIT 1
```

```
#define SHADOW 1

#define SHARPEN_KERNEL_SIZE 7

#define SIGMA_KERNEL_SIZE 5

#define SIPP_CBL_OFFSET 16

#define SIPP_CC_NLBRC 4

#define SIPP_CHROMA_NLBRC 3

#define SIPP_CONV_NLBRC 2

#define SIPP_CS_MASK 0xff8

#define SIPP_CS_OFFSET 16

#define SIPP_CTXUP_BIT 31

#define SIPP_CTXUP_BIT_MASK (1 << SIPP_CTXUP_BIT)

#define SIPP_DBYR_NLBRC 2

#define SIPP_DOG_NLBRC 1

#define SIPP_DOGL_NLBRC 1

#define SIPP_EDGE_OP_NLBRC 1

#define SIPP_FO_MASK 0x7

#define SIPP_FO_OFFSET 28
```

Referenced by sippIbufSetup(), and sippObufSetup().

```
#define SIPP_GCHR_NLBRC 3

#define SIPP_HARRIS_NLBRC 1

#define SIPP_IC_MASK 0x3

#define SIPP_IC_OFFSET 4

#define SIPP_IMGDIM_MASK 0xffff

#define SIPP_IMGDIM_SIZE 16
```

Referenced by sippInitMipiRx(), sippInitMipiTx(), and sippInitSigma().

```
#define SIPP_INCDEC_BIT 30
```

```
#define SIPP_INCDEC_BIT_MASK (1 << SIPP_INCDEC_BIT)

#define SIPP_IR_MASK 0xf

#define SIPP_IR_OFFSET 0

#define SIPP_KL_MASK 0xf

#define SIPP_LS_MASK 0x1ffffff

#define SIPP_LSC_NLBRC 2

#define SIPP_LUMA_NLBRC 4

#define SIPP_LUT_NLBRC 1

#define SIPP_MED_NLBRC 1

#define SIPP_MIPI_RX_INSTANCES 4

#define SIPP_MIPI_RX_NLBRC 0

#define SIPP_MIPI_TX_INSTANCES 2

#define SIPP_MIPI_TX_NLBRC 1

#define SIPP_NL_MASK 0x3ff

#define SIPP_NL_OFFSET 0

#define SIPP_NLBWC 1

#define SIPP_NP_MASK 0xf

#define SIPP_NP_OFFSET 24
```

Referenced by sippIbufSetup(), sippLoadMipiRx(), sippLoadSigma(), and sippObufSetup().

```
#define SIPP_OF_MASK 0x7

#define SIPP_OF_OFFSET 12

#define SIPP_PS_MASK 0x1ffffff

#define SIPP_RAW_NLBRC 2

#define SIPP_SB_MASK 0x1

#define SIPP_SC_MASK 0x1

#define SIPP_SC_OFFSET 10
```

```
#define SIPP_SHARPEN_NLBRC 2

#define SIPP_SIGMA_NLBRC 1

#define SIPP_SL_OFFSET 12

#define SIPP_SS_MASK 0xf

#define SIPP_SS_OFFSET 8

#define SIPP_START_BIT 30

#define SIPP_START_BIT_MASK (1 << SIPP_START_BIT)

#define SIPP_UPFIRDN_NLBRC 1

#define STATS_AE_BASE_ADR O_BASE(SIPP_STATS_ID)

#define STATS_AF_BASE_ADR O_BASE(SIPP_AF_STATS_ID)

#define STATS_HIST_LUMA_BASE_ADR O_BASE(SIPP_LUMA_HIST_ID)

#define STATS_HIST_RGB_BASE_ADR O_BASE(SIPP_RGB_HIST_ID)

#define UPFIRDN_KERNEL_SIZE 7

#define X_AXIS_REFL 1

#define XY_AXIS_REFL 2
```

7.230.2 Enumeration Type Documentation

enum **AddressType**

Enumerator

```
OTHER
CMX_ADDRESS
CMX_MIRRORED_ADDRESS
DDR_ADDRESS
DDR_MIRRORED_ADDRESS
```

enum **Revision**

Enumerator

```
REV1
REV2
```

7.231 sippHwConv.c File Reference

SIPP engine.

```
#include <sipp.h>
#include <sippInternal.h>
```

7.231.1 Detailed Description

SIPP engine.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.232 sippHwDebayer.c File Reference

SIPP engine.

```
#include <sipp.h>
#include <sippInternal.h>
```

7.232.1 Detailed Description

SIPP engine.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.233 sippHwDefs.h File Reference

SIPP engine.

Data Structures

- struct [DmaParam](#)
Parameter structure of the [DMA](#) filter.

7.233.1 Detailed Description

SIPP engine.

Copyright

All code copyright Movidius Ltd 2015, all rights reserved. For License Warranty see: common/license.txt

7.234 sippHwDefs_ma2x5x.h File Reference

Config data structures for MA2x5x SIPP HW filters. Most registers exposed through these data structures are described in detail in the MDK Programmer's Guide.

Data Structures

- struct [MedParam](#)
Parameter structure of the median filter.
- struct [LscParam](#)
Parameter structure of the lsc filter.
- struct [RawParam](#)
Parameter structure of the raw filter.
- struct [DbyrParam](#)
Parameter structure of the debayer filter.
- struct [UsmParam](#)
Parameter structure of the sharpen filter.
- struct [YDnsParam](#)
Parameter structure of the ydns filter.
- struct [ChrDnsParam](#)
Parameter structure of the chormadns filter.
- struct [LutParam](#)
*Parameter structure of the *lut* filter.*
- struct [ColCombParam](#)
Parameter structure of the colorcomb filter.
- struct [ConvParam](#)
Parameter structure of the convolution filter.
- struct [HarrisParam](#)
Parameter structure of the harriscorners filter.
- struct [PolyFirParam](#)
Parameter structure of the polyphasefir filter.
- struct [EdgeParam](#)
Parameter structure of the edgeoperator filter.
- struct [SigmaParam](#)
Parameter structure of the edgeoperator filter.
- struct [GenChrParam](#)
Parameter structure of the edgeoperator filter.
- struct [DogLtmParam](#)
- struct [MipiRxParam](#)
*Parameter structure of the *MIPI Rx* filter.*

- struct [MipiTxParam](#)

Parameter structure of the [MIPI Rx](#) filter.

- struct [MipiTxLoopbackParam](#)
- struct [MipiRxLoopbackParam](#)

Enumerations

- enum [PolyModes](#) { [POLY_MODE_AUTO](#) = 0, [POLY_MODE_ADVANCE](#) = 1 }
- enum [PolyScalerType](#) { [POLY_LANCZOS](#) = 0, [POLY_BICUBIC](#) = 1, [POLY_BILINEAR](#) = 2 }
- enum [PolyPlaneMode](#) {
[POLY_PLANE_ALL](#) = 0, [POLY_PLANE_Y](#) = 1, [POLY_PLANE_U](#) = 2, [POLY_PLANE_V](#) = 3,
[POLY_PLANE_UV](#) = 4 }

Functions

- void [packConv5x5CCM](#) ([ConvParam](#) *cfg, [UInt16](#) *ccm5x5)
- void [packConv3x3CCM](#) ([ConvParam](#) *cfg, [UInt16](#) *ccm3x3)
- void [packColCombCCM](#) ([ColCombParam](#) *cfg, float *ccm3x3)
- void [packLumaDnsGaussLut](#) ([YDnsParam](#) *cfg, [UInt8](#) *lut)
- void [cfgMipiRxLoopback](#) ([MipiRxLoopbackParam](#) *cfg)
- void [cfgMipiTxLoopback](#) ([MipiTxLoopbackParam](#) *cfg)
- void [startMipiTxLoopback](#) ([MipiTxLoopbackParam](#) *cfg)

7.234.1 Detailed Description

Config data structures for MA2x5x SIPP HW filters. Most registers exposed through these data structures are described in detail in the MDK Programmer's Guide.

Copyright

All code copyright Movidius Ltd 2015, all rights reserved. For License Warranty see: common/license.txt

Data members marked (Private) are computed internally by SIPP framework and should not be touched by user.

7.234.2 Function Documentation

void [cfgMipiRxLoopback](#) ([MipiRxLoopbackParam](#) * cfg)

void [cfgMipiTxLoopback](#) ([MipiTxLoopbackParam](#) * cfg)

void [packColCombCCM](#) ([ColCombParam](#) * cfg, float * ccm3x3)

void [packConv3x3CCM](#) ([ConvParam](#) * cfg, [UInt16](#) * ccm3x3)

void [packConv5x5CCM](#) ([ConvParam](#) * cfg, [UInt16](#) * ccm5x5)

```
void packLumaDnsGaussLut ( YDnsParam * cfg, UInt8 * lut )
```

```
void startMipiTxLoopback ( MipiTxLoopbackParam * cfg )
```

7.235 sippHwDogLtm.c File Reference

SIPP engine.

```
#include <sipp.h>
#include <sippInternal.h>
```

7.235.1 Detailed Description

SIPP engine.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see:
common/license.txt

7.236 sippHwEdge.c File Reference

SIPP engine.

```
#include <sipp.h>
#include <sippInternal.h>
```

7.236.1 Detailed Description

SIPP engine.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see:
common/license.txt

7.237 sippHwGenChroma.c File Reference

SIPP engine.

```
#include <sipp.h>
#include <sippInternal.h>
```

7.237.1 Detailed Description

SIPP engine.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.238 sippHwHarris.c File Reference

SIPP engine.

```
#include <sipp.h>
#include <sippInternal.h>
```

7.238.1 Detailed Description

SIPP engine.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.239 sippHwIds.h File Reference

HW filter related macros.

Macros

- #define [SIPP_OPIPE_ID](#) -1 /* Convention to mark the oPipe */
- #define [SIPP_SIGMA_ID](#) 0 /* Sigma denoise */
- #define [SIPP_LSC_ID](#) 1 /* Lens shading filter */
- #define [SIPP_RAW_ID](#) 2 /* RAW filter */
- #define [SIPP_DBYR_ID](#) 3 /* Debayer */
- #define [SIPP_DOGL_ID](#) 4 /* Difference of Gaussians/Local tone mapping */
- #define [SIPP_LUMA_ID](#) 5 /* Luma denoise in/out */
- #define [SIPP_SHARPEN_ID](#) 6 /* Sharpening */
- #define [SIPP_CGEN_ID](#) 7 /* Generate chroma filter */
- #define [SIPP_MED_ID](#) 8 /* Median */
- #define [SIPP_CHROMA_ID](#) 9 /* Chroma denoise */
- #define [SIPP_CC_ID](#) 10 /* Colour combination */
- #define [SIPP_LUT_ID](#) 11 /* Look-up table */
- #define [SIPP_EDGE_OP_ID](#) 12 /* Edge operator */
- #define [SIPP_CONV_ID](#) 13 /* Programmable convolution */
- #define [SIPP_HARRIS_ID](#) 14 /* Harris corners */
- #define [SIPP_UPFIRDN0_ID](#) 15 /* Polyphase FIR filter[0] */
- #define [SIPP_UPFIRDN1_ID](#) 16 /* Polyphase FIR filter[1] */
- #define [SIPP_UPFIRDN2_ID](#) 17 /* Polyphase FIR filter[2] */

- #define SIPP_MIPI_TX0_ID 18 /* MIPI Tx[0] filter (input buffer only) */
- #define SIPP_MIPI_TX1_ID 19 /* MIPI Tx[1] filter (input buffer only) */
- #define SIPP_MIPI_RX0_ID 20 /* MIPI Rx[0] filter (output buffer only) */
- #define SIPP_MIPI_RX1_ID 21 /* MIPI Rx[1] filter (output buffer only) */
- #define SIPP_MIPI_RX2_ID 22 /* MIPI Rx[2] filter (output buffer only) */
- #define SIPP_MIPI_RX3_ID 23 /* MIPI Rx[3] filter (output buffer only) */
- #define SIPP_LSC_GM_ID 20 /* Lens shading correction - gain mesh buffer */
- #define SIPP_MED_LUMA_ID 21 /* Median filter - chroma median reference luma buffer */
- #define SIPP_CC_CHROMA_ID 22 /* Colour combination - chroma buffer */
- #define SIPP_LUT_LOAD_ID 23 /* LUT filter - LUT buffer */
- #define SIPP_LUMA_C4LUT_ID 24 /* Luma denoise - cosine-4th law LUT buffer */
- #define SIPP_CC_3DLUT_ID 25 /* Colour combination - 3D LUT buffer */
- #define SIPP_RAW_DEFECT_ID 26 /* RAW filter - defect pixel list for static defect correction */
- #define SIPP_DBYR_LUMA_ID 18 /* Debayer luma buffer */
- #define SIPP_STATS_ID 19 /* RAW statistics */
- #define SIPP_AF_STATS_ID 24 /* RAW AF statistics */
- #define SIPP_LUMA_HIST_ID 25 /* RAW Luma histogram */
- #define SIPP_RGB_HIST_ID 26 /* RAW RGB histogram */
- #define SIPP_CHROMA_REF_ID 17 /* Chroma denoise reference in */
- #define SIPP_LUMA_REF_ID 18 /* Luma denoise reference in */
- #define SIPP_MAX_ID 26
- #define SIPP_MIN_FILTER_ID SIPP_SIGMA_ID
- #define SIPP_MAX_FILTER_ID SIPP_MIPI_RX3_ID
- #define SIPP_RESERVED_ID 13 /* 13 - Reserved */
- #define SIPP_V2_RESERVED_ID SIPP_V2_MAX_ID+1 /* No ID Reserved */
- #define SIPP_SIGMA_ID_MASK (1 << SIPP_SIGMA_ID)
- #define SIPP_RAW_ID_MASK (1 << SIPP_RAW_ID)
- #define SIPP_STATS_MASK (1 << SIPP_STATS_ID)
- #define SIPP_LSC_ID_MASK (1 << SIPP_LSC_ID)
- #define SIPP_LSC_GM_ID_MASK (1 << SIPP_LSC_GM_ID)
- #define SIPP_DBYR_ID_MASK (1 << SIPP_DBYR_ID)
- #define SIPP_CHROMA_ID_MASK (1 << SIPP_CHROMA_ID)
- #define SIPP_LUMA_ID_MASK (1 << SIPP_LUMA_ID)
- #define SIPP_LUMA_REF_ID_MASK (1 << SIPP_LUMA_REF_ID)
- #define SIPP_SHARPEN_ID_MASK (1 << SIPP_SHARPEN_ID)
- #define SIPP_UPFIRDN0_ID_MASK (1 << SIPP_UPFIRDN0_ID)
- #define SIPP_UPFIRDN1_ID_MASK (1 << SIPP_UPFIRDN1_ID)
- #define SIPP_UPFIRDN2_ID_MASK (1 << SIPP_UPFIRDN2_ID)
- #define SIPP_MED_ID_MASK (1 << SIPP_MED_ID)
- #define SIPP_LUT_ID_MASK (1 << SIPP_LUT_ID)
- #define SIPP_LUT_LOAD_MASK (1 << SIPP_LUT_LOAD_ID)
- #define SIPP_EDGE_OP_ID_MASK (1 << SIPP_EDGE_OP_ID)
- #define SIPP_CONV_ID_MASK (1 << SIPP_CONV_ID)
- #define SIPP_HARRIS_ID_MASK (1 << SIPP_HARRIS_ID)
- #define SIPP_CC_ID_MASK (1 << SIPP_CC_ID)
- #define SIPP_CC_CHROMA_ID_MASK (1 << SIPP_CC_CHROMA_ID)

- #define SIPP_DOGL_ID_MASK (1 << SIPP_DOGL_ID)
- #define SIPP_CGEN_ID_MASK (1 << SIPP_CGEN_ID)
- #define SIPP_MIPI_TX0_ID_MASK (1 << SIPP_MIPI_TX0_ID)
- #define SIPP_MIPI_TX1_ID_MASK (1 << SIPP_MIPI_TX1_ID)
- #define SIPP_MIPI_RX0_ID_MASK (1 << SIPP_MIPI_RX0_ID)
- #define SIPP_MIPI_RX1_ID_MASK (1 << SIPP_MIPI_RX1_ID)
- #define SIPP_MIPI_RX2_ID_MASK (1 << SIPP_MIPI_RX2_ID)
- #define SIPP_MIPI_RX3_ID_MASK (1 << SIPP_MIPI_RX3_ID)
- #define SIPP_DMA_ID (SIPP_MAX_ID+1)
- #define SIPP_SVU_ID (SIPP_MAX_ID+2)
- #define EXE_NUM (SIPP_MAX_ID+2+1)
- #define SIPP_FAKE_ID (EXE_NUM + 1)
- #define SIPP_HW_FILTER_MASK_SIZE ((EXE_NUM+31)/32)
- #define SIPP_FILTER_LL_B_MASK
- #define SIPP_FILTER_FULL_LL_B_MASK

7.239.1 Detailed Description

HW filter related macros.

Copyright

All code copyright Movidius Ltd 2016, all rights reserved. For License Warranty see: common/license.txt

7.239.2 Macro Definition Documentation

#define EXE_NUM (SIPP_MAX_ID+2+1)

Referenced by sippCoreHwInitialLoad(), and sippInitPipeline().

#define SIPP_AF_STATS_ID 24 /* RAW AF statistics */

#define SIPP_CC_3DLUT_ID 25 /* Colour combination - 3D LUT buffer */

#define SIPP_CC_CHROMA_ID 22 /* Colour combination - chroma **buffer** */

#define SIPP_CC_CHROMA_ID_MASK (1 << SIPP_CC_CHROMA_ID)

#define SIPP_CC_ID 10 /* Colour combination */

Referenced by sippCheckOPipeConnectionLut(), sippGenericSchCreateSEFromFilter(), sippGenericSchCreateSEFromOSE(), and sippGenericScheduleSetBufConsModels().

#define SIPP_CC_ID_MASK (1 << SIPP_CC_ID)

#define SIPP_CGEN_ID 7 /* Generate chroma filter */

Referenced by sippCheckOPipeConnectionMedian(), and sippGenericScheduleSetBufConsModels().

```
#define SIPP_CGEN_ID_MASK (1 << SIPP_CGEN_ID)
```

```
#define SIPP_CHROMA_ID 9 /* Chroma denoise */
```

Referenced by sippCheckOPipeConnectionColourComb().

```
#define SIPP_CHROMA_ID_MASK (1 << SIPP_CHROMA_ID)
```

```
#define SIPP_CHROMA_REF_ID 17 /* Chroma denoise reference in */
```

```
#define SIPP_CONV_ID 13 /* Programmable convolution */
```

```
#define SIPP_CONV_ID_MASK (1 << SIPP_CONV_ID)
```

```
#define SIPP_DBYR_ID 3 /* Debayer */
```

Referenced by sippBufGetObufCtx(), sippCheckOPipeConnectionDoGLTM(), sippCheckOPipeConnectionGenChroma(), and sippCheckOPipeConnectionMedian().

```
#define SIPP_DBYR_ID_MASK (1 << SIPP_DBYR_ID)
```

```
#define SIPP_DBYR_LUMA_ID 18 /* Debayer luma buffer */
```

Referenced by sippBufGetObufCtx(), sippCheckOPipeConnectionDoGLTM(), and sippCheckOPipeConnectionMedian().

```
#define SIPP_DMA_ID (SIPP_MAX_ID+1 )
```

Referenced by sippAddFilterToPipe(), sippAnalysePipe2x5x(), sippComputeChunkWidths(), sippCoreSetPaddingReqs(), sippCreateFilter(), sippGenericSchCreateSEFromFilter(), and sippLinkFilter().

```
#define SIPP_DOGL_ID 4 /* Difference of Gaussians/Local tone mapping */
```

Referenced by sippCheckOPipeConnectionLuma().

```
#define SIPP_DOGL_ID_MASK (1 << SIPP_DOGL_ID)
```

```
#define SIPP_EDGE_OP_ID 12 /* Edge operator */
```

```
#define SIPP_EDGE_OP_ID_MASK (1 << SIPP_EDGE_OP_ID)
```

```
#define SIPP_FAKE_ID (EXE_NUM + 1)
```

```
#define SIPP_FILTER_FULL_LLБ_MASK
```

Value:

```
(SIPP_SIGMA_ID_MASK | \
                                     SIPP_RAW_ID_MASK | \
                                     SIPP_DBYR_ID_MASK | \
```

```
SIPP_DOGL_ID_MASK | \
SIPP_LUMA_ID_MASK | \
SIPP_SHARPEN_ID_MASK | \
SIPP_CGEN_ID_MASK | \
SIPP_MED_ID_MASK | \
SIPP_CHROMA_ID_MASK | \
SIPP_CC_ID_MASK)
```

Referenced by sippOSEProcessFilterIBufs().

```
#define SIPP_FILTER_LL_B_MASK
```

Value:

```
(SIPP_SIGMA_ID_MASK | \
SIPP_RAW_ID_MASK | \
SIPP_DBYR_ID_MASK | \
SIPP_LUMA_ID_MASK | \
SIPP_SHARPEN_ID_MASK | \
SIPP_MED_ID_MASK | \
SIPP_CHROMA_ID_MASK)
```

Referenced by sippAnalysePipe2x5x(), and sippOSEProcessFilterIBufs().

```
#define SIPP_HARRIS_ID 14 /* Harris corners */
```

```
#define SIPP_HARRIS_ID_MASK (1 << SIPP_HARRIS_ID)
```

```
#define SIPP_HW_FILTER_MASK_SIZE ((EXE_NUM+31)/32)
```

Referenced by sippHWSessionAddActiveLists(), and sippHWSessionRemoveActiveLists().

```
#define SIPP_LSC_GM_ID 20 /* Lens shading correction - gain mesh buffer */
```

```
#define SIPP_LSC_GM_ID_MASK (1 << SIPP_LSC_GM_ID)
```

```
#define SIPP_LSC_ID 1 /* Lens shading filter */
```

Referenced by sippCheckOPipeConnectionRaw().

```
#define SIPP_LSC_ID_MASK (1 << SIPP_LSC_ID)
```

```
#define SIPP_LUMA_C4LUT_ID 24 /* Luma denoise - cosine-4th law LUT buffer */
```

```
#define SIPP_LUMA_HIST_ID 25 /* RAW Luma histogram */
```

```
#define SIPP_LUMA_ID 5 /* Luma denoise in/out */
```

Referenced by sippCheckOPipeConnectionSharpen().


```
#define SIPP_LUMA_ID_MASK (1 << SIPP_LUMA_ID)

#define SIPP_LUMA_REF_ID 18 /* Luma denoise reference in */

#define SIPP_LUMA_REF_ID_MASK (1 << SIPP_LUMA_REF_ID)

#define SIPP_LUT_ID 11 /* Look-up table */
```

Referenced by sippCheckOPipeConnectionPoly().

```
#define SIPP_LUT_ID_MASK (1 << SIPP_LUT_ID)

#define SIPP_LUT_LOAD_ID 23 /* LUT filter - LUT buffer */

#define SIPP_LUT_LOAD_MASK (1 << SIPP_LUT_LOAD_ID)

#define SIPP_MAX_FILTER_ID SIPP_MIPI_RX3_ID
```

Referenced by sippAddFilterToPipe(), sippAnalysePipe2x5x(), sippDbgDumpFilterOuts(), sippGenericBlockHWUnits2x5x(), sippGenericLinePrepare(), sippGenericRuntimeLoadPipeline(), sippGenericSchedWr(), sippGenericStartUnits(), sippGetFirstHwFiltIdx(), sippHWSessionCommand(), sippIdentifyOPipeSchedulingEntity(), and sippValidatePipe().

```
#define SIPP_MAX_ID 26
```

Referenced by sippAllocCmxLineBuffers().

```
#define SIPP_MED_ID 8 /* Median */
```

Referenced by sippCheckOPipeConnectionChrDns(), and sippGenericScheduleSetBufConsModels().

```
#define SIPP_MED_ID_MASK (1 << SIPP_MED_ID)

#define SIPP_MED_LUMA_ID 21 /* Median filter - chroma median reference luma buffer */

#define SIPP_MIN_FILTER_ID SIPP_SIGMA_ID

#define SIPP_MIPI_RX0_ID 20 /* MIPI Rx[0] filter (output buffer only) */
```

Referenced by sippCheckOPipeConnectionSigma(), sippGetOBufIdsMipiRx0(), sippLoadMipiRx(), and sippSetOBufLevelsMipiRx0().

```
#define SIPP_MIPI_RX0_ID_MASK (1 << SIPP_MIPI_RX0_ID)

#define SIPP_MIPI_RX1_ID 21 /* MIPI Rx[1] filter (output buffer only) */
```

Referenced by sippCheckOPipeConnectionSigma(), sippGetOBufIdsMipiRx1(), and sippSetOBufLevelsMipiRx1().

```
#define SIPP_MIPI_RX1_ID_MASK (1 << SIPP_MIPI_RX1_ID)
```

```
#define SIPP_MIPI_RX2_ID 22 /* MIPI Rx[2] filter (output buffer only) */
```

Referenced by sippCheckOPipeConnectionSigma(), sippGetOBufIdsMipiRx2(), and sippSetOBufLevelsMipiRx2().

```
#define SIPP_MIPI_RX2_ID_MASK (1 << SIPP_MIPI_RX2_ID)
```

```
#define SIPP_MIPI_RX3_ID 23 /* MIPI Rx[3] filter (output buffer only) */
```

Referenced by sippCheckOPipeConnectionSigma(), sippGetOBufIdsMipiRx3(), and sippSetOBufLevelsMipiRx3().

```
#define SIPP_MIPI_RX3_ID_MASK (1 << SIPP_MIPI_RX3_ID)
```

```
#define SIPP_MIPI_TX0_ID 18 /* MIPI Tx[0] filter (input buffer only) */
```

Referenced by sippGetOBufIdsMipiTx0(), and sippLoadMipiTx().

```
#define SIPP_MIPI_TX0_ID_MASK (1 << SIPP_MIPI_TX0_ID)
```

Referenced by sippGenericWaitUnits().

```
#define SIPP_MIPI_TX1_ID 19 /* MIPI Tx[1] filter (input buffer only) */
```

Referenced by sippGetOBufIdsMipiTx1().

```
#define SIPP_MIPI_TX1_ID_MASK (1 << SIPP_MIPI_TX1_ID)
```

Referenced by sippGenericWaitUnits().

```
#define SIPP_OPIPE_ID -1 /* Convention to mark the oPipe */
```

Referenced by sippIdentifyOPipeSchedulingEntity().

```
#define SIPP_RAW_DEFECT_ID 26 /* RAW filter - defect pixel list for static defect correction */
```

```
#define SIPP_RAW_ID 2 /* RAW filter */
```

Referenced by sippCheckOPipeConnectionDbyr().

```
#define SIPP_RAW_ID_MASK (1 << SIPP_RAW_ID)
```

```
#define SIPP_RESERVED_ID 13 /* 13 - Reserved */
```

```
#define SIPP_RGB_HIST_ID 26 /* RAW RGB histogram */
```

```
#define SIPP_SHARPEN_ID 6 /* Sharpening */
```

Referenced by sippCheckOPipeConnectionColourComb().

```
#define SIPP_SHARPEN_ID_MASK (1 << SIPP_SHARPEN_ID)
```

```
#define SIPP_SIGMA_ID 0 /* Sigma denoise */
```

Referenced by sippCheckOPipeConnectionLsc(), sippGetIBufIdsSigma(), sippGetOBufIdsSigma(), sippLoadSigma(), and sippSetOBufLevelsSigma().

```
#define SIPP_SIGMA_ID_MASK (1 << SIPP_SIGMA_ID)
```

```
#define SIPP_STATS_ID 19 /* RAW statistics */
```

```
#define SIPP_STATS_MASK (1 << SIPP_STATS_ID)
```

```
#define SIPP_SVU_ID (SIPP_MAX_ID+2 )
```

Referenced by adjustNodesRecursively(), sippAddFilterToPipe(), sippComputeChunkWidthsSW(), sippCoreSetPaddingReqs(), sippCreateFilter(), sippIniHwFilters(), sippLinkFilter(), and sippLinkFilter-SetOBuf().

```
#define SIPP_UPFIRDN0_ID 15 /* Polyphase FIR filter[0] */
```

Referenced by sippGenericSchCreateSEFromFilter(), and sippGenericScheduleSetBufConsModels().

```
#define SIPP_UPFIRDN0_ID_MASK (1 << SIPP_UPFIRDN0_ID)
```

```
#define SIPP_UPFIRDN1_ID 16 /* Polyphase FIR filter[1] */
```

Referenced by sippAddFilterToPipe(), sippGenericSchCreateSEFromFilter(), and sippGenericSchedule-SetBufConsModels().

```
#define SIPP_UPFIRDN1_ID_MASK (1 << SIPP_UPFIRDN1_ID)
```

```
#define SIPP_UPFIRDN2_ID 17 /* Polyphase FIR filter[2] */
```

Referenced by sippAddFilterToPipe(), sippGenericSchCreateSEFromFilter(), and sippGenericSchedule-SetBufConsModels().

```
#define SIPP_UPFIRDN2_ID_MASK (1 << SIPP_UPFIRDN2_ID)
```

```
#define SIPP_V2_RESERVED_ID SIPP_V2_MAX_ID+1 /* No ID Reserved */
```

7.240 sippHwLsc.c File Reference

SIPP engine.

```
#include <sipp.h>
#include <sippInternal.h>
```

7.240.1 Detailed Description

SIPP engine.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see:
common/license.txt

7.241 sippHwLumaDns.c File Reference

SIPP engine.

```
#include <sipp.h>
#include <sippInternal.h>
```

7.241.1 Detailed Description

SIPP engine.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see:
common/license.txt

7.242 sippHwLut.c File Reference

SIPP engine.

```
#include <sipp.h>
#include <sippInternal.h>
```

7.242.1 Detailed Description

SIPP engine.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see:
common/license.txt

7.243 sippHwMedian.c File Reference

SIPP engine.

```
#include <sipp.h>
#include <sippInternal.h>
```

7.243.1 Detailed Description

SIPP engine.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.244 sippHwMipiRx.c File Reference

SIPP engine.

```
#include <sipp.h>
#include <sippInternal.h>
```

Functions

- [u32 sippInitMipiRx \(SippFilter *fptr\)](#)
- [void sippLoadMipiRx \(SippFilter *fptr\)](#)
- [void sippSetBufLatenciesMipiRx \(SippFilter *fptr\)](#)
- [u32 sippGetOBufIdsMipiRx0 \(SippFilter *fptr, u32 oBufIdx\)](#)
- [u32 sippGetOBufIdsMipiRx1 \(SippFilter *fptr, u32 oBufIdx\)](#)
- [u32 sippGetOBufIdsMipiRx2 \(SippFilter *fptr, u32 oBufIdx\)](#)
- [u32 sippGetOBufIdsMipiRx3 \(SippFilter *fptr, u32 oBufIdx\)](#)
- [void sippSetOBufLevelsMipiRx0 \(SippFilter *fptr, eSippObufControl eOBufSetting\)](#)
- [void sippSetOBufLevelsMipiRx1 \(SippFilter *fptr, eSippObufControl eOBufSetting\)](#)
- [void sippSetOBufLevelsMipiRx2 \(SippFilter *fptr, eSippObufControl eOBufSetting\)](#)
- [void sippSetOBufLevelsMipiRx3 \(SippFilter *fptr, eSippObufControl eOBufSetting\)](#)

7.244.1 Detailed Description

SIPP engine.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.244.2 Function Documentation

u32 sippGetOBufIdsMipiRx0 (**SippFilter** * fptr, **u32** oBufIdx)

u32 sippGetOBufIdsMipiRx1 (**SippFilter** * fptr, **u32** oBufIdx)

u32 sippGetOBufIdsMipiRx2 (**SippFilter** * fptr, **u32** oBufIdx)

u32 sippGetOBufIdsMipiRx3 (**SippFilter** * fptr, **u32** oBufIdx)

u32 sippInitMipiRx (**SippFilter** * fptr)

void sippLoadMipiRx (**SippFilter** * fptr)

void sippSetBufLatenciesMipiRx (**SippFilter** * fptr)

void sippSetOBufLevelsMipiRx0 (**SippFilter** * fptr, **eSippObufControl** eOBufSetting)

void sippSetOBufLevelsMipiRx1 (**SippFilter** * fptr, **eSippObufControl** eOBufSetting)

void sippSetOBufLevelsMipiRx2 (**SippFilter** * fptr, **eSippObufControl** eOBufSetting)

void sippSetOBufLevelsMipiRx3 (**SippFilter** * fptr, **eSippObufControl** eOBufSetting)

7.245 sippHwMipiTx.c File Reference

SIPP engine.

```
#include <sipp.h>
#include <sippInternal.h>
```

Functions

- **u32** sippInitMipiTx (SippFilter *fptr)
- void sippLoadMipiTx (SippFilter *fptr)
- void sippSetBufLatenciesMipiTx (SippFilter *fptr)
- **u32** sippGetOBufIdsMipiTx0 (SippFilter *fptr, **u32** oBufIdx)
- **u32** sippGetOBufIdsMipiTx1 (SippFilter *fptr, **u32** oBufIdx)
- void sippSetOBufLevelsMipiTx0 (SippFilter *fptr, **eSippObufControl** eOBufSetting)
- void sippSetOBufLevelsMipiTx1 (SippFilter *fptr, **eSippObufControl** eOBufSetting)

7.245.1 Detailed Description

SIPP engine.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.245.2 Function Documentation

u32 sippGetOBufIdsMipiTx0 (**SippFilter** * fptr, **u32** oBufIdx)

u32 sippGetOBufIdsMipiTx1 (**SippFilter** * fptr, **u32** oBufIdx)

u32 sippInitMipiTx (**SippFilter** * fptr)

void sippLoadMipiTx (**SippFilter** * fptr)

void sippSetBufLatenciesMipiTx (**SippFilter** * fptr)

void sippSetOBufLevelsMipiTx0 (**SippFilter** * fptr, **eSippObufControl** eOBufSetting)

void sippSetOBufLevelsMipiTx1 (**SippFilter** * fptr, **eSippObufControl** eOBufSetting)

7.246 sippHwPolyFir.c File Reference

SIPP engine.

```
#include <sipp.h>
#include <sippInternal.h>
```

7.246.1 Detailed Description

SIPP engine.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.247 sippHwRaw.c File Reference

SIPP engine.

```
#include <sipp.h>
#include <sippInternal.h>
```

7.247.1 Detailed Description

SIPP engine.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.248 sippHWSessionControl.c File Reference

Establishes context for and makes calls to sipp HW and shaves This file provides the hardware facing aspect to the interface. It is part of a set with [sippPipeSessionControl.c](#) which provides pipeline functionality Platform(s) supported : ma2x5x.

```
#include <sipp.h>
#include <sippInternal.h>
```

Functions

- void [sippHWSessionInit](#) ()
- void [sippHWSessionAddActiveLists](#) (pSippPipeline pPipe, u32 uHWPipeID)
- void [sippHWSessionRemoveActiveLists](#) (pSippPipeline pPipe, u32 uHWPipeID)
- void [sippHWSessionRemoveLoadedPipe](#) (pSippPipeline pPipe)
- void [sippHWSessionCommand](#) (pSippPipeline pPipe, u32 uHWPipeID, eSIPP_ACCESS_SCHEDULER_EVENT eEvent, SIPP_ACCESS_SCHEDULER_EVENT_DATA pData)

Variables

- pSIPP_HW_SESSION pgSippHW
- u32 sippGlobalOBFLIncStatus

7.248.1 Detailed Description

Establishes context for and makes calls to sipp HW and shaves This file provides the hardware facing aspect to the interface. It is part of a set with [sippPipeSessionControl.c](#) which provides pipeline functionality Platform(s) supported : ma2x5x.

Copyright

All code copyright Movidius Ltd 2015, all rights reserved. For License Warranty see: common/license.txt

7.248.2 Function Documentation

```
void sippHWSessionAddActiveLists ( pSippPipeline pPipe, u32 uHWPipeID )
```

```
void sippHWSessionCommand ( pSippPipeline pPipe, u32 uHWPipeID, eSIPP_ACCESS-
_SCHEDULER_EVENT eEvent, SIPP_ACCESS_SCHEDULER_EVENT_DATA pData
)
```

Referenced by [sippPipeSessionControl\(\)](#).

```
void sippHWSessionInit ( void )
```

Referenced by [sippPipeSessionControl\(\)](#).


```
void sippHWSessionRemoveActiveLists ( pSippPipeline pPipe, u32 uHWPipeID )
```

Referenced by sippPipeSessionControl().

```
void sippHWSessionRemoveLoadedPipe ( pSippPipeline pPipe )
```

Referenced by sippGenericRunIterDone().

7.248.3 Variable Documentation

pSIPP_HW_SESSION pgSippHW

u32 sippGlobalOBFLIncStatus

7.249 sippHwSigma.c File Reference

SIPP engine.

```
#include <sipp.h>
#include <sippInternal.h>
```

Functions

- **u32** sippInitSigma (**SippFilter** *fptr)
- void sippLoadSigma (**SippFilter** *fptr)
- **u32** sippGetIBufCtxSigma (**SippFilter** *fptr, **u32** iBufIdx)
- **u32** sippGetIBufIdsSigma (**SippFilter** *fptr, **u32** iBufIdx)
- **u32** sippGetOBufIdsSigma (**SippFilter** *fptr, **u32** oBufIdx)
- void sippSetBufLatenciesSigma (**SippFilter** *fptr)
- void sippSetOBufLevelsSigma (**SippFilter** *fptr, **eSippOBufControl** eOBufSetting)

7.249.1 Detailed Description

SIPP engine.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.249.2 Function Documentation

u32 sippGetIBufCtxSigma (**SippFilter** * fptr, **u32** iBufIdx)

u32 sippGetIBufIdsSigma (**SippFilter** * fptr, **u32** iBufIdx)

u32 sippGetOBufIdsSigma (**SippFilter** * fptr, **u32** oBufIdx)

```
u32 sippInitSigma ( SippFilter * fptr )
```

```
void sippLoadSigma ( SippFilter * fptr )
```

```
void sippSetBufLatenciesSigma ( SippFilter * fptr )
```

```
void sippSetOBufLevelsSigma ( SippFilter * fptr, eSippObufControl eOBufSetting )
```

7.250 sippHwUnsharp.c File Reference

SIPP engine.

```
#include <sipp.h>
#include <sippInternal.h>
```

7.250.1 Detailed Description

SIPP engine.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.251 sippInternal.h File Reference

SIPP core : internal definitions.

```
#include <sippDefines.h>
#include <sippTypesPrivate.h>
#include <sippSessionControl.h>
#include <sippPal.h>
#include <sippAccessScheduler.h>
```

Functions

- void [sippHWInit](#) (void)
- bool [sippSWInit](#) (void)
- void [sippInitPipeline](#) (ptSippPipelineSuper ptSPipe, u32 sliceFirst, u32 sliceLast, u32 sliceSize, u8 *mbinImg)
- void [sippAddFilterToPipe](#) (pSippPipeline pPipe, pSippFilter pFilter, u32 flags, u32 out_W, u32 out_H, u32 num_pl, u32 bpp, FnSvuRun funcSvuRun, const char *name)
- eSIPP_STATUS [sippElaboratePipeline](#) (pSippPipeline pPipe)
- eSIPP_STATUS [sippTermInternal](#) (void)
- eSIPP_STATUS [sippFreePipeResource](#) (pSippPipeline pPipe)
- eSIPP_STATUS [sippRescheduleRequest](#) (pSippPipeline pPipe)
- eSIPP_STATUS [sippRunItersRequest](#) (pSippPipeline pPipe, u32 uNumIters)

- `eSIPP_STATUS sippResetFilterVariables (pSippPipeline pPipe)`
- `void sippCallbackInit (void)`
- `eSIPP_STATUS sippIssueCommand (pSippPipeline pPipe, eSippCommand eSippCmd, SippCommandData pCmdData)`
- `void sippEventNotify (pSippPipeline pPipe, eSIPP_PIPELINE_EVENT eEvent, SIPP_PIPELINE_EVENT_DATA *pData)`
- `void sippHWSessionInit (void)`
- `void sippHWSessionCommand (pSippPipeline pPipe, u32 uHWPipeID, eSIPP_ACCESS_SCHEDULER_EVENT eEvent, SIPP_ACCESS_SCHEDULER_EVENT_DATA pData)`
- `void sippHWSessionRemoveActiveLists (pSippPipeline pPipe, u32 uHWPipeID)`
- `void sippHWSessionRemoveLoadedPipe (pSippPipeline pPipe)`
- `void sippPipeSessionControlInit (void)`
- `void sippPipeSessionControl (eSIPP_ACCESS_SCHEDULER_EVENT eEvent, SIPP_ACCESS_SCHEDULER_EVENT_DATA pData, u32 uPipeIdx, u32 uHWPipeIdx)`
- `eSIPP_STATUS sippCoreFinalisePipeline (pSippPipeline pPipe)`
- `eSIPP_STATUS sippCoreResourceInit (void)`
- `eSIPP_STATUS sippCoreReschedulePipeline (pSippPipeline pPipe)`
- `bool sippValidatePipe (pSippPipeline pPipe)`
- `void sippCoreHwInitialLoad (pSippPipeline pPipe)`
- `void sippCoreHwInitialSave (pSippPipeline pPipe)`
- `void sippComputeSliceLayout (pSippPipeline pPipe)`
- `int sippUsingPrecompSched (pSippPipeline pPipe)`
- `void sippGetFirstHwFiltIdx (pSippPipeline pPipe)`
- `u32 sippIniHwFilters (pSippPipeline pPipe)`
- `u32 sippBuildLnBufs (pSippPipeline pPipe)`
- `void sippComputePaddingOffsets (pSippPipeline pPipe)`
- `void sippAsmOptSetup (pSippPipeline pPipe)`
- `void sippComputeSwOutCt (pSippPipeline pPipe)`
- `void sippIncrementOutBufs (pSippPipeline pPipe)`
- `void sippComputeBufferProps (pSippPipeline pPipe)`
- `void sippInitBufferLnPointers (pSippPipeline pPipe)`
- `void sippUtilOrderPixels (u32 bayerPattern, u32 inGr, u32 inR, u32 inB, u32 inGb, u32 *out)`
- `void sippListSort (s32 *pnList, s32 *pnSortIndices, u32 uSize, u8 descending)`
- `s32 sippFindInList (void *pElement, void **ppList, u32 numList)`
- `void sippCQInit (SippPipeline *pPipe)`
- `void sippDbgPrintNumPar (SippFilter *filters[], u32 nFilters)`
- `void sippDbgShowBuffPtr (SippFilter *fptr, const char *msg)`
- `void sippDbgDumpRunMask (u32 mask, int iteration, int dbgDump)`
- `void sippDbgFrameCheck (SippPipeline *pl)`
- `void sippDbgDumpSchedForVcs (SippPipeline *pl)`
- `void sippDbgDumpSchedForVcsCArr (SippPipeline *pl)`
- `void sippDbgDumpGraph (SippPipeline *pl, const char *fname)`
- `void sippDbgCreateDumpFiles (SippPipeline *pl)`
- `void sippDbgDumpFilterOuts (SippPipeline *pl)`
- `void sippDbgDumpAsmOffsets (SippPipeline *pl)`
- `void sippDumpHtmlMap (SippPipeline *pl)`
- `void sippPrintSliceWidth (SippPipeline *pl)`

- void `sippErrorInit` (void)
- void `sippAssert` (u32 condition, u32 errCode)
- u32 `sippGenericSchedule` (SippPipeline *pl, bool allocMem, bool reschedPipe)
- void `sippGenericScheduleSetBufConsModels` (pSippPipeline pipeLine)
- float `sippGetCoord2` (s32 in, float factor, float centreIn, float centreOut)
- void `sippGenericSchedWr` (SippPipeline *pl, u32 iteration)
- void `sippGenericDbgPrintRunnable` (SippSchEnt *SEs[], u32 nSE, u32 iteration)
- void `sippGenericDbgShowBufferReq` (SippSchEnt *SEs[], u32 nSE)
- void `sippGenericDbgDumpBuffState` (SippFilter *filters[], u32 nFilters, u32 iteration)
- void `ctxSwitchOnePar` (SippFilter *fptr, bool bSave, u32 unitID)
- void `ctxSwitchTwoPar` (SippFilter *fptr, bool bSave, u32 unitID)
- void `ctxSwitchChromaDns` (SippFilter *fptr, bool bSave, u32 unitID)
- void `ctxSwitchLut` (SippFilter *newF, SippFilter *oldF, u32 unitID)
- void `ctxSwitchPoly` (SippFilter *fptr, bool bSave, u32 unitID)
- void `ctxSwitchColComb` (SippFilter *newF, SippFilter *oldF, u32 unitID)
- void `ctxSwitchMipiRx` (SippFilter *newF, SippFilter *oldF, u32 unitID)
- void `ctxSwitchMipiTx` (SippFilter *fptr, bool bSave, u32 unitID)
- void `sippConfigSvus` (SippPipeline *pl)
- void `sippDataSectAction` (SippPipeline *pl, u64 action)
- void `sippChainDmaDesc` (SippPipeline *pl)
- void `sippSetupSvus` (SippPipeline *pl)
- void `sippInitSyncMutexes` (SippPipeline *pl)
- void `sippKickSvus` (SippPipeline *pl)
- void `sippGetCtxOrder` (SippPipeline *pl)
- void `sippComputeHwCtxChg` (SippPipeline *pl)
- void `sippHandleCtxSwitch` (SippPipeline *pl, int doLoop)
- void `sippGenericRuntimeFrameReset` (pSippPipeline pPipe)
- void `sippGenericRuntimeClaimHWResource` (pSippPipeline pPipe)
- void `sippGenericRuntime` (pSippPipeline pPipe, eSIPP_ACCESS_SCHEDULER_EVENT e-Event, SIPP_ACCESS_SCHEDULER_EVENT_DATA pData)
- void `sippGenericRuntimeHWProcessIters` (pSippPipeline pPipe, u32 numIters)
- u32 `sippGenericRunIterDone` (pSippPipeline pPipe)
- void `sippGenericRunNextIter` (pSippPipeline pPipe)
- void `sippGenericUpdateHWUnits2x5x` (SippPipeline *pPipe)
- void `sippGenericStartUnits` (SippPipeline *pl)
- void `sippGenericLinePrepare` (SippPipeline *pl, int iteration)
- void `sippGenericUpdateExecNums` (SippPipeline *pl)
- void `sippIbflDecHandler` (u32 irqSource)
- void `sippObflIncHandler` (u32 irqSource)
- void `sippSvuDoneIrqHandler` (u32 irqSource)
- void `sippCmxDmaDoneIrqHandler` (void)
- void `sippCheckIterComplete` (SippPipeline *pPipe, u32 Flag)
- void `sippIsrSetup` (void)
- void `sippIntBarrierSetup` (u32 useIntBar)
- void * `sippMemAlloc` (ptSippMCB pSippMCB, SippVirtualPool vPool, s32 n_bytes)
- void `sippInitLnMemPool` (ptSippMCB ptMCB, u8 *start)
- //

- void sippInitPhysicalPoolGlobal (void)
- void sippMemInitVirtPhysMaps (void)
- s32 sippAllocCmxLineBuffers (SippPipeline *pipe)
- s32 sippAllocCmxLineBuffersOPipe (SippPipeline *pipe)
- u8 sippMemLBMatchRegionsToChunks (pSippCmxBufferMap pCmxMap, u32 chunkStride, u32 numChunks)
- void sippMemLBConsolidateRegions (pSippCmxBufferMap pCmxMap)
- void sippInitSchedPool (ptSippMCB ptMCB, u32 sliceFirst, u32 sliceLast)
- void sippMemFreeList (ptSippMCB pSippMCB, SippVirtualPool vPool)
- void sippMemFree (ptSippMCB pSippMCB, SippVirtualPool vPool, void *pPtr)
- u32 sippInitPhysicalPoolPipe (ptSippMCB ptMCB)
- u32 sippInitLnMemPoolSlices (ptSippMCB ptMCB, u32 firstSlice, u32 lastSlice)
- SippVirtualPool sippMemFindMaxLnMemPoolFree (ptSippMCB pSippMCB)
- u32 sippIbufSetup (SippFilter *fptr, u32 parNo)
- u32 sippObufSetup (SippFilter *fptr, u32 oBufIdx)
- u32 sippBufSetupIrqRate (SippHwBuf *buf, u32 oBuf, u32 bufId, u32 numLinesPerIrq)
- u32 sippBufSetupIrqRateCQ (SippHwBuf *buf, u32 oBuf, u32 bufId, u32 numLinesPerIrq, ps-SippCMDQu pQu)
- u32 sippBufGetObufCtx (SippFilter *fptr, u32 oBufIdx)
- void sippCmxDmaInit (void)
- u32 sippInitDma (SippFilter *fptr)
- void sippKickDma (SippPipeline *pl)
- u32 sippWaitDma (void)
- void sippKickDmaCQ (SippPipeline *pl, u32 iteration)
- void sippRunDmaCQ (SippPipeline *pl)
- u32 sippDmaCQInit (SippPipeline *pPipe)
- void topLevelCmxDmaIrqHandler (u32 irqSource)
- void sippCmxDmaInitAsync (void)
- void dmaKickSequenceConcurrent (SippPipeline *pl)
- void sippKickShaveM1PC (SippPipeline *pl)
- void sippWaitShave (SippPipeline *pl)
- void sippStopSvus (SippPipeline *pl)
- SippPipeline * SVU_SYM (sipp_pl)
- tsSippHeap * sippHeapCreate (u8 *sippHeap_start, u32 sippHeap_size)
- void * sippAlloc (tsSippHeap *pSippHeap, void *pPrev, u32 size)
- void sippFreeList (tsSippHeap *pSippHeap, void *pStartPtr)
- void sippFree (tsSippHeap *pSippHeap, void *pPtr)
- u32 sippHeapCheck (tsSippHeap *pSippHeap)

7.251.1 Detailed Description

SIPP core : internal definitions.

Copyright

All code copyright Movidius Ltd 2015, all rights reserved. For License Warranty see: common/license.txt

7.251.2 Function Documentation

`void ctxSwitchChromaDns (SippFilter * fptr, bool bSave, u32 unitID)`

`void ctxSwitchColComb (SippFilter * newF, SippFilter * oldF, u32 unitID)`

`void ctxSwitchLut (SippFilter * newF, SippFilter * oldF, u32 unitID)`

`void ctxSwitchMipiRx (SippFilter * newF, SippFilter * oldF, u32 unitID)`

`void ctxSwitchMipiTx (SippFilter * fptr, bool bSave, u32 unitID)`

`void ctxSwitchOnePar (SippFilter * fptr, bool bSave, u32 unitID)`

`void ctxSwitchPoly (SippFilter * fptr, bool bSave, u32 unitID)`

`void ctxSwitchTwoPar (SippFilter * fptr, bool bSave, u32 unitID)`

`void dmaKickSequenceConcurrent (SippPipeline * pl)`

Referenced by `sippKickDma()`, and `sippRunDmaCQ()`.

`void sippAddFilterToPipe (pSippPipeline pPipe, pSippFilter pFilter, u32 flags, u32 out_W, u32 out_H, u32 num_pl, u32 bpp, FnSvuRun funcSvuRun, const char * name)`

Referenced by `sippCreateFilter()`.

`void* sippAlloc (tsSippHeap * pSippHeap, void * pPrev, u32 size)`

Referenced by `sippMemAlloc()`.

`s32 sippAllocCmxLineBuffers (SippPipeline * pipe)`

`s32 sippAllocCmxLineBuffersOPipe (SippPipeline * pipe)`

`void sippAsmOptSetup (pSippPipeline pPipe)`

`void sippAssert (u32 condition, u32 errCode)`

Referenced by `svuBoxFilter()`.

`u32 sippBufGetObufCtx (SippFilter * fptr, u32 oBufIdx)`

`u32 sippBufSetupIrqRate (SippHwBuf * buf, u32 oBuf, u32 bufId, u32 numLinesPerIrq)`

`u32 sippBufSetupIrqRateCQ (SippHwBuf * buf, u32 oBuf, u32 bufId, u32 numLinesPerIrq, psSippCMDQu pQu)`

`u32 sippBuildLnBufs (pSippPipeline pPipe)`

`void sippCallbackInit (void)`

Referenced by `sippSWInit()`.

`void sippChainDmaDesc (SippPipeline * pl)`

Referenced by `sippCoreFinalisePipeline()`, and `sippCoreReschedulePipeline()`.

`void sippCheckIterComplete (SippPipeline * pPipe, u32 Flag)`

Referenced by `sippCmxDmaDoneIrqHandler()`, `sippObflIncHandler()`, and `sippSvuDoneIrqHandler()`.

`void sippCmxDmaDoneIrqHandler (void)`

Referenced by `topLevelCmxDmaIrqHandler()`.

`void sippCmxDmaInit (void)`

Referenced by `sippHWSessionInit()`.

`void sippCmxDmaInitAsync (void)`

Referenced by `sippInitDma()`.

`void sippComputeBufferProps (pSippPipeline pPipe)`

`void sippComputeHwCtxChg (SippPipeline * pl)`

`void sippComputePaddingOffsets (pSippPipeline pPipe)`

`void sippComputeSliceLayout (pSippPipeline pPipe)`

`void sippComputeSwOutCt (pSippPipeline pPipe)`

`void sippConfigSvus (SippPipeline * pl)`

`eSIPP_STATUS sippCoreFinalisePipeline (pSippPipeline pPipe)`

Referenced by `sippPipeSessionControl()`.

`void sippCoreHwInitialLoad (pSippPipeline pPipe)`

`void sippCoreHwInitialSave (pSippPipeline pPipe)`

`eSIPP_STATUS sippCoreReschedulePipeline (pSippPipeline pPipe)`

Referenced by `sippPipeSessionControl()`.

eSIPP_STATUS sippCoreResourceInit (void)

Referenced by sippPipeSessionControlInit().

void sippCQInit (**SippPipeline** * pPipe)

Referenced by sippCoreFinalisePipeline().

void sippDataSectAction (**SippPipeline** * pl, **u64** action)

void sippDbgCreateDumpFiles (**SippPipeline** * pl)

Referenced by sippGenericRuntimeFrameReset().

void sippDbgDumpAsmOffsets (**SippPipeline** * pl)

Referenced by sippCoreFinalisePipeline(), and sippCoreReschedulePipeline().

void sippDbgDumpFilterOuts (**SippPipeline** * pl)

Referenced by sippGenericRunIterDone(), and sippGenericWaitUnits().

void sippDbgDumpGraph (**SippPipeline** * pl, const char * fname)

Referenced by sippCoreFinalisePipeline().

void sippDbgDumpRunMask (**u32** mask, int iteration, int dbgDump)

Referenced by sippGenericLinePrepare().

void sippDbgDumpSchedForVcs (**SippPipeline** * pl)

void sippDbgDumpSchedForVcsCArr (**SippPipeline** * pl)

Referenced by sippCoreFinalisePipeline(), and sippCoreReschedulePipeline().

void sippDbgFrameCheck (**SippPipeline** * pl)

void sippDbgPrintNumPar (**SippFilter** * filters[], **u32** nFilters)

void sippDbgShowBuffPtr (**SippFilter** * fptr, const char * msg)

u32 sippDmaCQInit (**SippPipeline** * pPipe)

Referenced by sippCoreFinalisePipeline().

`void sippDumpHtmlMap (SippPipeline * pl)`

Referenced by `sippCoreFinalisePipeline()`.

`eSIPP_STATUS sippElaboratePipeline (pSippPipeline pPipe)`

Referenced by `sippFinalizePipeline()`, `sippProcessFrame()`, and `sippProcessFrameNB()`.

`void sippErrorInit (void)`

Referenced by `sippSWInit()`.

`void sippEventNotify (pSippPipeline pPipe, eSIPP_PIPELINE_EVENT eEvent, SIPP_PIPELINE_EVENT_DATA * pData)`

Referenced by `sippCoreFinalisePipeline()`, `sippCoreReschedulePipeline()`, `sippGenericRunIterDone()`, and `sippGenericRuntimeProcessIters()`.

`s32 sippFindInList (void * pElement, void ** ppList, u32 numList)`

`void sippFree (tsSippHeap * pSippHeap, void * pPtr)`

Referenced by `sippFreeList()`.

`void sippFreeList (tsSippHeap * pSippHeap, void * pStartPtr)`

Referenced by `sippMemFree()`, and `sippMemFreeList()`.

`eSIPP_STATUS sippFreePipeResource (pSippPipeline pPipe)`

Referenced by `sippDeletePipeline()`.

`void sippGenericDbgDumpBuffState (SippFilter * filters[], u32 nFilters, u32 iteration)`

`void sippGenericDbgPrintRunnable (SippSchEnt * SEs[], u32 nSE, u32 iteration)`

`void sippGenericDbgShowBufferReq (SippSchEnt * SEs[], u32 nSE)`

`void sippGenericLinePrepare (SippPipeline * pl, int iteration)`

Referenced by `sippGenericRunNextIter()`, `sippGenericRuntimeHWProcessIters()`, and `sippGenericRuntimeProcessIters()`.

`u32 sippGenericRunIterDone (pSippPipeline pPipe)`

`void sippGenericRunNextIter (pSippPipeline pPipe)`

Referenced by `sippGenericRuntimeProcessIters()`.

```
void sippGenericRuntime ( pSippPipeline pPipe, eSIPP_ACCESS_SCHEDULER_EVENT eEvent,
SIPP_ACCESS_SCHEDULER_EVENT_DATA pData )
```

```
void sippGenericRuntimeClaimHWResource ( pSippPipeline pPipe )
```

```
void sippGenericRuntimeFrameReset ( pSippPipeline pPipe )
```

```
void sippGenericRuntimeHWProcessIters ( pSippPipeline pPipe, u32 numIters )
```

Referenced by sippGenericRuntimeProcessIters().

```
u32 sippGenericSchedule ( SippPipeline * pl, bool allocMem, bool reschedPipe )
```

```
void sippGenericScheduleSetBufConsModels ( pSippPipeline pipeLine )
```

```
void sippGenericSchedWr ( SippPipeline * pl, u32 iteration )
```

```
void sippGenericStartUnits ( SippPipeline * pl )
```

Referenced by sippGenericRunNextIter(), and sippGenericRuntimeHWProcessIters().

```
void sippGenericUpdateExecNums ( SippPipeline * pl )
```

Referenced by sippGenericRunIterDone(), and sippGenericWaitUnits().

```
void sippGenericUpdateHWUnits2x5x ( SippPipeline * pPipe )
```

Referenced by sippGenericRunIterDone(), and sippGenericRuntimeHWProcessIters().

```
float sippGetCoord2 ( s32 in, float factor, float centreIn, float centreOut )
```

Referenced by askResizer(), and askResizerLatency().

```
void sippGetCtxOrder ( SippPipeline * pl )
```

Referenced by sippCoreFinalisePipeline().

```
void sippGetFirstHwFiltIdx ( pSippPipeline pPipe )
```

```
void sippHandleCtxSwitch ( SippPipeline * pl, int doLoop )
```

```
u32 sippHeapCheck ( tsSippHeap * pSippHeap )
```

Referenced by sippMemCheck(), and sippMemStatus().

```
tsSippHeap* sippHeapCreate ( u8 * sippHeap_start, u32 sippHeap_size )
```

Referenced by sippInitPhysicalPoolGlobal().

```
void sippHWInit ( void )
```

Referenced by sippInitialize().

```
void sippHWSessionCommand ( pSippPipeline pPipe, u32 uHWPipeID, eSIPP_ACCESS-  
_SCHEDULER_EVENT eEvent, SIPP_ACCESS_SCHEDULER_EVENT_DATA pData  
)
```

Referenced by sippPipeSessionControl().

```
void sippHWSessionInit ( void )
```

Referenced by sippPipeSessionControl().

```
void sippHWSessionRemoveActiveLists ( pSippPipeline pPipe, u32 uHWPipeID )
```

Referenced by sippPipeSessionControl().

```
void sippHWSessionRemoveLoadedPipe ( pSippPipeline pPipe )
```

Referenced by sippGenericRunIterDone().

```
void sippIbflDecHandler ( u32 irqSource )
```

Referenced by sippIsrSetup().

```
u32 sippIbufSetup ( SippFilter * fptr, u32 parNo )
```

Referenced by sippInitMipiTx(), and sippInitSigma().

```
void sippIncrementOutBufs ( pSippPipeline pPipe )
```

```
u32 sippIniHwFilters ( pSippPipeline pPipe )
```

```
void sippInitBufferLnPointers ( pSippPipeline pPipe )
```

```
u32 sippInitDma ( SippFilter * fptr )
```

```
void sippInitLnMemPool ( ptSippMCB ptMCB, u8 * start )
```

```
//
```

Referenced by sippComputeSliceLayout(), and sippCoreReschedulePipeline().

```
u32 sippInitLnMemPoolSlices ( ptSippMCB ptMCB, u32 firstSlice, u32 lastSlice )
```

Referenced by sippCoreFinalisePipeline().

`void sippInitPhysicalPoolGlobal (void)`

Referenced by `sippCoreResourceInit()`.

`u32 sippInitPhysicalPoolPipe (ptSippMCB ptMCB)`

Referenced by `sippInitPipeline()`.

`void sippInitPipeline (ptSippPipelineSuper ptSPipe, u32 sliceFirst, u32 sliceLast, u32 sliceSize, u8 * mbinImg)`

Referenced by `sippCreatePipeline()`.

`void sippInitSchedPool (ptSippMCB ptMCB, u32 sliceFirst, u32 sliceLast)`

Referenced by `sippInitPipeline()`.

`void sippInitSyncMutexes (SippPipeline * pl)`

`void sippIntBarrierSetup (u32 useIntBar)`

Referenced by `sippHWSessionInit()`.

`void sippIsrSetup (void)`

Referenced by `sippDynRouteIrq()`, and `sippHWSessionInit()`.

`eSIPP_STATUS sippIssueCommand (pSippPipeline pPipe, eSippCommand eSippCmd, SippCommandData pCmdData)`

Referenced by `sippElaboratePipeline()`, `sippFreePipeResource()`, `sippHWInit()`, `sippRescheduleRequest()`, and `sippRunItersRequest()`.

`void sippKickDma (SippPipeline * pl)`

Referenced by `sippGenericStartUnits()`.

`void sippKickDmaCQ (SippPipeline * pl, u32 iteration)`

Referenced by `sippGenericLinePrepare()`.

`void sippKickShaveM1PC (SippPipeline * pl)`

Referenced by `sippGenericStartUnits()`.

```
void sippKickSvus ( SippPipeline * pl )
```

```
void sippListSort ( s32 * pnList, s32 * pnSortIndices, u32 uSize, u8 descending )
```

Referenced by sippAllocCmxLineBuffers(), sippAllocCmxLineBuffersOPipe(), sippIdentifyOPipeSchedulingEntity(), and sippOSEComplete().

```
void* sippMemAlloc ( ptSippMCB pSippMCB, SippVirtualPool vPool, s32 n_bytes )
```

Referenced by createTripleConv3x3(), sippAllocCmxLineBuffers(), sippAllocCmxLineBuffersOPipe(), sippAssignCmxMemRegion(), sippComputeBufferProps(), sippComputePaddingOffsets(), sippComputeSliceLayout(), sippComputeSwOutCt(), sippCoreFinalisePipeline(), sippCQInit(), sippCreateFilter(), sippCreatePipeline(), sippDmaCQInit(), sippFilterAddOBuf(), sippGenericAllocRuntimeSched(), sippGenericSchedAllocTempStorage(), sippGenericSchedule(), sippIbufSetup(), sippInitDma(), sippInitLnMemPoolSlices(), sippInitPhysicalPoolPipe(), sippLinkFilter(), sippMapRegionToCmx(), sippOBufSetup(), and sippOSECreate().

```
SippVirtualPool sippMemFindMaxLnMemPoolFree ( ptSippMCB pSippMCB )
```

```
void sippMemFree ( ptSippMCB pSippMCB, SippVirtualPool vPool, void * pPtr )
```

Referenced by sippDeletePipeline().

```
void sippMemFreeList ( ptSippMCB pSippMCB, SippVirtualPool vPool )
```

Referenced by sippCoreReschedulePipeline(), and sippFreePipeResource().

```
void sippMemInitVirtPhysMaps ( void )
```

Referenced by sippCoreResourceInit().

```
void sippMemLBConsolidateRegions ( pSippCmxBufferMap pCmxMap )
```

Referenced by sippCoreFinalisePipeline().

```
u8 sippMemLBMatchRegionsToChunks ( pSippCmxBufferMap pCmxMap, u32 chunkStride, u32 numChunks )
```

Referenced by sippCoreFinalisePipeline().

```
void sippObflIncHandler ( u32 irqSource )
```

Referenced by sippIsrSetup().

```
u32 sippOBufSetup ( SippFilter * fptr, u32 oBufIdx )
```

Referenced by sippInitMipiRx(), and sippInitSigma().

```
void sippPipeSessionControl ( eSIPP_ACCESS_SCHEDULER_EVENT eEvent,  
SIPP_ACCESS_SCHEDULER_EVENT_DATA pData, u32 uPipeIdx, u32 uHWPipeIdx )
```

```
void sippPipeSessionControlInit ( void )
```

Referenced by sippSWInit().

```
void sippPrintSliceWidth ( SippPipeline * pl )
```

```
eSIPP_STATUS sippRescheduleRequest ( pSippPipeline pPipe )
```

Referenced by sippReschedulePipeline().

```
eSIPP_STATUS sippResetFilterVariables ( pSippPipeline pPipe )
```

Referenced by sippProcessFrame(), and sippProcessFrameNB().

```
void sippRunDmaCQ ( SippPipeline * pl )
```

Referenced by sippGenericStartUnits().

```
eSIPP_STATUS sippRunItersRequest ( pSippPipeline pPipe, u32 uNumIters )
```

Referenced by sippProcessFrame(), and sippProcessFrameNB().

```
void sippSetupSvus ( SippPipeline * pl )
```

```
void sippStopSvus ( SippPipeline * pl )
```

```
void sippSvuDoneIrqHandler ( u32 irqSource )
```

Referenced by sippKickSvus().

```
bool sippSWInit ( void )
```

Referenced by sippInitialize().

```
eSIPP_STATUS sippTermInternal ( void )
```

Referenced by sippTerm().

```
int sippUsingPrecompSched ( pSippPipeline pPipe )
```

```
void sippUtilOrderPixels ( u32 bayerPattern, u32 inGr, u32 inR, u32 inB, u32 inGb, u32 * out )
```

Referenced by sippLoadSigma().

bool sippValidatePipe (**pSippPipeline** pPipe)

Referenced by sippPipeSessionControl().

u32 sippWaitDma (void)

Referenced by sippGenericWaitUnits(), and sippKickDma().

void sippWaitShave (**SippPipeline** * pl)

SippPipeline* SVU_SYM (**sipp_pl**)

Referenced by createTripleConv3x3(), sippBuildLnBufs(), sippKickShaveM1PC(), and sippKickSvus().

void topLevelCmxDmaIrqHandler (**u32** irqSource)

Referenced by sippCmxDmaInit().

7.252 sippIoPtrs.c File Reference

SIPP engine.

```
#include <sipp.h>
#include <sippInternal.h>
```

Functions

- **int** scGetShaveNumber ()
- **u32** getOutPtr (**SippFilter** *fptr, **u32** iterNo, **u32** planeNo)
- **u32** getInPtr (**SippFilter** *fptr, **u32** parNo, **u32** iterNo, **u32** lineNo, **u32** planeNo)
- void getIn3PlanePtr (**SippFilter** *fptr, **u32** parNo, **u32** iterNo, **u32** lineNo, void *out3Ptr)
- **u32** getInPtrAbs (**SippFilter** *fptr, **u32** parNo, **u32** iterNo, **u32** lineNo, **u32** planeNo, **u32** target-Slice)
- **u32** getPlaneIoPtrs (**SippFilter** *fptr, **u32** parNo, **u32** iterNo, **u32** planeNo, void *inPtrs)

Variables

- **SippPipeline** * sipp_pl
- **u32** dbg_svu_no

7.252.1 Detailed Description

SIPP engine.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.252.2 Function Documentation

`void getIn3PlanePtr (SippFilter * fptr, u32 parNo, u32 iterNo, u32 lineNo, void * out3Ptr)`

Referenced by svuCvtColorRGBtoChromaNV12(), svuCvtColorRGBtoLumaNV12(), svuGenChromaSS(), svuGenLumaU8Fp16(), and svuMixMedian().

`u32 getInPtr (SippFilter * fptr, u32 parNo, u32 iterNo, u32 lineNo, u32 planeNo)`

Referenced by svuAbsdiff(), svuAccumulateSquare(), svuAccumulateWeighted(), svuArithmeticAdd(), svuArithmeticAddmask(), svuArithmeticSub(), svuArithmeticSubFp16ToFp16(), svuArithmeticSubmask(), svuAvg(), svuBilateral5x5(), svuBitwiseAnd(), svubitwiseAndMask(), svuBitwiseNot(), svuBitwiseOr(), svuBitwiseOrMask(), svuBitwiseXor(), svuBitwiseXorMask(), svuBoxFilter(), svuCannyEdgeDetection(), svuCensusMatching16(), svuCensusMatching32(), svuCensusMatching64(), svuCensusMatching65(), svuCensusMatchingPyr(), svuCensusMin16(), svuCensusMin64(), svuCensusMin65(), svuCensusMin7(), svuCensusTransform5x5(), svuChannelExtract(), svuChromaBlock(), svuCombDecimDemosaicAwbGains(), svuCombDecimDemosaicAwbGainsStats(), svuContrast(), svuConv11x11(), svuConv1x5Fp16ToFp16(), svuConv1x7(), svuConv1x7Fp16ToFp16(), svuConv1x9(), svuConv5x1Fp16ToFp16(), svuConv7x1(), svuConv7x1Fp16ToFp16(), svuConv7x7Fp16ToU8(), svuConv9x1(), svuConv9x9(), svuConvert16bppTo8bpp(), svuConvertF16ToU8(), svuConvertFrom12BppTo8Bpp(), svuConvertPFp16U16(), svuConvertPU16Fp16(), svuConvertU8ToF16(), svuConvertYUV400ToYUV422(), svuConvGeneric(), svuConvSeparable11x11(), svuConvSeparable11x11Fp16ToFp16(), svuConvSeparable3x3(), svuConvSeparable3x3Fp16ToFp16(), svuConvSeparable5x5(), svuConvSeparable9x9(), svuConvSeparable9x9Fp16ToFp16(), svuCopy(), svuCornerMinEigenVal(), svuCornerMinEigenValpatched(), svuCrop(), svuCropCvtPlaneMode(), svuCvtColorChromaYUVtoNV12(), svucvtColorNV21toRGB(), svuCvtColorRGBFp16ToLumaU8(), svuCvtColorRGBFp16ToUV420U8(), svuCvtColorRGBtoLuma(), svuCvtColorRGBtoUV(), svuCvtColorRGBtoUV420(), svuCvtColorRGBtoYUV422(), svuCvtColorYUV422ToRGB(), svuCvtColorYUVtoRGB(), svuDilate3x3(), svuDilate5x5(), svuDilate7x7(), svuDilateGeneric(), svudisp2depth(), svuEqualizeHist(), svuErode3x3(), svuErode5x5(), svuErode7x7(), svuExtAfStats(), svuExtStatsSatPixelsU32(), svuFast9M2(), svuFast9ScoreCv(), svuGauss(), svuGaussHx2(), svuGaussHx2_fp16(), svuGaussVx2(), svuGaussVx2_fp16(), svuGenChroma(), svuGenChromaSS(), svuGenDnsRef(), svuGenDnsRefFp16(), svuGenLuma(), svuGenNoise(), svuGenNoiseFp16(), svuGreyDesat(), svuHammingDistance(), svuHarrisResponse(), svuHistogram(), svuHistogramStat(), svuIntegralImageSqSumF32M2(), svuIntegralImageSqSumU32M2(), svuIntegralImageSumF32M2(), svuIntegralImageSumU16U32(), svuIntegralImageSumU32M2(), svuInterpolatePixelBilinear(), svuLaplacian3x3(), svuLaplacian5x5(), svuLaplacian5x5Fp16ToFp16(), svuLaplacian7x7(), svuLaplacian7x7Fp16ToFp16(), svuLocalMaxMin3x3_fp16(), svuLocalTM(), svuLowLvlCorr(), svuLumaBlur(), svuLut10to16(), svuLut10to8(), svuLut12to16(), svuLut12to8(), svuLut8to8(), svuLutP10BppU16inU8out(), svuMaxTest3x3_fp16(), svuMeanStdDev(), svuMinMaxPos(), svuMinMaxValue(), svuMinTest3x3_fp16(), svuMixMedian(), svuMonoImbalance(), svuNegative(), svuNonMax3x3Fp32(), svuNonMax3x3U8(), svuPixelPacker10b(), svuPixelUnpacker(), svuPixelUnpackerMipi10b(), svuPixelUnpackerWB(), svuPositionKernel(), svuPurpleFlare(), svuPyrDown(), svuRgbYuv444(), svuSAD11x11(), svuSAD5x5(), svuScale05BilinHV_Fp16U8(), svuScale05BilinHVFp16(), svuScharr_fp16(), svuScl05Lanc7(), svuScl2xLancHV(), svuScl2xLancV(), svuSclBilinArb(), svusLaplacian3x3Fp16ToFp16(), svuSobel(), svuSSD11x11(), svuSSD5x5(), svuSSD7x7U8ToU32(), svuSsdPointLine7x7U8U32(), svuStart-

Bicubic(), svuStatsAwbSatPixels(), svuStatsAwbSatPixelsU32(), svuSubpixelFilter(), svuThreshold(), svuThresholdBinaryRange(), svuThresholdBinaryU8(), svuThresholdFilter(), svuWhiteBalanceBayerGBRG(), and svuWhiteBalanceRGB().

u32 getInPtrAbs (**SippFilter** * fptr, **u32** parNo, **u32** iterNo, **u32** lineNo, **u32** planeNo, **u32** targetSlice)

Referenced by svuHomography(), and svuUndistortBrown().

u32 getOutPtr (**SippFilter** * fptr, **u32** iterNo, **u32** planeNo)

Referenced by svuAbsdiff(), svuAccumulateSquare(), svuAccumulateWeighted(), svuArithmeticAdd(), svuArithmeticAddmask(), svuArithmeticSub(), svuArithmeticSubFp16ToFp16(), svuArithmeticSubmask(), svuAvg(), svuBilateral5x5(), svuBitwiseAnd(), svubitwiseAndMask(), svuBitwiseNot(), svuBitwiseOr(), svuBitwiseOrMask(), svuBitwiseXor(), svuBitwiseXorMask(), svuBoxFilter(), svuCannyEdgeDetection(), svuCensusMatching16(), svuCensusMatching32(), svuCensusMatching64(), svuCensusMatching65(), svuCensusMatchingPyr(), svuCensusMin16(), svuCensusMin64(), svuCensusMin65(), svuCensusMin7(), svuCensusTransform5x5(), svuChannelExtract(), svuChromaBlock(), svuCombDecimDemosaicAwbGains(), svuCombDecimDemosaicAwbGainsStats(), svuContrast(), svuConv11x11(), svuConv1x5Fp16ToFp16(), svuConv1x7(), svuConv1x7Fp16ToFp16(), svuConv1x9(), svuConv5x1Fp16ToFp16(), svuConv7x1(), svuConv7x1Fp16ToFp16(), svuConvert16bppTo8bpp(), svuConvertF16ToU8(), svuConvertFrom12BppTo8Bpp(), svuConvertPFp16-U16(), svuConvertPU16Fp16(), svuConvertU8ToF16(), svuConvertYUV400ToYUV422(), svuConvGeneric(), svuConvSeparable11x11(), svuConvSeparable11x11Fp16ToFp16(), svuConvSeparable3x3(), svuConvSeparable3x3Fp16ToFp16(), svuConvSeparable9x9(), svuConvSeparable9x9Fp16ToFp16(), svuCopy(), svuCornerMinEigenVal(), svuCornerMinEigenValpatched(), svuCrop(), svuCropCvtPlaneMode(), svuCvtColorChromaYUVToNV12(), svucvtColorNV21toRGB(), svuCvtColorRGBfp16ToLumaU8(), svuCvtColorRGBfp16ToUV420U8(), svuCvtColorRGBtoChromaNV12(), svuCvtColorRGBtoLuma(), svuCvtColorRGBtoLumaNV12(), svuCvtColorRGBtoUV(), svuCvtColorRGBtoUV420(), svuCvtColorRGBtoYUV422(), svuCvtColorYUV422ToRGB(), svuCvtColorYUVToRGB(), svuDilate3x3(), svuDilate5x5(), svuDilate7x7(), svuDilateGeneric(), svudisp2depth(), svuEqualizeHist(), svuErode3x3(), svuErode5x5(), svuErode7x7(), svuFast9M2(), svuFast9ScoreCv(), svuGauss(), svuGaussHx2(), svuGaussHx2_fp16(), svuGaussVx2(), svuGaussVx2_fp16(), svuGenChroma(), svuGenChromaSS(), svuGenDnsRef(), svuGenDnsRefFp16(), svuGenLuma(), svuGenLumaU8Fp16(), svuGenNoise(), svuGenNoiseFp16(), svuGreyDesat(), svuHammingDistance(), svuHarrisResponse(), svuHistogramStat(), svuHomography(), svuIntegralImageSqSumF32M2(), svuIntegralImageSqSum-U32M2(), svuIntegralImageSumF32M2(), svuIntegralImageSumU16U32(), svuIntegralImageSum-U32M2(), svuInterpolatePixelBilinear(), svuLaplacian3x3(), svuLaplacian5x5(), svuLaplacian5x5-Fp16ToFp16(), svuLaplacian7x7(), svuLaplacian7x7Fp16ToFp16(), svuLocalMaxMin3x3_fp16(), svuLocalTM(), svuLowLvlCorr(), svuLumaBlur(), svuLut10to16(), svuLut10to8(), svuLut12to16(), svuLut12to8(), svuLut8to8(), svuLutP10BppU16inU8out(), svuMaxTest3x3_fp16(), svuMeanStdDev(), svuMinMaxPos(), svuMinMaxValue(), svuMinTest3x3_fp16(), svuMixMedian(), svuMonoImbalance(), svuNegative(), svuNonMax3x3Fp32(), svuNonMax3x3U8(), svuPixelPacker10b(), svuPixelUnpacker(), svuPixelUnpackerMipi10b(), svuPixelUnpackerWB(), svuPositionKernel(), svuPurpleFlare(), svuPyrDown(), svuRgbyuv444(), svuSAD11x11(), svuSAD5x5(), svuScale05BilinHV_Fp16U8(), svuScale05BilinHVFp16(), svuScharr_fp16(), svuScl05Lanc7(), svuScl2xLancHV(), svuScl2xLancV(), svuSclBilinArb(), svuSclLaplacian3x3Fp16ToFp16(), svuSobel(), svuSSD11x11(), svuSSD5x5(), svuSSD7x7U8ToU32(), svuSsdPointLine7x7U8U32(), svuSubpixelFilter(), svuThreshold(), svuThresholdBinaryRange(), svuThresholdBinaryU8(), svuThresholdFilter(), svuUndistortBrown(), svuWhiteBalanceBayerGBRG(), svuWhiteBalanceRGB(), and svuXYgen().

u32 getPlaneIoPtrs (**SippFilter** * fptr, **u32** parNo, **u32** iterNo, **u32** planeNo, void * inPtrs)

Referenced by svuBoxFilter(), svuBoxFilter11x11(), svuBoxFilter13x13(), svuBoxFilter15x15(), svuBoxFilter3x3(), svuBoxFilter5x5(), svuBoxFilter7x7(), svuBoxFilter9x9(), svuConv15x1(), svuConv1x15(), svuConv1x5(), svuConv3x3(), svuConv3x3Fp16ToFp16(), svuConv5x1(), svuConv5x5(), svuConv5x5Fp16ToFp16(), svuConv7x7(), svuConv7x7Fp16ToFp16(), svuConv7x7Fp16ToU8(), svuConv9x1(), svuConv9x9(), svuConv9x9Fp16ToFp16(), svuConvSeparable5x5(), svuConvSeparable5x5Fp16ToFp16(), svuConvSeparable7x7(), svuConvSeparable7x7Fp16ToFp16(), svuGreyDesat(), svulowLvlCorrMultiplePlanes(), svuLumaBlur(), svuPadBayer5(), svuPadBayer5Frame(), svuScale2xBilinHV_025_075_Fp16ToFp16(), svuScale2xBilinHV_025_075_U16ToU16(), svuScale2xBilinHV_Fp16U8_phase025_075(), svuScale2xBilinHV_U8ToU8_phase025_075(), svuScl05BilinHV(), svuScl05Lanc6(), svuScl2xBilinHV(), and svuScl2xLancH().

int scGetShaveNumber ()

7.252.3 Variable Documentation

u32 dbg_svu_no

SippPipeline* sipp_pl

7.253 sippIsr.c File Reference

SIPP framework API - asynchronous runtime API Platform(s) supported : ma2x5x.

```
#include <sipp.h>
#include <sippInternal.h>
```

Functions

- void **sippIsrSetup** ()
- void **sippIntBarrierSetup** (**u32** useIntBar)

Variables

- **pSIPP_HW_SESSION** pgSippHW
- **tSippFramework** gSippFramework
- **u32** sippGlobalOBFLIncStatus

7.253.1 Detailed Description

SIPP framework API - asynchronous runtime API Platform(s) supported : ma2x5x.

Copyright

All code copyright Movidius Ltd 2015, all rights reserved. For License Warranty see: common/license.txt

7.253.2 Function Documentation

`void sippIntBarrierSetup (u32 useIntBar)`

Referenced by `sippHWSessionInit()`.

`void sippIsrSetup (void)`

Referenced by `sippDynRouteIrq()`, and `sippHWSessionInit()`.

7.253.3 Variable Documentation

tSippFramework `gSippFramework`

pSIPP_HW_SESSION `pgSippHW`

u32 `sippGlobalOBFLIncStatus`

7.254 sippManagerApi.c File Reference

SIPP framework API Platform(s) supported : ma2x5x.

```
#include <sipp.h>
#include <sippInternal.h>
```

Functions

- `void sippCallbackInit (void)`
- `eSIPP_STATUS sippIssueCommand (pSippPipeline pPipe, eSippCommand eSippCmd, Sipp-CommandData pCmdData)`
- `void sippEventNotify (pSippPipeline pPipe, eSIPP_PIPELINE_EVENT eEvent, SIPP_PIPELINE_EVENT_DATA *pData)`

Variables

- `tSippFramework gSippFramework`

7.254.1 Detailed Description

SIPP framework API Platform(s) supported : ma2x5x.

Copyright

All code copyright Movidius Ltd 2015, all rights reserved. For License Warranty see: `common/license.txt`

7.254.2 Function Documentation

`void sippCallbackInit (void)`

Referenced by `sippSWInit()`.

`void sippEventNotify (pSippPipeline pPipe, eSIPP_PIPELINE_EVENT eEvent, SIPP_PIPELINE_EVENT_DATA * pData)`

Referenced by `sippCoreFinalisePipeline()`, `sippCoreReschedulePipeline()`, `sippGenericRunIterDone()`, and `sippGenericRuntimeProcessIters()`.

`eSIPP_STATUS sippIssueCommand (pSippPipeline pPipe, eSippCommand eSippCmd, SippCommandData pCmdData)`

Referenced by `sippElaboratePipeline()`, `sippFreePipeResource()`, `sippHWInit()`, `sippRescheduleRequest()`, and `sippRunItersRequest()`.

7.254.3 Variable Documentation

`tSippFramework gSippFramework`

7.255 sippMem.c File Reference

SIPP engine.

```
#include <sipp.h>
#include <sippInternal.h>
```

Macros

- #define `SIPP_MEMPOOL_CMX` 0
- #define `SIPP_MEMPOOL_DDR` 1
- #define `SIPP_MEMPOOL_NUM_GENERAL_POOLS` 2
- #define `SIPP_MEMPOOL_TOTAL_POOLS` (`SIPP_MEMPOOL_NUM_GENERAL_POOLS` + (2 * `SIPP_MAX_SUPPORTED_PIPELINES`))

Functions

- `u8` `sippMemPoolCMX`[`SIPP_CMX_POOL_SZ`] `ALIGNED` (32)
- void `sippInitPhysicalPoolGlobal` ()
- `u32` `sippInitPhysicalPoolPipe` (`ptSippMCB` `ptMCB`)
- void `sippInitLnMemPool` (`ptSippMCB` `ptMCB`, `u8` *start)
- //
- `u32` `sippInitLnMemPoolSlices` (`ptSippMCB` `ptMCB`, `u32` firstSlice, `u32` lastSlice)
- void `sippInitSchedPool` (`ptSippMCB` `ptMCB`, `u32` sliceFirst, `u32` sliceLast)
- void `sippMemInitVirtPhysMaps` ()

- void `sippChooseMemPool` (ptSippMCB pSippMCB, SippVirtualPool vPool, u32 physPoolIdx)
- void * `sippMemAlloc` (ptSippMCB pSippMCB, SippVirtualPool vPool, s32 bytes)
- void `sippMemFreeList` (ptSippMCB pSippMCB, SippVirtualPool vPool)
- void `sippMemFree` (ptSippMCB pSippMCB, SippVirtualPool vPool, void *pPtr)
- SippVirtualPool `sippMemFindMaxLnMemPoolFree` (ptSippMCB pSippMCB)
- u32 `sippMemCheck` (ptSippMCB pSippMCB, SippVirtualPool vPool)
- void `sippSetCircularLnBuffBase` (u8 *base)
- void `sippMemStatus` ()

Variables

- u32 `gSippSliceSz`
- u8 * `sippCmxBase`
- u32 `sippPoolsUsage` [SIPP_MEMPOOL_TOTAL_POOLS] = {0}
- tSippVPhysMap `gsSippMemMap` [SIPP_MAX_SUPPORTED_PIPELINES]

7.255.1 Detailed Description

SIPP engine.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.255.2 Macro Definition Documentation

`#define SIPP_MEMPOOL_CMX 0`

Referenced by `sippChooseMemPool()`, `sippInitPhysicalPoolGlobal()`, `sippMemAlloc()`, `sippMemCheck()`, `sippMemFree()`, and `sippMemInitVirtPhysMaps()`.

`#define SIPP_MEMPOOL_DDR 1`

Referenced by `sippChooseMemPool()`, `sippInitPhysicalPoolGlobal()`, `sippMemAlloc()`, `sippMemCheck()`, `sippMemFree()`, and `sippMemInitVirtPhysMaps()`.

`#define SIPP_MEMPOOL_NUM_GENERAL_POOLS 2`

Referenced by `sippMemStatus()`.

`#define SIPP_MEMPOOL_TOTAL_POOLS (SIPP_MEMPOOL_NUM_GENERAL_POOLS + (2 * SIPP_MAX_SUPPORTED_PIPELINES))`

7.255.3 Function Documentation

`u8 sippMemPoolCMX [SIPP_CMX_POOL_SZ] ALIGNED (32)`

```
void sippChooseMemPool ( ptSippMCB pSippMCB, SippVirtualPool vPool, u32 physPoolIdx )
```

```
void sippInitLnMemPool ( ptSippMCB ptMCB, u8 * start )
```

```
//
```

Referenced by sippComputeSliceLayout(), and sippCoreReschedulePipeline().

```
u32 sippInitLnMemPoolSlices ( ptSippMCB ptMCB, u32 firstSlice, u32 lastSlice )
```

Referenced by sippCoreFinalisePipeline().

```
void sippInitPhysicalPoolGlobal ( void )
```

Referenced by sippCoreResourceInit().

```
u32 sippInitPhysicalPoolPipe ( ptSippMCB ptMCB )
```

Referenced by sippInitPipeline().

```
void sippInitSchedPool ( ptSippMCB ptMCB, u32 sliceFirst, u32 sliceLast )
```

Referenced by sippInitPipeline().

```
void* sippMemAlloc ( ptSippMCB pSippMCB, SippVirtualPool vPool, s32 bytes )
```

Referenced by createTripleConv3x3(), sippAllocCmxLineBuffers(), sippAllocCmxLineBuffersO-Pipe(), sippAssignCmxMemRegion(), sippComputeBufferProps(), sippComputePaddingOffsets(), sipp-ComputeSliceLayout(), sippComputeSwOutCt(), sippCoreFinalisePipeline(), sippCQInit(), sippCreate-Filter(), sippCreatePipeline(), sippDmaCQInit(), sippFilterAddOBuf(), sippGenericAllocRuntime-Sched(), sippGenericSchedAllocTempStorage(), sippGenericSchedule(), sippIbufSetup(), sippInit-Dma(), sippInitLnMemPoolSlices(), sippInitPhysicalPoolPipe(), sippLinkFilter(), sippMapRegion-ToCmx(), sippObufSetup(), and sippOSECreate().

```
u32 sippMemCheck ( ptSippMCB pSippMCB, SippVirtualPool vPool )
```

Referenced by sippMemAlloc().

```
SippVirtualPool sippMemFindMaxLnMemPoolFree ( ptSippMCB pSippMCB )
```

```
void sippMemFree ( ptSippMCB pSippMCB, SippVirtualPool vPool, void * pPtr )
```

Referenced by sippDeletePipeline().

```
void sippMemFreeList ( ptSippMCB pSippMCB, SippVirtualPool vPool )
```

Referenced by sippCoreReschedulePipeline(), and sippFreePipeResource().

```
void sippMemInitVirtPhysMaps ( void )
```

Referenced by sippCoreResourceInit().

```
void sippMemStatus ( void )
```

Referenced by sippAllocCmxLineBuffersOPipe().

```
void sippSetCircularLnBuffBase ( u8 * base )
```

7.255.4 Variable Documentation

u32 gSippSliceSz

Referenced by sippCmxDmaInit(), sippCreatePipeline(), sippInitLnMemPool(), sippInitLnMemPoolSlices(), sippInitPhysicalPoolGlobal(), sippInitSchedPool(), sippMemFindMaxLnMemPoolFree(), and sippSetSliceSize().

tSippVPhysMap gsSippMemMap[SIPP_MAX_SUPPORTED_PIPELINES]

u8* sippCmxBase

Referenced by sippComputeSliceLayout(), sippDumpHtmlMap(), sippInitLnMemPool(), sippInitLnMemPoolSlices(), sippInitPhysicalPoolGlobal(), sippInitSchedPool(), sippMapRegionMapAddrToSliceZero(), sippMapRegionToCmx(), and sippMemLBMatchRegionsToChunks().

u32 sippPoolsUsage[SIPP_MEMPOOL_TOTAL_POOLS] = {0}

7.256 sippMemLineBuffer.c File Reference

SIPP engine.

```
#include <sipp.h>
#include <sippInternal.h>
```

Functions

- **u32** sippMapRegionMapAddrToSliceZero (SippPipeline *pipe, u32 addrOffset)
- void sippMapRegionToCmx (SippPipeline *pipe, SippMemRegion *nextMemRegion)
- **u8** sippGetNextMemRegion (SippPipeline *pipe, SippMemRegion *memRegList, SippMemRegion **ppNextMemRegion)
- **u8** sippMemRegionAllocLineBuffer (pSippCmxBufferMap pCmxMap, u32 cmxSlice, SippFilter *ptrFilt, u32 oBufIdx, pSippMemRegionListNode *ppStartOfChainNode)
- **u8** sippMemAllocChainChunk (pSippMemRegionListNode pStartOfChainNode, u32 numChunks, SippFilter *ptrFilt, u32 oBufIdx)
- void sippConfirmChunkChain (pSippMemRegionListNode *nodeChain, u32 chainLength)
- **u8** sippMemLBMatchRegionsToChunks (pSippCmxBufferMap pCmxMap, u32 chunkStride, u32 numChunks)

- `bool sippMemLBRemoveNode (pSippCmxBufferMap pCmxMap, u32 sliceIdx, pSippMemRegionListNode pNodeRemove)`
- `void sippMemLBConsolidateRegions (pSippCmxBufferMap pCmxMap)`
- `s32 sippAssignCmxMemRegion (SippPipeline *pipe, SippMemRegion *memRegList)`
- `s32 sippAllocCmxLineBuffers (SippPipeline *pipe)`
- `s32 sippAllocCmxLineBuffersOPipe (SippPipeline *pipe)`
- `s32 sippAllocCmxMemRegion (SippPipeline *pipe, SippMemRegion *memRegList)`

Variables

- `u8 * sippCmxBase`

7.256.1 Detailed Description

SIPP engine.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.256.2 Function Documentation

`s32 sippAllocCmxLineBuffers (SippPipeline * pipe)`

`s32 sippAllocCmxLineBuffersOPipe (SippPipeline * pipe)`

`s32 sippAllocCmxMemRegion (SippPipeline * pipe, SippMemRegion * memRegList)`

`s32 sippAssignCmxMemRegion (SippPipeline * pipe, SippMemRegion * memRegList)`

Referenced by `sippAllocCmxMemRegion()`.

`void sippConfirmChunkChain (pSippMemRegionListNode * nodeChain, u32 chainLength)`

Referenced by `sippMemLBMatchRegionsToChunks()`.

`u8 sippGetNextMemRegion (SippPipeline * pipe, SippMemRegion * memRegList, SippMemRegion ** ppNextMemRegion)`

Referenced by `sippAssignCmxMemRegion()`.

`u32 sippMapRegionMapAddrToSliceZero (SippPipeline * pipe, u32 addrOffset)`

Referenced by `sippMapRegionToCmx()`.


```
void sippMapRegionToCmx ( SippPipeline * pipe, SippMemRegion * nextMemRegion )
```

Referenced by sippAssignCmxMemRegion().

```
u8 sippMemAllocChainChunk ( pSippMemRegionListNode pStartOfChainNode, u32 numChunks,  
SippFilter * ptrFilt, u32 oBufIdx )
```

Referenced by sippAllocCmxLineBuffers().

```
void sippMemLBConsolidateRegions ( pSippCmxBufferMap pCmxMap )
```

Referenced by sippCoreFinalisePipeline().

```
u8 sippMemLBMatchRegionsToChunks ( pSippCmxBufferMap pCmxMap, u32 chunkStride, u32  
numChunks )
```

Referenced by sippCoreFinalisePipeline().

```
bool sippMemLBRemoveNode ( pSippCmxBufferMap pCmxMap, u32 sliceIdx,  
pSippMemRegionListNode pNodeRemove )
```

Referenced by sippMemLBConsolidateRegions().

```
u8 sippMemRegionAllocLineBuffer ( pSippCmxBufferMap pCmxMap, u32 cmxSlice, SippFilter *  
ptrFilt, u32 oBufIdx, pSippMemRegionListNode * ppStartOfChainNode )
```

Referenced by sippAllocCmxLineBuffers().

7.256.3 Variable Documentation

```
u8* sippCmxBase
```

7.257 sippMLPIRuntime.c File Reference

Multiple Liner Per Iteration runtime.

7.257.1 Detailed Description

Multiple Liner Per Iteration runtime.

Copyright

All code copyright Movidius Ltd 2016, all rights reserved. For License Warranty see:
common/license.txt

7.258 sippMLPIRuntimeema2x5x.c File Reference

Multiple Line Per Iteration runtime, ma2x5x specific aspect.

7.258.1 Detailed Description

Multiple Line Per Iteration runtime, ma2x5x specific aspect.

Copyright

All code copyright Movidius Ltd 2016, all rights reserved. For License Warranty see: common/license.txt

7.259 sippMLPISchApi.c File Reference

7.260 sippMLPISchDebug.c File Reference

7.261 sippMLPISchReq.c File Reference

7.262 sippMLPISchWrite.c File Reference

SIPP engine.

7.262.1 Detailed Description

SIPP engine.

Copyright

All code copyright Movidius Ltd 2015, all rights reserved. For License Warranty see: common/license.txt

7.263 sippOPipeRuntime.c File Reference

SIPP engine.

7.263.1 Detailed Description

SIPP engine.

Copyright

All code copyright Movidius Ltd 2016, all rights reserved. For License Warranty see: common/license.txt

7.264 sippOPipeSchApi.c File Reference

7.265 sippOPipeSchedulingEntity.c File Reference

Establish the OPipe Scheduling Entities within a pipeline Platform(s) supported : ma2x5x.

```
#include <sipp.h>
#include <sippInternal.h>
```

Macros

- #define [SIPP_OPIPE_FILTER_LIST](#)
- #define [SIPP_OCTOPUS_MASK](#)

Functions

- [u32 sippOSEProcessFilterIBufs](#) (SippOse *pOSE, [SippFilter](#) *ptrFilt)
- void [sippOSEProcessFilterOBufs](#) (SippOse *pOSE, [SippFilter](#) *ptrFilt)
- [u32 sippOSEProcessFilter](#) (SippOse *pOSE, [u8](#) filterIdx)
- SippOse * [sippOSECreate](#) ([SippPipeline](#) *pPipe)
- void [sippOSEAddFilter](#) (SippOse *pOSE, [SippFilter](#) *ptrFilt)
- void [sippOSEComplete](#) (SippOse *pOSE)
- [u32 sippFindConnectionsListRecursive](#) ([SippFilter](#) *ptrFilt, [SippFilter](#) **filterList, [u32](#) numList, [u8](#) parents, [u32](#) connectedFilt)
- [u32 sippRemoveEntriesList](#) ([u32](#) entryMask, [SippFilter](#) **filterList, [u32](#) numEntries)
- [s32 sippIdentifyOPipeSchedulingEntity](#) ([pSippPipeline](#) pPipe)
- [u32 sippCheckOPipeConnectionSigma](#) ([SippFilter](#) *ptrFilt, [SippFilter](#) *parentFilt, [u32](#) *parOBufIdx)
- [u32 sippCheckOPipeConnectionLsc](#) ([SippFilter](#) *ptrFilt, [SippFilter](#) *parentFilt, [u32](#) *parOBufIdx)
- [u32 sippCheckOPipeConnectionRaw](#) ([SippFilter](#) *ptrFilt, [SippFilter](#) *parentFilt, [u32](#) *parOBufIdx)
- [u32 sippCheckOPipeConnectionDbyr](#) ([SippFilter](#) *ptrFilt, [SippFilter](#) *parentFilt, [u32](#) *parOBufIdx)
- [u32 sippCheckOPipeConnectionDoGLTM](#) ([SippFilter](#) *ptrFilt, [SippFilter](#) *parentFilt, [u32](#) *parOBufIdx)
- [u32 sippCheckOPipeConnectionLuma](#) ([SippFilter](#) *ptrFilt, [SippFilter](#) *parentFilt, [u32](#) *parOBufIdx)
- [u32 sippCheckOPipeConnectionSharpen](#) ([SippFilter](#) *ptrFilt, [SippFilter](#) *parentFilt, [u32](#) *parOBufIdx)
- [u32 sippCheckOPipeConnectionGenChroma](#) ([SippFilter](#) *ptrFilt, [SippFilter](#) *parentFilt, [u32](#) *parOBufIdx)
- [u32 sippCheckOPipeConnectionMedian](#) ([SippFilter](#) *ptrFilt, [SippFilter](#) *parentFilt, [u32](#) *parOBufIdx)
- [u32 sippCheckOPipeConnectionChrDns](#) ([SippFilter](#) *ptrFilt, [SippFilter](#) *parentFilt, [u32](#) *parOBufIdx)
- [u32 sippCheckOPipeConnectionColourComb](#) ([SippFilter](#) *ptrFilt, [SippFilter](#) *parentFilt, [u32](#) *parOBufIdx)

- `u32 sippCheckOPipeConnectionLut (SippFilter *ptrFilt, SippFilter *parentFilt, u32 *parOBufIdx)`
- `u32 sippCheckOPipeConnectionPoly (SippFilter *ptrFilt, SippFilter *parentFilt, u32 *parOBufIdx)`
- `u32 sippCheckOPipeConnectionDefault (SippFilter *ptrFilt, SippFilter *parentFilt, u32 *parOBufIdx)`

7.265.1 Detailed Description

Establish the OPipe Scheduling Entities within a pipeline Platform(s) supported : ma2x5x.

Copyright

All code copyright Movidius Ltd 2015, all rights reserved. For License Warranty see: common/license.txt

7.265.2 Macro Definition Documentation

`#define SIPP_OCTOPUS_MASK`

Value:

```
((1 << SIPP_SIGMA_ID) |\
(1 << SIPP_LSC_ID) |\
(1 << SIPP_RAW_ID) |\
(1 << SIPP_DBYR_ID) |\
(1 << SIPP_DOGL_ID) |\
(1 << SIPP_LUMA_ID) |\
(1 << SIPP_SHARPEN_ID) |\
(1 << SIPP_CGEN_ID) |\
(1 << SIPP_MED_ID) |\
(1 << SIPP_CHROMA_ID) |\
(1 << SIPP_CC_ID) |\
(1 << SIPP_LUT_ID))
```

Referenced by `sippIdentifyOPipeSchedulingEntity()`.

`#define SIPP_OPIPE_FILTER_LIST`

Value:

```
((1 << SIPP_SIGMA_ID) |\
(1 << SIPP_LSC_ID) |\
(1 << SIPP_RAW_ID) |\
(1 << SIPP_DBYR_ID) |\
(1 << SIPP_DOGL_ID) |\
(1 << SIPP_LUMA_ID) |\
(1 << SIPP_SHARPEN_ID) |\
(1 << SIPP_CGEN_ID) |\
(1 << SIPP_MED_ID) |\
(1 << SIPP_CHROMA_ID) |\
(1 << SIPP_CC_ID) |\
(1 << SIPP_LUT_ID) |\
(1 << SIPP_EDGE_OP_ID) |\
(1 << SIPP_CONV_ID) |\
(1 << SIPP_HARRIS_ID) |\
(1 << SIPP_UPFIRDN0_ID) |\
(1 << SIPP_UPFIRDN1_ID) |\
(1 << SIPP_UPFIRDN2_ID) |\
```

```
(1 << SIPP_MIPI_TX0_ID) |\
(1 << SIPP_MIPI_TX1_ID) |\
(1 << SIPP_MIPI_RX0_ID) |\
(1 << SIPP_MIPI_RX1_ID) |\
(1 << SIPP_MIPI_RX2_ID) |\
(1 << SIPP_MIPI_RX3_ID))
```

Referenced by `sippFindConnectionsListRecursive()`, `sippIdentifyOPipeSchedulingEntity()`, and `sippOSEProcessFilterOBufs()`.

7.265.3 Function Documentation

u32 `sippCheckOPipeConnectionChrDns (SippFilter * ptrFilt, SippFilter * parentFilt, u32 * parOBufIdx)`

u32 `sippCheckOPipeConnectionColourComb (SippFilter * ptrFilt, SippFilter * parentFilt, u32 * parOBufIdx)`

u32 `sippCheckOPipeConnectionDbyr (SippFilter * ptrFilt, SippFilter * parentFilt, u32 * parOBufIdx)`

u32 `sippCheckOPipeConnectionDefault (SippFilter * ptrFilt, SippFilter * parentFilt, u32 * parOBufIdx)`

u32 `sippCheckOPipeConnectionDoGLTM (SippFilter * ptrFilt, SippFilter * parentFilt, u32 * parOBufIdx)`

u32 `sippCheckOPipeConnectionGenChroma (SippFilter * ptrFilt, SippFilter * parentFilt, u32 * parOBufIdx)`

u32 `sippCheckOPipeConnectionLsc (SippFilter * ptrFilt, SippFilter * parentFilt, u32 * parOBufIdx)`

u32 `sippCheckOPipeConnectionLuma (SippFilter * ptrFilt, SippFilter * parentFilt, u32 * parOBufIdx)`

u32 `sippCheckOPipeConnectionLut (SippFilter * ptrFilt, SippFilter * parentFilt, u32 * parOBufIdx)`

u32 `sippCheckOPipeConnectionMedian (SippFilter * ptrFilt, SippFilter * parentFilt, u32 * parOBufIdx)`

u32 `sippCheckOPipeConnectionPoly (SippFilter * ptrFilt, SippFilter * parentFilt, u32 * parOBufIdx)`

u32 `sippCheckOPipeConnectionRaw (SippFilter * ptrFilt, SippFilter * parentFilt, u32 * parOBufIdx)`

u32 `sippCheckOPipeConnectionSharpen (SippFilter * ptrFilt, SippFilter * parentFilt, u32 * parOBufIdx)`

```
u32 sippCheckOPipeConnectionSigma ( SippFilter * ptrFilt, SippFilter * parentFilt, u32 *
parOBufIdx )
```

```
u32 sippFindConnectionsListRecursive ( SippFilter * ptrFilt, SippFilter ** filterList, u32 numList,
u8 parents, u32 connectedFilt )
```

Referenced by sippIdentifyOPipeSchedulingEntity().

```
s32 sippIdentifyOPipeSchedulingEntity ( pSippPipeline pPipe )
```

Referenced by sippAnalysePipe2x5x().

```
void sippOSEAddFilter ( SippOse * pOSE, SippFilter * ptrFilt )
```

Referenced by sippIdentifyOPipeSchedulingEntity().

```
void sippOSEComplete ( SippOse * pOSE )
```

Referenced by sippIdentifyOPipeSchedulingEntity().

```
SippOse* sippOSECreate ( SippPipeline * pPipe )
```

Referenced by sippIdentifyOPipeSchedulingEntity().

```
u32 sippOSEProcessFilter ( SippOse * pOSE, u8 filterIdx )
```

Referenced by sippOSEComplete().

```
u32 sippOSEProcessFilterIBufs ( SippOse * pOSE, SippFilter * ptrFilt )
```

Referenced by sippOSEProcessFilter().

```
void sippOSEProcessFilterOBufs ( SippOse * pOSE, SippFilter * ptrFilt )
```

Referenced by sippOSEProcessFilter().

```
u32 sippRemoveEntriesList ( u32 entryMask, SippFilter ** filterList, u32 numEntries )
```

Referenced by sippIdentifyOPipeSchedulingEntity().

7.266 sippPal.h File Reference

```
#include <stdio.h>
#include <sippTypes.h>
#include <sippPalTypes.h>
#include <stdarg.h>
```

Functions

- void `sippPalCriticalSectionBegin` (`SIPP_PAL_CRIT_STATE *pState`)
- void `sippPalCriticalSectionEnd` (`SIPP_PAL_CRIT_STATE PreviousState`)
- void * `sippPalMemcpy` (void *pDest, const void *pSrc, `u32` uSize)
- void `sippPalMemset` (void *pDest, `s32` nChar, `u32` uCount)
- `u8` `sippPalMemCompare` (void *pArea1, void *pArea2, `u32` uSizeInWords)
- void `sippPalTraceInit` (void)
- void `sippPalTrace` (`u32` uFlags, const char *psz_format,...)
- void `sippPalPrintInt` (`u32` uVal)
- `u32` `sippPalFindHighestBit` (`u32` uValue)

7.266.1 Function Documentation

`void sippPalCriticalSectionBegin (SIPP_PAL_CRIT_STATE * pState)`

Referenced by `dmaKickSequenceConcurrent()`, `sippAccessSchedulerQuPush()`, `sippCreatePipeline()`, `sippDeletePipeline()`, `sippEventNotify()`, `sippGenericRuntimeLoadPipeline()`, `sippGenericRuntimeProcessIters()`, `sippHWSessionAddActiveLists()`, `sippHWSessionRemoveActiveLists()`, `sippHWSessionRemoveLoadedPipe()`, `sippPalQuPost()`, `sippPalQuReceive()`, and `sippProcessFrame()`.

`void sippPalCriticalSectionEnd (SIPP_PAL_CRIT_STATE PreviousState)`

Referenced by `dmaKickSequenceConcurrent()`, `sippAccessSchedulerQuPush()`, `sippCreatePipeline()`, `sippDeletePipeline()`, `sippEventNotify()`, `sippGenericRuntimeLoadPipeline()`, `sippGenericRuntimeProcessIters()`, `sippHWSessionAddActiveLists()`, `sippHWSessionRemoveActiveLists()`, `sippHWSessionRemoveLoadedPipe()`, `sippPalQuPost()`, `sippPalQuReceive()`, and `sippProcessFrame()`.

`u32 sippPalFindHighestBit (u32 uValue)`

`u8 sippPalMemCompare (void * pArea1, void * pArea2, u32 uSizeInWords)`

`void* sippPalMemcpy (void * pDest, const void * pSrc, u32 uSize)`

Referenced by `sippHWSessionRemoveActiveLists()`, and `sippIssueCommand()`.

`void sippPalMemset (void * pDest, s32 nChar, u32 uCount)`

Referenced by `sippAssignCmxMemRegion()`, `sippCreateFilter()`, `sippCreatePipeline()`, `sippGenericSchedAllocTempStorage()`, `sippHeapInit()`, and `sippInitPhysicalPoolGlobal()`.

`void sippPalPrintInt (u32 uVal)`

```
void sippPalTrace ( u32 uFlags, const char * psz_format, ... )
```

Referenced by sippAccessSchedulerQuPush(), sippAllocCmxLineBuffersOPipe(), sippCoreFinalisePipeline(), sippCoreReschedulePipeline(), sippCreateFilter(), sippCreatePipeline(), sippDbgCompareDeltaU8(), sippDbgCompareU16(), sippDbgCompareU32(), sippDbgCompareU8(), sippDbgDumpRunMask(), sippDbgShowBuffPtr(), sippDeletePipeline(), sippError(), sippFinalizePipeline(), sippGenericAllocRuntimeSched(), sippGenericBlockHWUnits2x5x(), sippGenericRunNextIter(), sippGenericRuntimeHWProcessIters(), sippGenericSchedule(), sippGenericUpdateHWUnits2x5x(), sippHeapAlloc(), sippHeapCheck(), sippHeapDefrag(), sippInitDma(), sippInitLnMemPool(), sippInitPipeline(), sippIntBarrierSetup(), sippMemAlloc(), sippMemCheck(), sippMemStatus(), sippPipeSessionControl(), sippPipeSetLinesPerIter(), sippPrintSliceWidth(), sippProcessFrame(), sippProcessFrameNB(), sippRegisterEventCallback(), sippReschedulePipeline(), and sippTestCrcCheck().

```
void sippPalTraceInit ( void )
```

Referenced by sippSWInit().

7.267 sippPalTypes.h File Reference

```
#include <sippBaseTypes.h>
```

Data Structures

- struct [SIPP_TRACE_FLAGS](#)

Macros

- #define [SIPP_ERROR_FATAL](#) 0x80000000
- #define [SIPP_ERROR_WARNING](#) 0x00000000
- #define [SIPP_TRACE_MASK_MODULE](#) 0x0FFFFFFF
- #define [SIPP_TRACE_MASK_LEVEL](#) 0x70000000
- #define [SIPP_TRACE_MASK_TIMESTAMP](#) 0x80000000
- #define [SIPP_TRACE_SHIFT_LEVEL](#) 28
- #define [SIPP_TRACE_SHIFT_TIMESTAMP](#) 31
- #define [SIPP_TRACE_LEVEL_NEVER](#) 0x00000000
- #define [SIPP_TRACE_LEVEL_1](#) 0x10000000
- #define [SIPP_TRACE_LEVEL_2](#) 0x20000000
- #define [SIPP_TRACE_LEVEL_3](#) 0x30000000
- #define [SIPP_TRACE_LEVEL_4](#) 0x40000000
- #define [SIPP_TRACE_LEVEL_5](#) 0x50000000
- #define [SIPP_TRACE_LEVEL_6](#) 0x60000000
- #define [SIPP_TRACE_LEVEL_ALWAYS](#) 0x70000000
- #define [SIPP_TRACE_NO_TIMESTAMP](#) 0x80000000
- #define [SIPP_TRACE_API](#) 0x00000001
- #define [SIPP_TRACE_FRAMEWORK_MANAGER](#) 0x00000002
- #define [SIPP_TRACE_ACCESS_SCHEDULER](#) 0x00000003

- #define SIPP_TRACE_PAL 0x00000004
- #define SIPP_TRACE_CORE 0x00000005
- #define SIPP_TRACE_SESS_CONTROL 0x00000006
- #define SIPP_TRACE_RUNTIME 0x00000007
- #define SIPP_TRACE_HEAP 0x00000008
- #define SIPP_TRACE_MAX 0x0000001F
- #define SIPP_TRACE_ANY 0x00000000
- #define SIPP_TRACE_FLAGS_WORDS ((SIPP_TRACE_MAX+31)/32)
- #define SIPP_API_TL_CATASTROPHE (SIPP_TRACE_LEVEL_ALWAYS | SIPP_TRACE_ANY)
- #define SIPP_API_TL_SEVERE (SIPP_TRACE_LEVEL_ALWAYS | SIPP_TRACE_API)
- #define SIPP_API_TL_ERROR (SIPP_TRACE_LEVEL_6 | SIPP_TRACE_API)
- #define SIPP_API_TL_WARNING (SIPP_TRACE_LEVEL_5 | SIPP_TRACE_API)
- #define SIPP_API_TL_FUNC (SIPP_TRACE_LEVEL_4 | SIPP_TRACE_API)
- #define SIPP_API_TL_INFO (SIPP_TRACE_LEVEL_3 | SIPP_TRACE_API)
- #define SIPP_API_TL_VERBOSE (SIPP_TRACE_LEVEL_2 | SIPP_TRACE_API)
- #define SIPP_API_TL_DEBUG (SIPP_TRACE_LEVEL_1 | SIPP_TRACE_API)
- #define SIPP_FW_MGR_TL_CATASTROPHE (SIPP_TRACE_LEVEL_ALWAYS | SIPP_TRACE_ANY)
- #define SIPP_FW_MGR_TL_SEVERE (SIPP_TRACE_LEVEL_ALWAYS | SIPP_TRACE_FRAMEWORK_MANAGER)
- #define SIPP_FW_MGR_TL_ERROR (SIPP_TRACE_LEVEL_6 | SIPP_TRACE_FRAMEWORK_MANAGER)
- #define SIPP_FW_MGR_TL_WARNING (SIPP_TRACE_LEVEL_5 | SIPP_TRACE_FRAMEWORK_MANAGER)
- #define SIPP_FW_MGR_TL_FUNC (SIPP_TRACE_LEVEL_4 | SIPP_TRACE_FRAMEWORK_MANAGER)
- #define SIPP_FW_MGR_TL_INFO (SIPP_TRACE_LEVEL_3 | SIPP_TRACE_FRAMEWORK_MANAGER)
- #define SIPP_FW_MGR_TL_VERBOSE (SIPP_TRACE_LEVEL_2 | SIPP_TRACE_FRAMEWORK_MANAGER)
- #define SIPP_FW_MGR_TL_DEBUG (SIPP_TRACE_LEVEL_1 | SIPP_TRACE_FRAMEWORK_MANAGER)
- #define SIPP_ACC_SCH_TL_CATASTROPHE (SIPP_TRACE_LEVEL_ALWAYS | SIPP_TRACE_ANY)
- #define SIPP_ACC_SCH_TL_SEVERE (SIPP_TRACE_LEVEL_ALWAYS | SIPP_TRACE_ACCESS_SCHEDULER)
- #define SIPP_ACC_SCH_TL_ERROR (SIPP_TRACE_LEVEL_6 | SIPP_TRACE_ACCESS_SCHEDULER)
- #define SIPP_ACC_SCH_TL_WARNING (SIPP_TRACE_LEVEL_5 | SIPP_TRACE_ACCESS_SCHEDULER)
- #define SIPP_ACC_SCH_TL_FUNC (SIPP_TRACE_LEVEL_4 | SIPP_TRACE_ACCESS_SCHEDULER)
- #define SIPP_ACC_SCH_TL_INFO (SIPP_TRACE_LEVEL_3 | SIPP_TRACE_ACCESS_SCHEDULER)
- #define SIPP_ACC_SCH_TL_VERBOSE (SIPP_TRACE_LEVEL_2 | SIPP_TRACE_ACCESS_SCHEDULER)
- #define SIPP_ACC_SCH_TL_DEBUG (SIPP_TRACE_LEVEL_1 | SIPP_TRACE_ACCESS_SCHEDULER)

- #define SIPP_PAL_TL_CATASTROPHE (SIPP_TRACE_LEVEL_ALWAYS | SIPP_TRACE_ANY)
- #define SIPP_PAL_TL_SEVERE (SIPP_TRACE_LEVEL_ALWAYS | SIPP_TRACE_PAL)
- #define SIPP_PAL_TL_ERROR (SIPP_TRACE_LEVEL_6 | SIPP_TRACE_PAL)
- #define SIPP_PAL_TL_WARNING (SIPP_TRACE_LEVEL_5 | SIPP_TRACE_PAL)
- #define SIPP_PAL_TL_FUNC (SIPP_TRACE_LEVEL_4 | SIPP_TRACE_PAL)
- #define SIPP_PAL_TL_INFO (SIPP_TRACE_LEVEL_3 | SIPP_TRACE_PAL)
- #define SIPP_PAL_TL_VERBOSE (SIPP_TRACE_LEVEL_2 | SIPP_TRACE_PAL)
- #define SIPP_PAL_TL_DEBUG (SIPP_TRACE_LEVEL_1 | SIPP_TRACE_PAL)
- #define SIPP_CORE_TL_CATASTROPHE (SIPP_TRACE_LEVEL_ALWAYS | SIPP_TRACE_CORE_ANY)
- #define SIPP_CORE_TL_SEVERE (SIPP_TRACE_LEVEL_ALWAYS | SIPP_TRACE_CORE)
- #define SIPP_CORE_TL_ERROR (SIPP_TRACE_LEVEL_6 | SIPP_TRACE_CORE)
- #define SIPP_CORE_TL_WARNING (SIPP_TRACE_LEVEL_5 | SIPP_TRACE_CORE)
- #define SIPP_CORE_TL_FUNC (SIPP_TRACE_LEVEL_4 | SIPP_TRACE_CORE)
- #define SIPP_CORE_TL_INFO (SIPP_TRACE_LEVEL_3 | SIPP_TRACE_CORE)
- #define SIPP_CORE_TL_VERBOSE (SIPP_TRACE_LEVEL_2 | SIPP_TRACE_CORE)
- #define SIPP_CORE_TL_DEBUG (SIPP_TRACE_LEVEL_1 | SIPP_TRACE_CORE)
- #define SIPP_SES_CTRL_TL_CATASTROPHE (SIPP_TRACE_LEVEL_ALWAYS | SIPP_TRACE_SESS_CONTROL_ANY)
- #define SIPP_SES_CTRL_TL_SEVERE (SIPP_TRACE_LEVEL_ALWAYS | SIPP_TRACE_SESS_CONTROL)
- #define SIPP_SES_CTRL_TL_ERROR (SIPP_TRACE_LEVEL_6 | SIPP_TRACE_SESS_CONTROL)
- #define SIPP_SES_CTRL_TL_WARNING (SIPP_TRACE_LEVEL_5 | SIPP_TRACE_SESS_CONTROL)
- #define SIPP_SES_CTRL_TL_FUNC (SIPP_TRACE_LEVEL_4 | SIPP_TRACE_SESS_CONTROL)
- #define SIPP_SES_CTRL_TL_INFO (SIPP_TRACE_LEVEL_3 | SIPP_TRACE_SESS_CONTROL)
- #define SIPP_SES_CTRL_TL_VERBOSE (SIPP_TRACE_LEVEL_2 | SIPP_TRACE_SESS_CONTROL)
- #define SIPP_SES_CTRL_TL_DEBUG (SIPP_TRACE_LEVEL_1 | SIPP_TRACE_SESS_CONTROL)
- #define SIPP_RUNTIME_TL_CATASTROPHE (SIPP_TRACE_LEVEL_ALWAYS | SIPP_TRACE_RUNTIME_ANY)
- #define SIPP_RUNTIME_TL_SEVERE (SIPP_TRACE_LEVEL_ALWAYS | SIPP_TRACE_RUNTIME)
- #define SIPP_RUNTIME_TL_ERROR (SIPP_TRACE_LEVEL_6 | SIPP_TRACE_RUNTIME)
- #define SIPP_RUNTIME_TL_WARNING (SIPP_TRACE_LEVEL_5 | SIPP_TRACE_RUNTIME)
- #define SIPP_RUNTIME_TL_FUNC (SIPP_TRACE_LEVEL_4 | SIPP_TRACE_RUNTIME)
- #define SIPP_RUNTIME_TL_INFO (SIPP_TRACE_LEVEL_3 | SIPP_TRACE_RUNTIME)
- #define SIPP_RUNTIME_TL_VERBOSE (SIPP_TRACE_LEVEL_2 | SIPP_TRACE_RUNTIME)
- #define SIPP_RUNTIME_TL_DEBUG (SIPP_TRACE_LEVEL_1 | SIPP_TRACE_RUNTIME)
- #define SIPP_HEAP_TL_CATASTROPHE (SIPP_TRACE_LEVEL_ALWAYS | SIPP_TRACE_HEAP_ANY)

- `#define SIPP_HEAP_TL_SEVERE (SIPP_TRACE_LEVEL_ALWAYS | SIPP_TRACE_RUNTIME)`
- `#define SIPP_HEAP_TL_ERROR (SIPP_TRACE_LEVEL_6 | SIPP_TRACE_RUNTIME)`
- `#define SIPP_HEAP_TL_WARNING (SIPP_TRACE_LEVEL_5 | SIPP_TRACE_RUNTIME)`
- `#define SIPP_HEAP_TL_FUNC (SIPP_TRACE_LEVEL_4 | SIPP_TRACE_RUNTIME)`
- `#define SIPP_HEAP_TL_INFO (SIPP_TRACE_LEVEL_3 | SIPP_TRACE_RUNTIME)`
- `#define SIPP_HEAP_TL_VERBOSE (SIPP_TRACE_LEVEL_2 | SIPP_TRACE_RUNTIME)`
- `#define SIPP_HEAP_TL_DEBUG (SIPP_TRACE_LEVEL_1 | SIPP_TRACE_RUNTIME)`

Typedefs

- `typedef u32 SIPP_PAL_TIMER_ID`
- `typedef u32 SIPP_PAL_CRIT_STATE`

7.267.1 Macro Definition Documentation

`#define SIPP_ACC_SCH_TL_CATASTROPHE (SIPP_TRACE_LEVEL_ALWAYS | SIPP_TRACE_ANY)`

`#define SIPP_ACC_SCH_TL_DEBUG (SIPP_TRACE_LEVEL_1 | SIPP_TRACE_ACCESS_SCHEDULER)`

`#define SIPP_ACC_SCH_TL_ERROR (SIPP_TRACE_LEVEL_6 | SIPP_TRACE_ACCESS_SCHEDULER)`

Referenced by `sippAccessSchedulerQuPush()`, `sippAllocCmxLineBuffersOPipe()`, `sippDbgCompareDeltaU8()`, `sippDbgCompareU16()`, `sippDbgCompareU32()`, `sippDbgCompareU8()`, `sippDbgDumpRunMask()`, `sippDbgShowBuffPtr()`, `sippPrintSliceWidth()`, and `sippTestCrcCheck()`.

`#define SIPP_ACC_SCH_TL_FUNC (SIPP_TRACE_LEVEL_4 | SIPP_TRACE_ACCESS_SCHEDULER)`

`#define SIPP_ACC_SCH_TL_INFO (SIPP_TRACE_LEVEL_3 | SIPP_TRACE_ACCESS_SCHEDULER)`

`#define SIPP_ACC_SCH_TL_SEVERE (SIPP_TRACE_LEVEL_ALWAYS | SIPP_TRACE_ACCESS_SCHEDULER)`

`#define SIPP_ACC_SCH_TL_VERBOSE (SIPP_TRACE_LEVEL_2 | SIPP_TRACE_ACCESS_SCHEDULER)`

`#define SIPP_ACC_SCH_TL_WARNING (SIPP_TRACE_LEVEL_5 | SIPP_TRACE_ACCESS_SCHEDULER)`

`#define SIPP_API_TL_CATASTROPHE (SIPP_TRACE_LEVEL_ALWAYS | SIPP_TRACE_ANY)`

`#define SIPP_API_TL_DEBUG (SIPP_TRACE_LEVEL_1 | SIPP_TRACE_API)`

```
#define SIPP_API_TL_ERROR (SIPP_TRACE_LEVEL_6 | SIPP_TRACE_API)
```

Referenced by sippCoreFinalisePipeline(), sippCoreReschedulePipeline(), sippCreateFilter(), sippCreatePipeline(), sippFinalizePipeline(), sippInitPipeline(), sippPipeSetLinesPerIter(), sippProcessFrame(), sippProcessFrameNB(), sippRegisterEventCallback(), and sippReschedulePipeline().

```
#define SIPP_API_TL_FUNC (SIPP_TRACE_LEVEL_4 | SIPP_TRACE_API)
```

```
#define SIPP_API_TL_INFO (SIPP_TRACE_LEVEL_3 | SIPP_TRACE_API)
```

```
#define SIPP_API_TL_SEVERE (SIPP_TRACE_LEVEL_ALWAYS | SIPP_TRACE_API)
```

```
#define SIPP_API_TL_VERBOSE (SIPP_TRACE_LEVEL_2 | SIPP_TRACE_API)
```

Referenced by sippDeletePipeline(), and sippInitDma().

```
#define SIPP_API_TL_WARNING (SIPP_TRACE_LEVEL_5 | SIPP_TRACE_API)
```

```
#define SIPP_CORE_TL_CATASTROPHE (SIPP_TRACE_LEVEL_ALWAYS |  
SIPP_TRACE_ANY)
```

```
#define SIPP_CORE_TL_DEBUG (SIPP_TRACE_LEVEL_1 | SIPP_TRACE_CORE)
```

```
#define SIPP_CORE_TL_ERROR (SIPP_TRACE_LEVEL_6 | SIPP_TRACE_CORE)
```

Referenced by sippError(), and sippMemAlloc().

```
#define SIPP_CORE_TL_FUNC (SIPP_TRACE_LEVEL_4 | SIPP_TRACE_CORE)
```

```
#define SIPP_CORE_TL_INFO (SIPP_TRACE_LEVEL_3 | SIPP_TRACE_CORE)
```

Referenced by sippInitLnMemPool(), sippMemCheck(), and sippMemStatus().

```
#define SIPP_CORE_TL_SEVERE (SIPP_TRACE_LEVEL_ALWAYS | SIPP_TRACE_CORE)
```

Referenced by sippGenericAllocRuntimeSched(), and sippGenericSchedule().

```
#define SIPP_CORE_TL_VERBOSE (SIPP_TRACE_LEVEL_2 | SIPP_TRACE_CORE)
```

```
#define SIPP_CORE_TL_WARNING (SIPP_TRACE_LEVEL_5 | SIPP_TRACE_CORE)
```

Referenced by sippMemStatus().

```
#define SIPP_ERROR_FATAL 0x80000000
```

```
#define SIPP_ERROR_WARNING 0x00000000
```

```
#define SIPP_FW_MGR_TL_CATASTROPHE (SIPP_TRACE_LEVEL_ALWAYS |
SIPP_TRACE_ANY)

#define SIPP_FW_MGR_TL_DEBUG (SIPP_TRACE_LEVEL_1 | SIPP_TRACE_FRAMEWOR-
K_MANAGER)

#define SIPP_FW_MGR_TL_ERROR (SIPP_TRACE_LEVEL_6 | SIPP_TRACE_FRAMEWOR-
K_MANAGER)

#define SIPP_FW_MGR_TL_FUNC (SIPP_TRACE_LEVEL_4 | SIPP_TRACE_FRAMEWORK-
_MANAGER)

#define SIPP_FW_MGR_TL_INFO (SIPP_TRACE_LEVEL_3 | SIPP_TRACE_FRAMEWORK_-
MANAGER)

#define SIPP_FW_MGR_TL_SEVERE (SIPP_TRACE_LEVEL_ALWAYS |
SIPP_TRACE_FRAMEWORK_MANAGER)

#define SIPP_FW_MGR_TL_VERBOSE (SIPP_TRACE_LEVEL_2 | SIPP_TRACE_FRAMEWO-
RK_MANAGER)

#define SIPP_FW_MGR_TL_WARNING (SIPP_TRACE_LEVEL_5 | SIPP_TRACE_FRAMEW-
ORK_MANAGER)

#define SIPP_HEAP_TL_CATASTROPHE (SIPP_TRACE_LEVEL_ALWAYS |
SIPP_TRACE_ANY)

#define SIPP_HEAP_TL_DEBUG (SIPP_TRACE_LEVEL_1 | SIPP_TRACE_RUNTIME)

#define SIPP_HEAP_TL_ERROR (SIPP_TRACE_LEVEL_6 | SIPP_TRACE_RUNTIME)

#define SIPP_HEAP_TL_FUNC (SIPP_TRACE_LEVEL_4 | SIPP_TRACE_RUNTIME)

#define SIPP_HEAP_TL_INFO (SIPP_TRACE_LEVEL_3 | SIPP_TRACE_RUNTIME)

Referenced by sippHeapAlloc(), sippHeapCheck(), and sippHeapDefrag().

#define SIPP_HEAP_TL_SEVERE (SIPP_TRACE_LEVEL_ALWAYS |
SIPP_TRACE_RUNTIME)

#define SIPP_HEAP_TL_VERBOSE (SIPP_TRACE_LEVEL_2 | SIPP_TRACE_RUNTIME)

#define SIPP_HEAP_TL_WARNING (SIPP_TRACE_LEVEL_5 | SIPP_TRACE_RUNTIME)

#define SIPP_PAL_TL_CATASTROPHE (SIPP_TRACE_LEVEL_ALWAYS |
SIPP_TRACE_ANY)

#define SIPP_PAL_TL_DEBUG (SIPP_TRACE_LEVEL_1 | SIPP_TRACE_PAL)

#define SIPP_PAL_TL_ERROR (SIPP_TRACE_LEVEL_6 | SIPP_TRACE_PAL)
```

```
#define SIPP_PAL_TL_FUNC (SIPP_TRACE_LEVEL_4 | SIPP_TRACE_PAL)

#define SIPP_PAL_TL_INFO (SIPP_TRACE_LEVEL_3 | SIPP_TRACE_PAL)

#define SIPP_PAL_TL_SEVERE (SIPP_TRACE_LEVEL_ALWAYS | SIPP_TRACE_PAL)

#define SIPP_PAL_TL_VERBOSE (SIPP_TRACE_LEVEL_2 | SIPP_TRACE_PAL)

#define SIPP_PAL_TL_WARNING (SIPP_TRACE_LEVEL_5 | SIPP_TRACE_PAL)

#define SIPP_RUNTIME_TL_CATASTROPHE (SIPP_TRACE_LEVEL_ALWAYS |
SIPP_TRACE_ANY)

#define SIPP_RUNTIME_TL_DEBUG (SIPP_TRACE_LEVEL_1 | SIPP_TRACE_RUNTIME)

#define SIPP_RUNTIME_TL_ERROR (SIPP_TRACE_LEVEL_6 | SIPP_TRACE_RUNTIME)

#define SIPP_RUNTIME_TL_FUNC (SIPP_TRACE_LEVEL_4 | SIPP_TRACE_RUNTIME)

#define SIPP_RUNTIME_TL_INFO (SIPP_TRACE_LEVEL_3 | SIPP_TRACE_RUNTIME)
```

Referenced by sippGenericBlockHWUnits2x5x(), sippGenericRunNextIter(), sippGenericRuntimeHW-ProcessIters(), sippGenericUpdateHWUnits2x5x(), and sippIntBarrierSetup().

```
#define SIPP_RUNTIME_TL_SEVERE (SIPP_TRACE_LEVEL_ALWAYS |
SIPP_TRACE_RUNTIME)

#define SIPP_RUNTIME_TL_VERBOSE (SIPP_TRACE_LEVEL_2 | SIPP_TRACE_RUNTIME)

#define SIPP_RUNTIME_TL_WARNING (SIPP_TRACE_LEVEL_5 | SIPP_TRACE_RUNTIME)

#define SIPP_SES_CTRL_TL_CATASTROPHE (SIPP_TRACE_LEVEL_ALWAYS |
SIPP_TRACE_ANY)

#define SIPP_SES_CTRL_TL_DEBUG (SIPP_TRACE_LEVEL_1 | SIPP_TRACE_SESS_CONT-
ROL)

#define SIPP_SES_CTRL_TL_ERROR (SIPP_TRACE_LEVEL_6 | SIPP_TRACE_SESS_CONT-
ROL)

#define SIPP_SES_CTRL_TL_FUNC (SIPP_TRACE_LEVEL_4 | SIPP_TRACE_SESS_CONTR-
OL)

#define SIPP_SES_CTRL_TL_INFO (SIPP_TRACE_LEVEL_3 | SIPP_TRACE_SESS_CONTR-
OL)

#define SIPP_SES_CTRL_TL_SEVERE (SIPP_TRACE_LEVEL_ALWAYS |
SIPP_TRACE_SESS_CONTROL)
```

```
#define SIPP_SES_CTRL_TL_VERBOSE (SIPP_TRACE_LEVEL_2 |  
SIPP_TRACE_SESS_CONTROL)
```

Referenced by sippPipeSessionControl().

```
#define SIPP_SES_CTRL_TL_WARNING (SIPP_TRACE_LEVEL_5 |  
SIPP_TRACE_SESS_CONTROL)
```

```
#define SIPP_TRACE_ACCESS_SCHEDULER 0x00000003
```

```
#define SIPP_TRACE_ANY 0x00000000
```

```
#define SIPP_TRACE_API 0x00000001
```

```
#define SIPP_TRACE_CORE 0x00000005
```

```
#define SIPP_TRACE_FLAGS_WORDS ((SIPP_TRACE_MAX+31)/32)
```

```
#define SIPP_TRACE_FRAMEWORK_MANAGER 0x00000002
```

```
#define SIPP_TRACE_HEAP 0x00000008
```

```
#define SIPP_TRACE_LEVEL_1 0x10000000
```

```
#define SIPP_TRACE_LEVEL_2 0x20000000
```

```
#define SIPP_TRACE_LEVEL_3 0x30000000
```

```
#define SIPP_TRACE_LEVEL_4 0x40000000
```

Referenced by sippPalPrintInt(), and sippPalTrace().

```
#define SIPP_TRACE_LEVEL_5 0x50000000
```

Referenced by sippPalTraceInit().

```
#define SIPP_TRACE_LEVEL_6 0x60000000
```

```
#define SIPP_TRACE_LEVEL_ALWAYS 0x70000000
```

```
#define SIPP_TRACE_LEVEL_NEVER 0x00000000
```

```
#define SIPP_TRACE_MASK_LEVEL 0x70000000
```

```
#define SIPP_TRACE_MASK_MODULE 0x0FFFFFFF
```

```
#define SIPP_TRACE_MASK_TIMESTAMP 0x80000000
```

```
#define SIPP_TRACE_MAX 0x0000001F
```

```
#define SIPP_TRACE_NO_TIMESTAMP 0x80000000

#define SIPP_TRACE_PAL 0x00000004

#define SIPP_TRACE_RUNTIME 0x00000007

#define SIPP_TRACE_SESS_CONTROL 0x00000006

#define SIPP_TRACE_SHIFT_LEVEL 28

#define SIPP_TRACE_SHIFT_TIMESTAMP 31
```

7.267.2 Typedef Documentation

```
typedef u32 SIPP_PAL_CRIT_STATE
```

```
typedef u32 SIPP_PAL_TIMER_ID
```

7.268 sippPalTypes.h File Reference

```
#include <sippBaseTypes.h>
```

Data Structures

- struct [SIPP_TRACE_FLAGS](#)

Macros

- #define [SIPP_ERROR_FATAL](#) 0x80000000
- #define [SIPP_ERROR_WARNING](#) 0x00000000
- #define [SIPP_TRACE_MASK_MODULE](#) 0x0FFFFFFF
- #define [SIPP_TRACE_MASK_LEVEL](#) 0x70000000
- #define [SIPP_TRACE_MASK_TIMESTAMP](#) 0x80000000
- #define [SIPP_TRACE_SHIFT_LEVEL](#) 28
- #define [SIPP_TRACE_SHIFT_TIMESTAMP](#) 31
- #define [SIPP_TRACE_LEVEL_NEVER](#) 0x00000000
- #define [SIPP_TRACE_LEVEL_1](#) 0x10000000
- #define [SIPP_TRACE_LEVEL_2](#) 0x20000000
- #define [SIPP_TRACE_LEVEL_3](#) 0x30000000
- #define [SIPP_TRACE_LEVEL_4](#) 0x40000000
- #define [SIPP_TRACE_LEVEL_5](#) 0x50000000
- #define [SIPP_TRACE_LEVEL_6](#) 0x60000000
- #define [SIPP_TRACE_LEVEL_ALWAYS](#) 0x70000000
- #define [SIPP_TRACE_NO_TIMESTAMP](#) 0x80000000
- #define [SIPP_TRACE_API](#) 0x00000001
- #define [SIPP_TRACE_FRAMEWORK_MANAGER](#) 0x00000002
- #define [SIPP_TRACE_ACCESS_SCHEDULER](#) 0x00000003

- #define SIPP_TRACE_PAL 0x00000004
- #define SIPP_TRACE_CORE 0x00000005
- #define SIPP_TRACE_SESS_CONTROL 0x00000006
- #define SIPP_TRACE_RUNTIME 0x00000007
- #define SIPP_TRACE_HEAP 0x00000008
- #define SIPP_TRACE_MAX 0x0000001F
- #define SIPP_TRACE_ANY 0x00000000
- #define SIPP_TRACE_FLAGS_WORDS ((SIPP_TRACE_MAX+31)/32)
- #define SIPP_API_TL_CATASTROPHE (SIPP_TRACE_LEVEL_ALWAYS | SIPP_TRACE_ANY)
- #define SIPP_API_TL_SEVERE (SIPP_TRACE_LEVEL_ALWAYS | SIPP_TRACE_API)
- #define SIPP_API_TL_ERROR (SIPP_TRACE_LEVEL_6 | SIPP_TRACE_API)
- #define SIPP_API_TL_WARNING (SIPP_TRACE_LEVEL_5 | SIPP_TRACE_API)
- #define SIPP_API_TL_FUNC (SIPP_TRACE_LEVEL_4 | SIPP_TRACE_API)
- #define SIPP_API_TL_INFO (SIPP_TRACE_LEVEL_3 | SIPP_TRACE_API)
- #define SIPP_API_TL_VERBOSE (SIPP_TRACE_LEVEL_2 | SIPP_TRACE_API)
- #define SIPP_API_TL_DEBUG (SIPP_TRACE_LEVEL_1 | SIPP_TRACE_API)
- #define SIPP_FW_MGR_TL_CATASTROPHE (SIPP_TRACE_LEVEL_ALWAYS | SIPP_TRACE_ANY)
- #define SIPP_FW_MGR_TL_SEVERE (SIPP_TRACE_LEVEL_ALWAYS | SIPP_TRACE_FRAMEWORK_MANAGER)
- #define SIPP_FW_MGR_TL_ERROR (SIPP_TRACE_LEVEL_6 | SIPP_TRACE_FRAMEWORK_MANAGER)
- #define SIPP_FW_MGR_TL_WARNING (SIPP_TRACE_LEVEL_5 | SIPP_TRACE_FRAMEWORK_MANAGER)
- #define SIPP_FW_MGR_TL_FUNC (SIPP_TRACE_LEVEL_4 | SIPP_TRACE_FRAMEWORK_MANAGER)
- #define SIPP_FW_MGR_TL_INFO (SIPP_TRACE_LEVEL_3 | SIPP_TRACE_FRAMEWORK_MANAGER)
- #define SIPP_FW_MGR_TL_VERBOSE (SIPP_TRACE_LEVEL_2 | SIPP_TRACE_FRAMEWORK_MANAGER)
- #define SIPP_FW_MGR_TL_DEBUG (SIPP_TRACE_LEVEL_1 | SIPP_TRACE_FRAMEWORK_MANAGER)
- #define SIPP_ACC_SCH_TL_CATASTROPHE (SIPP_TRACE_LEVEL_ALWAYS | SIPP_TRACE_ANY)
- #define SIPP_ACC_SCH_TL_SEVERE (SIPP_TRACE_LEVEL_ALWAYS | SIPP_TRACE_ACCESS_SCHEDULER)
- #define SIPP_ACC_SCH_TL_ERROR (SIPP_TRACE_LEVEL_6 | SIPP_TRACE_ACCESS_SCHEDULER)
- #define SIPP_ACC_SCH_TL_WARNING (SIPP_TRACE_LEVEL_5 | SIPP_TRACE_ACCESS_SCHEDULER)
- #define SIPP_ACC_SCH_TL_FUNC (SIPP_TRACE_LEVEL_4 | SIPP_TRACE_ACCESS_SCHEDULER)
- #define SIPP_ACC_SCH_TL_INFO (SIPP_TRACE_LEVEL_3 | SIPP_TRACE_ACCESS_SCHEDULER)
- #define SIPP_ACC_SCH_TL_VERBOSE (SIPP_TRACE_LEVEL_2 | SIPP_TRACE_ACCESS_SCHEDULER)
- #define SIPP_ACC_SCH_TL_DEBUG (SIPP_TRACE_LEVEL_1 | SIPP_TRACE_ACCESS_SCHEDULER)

- #define SIPP_PAL_TL_CATASTROPHE (SIPP_TRACE_LEVEL_ALWAYS | SIPP_TRACE_ANY)
- #define SIPP_PAL_TL_SEVERE (SIPP_TRACE_LEVEL_ALWAYS | SIPP_TRACE_PAL)
- #define SIPP_PAL_TL_ERROR (SIPP_TRACE_LEVEL_6 | SIPP_TRACE_PAL)
- #define SIPP_PAL_TL_WARNING (SIPP_TRACE_LEVEL_5 | SIPP_TRACE_PAL)
- #define SIPP_PAL_TL_FUNC (SIPP_TRACE_LEVEL_4 | SIPP_TRACE_PAL)
- #define SIPP_PAL_TL_INFO (SIPP_TRACE_LEVEL_3 | SIPP_TRACE_PAL)
- #define SIPP_PAL_TL_VERBOSE (SIPP_TRACE_LEVEL_2 | SIPP_TRACE_PAL)
- #define SIPP_PAL_TL_DEBUG (SIPP_TRACE_LEVEL_1 | SIPP_TRACE_PAL)
- #define SIPP_CORE_TL_CATASTROPHE (SIPP_TRACE_LEVEL_ALWAYS | SIPP_TRACE_ANY)
- #define SIPP_CORE_TL_SEVERE (SIPP_TRACE_LEVEL_ALWAYS | SIPP_TRACE_CORE)
- #define SIPP_CORE_TL_ERROR (SIPP_TRACE_LEVEL_6 | SIPP_TRACE_CORE)
- #define SIPP_CORE_TL_WARNING (SIPP_TRACE_LEVEL_5 | SIPP_TRACE_CORE)
- #define SIPP_CORE_TL_FUNC (SIPP_TRACE_LEVEL_4 | SIPP_TRACE_CORE)
- #define SIPP_CORE_TL_INFO (SIPP_TRACE_LEVEL_3 | SIPP_TRACE_CORE)
- #define SIPP_CORE_TL_VERBOSE (SIPP_TRACE_LEVEL_2 | SIPP_TRACE_CORE)
- #define SIPP_CORE_TL_DEBUG (SIPP_TRACE_LEVEL_1 | SIPP_TRACE_CORE)
- #define SIPP_SES_CTRL_TL_CATASTROPHE (SIPP_TRACE_LEVEL_ALWAYS | SIPP_TRACE_SESS_CONTROL)
- #define SIPP_SES_CTRL_TL_SEVERE (SIPP_TRACE_LEVEL_ALWAYS | SIPP_TRACE_SESS_CONTROL)
- #define SIPP_SES_CTRL_TL_ERROR (SIPP_TRACE_LEVEL_6 | SIPP_TRACE_SESS_CONTROL)
- #define SIPP_SES_CTRL_TL_WARNING (SIPP_TRACE_LEVEL_5 | SIPP_TRACE_SESS_CONTROL)
- #define SIPP_SES_CTRL_TL_FUNC (SIPP_TRACE_LEVEL_4 | SIPP_TRACE_SESS_CONTROL)
- #define SIPP_SES_CTRL_TL_INFO (SIPP_TRACE_LEVEL_3 | SIPP_TRACE_SESS_CONTROL)
- #define SIPP_SES_CTRL_TL_VERBOSE (SIPP_TRACE_LEVEL_2 | SIPP_TRACE_SESS_CONTROL)
- #define SIPP_SES_CTRL_TL_DEBUG (SIPP_TRACE_LEVEL_1 | SIPP_TRACE_SESS_CONTROL)
- #define SIPP_RUNTIME_TL_CATASTROPHE (SIPP_TRACE_LEVEL_ALWAYS | SIPP_TRACE_RUNTIME)
- #define SIPP_RUNTIME_TL_SEVERE (SIPP_TRACE_LEVEL_ALWAYS | SIPP_TRACE_RUNTIME)
- #define SIPP_RUNTIME_TL_ERROR (SIPP_TRACE_LEVEL_6 | SIPP_TRACE_RUNTIME)
- #define SIPP_RUNTIME_TL_WARNING (SIPP_TRACE_LEVEL_5 | SIPP_TRACE_RUNTIME)
- #define SIPP_RUNTIME_TL_FUNC (SIPP_TRACE_LEVEL_4 | SIPP_TRACE_RUNTIME)
- #define SIPP_RUNTIME_TL_INFO (SIPP_TRACE_LEVEL_3 | SIPP_TRACE_RUNTIME)
- #define SIPP_RUNTIME_TL_VERBOSE (SIPP_TRACE_LEVEL_2 | SIPP_TRACE_RUNTIME)
- #define SIPP_RUNTIME_TL_DEBUG (SIPP_TRACE_LEVEL_1 | SIPP_TRACE_RUNTIME)
- #define SIPP_HEAP_TL_CATASTROPHE (SIPP_TRACE_LEVEL_ALWAYS | SIPP_TRACE_ANY)

- `#define SIPP_HEAP_TL_SEVERE (SIPP_TRACE_LEVEL_ALWAYS | SIPP_TRACE_RUNTIME)`
- `#define SIPP_HEAP_TL_ERROR (SIPP_TRACE_LEVEL_6 | SIPP_TRACE_RUNTIME)`
- `#define SIPP_HEAP_TL_WARNING (SIPP_TRACE_LEVEL_5 | SIPP_TRACE_RUNTIME)`
- `#define SIPP_HEAP_TL_FUNC (SIPP_TRACE_LEVEL_4 | SIPP_TRACE_RUNTIME)`
- `#define SIPP_HEAP_TL_INFO (SIPP_TRACE_LEVEL_3 | SIPP_TRACE_RUNTIME)`
- `#define SIPP_HEAP_TL_VERBOSE (SIPP_TRACE_LEVEL_2 | SIPP_TRACE_RUNTIME)`
- `#define SIPP_HEAP_TL_DEBUG (SIPP_TRACE_LEVEL_1 | SIPP_TRACE_RUNTIME)`

Typedefs

- `typedef u32 SIPP_PAL_TIMER_ID`
- `typedef u32 SIPP_PAL_CRIT_STATE`

7.268.1 Macro Definition Documentation

`#define SIPP_ACC_SCH_TL_CATASTROPHE (SIPP_TRACE_LEVEL_ALWAYS | SIPP_TRACE_ANY)`

`#define SIPP_ACC_SCH_TL_DEBUG (SIPP_TRACE_LEVEL_1 | SIPP_TRACE_ACCESS_SCHEDULER)`

`#define SIPP_ACC_SCH_TL_ERROR (SIPP_TRACE_LEVEL_6 | SIPP_TRACE_ACCESS_SCHEDULER)`

`#define SIPP_ACC_SCH_TL_FUNC (SIPP_TRACE_LEVEL_4 | SIPP_TRACE_ACCESS_SCHEDULER)`

`#define SIPP_ACC_SCH_TL_INFO (SIPP_TRACE_LEVEL_3 | SIPP_TRACE_ACCESS_SCHEDULER)`

`#define SIPP_ACC_SCH_TL_SEVERE (SIPP_TRACE_LEVEL_ALWAYS | SIPP_TRACE_ACCESS_SCHEDULER)`

`#define SIPP_ACC_SCH_TL_VERBOSE (SIPP_TRACE_LEVEL_2 | SIPP_TRACE_ACCESS_SCHEDULER)`

`#define SIPP_ACC_SCH_TL_WARNING (SIPP_TRACE_LEVEL_5 | SIPP_TRACE_ACCESS_SCHEDULER)`

`#define SIPP_API_TL_CATASTROPHE (SIPP_TRACE_LEVEL_ALWAYS | SIPP_TRACE_ANY)`

`#define SIPP_API_TL_DEBUG (SIPP_TRACE_LEVEL_1 | SIPP_TRACE_API)`

`#define SIPP_API_TL_ERROR (SIPP_TRACE_LEVEL_6 | SIPP_TRACE_API)`

`#define SIPP_API_TL_FUNC (SIPP_TRACE_LEVEL_4 | SIPP_TRACE_API)`

`#define SIPP_API_TL_INFO (SIPP_TRACE_LEVEL_3 | SIPP_TRACE_API)`

```
#define SIPP_API_TL_SEVERE (SIPP_TRACE_LEVEL_ALWAYS | SIPP_TRACE_API)

#define SIPP_API_TL_VERBOSE (SIPP_TRACE_LEVEL_2 | SIPP_TRACE_API)

#define SIPP_API_TL_WARNING (SIPP_TRACE_LEVEL_5 | SIPP_TRACE_API)

#define SIPP_CORE_TL_CATASTROPHE (SIPP_TRACE_LEVEL_ALWAYS |
SIPP_TRACE_ANY)

#define SIPP_CORE_TL_DEBUG (SIPP_TRACE_LEVEL_1 | SIPP_TRACE_CORE)

#define SIPP_CORE_TL_ERROR (SIPP_TRACE_LEVEL_6 | SIPP_TRACE_CORE)

#define SIPP_CORE_TL_FUNC (SIPP_TRACE_LEVEL_4 | SIPP_TRACE_CORE)

#define SIPP_CORE_TL_INFO (SIPP_TRACE_LEVEL_3 | SIPP_TRACE_CORE)

#define SIPP_CORE_TL_SEVERE (SIPP_TRACE_LEVEL_ALWAYS | SIPP_TRACE_CORE)

#define SIPP_CORE_TL_VERBOSE (SIPP_TRACE_LEVEL_2 | SIPP_TRACE_CORE)

#define SIPP_CORE_TL_WARNING (SIPP_TRACE_LEVEL_5 | SIPP_TRACE_CORE)

#define SIPP_ERROR_FATAL 0x80000000

#define SIPP_ERROR_WARNING 0x00000000

#define SIPP_FW_MGR_TL_CATASTROPHE (SIPP_TRACE_LEVEL_ALWAYS |
SIPP_TRACE_ANY)

#define SIPP_FW_MGR_TL_DEBUG (SIPP_TRACE_LEVEL_1 | SIPP_TRACE_FRAMEWORK-
K_MANAGER)

#define SIPP_FW_MGR_TL_ERROR (SIPP_TRACE_LEVEL_6 | SIPP_TRACE_FRAMEWORK-
K_MANAGER)

#define SIPP_FW_MGR_TL_FUNC (SIPP_TRACE_LEVEL_4 | SIPP_TRACE_FRAMEWORK-
_MANAGER)

#define SIPP_FW_MGR_TL_INFO (SIPP_TRACE_LEVEL_3 | SIPP_TRACE_FRAMEWORK_-
MANAGER)

#define SIPP_FW_MGR_TL_SEVERE (SIPP_TRACE_LEVEL_ALWAYS |
SIPP_TRACE_FRAMEWORK_MANAGER)

#define SIPP_FW_MGR_TL_VERBOSE (SIPP_TRACE_LEVEL_2 | SIPP_TRACE_FRAMEWO-
RK_MANAGER)

#define SIPP_FW_MGR_TL_WARNING (SIPP_TRACE_LEVEL_5 | SIPP_TRACE_FRAMEW-
ORK_MANAGER)
```

```
#define SIPP_HEAP_TL_CATASTROPHE (SIPP_TRACE_LEVEL_ALWAYS |
SIPP_TRACE_ANY)

#define SIPP_HEAP_TL_DEBUG (SIPP_TRACE_LEVEL_1 | SIPP_TRACE_RUNTIME)

#define SIPP_HEAP_TL_ERROR (SIPP_TRACE_LEVEL_6 | SIPP_TRACE_RUNTIME)

#define SIPP_HEAP_TL_FUNC (SIPP_TRACE_LEVEL_4 | SIPP_TRACE_RUNTIME)

#define SIPP_HEAP_TL_INFO (SIPP_TRACE_LEVEL_3 | SIPP_TRACE_RUNTIME)

#define SIPP_HEAP_TL_SEVERE (SIPP_TRACE_LEVEL_ALWAYS |
SIPP_TRACE_RUNTIME)

#define SIPP_HEAP_TL_VERBOSE (SIPP_TRACE_LEVEL_2 | SIPP_TRACE_RUNTIME)

#define SIPP_HEAP_TL_WARNING (SIPP_TRACE_LEVEL_5 | SIPP_TRACE_RUNTIME)

#define SIPP_PAL_TL_CATASTROPHE (SIPP_TRACE_LEVEL_ALWAYS |
SIPP_TRACE_ANY)

#define SIPP_PAL_TL_DEBUG (SIPP_TRACE_LEVEL_1 | SIPP_TRACE_PAL)

#define SIPP_PAL_TL_ERROR (SIPP_TRACE_LEVEL_6 | SIPP_TRACE_PAL)

#define SIPP_PAL_TL_FUNC (SIPP_TRACE_LEVEL_4 | SIPP_TRACE_PAL)

#define SIPP_PAL_TL_INFO (SIPP_TRACE_LEVEL_3 | SIPP_TRACE_PAL)

#define SIPP_PAL_TL_SEVERE (SIPP_TRACE_LEVEL_ALWAYS | SIPP_TRACE_PAL)

#define SIPP_PAL_TL_VERBOSE (SIPP_TRACE_LEVEL_2 | SIPP_TRACE_PAL)

#define SIPP_PAL_TL_WARNING (SIPP_TRACE_LEVEL_5 | SIPP_TRACE_PAL)

#define SIPP_RUNTIME_TL_CATASTROPHE (SIPP_TRACE_LEVEL_ALWAYS |
SIPP_TRACE_ANY)

#define SIPP_RUNTIME_TL_DEBUG (SIPP_TRACE_LEVEL_1 | SIPP_TRACE_RUNTIME)

#define SIPP_RUNTIME_TL_ERROR (SIPP_TRACE_LEVEL_6 | SIPP_TRACE_RUNTIME)

#define SIPP_RUNTIME_TL_FUNC (SIPP_TRACE_LEVEL_4 | SIPP_TRACE_RUNTIME)

#define SIPP_RUNTIME_TL_INFO (SIPP_TRACE_LEVEL_3 | SIPP_TRACE_RUNTIME)

#define SIPP_RUNTIME_TL_SEVERE (SIPP_TRACE_LEVEL_ALWAYS |
SIPP_TRACE_RUNTIME)

#define SIPP_RUNTIME_TL_VERBOSE (SIPP_TRACE_LEVEL_2 | SIPP_TRACE_RUNTIME)
```

```
#define SIPP_RUNTIME_TL_WARNING (SIPP_TRACE_LEVEL_5 | SIPP_TRACE_RUNTIME)

#define SIPP_SES_CTRL_TL_CATASTROPHE (SIPP_TRACE_LEVEL_ALWAYS |
SIPP_TRACE_ANY)

#define SIPP_SES_CTRL_TL_DEBUG (SIPP_TRACE_LEVEL_1 | SIPP_TRACE_SESS_CONT-
ROL)

#define SIPP_SES_CTRL_TL_ERROR (SIPP_TRACE_LEVEL_6 | SIPP_TRACE_SESS_CONT-
ROL)

#define SIPP_SES_CTRL_TL_FUNC (SIPP_TRACE_LEVEL_4 | SIPP_TRACE_SESS CONTR-
OL)

#define SIPP_SES_CTRL_TL_INFO (SIPP_TRACE_LEVEL_3 | SIPP_TRACE_SESS CONTR-
OL)

#define SIPP_SES_CTRL_TL_SEVERE (SIPP_TRACE_LEVEL_ALWAYS |
SIPP_TRACE_SESS_CONTROL)

#define SIPP_SES_CTRL_TL_VERBOSE (SIPP_TRACE_LEVEL_2 |
SIPP_TRACE_SESS_CONTROL)

#define SIPP_SES_CTRL_TL_WARNING (SIPP_TRACE_LEVEL_5 |
SIPP_TRACE_SESS_CONTROL)

#define SIPP_TRACE_ACCESS_SCHEDULER 0x00000003

#define SIPP_TRACE_ANY 0x00000000

#define SIPP_TRACE_API 0x00000001

#define SIPP_TRACE_CORE 0x00000005

#define SIPP_TRACE_FLAGS_WORDS ((SIPP_TRACE_MAX+31)/32)

#define SIPP_TRACE_FRAMEWORK_MANAGER 0x00000002

#define SIPP_TRACE_HEAP 0x00000008

#define SIPP_TRACE_LEVEL_1 0x10000000

#define SIPP_TRACE_LEVEL_2 0x20000000

#define SIPP_TRACE_LEVEL_3 0x30000000

#define SIPP_TRACE_LEVEL_4 0x40000000

#define SIPP_TRACE_LEVEL_5 0x50000000

#define SIPP_TRACE_LEVEL_6 0x60000000
```

```
#define SIPP_TRACE_LEVEL_ALWAYS 0x70000000

#define SIPP_TRACE_LEVEL_NEVER 0x00000000

#define SIPP_TRACE_MASK_LEVEL 0x70000000

#define SIPP_TRACE_MASK_MODULE 0x0FFFFFFF

#define SIPP_TRACE_MASK_TIMESTAMP 0x80000000

#define SIPP_TRACE_MAX 0x0000001F

#define SIPP_TRACE_NO_TIMESTAMP 0x80000000

#define SIPP_TRACE_PAL 0x00000004

#define SIPP_TRACE_RUNTIME 0x00000007

#define SIPP_TRACE_SESS_CONTROL 0x00000006

#define SIPP_TRACE_SHIFT_LEVEL 28

#define SIPP_TRACE_SHIFT_TIMESTAMP 31
```

7.268.2 Typedef Documentation

```
typedef u32 SIPP_PAL_CRIT_STATE
```

```
typedef u32 SIPP_PAL_TIMER_ID
```

7.269 sippPalTypes.h File Reference

```
#include <sippBaseTypes.h>
#include <rtems.h>
#include <mqueue.h>
#include <pthread.h>
```

Data Structures

- struct [SIPP_PAL_THREAD](#)
- struct [SIPP_PAL_QU](#)
- struct [SIPP_TRACE_FLAGS](#)

Macros

- #define [SIPP_QU_FLAG_STRUCT_FULL](#) 0x1
- #define [SIPP_ERROR_FATAL](#) 0x80000000
- #define [SIPP_ERROR_WARNING](#) 0x00000000
- #define [SIPP_TRACE_MASK_MODULE](#) 0x0FFFFFFF

- #define SIPP_TRACE_MASK_LEVEL 0x70000000
- #define SIPP_TRACE_MASK_TIMESTAMP 0x80000000
- #define SIPP_TRACE_SHIFT_LEVEL 28
- #define SIPP_TRACE_SHIFT_TIMESTAMP 31
- #define SIPP_TRACE_LEVEL_NEVER 0x00000000
- #define SIPP_TRACE_LEVEL_1 0x10000000
- #define SIPP_TRACE_LEVEL_2 0x20000000
- #define SIPP_TRACE_LEVEL_3 0x30000000
- #define SIPP_TRACE_LEVEL_4 0x40000000
- #define SIPP_TRACE_LEVEL_5 0x50000000
- #define SIPP_TRACE_LEVEL_6 0x60000000
- #define SIPP_TRACE_LEVEL_ALWAYS 0x70000000
- #define SIPP_TRACE_NO_TIMESTAMP 0x80000000
- #define SIPP_TRACE_API 0x00000001
- #define SIPP_TRACE_FRAMEWORK_MANAGER 0x00000002
- #define SIPP_TRACE_ACCESS_SCHEDULER 0x00000003
- #define SIPP_TRACE_PAL 0x00000004
- #define SIPP_TRACE_CORE 0x00000005
- #define SIPP_TRACE_SESS_CONTROL 0x00000006
- #define SIPP_TRACE_RUNTIME 0x00000007
- #define SIPP_TRACE_HEAP 0x00000008
- #define SIPP_TRACE_MAX 0x0000001F
- #define SIPP_TRACE_ANY 0x00000000
- #define SIPP_TRACE_FLAGS_WORDS ((SIPP_TRACE_MAX+31)/32)
- #define SIPP_API_TL_CATASTROPHE (SIPP_TRACE_LEVEL_ALWAYS | SIPP_TRACE_ANY)
- #define SIPP_API_TL_SEVERE (SIPP_TRACE_LEVEL_ALWAYS | SIPP_TRACE_API)
- #define SIPP_API_TL_ERROR (SIPP_TRACE_LEVEL_6 | SIPP_TRACE_API)
- #define SIPP_API_TL_WARNING (SIPP_TRACE_LEVEL_5 | SIPP_TRACE_API)
- #define SIPP_API_TL_FUNC (SIPP_TRACE_LEVEL_4 | SIPP_TRACE_API)
- #define SIPP_API_TL_INFO (SIPP_TRACE_LEVEL_3 | SIPP_TRACE_API)
- #define SIPP_API_TL_VERBOSE (SIPP_TRACE_LEVEL_2 | SIPP_TRACE_API)
- #define SIPP_API_TL_DEBUG (SIPP_TRACE_LEVEL_1 | SIPP_TRACE_API)
- #define SIPP_FW_MGR_TL_CATASTROPHE (SIPP_TRACE_LEVEL_ALWAYS | SIPP_TRACE_ANY)
- #define SIPP_FW_MGR_TL_SEVERE (SIPP_TRACE_LEVEL_ALWAYS | SIPP_TRACE_FRAMEWORK_MANAGER)
- #define SIPP_FW_MGR_TL_ERROR (SIPP_TRACE_LEVEL_6 | SIPP_TRACE_FRAMEWORK_MANAGER)
- #define SIPP_FW_MGR_TL_WARNING (SIPP_TRACE_LEVEL_5 | SIPP_TRACE_FRAMEWORK_MANAGER)
- #define SIPP_FW_MGR_TL_FUNC (SIPP_TRACE_LEVEL_4 | SIPP_TRACE_FRAMEWORK_MANAGER)
- #define SIPP_FW_MGR_TL_INFO (SIPP_TRACE_LEVEL_3 | SIPP_TRACE_FRAMEWORK_MANAGER)
- #define SIPP_FW_MGR_TL_VERBOSE (SIPP_TRACE_LEVEL_2 | SIPP_TRACE_FRAMEWORK_MANAGER)

- #define SIPP_FW_MGR_TL_DEBUG (SIPP_TRACE_LEVEL_1 | SIPP_TRACE_FRAMEWORK_MANAGER)
- #define SIPP_ACC_SCH_TL_CATASTROPHE (SIPP_TRACE_LEVEL_ALWAYS | SIPP_TRACE_ANY)
- #define SIPP_ACC_SCH_TL_SEVERE (SIPP_TRACE_LEVEL_ALWAYS | SIPP_TRACE_ACCESS_SCHEDULER)
- #define SIPP_ACC_SCH_TL_ERROR (SIPP_TRACE_LEVEL_6 | SIPP_TRACE_ACCESS_SCHEDULER)
- #define SIPP_ACC_SCH_TL_WARNING (SIPP_TRACE_LEVEL_5 | SIPP_TRACE_ACCESS_SCHEDULER)
- #define SIPP_ACC_SCH_TL_FUNC (SIPP_TRACE_LEVEL_4 | SIPP_TRACE_ACCESS_SCHEDULER)
- #define SIPP_ACC_SCH_TL_INFO (SIPP_TRACE_LEVEL_3 | SIPP_TRACE_ACCESS_SCHEDULER)
- #define SIPP_ACC_SCH_TL_VERBOSE (SIPP_TRACE_LEVEL_2 | SIPP_TRACE_ACCESS_SCHEDULER)
- #define SIPP_ACC_SCH_TL_DEBUG (SIPP_TRACE_LEVEL_1 | SIPP_TRACE_ACCESS_SCHEDULER)
- #define SIPP_PAL_TL_CATASTROPHE (SIPP_TRACE_LEVEL_ALWAYS | SIPP_TRACE_ANY)
- #define SIPP_PAL_TL_SEVERE (SIPP_TRACE_LEVEL_ALWAYS | SIPP_TRACE_PAL)
- #define SIPP_PAL_TL_ERROR (SIPP_TRACE_LEVEL_6 | SIPP_TRACE_PAL)
- #define SIPP_PAL_TL_WARNING (SIPP_TRACE_LEVEL_5 | SIPP_TRACE_PAL)
- #define SIPP_PAL_TL_FUNC (SIPP_TRACE_LEVEL_4 | SIPP_TRACE_PAL)
- #define SIPP_PAL_TL_INFO (SIPP_TRACE_LEVEL_3 | SIPP_TRACE_PAL)
- #define SIPP_PAL_TL_VERBOSE (SIPP_TRACE_LEVEL_2 | SIPP_TRACE_PAL)
- #define SIPP_PAL_TL_DEBUG (SIPP_TRACE_LEVEL_1 | SIPP_TRACE_PAL)
- #define SIPP_CORE_TL_CATASTROPHE (SIPP_TRACE_LEVEL_ALWAYS | SIPP_TRACE_ANY)
- #define SIPP_CORE_TL_SEVERE (SIPP_TRACE_LEVEL_ALWAYS | SIPP_TRACE_CORE)
- #define SIPP_CORE_TL_ERROR (SIPP_TRACE_LEVEL_6 | SIPP_TRACE_CORE)
- #define SIPP_CORE_TL_WARNING (SIPP_TRACE_LEVEL_5 | SIPP_TRACE_CORE)
- #define SIPP_CORE_TL_FUNC (SIPP_TRACE_LEVEL_4 | SIPP_TRACE_CORE)
- #define SIPP_CORE_TL_INFO (SIPP_TRACE_LEVEL_3 | SIPP_TRACE_CORE)
- #define SIPP_CORE_TL_VERBOSE (SIPP_TRACE_LEVEL_2 | SIPP_TRACE_CORE)
- #define SIPP_CORE_TL_DEBUG (SIPP_TRACE_LEVEL_1 | SIPP_TRACE_CORE)
- #define SIPP_SES_CTRL_TL_CATASTROPHE (SIPP_TRACE_LEVEL_ALWAYS | SIPP_TRACE_ANY)
- #define SIPP_SES_CTRL_TL_SEVERE (SIPP_TRACE_LEVEL_ALWAYS | SIPP_TRACE_SESSION_CONTROL)
- #define SIPP_SES_CTRL_TL_ERROR (SIPP_TRACE_LEVEL_6 | SIPP_TRACE_SESSION_CONTROL)
- #define SIPP_SES_CTRL_TL_WARNING (SIPP_TRACE_LEVEL_5 | SIPP_TRACE_SESSION_CONTROL)
- #define SIPP_SES_CTRL_TL_FUNC (SIPP_TRACE_LEVEL_4 | SIPP_TRACE_SESSION_CONTROL)
- #define SIPP_SES_CTRL_TL_INFO (SIPP_TRACE_LEVEL_3 | SIPP_TRACE_SESSION_CONTROL)

- `#define SIPP_SES_CTRL_TL_VERBOSE (SIPP_TRACE_LEVEL_2 | SIPP_TRACE_SESS_CTRL)`
- `#define SIPP_SES_CTRL_TL_DEBUG (SIPP_TRACE_LEVEL_1 | SIPP_TRACE_SESS_CTRL)`
- `#define SIPP_RUNTIME_TL_CATASTROPHE (SIPP_TRACE_LEVEL_ALWAYS | SIPP_TRACE_ANY)`
- `#define SIPP_RUNTIME_TL_SEVERE (SIPP_TRACE_LEVEL_ALWAYS | SIPP_TRACE_RUNTIME)`
- `#define SIPP_RUNTIME_TL_ERROR (SIPP_TRACE_LEVEL_6 | SIPP_TRACE_RUNTIME)`
- `#define SIPP_RUNTIME_TL_WARNING (SIPP_TRACE_LEVEL_5 | SIPP_TRACE_RUNTIME)`
- `#define SIPP_RUNTIME_TL_FUNC (SIPP_TRACE_LEVEL_4 | SIPP_TRACE_RUNTIME)`
- `#define SIPP_RUNTIME_TL_INFO (SIPP_TRACE_LEVEL_3 | SIPP_TRACE_RUNTIME)`
- `#define SIPP_RUNTIME_TL_VERBOSE (SIPP_TRACE_LEVEL_2 | SIPP_TRACE_RUNTIME)`
- `#define SIPP_RUNTIME_TL_DEBUG (SIPP_TRACE_LEVEL_1 | SIPP_TRACE_RUNTIME)`
- `#define SIPP_HEAP_TL_CATASTROPHE (SIPP_TRACE_LEVEL_ALWAYS | SIPP_TRACE_RUNTIME)`
- `#define SIPP_HEAP_TL_SEVERE (SIPP_TRACE_LEVEL_ALWAYS | SIPP_TRACE_RUNTIME)`
- `#define SIPP_HEAP_TL_ERROR (SIPP_TRACE_LEVEL_6 | SIPP_TRACE_RUNTIME)`
- `#define SIPP_HEAP_TL_WARNING (SIPP_TRACE_LEVEL_5 | SIPP_TRACE_RUNTIME)`
- `#define SIPP_HEAP_TL_FUNC (SIPP_TRACE_LEVEL_4 | SIPP_TRACE_RUNTIME)`
- `#define SIPP_HEAP_TL_INFO (SIPP_TRACE_LEVEL_3 | SIPP_TRACE_RUNTIME)`
- `#define SIPP_HEAP_TL_VERBOSE (SIPP_TRACE_LEVEL_2 | SIPP_TRACE_RUNTIME)`
- `#define SIPP_HEAP_TL_DEBUG (SIPP_TRACE_LEVEL_1 | SIPP_TRACE_RUNTIME)`

Typedefs

- `typedef void (*)(SIPP_PAL_PFTHREAD)(void **)`
- `typedef void (*)(SIPP_RTEMS_PFTHREAD)(void *)`
- `typedef struct SIPP_PAL_THREAD * SIPP_PAL_THREAD_ID`
- `typedef struct SIPP_PAL_QU * SIPP_PAL_QU_ID`
- `typedef u32 SIPP_PAL_QU_MSG`
- `typedef u32 SIPP_PAL_TIMER_ID`
- `typedef u32 SIPP_PAL_CRIT_STATE`

7.269.1 Macro Definition Documentation

`#define SIPP_ACC_SCH_TL_CATASTROPHE (SIPP_TRACE_LEVEL_ALWAYS | SIPP_TRACE_ANY)`

`#define SIPP_ACC_SCH_TL_DEBUG (SIPP_TRACE_LEVEL_1 | SIPP_TRACE_ACCESS_SCHEDULER)`

`#define SIPP_ACC_SCH_TL_ERROR (SIPP_TRACE_LEVEL_6 | SIPP_TRACE_ACCESS_SCHEDULER)`

```
#define SIPP_ACC_SCH_TL_FUNC (SIPP_TRACE_LEVEL_4 | SIPP_TRACE_ACCESS_SCH-
EDULER)

#define SIPP_ACC_SCH_TL_INFO (SIPP_TRACE_LEVEL_3 | SIPP_TRACE_ACCESS_SCHE-
DULER)

#define SIPP_ACC_SCH_TL_SEVERE (SIPP_TRACE_LEVEL_ALWAYS |
SIPP_TRACE_ACCESS_SCHEDULER)

#define SIPP_ACC_SCH_TL_VERBOSE (SIPP_TRACE_LEVEL_2 | SIPP_TRACE_ACCESS_S-
CHEDULER)

#define SIPP_ACC_SCH_TL_WARNING (SIPP_TRACE_LEVEL_5 |
SIPP_TRACE_ACCESS_SCHEDULER)

#define SIPP_API_TL_CATASTROPHE (SIPP_TRACE_LEVEL_ALWAYS |
SIPP_TRACE_ANY)

#define SIPP_API_TL_DEBUG (SIPP_TRACE_LEVEL_1 | SIPP_TRACE_API)

#define SIPP_API_TL_ERROR (SIPP_TRACE_LEVEL_6 | SIPP_TRACE_API)

#define SIPP_API_TL_FUNC (SIPP_TRACE_LEVEL_4 | SIPP_TRACE_API)

#define SIPP_API_TL_INFO (SIPP_TRACE_LEVEL_3 | SIPP_TRACE_API)

#define SIPP_API_TL_SEVERE (SIPP_TRACE_LEVEL_ALWAYS | SIPP_TRACE_API)

#define SIPP_API_TL_VERBOSE (SIPP_TRACE_LEVEL_2 | SIPP_TRACE_API)

#define SIPP_API_TL_WARNING (SIPP_TRACE_LEVEL_5 | SIPP_TRACE_API)

#define SIPP_CORE_TL_CATASTROPHE (SIPP_TRACE_LEVEL_ALWAYS |
SIPP_TRACE_ANY)

#define SIPP_CORE_TL_DEBUG (SIPP_TRACE_LEVEL_1 | SIPP_TRACE_CORE)

#define SIPP_CORE_TL_ERROR (SIPP_TRACE_LEVEL_6 | SIPP_TRACE_CORE)

#define SIPP_CORE_TL_FUNC (SIPP_TRACE_LEVEL_4 | SIPP_TRACE_CORE)

#define SIPP_CORE_TL_INFO (SIPP_TRACE_LEVEL_3 | SIPP_TRACE_CORE)

#define SIPP_CORE_TL_SEVERE (SIPP_TRACE_LEVEL_ALWAYS | SIPP_TRACE_CORE)

#define SIPP_CORE_TL_VERBOSE (SIPP_TRACE_LEVEL_2 | SIPP_TRACE_CORE)

#define SIPP_CORE_TL_WARNING (SIPP_TRACE_LEVEL_5 | SIPP_TRACE_CORE)

#define SIPP_ERROR_FATAL 0x80000000

#define SIPP_ERROR_WARNING 0x00000000
```

```
#define SIPP_FW_MGR_TL_CATASTROPHE (SIPP_TRACE_LEVEL_ALWAYS |
SIPP_TRACE_ANY)

#define SIPP_FW_MGR_TL_DEBUG (SIPP_TRACE_LEVEL_1 | SIPP_TRACE_FRAMEWOR-
K_MANAGER)

#define SIPP_FW_MGR_TL_ERROR (SIPP_TRACE_LEVEL_6 | SIPP_TRACE_FRAMEWOR-
K_MANAGER)

#define SIPP_FW_MGR_TL_FUNC (SIPP_TRACE_LEVEL_4 | SIPP_TRACE_FRAMEWORK-
_MANAGER)

#define SIPP_FW_MGR_TL_INFO (SIPP_TRACE_LEVEL_3 | SIPP_TRACE_FRAMEWORK_-
MANAGER)

#define SIPP_FW_MGR_TL_SEVERE (SIPP_TRACE_LEVEL_ALWAYS |
SIPP_TRACE_FRAMEWORK_MANAGER)

#define SIPP_FW_MGR_TL_VERBOSE (SIPP_TRACE_LEVEL_2 | SIPP_TRACE_FRAMEWO-
RK_MANAGER)

#define SIPP_FW_MGR_TL_WARNING (SIPP_TRACE_LEVEL_5 | SIPP_TRACE_FRAMEW-
ORK_MANAGER)

#define SIPP_HEAP_TL_CATASTROPHE (SIPP_TRACE_LEVEL_ALWAYS |
SIPP_TRACE_ANY)

#define SIPP_HEAP_TL_DEBUG (SIPP_TRACE_LEVEL_1 | SIPP_TRACE_RUNTIME)

#define SIPP_HEAP_TL_ERROR (SIPP_TRACE_LEVEL_6 | SIPP_TRACE_RUNTIME)

#define SIPP_HEAP_TL_FUNC (SIPP_TRACE_LEVEL_4 | SIPP_TRACE_RUNTIME)

#define SIPP_HEAP_TL_INFO (SIPP_TRACE_LEVEL_3 | SIPP_TRACE_RUNTIME)

#define SIPP_HEAP_TL_SEVERE (SIPP_TRACE_LEVEL_ALWAYS |
SIPP_TRACE_RUNTIME)

#define SIPP_HEAP_TL_VERBOSE (SIPP_TRACE_LEVEL_2 | SIPP_TRACE_RUNTIME)

#define SIPP_HEAP_TL_WARNING (SIPP_TRACE_LEVEL_5 | SIPP_TRACE_RUNTIME)

#define SIPP_PAL_TL_CATASTROPHE (SIPP_TRACE_LEVEL_ALWAYS |
SIPP_TRACE_ANY)

#define SIPP_PAL_TL_DEBUG (SIPP_TRACE_LEVEL_1 | SIPP_TRACE_PAL)

#define SIPP_PAL_TL_ERROR (SIPP_TRACE_LEVEL_6 | SIPP_TRACE_PAL)

#define SIPP_PAL_TL_FUNC (SIPP_TRACE_LEVEL_4 | SIPP_TRACE_PAL)
```

```
#define SIPP_PAL_TL_INFO (SIPP_TRACE_LEVEL_3 | SIPP_TRACE_PAL)

#define SIPP_PAL_TL_SEVERE (SIPP_TRACE_LEVEL_ALWAYS | SIPP_TRACE_PAL)

#define SIPP_PAL_TL_VERBOSE (SIPP_TRACE_LEVEL_2 | SIPP_TRACE_PAL)

#define SIPP_PAL_TL_WARNING (SIPP_TRACE_LEVEL_5 | SIPP_TRACE_PAL)

#define SIPP_QU_FLAG_STRUCT_FULL 0x1
```

Referenced by sippPalQuPost(), and sippPalQuReceive().

```
#define SIPP_RUNTIME_TL_CATASTROPHE (SIPP_TRACE_LEVEL_ALWAYS |
SIPP_TRACE_ANY)

#define SIPP_RUNTIME_TL_DEBUG (SIPP_TRACE_LEVEL_1 | SIPP_TRACE_RUNTIME)

#define SIPP_RUNTIME_TL_ERROR (SIPP_TRACE_LEVEL_6 | SIPP_TRACE_RUNTIME)

#define SIPP_RUNTIME_TL_FUNC (SIPP_TRACE_LEVEL_4 | SIPP_TRACE_RUNTIME)

#define SIPP_RUNTIME_TL_INFO (SIPP_TRACE_LEVEL_3 | SIPP_TRACE_RUNTIME)

#define SIPP_RUNTIME_TL_SEVERE (SIPP_TRACE_LEVEL_ALWAYS |
SIPP_TRACE_RUNTIME)

#define SIPP_RUNTIME_TL_VERBOSE (SIPP_TRACE_LEVEL_2 | SIPP_TRACE_RUNTIME)

#define SIPP_RUNTIME_TL_WARNING (SIPP_TRACE_LEVEL_5 | SIPP_TRACE_RUNTIME)

#define SIPP_SES_CTRL_TL_CATASTROPHE (SIPP_TRACE_LEVEL_ALWAYS |
SIPP_TRACE_ANY)

#define SIPP_SES_CTRL_TL_DEBUG (SIPP_TRACE_LEVEL_1 | SIPP_TRACE_SESS_CONT-
ROL)

#define SIPP_SES_CTRL_TL_ERROR (SIPP_TRACE_LEVEL_6 | SIPP_TRACE_SESS_CONT-
ROL)

#define SIPP_SES_CTRL_TL_FUNC (SIPP_TRACE_LEVEL_4 | SIPP_TRACE_SESS_CONTR-
OL)

#define SIPP_SES_CTRL_TL_INFO (SIPP_TRACE_LEVEL_3 | SIPP_TRACE_SESS_CONTR-
OL)

#define SIPP_SES_CTRL_TL_SEVERE (SIPP_TRACE_LEVEL_ALWAYS |
SIPP_TRACE_SESS_CONTROL)

#define SIPP_SES_CTRL_TL_VERBOSE (SIPP_TRACE_LEVEL_2 |
SIPP_TRACE_SESS_CONTROL)
```

```
#define SIPP_SES_CTRL_TL_WARNING (SIPP_TRACE_LEVEL_5 |
SIPP_TRACE_SESS_CONTROL)

#define SIPP_TRACE_ACCESS_SCHEDULER 0x00000003

#define SIPP_TRACE_ANY 0x00000000

#define SIPP_TRACE_API 0x00000001

#define SIPP_TRACE_CORE 0x00000005

#define SIPP_TRACE_FLAGS_WORDS ((SIPP_TRACE_MAX+31)/32)

#define SIPP_TRACE_FRAMEWORK_MANAGER 0x00000002

#define SIPP_TRACE_HEAP 0x00000008

#define SIPP_TRACE_LEVEL_1 0x10000000

#define SIPP_TRACE_LEVEL_2 0x20000000

#define SIPP_TRACE_LEVEL_3 0x30000000

#define SIPP_TRACE_LEVEL_4 0x40000000

#define SIPP_TRACE_LEVEL_5 0x50000000

#define SIPP_TRACE_LEVEL_6 0x60000000

#define SIPP_TRACE_LEVEL_ALWAYS 0x70000000

#define SIPP_TRACE_LEVEL_NEVER 0x00000000

#define SIPP_TRACE_MASK_LEVEL 0x70000000

#define SIPP_TRACE_MASK_MODULE 0xFFFFFFFF

#define SIPP_TRACE_MASK_TIMESTAMP 0x80000000

#define SIPP_TRACE_MAX 0x0000001F

#define SIPP_TRACE_NO_TIMESTAMP 0x80000000

#define SIPP_TRACE_PAL 0x00000004

#define SIPP_TRACE_RUNTIME 0x00000007

#define SIPP_TRACE_SESS_CONTROL 0x00000006

#define SIPP_TRACE_SHIFT_LEVEL 28

#define SIPP_TRACE_SHIFT_TIMESTAMP 31
```

7.269.2 Typedef Documentation

```
typedef u32 SIPP_PAL_CRIT_STATE
```

```
typedef void>(* SIPP_PAL_PFNTHREAD)(void **)
```

```
typedef struct SIPP_PAL_QU * SIPP_PAL_QU_ID
```

```
typedef u32 SIPP_PAL_QU_MSG
```

```
typedef struct SIPP_PAL_THREAD * SIPP_PAL_THREAD_ID
```

```
typedef u32 SIPP_PAL_TIMER_ID
```

```
typedef void>(* SIPP_RTEMS_PFNTHREAD)(void *)
```

7.270 sippPipelineValidate.c File Reference

Examine a pipeline for validity & features pertinent to implementation Platform(s) supported : MA2x5x.

```
#include <sipp.h>
#include <sippInternal.h>
```

Functions

- void [sippAnalysePipe](#) (pSippPipeline pPipe)
- bool [sippValidatePipe](#) (pSippPipeline pPipe)

7.270.1 Detailed Description

Examine a pipeline for validity & features pertinent to implementation Platform(s) supported : MA2x5x.

Copyright

All code copyright Movidius Ltd 2015, all rights reserved. For License Warranty see: common/license.txt

7.270.2 Function Documentation

```
void sippAnalysePipe ( pSippPipeline pPipe )
```

Referenced by [sippValidatePipe](#)().

```
bool sippValidatePipe ( pSippPipeline pPipe )
```

Referenced by [sippPipeSessionControl](#)().

7.271 sippPipeSessionControl.c File Reference

Processes commands from sipp access scheduler Establishes context for and makes calls to sipp HW and shaves This file provides the pipeline aspect to the interface. It is part of a set with SippHWSessionControl.c which provides the HW functionality Platform(s) supported : ma2x5x.

```
#include <sipp.h>
#include <sippInternal.h>
```

Functions

- void [sippPipeSessionControlInit](#) ()
- void [sippPipeSessionControl](#) ([eSIPP_ACCESS_SCHEDULER_EVENT](#) eEvent, [SIPP_ACCESS_SCHEDULER_EVENT_DATA](#) pData, [u32](#) uPipeIdx, [u32](#) uHWPipeIdx)

Variables

- [tSippFramework](#) gSippFramework
- [pSIPP_HW_SESSION](#) pgSippHW

7.271.1 Detailed Description

Processes commands from sipp access scheduler Establishes context for and makes calls to sipp HW and shaves This file provides the pipeline aspect to the interface. It is part of a set with SippHWSessionControl.c which provides the HW functionality Platform(s) supported : ma2x5x.

Copyright

All code copyright Movidius Ltd 2015, all rights reserved. For License Warranty see: common/license.txt

7.271.2 Function Documentation

```
void sippPipeSessionControl ( eSIPP\_ACCESS\_SCHEDULER\_EVENT eEvent,
SIPP\_ACCESS\_SCHEDULER\_EVENT\_DATA pData, u32 uPipeIdx, u32 uHWPipeIdx )
```

```
void sippPipeSessionControlInit ( void )
```

Referenced by [sippSWInit](#)().

7.271.3 Variable Documentation

[tSippFramework](#) gSippFramework

[pSIPP_HW_SESSION](#) pgSippHW

7.272 sippPlatform.h File Reference

SIPP engine.

7.272.1 Detailed Description

SIPP engine.

Copyright

All code copyright Movidius Ltd 2015, all rights reserved. For License Warranty see: common/license.txt

7.273 sippPlatform_ma2x5x.h File Reference

SIPP engine.

```
#include <mv_types.h>
#include <stdio.h>
#include <string.h>
#include <stdint.h>
#include <stdlib.h>
#include <math.h>
#include "sippHw.h"
#include <half.h>
```

Macros

- #define [INLINE](#)
- #define [ALIGNED](#)(x)
- #define [SECTION](#)(x)
- #define [DBG_PRINT](#) printf
- #define [unitTestFinalReport](#)()
- #define [SIPP_PC](#)
- #define [SVU_SYM](#)(s) s
- #define [SHAVE_MAIN](#) shave_main
- #define [U32_MEMSET](#)(addr, len, val) memset((void*)(addr), (int)(len), (size_t)(val))
- #define [U32_MEMCPY](#)(dest, src, len) memcpy((void*)(dest), (void*)(src), (size_t)(len))
- #define [SIPP_MBIN](#)(x) 0
- #define [VCS_PRINT_INT](#)(a)
- #define [CMX_TEXT](#)
- #define [CMX_RODATA](#)
- #define [CMX_DATA](#)
- #define [CMX_BSS](#)
- #define [DDR_TEXT](#)
- #define [DDR_RODATA](#)
- #define [DDR_DATA](#)
- #define [DDR_BSS](#)
- #define [UNUSED](#)(x) (void)x
- #define [NOP](#)
- #define [SIPP_SVU_STACK_SZ](#) (8*1024)

Functions

- int `scGetShaveNumber` ()

7.273.1 Detailed Description

SIPP engine.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: `common/license.txt`

7.273.2 Macro Definition Documentation

```
#define ALIGNED( x )
```

```
#define CMX_BSS
```

```
#define CMX_DATA
```

```
#define CMX_RODATA
```

```
#define CMX_TEXT
```

```
#define DBG_PRINT printf
```

```
#define DDR_BSS
```

```
#define DDR_DATA
```

```
#define DDR_RODATA
```

```
#define DDR_TEXT
```

```
#define INLINE
```

```
#define NOP
```

Referenced by `sippCmxDmaInit()`, `sippGenericStartHWUnits2x5x()`, `sippGenericStartUnits()`, `sippGenericWaitUnits()`, `sippHWSessionInit()`, `sippWait()`, and `sippWaitDma()`.

```
#define SECTION( x )
```

```
#define SHAVE_MAIN shave_main
```

Referenced by `sippKickShaveM1PC()`, and `sippKickSvus()`.

```
#define SIPP_MBIN( x ) 0
```

```
#define SIPP_PC
```

```
#define SIPP_SVU_STACK_SZ (8*1024)
```

Referenced by sippComputeSliceLayout().

```
#define SVU_SYM( s ) s
```

```
#define U32_MEMCPY( dest, src, len ) memcpy((void*)(dest), (void*)(src), (size_t)(len))
```

```
#define U32_MEMSET( addr, len, val ) memset((void*)(addr), (int)(len), (size_t)(val))
```

```
#define unitTestFinalReport( )
```

```
#define UNUSED( x ) (void)x
```

Referenced by askChromaGenDownsizer(), askCrop(), askCropLatency(), askHwColorCombChroma(), askHwMedLumaLatency(), askPolyFirResizer(), askRegular(), askRegularLatency(), askResizer(), askResizerLatency(), HarrisResponse(), lumaGenLut(), sharpenSigmaToCoefficients(), sippAddFilterToPipe(), sippCheckOPipeConnectionChrDns(), sippCheckOPipeConnectionColourComb(), sippCheckOPipeConnectionDbyr(), sippCheckOPipeConnectionDefault(), sippCheckOPipeConnectionDoGLTM(), sippCheckOPipeConnectionGenChroma(), sippCheckOPipeConnectionLsc(), sippCheckOPipeConnectionLuma(), sippCheckOPipeConnectionMedian(), sippCheckOPipeConnectionPoly(), sippCheckOPipeConnectionRaw(), sippCheckOPipeConnectionSharpen(), sippCheckOPipeConnectionSigma(), sippChooseMemPool(), sippCoreHwInitialSave(), sippDbgCreateDumpFiles(), sippDbgDumpAsmOffsets(), sippDbgDumpFilterOuts(), sippDbgDumpGraph(), sippDbgDumpRunMask(), sippDbgDumpSchedForVcsCArr(), sippDbgShowBuffPtr(), sippDumpHtmlMap(), sippGenericDbgPrintRunnable(), sippGenericDbgShowBufferReq(), sippGenericDMACheck(), sippGenericRunNextIter(), sippGenericSchedule(), sippGetIBufCtxSigma(), sippGetIBufIdsSigma(), sippGetOBufIdsMipiRx0(), sippGetOBufIdsMipiRx1(), sippGetOBufIdsMipiRx2(), sippGetOBufIdsMipiRx3(), sippGetOBufIdsMipiTx0(), sippGetOBufIdsMipiTx1(), sippGetOBufIdsSigma(), sippMapRegionMapAddrToSliceZero(), sippObflIncHandler(), sippPalFindHighestBit(), sippPrintSliceWidth(), sippRdFileU8toF16(), sippSetCircularLnBuffBase(), sippSetOBufLevelsMipiTx0(), sippSetOBufLevelsMipiTx1(), sippShaveDebug(), sippSizeList(), sippTestCrcCheck(), sippUtilComputeFp16Lut(), sippUtilComputeFp16LutChannelMode(), sippUtilPrintFp16Lut(), and sippWrFileF16toU8().

```
#define VCS_PRINT_INT( a )
```

Referenced by dmaKickSequenceConcurrent(), sippComputeSliceLayout(), sippGenericRunIterDone(), sippGenericRunNextIter(), sippGenericRuntimeHWProcessIters(), sippGenericStartUnits(), sippIbflDecHandler(), sippMemAlloc(), and sippObflIncHandler().

7.273.3 Function Documentation

```
int scGetShaveNumber ( )
```

Referenced by configureBicubicHWblock(), genXYlist(), getIn3PlanePtr(), getInPtr(), getOutPtr(), getPlaneIoPtrs(), padBayer5Reference(), randNoiseFp16(), sippGetChunkStartPixelPos(), sippHorizontal-

Padding(), svuCensusMin64(), svuExtAfStats(), svuExtStatsSatPixelsU32(), svuGenDnsRef(), svuGenDnsRefFp16(), svuGenNoise(), svuHomography(), and svuUndistortBrown().

7.274 sippPlatformAbstractionLayer.c File Reference

```
#include <sippPalTypes.h>
#include <stdio.h>
#include <stdarg.h>
#include <string.h>
#include <VcsHooksApi.h>
#include <DrvIcb.h>
```

Macros

- #define [SIPP_PRT_BUFF_SIZE](#) 1024

Functions

- void [sippPalTraceInit](#) (void)
- void [sippPalTrace](#) (u32 uFlags, const char *psz_format,...)
- void [sippPalPrintInt](#) (u32 uVal)
- void [sippPalCriticalSectionBegin](#) (SIPP_PAL_CRIT_STATE *pState)
- void [sippPalCriticalSectionEnd](#) (SIPP_PAL_CRIT_STATE PreviousState)
- void * [sippPalMemcpy](#) (void *pDest, const void *pSrc, u32 uSize)
- void [sippPalMemset](#) (void *pDest, s32 nChar, u32 uCount)
- u8 [sippPalMemCompare](#) (void *pArea1, void *pArea2, u32 uSizeInWords)
- u32 [sippPalFindHighestBit](#) (u32 uValue)

7.274.1 Macro Definition Documentation

```
#define SIPP_PRT_BUFF_SIZE 1024
```

Referenced by [sippPalTrace\(\)](#).

7.274.2 Function Documentation

```
void sippPalCriticalSectionBegin ( SIPP_PAL_CRIT_STATE * pState )
```

```
void sippPalCriticalSectionEnd ( SIPP_PAL_CRIT_STATE PreviousState )
```

```
u32 sippPalFindHighestBit ( u32 uValue )
```

```
u8 sippPalMemCompare ( void * pArea1, void * pArea2, u32 uSizeInWords )
```

```
void* sippPalMemcpy ( void * pDest, const void * pSrc, u32 uSize )
```

```
void sippPalMemset ( void * pDest, s32 nChar, u32 uCount )
```

```
void sippPalPrintInt ( u32 uVal )
```

```
void sippPalTrace ( u32 uFlags, const char * psz_format, ... )
```

```
void sippPalTraceInit ( void )
```

7.275 sippPlatformAbstractionLayer.c File Reference

```
#include <sippPalTypes.h>
#include "sippPal.h"
#include <stdio.h>
#include <stdarg.h>
#include <string.h>
#include "wrapperSem.h"
```

Macros

- #define FALSE 0
- #define TRUE 1
- #define SIPP_PRT_BUFF_SIZE 1024

Functions

- void sippPalTraceInit (void)
- void sippPalTrace (u32 uFlags, const char *psz_format,...)
- void sippPalCriticalSectionBegin (SIPP_PAL_CRIT_STATE *pState)
- void sippPalCriticalSectionEnd (SIPP_PAL_CRIT_STATE PreviousState)
- void * sippPalMemcpy (void *pDest, const void *pSrc, u32 uSize)
- void sippPalMemset (void *pDest, s32 nChar, u32 uCount)
- u8 sippPalMemCompare (void *pArea1, void *pArea2, u32 uSizeInWords)
- u32 sippPalFindHighestBit (u32 uValue)

Variables

- Semaphore * palTraceSerialiseSem

7.275.1 Macro Definition Documentation

```
#define FALSE 0
```

```
#define SIPP_PRT_BUFF_SIZE 1024
```

Referenced by sippPalTrace().

```
#define TRUE 1
```

7.275.2 Function Documentation

```
void sippPalCriticalSectionBegin ( SIPP_PAL_CRIT_STATE * pState )
```

```
void sippPalCriticalSectionEnd ( SIPP_PAL_CRIT_STATE PreviousState )
```

```
u32 sippPalFindHighestBit ( u32 uValue )
```

```
u8 sippPalMemCompare ( void * pArea1, void * pArea2, u32 uSizeInWords )
```

```
void* sippPalMemcpy ( void * pDest, const void * pSrc, u32 uSize )
```

```
void sippPalMemset ( void * pDest, s32 nChar, u32 uCount )
```

```
void sippPalTrace ( u32 uFlags, const char * psz_format, ... )
```

```
void sippPalTraceInit ( void )
```

7.275.3 Variable Documentation

Semaphore* palTraceSerialiseSem

7.276 sippPlatformAbstractionLayer.c File Reference

SIPP engine.

```
#include <sipp.h>
#include <sippPalTypes.h>
#include <sippPal.h>
#include <stdio.h>
#include <stdarg.h>
#include <string.h>
#include <rtems.h>
#include <rtems/libio.h>
#include <rtems/bspIo.h>
#include <sys/stat.h>
#include <mqueue.h>
#include <pthread.h>
#include <sched.h>
#include <fcntl.h>
#include <VcsHooksApi.h>
```

Functions

- void [sippPalTraceInit](#) (void)
- void [sippPalTrace](#) (**u32** uFlags, const char *psz_format,...)
- void [sippPalPrintInt](#) (**u32** uVal)
- void [sippPalCriticalSectionBegin](#) (**SIPP_PAL_CRIT_STATE** *pState)

- void `sippPalCriticalSectionEnd` (`SIPP_PAL_CRIT_STATE` PreviousState)
- `eSIPP_STATUS` `sippPalQuCreate` (`uint32_t` maxElements, `const char *pszName`, `SIPP_PAL_QU_ID` pQu)
- `eSIPP_STATUS` `sippPalQuDestroy` (`SIPP_PAL_QU_ID` pQu)
- `eSIPP_STATUS` `sippPalQuAttach` (`const char *pszName`, `SIPP_PAL_QU_ID` pQu)
- `eSIPP_STATUS` `sippPalQuPost` (`SIPP_PAL_QU_ID` quId, `void *message`)
- `eSIPP_STATUS` `sippPalQuReceive` (`SIPP_PAL_QU_ID` quId, `SIPP_PAL_QU_MSG` *message)
- `eSIPP_STATUS` `sippPalThreadCreate` (`SIPP_PAL_PFTHREAD` pfnEntryPoint, `int` nArgC, `void **ppArgV`, `uint32_t` stackSize, `uint8_t` priority, `const char *pszName`, `SIPP_PAL_THREAD_ID` pThread)
- void `sippPalThreadTerminate` (`SIPP_PAL_THREAD_ID` pThread)
- void * `sippPalMemcpy` (`void *pDest`, `const void *pSrc`, `u32` uSize)
- void `sippPalMemset` (`void *pDest`, `s32` nChar, `u32` uCount)
- `u8` `sippPalMemCompare` (`void *pArea1`, `void *pArea2`, `u32` uSizeInWords)
- `u32` `sippPalFindHighestBit` (`u32` uValue)

7.276.1 Detailed Description

SIPP engine.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.276.2 Function Documentation

`void sippPalCriticalSectionBegin (SIPP_PAL_CRIT_STATE * pState)`

Referenced by `dmaKickSequenceConcurrent()`, `sippAccessSchedulerQuPush()`, `sippCreatePipeline()`, `sippDeletePipeline()`, `sippEventNotify()`, `sippGenericRuntimeLoadPipeline()`, `sippGenericRuntimeProcessIters()`, `sippHWSessionAddActiveLists()`, `sippHWSessionRemoveActiveLists()`, `sippHWSessionRemoveLoadedPipe()`, `sippPalQuPost()`, `sippPalQuReceive()`, and `sippProcessFrame()`.

`void sippPalCriticalSectionEnd (SIPP_PAL_CRIT_STATE PreviousState)`

Referenced by `dmaKickSequenceConcurrent()`, `sippAccessSchedulerQuPush()`, `sippCreatePipeline()`, `sippDeletePipeline()`, `sippEventNotify()`, `sippGenericRuntimeLoadPipeline()`, `sippGenericRuntimeProcessIters()`, `sippHWSessionAddActiveLists()`, `sippHWSessionRemoveActiveLists()`, `sippHWSessionRemoveLoadedPipe()`, `sippPalQuPost()`, `sippPalQuReceive()`, and `sippProcessFrame()`.

`u32 sippPalFindHighestBit (u32 uValue)`

`u8 sippPalMemCompare (void * pArea1, void * pArea2, u32 uSizeInWords)`

`void* sippPalMemcpy (void * pDest, const void * pSrc, u32 uSize)`

Referenced by `sippHWSessionRemoveActiveLists()`, and `sippIssueCommand()`.

```
void sippPalMemset ( void * pDest, s32 nChar, u32 uCount )
```

Referenced by sippAssignCmxMemRegion(), sippCreateFilter(), sippCreatePipeline(), sippGenericSchedAllocTempStorage(), sippHeapInit(), and sippInitPhysicalPoolGlobal().

```
void sippPalPrintInt ( u32 uVal )
```

```
eSIPP_STATUS sippPalQuAttach ( const char * pszName, SIPP_PAL_QU_ID pQu )
```

```
eSIPP_STATUS sippPalQuCreate ( uint32_t maxElements, const char * pszName,  
SIPP_PAL_QU_ID pQu )
```

Referenced by sippAccessSchedulerInit().

```
eSIPP_STATUS sippPalQuDestroy ( SIPP_PAL_QU_ID pQu )
```

```
eSIPP_STATUS sippPalQuPost ( SIPP_PAL_QU_ID quId, void * message )
```

Referenced by sippAccessSchedulerQuPush().

```
eSIPP_STATUS sippPalQuReceive ( SIPP_PAL_QU_ID quId, SIPP_PAL_QU_MSG * message )
```

```
eSIPP_STATUS sippPalThreadCreate ( SIPP_PAL_PFTHREAD pfnEntryPoint, int nArgC, void  
** ppArgV, uint32_t stackSize, uint8_t priority, const char * pszName, SIPP_PAL_THREAD_ID  
pThread )
```

```
void sippPalThreadTerminate ( SIPP_PAL_THREAD_ID pThread )
```

```
void sippPalTrace ( u32 uFlags, const char * psz_format, ... )
```

Referenced by sippAccessSchedulerQuPush(), sippAllocCmxLineBuffersOPipe(), sippCoreFinalisePipeline(), sippCoreReschedulePipeline(), sippCreateFilter(), sippCreatePipeline(), sippDbgCompareDeltaU8(), sippDbgCompareU16(), sippDbgCompareU32(), sippDbgCompareU8(), sippDbgDumpRunMask(), sippDbgShowBuffPtr(), sippDeletePipeline(), sippError(), sippFinalizePipeline(), sippGenericAllocRuntimeSched(), sippGenericBlockHWUnits2x5x(), sippGenericRunNextIter(), sippGenericRuntimeHWProcessIters(), sippGenericSchedule(), sippGenericUpdateHWUnits2x5x(), sippHeapAlloc(), sippHeapCheck(), sippHeapDefrag(), sippInitDma(), sippInitLnMemPool(), sippInitPipeline(), sippIntBarrierSetup(), sippMemAlloc(), sippMemCheck(), sippMemStatus(), sippPipeSessionControl(), sippPipeSetLinesPerIter(), sippPrintSliceWidth(), sippProcessFrame(), sippProcessFrameNB(), sippRegisterEventCallback(), sippReschedulePipeline(), and sippTestCrcCheck().

```
void sippPalTraceInit ( void )
```

Referenced by sippSWInit().

7.277 sippScheduleIsr.c File Reference

SIPP framework ISRs used by the schedule based runtimes in async mode Platform(s) supported - : ma2x5x.


```
#include <sipp.h>
#include <sippInternal.h>
```

Functions

- void [sippIbflDecHandler](#) (u32 irqSource)
- void [sippObflIncHandler](#) (u32 irqSource)
- void [sippSvuDoneIrqHandler](#) (u32 irqSource)
- void [sippCheckIterComplete](#) (SippPipeline *pPipe, u32 flag)

Variables

- [pSIPP_HW_SESSION](#) pgSippHW
- [u32 sippGlobalOBFLIncStatus](#) = 0

7.277.1 Detailed Description

SIPP framework ISRs used by the schedule based runtimes in async mode Platform(s) supported - : ma2x5x.

Copyright

All code copyright Movidius Ltd 2015, all rights reserved. For License Warranty see: common/license.txt

7.277.2 Function Documentation

void [sippCheckIterComplete](#) ([SippPipeline](#) * pPipe, [u32](#) flag)

Referenced by [sippCmxDmaDoneIrqHandler\(\)](#), [sippObflIncHandler\(\)](#), and [sippSvuDoneIrqHandler\(\)](#).

void [sippIbflDecHandler](#) ([u32](#) irqSource)

Referenced by [sippIsrSetup\(\)](#).

void [sippObflIncHandler](#) ([u32](#) irqSource)

Referenced by [sippIsrSetup\(\)](#).

void [sippSvuDoneIrqHandler](#) ([u32](#) irqSource)

Referenced by [sippKickSvus\(\)](#).

7.277.3 Variable Documentation

pSIPP_HW_SESSION pgSippHW

u32 sippGlobalOBFLIncStatus = 0

Referenced by sippGenericRuntimeLoadPipeline(), sippHWSessionRemoveActiveLists(), sippIntBarrierSetup(), and sippObflIncHandler().

7.278 sippSchTypes.h File Reference

```
#include <sippCfg.h>
#include <sippTypes.h>
```

Data Structures

- struct [sSchLineBufferS](#)
- struct [SippSchEntS](#)
- struct [SippFilterSchedInfoS](#)
- struct [SippManagedBufSchedInfo](#)

Typedefs

- typedef void(* [SchedAddCheck](#))(void *pPipe, void *pSE, void *pParam)
- typedef struct [SippManagedBufSchedInfo](#) * [pSippManagedBufSchedInfo](#)

Enumerations

- enum [RunStatus](#) { [RS_DONT_KNOW](#) = 0, [RS_CAN_RUN](#) = 1, [RS_CANNOT](#) = 2 }

7.278.1 Typedef Documentation

typedef struct [SippManagedBufSchedInfo](#) * [pSippManagedBufSchedInfo](#)

typedef void(* [SchedAddCheck](#))(void *pPipe, void *pSE, void *pParam)

7.278.2 Enumeration Type Documentation

enum [RunStatus](#)

Enumerator

RS_DONT_KNOW
RS_CAN_RUN
RS_CANNOT

7.279 sippSessionControl.h File Reference

SIPP framework API Platform(s) supported : ma2x5x.

Macros

- #define [SIPP_NULL_PIPELINE_ID](#) 0xFF

Enumerations

- enum [SIPP_PIPE_STATE](#) { [SIPP_PIPE_WAIT_INIT](#) = 0x0, [SIPP_PIPE_WAIT_FINALISE](#), [SIPP_PIPE_ACTIVE](#), [SIPP_PIPE_END_SESSION](#) }

7.279.1 Detailed Description

SIPP framework API Platform(s) supported : ma2x5x.

Copyright

All code copyright Movidius Ltd 2015, all rights reserved. For License Warranty see: common/license.txt

7.279.2 Macro Definition Documentation

```
#define SIPP_NULL_PIPELINE_ID 0xFF
```

7.279.3 Enumeration Type Documentation

```
enum SIPP\_PIPE\_STATE
```

Enumerator

```
SIPP_PIPE_WAIT_INIT
SIPP_PIPE_WAIT_FINALISE
SIPP_PIPE_ACTIVE
SIPP_PIPE_END_SESSION
```

7.280 sippShave.c File Reference

SIPP engine.

```
#include <sipp.h>
#include <sippInternal.h>
```

Functions

- void [sippHorizontalPadding](#) ([SippPipeline](#) *pl)

- int `scGetShaveNumber` ()
- `u32 sippGetChunkStartPixelPos` (`SippFilter` *fptr, `u32` *YPos, `u32` *XPos)
- int `SHAVE_MAIN` (void)

Variables

- `SippPipeline` * `sipp_pl`
- `u32 dbg_svu_no`

7.280.1 Detailed Description

SIPP engine.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.280.2 Function Documentation

`int scGetShaveNumber` ()

Referenced by `configureBicubicHWblock()`, `genXYlist()`, `getIn3PlanePtr()`, `getInPtr()`, `getOutPtr()`, `getPlaneIoPtrs()`, `padBayer5Reference()`, `randNoiseFp16()`, `sippGetChunkStartPixelPos()`, `sippHorizontalPadding()`, `svuCensusMin64()`, `svuExtAfStats()`, `svuExtStatsSatPixelsU32()`, `svuGenDnsRef()`, `svuGenDnsRefFp16()`, `svuGenNoise()`, `svuHomography()`, and `svuUndistortBrown()`.

`int SHAVE_MAIN` (void)

`u32 sippGetChunkStartPixelPos` (`SippFilter` * fptr, `u32` * YPos, `u32` * XPos)

`void sippHorizontalPadding` (`SippPipeline` * pl)

Referenced by `SHAVE_MAIN()`.

7.280.3 Variable Documentation

`u32 dbg_svu_no`

Referenced by `scGetShaveNumber()`, and `sippKickShaveM1PC()`.

`SippPipeline`* `sipp_pl`

Referenced by `sippKickSvus()`.

7.281 sippShaveIf.c File Reference

SIPP engine.

```
#include <sipp.h>
#include <sippInternal.h>
```

Functions

- int **SVU_SYM()** **SHAVE_MAIN** (void)
- void **sippStopSvus** (**SippPipeline** *pl)
- void **sippSetupSvus** (**SippPipeline** *pl)
- void **sippKickShaveM1PC** (**SippPipeline** *pl)
- void **sippWaitShave** (**SippPipeline** *pl)
- void **sippKickSvus** (**SippPipeline** *pl)

7.281.1 Detailed Description

SIPP engine.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.281.2 Function Documentation

int **SVU_SYM()** **SHAVE_MAIN** (void)

void **sippKickShaveM1PC** (**SippPipeline** * pl)

Referenced by **sippGenericStartUnits()**.

void **sippKickSvus** (**SippPipeline** * pl)

void **sippSetupSvus** (**SippPipeline** * pl)

void **sippStopSvus** (**SippPipeline** * pl)

Referenced by **sippHWSessionRemoveActiveLists()**.

void **sippWaitShave** (**SippPipeline** * pl)

7.282 sippShaveMacros.h File Reference

SIPP engine.

```
#include <sipp.h>
```

7.282.1 Detailed Description

SIPP engine.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.283 sippShaveSym.h File Reference

7.284 sippShvDbg.c File Reference

SIPP engine.

```
#include <sipp.h>
#include <sippInternal.h>
```

Functions

- void [sippShaveDebug](#) ([SippPipeline](#) *pl, [u32](#) val)

Variables

- [SippPipeline](#) * [sipp_pl](#)

7.284.1 Detailed Description

SIPP engine.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.284.2 Function Documentation

```
void sippShaveDebug ( SippPipeline * pl, u32 val )
```

7.284.3 Variable Documentation

[SippPipeline](#)* [sipp_pl](#)

7.285 sippTestCommon.c File Reference

SIPP engine.

```
#include <sipp.h>
#include <sippTestCommon.h>
```

Functions

- void [sippPlatformInit](#) ()
- void [sippPlatformInitAsync](#) (void)

7.285.1 Detailed Description

SIPP engine.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.285.2 Function Documentation

[void sippPlatformInit \(void \)](#)

[void sippPlatformInitAsync \(void \)](#)

7.286 sippTestCommon.h File Reference

SIPP engine.

7.286.1 Detailed Description

SIPP engine.

Copyright

All code copyright Movidius Ltd 2015, all rights reserved. For License Warranty see: [common/license.txt](#)

7.287 sippTestCommon_ma2x5x.h File Reference

SIPP engine.

Functions

- void [sippPlatformInit](#) (void)
- void [sippPlatformInitAsync](#) (void)

Variables

- [UInt8 mbinImgSipp \[\]](#)

7.287.1 Detailed Description

SIPP engine.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.287.2 Function Documentation

[void sippPlatformInit \(void \)](#)

[void sippPlatformInitAsync \(void \)](#)

7.287.3 Variable Documentation

[UInt8 mbinImgSipp\[\]](#)

7.288 sippThread.c File Reference

SIPP engine, RTOS thread.

```
#include <sipp.h>
#include <sippInternal.h>
```

7.288.1 Detailed Description

SIPP engine, RTOS thread.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.289 sippTypes.h File Reference

```
#include <sippBaseTypes.h>
#include <sippAccessSchedulerTypes.h>
#include <sippEvents.h>
#include <mv_types.h>
```


Data Structures

- struct [CommInfo](#)
- struct [HorizPaddingOffS](#)
- struct [HPadInfoS](#)
- struct [SchedInfoS](#)
- struct [ParentInfoS](#)
- struct [DmaDesc](#)
- union [DMAPartialCfg](#)
- union [DMAExtCfg](#)
- struct [SippMemRegion](#)
- struct [tSippMCB](#)
- struct [SippPixelChunkPos](#)
- struct [sSippCdmaQuEntryS](#)
- struct [sSippCdmaQuS](#)
- struct [sSippCMDQuEntryS](#)
- struct [sSippCMDQuS](#)
- struct [tRTStats](#)
- struct [SippPipelineS](#)
- struct [SippFilterS](#)
- struct [ae_patch_stats](#)
- struct [AF_paxel_statistics](#)

Macros

- #define [NULL](#) 0
- #define [FALSE](#) 0
- #define [TRUE](#) 1
- #define [SIPP_ERROR_MASK_SIZE](#) ((E_LAST/32)+1)

Typedefs

- typedef struct [SippFilterS](#) [SippFilter](#)
- typedef struct [SippFilterS](#) * [pSippFilter](#)
- typedef struct [SippPipelineS](#) [SippPipeline](#)
- typedef struct [SippPipelineS](#) * [pSippPipeline](#)
- typedef struct [SippSchEntS](#) [SippSchEnt](#)
- typedef struct [SippHwBufS](#) [SippHwBuf](#)
- typedef struct [SippCmxBufferMapS](#) [SippCmxBufferMap](#)
- typedef struct [SippCmxBufferMapS](#) * [pSippCmxBufferMap](#)
- typedef struct [SippVPhysMapS](#) [tSippVPhysMap](#)
- typedef struct [SippVPhysMapS](#) * [pSippVPhysMap](#)
- typedef struct [SippFilterSchedInfoS](#) [SippFilterSchedInfo](#)
- typedef struct [SippFilterSchedInfoS](#) * [pSippFilterSchedInfo](#)
- typedef [u8](#)(* [SchedFuncAsk](#))([SippSchEnt](#) *ptrSE, [u32](#) iBufId, [s32](#) line_no, [u32](#) linesPerIter)
- typedef void(* [FnSvuRun](#))([SippFilter](#) *fptr)

- typedef void(* SippCallback)(SippPipeline *sourcePl)
- typedef struct sSchLineBufferS sSchLineBuffer
- typedef struct sSchLineBufferS * psSchLineBuffer
- typedef struct sSchedIBufUsageInfoS SchedIBufUsageInfo
- typedef struct sSippCdmaQuS sSippCdmaQu
- typedef struct sSippCdmaQuS * psSippCdmaQu
- typedef struct sSippCdmaQuEntryS sSippCdmaQuEntry
- typedef struct sSippCdmaQuEntryS * psSippCdmaQuEntry
- typedef struct sSippCMDQuS sSippCMDQu
- typedef struct sSippCMDQuS * psSippCMDQu
- typedef struct sSippCMDQuEntryS sSippCMDQuEntry
- typedef struct sSippCMDQuEntryS * psSippCMDQuEntry
- typedef struct HorizPaddingOffS HorizPaddingOff
- typedef struct HPadInfoS HPadInfo
- typedef struct SchedInfoS SchedInfo
- typedef struct ParentInfoS ParentInfo
- typedef struct tSippMCB * ptSippMCB
- typedef void(* sippEventCallback_t)(SippPipeline *pPipeline, eSIPP_PIPELINE_EVENT e-Event, SIPP_PIPELINE_EVENT_DATA *ptEventData)
- typedef u32(* sippSchedFunc)(pSippPipeline pipeLine, bool allocMem, bool reschedPipe)
- typedef void(* sippSchedSetBufConsModels)(pSippPipeline pPipe)
- typedef void(* sippRuntimeClaimHWResourceFunc)(pSippPipeline pPipe)
- typedef void(* sippRuntimeFunc)(pSippPipeline pPipe, eSIPP_ACCESS_SCHEDULER_EVE-NT eEvent, SIPP_ACCESS_SCHEDULER_EVENT_DATA pData)
- typedef u32(* sippRunIterDoneFunc)(pSippPipeline pPipe)
- typedef void(* sippRunNextIterFunc)(pSippPipeline pPipe)
- typedef void(* sippRunFrameReset)(pSippPipeline pPipe)
- typedef struct tRTStats * ptRTStats

Enumerations

- enum eSIPP_STATUS {
eSIPP_STATUS_OK = 0, eSIPP_STATUS_ALREADY_INIT, eSIPP_STATUS_NOT_INIT, e-
SIPP_STATUS_INTERNAL_ERROR,
eSIPP_STATUS_BAD_HANDLE, eSIPP_STATUS_BAD_PARAMETER, eSIPP_STATUS_B-
AD_LENGTH, eSIPP_STATUS_BAD_UNIT,
eSIPP_STATUS_RESOURCE_ERROR, eSIPP_STATUS_CLOSED_HANDLE, eSIPP_STAT-
US_TIMEOUT, eSIPP_STATUS_NOT_ATTACHED,
eSIPP_STATUS_NOT_SUPPORTED, eSIPP_STATUS_REOPENED_HANDLE, eSIPP_STA-
TUS_INVALID, eSIPP_STATUS_DESTROYED,
eSIPP_STATUS_DISCONNECTED, eSIPP_STATUS_BUSY, eSIPP_STATUS_IN_USE, eSIP-
P_STATUS_CANCELLED,
eSIPP_STATUS_UNDEFINED, eSIPP_STATUS_UNKNOWN, eSIPP_STATUS_NOT_FOU-
ND, eSIPP_STATUS_NOT_AVAILABLE,
eSIPP_STATUS_NOT_COMPATIBLE, eSIPP_STATUS_NOT_IMPLEMENTED, eSIPP_ST-
ATUS_EMPTY, eSIPP_STATUS_FULL,
eSIPP_STATUS_FAILURE, eSIPP_STATUS_ALREADY_ATTACHED, eSIPP_STATUS_A-
LREADY_DONE, eSIPP_STATUS_ASLEEP,
eSIPP_STATUS_BAD_ATTACHMENT, eSIPP_STATUS_BAD_COMMAND, eSIPP_STAT-

```

US_INT_HANDLED, eSIPP_STATUS_INT_NOT_HANDLED,
eSIPP_STATUS_NOT_SET, eSIPP_STATUS_NOT_HOOKED, eSIPP_STATUS_COMPLET-
E, eSIPP_STATUS_INVALID_NODE,
eSIPP_STATUS_DUPLICATE_NODE, eSIPP_STATUS_HARDWARE_NOT_FOUND, eSIP-
P_STATUS_ILLEGAL_OPERATION, eSIPP_STATUS_INCOMPATIBLE_FORMATS,
eSIPP_STATUS_INVALID_DEVICE, eSIPP_STATUS_INVALID_EDGE, eSIPP_STATUS_I-
NVALID_NUMBER, eSIPP_STATUS_INVALID_STATE,
eSIPP_STATUS_INVALID_TYPE, eSIPP_STATUS_STOPPED, eSIPP_STATUS_SUSPEND-
ED, eSIPP_STATUS_TERMINATED,
eSIPP_STATUS_CODE_LAST = eSIPP_STATUS_TERMINATED }
• enum {
  E_SUCCESS = 0, E_OUT_OF_MEM = 1, E_INVALID_MEM_P = 2, E_PAR_NOT_FOUND =
  3,
  E_DATA_NOT_FOUND = 4, E_RUN_DON_T_KNOW = 5, E_INVALID_HW_PARAM = 6,
  E_INVLD_FILT_FIRST_SLICE = 7,
  E_INVLD_FILT_LAST_SLICE = 8, E_MISSING_SHAVE_IMAGE = 9, E_UNIMPLEMENT-
  ED_FEAT = 10, E_PC_CMx_MEM_ALLOC_ERR = 11,
  E_OPT_EXEC_NUM = 12, E_CANNOT_FINISH_FILTER = 13, E_DATA_ALIGN = 14, E_I-
  NVLD_MIPI_RX_LOOPBACK = 15,
  E_TOO_MANY_FILTERS = 16, E_INVLD_MULTI_INSTANCE = 17, E_INVLD_HW_ID =
  18, E_TOO_MANY_PARENTS = 19,
  E_TOO_MANY_CONSUMERS = 20, E_RUNS_ITER_GROUPS = 21, E_TOO_MANY_DM-
  AS = 22, E_INVLD_SLICE_WIDTH = 23,
  E_OSE_CREATION_ERROR = 24, E_CDMA_QU_OVERFLOW = 25, E_PC_RUNTIME_FA-
  ILURE = 26, E_SCHEDULING_OVF = 27,
  E_BLOCK_CALL_REJECTED = 28, E_PRECOMP_SCHED = 29, E_FINALISE_FAIL = 30,
  E_HEAP_CREATION_FAIL = 31,
  E_LAST = E_HEAP_CREATION_FAIL }
• enum eDmaMode { DMA_MODE_STANDARD = 0x0, DMA_MODE_PARTIAL_LINE }
• enum SippVirtualPool {
  vPoolGeneral = 0, vPoolPipeStructs, vPoolFilterLineBuf, vPoolCMxDMADesc,
  vPoolSchedule, vPoolScheduleTemp, vPoolFilterLineBuf0, vPoolFilterLineBuf1,
  vPoolFilterLineBuf2, vPoolFilterLineBuf3, vPoolFilterLineBuf4, vPoolFilterLineBuf5,
  vPoolFilterLineBuf6, vPoolFilterLineBuf7, vPoolFilterLineBuf8, vPoolFilterLineBuf9,
  vPoolFilterLineBuf10, vPoolFilterLineBuf11, vPoolLast }

```

7.289.1 Macro Definition Documentation

```
#define FALSE 0
```

Referenced by sippAccessSchedulerInit(), sippAccessSchedulerQuPush(), sippAllocCmxLineBuffers-
OPipe(), sippAnalysePipe2x5x(), sippCreateFilter(), sippCreatePipeline(), sippEventNotify(), sipp-
FinalizePipeline(), sippHWSessionCommand(), sippInitialize(), sippIssueCommand(), sippMemFind-
MaxLnMemPoolFree(), sippMemLBConsolidateRegions(), sippMemLBRemoveNode(), sippOSE-
ProcessFilterIBufs(), sippPipeSessionControl(), sippPipeSetLinesPerIter(), sippPlatformInit(), sipp-
ProcessFrame(), sippProcessFrameNB(), sippRegisterEventCallback(), sippReschedulePipeline(), sipp-
SWInit(), sippTerm(), sippValidatePipe(), and topLevelCmxDmaIrqHandler().

```
#define NULL 0
```

Referenced by dmaKickSequenceConcurrent(), minMaxKernel(), Semaphore::Post(), Semaphore::Semaphore(), sippAccessSchedulerControl(), sippAlloc(), sippAssert(), sippCmxDmaDoneIrqHandler(), sippComputeBufferProps(), sippCoreFinalisePipeline(), sippCoreReschedulePipeline(), sippCreateFilter(), sippCreatePipeline(), sippDbgCreateDumpFiles(), sippDbgDumpGraph(), sippDbgDumpRunMask(), sippDbgDumpSchedForVcsCarr(), sippDeletePipeline(), sippElaboratePipeline(), sippFilterAddOBuf(), sippFreePipeResource(), sippGenericAllocRuntimeSched(), sippGenericRecordParentKS(), sippGenericRuntimeProcessIters(), sippGenericSchedAllocTempStorage(), sippGenericSchedule(), sippHeapAlloc(), sippHeapCreate(), sippHWInit(), sippHWSessionInit(), sippHWSessionRemoveActiveLists(), sippHWSessionRemoveLoadedPipe(), sippIbufSetup(), sippInitDma(), sippInitLnMemPool(), sippInitLnMemPoolSlices(), sippInitPhysicalPoolGlobal(), sippInitPhysicalPoolPipe(), sippInitPipeline(), sippInitSchedPool(), sippIssueCommand(), sippMemAlloc(), sippMemLBRemoveNode(), sippOBufSetup(), sippOSECreate(), sippPalMemcpy(), sippPalQuAttach(), sippPlatformInit(), sippRdFileU8(), sippRdFileU8toF16(), sippRescheduleRequest(), sippUtilPrintFp16Lut(), and sippWrFileU8().

```
#define SIPP_ERROR_MASK_SIZE ((E_LAST/32)+1)
```

Referenced by sippPipeGetErrorStatus().

```
#define TRUE 1
```

Referenced by sippAccessSchedulerInit(), sippAccessSchedulerQuPush(), sippAllocCmxLineBuffersOPipe(), sippAnalysePipe2x5x(), sippCreatePipeline(), sippHWSessionCommand(), sippInitialize(), sippMemFindMaxLnMemPoolFree(), sippMemLBConsolidateRegions(), sippMemLBRemoveNode(), sippOSEProcessFilterIBufs(), sippPipeSessionControl(), sippSWInit(), sippValidatePipe(), and topLevelCmxDmaIrqHandler().

7.289.2 Typedef Documentation

```
typedef void(* FnSvuRun)(SippFilter *fptr)
```

```
typedef struct HorizPaddingOffS HorizPaddingOff
```

```
typedef struct HPadInfoS HPadInfo
```

```
typedef struct ParentInfoS ParentInfo
```

```
typedef struct SippCmxBufferMapS* pSippCmxBufferMap
```

```
typedef struct SippFilterS * pSippFilter
```

```
typedef struct SippFilterSchedInfoS * pSippFilterSchedInfo
```

```
typedef struct SippPipelineS * pSippPipeline
```

```
typedef struct SippVPhysMapS * pSippVPhysMap
```

```
typedef struct sSchLineBufferS * psSchLineBuffer
```

```

typedef struct sSippCdmaQuS * psSippCdmaQu

typedef struct sSippCdmaQuEntryS * psSippCdmaQuEntry

typedef struct sSippCMDQuS * psSippCMDQu

typedef struct sSippCMDQuEntryS * psSippCMDQuEntry

typedef struct tRTStats * ptRTStats

typedef struct tSippMCB * ptSippMCB

typedef u8(* SchedFuncAsk)(SippSchEnt *ptrSE, u32 iBufId, s32 line_no, u32 linesPerIter)

typedef struct sSchedIBufUsageInfoS SchedIBufUsageInfo

typedef struct SchedInfoS SchedInfo

typedef void(* SippCallback)(SippPipeline *sourcePl)

typedef struct SippCmxBufferMapS SippCmxBufferMap

typedef void( * sippEventCallback_t)(SippPipeline *pPipeline, eSIPP_PIPELINE_EVENT eEvent,
SIPP_PIPELINE_EVENT_DATA *ptEventData)

typedef struct SippFilterS SippFilter

typedef struct SippFilterSchedInfoS SippFilterSchedInfo

typedef struct SippHwBufS SippHwBuf

typedef struct SippPipelineS SippPipeline

typedef void( * sippRunFrameReset)(pSippPipeline pPipe)

typedef u32( * sippRunIterDoneFunc)(pSippPipeline pPipe)

typedef void( * sippRunNextIterFunc)(pSippPipeline pPipe)

typedef void( * sippRuntimeClaimHWResourceFunc)(pSippPipeline pPipe)

typedef void( * sippRuntimeFunc)(pSippPipeline pPipe, eSIPP_ACCESS_SCHEDULER_EVENT
eEvent, SIPP_ACCESS_SCHEDULER_EVENT_DATA pData)

typedef u32( * sippSchedFunc)(pSippPipeline pipeLine, bool allocMem, bool reschedPipe)

typedef void( * sippSchedSetBufConsModels)(pSippPipeline pPipe)

typedef struct SippSchEnts SippSchEnt

typedef struct sSchLineBufferS sSchLineBuffer

```

```
typedef struct sSippCdmaQuS sSippCdmaQu
```

```
typedef struct sSippCdmaQuEntryS sSippCdmaQuEntry
```

```
typedef struct sSippCMDQuS sSippCMDQu
```

```
typedef struct sSippCMDQuEntryS sSippCMDQuEntry
```

```
typedef struct SippVPhysMapS tSippVPhysMap
```

7.289.3 Enumeration Type Documentation

anonymous enum

Enumerator

E_SUCCESS
E_OUT_OF_MEM
E_INVALID_MEM_P
E_PAR_NOT_FOUND
E_DATA_NOT_FOUND
E_RUN_DON_T_KNOW
E_INVALID_HW_PARAM
E_INVLD_FILT_FIRST_SLICE
E_INVLD_FILT_LAST_SLICE
E_MISSING_SHAVE_IMAGE
E_UNIMPLEMENTED_FEAT
E_PC_CMX_MEM_ALLOC_ERR
E_OPT_EXEC_NUM
E_CANNOT_FINISH_FILTER
E_DATA_ALIGN
E_INVLD_MIPI_RX_LOOPBACK
E_TOO_MANY_FILTERS
E_INVLD_MULTI_INSTANCE
E_INVLD_HW_ID
E_TOO_MANY_PARENTS
E_TOO_MANY_CONSUMERS
E_RUNS_ITER_GROUPS
E_TOO_MANY_DMAS
E_INVLD_SLICE_WIDTH
E_OSE_CREATION_ERROR
E_CDMA_QU_OVERFLOW
E_PC_RUNTIME_FAILURE
E_SCHEDULING_OVF
E_BLOCK_CALL_REJECTED

E_PRECOMP_SCHED
E_FINALISE_FAIL
E_HEAP_CREATION_FAIL
E_LAST

enum **eDmaMode**

Enumerator

DMA_MODE_STANDARD
DMA_MODE_PARTIAL_LINE

enum **eSIPP_STATUS**

Enumerator

eSIPP_STATUS_OK
eSIPP_STATUS_ALREADY_INIT
eSIPP_STATUS_NOT_INIT
eSIPP_STATUS_INTERNAL_ERROR
eSIPP_STATUS_BAD_HANDLE
eSIPP_STATUS_BAD_PARAMETER
eSIPP_STATUS_BAD_LENGTH
eSIPP_STATUS_BAD_UNIT
eSIPP_STATUS_RESOURCE_ERROR
eSIPP_STATUS_CLOSED_HANDLE
eSIPP_STATUS_TIMEOUT
eSIPP_STATUS_NOT_ATTACHED
eSIPP_STATUS_NOT_SUPPORTED
eSIPP_STATUS_REOPENED_HANDLE
eSIPP_STATUS_INVALID
eSIPP_STATUS_DESTROYED
eSIPP_STATUS_DISCONNECTED
eSIPP_STATUS_BUSY
eSIPP_STATUS_IN_USE
eSIPP_STATUS_CANCELLED
eSIPP_STATUS_UNDEFINED
eSIPP_STATUS_UNKNOWN
eSIPP_STATUS_NOT_FOUND
eSIPP_STATUS_NOT_AVAILABLE
eSIPP_STATUS_NOT_COMPATIBLE
eSIPP_STATUS_NOT_IMPLEMENTED

eSIPP_STATUS_EMPTY
eSIPP_STATUS_FULL
eSIPP_STATUS_FAILURE
eSIPP_STATUS_ALREADY_ATTACHED
eSIPP_STATUS_ALREADY_DONE
eSIPP_STATUS_ASLEEP
eSIPP_STATUS_BAD_ATTACHMENT
eSIPP_STATUS_BAD_COMMAND
eSIPP_STATUS_INT_HANDLED
eSIPP_STATUS_INT_NOT_HANDLED
eSIPP_STATUS_NOT_SET
eSIPP_STATUS_NOT_HOOKED
eSIPP_STATUS_COMPLETE
eSIPP_STATUS_INVALID_NODE
eSIPP_STATUS_DUPLICATE_NODE
eSIPP_STATUS_HARDWARE_NOT_FOUND
eSIPP_STATUS_ILLEGAL_OPERATION
eSIPP_STATUS_INCOMPATIBLE_FORMATS
eSIPP_STATUS_INVALID_DEVICE
eSIPP_STATUS_INVALID_EDGE
eSIPP_STATUS_INVALID_NUMBER
eSIPP_STATUS_INVALID_STATE
eSIPP_STATUS_INVALID_TYPE
eSIPP_STATUS_STOPPED
eSIPP_STATUS_SUSPENDED
eSIPP_STATUS_TERMINATED
eSIPP_STATUS_CODE_LAST

enum SippVirtualPool

Enumerator

vPoolGeneral
vPoolPipeStructs
vPoolFilterLineBuf
vPoolCMXDMADesc
vPoolSchedule
vPoolScheduleTemp
vPoolFilterLineBuf0
vPoolFilterLineBuf1
vPoolFilterLineBuf2

vPoolFilterLineBuf3
vPoolFilterLineBuf4
vPoolFilterLineBuf5
vPoolFilterLineBuf6
vPoolFilterLineBuf7
vPoolFilterLineBuf8
vPoolFilterLineBuf9
vPoolFilterLineBuf10
vPoolFilterLineBuf11
vPoolLast

7.290 sippTypesPrivate.h File Reference

```

#include <sippCfg.h>
#include <sippTypes.h>
#include <sippAccessSchedulerTypes.h>
#include <sippSchTypes.h>
#include "DrvSippDefines.h"

```

Data Structures

- struct [tSippPipelineSuper](#)
- struct [tSippFramework](#)
- struct [UnitInfo](#)
- struct [SippGlobals](#)
- struct [SippHwBufS](#)
- struct [SIPP_HW_SESSION](#)
- struct [SippMemRegionListNode](#)
- struct [SippCmxBufferMapS](#)
- struct [cmxRegUsage](#)
- struct [memRegDescriptor](#)
- struct [sSchedIBufUsageInfoS](#)
- struct [SippOseS](#)
- struct [tsSippHeap](#)
- struct [SippHeapCB](#)
- struct [tSippPhysicalPool](#)
- struct [SippVPhysMapS](#)
- struct [tMLPISStartCQCtrl](#)
- struct [sippOpipeBufInfo](#)

Macros

- #define [SIPP_HCB_SIZE](#) 8

Typedefs

- typedef struct [tSippPipelineSuper](#) * [ptSippPipelineSuper](#)
- typedef [u32](#)(* [FnHwFltInit](#))([SippFilter](#) *fptr)
- typedef void(* [FnHwFltLoad](#))([SippFilter](#) *fptr)
- typedef [u32](#)(* [FnGetIBufCtx](#))([SippFilter](#) *fptr, [u32](#) bufIdx)
- typedef [u32](#)(* [FnGetIBufIds](#))([SippFilter](#) *fptr, [u32](#) bufIdx)
- typedef [u32](#)(* [FnGetOBufIds](#))([SippFilter](#) *fptr, [u32](#) bufIdx)
- typedef [u32](#)(* [FnCheckOpPipeCons](#))([SippFilter](#) *fptr, [SippFilter](#) *parfptr, [u32](#) *parOBufIdx)
- typedef void(* [FnHwSetOBufLatencies](#))([SippFilter](#) *fptr)
- typedef void(* [FnSetOBufs](#))([SippFilter](#) *fptr, [eSippOBufControl](#) eOBufSetting)
- typedef void(* [FnHwSetupUpdate](#))([SippFilter](#) *fptr)
- typedef void * [SippCommandData](#)
- typedef struct [SIPP_HW_SESSION](#) * [pSIPP_HW_SESSION](#)
- typedef struct [SippMemRegionListNode](#) * [pSippMemRegionListNode](#)
- typedef struct [memRegDescriptor](#) * [pmemRegDescriptor](#)
- typedef struct [SippHeapCB](#) * [pSippHeapCB](#)
- typedef struct [tSippPhysicalPool](#) * [ptSippPhysicalPool](#)
- typedef struct [tMLPIStartCQCtrl](#) * [ptMLPIStartCQCtrl](#)

Enumerations

- enum [eSippPipeState](#) { [eSIPP_PIPE_WAIT_INIT](#) = 0x0, [eSIPP_PIPE_WAIT_FINALISE](#), [eSIPP_PIPE_ACTIVE](#), [eSIPP_PIPE_END_SESSION](#) }
- enum [eSippOBufControl](#) { [eSIPP_SET_OBUF_SPACE_FULL](#) = 0x0, [eSIPP_SET_OBUF_SPACE_EMPTY](#), [eSIPP_SET_OBUF_SPACE_ITER](#) }
- enum [eSippCommand](#) { [eSIPP_CMD_INIT_HW](#) = 0x0, [eSIPP_CMD_FINALISE_PIPE_SW](#), [eSIPP_CMD_RESCHEDULE_PIPE_SW](#), [eSIPP_CMD_DELETE_PIPE_SW](#), [eSIPP_CMD_PROCESS_ITERS_HW](#) }
- enum [eSIPP_HW_STATE](#) { [eSIPP_HW_INACTIVE](#) = 0, [eSIPP_HW_ACTIVE](#) }

Variables

- [SippGlobals](#) [gSipp](#)

7.290.1 Macro Definition Documentation

[#define SIPP_HCB_SIZE](#) 8

Referenced by [sippAlloc\(\)](#), [sippFree\(\)](#), [sippFreeList\(\)](#), [sippHeapAlloc\(\)](#), and [sippSizeList\(\)](#).

7.290.2 Typedef Documentation

typedef **u32**(* FnCheckOpipeCons)(**SippFilter** *fptr, **SippFilter** *parfptr, **u32** *parOBufIdx)

typedef **u32**(* FnGetIBufCtx)(**SippFilter** *fptr, **u32** bufIdx)

typedef **u32**(* FnGetIBufIds)(**SippFilter** *fptr, **u32** bufIdx)

typedef **u32**(* FnGetOBufIds)(**SippFilter** *fptr, **u32** bufIdx)

typedef **u32**(* FnHwFltInit)(**SippFilter** *fptr)

typedef void(* FnHwFltLoad)(**SippFilter** *fptr)

typedef void(* FnHwSetObufLatencies)(**SippFilter** *fptr)

typedef void(* FnHwSetupUpdate)(**SippFilter** *fptr)

typedef void(* FnSetOBufs)(**SippFilter** *fptr, **eSippObufControl** eOBufSetting)

typedef struct **memRegDescriptor** * **pmemRegDescriptor**

typedef struct **SIPP_HW_SESSION** * **pSIPP_HW_SESSION**

typedef struct **SippHeapCB** * **pSippHeapCB**

typedef struct **SippMemRegionListNode** * **pSippMemRegionListNode**

typedef struct **tMLPISstartCQCtrl** * **ptMLPISstartCQCtrl**

typedef struct **tSippPhysicalPool** * **ptSippPhysicalPool**

typedef struct **tSippPipelineSuper** * **ptSippPipelineSuper**

typedef void* **SippCommandData**

7.290.3 Enumeration Type Documentation

enum **eSIPP_HW_STATE**

Enumerator

eSIPP_HW_INACTIVE

eSIPP_HW_ACTIVE

enum **eSippCommand**

Enumerator

eSIPP_CMD_INIT_HW

eSIPP_CMD_FINALISE_PIPE_SW

eSIPP_CMD_RESCHEDULE_PIPE_SW

eSIPP_CMD_DELETE_PIPE_SW
eSIPP_CMD_PROCESS_ITERS_HW

enum eSippObufControl

Enumerator

eSIPP_SET_OBUF_SPACE_FULL
eSIPP_SET_OBUF_SPACE_EMPTY
eSIPP_SET_OBUF_SPACE_ITER

enum eSippPipeState

Enumerator

eSIPP_PIPE_WAIT_INIT
eSIPP_PIPE_WAIT_FINALISE
eSIPP_PIPE_ACTIVE
eSIPP_PIPE_END_SESSION

7.290.4 Variable Documentation

SippGlobals gSipp

Referenced by sippAnalysePipe2x5x(), sippCoreUnitLoad(), sippCreateFilter(), sippFindConnections-ListRecursive(), sippGenericBlockHWUnits2x5x(), sippGenericRuntimeLoadPipeline(), sippIdentify-OPipeSchedulingEntity(), sippIniHwFilters(), sippOSEProcessFilterIBufs(), sippOSEProcessFilterO-Bufs(), and sippValidatePipe().

7.291 sippUtils.c File Reference

SIPP engine.

```
#include <sipp.h>
#include <sippInternal.h>
```

Macros

- #define **CCM_COEF**(x) (**u16**)(x*1024)

Functions

- void **packConv5x5CCM** (**ConvParam** *cfg, **u16** *ccm5x5)
- void **packConv3x3CCM** (**ConvParam** *cfg, **u16** *ccm3x3)
- void **packColCombCCM** (**ColCombParam** *cfg, float *ccm3x3)

- void `packLumaDnsGaussLut` (`YDnsParam *cfg`, `u8 *lut`)
- void `sippUtilComputeFp16Lut` (`half(*formula)(half input)`, `half *outLut`, `u32 lutSize`)
- void `sippUtilComputeFp16LutChannelMode` (`half(*formula)(half input)`, `half *outLut`, `u32 lutSize`)
- void `sippUtilPrintFp16Lut` (`half *fp16Lut`, `u32 lutSize`, `const char *fName`)
- void `sharpenSigmaToCoefficients` (`float sigma`, `u16 *coeffs`)
- void `lumaGenLut` (`float strength`, `u8 *lut`, `int *bitpos`)
- void `sippWait` (`u32 numx100`)
- void `sippUtilOrderPixels` (`u32 bayerPattern`, `u32 inGr`, `u32 inR`, `u32 inB`, `u32 inGb`, `u32 *out`)
- void `sippListSort` (`s32 *pnList`, `s32 *pnSortIndices`, `u32 uSize`, `u8 descending`)
- `s32 sippFindInList` (`void *pElement`, `void **ppList`, `u32 numList`)

7.291.1 Detailed Description

SIPP engine.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: `common/license.txt`

7.291.2 Macro Definition Documentation

```
#define CCM_COEF( x ) (u16)(x*1024)
```

Referenced by `packColCombCCM()`.

7.291.3 Function Documentation

```
void lumaGenLut ( float strength, u8 * lut, int * bitpos )
```

```
void packColCombCCM ( ColCombParam * cfg, float * ccm3x3 )
```

```
void packConv3x3CCM ( ConvParam * cfg, u16 * ccm3x3 )
```

```
void packConv5x5CCM ( ConvParam * cfg, u16 * ccm5x5 )
```

```
void packLumaDnsGaussLut ( YDnsParam * cfg, u8 * lut )
```

```
void sharpenSigmaToCoefficients ( float sigma, u16 * coeffs )
```

```
s32 sippFindInList ( void * pElement, void ** ppList, u32 numList )
```

```
void sippListSort ( s32 * pnList, s32 * pnSortIndices, u32 uSize, u8 descending )
```

Referenced by `sippAllocCmxLineBuffers()`, `sippAllocCmxLineBuffersOPipe()`, `sippIdentifyOPipeSchedulingEntity()`, and `sippOSEComplete()`.

```
void sippUtilComputeFp16Lut ( half(*) (half input) formula, half * outLut, u32 lutSize )

void sippUtilComputeFp16LutChannelMode ( half(*) (half input) formula, half * outLut, u32 lutSize
)

void sippUtilOrderPixels ( u32 bayerPattern, u32 inGr, u32 inR, u32 inB, u32 inGb, u32 * out )
```

Referenced by sippLoadSigma().

```
void sippUtilPrintFp16Lut ( half * fp16Lut, u32 lutSize, const char * fName )

void sippWait ( u32 numx100 )
```

Referenced by sippGenericRunIterDone(), and sippGenericRuntimeProcessIters().

7.292 sLaplacian3x3Fp16ToFp16.h File Reference

This file contains the declaration of the [Laplacian 3x3 Fp16 To Fp16](#) SIPP filter API.

```
#include <sipp.h>
```

Functions

- void [SVU_SYM\(\)](#) [svusLaplacian3x3Fp16ToFp16](#) ([SippFilter](#) *fptr)
Shave function of the [Laplacian 3x3 Fp16 To Fp16](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svusLaplacian3x3Fp16ToFp16](#))

7.292.1 Detailed Description

This file contains the declaration of the [Laplacian 3x3 Fp16 To Fp16](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see:
common/license.txt

7.293 sobel.h File Reference

This file contains the declaration of the [Sobel](#) SIPP filter API.

```
#include <sipp.h>
```

Functions

- void [SVU_SYM\(\)](#) [svuSobel](#) ([SippFilter](#) *fptr)
Shave function of the [Sobel](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuSobel](#))

7.293.1 Detailed Description

This file contains the declaration of the [Sobel](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.294 [ssd11x11.h](#) File Reference

This file contains the declaration of the [Sum of Squared Differences 11x11](#) SIPP filter API.

```
#include <sipp.h>
```

Functions

- void [SVU_SYM\(\)](#) [svuSSD11x11](#) ([SippFilter](#) *fptr)
Shave function of the [Sum of Squared Differences 11x11](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuSSD11x11](#))

7.294.1 Detailed Description

This file contains the declaration of the [Sum of Squared Differences 11x11](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.295 [ssd5x5.h](#) File Reference

This file contains the declaration of the [Sum of Squared Differences 5x5](#) SIPP filter API.

```
#include <sipp.h>
```

Functions

- void [SVU_SYM\(\)](#) [svuSSD5x5](#) ([SippFilter](#) *fptr)
Shave function of the [Sum of Squared Differences 5x5](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuSSD5x5](#))

7.295.1 Detailed Description

This file contains the declaration of the [Sum of Squared Differences 5x5](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.296 ssd7x7U8ToU32.h File Reference

This file contains the declaration of the [Sum of Squared Differences 7x7 \(U8 to U32\)](#) SIPP filter API.

```
#include <sipp.h>
```

Functions

- void [SVU_SYM\(\)](#) [svuSSD7x7U8ToU32](#) ([SippFilter](#) *fptr)
Shave function of the [Sum of Squared Differences 7x7 \(U8 to U32\)](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuSSD7x7U8ToU32](#))

7.296.1 Detailed Description

This file contains the declaration of the [Sum of Squared Differences 7x7 \(U8 to U32\)](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.297 ssdPointLine7x7U8U32.h File Reference

This file contains the declaration of the [Sum of Squared Differences 7x7](#) SIPP filter API.

```
#include <sipp.h>
```

Functions

- void [SVU_SYM\(\)](#) [svuSsdPointLine7x7U8U32](#) ([SippFilter](#) *fptr)
Shave function of the [Sum of Squared Differences 7x7](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuSsdPointLine7x7U8U32](#))

7.297.1 Detailed Description

This file contains the declaration of the [Sum of Squared Differences 7x7](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.298 startBicubic.h File Reference

```
#include <sipp.h>
#include <sippBicubic.h>
```

Data Structures

- struct [StartBicubicParam](#)

Functions

- void [SVU_SYM\(\)](#) [svuStartBicubic](#) ([SippFilter](#) *fptr)
- [SHAVE_SYM_EXPORT](#) ([svuStartBicubic](#))

7.298.1 Function Documentation

[SHAVE_SYM_EXPORT](#) ([svuStartBicubic](#))

void [SVU_SYM\(\)](#) [svuStartBicubic](#) ([SippFilter](#) * fptr)

7.299 statsAwbSatPixels.h File Reference

```
#include <sipp.h>
```

Data Structures

- struct [StatsAwbSatPixelsParam](#)

Functions

- void [SVU_SYM\(\)](#) [svuStatsAwbSatPixels](#) ([SippFilter](#) *fptr)
- [SHAVE_SYM_EXPORT](#) ([svuStatsAwbSatPixels](#))

7.299.1 Function Documentation

[SHAVE_SYM_EXPORT](#) ([svuStatsAwbSatPixels](#))

void [SVU_SYM\(\)](#) [svuStatsAwbSatPixels](#) ([SippFilter](#) * fptr)

7.300 statsAwbSatPixelsU32.h File Reference

```
#include <sipp.h>
```

Data Structures

- struct `StatsAwbSatPixelsParamU32`

Functions

- void `SVU_SYM()` `svuStatsAwbSatPixelsU32` (`SippFilter *fptr`)
- `SHAVE_SYM_EXPORT` (`svuStatsAwbSatPixelsU32`)

7.300.1 Function Documentation

`SHAVE_SYM_EXPORT` (`svuStatsAwbSatPixelsU32`)

void `SVU_SYM()` `svuStatsAwbSatPixelsU32` (`SippFilter * fptr`)

7.301 subpixelFilter.h File Reference

This file contains the declaration of the subpixel SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct `SubpixelFilterParam`
Parameter structure of the [Threshold](#) filter.

Typedefs

- typedef struct `SubpixelFilterParam` `SubpixelFilterParam`
Parameter structure of the [Threshold](#) filter.

Functions

- void `SVU_SYM()` `svuSubpixelFilter` (`SippFilter *fptr`)

7.301.1 Detailed Description

This file contains the declaration of the subpixel SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.301.2 Typedef Documentation

typedef struct **SubpixelFilterParam** **SubpixelFilterParam**

Parameter structure of the **Threshold** filter.

7.301.3 Function Documentation

void **SVU_SYM()** svuSubpixelFilter (**SippFilter** * fptr)

7.302 svuAbsdiff.c File Reference

```
#include <stdlib.h>
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/absdiff/absdiff.h>
```

Functions

- void **AbsoluteDiff** (**UInt8** **in1, **UInt8** **in2, **UInt8** **out, **UInt32** width)
- void **svuAbsdiff** (**SippFilter** *fptr)

*Shave function of the **Absolute difference** filter.*

7.302.1 Function Documentation

void **AbsoluteDiff** (**UInt8** ** in1, **UInt8** ** in2, **UInt8** ** out, **UInt32** width)

AbsoluteDiff kernel computes the absolute difference of two images given as parameters(used to estimate motion)

Parameters

in	<i>in1</i>	- array of pointers to input lines of the first image
in	<i>in2</i>	- array of pointers to input lines of the second image
out	<i>out</i>	- array of pointers to output line
in	<i>width</i>	- width of the input lines

Referenced by svuAbsdiff().

7.303 svuAccumulateSquare.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/accumulateSquare/accumulateSquare.h>
```

Functions

- void [AccumulateSquare](#) ([UInt8](#) **srcAddr, [UInt8](#) **maskAddr, float **destAddr, [UInt32](#) width, [UInt32](#) height)
- void [svuAccumulateSquare](#) ([SippFilter](#) *fptr)
Shave function of the [Accumulate Square](#) filter.

7.303.1 Function Documentation

void AccumulateSquare ([UInt8](#) ** srcAddr, [UInt8](#) ** maskAddr, float ** destAddr, [UInt32](#) width, [UInt32](#) height)

Adds the square of the source image to the accumulator.

Parameters

<i>srcAddr</i>	The input image, 1- or 3-channel, 8-bit or 32-bit floating point
<i>destAddr</i>	The accumulator image with the same number of channels as input image, 32-bit or 64-bit floating-point
<i>maskAddr</i>	Optional operation mask
<i>width</i>	Width of input image
<i>height</i>	Number of lines of input images (defaulted to one line)

Referenced by [svuAccumulateSquare](#)().

7.304 svuAccumulateWeighted.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/accumulateWeighted/accumulateWeighted.h>
```

Functions

- void [AccumulateWeighted](#) ([UInt8](#) **srcAddr, [UInt8](#) **maskAddr, float **destAddr, [UInt32](#) width, float alpha)
- void [svuAccumulateWeighted](#) ([SippFilter](#) *fptr)
Shave function of the [Accumulate Weighted](#) filter.

7.304.1 Function Documentation

void AccumulateWeighted ([UInt8](#) ** srcAddr, [UInt8](#) ** maskAddr, float ** destAddr, [UInt32](#) width, float alpha)

AccumulateWeighted kernel - The function calculates the weighted sum of the input image (srcAddr) and the accumulator (destAddr) so that accumulator becomes a running average of frame sequence

Parameters

in	<i>srcAddr</i>	- array of pointers to input lines
in	<i>maskAddr</i>	- array of pointers to input lines of mask
out	<i>destAddr</i>	- array of pointers for output lines
in	<i>width</i>	- width of input line
in	<i>alpha</i>	- Weight of the input image must be a fp32 between 0 and 1

Referenced by `svuAccumulateWeighted()`.

7.305 `svuArithmeticAdd.c` File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/arithmeticAdd/arithmeticAdd.h>
```

Functions

- void `arithmeticAddImplementation` (`UInt8 **src1`, `UInt8 **src2`, `UInt8 **dst`, `UInt32 width`)
- void `svuArithmeticAdd` (`SippFilter *fptr`)

Shave function of the [Arithmetic addition](#) filter.

7.305.1 Function Documentation

`void arithmeticAddImplementation (UInt8 ** src1, UInt8 ** src2, UInt8 ** dst, UInt32 width)`

ArithmeticAdd for two arrays

Parameters

in	<i>src1Addr</i>	- array of pointers to input lines from the first image
in	<i>src2Addr</i>	- array of pointers to input lines from the second image
out	<i>destAddr</i>	- array of pointers for output line
in	<i>width</i>	- width of input line

Referenced by `svuArithmeticAdd()`.

7.306 `svuArithmeticAddmask.c` File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/arithmeticAddmask/arithmeticAddmask.h>
```

Functions

- void `arithmeticAddmaskImplementation` (`UInt8 **src1`, `UInt8 **src2`, `UInt8 **dst`, `UInt8 **mask`, `UInt32 width`)

- void [svuArithmeticAddmask](#) ([SippFilter](#) *fptr)
Shave function of the [Arithmetic addition with mask](#) filter.

7.306.1 Function Documentation

void [arithmeticAddmaskImplementation](#) ([UInt8](#) ** src1, [UInt8](#) ** src2, [UInt8](#) ** dst, [UInt8](#) ** mask, [UInt32](#) width)

ArithmeticAdd with mask for two arrays

Parameters

in	<i>src1Addr</i>	- array of pointers to input lines from the first image
in	<i>src2Addr</i>	- array of pointers to input lines from the second image
out	<i>destAddr</i>	- array of pointers for output line
in	<i>width</i>	- width of input line

Referenced by [svuArithmeticAddmask\(\)](#).

7.307 [svuArithmeticSub.c](#) File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/arithmeticAdd/arithmeticAdd.h>
```

Functions

- void [arithmeticSubImplementation](#) ([UInt8](#) **src1, [UInt8](#) **src2, [UInt8](#) **dst, [UInt32](#) width)
- void [svuArithmeticSub](#) ([SippFilter](#) *fptr)
Shave function of the [Arithmetic subtraction](#) filter.

7.307.1 Function Documentation

void [arithmeticSubImplementation](#) ([UInt8](#) ** src1, [UInt8](#) ** src2, [UInt8](#) ** dst, [UInt32](#) width)

ArithmeticSub for two arrays

Parameters

in	<i>src1Addr</i>	- array of pointers to input lines from the first image
in	<i>src2Addr</i>	- array of pointers to input lines from the second image
out	<i>destAddr</i>	- array of pointers for output line
in	<i>width</i>	- width of input line

Referenced by [svuArithmeticSub\(\)](#).

7.308 [svuArithmeticSubFp16ToFp16.c](#) File Reference

```
#include <sipp.h>
```

```
#include <sippShaveMacros.h>
#include <filters/arithmeticSubFp16ToFp16/arithmeticSubFp16ToFp16.h>
```

Functions

- void [arithmeticSubFp16ToFp16](#) (half **src1Addr, half **src2Addr, half **destAddr, [UInt32](#) width)
- void [svuArithmeticSubFp16ToFp16](#) ([SippFilter](#) *fptr)
Shave function of the [Arithmetic subtraction fp16](#) filter.

7.308.1 Function Documentation

[void arithmeticSubFp16ToFp16](#) (half ** src1Addr, half ** src2Addr, half ** destAddr, [UInt32](#) width)

ArithmeticSub for two fp16 arrays

Parameters

in	<i>src1Addr</i>	- array of pointers to input lines from the first image
in	<i>src2Addr</i>	- array of pointers to input lines from the second image
out	<i>destAddr</i>	- array of pointers for output line
in	<i>width</i>	- width of input line

Referenced by [svuArithmeticSubFp16ToFp16\(\)](#).

7.309 svuArithmeticSubmask.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/arithmeticSubmask/arithmeticSubmask.h>
```

Functions

- void [arithmeticSubmaskImplementation](#) ([UInt8](#) **src1, [UInt8](#) **src2, [UInt8](#) **dst, [UInt8](#) **mask, [UInt32](#) width)
- void [svuArithmeticSubmask](#) ([SippFilter](#) *fptr)
Shave function of the [Arithmetic subtraction with mask](#) filter.

7.309.1 Function Documentation

[void arithmeticSubmaskImplementation](#) ([UInt8](#) ** src1, [UInt8](#) ** src2, [UInt8](#) ** dst, [UInt8](#) ** mask, [UInt32](#) width)

ArithmeticSub with mask for two arrays

Parameters

in	<i>src1Addr</i>	- array of pointers to input lines from the first image
in	<i>src2Addr</i>	- array of pointers to input lines from the second image
out	<i>destAddr</i>	- array of pointers for output line
in	<i>width</i>	- width of input line

Referenced by `svuArithmeticSubmask()`.

7.310 `svuAvg.c` File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
```

Functions

- void `svuAvg` (`SippFilter` *fptr)
Shave function of the [Average](#) filter.

7.311 `svuBilateral5x5.c` File Reference

```
#include <math.h>
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/bilateral5x5/bilateral5x5.h>
```

Functions

- `UInt16 bilateralVal` (`UInt16` *pDepthData, int depthStride, `UInt16` sigma)
- void `mvcbBilateral5x5` (`UInt16` *input, `UInt16` *sigma, `UInt32` width, `UInt16` *output)
- `UInt16 max` (`UInt16` a, `UInt16` b)
- void `svuBilateral5x5` (`SippFilter` *fptr)
Shave function of the [bilateral5x5](#) filter.

7.311.1 Function Documentation

`UInt16 bilateralVal` (`UInt16` * pDepthData, int depthStride, `UInt16` sigma)

This kernels performs a bilateral filter on the input image of 5x5 dimensions

Parameters

in	<i>input</i>	- Input lines, 16-bits unsigned char
in	<i>sigma</i>	- Sigma values, 16-bits unsigned char
in	<i>width</i>	- Width of input line
out	<i>output</i>	- Output lines, 16-bits unsigned char

Referenced by `mvcbBilateral5x5()`.

`UInt16 max (UInt16 a, UInt16 b)`

Referenced by `bilateralVal()`, `Dilate3x3()`, `Dilate5x5()`, `Dilate7x7()`, and `DilateGeneric()`.

`void mvcbBilateral5x5 (UInt16 * input, UInt16 * sigma, UInt32 width, UInt16 * output)`

Referenced by `svuBilateral5x5()`.

7.312 svuBitwiseAnd.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/bitwiseAnd/bitwiseAnd.h>
```

Functions

- void `bitwiseAnd` (`UInt8 **src1Addr`, `UInt8 **src2Addr`, `UInt8 **destAddr`, `UInt32 width`)
- void `svuBitwiseAnd` (`SippFilter *fptr`)
Shave function of the `Bitwise And` filter.

7.312.1 Function Documentation

`void bitwiseAnd (UInt8 ** src1Addr, UInt8 ** src2Addr, UInt8 ** destAddr, UInt32 width)`

per-element bit-wise logical conjunction(AND) for two arrays

Parameters

in	<i>src1Addr</i>	- array of pointers to input lines from the first image
in	<i>src2Addr</i>	- array of pointers to input lines from the second image
out	<i>destAddr</i>	- array of pointers for output line
in	<i>width</i>	- width of input line

Referenced by `svuBitwiseAnd()`.

7.313 svuBitwiseAndMask.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/bitwiseAndMask/bitwiseAndMask.h>
```

Functions

- void **bitwiseAndMask** (**UInt8** **src1Addr, **UInt8** **src2Addr, **UInt8** **destAddr, **UInt8** **maskAddr, **UInt32** width)
- void **svubitwiseAndMask** (**SippFilter** *fptr)

*Shave function of the **Bitwise And with mask** filter.*

7.313.1 Function Documentation

void bitwiseAndMask (**UInt8** ** src1Addr, **UInt8** ** src2Addr, **UInt8** ** destAddr, **UInt8** ** maskAddr, **UInt32** width)

per-element bit-wise logical conjunction(AND) for two arrays if mask == 1

Parameters

in	<i>src1Addr</i>	- array of pointers to input lines from the first image
in	<i>src2Addr</i>	- array of pointers to input lines from the second image
in	<i>maskAddr</i>	- array of pointers to input line that contain the mask
out	<i>destAddr</i>	- array of pointers for output line
in	<i>width</i>	- width of input line

Referenced by svubitwiseAndMask().

7.314 svuBitwiseNot.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/bitwiseNot/bitwiseNot.h>
```

Functions

- void **bitwiseNot** (**UInt8** **srcAddr, **UInt8** **destAddr, **UInt32** width)
- void **svuBitwiseNot** (**SippFilter** *fptr)

*Shave function of the **Bitwise Not** filter.*

7.314.1 Function Documentation

void bitwiseNot (**UInt8** ** srcAddr, **UInt8** ** destAddr, **UInt32** width)

per-element bit-wise NOT

Parameters

in	<i>srcAddr</i>	- array of pointers to input line
out	<i>destAddr</i>	- array of pointers for output line
in	<i>width</i>	- width of input line

Referenced by `svuBitwiseNot()`.

7.315 `svuBitwiseOr.c` File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/bitwiseOr/bitwiseOr.h>
```

Functions

- void `bitwiseOr` (`UInt8 **src1Addr`, `UInt8 **src2Addr`, `UInt8 **destAddr`, `UInt32 width`)
- void `svuBitwiseOr` (`SippFilter *fptr`)

*Shave function of the **Bitwise Or** filter.*

7.315.1 Function Documentation

`void bitwiseOr (UInt8 ** src1Addr, UInt8 ** src2Addr, UInt8 ** destAddr, UInt32 width)`

per-element bit-wise logical conjunction(OR) for two arrays

Parameters

in	<i>src1Addr</i>	- array of pointers to input lines from the first image
in	<i>src2Addr</i>	- array of pointers to input lines from the second image
out	<i>destAddr</i>	- array of pointers for output line
in	<i>width</i>	- width of input line

Referenced by `svuBitwiseOr()`.

7.316 `svuBitwiseOrMask.c` File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/bitwiseOrMask/bitwiseOrMask.h>
```

Functions

- void `bitwiseOrMask` (`UInt8 **src1Addr`, `UInt8 **src2Addr`, `UInt8 **destAddr`, `UInt8 **maskAddr`, `UInt32 width`)
- void `svuBitwiseOrMask` (`SippFilter *fptr`)

*Shave function of the **Bitwise Or with mask** filter.*

7.316.1 Function Documentation

```
void bitwiseOrMask ( UInt8 ** src1Addr, UInt8 ** src2Addr, UInt8 ** destAddr, UInt8 **
maskAddr, UInt32 width )
```

per-element bit-wise logical conjunction(OR) for two arrays if mask == 1

Parameters

in	<i>src1Addr</i>	- array of pointers to input lines from the first image
in	<i>src2Addr</i>	- array of pointers to input lines from the second image
in	<i>maskAddr</i>	- array of pointers to input line that contain the mask
out	<i>destAddr</i>	- array of pointers for output line
in	<i>width</i>	- width of input line

Referenced by svuBitwiseOrMask().

7.317 svuBitwiseXor.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/bitwiseXor/bitwiseXor.h>
```

Functions

- void **bitwiseXor** (UInt8 **src1Addr, UInt8 **src2Addr, UInt8 **destAddr, UInt32 width)
- void **svuBitwiseXor** (SippFilter *fptr)
Shave function of the Bitwise Xor filter.

7.317.1 Function Documentation

```
void bitwiseXor ( UInt8 ** src1Addr, UInt8 ** src2Addr, UInt8 ** destAddr, UInt32 width )
```

per-element bit-wise logical conjunction(XOR) for two arrays

Parameters

in	<i>src1Addr</i>	- array of pointers to input lines from the first image
in	<i>src2Addr</i>	- array of pointers to input lines from the second image
out	<i>destAddr</i>	- array of pointers for output line
in	<i>width</i>	- width of input line

Referenced by svuBitwiseXor().

7.318 svuBitwiseXorMask.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/bitwiseXorMask/bitwiseXorMask.h>
```

Functions

- void **bitwiseXorMask** (UInt8 **src1Addr, UInt8 **src2Addr, UInt8 **destAddr, UInt8 **maskAddr, UInt32 width)
- void **svuBitwiseXorMask** (SippFilter *fptr)

*Shave function of the **Bitwise And with mask** filter.*

7.318.1 Function Documentation

void bitwiseXorMask (UInt8 ** src1Addr, UInt8 ** src2Addr, UInt8 ** destAddr, UInt8 ** maskAddr, UInt32 width)

per-element bit-wise logical conjunction(XOR) for two arrays if mask == 1

Parameters

in	<i>src1Addr</i>	- array of pointers to input lines from the first image
in	<i>src2Addr</i>	- array of pointers to input lines from the second image
in	<i>maskAddr</i>	- array of pointers to input line that contain the mask
out	<i>destAddr</i>	- array of pointers for output line
in	<i>width</i>	- width of input line

Referenced by svuBitwiseXorMask().

7.319 svuBoxFilter.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/boxFilter/boxFilter.h>
```

Macros

- #define **BOX_FILT**(ACC_TYPE, DATA_TYPE)

Functions

- void **boxfilter** (void *in[], void *out, UInt32 dataFmt, UInt32 kerSzH, UInt32 kerSzV, UInt32 normalize, UInt32 width)
- void **svuBoxFilter** (SippFilter *fptr)

*Shave function of the **Generic Box Filter** filter.*

7.319.1 Macro Definition Documentation

```
#define BOX_FILT( ACC_TYPE, DATA_TYPE )
```

Value:

```

for (curPix=0; curPix<(int)width; curPix++)
{
    ACC_TYPE sum = 0;
    for (y=0; y<(int)kerSzV; y++)
        for (x=-lobeH; x<= (int)lobeH; x++)
            sum += * ( (DATA_TYPE*)in[y]) + x + curPix);
    if(normalize)
        ((DATA_TYPE*)out)[curPix] = sum/norm;
    else
        ((DATA_TYPE*)out)[curPix] = sum;
}

```

Referenced by boxfilter().

7.319.2 Function Documentation

void boxfilter (void * in[], void * out, **UInt32** dataFmt, **UInt32** kerSzH, **UInt32** kerSzV, **UInt32** normalize, **UInt32** width)

Referenced by svuBoxFilter().

7.320 svuBoxFilter11x11.c File Reference

```

#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/boxFilter11x11/boxFilter11x11.h>

```

Functions

- void **boxfilter11x11** (**UInt8** **in, **UInt8** **out, **UInt32** normalize, **UInt32** width)
- void **svuBoxFilter11x11** (**SippFilter** *fptr)

*Shave function of the **Box Filter 11x11** filter.*

7.320.1 Function Documentation

void boxfilter11x11 (**UInt8** ** in, **UInt8** ** out, **UInt32** normalize, **UInt32** width)

boxfilter kernel that makes average on 11x11 kernel size

Parameters

in	<i>in</i>	- array of pointers to input lines
out	<i>out</i>	- array of pointers for output lines
in	<i>normalize</i>	- parameter to check if we want to do a normalize boxfilter or not 1 for normalized values , 0 in the other case
in	<i>width</i>	- width of input line

Referenced by svuBoxFilter11x11().

7.321 svuBoxFilter13x13.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/boxFilter13x13/boxFilter13x13.h>
```

Functions

- void **boxfilter13x13** (UInt8 **in, UInt8 **out, UInt32 normalize, UInt32 width)
- void **svuBoxFilter13x13** (SippFilter *fptr)
*Shave function of the **Box Filter 13x13** filter.*

7.321.1 Function Documentation

void **boxfilter13x13** (UInt8 ** in, UInt8 ** out, UInt32 normalize, UInt32 width)

boxfilter kernel that makes average on 13x13 kernel size

Parameters

in	<i>in</i>	- array of pointers to input lines
out	<i>out</i>	- array of pointers for output lines
in	<i>normalize</i>	- parameter to check if we want to do a normalize boxfilter or not 1 for normalized values , 0 in the other case
in	<i>width</i>	- width of input line

Referenced by svuBoxFilter13x13().

7.322 svuBoxFilter15x15.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/boxFilter15x15/boxFilter15x15.h>
```

Functions

- void **boxfilter15x15** (UInt8 **in, UInt8 **out, UInt32 normalize, UInt32 width)
- void **svuBoxFilter15x15** (SippFilter *fptr)
*Shave function of the **Box Filter 15x15** filter.*

7.322.1 Function Documentation

void **boxfilter15x15** (UInt8 ** in, UInt8 ** out, UInt32 normalize, UInt32 width)

boxfilter kernel that makes average on 15x15 kernel size

Parameters

in	<i>in</i>	- array of pointers to input lines
out	<i>out</i>	- array of pointers for output lines
in	<i>normalize</i>	- parameter to check if we want to do a normalize boxfilter or not 1 for normalized values , 0 in the other case
in	<i>width</i>	- width of input line

Referenced by svuBoxFilter15x15().

7.323 svuBoxFilter3x3.c File Reference

```
#include <sipp.h>
#include <sippInternal.h>
#include <sippShaveMacros.h>
#include <filters/boxFilter3x3/boxFilter3x3.h>
```

Functions

- void **boxfilter3x3** (UInt8 **in, UInt8 **out, UInt32 normalize, UInt32 width)
- void **svuBoxFilter3x3** (SippFilter *fptr)

*Shave function of the **Box Filter 3x3** filter.*

7.323.1 Function Documentation

```
void boxfilter3x3 ( UInt8 ** in, UInt8 ** out, UInt32 normalize, UInt32 width )
```

boxfilter kernel that makes average on 3x3 kernel size

Parameters

in	<i>in</i>	- array of pointers to input lines
out	<i>out</i>	- array of pointers for output lines
in	<i>normalize</i>	- parameter to check if we want to do a normalize boxfilter or not 1 for normalized values , 0 in the other case
in	<i>width</i>	- width of input line

Referenced by svuBoxFilter3x3().

7.324 svuBoxFilter5x5.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/boxFilter5x5/boxFilter5x5.h>
```

Functions

- void **boxfilter5x5** (UInt8 **in, UInt8 **out, UInt32 normalize, UInt32 width)

- void `svuBoxFilter5x5` (`SippFilter *fptr`)
Shave function of the [Box Filter 5x5](#) filter.

7.324.1 Function Documentation

void `boxfilter5x5` (`UInt8 ** in`, `UInt8 ** out`, `UInt32 normalize`, `UInt32 width`)

boxfilter kernel that makes average on 5x5 kernel size

Parameters

in	<i>in</i>	- array of pointers to input lines
out	<i>out</i>	- array of pointers for output lines
in	<i>normalize</i>	- parameter to check if we want to do a normalize boxfilter or not 1 for normalized values , 0 in the other case
in	<i>width</i>	- width of input line

Referenced by `svuBoxFilter5x5()`.

7.325 svuBoxFilter7x7.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/boxFilter7x7/boxFilter7x7.h>
```

Functions

- void `boxfilter7x7` (`UInt8 **in`, `UInt8 **out`, `UInt32 normalize`, `UInt32 width`)
- void `svuBoxFilter7x7` (`SippFilter *fptr`)
Shave function of the [Box Filter 7x7](#) filter.

7.325.1 Function Documentation

void `boxfilter7x7` (`UInt8 ** in`, `UInt8 ** out`, `UInt32 normalize`, `UInt32 width`)

boxfilter kernel that makes average on 7x7 kernel size

Parameters

in	<i>in</i>	- array of pointers to input lines
out	<i>out</i>	- array of pointers for output lines
in	<i>normalize</i>	- parameter to check if we want to do a normalize boxfilter or not 1 for normalized values , 0 in the other case
in	<i>width</i>	- width of input line

Referenced by `svuBoxFilter7x7()`.

7.326 svuBoxFilter9x9.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/boxFilter9x9/boxFilter9x9.h>
```

Functions

- void **boxfilter9x9** (**UInt8** **in, **UInt8** **out, **UInt32** normalize, **UInt32** width)
- void **svuBoxFilter9x9** (**SippFilter** *fptr)

*Shave function of the **Box Filter 9x9** filter.*

7.326.1 Function Documentation

```
void boxfilter9x9 ( UInt8 ** in, UInt8 ** out, UInt32 normalize, UInt32 width )
```

boxfilter kernel that makes average on 9x9 kernel size

Parameters

in	<i>in</i>	- array of pointers to input lines
out	<i>out</i>	- array of pointers for output lines
in	<i>normalize</i>	- parameter to check if we want to do a normalize boxfilter or not 1 for normalized values , 0 in the other case
in	<i>width</i>	- width of input line

Referenced by **svuBoxFilter9x9()**.

7.327 svuCannyEdgeDetection.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <math.h>
#include <filters/cannyEdgeDetection/cannyEdgeDetection.h>
```

Macros

- #define **MAX_WIDTH** 1050
- #define **PADDING** 16
- #define **INTERMEDIARY_BUFFER_LINE_NUMBER** 13

Functions

- void **canny** (**UInt8** **srcAddr, **UInt8** **dstAddr, **UInt8** *buffer, **UInt32** threshold1, **UInt32** threshold2, **UInt32** width)
- void **svuCannyEdgeDetection** (**SippFilter** *fptr)

*Shave function of the **Canny Edge Detection** filter.*

Variables

- `UInt8 buffer [(MAX_WIDTH+PADDING)*INTERMEDIARY_BUFFER_LINE_NUMBER]`

7.327.1 Macro Definition Documentation

```
#define INTERMEDIARY_BUFFER_LINE_NUMBER 13
```

```
#define MAX_WIDTH 1050
```

```
#define PADDING 16
```

7.327.2 Function Documentation

```
void canny ( UInt8 ** srcAddr, UInt8 ** dstAddr, UInt8 * buffer, UInt32 threshold1, UInt32 threshold2, UInt32 width )
```

cannyEdge filter - The function finds edges in the input image image and marks them in the output map edges using the Canny algorithm(9x9 kernel size). The smallest value between threshold1 and threshold2 is used for edge linking. The largest value is used to find initial segments of strong edges.

Parameters

in	<i>srcAddr</i>	- array of pointers to input lines
out	<i>dstAddr</i>	- pointers for output line
in	<i>threshold1</i>	- lower threshold - value between 0 -255
in	<i>threshold2</i>	- upper threshold - value between 0 -255
in	<i>width</i>	- width of input line

Referenced by `svuCannyEdgeDetection()`.

7.327.3 Variable Documentation

`UInt8 buffer[(MAX_WIDTH+PADDING)*INTERMEDIARY_BUFFER_LINE_NUMBER]`

Referenced by `canny()`, `svuCannyEdgeDetection()`, and `svuCornerMinEigenVal()`.

7.328 svuCensusMatching16.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/censusMatching16/censusMatching16.h>
```

Functions

- void `mvCvCensusMatching16` (UInt32 *in1, UInt32 *in2, UInt8 *out, UInt32 width)
- void `svuCensusMatching16` (SippFilter *fptr)

Parameter structure of the *censusMatching16* filter.

7.328.1 Function Documentation

`void mvCvCensusMatching16 (UInt32 * in1, UInt32 * in2, UInt8 * out, UInt32 width)`

`mvCvCensusMatching` - performs an XOR operation between pixel one pixel in `*in1` and 16 pixels from `*in2` and counts up how many values of 1 are in the result

Parameters

<code>in</code>	<code>in1</code>	- Input lines of the left image
<code>in</code>	<code>in2</code>	- Input lines of the right image
<code>out</code>	<code>out</code>	- array of disparity cost
<code>in</code>	<code>width</code>	- Width of the input lines

Returns

Nothing

Referenced by `svuCensusMatching16()`.

7.329 svuCensusMatching32.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/censusMatching32/censusMatching32.h>
```

Functions

- void `mvCvCensusMatching32` (UInt32 *in1, UInt32 *in2, UInt8 *out, UInt32 width, UInt32 flag)
- void `svuCensusMatching32` (SippFilter *fptr)
Shave function of the `censusMatching32` filter.

7.329.1 Function Documentation

`void mvCvCensusMatching32 (UInt32 * in1, UInt32 * in2, UInt8 * out, UInt32 width, UInt32 flag)`

`mvCvCensusMatching` - performs an XOR operation between pixel one pixel in `*in1` and 16 pixels from `*in2` and counts up how many values of 1 are in the result

Parameters

<code>in</code>	<code>in1</code>	- Input lines of the left image
<code>in</code>	<code>in2</code>	- Input lines of the right image
<code>in</code>	<code>flag</code>	- enable right crossing (by default left crossing)

out	<i>out</i>	- array of disparity cost
in	<i>width</i>	- Width of the input lines

Referenced by `svuCensusMatching32()`.

7.330 `svuCensusMatching64.c` File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/censusMatching64/censusMatching64.h>
```

Functions

- void `mvcvCensusMatching64` (`UInt32 *in1`, `UInt32 *in2`, `UInt8 *out`, `UInt32 width`, `UInt32 flag`)
- void `svuCensusMatching64` (`SippFilter *fptr`)

Shave function of the `censusMatching64` filter.

7.330.1 Function Documentation

`void mvcvCensusMatching64 (UInt32 * in1, UInt32 * in2, UInt8 * out, UInt32 width, UInt32 flag)`

`mvcvCensusMatching` - performs an XOR operation between pixel one pixel in `*in1` and 16 pixels from `*in2` and counts up how many values of 1 are in the result

Parameters

in	<i>in1</i>	- Input lines of the left image
in	<i>in2</i>	- Input lines of the right image
in	<i>flag</i>	- enable right crossing (by default left crossing)
out	<i>out</i>	- array of disparity cost
in	<i>width</i>	- Width of the input lines

Referenced by `svuCensusMatching64()`.

7.331 `svuCensusMatching65.c` File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/censusMatching65/censusMatching65.h>
```

Functions

- void `mvcvCensusMatching65` (`UInt32 *in1`, `UInt32 *in2`, `UInt8 *out`, `UInt32 width`)
- void `svuCensusMatching65` (`SippFilter *fptr`)

Parameter structure of the `censusMatching65` filter.

7.331.1 Function Documentation

`void mvCvCensusMatching65 (UInt32 * in1, UInt32 * in2, UInt8 * out, UInt32 width)`

`mvCvCensusMatching` - performs an XOR operation between pixel one pixel in `*in1` and 16 pixels from `*in2` and counts up how many values of 1 are in the result

Parameters

<code>in</code>	<code>in1</code>	- Input lines of the left image
<code>in</code>	<code>in2</code>	- Input lines of the right image
<code>out</code>	<code>out</code>	- array of disparity cost
<code>in</code>	<code>width</code>	- Width of the input lines

Referenced by `svuCensusMatching65()`.

7.332 svuCensusMatchingPyr.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/censusMatchingPyr/censusMatchingPyr.h>
```

Functions

- `void mvCvCensusMatchingPyr (UInt32 *in1, UInt32 *in2, UInt8 *predicted, UInt8 *out, UInt32 width)`
- `void svuCensusMatchingPyr (SippFilter *fptr)`
Shave function of the `censusMatchingPyr` filter.

7.332.1 Function Documentation

`void mvCvCensusMatchingPyr (UInt32 * in1, UInt32 * in2, UInt8 * predicted, UInt8 * out, UInt32 width)`

`mvCvCensusMatching` - performs an XOR operation between pixel one pixel in `*in1` and 7 pixels from `*in2`, based on predicted disparities, and counts up how many values of 1 are in the result

Parameters

<code>in</code>	<code>in1</code>	- pointer to input lines of the left image
<code>in</code>	<code>in2</code>	- pointer to input lines of the right image
<code>in</code>	<code>predicted</code>	- pointer to predicted disparities
<code>out</code>	<code>out</code>	- array of disparity cost
<code>in</code>	<code>width</code>	- width of the input lines

Returns

Nothing

Referenced by `svuCensusMatchingPyr()`.

7.333 svuCensusMin16.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/censusMin16/censusMin16.h>
```

Functions

- void **mvcvCensusMin16** (UInt8 *in, UInt8 *out, UInt32 width)
- void **svuCensusMin16** (SippFilter *fptr)

*Parameter structure of the **censusMin16** filter.*

7.333.1 Function Documentation

```
void mvcvCensusMin16 ( UInt8 * in, UInt8 * out, UInt32 width )
```

mvcvCensusMin - computes minimum of 16 disparity costs values

Parameters

in	<i>in</i>	- pointer to disparity costs
out	<i>out</i>	- array of disparity cost
in	<i>width</i>	- width of the input lines

Returns

Nothing

Referenced by svuCensusMin16().

7.334 svuCensusMin64.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/censusMin64/censusMin64.h>
```

Functions

- void **mvcvCensusMin64** (UInt8 *in, UInt8 *out, UInt8 *outm, UInt32 width)
- void **svuCensusMin64** (SippFilter *fptr)

*Parameter structure of the **censusMin64** filter.*

7.334.1 Function Documentation

```
void mvcvCensusMin64 ( UInt8 * in, UInt8 * out, UInt8 * outm, UInt32 width )
```

mvcvCensusMin - computes minimum of 64 disparity costs values

Parameters

in	<i>in</i>	- pointer to disparity costs
out	<i>out</i>	- array of minimum positions
out	<i>outm</i>	- array of minimum values
in	<i>width</i>	- width of the input lines

Returns

Nothing

Referenced by svuCensusMin64().

7.335 svuCensusMin65.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/censusMin65/censusMin65.h>
```

Functions

- void [mvcvCensusMin65](#) (UInt8 *in, UInt8 *out, UInt32 width)
- void [svuCensusMin65](#) (SippFilter *fptr)

Parameter structure of the [censusMin65](#) filter.

7.335.1 Function Documentation

void mvcvCensusMin65 (UInt8 * in, UInt8 * out, UInt32 width)

mvcvCensusMin - computes minimum of 65 disparity costs values

Parameters

in	<i>in</i>	- pointer to disparity costs
out	<i>out</i>	- array of disparity cost
in	<i>width</i>	- width of the input lines

Returns

Nothing

Referenced by svuCensusMin65().

7.336 svuCensusMin7.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/censusMin7/censusMin7.h>
```


Functions

- void [mvCvCensusMin7](#) (UInt8 *in, UInt8 *out, UInt32 width)
- void [svuCensusMin7](#) (SippFilter *fptr)

Parameter structure of the [censusMin7](#) filter.

7.336.1 Function Documentation

void mvCvCensusMin7 (UInt8 * in, UInt8 * out, UInt32 width)

mvCvCensusMin - computes minimum of 7 disparity costs values

Parameters

in	<i>in</i>	- pointer to disparity costs
out	<i>out</i>	- array of disparity cost
in	<i>width</i>	- width of the input lines

Returns

Nothing

Referenced by svuCensusMin7().

7.337 svuCensusTransform5x5.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/censusTransform5x5/censusTransform5x5.h>
```

Functions

- void [mvCvCensusTransform5x5](#) (UInt8 **in, UInt32 *out, UInt32 width)
- void [svuCensusTransform5x5](#) (SippFilter *fptr)

Parameter structure of the [CensusTransform5x5](#) filter.

7.337.1 Function Documentation

void mvCvCensusTransform5x5 (UInt8 ** in, UInt32 * out, UInt32 width)

censusTransform5x5 kernel

Parameters

in	<i>in</i>	- array of pointers to input lines
----	-----------	------------------------------------

in	out	- pointer to output line
in	inWidth	- width of input line

Referenced by svuCensusTransform5x5().

7.338 svuChannelExtract.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/channelExtract/channelExtract.h>
```

Functions

- void [channelExtract](#) ([UInt8](#) **in, [UInt8](#) **out, [UInt32](#) width, [UInt32](#) plane)
- void [svuChannelExtract](#) ([SippFilter](#) *fptr)

Shave function of the [channelExtract](#) filter.

7.338.1 Function Documentation

void [channelExtract](#) ([UInt8](#) ** in, [UInt8](#) ** out, [UInt32](#) width, [UInt32](#) plane)

[channelExtract](#) kernel - This kernel extracts one of the R, G, B, plane from an interleaved RGB line

Parameters

in	in	- array of pointers to input lines
out		- array of pointers for output lines
in	width	- width of input line
in	plane	- number 0 to extract plane R, 1 for extracting G, 2 for extracting B

Referenced by svuChannelExtract().

7.339 svuChromaBlock.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/chromaBlock/chromaBlock.h>
```

Functions

- void [svuChromaBlock](#) ([SippFilter](#) *fptr)

Shave function of the [Chroma Block](#) filter.

7.340 svuCombDecimDemosaicAwbGains.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/combDecimDemosaicAwbGains/combDecimDemosaicAwb-
Gains.h>
```

Macros

- #define [CLAMPZ255](#)(X) ((X)>255 ? 255 : ((X)<0 ? 0 : (X)))

Functions

- void [combDecimDemosaicAwbGainsGR](#) (unsigned char **output, unsigned short **iline, unsigned short gains[3], unsigned int width)
- void [combDecimDemosaicAwbGainsRG](#) (unsigned char **output, unsigned short **iline, unsigned short gains[3], unsigned int width)
- void [combDecimDemosaicAwbGainsGB](#) (unsigned char **output, unsigned short **iline, unsigned short gains[3], unsigned int width)
- void [combDecimDemosaicAwbGainsBG](#) (unsigned char **output, unsigned short **iline, unsigned short gains[3], unsigned int width)
- void [svuCombDecimDemosaicAwbGains](#) ([SippFilter](#) *fptr)

7.340.1 Macro Definition Documentation

```
#define CLAMPZ255( X ) ((X)>255 ? 255 : ((X)<0 ? 0 : (X)))
```

Referenced by [combDecimDemosaicAwbGainsBG\(\)](#), [combDecimDemosaicAwbGainsGB\(\)](#), [combDecimDemosaicAwbGainsGR\(\)](#), and [combDecimDemosaicAwbGainsRG\(\)](#).

7.340.2 Function Documentation

```
void combDecimDemosaicAwbGainsBG ( unsigned char ** output, unsigned short ** iline, unsigned
short gains[3], unsigned int width )
```

Referenced by [svuCombDecimDemosaicAwbGains\(\)](#).

```
void combDecimDemosaicAwbGainsGB ( unsigned char ** output, unsigned short ** iline, unsigned
short gains[3], unsigned int width )
```

Referenced by [svuCombDecimDemosaicAwbGains\(\)](#).

```
void combDecimDemosaicAwbGainsGR ( unsigned char ** output, unsigned short ** iline, unsigned
short gains[3], unsigned int width )
```

Referenced by [svuCombDecimDemosaicAwbGains\(\)](#).

```
void combDecimDemosaicAwbGainsRG ( unsigned char ** output, unsigned short ** iline, unsigned short gains[3], unsigned int width )
```

Referenced by svuCombDecimDemosaicAwbGains().

```
void svuCombDecimDemosaicAwbGains ( SippFilter * fptr )
```

7.341 svuContrast.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/contrast/contrast.h>
```

Functions

- void **svuContrast** (**SippFilter** *fptr)
Shave function of the Contrast filter.

7.342 svuConv11x11.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/conv11x11/conv11x11.h>
```

Functions

- void **Convolution11x11** (**UInt8** **in, **UInt8** **out, half *conv, **UInt32** inWidth)
- void **svuConv11x11** (**SippFilter** *fptr)
Shave function of the Convolution 11x11 filter.

7.342.1 Function Documentation

```
void Convolution11x11 ( UInt8 ** in, UInt8 ** out, half * conv, UInt32 inWidth )
```

Convolution 11x11 kernel

Parameters

in	in	- array of pointers to input lines
in	out	- array of pointers to output lines
in	conv	- array of values from convolution
in	inWidth	- width of input line

Referenced by svuConv11x11().

7.343 svuConv15x1.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/conv15x1/conv15x1.h>
```

Functions

- void [Convolution15x1](#) (UInt8 **in, UInt8 **out, half *conv, UInt32 inWidth)
- void [svuConv15x1](#) (SippFilter *fptr)

Shave function of the [Convolution 15x1](#) filter.

7.343.1 Function Documentation

```
void Convolution15x1 ( UInt8 ** in, UInt8 ** out, half * conv, UInt32 inWidth )
```

Convolution 15x1 kernel

Parameters

in	<i>in</i>	- array of pointers to input lines
in	<i>out</i>	- array of pointers to output lines
in	<i>conv</i>	- array of values from convolution
in	<i>inWidth</i>	- width of input line

Referenced by [svuConv15x1\(\)](#).

7.344 svuConv1x15.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/conv1x15/conv1x15.h>
```

Functions

- void [Convolution1x15](#) (UInt8 **in, UInt8 **out, half *conv, UInt32 inWidth)
- void [svuConv1x15](#) (SippFilter *fptr)

Shave function of the [Convolution 1x15](#) filter.

7.344.1 Function Documentation

```
void Convolution1x15 ( UInt8 ** in, UInt8 ** out, half * conv, UInt32 inWidth )
```

Convolution 1x15 kernel

Parameters

in	<i>in</i>	- array of pointers to input lines
in	<i>out</i>	- array of pointers to output lines
in	<i>conv</i>	- array of values from convolution
in	<i>inWidth</i>	- width of input line

Referenced by `svuConv1x15()`.

7.345 `svuConv1x5.c` File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/conv1x5/conv1x5.h>
```

Functions

- void `Convolution1x5` (`UInt8 **in`, `UInt8 **out`, `half *conv`, `UInt32 inWidth`)
- void `svuConv1x5` (`SippFilter *fptr`)

Shave function of the `Convolution 1x5` filter.

7.345.1 Function Documentation

void `Convolution1x5` (`UInt8 ** in`, `UInt8 ** out`, `half * conv`, `UInt32 inWidth`)

Convolution 1x5 kernel

Parameters

in	<i>in</i>	- array of pointers to input lines
in	<i>out</i>	- array of pointers to output lines
in	<i>conv</i>	- array of values from convolution
in	<i>inWidth</i>	- width of input line

Referenced by `svuConv1x5()`.

7.346 `svuConv1x5Fp16ToFp16.c` File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/conv1x5Fp16ToFp16/conv1x5Fp16ToFp16.h>
```

Functions

- void `Convolution1x5Fp16ToFp16` (`half **in`, `half **out`, `half *conv`, `UInt32 inWidth`)
- void `svuConv1x5Fp16ToFp16` (`SippFilter *fptr`)

Shave function of the `Convolution 1x5 Fp16ToFp16` filter.

7.346.1 Function Documentation

`void Convolution1x5Fp16ToFp16 (half ** in, half ** out, half * conv, UInt32 inWidth)`

Convolution 1x5Fp16ToFp16 kernel

Parameters

<code>in</code>	<i>in</i>	- array of pointers to input lines
<code>out</code>	<i>out</i>	- array of pointers to output lines
<code>in</code>	<i>conv</i>	- array of values from convolution
<code>in</code>	<i>inWidth</i>	- width of input line

Referenced by `svuConv1x5Fp16ToFp16()`.

7.347 svuConv1x7.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/conv1x7/conv1x7.h>
```

Functions

- `void Convolution1x7 (UInt8 **in, UInt8 **out, half *conv, UInt32 inWidth)`
- `void svuConv1x7 (SippFilter *fptr)`

Shave function of the Convolution 1x7 filter.

7.347.1 Function Documentation

`void Convolution1x7 (UInt8 ** in, UInt8 ** out, half * conv, UInt32 inWidth)`

Convolution 1x7 kernel

Parameters

<code>in</code>	<i>in</i>	- array of pointers to input lines
<code>in</code>	<i>out</i>	- array of pointers to output lines
<code>in</code>	<i>conv</i>	- array of values from convolution
<code>in</code>	<i>inWidth</i>	- width of input line

Referenced by `svuConv1x7()`.

7.348 svuConv1x7Fp16ToFp16.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/conv1x7Fp16ToFp16/conv1x7Fp16ToFp16.h>
```

Functions

- void [Convolution1x7Fp16ToFp16](#) (half *in, half *out, half *conv, [UInt32](#) inWidth)
- void [svuConv1x7Fp16ToFp16](#) ([SippFilter](#) *fptr)

Shave function of the [Convolution 1x7 Fp16ToFp16](#) filter.

7.348.1 Function Documentation

[void Convolution1x7Fp16ToFp16](#) (half * in, half * out, half * conv, [UInt32](#) inWidth)

Convolution 1x7 kernel fp16

Parameters

in	<i>in</i>	- pointer to input line
out	<i>out</i>	- pointer to output line
in	<i>conv</i>	- array of values from convolution
in	<i>inWidth</i>	- width of input line

Referenced by [svuConv1x7Fp16ToFp16\(\)](#).

7.349 svuConv1x9.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/conv1x9/conv1x9.h>
```

Functions

- void [Convolution1x9](#) ([UInt8](#) **in, [UInt8](#) **out, half *conv, [UInt32](#) inWidth)
- void [svuConv1x9](#) ([SippFilter](#) *fptr)

Shave function of the [Convolution 1x9](#) filter.

7.349.1 Function Documentation

[void Convolution1x9](#) ([UInt8](#) ** in, [UInt8](#) ** out, half * conv, [UInt32](#) inWidth)

Convolution 1x9 kernel

Parameters

in	<i>in</i>	- array of pointers to input lines
in	<i>out</i>	- array of pointers to output lines
in	<i>conv</i>	- array of values from convolution
in	<i>inWidth</i>	- width of input line

Referenced by [svuConv1x9\(\)](#).

7.350 svuConv3x3.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/conv3x3/conv3x3.h>
```

Functions

- void [conv3x3FilterImplementation](#) (UInt8 *inLine[3], UInt8 *outLine, half *f, UInt32 widthLine)
- void [svuConv3x3](#) (SippFilter *fptr)

Shave function of the [Convolution 3x3](#) filter.

7.350.1 Function Documentation

```
void conv3x3FilterImplementation ( UInt8 * inLine[3], UInt8 * outLine, half * f, UInt32 widthLine
)
```

Convolution 3x3 kernel

Parameters

in	<i>in</i>	- array of pointers to input lines
in	<i>out</i>	- array of pointers to output lines
in	<i>conv</i>	- array of values from convolution
in	<i>inWidth</i>	- width of input line

Referenced by [svuConv3x3\(\)](#).

7.351 svuConv3x3Fp16ToFp16.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/conv3x3Fp16ToFp16/conv3x3Fp16ToFp16.h>
```

Functions

- void [convolution3x3Fp16ToFp16Implementation](#) (half **in, half **out, half *conv, UInt32 inWidth)
- void [svuConv3x3Fp16ToFp16](#) (SippFilter *fptr)

Shave function of the [Convolution 3x3 Fp16ToFp16](#) filter.

7.351.1 Function Documentation

```
void convolution3x3Fp16ToFp16Implementation ( half ** in, half ** out, half * conv, UInt32
inWidth )
```

Convolution 3x3Fp16ToFp16 kernel

Parameters

in	<i>in</i>	- array of pointers to input lines
in	<i>out</i>	- array of pointers to output lines
in	<i>conv</i>	- array of values from convolution
in	<i>inWidth</i>	- width of input line

Referenced by `svuConv3x3Fp16ToFp16()`.

7.352 svuConv5x1.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/conv5x1/conv5x1.h>
```

Functions

- void `Convolution5x1` (`UInt8 **in`, `UInt8 **out`, `half *conv`, `UInt32 inWidth`)
- void `svuConv5x1` (`SippFilter *fptr`)

Shave function of the `Convolution 5x1` filter.

7.352.1 Function Documentation

`void Convolution5x1 (UInt8 ** in, UInt8 ** out, half * conv, UInt32 inWidth)`

Convolution 5x1 kernel

Parameters

in	<i>in</i>	- array of pointers to input lines
in	<i>out</i>	- array of pointers to output lines
in	<i>conv</i>	- array of values from convolution
in	<i>inWidth</i>	- width of input line

Referenced by `svuConv5x1()`.

7.353 svuConv5x1Fp16ToFp16.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/conv5x1Fp16ToFp16/conv5x1Fp16ToFp16.h>
```

Functions

- void `Convolution5x1Fp16ToFp16` (`half **in`, `half **out`, `half *conv`, `UInt32 inWidth`)
- void `svuConv5x1Fp16ToFp16` (`SippFilter *fptr`)

Shave function of the `Convolution 5x1 Fp16ToFp16` filter.

7.353.1 Function Documentation

`void Convolution5x1Fp16ToFp16 (half ** in, half ** out, half * conv, UInt32 inWidth)`

Convolution 5x1Fp16ToFp16 kernel

Parameters

<code>in</code>	<i>in</i>	- array of pointers to input lines
<code>out</code>	<i>out</i>	- array of pointers to output lines
<code>in</code>	<i>conv</i>	- array of values from convolution
<code>in</code>	<i>inWidth</i>	- width of input line

Referenced by `svuConv5x1Fp16ToFp16()`.

7.354 svuConv5x5.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/conv5x5/conv5x5.h>
```

Functions

- `void Convolution5x5 (UInt8 **in, UInt8 **out, half conv[25], UInt32 inWidth)`
- `void svuConv5x5 (SippFilter *fptr)`

Shave function of the Convolution 5x5 filter.

7.354.1 Function Documentation

`void Convolution5x5 (UInt8 ** in, UInt8 ** out, half conv[25], UInt32 inWidth)`

Convolution 5x5 kernel

Parameters

<code>in</code>	<i>in</i>	- array of pointers to input lines
<code>in</code>	<i>out</i>	- array of pointers to output lines
<code>in</code>	<i>conv</i>	- array of values from convolution
<code>in</code>	<i>inWidth</i>	- width of input line

Referenced by `svuConv5x5()`.

7.355 svuConv5x5Fp16ToFp16.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/conv5x5Fp16ToFp16/conv5x5Fp16ToFp16.h>
```

Functions

- void [Convolution5x5Fp16ToFp16](#) (half **in, half **out, half conv[25], [UInt32](#) inWidth)
- void [svuConv5x5Fp16ToFp16](#) ([SippFilter](#) *fptr)

Shave function of the [Convolution 5x5 Fp16ToFp16](#) filter.

7.355.1 Function Documentation

[void Convolution5x5Fp16ToFp16](#) (half ** in, half ** out, half conv[25], [UInt32](#) inWidth)

Convolution 5x5Fp16ToFp16 kernel

Parameters

in	<i>in</i>	- array of pointers to input lines
in	<i>out</i>	- array of pointers to output lines
in	<i>conv</i>	- array of values from convolution
in	<i>inWidth</i>	- width of input line

Referenced by [svuConv5x5Fp16ToFp16\(\)](#).

7.356 svuConv7x1.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/conv7x1/conv7x1.h>
```

Functions

- void [Convolution7x1](#) ([UInt8](#) **in, [UInt8](#) **out, half *conv, [UInt32](#) inWidth)
- void [svuConv7x1](#) ([SippFilter](#) *fptr)

Shave function of the [Convolution 7x1](#) filter.

7.356.1 Function Documentation

[void Convolution7x1](#) ([UInt8](#) ** in, [UInt8](#) ** out, half * conv, [UInt32](#) inWidth)

Convolution 7x1 kernel

Parameters

in	<i>in</i>	- array of pointers to input lines
in	<i>out</i>	- array of pointers to output lines
in	<i>conv</i>	- array of values from convolution
in	<i>inWidth</i>	- width of input line

Referenced by [svuConv7x1\(\)](#).

7.357 svuConv7x1Fp16ToFp16.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/conv7x1Fp16ToFp16/conv7x1Fp16ToFp16.h>
```

Functions

- void [convolution7x1Fp16ToFp16](#) (half **in, half *out, half *conv, [UInt32](#) inWidth)
- void [svuConv7x1Fp16ToFp16](#) ([SippFilter](#) *fptr)

Shave function of the [Convolution 7x1 Fp16ToFp16](#) filter.

7.357.1 Function Documentation

```
void convolution7x1Fp16ToFp16 ( half ** in, half * out, half * conv, UInt32 inWidth )
```

Convolution 7x1Fp16ToFp16 kernel

Parameters

in	<i>in</i>	- array of pointers to input lines
out	<i>out</i>	- pointer to output line
in	<i>conv</i>	- array of values from convolution
in	<i>inWidth</i>	- width of input line

Referenced by [svuConv7x1Fp16ToFp16\(\)](#).

7.358 svuConv7x7.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/conv7x7/conv7x7.h>
```

Functions

- void [convolution7x7](#) ([UInt8](#) **in, [UInt8](#) **out, half *conv, [UInt32](#) inWidth)
- void [svuConv7x7](#) ([SippFilter](#) *fptr)

Shave function of the [Convolution 7x7](#) filter.

7.358.1 Function Documentation

```
void convolution7x7 ( UInt8 ** in, UInt8 ** out, half * conv, UInt32 inWidth )
```

Convolution 7x7 kernel

Parameters

in	<i>in</i>	- array of pointers to input lines
in	<i>out</i>	- array of pointers to output lines
in	<i>conv</i>	- array of values from convolution
in	<i>inWidth</i>	- width of input line

Referenced by svuConv7x7().

7.359 svuConv7x7Fp16ToFp16.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/conv7x7Fp16ToFp16/conv7x7Fp16ToFp16.h>
```

Functions

- void [convolution7x7Fp16ToFp16](#) (half **in, half **out, half *conv, [UInt32](#) inWidth)
- void [svuConv7x7Fp16ToFp16](#) ([SippFilter](#) *fptr)
Shave function of the [Convolution 7x7 Fp16ToFp16](#) filter.

7.359.1 Function Documentation

void [convolution7x7Fp16ToFp16](#) (half ** in, half ** out, half * conv, [UInt32](#) inWidth)

Convolution 7x7 kernel

Parameters

in	<i>in</i>	- array of pointers to input lines
in	<i>out</i>	- array of pointers to output lines
in	<i>conv</i>	- array of values from convolution
in	<i>inWidth</i>	- width of input line

Referenced by svuConv7x7Fp16ToFp16().

7.360 svuConv7x7Fp16ToU8.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/conv7x7Fp16ToU8/conv7x7Fp16ToU8.h>
```

Functions

- void [convolution7x7Fp16ToU8](#) (half **in, [UInt8](#) **out, half *conv, [UInt32](#) inWidth)
- void [svuConv7x7Fp16ToU8](#) ([SippFilter](#) *fptr)
Shave function of the [Convolution 7x7 Fp16ToU8](#) filter.

7.360.1 Function Documentation

`void convolution7x7Fp16ToU8 (half ** in, UInt8 ** out, half * conv, UInt32 inWidth)`

Convolution 7x7 kernel

Parameters

<code>in</code>	<i>in</i>	- array of pointers to input lines
<code>in</code>	<i>out</i>	- array of pointers to output lines
<code>in</code>	<i>conv</i>	- array of values from convolution
<code>in</code>	<i>inWidth</i>	- width of input line

Referenced by `svuConv7x7Fp16ToU8()`.

7.361 svuConv9x1.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/conv9x1/conv9x1.h>
```

Functions

- void `Convolution9x1` (UInt8 **in, UInt8 **out, half *conv, UInt32 inWidth)
- void `svuConv9x1` (SippFilter *fptr)
Shave function of the `Convolution 9x1` filter.

7.361.1 Function Documentation

`void Convolution9x1 (UInt8 ** in, UInt8 ** out, half * conv, UInt32 inWidth)`

Convolution 9x1 kernel

Parameters

<code>in</code>	<i>in</i>	- array of pointers to input lines
<code>in</code>	<i>out</i>	- array of pointers to output lines
<code>in</code>	<i>conv</i>	- array of values from convolution
<code>in</code>	<i>inWidth</i>	- width of input line

Referenced by `svuConv9x1()`.

7.362 svuConv9x9.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/conv9x9/conv9x9.h>
```

Functions

- void [Convolution9x9](#) (UInt8 **in, UInt8 **out, half *conv, UInt32 inWidth)
- void [svuConv9x9](#) (SippFilter *fptr)

Shave function of the [Convolution 9x9](#) filter.

7.362.1 Function Documentation

void [Convolution9x9](#) (UInt8 ** in, UInt8 ** out, half * conv, UInt32 inWidth)

Convolution 9x9 kernel

Parameters

in	<i>in</i>	- array of pointers to input lines
in	<i>out</i>	- array of pointers to output lines
in	<i>conv</i>	- array of values from convolution
in	<i>inWidth</i>	- width of input line

Referenced by [svuConv9x9\(\)](#).

7.363 svuConv9x9Fp16ToFp16.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/conv9x9Fp16ToFp16/conv9x9Fp16ToFp16.h>
```

Functions

- void [Convolution9x9Fp16ToFp16](#) (half **in, half **out, half *conv, UInt32 inWidth)
- void [svuConv9x9Fp16ToFp16](#) (SippFilter *fptr)

Shave function of the [Convolution 9x9 Fp16ToFp16](#) filter.

7.363.1 Function Documentation

void [Convolution9x9Fp16ToFp16](#) (half ** in, half ** out, half * conv, UInt32 inWidth)

Convolution 9x9Fp16ToFp16 kernel

Parameters

in	<i>in</i>	- array of pointers to input lines
in	<i>out</i>	- array of pointers to output lines
in	<i>conv</i>	- array of values from convolution
in	<i>inWidth</i>	- width of input line

Referenced by [svuConv9x9Fp16ToFp16\(\)](#).

7.364 svuConvert16bppTo8bpp.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/convert16bppTo8bpp/convert16bppTo8bpp.h>
```

Macros

- #define CLAMPU8(x) (x < 0 ? 0 : x > 255 ? 255 : x)

Functions

- void svuConvert16bppTo8bpp (SippFilter *fptr)
Shave function of the Convert 16bpp To 8bpp filter.

7.364.1 Macro Definition Documentation

```
#define CLAMPU8( x ) (x < 0 ? 0 : x > 255 ? 255 : x)
```

Referenced by svuConvert16bppTo8bpp().

7.365 svuConvertF16ToU8.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/convertF16ToU8/convertF16ToU8.h>
```

Functions

- void convertF16ToU8 (half *in, UInt8 *out, UInt32 width)
- void svuConvertF16ToU8 (SippFilter *fptr)
Shave function of the Convert F16 To U8 filter.

7.365.1 Function Documentation

```
void convertF16ToU8 ( half * in, UInt8 * out, UInt32 width )
```

Convert a 16-bit floating point line to 8-bit line, with saturation

Parameters

in	in	- Input line
----	----	--------------

out	<i>out</i>	- Output line
in	<i>width</i>	- Width of the input line

Returns

Nothing

Referenced by svuConvertF16ToU8().

7.366 svuConvertFrom12BppTo8Bpp.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/convertFrom12BppTo8Bpp/convertFrom12BppTo8Bpp.h>
```

Functions

- void [mvcvConvert12BppTo8Bpp](#) (UInt8 *out, UInt8 *in, u32 width)
- void [svuConvertFrom12BppTo8Bpp](#) (SippFilter *fptr)
Shave function of the [12Bpp to 8Bpp conversion filter](#).

7.366.1 Function Documentation

void [mvcvConvert12BppTo8Bpp](#) (UInt8 * out, UInt8 * in, u32 width)

ConvertFrom12BppTo8Bpp kernel This kernel converts from 12 bpp to 8 bpp

Parameters

in	<i>in</i>	- Input line
out	<i>out</i>	- Output line
in	<i>width</i>	- Width of the input line

Referenced by svuConvertFrom12BppTo8Bpp().

7.367 svuConvertPFp16U16.c File Reference

```
#include "sipp.h"
#include "sippMacros.h"
#include "filters/convertPFp16U16/convertPFp16U16.h"
```

Macros

- #define [MAX_U16_VAL](#) ((1<<10))

Functions

- void [svuConvertPFp16U16](#) (SippFilter *fptr)
Shave function of the [Convert Fp16 to U16](#) filter.

7.367.1 Macro Definition Documentation

```
#define MAX_U16_VAL ((1<<10))
```

Referenced by [svuConvertPFp16U16\(\)](#).

7.368 svuConvertPU16Fp16.c File Reference

```
#include "sipp.h"
#include "sippMacros.h"
#include "filters/convertPU16Fp16/convertPU16Fp16.h"
```

Macros

- #define [MAX_U16_VAL](#) ((1<<10))

Functions

- void [svuConvertPU16Fp16](#) (SippFilter *fptr)
Shave function of the [Convert U16 to Fp16](#) filter.

7.368.1 Macro Definition Documentation

```
#define MAX_U16_VAL ((1<<10))
```

Referenced by [svuConvertPU16Fp16\(\)](#).

7.369 svuConvertU8ToF16.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/convertU8ToF16/convertU8ToF16.h>
```

Functions

- void [convertU8ToF16](#) (UInt8 *in, half *out, UInt32 width)
- void [svuConvertU8ToF16](#) (SippFilter *fptr)
Shave function of the [Convert U8 To F16](#) filter.

7.369.1 Function Documentation

`void convertU8ToF16 (UInt8 * in, half * out, UInt32 width)`

Convert a 8-bit line to 16-bit floating point line

Parameters

<code>in</code>	<i>in</i>	- Input line
<code>out</code>	<i>out</i>	- Output line
<code>in</code>	<i>width</i>	- Width of the input line

Returns

Nothing

Referenced by `svuConvertU8ToF16()`.

7.370 svuConvertYUV400ToYUV422.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/convertYUV400ToYUV422/convertYUV400ToYUV422.h>
```

Functions

- void `ConvertYUV400ToYUV422` (UInt8 *in, UInt16 *out, UInt32 width)
- void `svuConvertYUV400ToYUV422` (SippFilter *fptr)
Shave function of the YUV400 to YUV422 conversion filter.

7.370.1 Function Documentation

`void ConvertYUV400ToYUV422 (UInt8 * in, UInt16 * out, UInt32 width)`

`ConvertYUV400ToYUV422`

Parameters

<code>in</code>	<i>out</i>	- pointer to output line
<code>in</code>	<i>in</i>	- pointer to input line
<code>in</code>	<i>width</i>	- width of input line

Referenced by `svuConvertYUV400ToYUV422()`.

7.371 svuConvGeneric.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/convGeneric/convGeneric.h>
```

Functions

- void [Convolution](#) ([UInt8](#) **in, [UInt8](#) **out, [UInt32](#) kernelSize, half *conv, [UInt32](#) inWidth)
- void [svuConvGeneric](#) ([SippFilter](#) *fptr)

Shave function of the [Generic Convolution](#) filter.

7.371.1 Function Documentation

void [Convolution](#) ([UInt8](#) ** in, [UInt8](#) ** out, [UInt32](#) kernelSize, half * conv, [UInt32](#) inWidth)

Referenced by [svuConvGeneric\(\)](#).

7.372 [svuConvSeparable11x11.c](#) File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/convSeparable11x11/convSeparable11x11.h>
```

Functions

- void [convSeparable11x11](#) ([UInt8](#) **out, [UInt8](#) **in, float conv[6], [UInt32](#) inWidth)
- void [svuConvSeparable11x11](#) ([SippFilter](#) *fptr)

Shave function of the [Convolution Separable 11x11](#) filter.

7.372.1 Function Documentation

void [convSeparable11x11](#) ([UInt8](#) ** out, [UInt8](#) ** in, float conv[6], [UInt32](#) inWidth)

Convolution Separable 11x11 kernel

Parameters

in	<i>out</i>	- array of pointers to output lines
in	<i>in</i>	- array of pointers to input lines
in	<i>conv</i>	- array of values from convolution
in	<i>inWidth</i>	- width of input line

Referenced by [svuConvSeparable11x11\(\)](#).

7.373 [svuConvSeparable11x11Fp16ToFp16.c](#) File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/convSeparable11x11Fp16ToFp16/convSeparable11x11-
Fp16ToFp16.h>
```

Functions

- void [convSeparable11x11Fp16ToFp16](#) (half **out, half **in, half conv[6], [UInt32](#) inWidth)
- void [svuConvSeparable11x11Fp16ToFp16](#) ([SippFilter](#) *fptr)

Shave function of the [Convolution Separable 11x11 Fp16ToFp16](#) filter.

7.373.1 Function Documentation

[void convSeparable11x11Fp16ToFp16](#) (half ** out, half ** in, half conv[6], [UInt32](#) inWidth)

Convolution Separable 11x11 Fp16ToFp16 kernel

Parameters

in	<i>out</i>	- array of pointers to output lines
in	<i>in</i>	- array of pointers to input lines
in	<i>conv</i>	- array of values from convolution
in	<i>inWidth</i>	- width of input line

Referenced by [svuConvSeparable11x11Fp16ToFp16\(\)](#).

7.374 [svuConvSeparable3x3.c](#) File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/convSeparable3x3/convSeparable3x3.h>
```

Functions

- void [convSeparable3x3](#) ([UInt8](#) *out, [UInt8](#) **in, float conv[2], [UInt32](#) inWidth)
- void [svuConvSeparable3x3](#) ([SippFilter](#) *fptr)

Shave function of the [Convolution Separable 3x3](#) filter.

7.374.1 Function Documentation

[void convSeparable3x3](#) ([UInt8](#) * out, [UInt8](#) ** in, float conv[2], [UInt32](#) inWidth)

Convolution Separable 3x3 kernel

Parameters

in	<i>out</i>	- array of pointers to output lines
in	<i>in</i>	- array of pointers to input lines
in	<i>conv</i>	- array of values from convolution
in	<i>inWidth</i>	- width of input line

Referenced by [svuConvSeparable3x3\(\)](#).

7.375 svuConvSeparable3x3Fp16ToFp16.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/convSeparable3x3Fp16ToFp16/convSeparable3x3Fp16ToFp16.h>
```

Functions

- void [convSeparable3x3Fp16ToFp16](#) (half *out, half **in, half conv[2], [UInt32](#) inWidth)
- void [svuConvSeparable3x3Fp16ToFp16](#) ([SippFilter](#) *fptr)
Shave function of the [Convolution Separable 3x3 Fp16ToFp16](#) filter.

7.375.1 Function Documentation

void [convSeparable3x3Fp16ToFp16](#) (half * out, half ** in, half conv[2], [UInt32](#) inWidth)

Convolution Separable 3x3 Fp16ToFp16 kernel

Parameters

in	<i>out</i>	- array of pointers to output lines
in	<i>in</i>	- array of pointers to input lines
in	<i>conv</i>	- array of values from convolution
in	<i>inWidth</i>	- width of input line

Referenced by [svuConvSeparable3x3Fp16ToFp16\(\)](#).

7.376 svuConvSeparable5x5.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/convSeparable5x5/convSeparable5x5.h>
```

Functions

- void [convSeparable5x5](#) ([UInt8](#) **out, [UInt8](#) **in, float conv[3], [UInt32](#) inWidth)
- void [svuConvSeparable5x5](#) ([SippFilter](#) *fptr)
Shave function of the [Convolution Separable 5x5](#) filter.

7.376.1 Function Documentation

void [convSeparable5x5](#) ([UInt8](#) ** out, [UInt8](#) ** in, float conv[3], [UInt32](#) inWidth)

Convolution Separable 5x5 kernel

Parameters

in	out	- array of pointers to output lines
in	in	- array of pointers to input lines
in	conv	- array of values from convolution
in	inWidth	- width of input line

Referenced by svuConvSeparable5x5().

7.377 svuConvSeparable5x5Fp16ToFp16.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/convSeparable5x5Fp16ToFp16/convSeparable5x5Fp16ToFp16.h>
```

Functions

- void [convSeparable5x5Fp16ToFp16](#) (half **out, half **in, half conv[3], [UInt32](#) inWidth)
- void [svuConvSeparable5x5Fp16ToFp16](#) ([SippFilter](#) *fptr)

Shave function of the [Convolution Separable 5x5 Fp16ToFp16](#) filter.

7.377.1 Function Documentation

[void convSeparable5x5Fp16ToFp16](#) (half ** out, half ** in, half conv[3], [UInt32](#) inWidth)

Convolution Separable 5x5 Fp16ToFp16 kernel

Parameters

in	out	- array of pointers to output lines
in	in	- array of pointers to input lines
in	conv	- array of values from convolution
in	inWidth	- width of input line

Referenced by svuConvSeparable5x5Fp16ToFp16().

7.378 svuConvSeparable7x7.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/convSeparable7x7/convSeparable7x7.h>
```

Functions

- void [convSeparable7x7](#) ([UInt8](#) **out, [UInt8](#) **in, float conv[4], [UInt32](#) inWidth)
- void [svuConvSeparable7x7](#) ([SippFilter](#) *fptr)

Shave function of the [Convolution Separable 7x7](#) filter.

7.378.1 Function Documentation

```
void convSeparable7x7 ( UInt8 ** out, UInt8 ** in, float conv[4], UInt32 inWidth )
```

Convolution Separable 7x7 kernel

Parameters

in	out	- array of pointers to output lines
in	in	- array of pointers to input lines
in	conv	- array of values from convolution
in	inWidth	- width of input line

Referenced by svuConvSeparable7x7().

7.379 svuConvSeparable7x7Fp16ToFp16.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/convSeparable7x7Fp16ToFp16/convSeparable7x7Fp16To-
Fp16.h>
```

Functions

- void [convSeparable7x7Fp16ToFp16](#) (half **out, half **in, half conv[4], UInt32 inWidth)
- void [svuConvSeparable7x7Fp16ToFp16](#) (SippFilter *fptr)

Shave function of the [Convolution Separable 7x7 Fp16ToFp16](#) filter.

7.379.1 Function Documentation

```
void convSeparable7x7Fp16ToFp16 ( half ** out, half ** in, half conv[4], UInt32 inWidth )
```

Convolution Separable 7x7 Fp16ToFp16 kernel

Parameters

in	out	- array of pointers to output lines
in	in	- array of pointers to input lines
in	conv	- array of values from convolution
in	inWidth	- width of input line

Referenced by svuConvSeparable7x7Fp16ToFp16().

7.380 svuConvSeparable9x9.c File Reference

```
#include <sipp.h>
```

```
#include <sippShaveMacros.h>
#include <filters/convSeparable9x9/convSeparable9x9.h>
```

Functions

- void [convSeparable9x9](#) (UInt8 **out, UInt8 **in, float conv[5], UInt32 inWidth)
- void [svuConvSeparable9x9](#) (SippFilter *fptr)

Shave function of the [Convolution Separable 9x9](#) filter.

7.380.1 Function Documentation

```
void convSeparable9x9 ( UInt8 ** out, UInt8 ** in, float conv[5], UInt32 inWidth )
```

Convolution Separable 9x9 kernel

Parameters

in	<i>out</i>	- array of pointers to output lines
in	<i>in</i>	- array of pointers to input lines
in	<i>conv</i>	- array of values from convolution
in	<i>inWidth</i>	- width of input line

Referenced by [svuConvSeparable9x9\(\)](#).

7.381 svuConvSeparable9x9Fp16ToFp16.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/convSeparable9x9Fp16ToFp16/convSeparable9x9Fp16To-
Fp16.h>
```

Functions

- void [convSeparable9x9Fp16ToFp16](#) (half **out, half **in, half conv[5], UInt32 inWidth)
- void [svuConvSeparable9x9Fp16ToFp16](#) (SippFilter *fptr)

Shave function of the [Convolution Separable 9x9 Fp16ToFp16](#) filter.

7.381.1 Function Documentation

```
void convSeparable9x9Fp16ToFp16 ( half ** out, half ** in, half conv[5], UInt32 inWidth )
```

Convolution Separable 9x9 Fp16ToFp16 kernel

Parameters

<code>in</code>	<code>out</code>	- array of pointers to output lines
<code>in</code>	<code>in</code>	- array of pointers to input lines
<code>in</code>	<code>conv</code>	- array of values from convolution
<code>in</code>	<code>inWidth</code>	- width of input line

Referenced by `svuConvSeparable9x9Fp16ToFp16()`.

7.382 `svuConvYuv444.c` File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/convYuv444/convYuv444.h>
```

Macros

- `#define SCALE 255.0f`

Functions

- void `svuRgbYuv444` (`SippFilter *fptr`)
Shave function of the [Convert to YUV444](#) filter.

7.382.1 Macro Definition Documentation

```
#define SCALE 255.0f
```

Referenced by `svuRgbYuv444()`.

7.383 `svuCopy.c` File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
```

Functions

- void `svuCopy` (`SippFilter *fptr`)
Shave function of the [Copy](#) filter.

7.384 `svuCornerMinEigenVal.c` File Reference

```
#include <sipp.h>
```

```
#include <sippShaveMacros.h>
#include <filters/cornerMinEigenVal/cornerMinEigenVal.h>
#include <math.h>
```

Macros

- #define MAX_WIDTH 1050
- #define PADDING 16
- #define INTERMEDIARY_BUFFER_LINE_NUMBER 24

Functions

- void CornerMinEigenVal (UInt8 **in_lines, UInt8 **out_line, UInt8 *buffer, UInt32 width)
- void svuCornerMinEigenVal (SippFilter *fptr)
Shave function of the Corner Min Eigenvalue filter.

Variables

- UInt8 buffer [(MAX_WIDTH+PADDING)*INTERMEDIARY_BUFFER_LINE_NUMBER]

7.384.1 Macro Definition Documentation

```
#define INTERMEDIARY_BUFFER_LINE_NUMBER 24

#define MAX_WIDTH 1050

#define PADDING 16
```

7.384.2 Function Documentation

```
void CornerMinEigenVal ( UInt8 ** in_lines, UInt8 ** out_line, UInt8 * buffer, UInt32 width )
```

CornerMinEigenVal filter - is 5x5 kernel size

Parameters

in	<i>input_lines</i>	- pointer to input pixel
out	<i>output_line</i>	- position on line
in	<i>width</i>	- width of line

Referenced by svuCornerMinEigenVal().

7.384.3 Variable Documentation

```
UInt8 buffer[(MAX_WIDTH+PADDING)*INTERMEDIARY_BUFFER_LINE_NUMBER]
```

7.385 svuCornerMinEigenValpatched.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/cornerMinEigenValpatched/cornerMinEigenValpatched.h>
```

Functions

- void [CornerMinEigenVal_patched](#) (UInt8 **in_lines, Int32 posx, UInt8 *out_pix, UInt32 width)
- void [svuCornerMinEigenValpatched](#) (SippFilter *fptr)
Shave function of the [Corner Min Eigenvalue Patched](#) filter.

7.385.1 Function Documentation

```
void CornerMinEigenVal_patched ( UInt8 ** in_lines, Int32 posx, UInt8 * out_pix, UInt32 width )
```

CornerMinEigenVal_patched filter - is 5x5 kernel size

Parameters

in	<i>input_buffer</i>	- pointer to input pixel
in	<i>posx</i>	- position on line
out	<i>out_pix</i>	- pointer to output pixel
in	<i>width</i>	- width of line
in	<i>stride</i>	- if it not exists we can put it to 0

Referenced by svuCornerMinEigenValpatched().

7.386 svuCrop.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/crop/crop.h>
```

Functions

- void [svuCrop](#) (SippFilter *fptr)
Shave function of the [Crop](#) filter.

7.387 svuCropCvtPlaneMode.c File Reference

```
#include <sipp.h>
#include <stdio.h>
#include <sippShaveMacros.h>
#include <filters/cropCvtPlaneMode/cropCvtPlaneMode.h>
```

Functions

- void `cropCvtPlaneMode` (`UInt8 **inR`, `UInt8 **inG`, `UInt8 **inB`, `UInt8 **Out`, `UInt32 width`)
- void `svuCropCvtPlaneMode` (`SippFilter *fptr`)

Shave function of the `Crop` filter.

7.387.1 Function Documentation

```
void cropCvtPlaneMode ( UInt8 ** inR, UInt8 ** inG, UInt8 ** inB, UInt8 ** Out, UInt32 width
)
```

Referenced by `svuCropCvtPlaneMode()`.

7.388 svuCvtColorChromaYUVToNV12.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/cvtColorChromaYUVToNV12/cvtColorChromaYUVToNV12.h>
```

Functions

- void `cvtColorChromaYUV420ToNV12` (`u8 *inU`, `u8 *inV`, `u8 *outUV`, `u32 width`)
- void `cvtColorChromaYUV444ToNV12` (`u8 *inU[2]`, `u8 *inV[2]`, `u8 *outUV`, `u32 width`)
- void `svuCvtColorChromaYUVToNV12` (`SippFilter *fptr`)

Shave function of the `YUV to NV12 chroma conversion` filter.

7.388.1 Function Documentation

```
void cvtColorChromaYUV420ToNV12 ( u8 * inU, u8 * inV, u8 * outUV, u32 width )
```

`cvtColorKernel` conversion YUV420p to NV12 chroma part only

Parameters

<code>in</code>	<code>inU</code>	- line from input U plane
<code>in</code>	<code>inV</code>	- line from input V plane
	<code>Out]</code>	<code>outUV</code> - UV plane in the first of the line the output NV12 image
<code>in</code>	<code>width</code>	- line width in pixels

Returns

Nothing

Referenced by `svuCvtColorChromaYUVToNV12()`.

```
void cvtColorChromaYUV444ToNV12 ( u8 * inU[2], u8 * inV[2], u8 * outUV, u32 width )
```

`cvtColorKernel` conversion YUV444p to NV12 chroma part only

Parameters

in	<i>inU[2]</i>	- 2 lines from input U plane
in	<i>inV[2]</i>	- 2 lines from input V plane
	<i>Out</i>	outUV - UV plane in the first of the line the output NV12 image
in	<i>width</i>	- line width in pixels

Returns

Nothing

Referenced by `svuCvtColorChromaYUVToNV12()`.

`void svuCvtColorChromaYUVToNV12 (SippFilter * fptr)`

Shave function of the [YUV to NV12 chroma conversion](#) filter.

7.389 svucvtColorNV21toRGB.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/cvtColorNV21toRGB/cvtColorNV21toRGB.h>
```

Functions

- void [cvtColorNV21toRGBImplementation](#) ([UInt8](#) **yin, [UInt8](#) **uvin, [UInt8](#) **outR, [UInt8](#) **outG, [UInt8](#) **outB, [UInt32](#) width)
- void [svucvtColorNV21toRGB](#) ([SippFilter](#) *fptr)

Shave function of the [NV21 to RGB conversion](#) filter.

7.389.1 Function Documentation

`void cvtColorNV21toRGBImplementation (UInt8 ** yin, UInt8 ** uvin, UInt8 ** outR, UInt8 ** outG, UInt8 ** outB, UInt32 width)`

`cvtColorKernel` to conversion NV21 to RGB Performs color space conversion: NV21 to RGB

Parameters

in	<i>yin</i>	input Y channel
in	<i>uvin</i>	input UV channel interleaved
out	<i>outR</i>	output R channel
out	<i>outG</i>	output G channel
out	<i>outB</i>	output B channel

<code>in</code>	<code>width</code>	- image width in pixels
-----------------	--------------------	-------------------------

Referenced by `svucvtColorNV21toRGB()`.

7.390 `svuCvtColorRGBfp16ToLumaU8.c` File Reference

```
#include <sipp.h>
#include <filters/cvtColorRGBfp16ToLumaU8/cvtColorRGBfp16ToLumaU8.h>
```

Functions

- void `cvtColorRGBfp16ToLumaU8` (half *inRGB[3], `UInt8` *yOut, `UInt32` width)
- void `svuCvtColorRGBfp16ToLumaU8` (`SippFilter` *fptr)
Shave function of the `RGB(fp16)` to `Luma(u8)` conversion filter.

7.390.1 Function Documentation

```
void cvtColorRGBfp16ToLumaU8 ( half * inRGB[3], UInt8 * yOut, UInt32 width )
```

Performs color space conversion RGBfp16 to LumaU8 for one line in an image

Parameters

<code>in</code>	<code>inRGB</code>	- pointer to the first lines from input RGB planes
	<code>Out]</code>	yOut - pointer to the luma plane
<code>in</code>	<code>width</code>	- line width in pixels

Referenced by `svuCvtColorRGBfp16ToLumaU8()`.

7.391 `svuCvtColorRGBfp16ToUV420U8.c` File Reference

```
#include <sipp.h>
#include <filters/cvtColorRGBfp16ToUV420U8/cvtColorRGBfp16ToUV420-
U8.h>
```

Functions

- void `cvtColorRGBfp16ToUV420U8` (half *inRGB[3], `UInt8` *uOut, `UInt8` *vOut, `UInt32` width)
- void `svuCvtColorRGBfp16ToUV420U8` (`SippFilter` *fptr)
Shave function of the `RGB(fp16)` to `UV420(u8)` conversion filter.

7.391.1 Function Documentation

```
void cvtColorRGBfp16ToUV420U8 ( half * inRGB[3], UInt8 * uOut, UInt8 * vOut, UInt32 width )
```

Performs color space conversion RGBfp16 to UV420U8 for one line in an image

Parameters

in	<i>inRGB</i>	- pointer to the first lines from input RGB planes
	<i>Out]</i>	uOut - pointer to the U plane
	<i>Out]</i>	vOut - pointer to the V plane
in	<i>width</i>	- line width in pixels

Referenced by `svuCvtColorRGBfp16ToUV420U8()`.

7.392 `svuCvtColorRGBtoChromaNV12.c` File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/cvtColorRGBtoChromaNV12/cvtColorRGBtoChromaNV12.h>
```

Functions

- void `cvtColorRGBtoChromaNV12` (`UInt8 *ln1RGB[3]`, `UInt8 *ln2RGB[3]`, `UInt8 *vuOut`, float `*coefsMat`, float `*offset`, `UInt32 width`)
- void `svuCvtColorRGBtoChromaNV12` (`SippFilter *fptr`)
Shave function of the [RGB to Chroma NV12 conversion filter](#).

7.392.1 Function Documentation

void `cvtColorRGBtoChromaNV12` (`UInt8 * ln1RGB[3]`, `UInt8 * ln2RGB[3]`, `UInt8 * vuOut`, float `* coefsMat`, float `* offset`, `UInt32 width`)

`cvtColorKernel` to conversion RGB to chromaNV12

Parameters

in	<i>inR</i>	input R channel
in	<i>inG</i>	input G channel
in	<i>inB</i>	input B channel
out	<i>uvOut</i>	output UV interleaved channel

Referenced by `svuCvtColorRGBtoChromaNV12()`.

7.393 `svuCvtColorRGBtoLuma.c` File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/cvtColorRGBtoLuma/cvtColorRGBtoLuma.h>
```

Functions

- void `cvtColorRGBtoLuma` (`UInt8 **inR`, `UInt8 **inG`, `UInt8 **inB`, `UInt8 **yOut`, `UInt32 width`)

- void `svuCvtColorRGBtoLuma` (`SippFilter *fptr`)
Shave function of the [RGB to Luma conversion filter](#).

7.393.1 Function Documentation

void `cvtColorRGBtoLuma` (`UInt8 ** inR`, `UInt8 ** inG`, `UInt8 ** inB`, `UInt8 ** yOut`, `UInt32 width`)

`cvtColorKernel` to conversion RGB to Luma

Parameters

in	<i>inR</i>	input R channel
in	<i>inG</i>	input G channel
in	<i>inB</i>	input B channel
out	<i>yOut</i>	output Y channel

Referenced by `svuCvtColorRGBtoLuma()`.

7.394 `svuCvtColorRGBtoLumaNV12.c` File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/cvtColorRGBtoLumaNV12/cvtColorRGBtoLumaNV12.h>
```

Functions

- void `cvtColorRGBtoLumaNV12` (`UInt8 *inR`, `UInt8 *inG`, `UInt8 *inB`, `UInt8 *yOut`, float *coefsMat, float *offset, `UInt32 width`)
- void `svuCvtColorRGBtoLumaNV12` (`SippFilter *fptr`)
Shave function of the [RGB to Luma NV12 conversion filter](#).

7.394.1 Function Documentation

void `cvtColorRGBtoLumaNV12` (`UInt8 * inR`, `UInt8 * inG`, `UInt8 * inB`, `UInt8 * yOut`, float * coefsMat, float * offset, `UInt32 width`)

`cvtColorKernel` to conversion RGB to Luma

Parameters

in	<i>inR</i>	input R channel
in	<i>inG</i>	input G channel
in	<i>inB</i>	input B channel
out	<i>yOut</i>	output Y channel

Referenced by `svuCvtColorRGBtoLumaNV12()`.

7.395 svuCvtColorRGBtoUV.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/cvtColorRGBtoUV/cvtColorRGBtoUV.h>
```

Functions

- void **cvtColorRGBtoUV** (**UInt8** **inR, **UInt8** **inG, **UInt8** **inB, **UInt8** **uvOut, **UInt32** width, **UInt32** line)
- void **svuCvtColorRGBtoUV** (**SippFilter** *fptr)
*Shave function of the **RGB to UV conversion** filter.*

7.395.1 Function Documentation

void cvtColorRGBtoUV (**UInt8** ** inR, **UInt8** ** inG, **UInt8** ** inB, **UInt8** ** uvOut, **UInt32** width, **UInt32** line)

cvtColorKernel to conversion RGB to UV

Parameters

in	<i>inR</i>	input R channel
in	<i>inG</i>	input G channel
in	<i>inB</i>	input B channel
out	<i>uvOut</i>	output UV channels

Referenced by svuCvtColorRGBtoUV().

7.396 svuCvtColorRGBtoUV420.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/cvtColorRGBtoUV420/cvtColorRGBtoUV420.h>
```

Functions

- void **cvtColorRGBtoUV420** (**UInt8** **inR, **UInt8** **inG, **UInt8** **inB, **UInt8** **uOut, **UInt8** **vOut, **UInt32** width)
- void **svuCvtColorRGBtoUV420** (**SippFilter** *fptr)
*Shave function of the **RGB to UV420 conversion** filter.*

7.396.1 Function Documentation

```
void cvtColorRGBtoUV420 ( UInt8 ** inR, UInt8 ** inG, UInt8 ** inB, UInt8 ** uOut, UInt8 **  
vOut, UInt32 width )
```

cvtColorKernel to conversion RGB to UV420

Parameters

in	<i>inR</i>	input R channel
in	<i>inG</i>	input G channel
in	<i>inB</i>	input B channel
out	<i>uOut</i>	output U channel
out	<i>vOut</i>	output V channel

Referenced by `svuCvtColorRGBtoUV420()`.

7.397 `svuCvtColorRGBToYUV422.c` File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/cvtColorRGBToYUV422/cvtColorRGBToYUV422.h>
```

Functions

- void `cvtColorKernelRGBToYUV422` (`UInt8 **rIn`, `UInt8 **gIn`, `UInt8 **bIn`, `UInt8 **output`, `UInt32 width`)
- void `svuCvtColorRGBToYUV422` (`SippFilter *fptr`)
Shave function of the [RGB to YUV422 conversion](#) filter.

7.397.1 Function Documentation

void `cvtColorKernelRGBToYUV422` (`UInt8 ** rIn`, `UInt8 ** gIn`, `UInt8 ** bIn`, `UInt8 ** output`, `UInt32 width`)

Performs color space conversion RGB to YUV422 for one line

Parameters

in	<i>rIn</i>	- pointer to the input line that contain R values from RGB
in	<i>gIn</i>	- pointer to the input line that contain G values from RGB
in	<i>bIn</i>	- pointer to the input line that contain B values from RGB
out	<i>output</i>	- pointer to the output line YUV422 interleaved
in	<i>width</i>	- line width

Referenced by `svuCvtColorRGBToYUV422()`.

7.398 `svuCvtColorYUV422ToRGB.c` File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/cvtColorYUV422ToRGB/cvtColorYUV422ToRGB.h>
```

Functions

- void `cvtColorKernelYUV422ToRGB` (`UInt8 **input`, `UInt8 **rOut`, `UInt8 **gOut`, `UInt8 **bOut`, `UInt32 width`)
- void `svuCvtColorYUV422ToRGB` (`SippFilter *fptr`)
Shave function of the `cvtColorYUV422ToRGB` filter.

7.398.1 Function Documentation

void `cvtColorKernelYUV422ToRGB` (`UInt8 ** input`, `UInt8 ** rOut`, `UInt8 ** gOut`, `UInt8 ** bOut`, `UInt32 width`)

Referenced by `svuCvtColorYUV422ToRGB()`.

7.399 svuCvtColorYUVToRGB.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/cvtColorYUVToRGB/cvtColorYUVToRGB.h>
```

Functions

- void `cvtColorKernelYUVToRGB` (`UInt8 *yIn`, `UInt8 *uIn`, `UInt8 *vIn`, `UInt8 *out`, `UInt32 width`)
- void `svuCvtColorYUVToRGB` (`SippFilter *fptr`)
Shave function of the `YUV to RGB conversion` filter.

7.399.1 Function Documentation

void `cvtColorKernelYUVToRGB` (`UInt8 * yIn`, `UInt8 * uIn`, `UInt8 * vIn`, `UInt8 * out`, `UInt32 width`)

Referenced by `svuCvtColorYUVToRGB()`.

7.400 svuDilate3x3.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/dilate3x3/dilate3x3.h>
```

Functions

- void `Dilate3x3` (`UInt8 **src`, `UInt8 **dst`, `UInt8 **kernel`, `UInt32 width`)
- void `svuDilate3x3` (`SippFilter *fptr`)
Shave function of the `Dilate 3x3` filter.

7.400.1 Function Documentation

void Dilate3x3 (UInt8 ** src, UInt8 ** dst, UInt8 ** kernel, UInt32 width)

Dilate3x3 kernel

Parameters

in	src	- array of pointers to input lines of the input image
out	dst	- array of pointers to output lines
in	kernel	- array of pointers to input kernel
in	width	- width of the input line

Referenced by svuDilate3x3().

7.401 svuDilate5x5.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/dilate5x5/dilate5x5.h>
```

Functions

- void Dilate5x5 (UInt8 **src, UInt8 **dst, UInt8 **kernel, UInt32 width)
- void svuDilate5x5 (SippFilter *fptr)
Shave function of the Dilate 5x5 filter.

7.401.1 Function Documentation

void Dilate5x5 (UInt8 ** src, UInt8 ** dst, UInt8 ** kernel, UInt32 width)

Dilate5x5 kernel

Parameters

in	src	- array of pointers to input lines of the input image
out	dst	- array of pointers to output lines
in	kernel	- array of pointers to input kernel
in	width	- width of the input line

Referenced by svuDilate5x5().

7.402 svuDilate7x7.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/dilate7x7/dilate7x7.h>
```

Functions

- void [Dilate7x7](#) ([UInt8](#) **src, [UInt8](#) **dst, [UInt8](#) **kernel, [UInt32](#) width)
- void [svuDilate7x7](#) ([SippFilter](#) *fptr)
Shave function of the [Dilate 7x7](#) filter.

7.402.1 Function Documentation

void [Dilate7x7](#) ([UInt8](#) ** src, [UInt8](#) ** dst, [UInt8](#) ** kernel, [UInt32](#) width)

Referenced by [svuDilate7x7\(\)](#).

7.403 [svuDilateGeneric.c](#) File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/dilateGeneric/dilateGeneric.h>
```

Functions

- void [DilateGeneric](#) ([UInt8](#) **src, [UInt8](#) **dst, [UInt8](#) **kernel, [UInt32](#) width, [UInt32](#) height, [UInt32](#) k)
- void [svuDilateGeneric](#) ([SippFilter](#) *fptr)
Shave function of the [Generic Dilate](#) filter.

7.403.1 Function Documentation

void [DilateGeneric](#) ([UInt8](#) ** src, [UInt8](#) ** dst, [UInt8](#) ** kernel, [UInt32](#) width, [UInt32](#) height, [UInt32](#) k)

Referenced by [svuDilateGeneric\(\)](#).

7.404 [svudisp2depth.c](#) File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/disp2depth/disp2depth.h>
#include <filters/disp2depth/shave/disp2depth_exec.h>
```

Functions

- void [flip_disp2depth_explic_vect](#) (uint8_t *input, half8 *output, half *LUT, uint32_t width)
disp2depth kernel
- void [disp2depth_explic_vect](#) (uint8_t *input, half8 *output, half *LUT, uint32_t width)
- void [svudisp2depth](#) ([SippFilter](#) *fptr)

7.404.1 Function Documentation

```
void dsp2depth_explic_vect ( uint8_t * input, half8 * output, half * LUT, uint32_t width )  
  
void flip_dsp2depth_explic_vect ( uint8_t * input, half8 * output, half * LUT, uint32_t width )
```

disp2depth kernel

```
void svudisp2depth ( SippFilter * fptr )
```

7.405 svuEqualizeHist.c File Reference

```
#include <sipp.h>  
#include <sippShaveMacros.h>  
#include <filters/equalizeHist/equalizeHist.h>
```

Functions

- void [equalizeHist](#) (UInt8 **in, UInt8 **out, UInt32 *hist, UInt32 width)
- void [svuEqualizeHist](#) (SippFilter *fptr)
Shave function of the [Equalize Histogram](#) filter.

7.405.1 Function Documentation

```
void equalizeHist ( UInt8 ** in, UInt8 ** out, UInt32 * hist, UInt32 width )
```

equalizehistogram kernel - makes an equalization trough an image with a given histogram

Parameters

in	<i>in</i>	- array of pointers to input lines
out		- array of pointers to output lines
in	<i>hist</i>	- pointer to an input array that indicates the cumulative histogram of the image
in	<i>width</i>	- width of input line

Referenced by [svuEqualizeHist](#)().

7.406 svuErode3x3.c File Reference

```
#include <sipp.h>  
#include <sippShaveMacros.h>  
#include <filters/erode3x3/erode3x3.h>
```

Functions

- void [Erode3x3](#) (UInt8 **src, UInt8 **dst, UInt8 **kernel, UInt32 width)

- void `svuErode3x3` (`SippFilter *fptr`)
Shave function of the `Erode 3x3` filter.

7.406.1 Function Documentation

void `Erode3x3` (`UInt8 ** src`, `UInt8 ** dst`, `UInt8 ** kernel`, `UInt32 width`)

`Erode3x3` kernel

Parameters

in	<i>src</i>	- array of pointers to input lines of the input image
out	<i>dst</i>	- array of pointers to output lines
in	<i>kernel</i>	- array of pointers to input kernel
in	<i>width</i>	- width of the input line

Referenced by `svuErode3x3()`.

7.407 `svuErode5x5.c` File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/erode5x5/erode5x5.h>
```

Functions

- void `Erode5x5` (`UInt8 **src`, `UInt8 **dst`, `UInt8 **kernel`, `UInt32 width`)
- void `svuErode5x5` (`SippFilter *fptr`)
Shave function of the `Erode 5x5` filter.

7.407.1 Function Documentation

void `Erode5x5` (`UInt8 ** src`, `UInt8 ** dst`, `UInt8 ** kernel`, `UInt32 width`)

`Erode5x5` kernel

Parameters

in	<i>src</i>	- array of pointers to input lines of the input image
out	<i>dst</i>	- array of pointers to output lines
in	<i>kernel</i>	- array of pointers to input kernel
in	<i>width</i>	- width of the input line

Referenced by `svuErode5x5()`.

7.408 `svuErode7x7.c` File Reference

```
#include <sipp.h>
```

```
#include <sippShaveMacros.h>
#include <filters/erode7x7/erode7x7.h>
```

Functions

- void [Erode7x7](#) ([UInt8](#) **src, [UInt8](#) **dst, [UInt8](#) **kernel, [UInt32](#) width)
- void [svuErode7x7](#) ([SippFilter](#) *fptr)

Shave function of the [Erode 7x7](#) filter.

7.408.1 Function Documentation

```
void Erode7x7 ( UInt8 ** src, UInt8 ** dst, UInt8 ** kernel, UInt32 width )
```

Erode7x7 kernel

Parameters

in	<i>src</i>	- array of pointers to input lines of the input image
out	<i>dst</i>	- array of pointers to output lines
in	<i>kernel</i>	- array of pointers to input kernel
in	<i>width</i>	- width of the input line

Referenced by [svuErode7x7\(\)](#).

7.409 svuExtAfStats.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <filters/extAfStats/extAfStats.h>
#include "string.h"
```

Macros

- #define [OCR_STOP_GO](#) 0x0004
- #define [OSR_SWI_HALT](#) 0x00008
- #define [IRF_BASE](#) 0x180
- #define [SVU_PTR](#) 0x024
- #define [SVU_OCR](#) 0x000
- #define [SVU_IRR](#) 0x010
- #define [SVU_OSR](#) 0x004
- #define [SET_REG_WORD](#)(a, x) (((void)(*(volatile [u32*](#))((((unsigned)(a)))) = ([u32](#))(x))))
- #define [GET_REG_WORD_VAL](#)(a) (*(volatile [u32*](#))((((unsigned)(a))))

Functions

- void `svuExtAfStats` (`SippFilter *fptr`)
Shave function of the AF Stats filter.

7.409.1 Macro Definition Documentation

```
#define GET_REG_WORD_VAL( a ) (*(volatile u32*)((unsigned)(a)))
```

Referenced by `sippCoreHwInitialLoad()`, `sippGenericRuntimeClaimHWResource()`, `sippGenericRuntimeLoadPipeline()`, `sippGenericStartHWUnits2x5x()`, `sippGenericWaitUnits()`, `sippHWSessionRemoveActiveLists()`, `sippIbflDecHandler()`, `sippObflIncHandler()`, `svuExtAfStats()`, `svuExtStatsSatPixelsU32()`, and `topLevelCmxDmaIrqHandler()`.

```
#define IRF_BASE 0x180
```

Referenced by `svuExtAfStats()`, and `svuExtStatsSatPixelsU32()`.

```
#define OCR_STOP_GO 0x0004
```

Referenced by `svuExtAfStats()`, and `svuExtStatsSatPixelsU32()`.

```
#define OSR_SWI_HALT 0x00008
```

Referenced by `svuExtAfStats()`, and `svuExtStatsSatPixelsU32()`.

```
#define SET_REG_WORD( a, x ) ((void)(*(volatile u32*)((unsigned)(a))) = (u32)(x))
```

Referenced by `sippBufSetupIrqRate()`, `sippCmxDmaInitAsync()`, `sippCoreHwInitialLoad()`, `sippGenericRuntimeLoadPipeline()`, `sippGenericStartHWUnits2x5x()`, `sippGenericStartUnits()`, `sippGenericUpdateHWUnits2x5x()`, `sippGenericWaitUnits()`, `sippHWSessionInit()`, `sippHWSessionRemoveActiveLists()`, `sippIbflDecHandler()`, `sippInitDma()`, `sippIntBarrierSetup()`, `sippKickSvus()`, `sippLoadMipiRx()`, `sippLoadMipiTx()`, `sippLoadSigma()`, `sippObflIncHandler()`, `sippSetOBufLevelsMipiRx0()`, `sippSetOBufLevelsMipiRx1()`, `sippSetOBufLevelsMipiRx2()`, `sippSetOBufLevelsMipiRx3()`, `sippSetOBufLevelsSigma()`, `sippStopSvus()`, `sippSvuDoneIrqHandler()`, `svuExtAfStats()`, and `svuExtStatsSatPixelsU32()`.

```
#define SVU_IRR 0x010
```

Referenced by `sippStopSvus()`, `svuExtAfStats()`, and `svuExtStatsSatPixelsU32()`.

```
#define SVU_OCR 0x000
```

Referenced by `svuExtAfStats()`, and `svuExtStatsSatPixelsU32()`.

```
#define SVU_OSR 0x004
```

Referenced by `svuExtAfStats()`, and `svuExtStatsSatPixelsU32()`.

```
#define SVU_PTR 0x024
```

Referenced by `svuExtAfStats()`, and `svuExtStatsSatPixelsU32()`.

7.410 svuExtStatsSatPixelsU32.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/extStatsSatPixelsU32/extStatsSatPixelsU32.h>
#include <string.h>
#include <stdlib.h>
```

Macros

- `#define OCR_STOP_GO 0x0004`
- `#define OSR_SWI_HALT 0x00008`
- `#define IRF_BASE 0x180`
- `#define SVU_PTR 0x024`
- `#define SVU_OCR 0x000`
- `#define SVU_IRR 0x010`
- `#define SVU_OSR 0x004`
- `#define SET_REG_WORD(a, x) ((void)(*(volatile u32*)((unsigned)(a))) = (u32)(x))`
- `#define GET_REG_WORD_VAL(a) (*(volatile u32*)((unsigned)(a)))`

Functions

- `void svuExtStatsSatPixelsU32 (SippFilter *fptr)`

7.410.1 Macro Definition Documentation

```
#define GET_REG_WORD_VAL( a ) (*(volatile u32*)((unsigned)(a)))
```

```
#define IRF_BASE 0x180
```

```
#define OCR_STOP_GO 0x0004
```

```
#define OSR_SWI_HALT 0x00008
```

```
#define SET_REG_WORD( a, x ) ((void)(*(volatile u32*)((unsigned)(a))) = (u32)(x))
```

```
#define SVU_IRR 0x010
```

```
#define SVU_OCR 0x000
```

```
#define SVU_OSR 0x004
```

```
#define SVU_PTR 0x024
```

7.410.2 Function Documentation

void svuExtStatsSatPixelsU32 (**SippFilter** * fptr)

7.411 svuFast9M2.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/fast9M2/fast9M2.h>
```

Macros

- #define **adiff**(a, b) ((a)>(b)?((a)-(b)):((b)-(a)))

Functions

- void **fastExclude** (**UInt8** **row, **UInt32** *posValid, **UInt32** *nrOfPoints, **UInt32** thresh, **UInt32** width)
- **UInt8** satu8add (**UInt8** a, **UInt8** b)
- **UInt8** satu8sub (**UInt8** a, **UInt8** b)
- void **fastBitFlag** (**UInt8** **row, **UInt32** *posValid, **UInt8** *scores, **UInt16** *cornerPositions, **UInt32** thresh, **UInt32** nrOfPoints)
- void **fast9M2** (**UInt8** **row, **UInt8** *score, **UInt16** *base, **UInt32** thresh, **UInt32** width)
- void **svuFast9M2** (**SippFilter** *fptr)
*Shave function of the **Fast9M2** filter.*

7.411.1 Macro Definition Documentation

```
#define adiff( a, b ) ((a)>(b)?((a)-(b)):((b)-(a)))
```

Fast9 - corner detection

Parameters

in	<i>in_lines</i>	- array of pointers to input lines
out	<i>score</i>	- pointer to corner score buffer ;; first unsigned int element is the number of candidates,
out	<i>base</i>	- pointer to corner candidates buffer ; first unsigned int element is the number of candidates, the rest are the position of coordinates
in	<i>thresh</i>	- threshold
in	<i>width</i>	- number of pixels to process

Referenced by fastExclude().

7.411.2 Function Documentation

void fast9M2 (**UInt8** ** row, **UInt8** * score, **UInt16** * base, **UInt32** thresh, **UInt32** width)

Referenced by svuFast9M2().

```
void fastBitFlag ( UInt8 ** row, UInt32 * posValid, UInt8 * scores, UInt16 * cornerPositions,
  UInt32 thresh, UInt32 nrOfPoints )
```

Referenced by fast9M2().

```
void fastExclude ( UInt8 ** row, UInt32 * posValid, UInt32 * nrOfPoints, UInt32 thresh, UInt32
  width )
```

Referenced by fast9M2().

```
UInt8 satu8add ( UInt8 a, UInt8 b )
```

Referenced by fastBitFlag().

```
UInt8 satu8sub ( UInt8 a, UInt8 b )
```

Referenced by fastBitFlag().

7.412 svuFast9ScoreCv.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/fast9ScoreCv/fast9ScoreCv.h>
#include <string.h>
#include <stdlib.h>
```

Macros

- #define `adiff(a, b)` $((a) > (b) ? ((a) - (b)) : ((b) - (a)))$

Functions

- void `fastExcludePos` (UInt8 **row, unsigned int *posValid, unsigned int *nrOfPoints, unsigned int thresh, unsigned int width)
- void `fastFlagBit` (UInt8 **row, unsigned int *posValid, UInt8 *scores, UInt16 *cornerPositions, unsigned int thresh, unsigned int nrOfPoints)
- void `fastScore` (UInt8 *score, UInt8 *scoresInput, unsigned int thresh, unsigned int nrOfPoints)
- void `mvcvfast9ScoreCv` (UInt8 **row, UInt8 *score, UInt16 *base, unsigned int thresh, unsigned int width, void *bulkBuff)
- UInt8 `satUInt8add` (UInt8 a, UInt8 b)
- UInt8 `satUInt8sub` (UInt8 a, UInt8 b)
- unsigned short `rightrot` (unsigned short x, unsigned int n)
- void `vectorRotate` (UInt8 *vect, UInt32 rotVal)
- UInt8 `minimumCalc` (UInt8 a, UInt8 b)
- void `vectorShift` (UInt8 *vect, UInt32 shiftVal)
- UInt16 `ones` (UInt16 f)

- void **svuFast9ScoreCv** (**SippFilter** *fptr)
*Shave function of the **Fast9ScoreCv** filter.*

Variables

- **UInt8** **bulkBuff** [20 * 1920]

7.412.1 Macro Definition Documentation

```
#define adiff( a, b ) ((a)>(b)?((a)-(b)):((b)-(a)))
```

Fast9 - corner detection

Parameters

in	<i>in_lines</i>	- array of pointers to input lines
out	<i>score</i>	- pointer to corner score buffer ;; first unsigned int element is the number of candidates,
out	<i>base</i>	- pointer to corner candidates buffer ; first unsigned int element is the number of candidates, the rest are the position of coordinates
in	<i>thresh</i>	- threshold
in	<i>width</i>	- number of pixels to process

Referenced by fastExcludePos().

7.412.2 Function Documentation

void fastExcludePos (**UInt8** ** row, unsigned int * posValid, unsigned int * nrOfPoints, unsigned int thresh, unsigned int width)

Referenced by mvcvfast9ScoreCv().

void fastFlagBit (**UInt8** ** row, unsigned int * posValid, **UInt8** * scores, **UInt16** * cornerPositions, unsigned int thresh, unsigned int nrOfPoints)

Referenced by mvcvfast9ScoreCv().

void fastScore (**UInt8** * score, **UInt8** * scoresInput, unsigned int thresh, unsigned int nrOfPoints)

Referenced by mvcvfast9ScoreCv().

UInt8 minimumCalc (**UInt8** a, **UInt8** b)

Referenced by fastScore().

void mvcvfast9ScoreCv (**UInt8** ** row, **UInt8** * score, **UInt16** * base, unsigned int thresh, unsigned int width, void * bulkBuff)

Referenced by svuFast9ScoreCv().

UInt16 ones (**UInt16** f)

Referenced by fastScore().

unsigned short rightrot (unsigned short x, unsigned int n)

UInt8 satUInt8add (**UInt8** a, **UInt8** b)

Referenced by fastFlagBit().

UInt8 satUInt8sub (**UInt8** a, **UInt8** b)

Referenced by fastFlagBit().

void vectorRotate (**UInt8** * vect, **UInt32** rotVal)

Referenced by fastScore().

void vectorShift (**UInt8** * vect, **UInt32** shiftVal)

7.412.3 Variable Documentation

UInt8 bulkBuff[20 * 1920]

Referenced by mvcvfast9ScoreCv(), and svuFast9ScoreCv().

7.413 svuGauss.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/gauss/gauss.h>
```

Functions

- void **gauss** (**UInt8** **inLine, **UInt8** **out, **UInt32** width)
- void **svuGauss** (**SippFilter** *fptr)

*Shave function of the **Gauss Blur** filter.*

7.413.1 Function Documentation

void **gauss** (**UInt8** ** inLine, **UInt8** ** out, **UInt32** width)

gaussian filter

Parameters

in	<i>in</i>	- array of pointers to input lines
out	<i>out</i>	- array of pointers for output lines
in	<i>width</i>	- width of input line

Referenced by `svuGauss()`.

7.414 `svuGaussHx2.c` File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/gaussHx2/gaussHx2.h>
```

Functions

- void `mvGaussHx2` (`UInt8 *inLine`, `UInt8 *outLine`, int width)
- void `svuGaussHx2` (`SippFilter *fptr`)

Shave function of the `GaussHx2` filter.

7.414.1 Function Documentation

`void mvGaussHx2 (UInt8 * inLine, UInt8 * outLine, int width)`

Apply downscale 2x horizontal with a gaussian filters with kernel 5x5. Have to be used in combination with `GaussVx2` to obtain correct output.

Parameters

in	<i>inLine</i>	- input line
out	<i>outLine</i>	- output resized line
in	<i>width</i>	- input line width (must be multiple of 16)

Referenced by `svuGaussHx2()`.

7.415 `svuGaussHx2_fp16.c` File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/gaussHx2_fp16/gaussHx2_fp16.h>
```

Functions

- void `GaussHx2_fp16` (half **inLine, half **outLine, `Int32` width)
- void `svuGaussHx2_fp16` (`SippFilter *fptr`)

Shave function of the `GaussHx2_fp16` filter.

7.415.1 Function Documentation

`void GaussHx2_fp16 (half ** inLine, half ** outLine, Int32 width)`

Apply downscale 2x horizontal with a gaussian filters with kernel 5x5. Have to be used in combination with GaussVx2 to obtain correct output.

Parameters

in	<i>inLine</i>	- input line
out	<i>outLine</i>	- output resized line
in	<i>width</i>	- input line width (must be multiple of 16)

Referenced by `svuGaussHx2_fp16()`.

7.416 svuGaussVx2.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/gaussVx2/gaussVx2.h>
```

Functions

- void `mvcvGaussVx2` (`UInt8 **inLine`, `UInt8 *outLine`, `int width`)
- void `svuGaussVx2` (`SippFilter *fptr`)

Shave function of the [GaussVx2](#) filter.

7.416.1 Function Documentation

`void mvcvGaussVx2 (UInt8 ** inLine, UInt8 * outLine, int width)`

Apply downscale 2x vertical with a gaussian filters with kernel 5x5. Have to be used in combination with GaussVx2 to obtain correct output.

Parameters

in	<i>inLine</i>	- input line
out	<i>outLine</i>	- output resized line
in	<i>width</i>	- input line width (must be multiple of 16)

Referenced by `svuGaussVx2()`.

7.417 svuGaussVx2_fp16.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/gaussVx2_fp16/gaussVx2_fp16.h>
```

Functions

- void **GaussVx2_fp16** (half **inLine, half **outLine, **Int32** width)
- void **svuGaussVx2_fp16** (**SippFilter** *fptr)
*Shave function of the **GaussVx2_fp16** filter.*

7.417.1 Function Documentation

void GaussVx2_fp16 (half ** inLine, half ** outLine, **Int32** width)

Apply downscale 2x horizontal with a gaussian filters with kernel 5x5. Have to be used in combination with GaussVx2 to obtain correct output.

Parameters

in	<i>inLine</i>	- input line
out	<i>outLine</i>	- output resized line
in	<i>width</i>	- input line width (must be multiple of 16)

Referenced by svuGaussVx2_fp16().

7.418 svuGenChroma.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/genChroma/genChroma.h>
```

Functions

- void **genChroma** (**UInt8** *inRGB[3], **UInt8** *inY, **UInt8** *outC[3], int width, int eps)
- void **svuGenChroma** (**SippFilter** *fptr)
*Shave function of the **Generate Chroma** filter.*

7.418.1 Function Documentation

void genChroma (**UInt8** * inRGB[3], **UInt8** * inY, **UInt8** * outC[3], int width, int eps)

Referenced by svuGenChroma().

7.419 svuGenChromaSS.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/genChromaSS/genChromaSS.h>
```

Functions

- void [GenChromaSS](#) ([UInt8](#) *outC[3], [UInt8](#) **inRGB, int eps, float chromaScale[3], [UInt32](#) width)
- void [svuGenChromaSS](#) ([SippFilter](#) *fptr)

Shave function of the [Generate Chroma with subsampling filter](#).

7.419.1 Function Documentation

void [GenChromaSS](#) ([UInt8](#) * outC[3], [UInt8](#) ** inRGB, int eps, float chromaScale[3], [UInt32](#) width)

Referenced by [svuGenChromaSS\(\)](#).

7.420 [svuGenDnsRef.c](#) File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/genDnsRef/genDnsRef.h>
```

Functions

- void [genDnsRef](#) ([UInt8](#) *inY, [UInt32](#) yc, [UInt32](#) x0, [YDnsRefParam](#) *param, [UInt8](#) *outRef, [UInt32](#) width)
- void [svuGenDnsRef](#) ([SippFilter](#) *fptr)

Shave function of the [Generate Reference for Luma Denoise filter](#).

7.420.1 Function Documentation

void [genDnsRef](#) ([UInt8](#) * inY, [UInt32](#) yc, [UInt32](#) x0, [YDnsRefParam](#) * param, [UInt8](#) * outRef, [UInt32](#) width)

Referenced by [svuGenDnsRef\(\)](#).

7.421 [svuGenDnsRefFp16.c](#) File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/genDnsRefFp16/genDnsRefFp16.h>
```

Functions

- void [genDnsRefFp16](#) (half *inY, unsigned int yc, int x0, [YDnsRefFp16Param](#) *param, unsigned char *outRef, unsigned int width)

- void **svuGenDnsRefFp16** (**SippFilter** *fptr)
Shave function of the [Generate Reference for Luma Denoise\(fp16 input\)](#) filter.

7.421.1 Function Documentation

void **genDnsRefFp16** (half * inY, unsigned int yc, int x0, **YDnsRefFp16Param** * param, unsigned char * outRef, unsigned int width)

Referenced by **svuGenDnsRefFp16**().

7.422 svuGenLuma.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/genLuma/genLuma.h>
```

Functions

- void **genLuma** (**UInt8** *inR, **UInt8** *inG, **UInt8** *inB, **UInt8** *outY, int width)
- void **svuGenLuma** (**SippFilter** *fptr)
Shave function of the [Luma Blur](#) filter.

7.422.1 Function Documentation

void **genLuma** (**UInt8** * inR, **UInt8** * inG, **UInt8** * inB, **UInt8** * outY, int width)

Referenced by **svuGenLuma**().

7.423 svuGenLumaU8Fp16.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/genLumaU8Fp16/genLumaU8Fp16.h>
```

Functions

- void **genLumaU8Fp16** (**UInt8** *inR, **UInt8** *inG, **UInt8** *inB, half *outY, half *coefs, int width)
- void **svuGenLumaU8Fp16** (**SippFilter** *fptr)

7.423.1 Function Documentation

```
void genLumaU8Fp16 ( UInt8 * inR, UInt8 * inG, UInt8 * inB, half * outY, half * coefs, int width
)
```

Referenced by svuGenLumaU8Fp16().

7.424 svuGreyDesat.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/greyDesat/greyDesat.h>
```

Functions

- void [greyDesat](#) (UInt8 **in, UInt8 **out, Int32 offset, Int32 slope, Int32 grey[3], UInt32 width)
- void [svuGreyDesat](#) (SippFilter *fptr)
Shave function of the [greyDesat](#) filter.

7.424.1 Function Documentation

```
void greyDesat ( UInt8 ** in, UInt8 ** out, Int32 offset, Int32 slope, Int32 grey[3], UInt32 width
)
```

Referenced by svuGreyDesat().

7.425 svuHammingDistance.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/hammingDistance/hammingDistance.h>
```

Functions

- int [countBit](#) (unsigned int resultXOR)
- void [mvcvHammingDistance](#) (UInt8 *d1, UInt8 *d2, UInt32 array_size, UInt32 descriptor_size, UInt16 *distances)
- void [svuHammingDistance](#) (SippFilter *fptr)
Shave function of the [hammingDistance](#) filter.

7.425.1 Function Documentation

```
int countBit ( unsigned int resultXOR )
```

This kernel find matches between two descriptors

Parameters

in	<i>d1</i>	- First descriptor
in	<i>d2</i>	- Second descriptor
in	<i>array_size</i>	- Size of array used to compute the distances
in	<i>descriptor_size</i>	- Size of the descriptor (32, 64 or 128 bytes)
out	<i>out</i>	- Pointer to distances

Returns

Nothing

Referenced by `mvsvHammingDistance()`.

```
void mvsvHammingDistance ( UInt8 * d1, UInt8 * d2, UInt32 array_size, UInt32 descriptor_size,
                          UInt16 * distances )
```

Referenced by `svuHammingDistance()`.

7.426 svuHarrisResponse.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/harrisResponse/harrisResponse.h>
```

Macros

- #define `HARRIS_SW_RADIUS` (3)

Functions

- float `HarrisResponse` (UInt8 *patchStart, UInt32 x, UInt32 y, UInt32 stepWidth, float k)
- void `svuHarrisResponse` (SippFilter *fptr)
Shave function of the `harrisResponse` filter.

7.426.1 Macro Definition Documentation

```
#define HARRIS_SW_RADIUS (3)
```

Referenced by `HarrisResponse()`.

7.426.2 Function Documentation

```
float HarrisResponse ( UInt8 * patchStart, UInt32 x, UInt32 y, UInt32 stepWidth, float k )
```

Computes Harris response over a patch of the image with a radius of 3. The patch size is actually 8x8 to account for borders

Parameters

in	<i>data</i>	- Input patch including borders
in	<i>x</i>	- X coordinate inside the patch. Only a value of 3 supported
in	<i>y</i>	- Y coordinate inside the patch. Only a value of 3 supported
in	<i>step_width</i>	- Step of the patch. Only a value 8 supported (2xradius + 2xborder)
in	<i>k</i>	- Constant that changes the response to the edges. Typically 0.02 is used return - Corner response value

Referenced by `svuHarrisResponse()`.

7.427 svuHistogram.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/histogram/histogram.h>
```

Functions

- void `histogram` (`UInt8 **in`, `UInt32 *hist`, `UInt32 width`)
- void `svuHistogram` (`SippFilter *fptr`)

Shave function of the [Histogram](#) filter.

7.427.1 Function Documentation

`void histogram (UInt8 ** in, UInt32 * hist, UInt32 width)`

histogram kernel - makes a histogram on a given plane

Parameters

in	<i>in</i>	- array of pointers to input lines
out		- array oh values from histogram
in	<i>width</i>	- width of input line

Referenced by `svuHistogram()`.

7.428 svuHistogramStat.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/histogramStat/histogramStat.h>
```

Functions

- void `mvispHistogramStat` (`UInt8 *inputR`, `UInt8 *inputG`, `UInt8 *inputB`, `UInt32 *histR`, `UInt32 *histG`, `UInt32 *histB`, `UInt32 width`, `UInt32 step`)

- void `svuHistogramStat` (`SippFilter` *fptr)
Shave function of the *histogramStat* filter.

7.428.1 Function Documentation

void `mvispHistogramStat` (`UInt8` * inputR, `UInt8` * inputG, `UInt8` * inputB, `UInt32` * histR, `UInt32` * histG, `UInt32` * histB, `UInt32` width, `UInt32` step)

HistogramStats kernel

Parameters

in	<i>inputR</i>	- pointer to the R component
in	<i>inputG</i>	- pointer to the G component
in	<i>inputB</i>	- pointer to the B component
	<i>Out]</i>	histR - pointer to histogram for R component
	<i>Out]</i>	histG - pointer to histogram for G component
	<i>Out]</i>	histB - pointer to histogram for B component
in	<i>width</i>	- width of the input line
in	<i>step</i>	- step*8 will be the number of skipped pixels

Returns

Nothing

Referenced by `svuHistogramStat()`.

7.429 svuHomography.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/homography/homography.h>
```

Functions

- `UInt32` `getIntPtrAbs` (`SippFilter` *fptr, `UInt32` parNo, `UInt32` lineNo, `UInt32` planeNo, `UInt32` targetSlice)
- void `svuHomography` (`SippFilter` *fptr)
Shave function of the *Homography* filter.

7.429.1 Function Documentation

`UInt32` `getIntPtrAbs` (`SippFilter` * fptr, `UInt32` parNo, `UInt32` lineNo, `UInt32` planeNo, `UInt32` targetSlice)

7.430 svuIntegralImageSqSumF32M2.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/integralImageSqSumF32M2/integralImageSqSumF32M2.h>
```

Macros

- #define MAX_NR_PLANE_ON_INTEGRAL_IMAGE 8

Functions

- void integralimage_sqsum_f32_M2 (float *out, UInt8 *in, UInt32 runNr, float **prevPointer, UInt32 width)
- void svuIntegralImageSqSumF32M2 (SippFilter *fptr)
Shave function of the Integral Image Square Sum(f32) filter.

7.430.1 Macro Definition Documentation

```
#define MAX_NR_PLANE_ON_INTEGRAL_IMAGE 8
```

7.430.2 Function Documentation

```
void integralimage_sqsum_f32_M2 ( float * out, UInt8 * in, UInt32 runNr, float ** prevPointer,
UInt32 width )
```

integral image kernel - this kernel makes the sum of all pixels before it and on the left of it's column (this particular case makes square sum of pixels in f32 format)

Parameters

in	in	- array of pointers to input lines
out		- array of pointers for output lines
in	sum	- sum of previous pixels . for this parameter we must have an array of u32 declared as global and having the width of the line
in	width	- width of input line

Referenced by svuIntegralImageSqSumF32M2().

7.431 svuIntegralImageSqSumU32M2.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/integralImageSqSumU32M2/integralImageSqSumU32M2.h>
```

Macros

- #define MAX_NR_PLANE_ON_INTEGRAL_IMAGE 8

Functions

- void `integralimage_sqsum_u32M2` (`UInt32` *out, `UInt8` *in, `UInt32` runNr, `UInt32` **prevPointer, `UInt32` width)
- void `svuIntegralImageSqSumU32M2` (`SippFilter` *fptr)
Shave function of the `Integral Image Square Sum(U32)` filter.

7.431.1 Macro Definition Documentation

```
#define MAX_NR_PLANE_ON_INTEGRAL_IMAGE 8
```

7.431.2 Function Documentation

```
void integralimage_sqsum_u32M2 ( UInt32 * out, UInt8 * in, UInt32 runNr, UInt32 ** prevPointer,
UInt32 width )
```

integral image kernel - this kernel makes the sum of all pixels before it and on the left of it's column (this particular case makes square sum of pixels in u32 format)

Parameters

in	<i>in</i>	- array of pointers to input lines
out		- array of pointers for output lines
in	<i>sum</i>	- sum of previous pixels . for this parameter we must have an array of u32 declared as global and having the width of the line
in	<i>width</i>	- width of input line

Referenced by `svuIntegralImageSqSumU32M2()`.

7.432 svuIntegralImageSumF32M2.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/integralImageSumF32M2/integralImageSumF32M2.h>
```

Macros

- #define `MAX_NR_PLANE_ON_INTEGRAL_IMAGE` 8

Functions

- void `integralimage_sum_f32M2` (`float` *out, `UInt8` *in, `UInt32` runNr, `float` **prevPointer, `UInt32` width)
- void `svuIntegralImageSumF32M2` (`SippFilter` *fptr)
Shave function of the `Integral Image Sum(f32)` filter.

7.432.1 Macro Definition Documentation

```
#define MAX_NR_PLANE_ON_INTEGRAL_IMAGE 8
```

7.432.2 Function Documentation

```
void integralimage_sum_f32M2 ( float * out, UInt8 * in, UInt32 runNr, float ** prevPointer, UInt32 width )
```

integral image kernel - this kernel makes the sum of all pixels before it and on the left of it's column (this particular case makes sum of pixels in f32 format)

Parameters

in	<i>in</i>	- array of pointers to input lines
out		- array of pointers for output lines
in	<i>sum</i>	- sum of previous pixels . for this parameter we must have an array of u32 declared as global and having the width of the line
in	<i>width</i>	- width of input line

Referenced by svuIntegralImageSumF32M2().

7.433 svuIntegralImageSumU16U32.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/integralImageSumU16U32/integralImageSumU16U32.h>
```

Macros

- #define [MAX_NR_PLANE_ON_INTEGRAL_IMAGE](#) 8

Functions

- void [integralImageSumU16U32](#) (UInt32 *out, UInt16 *in, UInt32 runNr, UInt32 **previsionOutLnPointer, UInt32 width)
- void [svuIntegralImageSumU16U32](#) (SippFilter *fptr)
Shave function of the [Integral Image Sum\(U16toU32\)](#) filter.

7.433.1 Macro Definition Documentation

```
#define MAX_NR_PLANE_ON_INTEGRAL_IMAGE 8
```

7.433.2 Function Documentation

```
void integralImageSumU16U32 ( UInt32 * out, UInt16 * in, UInt32 runNr, UInt32 **  
previsionOutLnPointer, UInt32 width )
```

integral image kernel - this kernel makes the sum of all pixels before it and on the left of it's column (this particular case makes sum of pixels in u32 format)

Parameters

out	<i>out</i>	- array of pointers for output lines U32 format
in	<i>in</i>	- array of pointers to input lines U16 data format
	<i>runNr/in</i>	- output line number
in	<i>sum</i>	- pointer to prevision generated line, will be used for calculate current pixels, and will be updated at the end to point to new output generated line
in	<i>width</i>	- width of input line

Referenced by `svuIntegralImageSumU16U32()`.

7.434 svuIntegralImageSumU32M2.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/integralImageSumU32M2/integralImageSumU32M2.h>
```

Macros

- #define `MAX_NR_PLANE_ON_INTEGRAL_IMAGE` 8

Functions

- void `integralimage_sum_u32M2` (`UInt32` *out, `UInt8` *in, `UInt32` runNr, `UInt32` **prevPointer, `UInt32` width)
- void `svuIntegralImageSumU32M2` (`SippFilter` *fptr)
Shave function of the `Integral Image Sum(U32)` filter.

7.434.1 Macro Definition Documentation

```
#define MAX_NR_PLANE_ON_INTEGRAL_IMAGE 8
```

7.434.2 Function Documentation

```
void integralimage_sum_u32M2 ( UInt32 * out, UInt8 * in, UInt32 runNr, UInt32 ** prevPointer,
UInt32 width )
```

integral image kernel - this kernel makes the sum of all pixels before it and on the left of it's column (this particular case makes sum of pixels in u32 format)

Parameters

in	<i>in</i>	- array of pointers to input lines
out		- array of pointers for output lines

in	<i>sum</i>	- sum of previous pixels . for this parameter we must have an array of u32 declared as global and having the width of the line
in	<i>width</i>	- width of input line

Referenced by `svuIntegralImageSumU32M2()`.

7.435 `svuInterpolatePixelBilinear.c` File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/interpolatePixelBilinear/interpolatePixelBilinear.-
h>
```

Functions

- half `mvcvInterpolatePixelBilinear` (half *line1, half *line2, float x, float y)
- void `svuInterpolatePixelBilinear` (SippFilter *fptr)
Shave function of the `interpolatePixelBilinear` filter.

7.435.1 Function Documentation

`half mvcvInterpolatePixelBilinear (half * line1, half * line2, float x, float y)`

Bilinear interpolation of four pixels

Parameters

in	<i>Line1</i>	- First input line
in	<i>Line2</i>	- Second input line
in	<i>x</i>	- The x coordinate of the pixel.
in	<i>y</i>	- The y coordinate of the pixel. Because we work with two lines only, only the fractional part of the number matters.

Returns

The value of the interpolated pixel.

Referenced by `svuInterpolatePixelBilinear()`.

7.436 `svuLaplacian3x3.c` File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/laplacian3x3/laplacian3x3.h>
```


Functions

- void [Laplacian3x3Implementation](#) (UInt8 **in, UInt8 **out, UInt32 inWidth)
- void [svuLaplacian3x3](#) (SippFilter *fptr)

Shave function of the [Laplacian 3x3](#) filter.

7.436.1 Function Documentation

[void Laplacian3x3Implementation](#) (UInt8 ** in, UInt8 ** out, UInt32 inWidth)

Laplacian filter - applies a Laplacian filter with custom size (see http://en.wikipedia.org/wiki/Discrete_Laplace_operator)

Parameters

in	<i>in</i>	- array of pointers to input lines
out	<i>out</i>	- pointer to output line
in	<i>width</i>	- width of input line

Referenced by [svuLaplacian3x3\(\)](#).

7.437 svuLaplacian5x5.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/laplacian5x5/laplacian5x5.h>
```

Functions

- void [sLaplacian5x5Implementation](#) (UInt8 **in, UInt8 **out, UInt32 inWidth)
- void [svuLaplacian5x5](#) (SippFilter *fptr)

Shave function of the [Laplacian 5x5](#) filter.

7.437.1 Function Documentation

[void sLaplacian5x5Implementation](#) (UInt8 ** in, UInt8 ** out, UInt32 inWidth)

Laplacian filter - applies a Laplacian filter with custom size (see http://en.wikipedia.org/wiki/Discrete_Laplace_operator)

Parameters

in	<i>in</i>	- array of pointers to input lines
out	<i>out</i>	- pointer to output line
in	<i>width</i>	- width of input line

Referenced by [svuLaplacian5x5\(\)](#).

7.438 svuLaplacian5x5Fp16ToFp16.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/laplacian5x5Fp16ToFp16/laplacian5x5Fp16ToFp16.h>
```

Functions

- void [sLaplacian5x5Fp16ToFp16Implementation](#) (half **in, half **out, [UInt32](#) inWidth)
- void [svuLaplacian5x5Fp16ToFp16](#) ([SippFilter](#) *fptr)
Shave function of the [Laplacian 5x5 Fp16 To Fp16](#) filter.

7.438.1 Function Documentation

[void sLaplacian5x5Fp16ToFp16Implementation](#) (half ** in, half ** out, [UInt32](#) inWidth)

Laplacian filter - applies a Laplacian filter with custom size (see http://en.wikipedia.org/wiki/Discrete_Laplace_operator)

Parameters

in	<i>in</i>	- array of pointers to input lines
out	<i>out</i>	- pointer to output line
in	<i>width</i>	- width of input line

Referenced by [svuLaplacian5x5Fp16ToFp16\(\)](#).

7.439 svuLaplacian7x7.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/laplacian7x7/laplacian7x7.h>
```

Functions

- void [sLaplacian7x7Implementation](#) ([UInt8](#) **in, [UInt8](#) **out, [UInt32](#) inWidth)
- void [svuLaplacian7x7](#) ([SippFilter](#) *fptr)
Shave function of the [Laplacian 7x7](#) filter.

7.439.1 Function Documentation

[void sLaplacian7x7Implementation](#) ([UInt8](#) ** in, [UInt8](#) ** out, [UInt32](#) inWidth)

Laplacian filter - applies a Laplacian filter with custom size (see http://en.wikipedia.org/wiki/Discrete_Laplace_operator)

Parameters

in	<i>in</i>	- array of pointers to input lines
out	<i>out</i>	- pointer to output line
in	<i>width</i>	- width of input line

Referenced by `svuLaplacian7x7()`.

7.440 `svuLaplacian7x7Fp16ToFp16.c` File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/laplacian7x7Fp16ToFp16/laplacian7x7Fp16ToFp16.h>
```

Functions

- void `sLaplacian7x7Fp16ToFp16Implementation` (half **in, half **out, `UInt32` inWidth)
- void `svuLaplacian7x7Fp16ToFp16` (`SippFilter` *fptr)
Shave function of the `Laplacian 7x7 Fp16 To Fp16` filter.

7.440.1 Function Documentation

`void sLaplacian7x7Fp16ToFp16Implementation (half ** in, half ** out, UInt32 inWidth)`

Laplacian filter - applies a Laplacian filter with custom size (see http://en.wikipedia.org/wiki/Discrete_Laplace_operator)

Parameters

in	<i>in</i>	- array of pointers to input lines
out	<i>out</i>	- pointer to output line
in	<i>width</i>	- width of input line

Referenced by `svuLaplacian7x7Fp16ToFp16()`.

7.441 `svuLocalMaxMin3x3_fp16.c` File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/localMaxMin3x3_fp16/localMaxMin3x3_fp16.h>
```

Functions

- void `mvcvLocalMaxMin3x3_fp16` (half **inBuffer, `UInt32` width, `UInt32` minLocationList[], `UInt32` maxLocationList[], `UInt32` *minCount, `UInt32` *maxCount)
- void `svuLocalMaxMin3x3_fp16` (`SippFilter` *fptr)
Shave function of the `localMaxMin3x3_fp16` filter.

7.441.1 Function Documentation

```
void mvcvLocalMaxMin3x3_fp16 ( half ** inBuffer, UInt32 width, UInt32 minLocationList[],
    UInt32 maxLocationList[], UInt32 * minCount, UInt32 * maxCount )
```

This kernel will find the points which are minimums or maximums in their 3x3 zone. The points from the middle line are compared to their neighbors

Parameters

in	<i>inBuffer</i>	- Address of the fp16 image buffer. This buffer should have a size of 3 * width * sizeof(fp16) bytes (3 lines)
in	<i>width</i>	- Line width in pixels as UInt32
out	<i>minLocation-List</i>	- UInt32 buffer, where the function will save the X coordinates of the minimums found
out	<i>maxLocation-List</i>	- UInt32 buffer, where the function will save the X coordinates of the maximums found
out	<i>minCount</i>	- The function will save the number of minimums found to this address
out	<i>maxCount</i>	- The function will save the number of maximums found to this address

The value of the pixel which is checked.

The candidate is a minimum, add the X coordinate to the list

The candidate is a maximum, add the X coordinate to the list

Referenced by svuLocalMaxMin3x3_fp16().

7.442 svuLocalTM.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/localTM/localTM.h>
```

Functions

- void **localTM** (half **luma_in, UInt8 **bg8, half **output, half curves[160], UInt32 width, UInt32 run_no)
- void **svuLocalTM** (SippFilter *fptr)
Shave function of the *localTM* filter.

7.442.1 Function Documentation

```
void localTM ( half ** luma_in, UInt8 ** bg8, half ** output, half curves[160], UInt32 width,
    UInt32 run_no )
```

Referenced by svuLocalTM().

7.443 svuLowLvlCorr.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/lowLvlCorr/lowLvlCorr.h>
```

Macros

- #define [SUB_SATURATE](#)(x, y) (x < y ? 0 : (x - y))

Functions

- void [svuLowLvlCorr](#) ([SippFilter](#) *fptr)
Shave function of the [Low Level Correction on Multiple Planes](#) filter.

7.443.1 Macro Definition Documentation

```
#define SUB_SATURATE( x, y ) (x < y ? 0 : (x - y))
```

Referenced by [svuLowLvlCorr\(\)](#).

7.444 svuLumaBlur.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
```

Functions

- void [lumaBlur](#) ([UInt8](#) *inY0, [UInt8](#) *inY1, [UInt8](#) *inY2, [UInt8](#) *outY, int width)
- void [svuLumaBlur](#) ([SippFilter](#) *fptr)
Shave function of the [Luma Blur](#) filter.

7.444.1 Function Documentation

```
void lumaBlur ( UInt8 * inY0, UInt8 * inY1, UInt8 * inY2, UInt8 * outY, int width )
```

Referenced by [svuLumaBlur\(\)](#).

7.445 svuLut10to16.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/lut10to16/lut10to16.h>
```

Functions

- void `LUT10to16` (`UInt16 **src`, `UInt16 **dest`, const `UInt16 *lut`, `UInt32 width`, `UInt32 height`)
- void `svuLut10to16` (`SippFilter *fptr`)

Shave function of the `LUT 10 to 16` filter.

7.445.1 Function Documentation

`void LUT10to16 (UInt16 ** src, UInt16 ** dest, const UInt16 * lut, UInt32 width, UInt32 height)`

Performs a look-up table transform of a line. The function fills the destination line with values from the look-up table. Indices of the entries are taken from the source line

Parameters

in	<i>src</i>	- Pointer to input line
out	<i>dest</i>	- Pointer to output line
in	<i>lut</i>	- Look-up table of 256 elements; should have the same depth as the input line. In the case of multi-channel source and destination lines, the table should either have a single-channel (in this case the same table is used for all channels) or the same number of channels as the source/destination line.
in	<i>width</i>	- width of input line
in	<i>height</i>	- the number of lines (defaulted to one line)

Referenced by `svuLut10to16()`.

7.446 svuLut10to8.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/lut10to8/lut10to8.h>
```

Functions

- void `LUT10to8` (`UInt16 **src`, `UInt8 **dest`, const `UInt8 *lut`, `UInt32 width`, `UInt32 height`)
- void `svuLut10to8` (`SippFilter *fptr`)

Shave function of the `LUT 10 to 8` filter.

7.446.1 Function Documentation

`void LUT10to8 (UInt16 ** src, UInt8 ** dest, const UInt8 * lut, UInt32 width, UInt32 height)`

Performs a look-up table transform of a line. The function fills the destination line with values from the look-up table. Indices of the entries are taken from the source line

Parameters

in	src	- Pointer to input line
out	dest	- Pointer to output line
in	lut	- Look-up table of 256 elements; should have the same depth as the input line. In the case of multi-channel source and destination lines, the table should either have a single-channel (in this case the same table is used for all channels) or the same number of channels as the source/destination line.
in	width	- width of input line
in	height	- the number of lines (defaulted to one line)

Referenced by svuLut10to8().

7.447 svuLut12to16.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/lut12to16/lut12to16.h>
```

Functions

- void [LUT12to16](#) ([UInt16](#) **src, [UInt16](#) **dest, const [UInt16](#) *lut, [UInt32](#) width, [UInt32](#) height)
- void [svuLut12to16](#) ([SippFilter](#) *fptr)

Shave function of the [LUT 12 to 16](#) filter.

7.447.1 Function Documentation

```
void LUT12to16 ( UInt16 ** src, UInt16 ** dest, const UInt16 * lut, UInt32 width, UInt32 height
)
```

Performs a look-up table transform of a line. The function fills the destination line with values from the look-up table. Indices of the entries are taken from the source line

Parameters

in	src	- Pointer to input line
out	dest	- Pointer to output line
in	lut	- Look-up table of 256 elements; should have the same depth as the input line. In the case of multi-channel source and destination lines, the table should either have a single-channel (in this case the same table is used for all channels) or the same number of channels as the source/destination line.
in	width	- width of input line
in	height	- the number of lines (defaulted to one line)

Referenced by svuLut12to16().

7.448 svuLut12to8.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/lut12to8/lut12to8.h>
```

Functions

- void [LUT12to8](#) ([UInt16](#) **src, [UInt8](#) **dest, const [UInt8](#) *lut, [UInt32](#) width, [UInt32](#) height)
- void [svuLut12to8](#) ([SippFilter](#) *fptr)

Shave function of the [LUT 12 to 8](#) filter.

7.448.1 Function Documentation

void [LUT12to8](#) ([UInt16](#) ** src, [UInt8](#) ** dest, const [UInt8](#) * lut, [UInt32](#) width, [UInt32](#) height)

Performs a look-up table transform of a line. The function fills the destination line with values from the look-up table. Indices of the entries are taken from the source line

Parameters

in	<i>src</i>	- Pointer to input line
out	<i>dest</i>	- Pointer to output line
in	<i>lut</i>	- Look-up table of 256 elements; should have the same depth as the input line. In the case of multi-channel source and destination lines, the table should either have a single-channel (in this case the same table is used for all channels) or the same number of channels as the source/destination line.
in	<i>width</i>	- width of input line
in	<i>height</i>	- the number of lines (defaulted to one line)

Referenced by [svuLut12to8\(\)](#).

7.449 svuLut8to8.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/lut8to8/lut8to8.h>
```

Functions

- void [LUT8to8](#) ([UInt8](#) **src, [UInt8](#) **dest, const [UInt8](#) *lut, [UInt32](#) width, [UInt32](#) height)
- void [svuLut8to8](#) ([SippFilter](#) *fptr)

Shave function of the [LUT 8 to 8](#) filter.

7.449.1 Function Documentation

```
void LUT8to8 ( UInt8 ** src, UInt8 ** dest, const UInt8 * lut, UInt32 width, UInt32 height )
```

Performs a look-up table transform of a line. The function fills the destination line with values from the look-up table. Indices of the entries are taken from the source line

Parameters

in	src	- Pointer to input line
out	dest	- Pointer to output line
in	lut	- Look-up table of 256 elements; should have the same depth as the input line. In the case of multi-channel source and destination lines, the table should either have a single-channel (in this case the same table is used for all channels) or the same number of channels as the source/destination line.
in	width	- width of input line
in	height	- the number of lines (defaulted to one line)

Referenced by `svuLut8to8()`.

7.450 svuLutP10BppU16inU8out.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/lutP10BppU16inU8out/lutP10BppU16inU8out.h>
```

Macros

- #define `MAX_LUT_IDX` $((1 < 10) - 1)$

Functions

- void `svuLutP10BppU16inU8out` (`SippFilter` *fptr)
Shave function of the *LUT 16 to 8* filter.

7.450.1 Macro Definition Documentation

```
#define MAX_LUT_IDX ((1 < 10) - 1)
```

Referenced by `svuLutP10BppU16inU8out()`.

7.451 svuMaxTest3x3_fp16.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/maxTest3x3_fp16/maxTest3x3_fp16.h>
```

Functions

- void `mvcvMaxTest3x3_fp16` (half *inBufferCandidates, half **inBuffer, `UInt32` width, `UInt32` maxLocationsIn[], `UInt32` maxLocationsOut[], `UInt32` maxCountIn, `UInt32` *maxCountOut)

- void `svuMaxTest3x3_fp16` (`SippFilter *fptr`)
Shave function of the `maxTest3x3_fp16` filter.

7.451.1 Function Documentation

void `mvcvMaxTest3x3_fp16` (half * `inBufferCandidates`, half ** `inBuffer`, `UInt32` `width`, `UInt32` `maxLocationsIn`[], `UInt32` `maxLocationsOut`[], `UInt32` `maxCountIn`, `UInt32` * `maxCountOut`)

This kernel will compare the points from `inBufferCandidates` to the corresponding 3x3 zone of `inBuffer`. The function will check `maxCountIn` locations.

Parameters

in	<i>inBuffer-Candidates</i>	- Fp16 buffer, where the candidates can be found
in	<i>inBuffer</i>	- Address of the fp16 image buffer. This buffer should have a size of 3 * width * sizeof(fp16) bytes (3 lines)
in	<i>width</i>	- Line width in pixels as u32
in	<i>maxLocations-In</i>	- Gives the x coordinates of the candidates. Only these candidates are checked.
out	<i>maxLocations-Out</i>	- The values from <code>maxLocationsIn</code> which passed the filter.
in	<i>maxCountIn</i>	- Number of values in <code>maxLocationsIn</code> buffer
out	<i>maxCountOut</i>	- Number of values in <code>maxLocationsOut</code> buffer

The candidate is a maximum, add the X coordinate to the list

Referenced by `svuMaxTest3x3_fp16()`.

7.452 svuMeanStdDev.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/meanStdDev/meanStdDev.h>
#include <math.h>
```

Functions

- void `mvcvMeanstddev` (`UInt8 **in`, float * `mean`, float * `stddev`, `u32` `width`)
- void `svuMeanStdDev` (`SippFilter *fptr`)
Parameter structure of the `meanStdDev` filter.

7.452.1 Function Documentation

void `mvcvMeanstddev` (`UInt8 **in`, float * `mean`, float * `stddev`, `u32` `width`)

This kernel calculates mean and standard deviation of an array of elements

Parameters

in	<i>in</i>	- Input line
out	<i>mean</i>	- Computed mean value
out	<i>stddev</i>	- Computed standard deviation
out	<i>width</i>	- Width of line

Referenced by `svuMeanStdDev()`.

7.453 `svuMinMaxPos.c` File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/minMaxPos/minMaxPos.h>
```

Functions

- void `minMaxPos` (`UInt8 **in`, `UInt32 width`, `UInt8 *minVal`, `UInt8 *maxVal`, `UInt32 *minPos`, `UInt32 *maxPos`, `UInt8 *maskAddr`)
- void `svuMinMaxPos` (`SippFilter *fptr`)

Shave function of the [Min/Max Value Position](#) filter.

7.453.1 Function Documentation

`void minMaxPos (UInt8 ** in, UInt32 width, UInt8 * minVal, UInt8 * maxVal, UInt32 * minPos, UInt32 * maxPos, UInt8 * maskAddr)`

`minMaxPos` kernel - computes the minimum and the maximum value of a given input line and their position

Parameters

in	<i>in</i>	- input line
in	<i>width</i>	- line's width(length)
in	<i>height</i>	- height of image (defaulted to one line)
in	<i>minVal</i>	- stores the minimum value on the line
in	<i>maxVal</i>	- stores the maximum value on the line
out	<i>minPos</i>	- stores the position occupied by the MIN value within line
out	<i>maxPos</i>	- stores the position occupied by the MAX value within line
in	<i>maskAddr</i>	- mask filled with 1s and 0s which determines the image area to compute minimum and maximum

Returns

- Nothing

Referenced by `svuMinMaxPos()`.

7.454 svuMinMaxValue.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/minMaxValue/minMaxValue.h>
```

Functions

- void [minMaxKernel](#) (UInt8 **in, UInt32 width, UInt32 height, UInt8 *minVal, UInt8 *maxVal, UInt8 *maskAddr)
- void [svuMinMaxValue](#) (SippFilter *fptr)
Shave function of the [Min/Max Value](#) filter.

7.454.1 Function Documentation

```
void minMaxKernel ( UInt8 ** in, UInt32 width, UInt32 height, UInt8 * minVal, UInt8 * maxVal,
UInt8 * maskAddr )
```

minMax kernel - computes the minimum and the maximum value of a given input image

Parameters

in	<i>in</i>	- array of pointers to input lines
in	<i>width</i>	- line's width(length)
in	<i>height</i>	- height of image (defaulted to one line)
in	<i>minVal</i>	- stores the minimum value on the line
in	<i>maxVal</i>	- stores the maximum value on the line
in	<i>maskAddr</i>	- mask filled with 1s and 0s which determines the image area to compute minimum and maximum

Returns

- Nothing

Referenced by [svuMinMaxValue\(\)](#).

7.455 svuMinTest3x3_fp16.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/minTest3x3_fp16/minTest3x3_fp16.h>
```

Functions

- void [mvcvMinTest3x3_fp16](#) (half *inBufferCandidates, half **inBuffer, UInt32 width, UInt32 minLocationsIn[], UInt32 minLocationsOut[], UInt32 minCountIn, UInt32 *minCountOut)
- void [svuMinTest3x3_fp16](#) (SippFilter *fptr)
Shave function of the [minTest3x3_fp16](#) filter.

7.455.1 Function Documentation

```
void mvcvMinTest3x3_fp16 ( half * inBufferCandidates, half ** inBuffer, UInt32 width, UInt32
minLocationsIn[], UInt32 minLocationsOut[], UInt32 minCountIn, UInt32 * minCountOut )
```

This function will compare the points from inBufferCandidates to the corresponding 3x3 zone of in-Buffer. The function will check minCountIn locations.

Parameters

in	<i>inBuffer-Candidates</i>	- fp16 buffer, where the candidates can be found
in	<i>inBuffer</i>	- address of the fp16 image buffer. This buffer should have a size of 3 * width * sizeof(fp16) bytes (3 lines)
in	<i>width</i>	- Line width in pixels as u32
in	<i>minLocations-In</i>	- gives the x coordinates of the candidates. Only these candidates are checked.
out	<i>minLocations-Out</i>	- the values from minLocationsIn which passed the filter.
in	<i>minCountIn</i>	- number of values in minLocationsIn buffer
out	<i>minCountOut</i>	- number of values in minLocationsOut buffer

The candidate is a minimum, add the X coordinate to the list

Referenced by svuMinTest3x3_fp16().

7.456 svuMixMedian.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/mixMedian/mixMedian.h>
```

Functions

- void [mixMedian](#) (UInt8 *out[3], UInt8 *in0[3], UInt8 *in1[3], UInt8 *ref, float offset, float slope, unsigned int width)
- void [svuMixMedian](#) (SippFilter *fptr)

Shave function of the [Mix Median](#) filter.

7.456.1 Function Documentation

```
void mixMedian ( UInt8 * out[3], UInt8 * in0[3], UInt8 * in1[3], UInt8 * ref, float offset, float
slope, unsigned int width )
```

mix median - Mix two 8-bit planes according to a third 8-bit reference plane

Parameters

out	-	first unsigned int contain number of valid points in the line, after score for every point
in	<i>in0</i>	- array of pointers to input line of the image, 3 planes
in	<i>in1</i>	- array of pointers to input line of the image, 3 planes
in	<i>ref</i>	- array of pointers to input line of the image, 1 planes, reference
in	<i>offset</i>	- offset
in	<i>slope</i>	- slope
in	<i>width</i>	- width of the input lines

Returns

Nothing

Referenced by `svuMixMedian()`.

7.457 svuMonoImbalance.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/monoImbalance/monoImbalance.h>
```

Functions

- void `mvcvMonoImbalance` (`UInt16 **in`, `half **out`, `int thrDark`, `int thrBright`, `int inputBits`, `UInt32 inWidth`)
- void `svuMonoImbalance` (`SippFilter *fptr`)
Shave function of the `MonoImbalance` filter.

7.457.1 Function Documentation

```
void mvcvMonoImbalance ( UInt16 ** in, half ** out, int thrDark, int thrBright, int inputBits,
UInt32 inWidth )
```

Referenced by `svuMonoImbalance()`.

```
void svuMonoImbalance ( SippFilter * fptr )
```

Shave function of the `MonoImbalance` filter.

7.458 svuNegative.c File Reference

main leon file

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/negative/negative.h>
```

Functions

- void [negativeFilterImplementation](#) (int *inLine, [UInt8](#) *outLine, int widthLine)
- void [svuNegative](#) ([SippFilter](#) *fptr)
Shave function of the [Negative](#) filter.

7.458.1 Detailed Description

main leon file

Copyright

All code copyright Movidius Ltd 2012, all rights reserved. For License Warranty see: common/license.txt

7.458.2 Function Documentation

void [negativeFilterImplementation](#) (int * inLine, [UInt8](#) * outLine, int widthLine)

Referenced by [svuNegative\(\)](#).

7.459 svuNonMax3x3Fp32.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/nonMax3x3Fp32/nonMax3x3Fp32.h>
```

Functions

- void [mvcvNonMax3x3_fp32](#) ([UInt16](#) *candPos, [UInt32](#) candCount, float *corners1, float *corners2, float *corners3, [UInt16](#) *candOutPos, [UInt32](#) *candOutCount)
- void [svuNonMax3x3Fp32](#) ([SippFilter](#) *fptr)
Shave function of the [RGB to Luma conversion](#) filter.

7.459.1 Function Documentation

void [mvcvNonMax3x3_fp32](#) ([UInt16](#) * candPos, [UInt32](#) candCount, float * corners1, float * corners2, float * corners3, [UInt16](#) * candOutPos, [UInt32](#) * candOutCount)

This kernel verifies whether each element from the central line is the maximum within a 3x3 range

Parameters

<code>in</code>	<code>candPos</code>	- X coordinate of the candidates
-----------------	----------------------	----------------------------------

in	<i>candCount</i>	- Number of candidates
in	<i>corners1</i>	- Input line (N-1 lines)
in	<i>corners2</i>	- Input line (N lines)
in	<i>corners3</i>	- Input line (N+1 lines)
out	<i>candOutPos</i>	- X coordinate of selected elements
out	<i>candOutCount</i>	- Output elements

Referenced by `svuNonMax3x3Fp32()`.

7.460 `svuNonMax3x3U8.c` File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/nonMax3x3U8/nonMax3x3U8.h>
```

Functions

- void `mvcvNonMax3x3_u8` (`UInt32` width, `UInt8` *corners1, `UInt8` *corners2, `UInt8` *corners3, `UInt16` *candOutPos, `UInt32` *candOutCount)
- void `svuNonMax3x3U8` (`SippFilter` *fptr)
Shave function of the `nonMax3x3U8` filter.

7.460.1 Function Documentation

```
void mvcvNonMax3x3_u8 ( UInt32 width, UInt8 * corners1, UInt8 * corners2, UInt8 * corners3,
UInt16 * candOutPos, UInt32 * candOutCount )
```

This kernel verifies whether each element from the central line is the maximum within a 3x3 range

Parameters

in	<i>width</i>	- line width
in	<i>corners1</i>	- input N-1 line
in	<i>corners2</i>	- input N line
in	<i>corners3</i>	- input N+1 line
out	<i>candOutPos</i>	- X coordinate of output elements
out	<i>candOutCount</i>	- number of output elements

Referenced by `svuNonMax3x3U8()`.

7.461 `svuPadBayer5.c` File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/padBayer5/padBayer5.h>
```

Functions

- void [padBayer5Reference](#) (uint8_t *inLine[13], uint8_t *outLine, uint32_t exeNo, uint32_t sliceWidth, uint32_t fullW, uint32_t outH, uint32_t svuF, uint32_t svuL)
- void [svuPadBayer5](#) ([SippFilter](#) *fptr)

7.461.1 Function Documentation

void padBayer5Reference (uint8_t * inLine[13], uint8_t * outLine, uint32_t exeNo, uint32_t sliceWidth, uint32_t fullW, uint32_t outH, uint32_t svuF, uint32_t svuL)

Referenced by svuPadBayer5(), and svuPadBayer5Frame().

void svuPadBayer5 ([SippFilter](#) * fptr)

7.462 svuPadBayer5Frame.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/padBayer5Frame/padBayer5Frame.h>
```

Functions

- void [padBayer5Reference](#) (uint16_t *inLine[13], uint16_t *outLine, uint32_t exeNo, uint32_t sliceWidth, uint32_t fullW, uint32_t outH, uint32_t svuF, uint32_t svuL)
- void [svuPadBayer5Frame](#) ([SippFilter](#) *fptr)
Shave function of the [Threshold](#) filter.

7.462.1 Function Documentation

void padBayer5Reference (uint16_t * inLine[13], uint16_t * outLine, uint32_t exeNo, uint32_t sliceWidth, uint32_t fullW, uint32_t outH, uint32_t svuF, uint32_t svuL)

7.463 svuPixelPacker10b.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/pixelPacker10b/pixelPacker10b.h>
```

Functions

- void [pixelPacker10b](#) (UInt16 *srcAddr, UInt32 *ls8b, UInt8 *ms2b, UInt32 width)
- void [svuPixelPacker10b](#) ([SippFilter](#) *fptr)
Shave function of the [Pixel packer](#) filter.

7.463.1 Function Documentation

```
void pixelPacker10b ( UInt16 * srcAddr, UInt32 * ls8b, UInt8 * ms2b, UInt32 width )
```

pixelPacker 16b -> 10b

Parameters

in	<i>srcAddr</i>	- array of pointers to input line
out	<i>ls8b</i>	- array of pointers for output line of ls8b
out	<i>ms2b</i>	- array of pointers for output line of ms2b
in	<i>width</i>	- width of input line

Referenced by svuPixelPacker10b().

7.464 svuPixelUnpacker.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/pixelUnpacker/pixelUnpacker.h>
```

Functions

- void [pixelUnpacker](#) (UInt32 *ls8b, UInt8 *ms2b, UInt16 *output, UInt32 width, UInt8 shift)
- void [svuPixelUnpacker](#) (SippFilter *fptr)
Shave function of the [Pixel Unpacker](#) filter.

7.464.1 Function Documentation

```
void pixelUnpacker ( UInt32 * ls8b, UInt8 * ms2b, UInt16 * output, UInt32 width, UInt8 shift )
```

pixelUnpacker 10b -> 16b/15b/14b/13b/12b/11b/10b depending on the shift value

Parameters

in	<i>ms8b</i>	- array of pointers to the 8 MSBs
in	<i>ms2b</i>	- array of pointers to the 2 LSBs
out	<i>output</i>	- array of pointers for output line
in	<i>width</i>	- width of input line
in	<i>shift</i>	- number of bits for shifting the result to left

Referenced by svuPixelUnpacker().

7.465 svuPixelUnpackerMipi10b.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/pixelUnpackerMipi10b/pixelUnpackerMipi10b.h>
```

Functions

- void [pixelUnpackerMipi10b](#) ([UInt32](#) *in, [UInt16](#) *out, [UInt32](#) width, [UInt32](#) lineNo, [UInt32](#) coefs[4], int noMipiRxWorkaround)
- void [svuPixelUnpackerMipi10b](#) ([SippFilter](#) *fptr)

Shave function of the [Pixel Unpacker Mipi 10b](#) filter.

7.465.1 Function Documentation

void [pixelUnpackerMipi10b](#) ([UInt32](#) * in, [UInt16](#) * out, [UInt32](#) width, [UInt32](#) lineNo, [UInt32](#) coefs[4], int noMipiRxWorkaround)

[pixelUnpackerMipi10b](#) input format 32b: xxxxxxxxxxxxa0a1a2a3a4a5a6a7a8a9b0b1b2b3b4b5b6b7b8b9

output format 16b: 000000a0a1a2a3a4a5a6a7a8a9 000000b0b1b2b3b4b5b6b7b8b9

Parameters

in	<i>in</i>	- pointer to input line
out	<i>out</i>	- pointer to output line
in	<i>width</i>	- number of processed pixels
in	<i>lineNo</i>	- line number
in	<i>corfs</i>	- coefficients for black correction

Referenced by [svuPixelUnpackerMipi10b\(\)](#).

7.466 [svuPixelUnpackerWB.c](#) File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/pixelUnpackerWB/pixelUnpackerWB.h>
```

Functions

- void [pixelUnpackerWB](#) ([UInt32](#) **ms8b, [UInt8](#) **ls2b, [UInt16](#) **output, [UInt32](#) width, [UInt8](#) shift, [UInt16](#) *awbCoef, [UInt32](#) line)
- void [svuPixelUnpackerWB](#) ([SippFilter](#) *fptr)

Shave function of the [Pixel Unpacker WB](#) filter.

7.466.1 Function Documentation

void [pixelUnpackerWB](#) ([UInt32](#) ** ms8b, [UInt8](#) ** ls2b, [UInt16](#) ** output, [UInt32](#) width, [UInt8](#) shift, [UInt16](#) * awbCoef, [UInt32](#) line)

[pixelUnpacker 10b](#) -> 16b/15b/14b/13b/12b/11b/10b depending on the shift value

Parameters

in	<i>ms8b</i>	- array of pointers to the 8 MSBs
in	<i>ms2b</i>	- array of pointers to the 2 LSBs
out	<i>output</i>	- array of pointers for output line
in	<i>width</i>	- width of input line
in	<i>shift</i>	- number of bits for shifting the result to left

Unpacker

White balance gains

Referenced by `svuPixelUnpackerWB()`.

7.467 `svuPositionKernel.c` File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/positionKernel/positionKernel.h>
```

Functions

- void `pixelPos` (`UInt8 **srcAddr`, `UInt8 *maskAddr`, `UInt32 width`, `UInt8 pixelValue`, `UInt32 *pixelPosition`, `UInt8 *status`)
- void `svuPositionKernel` (`SippFilter *fptr`)
Shave function of the `Pixel Position` filter.

7.467.1 Function Documentation

void `pixelPos` (`UInt8 ** srcAddr`, `UInt8 * maskAddr`, `UInt32 width`, `UInt8 pixelValue`, `UInt32 * pixelPosition`, `UInt8 * status`)

pixel Position kernel - returns the position of a given pixel value

Parameters

in	<i>srcAddr</i>	- array of pointers to input lines
in	<i>maskAddr</i>	- mask filled with 1s and 0s which determines the image area to find position
in	<i>width</i>	- line's width(length)
in	<i>pixelValue</i>	- stores the pixel value to be searched
out	<i>pixelPosition</i>	- stores the position occupied by the searched value within line
out	<i>status</i>	- stores 0x11 if pixel value found, else 0x00

Returns

- Nothing

Referenced by `svuPositionKernel()`.

7.468 svuPurpleFlare.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/purpleFlare/purpleFlare.h>
```

Macros

- #define **MIN**(a, b) ((a) < (b) ? (a) : (b))
- #define **MAX**(a, b) ((a) > (b) ? (a) : (b))

Functions

- void **purpleFlare** (**UInt16** *inGreen, **UInt16** *blue[3], **UInt16** *outBlue, **UInt8** strength, **UInt32** width)
- void **svuPurpleFlare** (**SippFilter** *fptr)
*Shave function of the **purpleFlare** filter.*

7.468.1 Macro Definition Documentation

```
#define MAX( a, b ) ((a) > (b) ? (a) : (b))
```

Referenced by **purpleFlare()**, **svuLowLvlCorr()**, and **svulowLvlCorrMultiplePlanes()**.

```
#define MIN( a, b ) ((a) < (b) ? (a) : (b))
```

Referenced by **purpleFlare()**, **svuLowLvlCorr()**, and **svulowLvlCorrMultiplePlanes()**.

7.468.2 Function Documentation

```
void purpleFlare ( UInt16 * inGreen, UInt16 * blue[3], UInt16 * outBlue, UInt8 strength, UInt32 width )
```

Referenced by **svuPurpleFlare()**.

7.469 svuPyrDown.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/pyrDown/pyrDown.h>
```

Functions

- void **pyrdown** (**UInt8** **inLine, **UInt8** **out, int width)

- void `svuPyrDown` (`SippFilter *fptr`)
Shave function of the `Pyramid Downscale` filter.

7.469.1 Function Documentation

```
void pyrdown ( UInt8 ** inLine, UInt8 ** out, int width )
```

pyrdown filter - downsample even lines and even cols

Parameters

in	<i>in</i>	- array of pointers to input lines
out		- array of pointers for output lines
in	<i>width</i>	- width of input line

Referenced by `svuPyrDown()`.

7.470 svuRandNoise.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/randNoise/randNoise.h>
```

Macros

- #define `ONE_OVER_UINT32_MAX` (1.0f/4294967295.0f)

Functions

- float `getRandom` (`UInt32 *m_w`, `UInt32 *m_z`)
- void `svuGenNoise` (`SippFilter *fptr`)
Shave function of the `Random Noise` filter.

Variables

- `UInt32 seeds` [12][2]

7.470.1 Macro Definition Documentation

```
#define ONE_OVER_UINT32_MAX (1.0f/4294967295.0f)
```

Referenced by `getRandom()`.

7.470.2 Function Documentation

```
float getRandom ( UInt32 * m_w, UInt32 * m_z )
```

Referenced by `svuGenNoise()`.

7.470.3 Variable Documentation

UInt32 seeds[12][2]

Initial value:

```
=
{
    {0x39f4d2fa, 0x2d9cbf10},
    {0x126eb1d1, 0x32a39600},
    {0x1475525d, 0x80f7f9b5},
    {0x0427cf25, 0x51f28cb3},
    {0x29d6a2bc, 0x2b1bc9cd},
    {0x1a1dbe0a, 0x77b33da8},
    {0x241e7fbb, 0x2e95a583},
    {0x44066b1d, 0x4c5d77df},
    {0x09fee756, 0x035a1244},
    {0x3c3420b5, 0x042e1913},
    {0x334fee6b, 0x89315fc8},
    {0x2ceb7dca, 0x6faf23ad}
}
```

Referenced by svuGenNoise().

7.471 svuRandNoiseFp16.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/randNoiseFp16/randNoiseFp16.h>
```

Functions

- unsigned short **sauOnesX16** (unsigned short in)
- unsigned short **genRand** (unsigned short seed)
- void **randNoiseFp16** (half *output, half *input, float noiseStrength, unsigned int width)
- void **svuGenNoiseFp16** (**SippFilter** *fptr)

*Shave function of the **Random Noise (high speed)** filter.*

Variables

- unsigned short **seedlist** [12 *8]
- const unsigned int **mskBit** = ((1<<15) + (1<<13) + (1<<4) + (1<<0))
- half **normValue** = (half)0.003921509

7.471.1 Function Documentation

unsigned short **genRand** (unsigned short seed)

Referenced by randNoiseFp16().


```
void randNoiseFp16 ( half * output, half * input, float noiseStrength, unsigned int width )
```

Referenced by svuGenNoiseFp16().

```
unsigned short sauOnesX16 ( unsigned short in )
```

Referenced by genRand().

7.471.2 Variable Documentation

```
const unsigned int mskBit = ((1<<15) + (1<<13) + (1<<4) + (1<<0))
```

Referenced by genRand().

```
half normValue = (half)0.003921509
```

Referenced by randNoiseFp16().

```
unsigned short seedlist[12 * 8]
```

Initial value:

```
= {
  48983, 45013, 19197, 10539, 13476, 3369, 31472, 22447,
  30400, 22649, 46471, 44385, 60118, 49067, 18879, 42139,
  23493, 18725, 21065, 19968, 12366, 25348, 52182, 63519,
  25325, 21617, 337, 20501, 18752, 29257, 457, 12295,
  27449, 9644, 50556, 31831, 40725, 53234, 49595, 64496,
  12042, 8516, 29204, 24711, 51586, 31526, 18924, 58697,
  23228, 25686, 44433, 44726, 58042, 22242, 50523, 41777,
  6150, 50968, 5916, 369, 2049, 4224, 55504, 58211,
  46402, 45323, 62350, 41020, 54912, 17622, 55249, 30196,
  9648, 50460, 3153, 17172, 44300, 3994, 52286, 58207,
  4872, 13075, 2892, 45101, 32394, 8084, 32543, 54467,
  8175, 25086, 16030, 36775, 7742, 29155, 41415, 32929
}
```

Referenced by randNoiseFp16().

7.472 svuSAD11x11.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/sad11x11/sad11x11.h>
```

Functions

- void [sumOfAbsDiff11x11](#) (UInt8 **in1, UInt8 **in2, UInt8 **out, UInt32 width)
- void [svuSAD11x11](#) (SippFilter *fptr)

Shave function of the [Sum of Absolute Differences 11x11](#) filter.

7.472.1 Function Documentation

```
void sumOfAbsDiff11x11 ( UInt8 ** in1, UInt8 ** in2, UInt8 ** out, UInt32 width )
```

SAD (sum of absolute differences) 11x11

Parameters

in	<i>in1</i>	- array of pointers to input lines from the first image
in	<i>in2</i>	- array of pointers to input lines from the second image
out	<i>out</i>	- array of pointers for output line
in	<i>width</i>	- width of input line

Referenced by svuSAD11x11().

7.473 svuSAD5x5.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/sad5x5/sad5x5.h>
```

Functions

- void [sumOfAbsDiff5x5](#) (UInt8 **in1, UInt8 **in2, UInt8 **out, UInt32 width)
- void [svuSAD5x5](#) (SippFilter *fptr)

Shave function of the [Sum of Absolute Differences 5x5](#) filter.

7.473.1 Function Documentation

```
void sumOfAbsDiff5x5 ( UInt8 ** in1, UInt8 ** in2, UInt8 ** out, UInt32 width )
```

SAD (sum of absolute differences) 5x5

Parameters

in	<i>in1</i>	- array of pointers to input lines from the first image
in	<i>in2</i>	- array of pointers to input lines from the second image
out	<i>out</i>	- array of pointers for output line
in	<i>width</i>	- width of input line

Referenced by svuSAD5x5().

7.474 svuScale05BilinHV.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
```

Functions

- void [scale05BilinHV_U16ToU16](#) ([UInt16](#) **input, [UInt16](#) *output, int width)
- void [scale05BilinHV_U8ToU8](#) ([UInt8](#) **input, [UInt8](#) *output, int width)
- void [svuScl05BilinHV](#) ([SippFilter](#) *fptr)

Shave function of the [Downscale by 2](#) filter.

7.474.1 Function Documentation

[void scale05BilinHV_U16ToU16](#) ([UInt16](#) ** input, [UInt16](#) * output, int width)

[scale05BilinHV_U16ToU16](#) kernel

Parameters

in	in	- array of pointers to input lines
in	out	- array of pointers to output lines
in	inWidth	- width of input line

Referenced by [svuScl05BilinHV](#)().

[void scale05BilinHV_U8ToU8](#) ([UInt8](#) ** input, [UInt8](#) * output, int width)

[scale05BilinHV_U8ToU8](#) kernel

Parameters

in	in	- array of pointers to input lines
in	out	- array of pointers to output lines
in	inWidth	- width of input line

Referenced by [svuScl05BilinHV](#)().

7.475 [svuScale05BilinHV_Fp16U8.c](#) File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/scale05BilinHV_Fp16U8/scale05BilinHV_Fp16U8.h>
```

Functions

- void [scale05BilinHV_Fp16U8](#) (half **in, [UInt8](#) **out, [UInt32](#) inWidth)
- void [svuScale05BilinHV_Fp16U8](#) ([SippFilter](#) *fptr)

Shave function of the [Downscale by 2 \(fp16/u8\)](#) filter.

7.475.1 Function Documentation

[void scale05BilinHV_Fp16U8](#) (half ** in, [UInt8](#) ** out, [UInt32](#) inWidth)

[scale05BilinHV_fp16U8](#) kernel

Parameters

in	<i>in</i>	- array of pointers to input lines
in	<i>out</i>	- array of pointers to output lines
in	<i>inWidth</i>	- width of input line

Referenced by svuScale05BilinHV_Fp16U8().

7.476 svuScale05BilinHVFp16.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
```

Functions

- void [scale05BilinHVFp16](#) (half **in, half **out, [UInt32](#) width)
- void [svuScale05BilinHVFp16](#) ([SippFilter](#) *fptr)

Shave function of the [Downscale by 2 \(fp16/fp16\)](#) filter.

7.476.1 Function Documentation

```
void scale05BilinHVFp16 ( half ** in, half ** out, UInt32 width )
```

Scale05BilinHVFp16 kernel

Parameters

in	<i>in</i>	- array of pointers to input lines
in	<i>out</i>	- array of pointers to output lines
in	<i>width</i>	- width of input line

Referenced by svuScale05BilinHVFp16().

7.477 svuScale05Lanc6HV.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
```

Functions

- int [vStep](#) ([UInt8](#) *inRGB[6], int pos)
- void [scale05Lanc6HV](#) ([UInt8](#) *in[6], [UInt8](#) *out, unsigned int width)
- void [svuScI05Lanc6](#) ([SippFilter](#) *fptr)

Shave function of the [Lanczos Downscale by 2 \(6 taps\)](#) filter.

7.477.1 Function Documentation

`void scale05Lanc6HV (UInt8 * in[6], UInt8 * out, unsigned int width)`

Referenced by `svuScI05Lanc6()`.

`int vStep (UInt8 * inRGB[6], int pos)`

Referenced by `scale05Lanc6HV()`, and `subs05sync7()`.

7.478 svuScale05Lanc7HV.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
```

Functions

- float `vStep` (`UInt8 *inRGB[7]`, `int pos`)
- void `subs05sync7` (`SippFilter *fptr`, `UInt8 *in[7]`, `UInt8 *out`)
- void `svuScI05Lanc7` (`SippFilter *fptr`)

Shave function of the [Lanczos Downscale by 2 \(7 taps\) filter](#).

7.478.1 Function Documentation

`void subs05sync7 (SippFilter * fptr, UInt8 * in[7], UInt8 * out)`

Referenced by `svuScI05Lanc7()`.

`float vStep (UInt8 * inRGB[7], int pos)`

7.479 svuScale2xBilinHV.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
```

Functions

- void `svuScI2xBilinHV` (`SippFilter *fptr`)
- Shave function of the [Upscale by 2 filter](#).*

7.480 svuScale2xBilinHV_025_075_Fp16ToFp16.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/scale2xBilinHV_025_075_Fp16ToFp16/scale2xBilinHV_025_075_Fp16ToFp16.h>
```

Functions

- void [scale2xBilinHV025_Fp16ToFp16](#) (half **in, half **out, **UInt32** inWidth)
- void [scale2xBilinHV075_Fp16ToFp16](#) (half **in, half **out, **UInt32** inWidth)
- void [svuScale2xBilinHV_025_075_Fp16ToFp16](#) (SippFilter *fptr)

Shave function of the [Upscale by 2 with phases 0.25 and 0.75 fp16 to fp16](#) filter.

7.480.1 Function Documentation

void [scale2xBilinHV025_Fp16ToFp16](#) (half ** in, half ** out, **UInt32** inWidth)

mvispScale2xBilinHV_025_075_Fp16ToFp16 Upscale by 2

Parameters

in	<i>in</i>	- Input lines
out	<i>out</i>	- Output line
in	<i>inWidth</i>	- Width of output line

Referenced by [svuScale2xBilinHV_025_075_Fp16ToFp16\(\)](#).

void [scale2xBilinHV075_Fp16ToFp16](#) (half ** in, half ** out, **UInt32** inWidth)

Referenced by [svuScale2xBilinHV_025_075_Fp16ToFp16\(\)](#).

7.481 svuScale2xBilinHV_025_075_U16ToU16.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/scale2xBilinHV_025_075_U16ToU16/scale2xBilinHV_025_075_U16ToU16.h>
```

Functions

- void [scale2xBilinHV025_U16ToU16](#) (UInt16 **in, UInt16 **out, **UInt32** inWidth)
- void [scale2xBilinHV075_U16ToU16](#) (UInt16 **in, UInt16 **out, **UInt32** inWidth)
- void [svuScale2xBilinHV_025_075_U16ToU16](#) (SippFilter *fptr)

Shave function of the [Upscale by 2 with phases 0.25 and 0.75 u16 to u16](#) filter.

7.481.1 Function Documentation

```
void scale2xBilinHV025_U16ToU16 ( UInt16 ** in, UInt16 ** out, UInt32 inWidth )
```

mvispScale2xBilinHV_025_075_U16ToU16 Upscale by 2

Parameters

in	<i>in</i>	- Input lines
out	<i>out</i>	- Output line
in	<i>inWidth</i>	- Width of output line

Referenced by svuScale2xBilinHV_025_075_U16ToU16().

```
void scale2xBilinHV075_U16ToU16 ( UInt16 ** in, UInt16 ** out, UInt32 inWidth )
```

Referenced by svuScale2xBilinHV_025_075_U16ToU16().

7.482 svuScale2xBilinHV_Fp16U8_phase025_075.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/scale2xBilinHV_Fp16U8_phase025_075/scale2xBilinHV-
_Fp16U8_phase025_075.h>
```

Functions

- void [scale2xBilinHV025_Fp16U8](#) (half **in, UInt8 **out, u32 inWidth)
- void [scale2xBilinHV075_Fp16U8](#) (half **in, UInt8 **out, u32 inWidth)
- void [svuScale2xBilinHV_Fp16U8_phase025_075](#) (SippFilter *fptr)
Shave function of the Upscale by 2 with phases 0.25 and 0.75 fp16 to u8 filter.

7.482.1 Function Documentation

```
void scale2xBilinHV025_Fp16U8 ( half ** in, UInt8 ** out, u32 inWidth )
```

mvispScale2xBilinHV_025_075_Fp16U8 Upscale by 2

Parameters

in	<i>in</i>	- Input lines
out	<i>out</i>	- Output line
in	<i>inWidth</i>	- Width of output line

Referenced by svuScale2xBilinHV_Fp16U8_phase025_075().

```
void scale2xBilinHV075_Fp16U8 ( half ** in, UInt8 ** out, u32 inWidth )
```

Referenced by svuScale2xBilinHV_Fp16U8_phase025_075().

7.483 svuScale2xBilinHV_U8ToU8_phase025_075.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/scale2xBilinHV_U8ToU8_phase025_075/scale2xBilinHV-
_U8ToU8_phase025_075.h>
```

Functions

- void [scale2xBilinHV025_U8ToU8](#) (UInt8 **in, UInt8 **out, UInt32 inWidth)
- void [scale2xBilinHV075_U8ToU8](#) (UInt8 **in, UInt8 **out, UInt32 inWidth)
- void [svuScale2xBilinHV_U8ToU8_phase025_075](#) (SippFilter *fptr)
Shave function of the [Upscale by 2 with phases 0.25 and 0.75 u8 to u8 filter](#).

7.483.1 Function Documentation

```
void scale2xBilinHV025_U8ToU8 ( UInt8 ** in, UInt8 ** out, UInt32 inWidth )
```

mvispScale2xBilinHV_025_075_U8ToU8 Upscale by 2

Parameters

in	<i>in</i>	- Input lines
out	<i>out</i>	- Output line
in	<i>inWidth</i>	- Width of output line

Referenced by [svuScale2xBilinHV_U8ToU8_phase025_075\(\)](#).

```
void scale2xBilinHV075_U8ToU8 ( UInt8 ** in, UInt8 ** out, UInt32 inWidth )
```

Referenced by [svuScale2xBilinHV_U8ToU8_phase025_075\(\)](#).

7.484 svuScale2xLancH.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
```

Functions

- void [svuScl2xLancH](#) (SippFilter *fptr)
Shave function of the [Lanczos Horizontal Upscale by 2 filter](#).

7.485 svuScale2xLancHV.c File Reference

```
#include <sipp.h>
```



```
#include <sippShaveMacros.h>
```

Macros

- #define **MARGIN** 4

Functions

- void **upscale2xH** (**SippFilter** *fptr, **UInt8** *out)
- void **upscale2xV** (**SippFilter** *fptr, **UInt8** *in[4], **UInt8** *out, int runNo)
- void **svuSc12xLancHV** (**SippFilter** *fptr)

*Shave function of the **Lanczos Upscale by 2** filter.*

Variables

- float **kern** [4]

7.485.1 Macro Definition Documentation

```
#define MARGIN 4
```

Referenced by **upscale2xV**().

7.485.2 Function Documentation

```
void upscale2xH ( SippFilter * fptr, UInt8 * out )
```

Referenced by **svuSc12xLancHV**().

```
void upscale2xV ( SippFilter * fptr, UInt8 * in[4], UInt8 * out, int runNo )
```

Referenced by **svuSc12xLancHV**().

7.485.3 Variable Documentation

```
float kern[4]
```

Initial value:

```
= { -1.0f/16.0f,
    9.0f/16.0f,
    9.0f/16.0f,
    -1.0f/16.0f }
```

Referenced by **scale05Lanc6HV**(), **subs05sync7**(), **svuSc12xLancH**(), **svuSc12xLancV**(), **upscale2xH**(), **upscale2xV**(), and **vStep**().

7.486 svuScale2xLancV.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
```

Functions

- void [svuScl2xLancV](#) ([SippFilter](#) *fptr)
Shave function of the [Lanczos Vertical Upscale by 2](#) filter.

7.487 svuScaleBilinArb.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
```

Functions

- void [svuSclBilinArb](#) ([SippFilter](#) *fptr)
Shave function of the [Arbitrary Downscale](#) filter.

7.488 svuScharr_fp16.c File Reference

main leon file

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/scharr_fp16/scharr_fp16.h>
#include <math.h>
#include <float.h>
```

Functions

- void [mvcvScharr_fp16](#) (half **in, half **out, [UInt32](#) width)
- void [svuScharr_fp16](#) ([SippFilter](#) *fptr)
Shave function of the [scharr_fp16](#) filter.

7.488.1 Detailed Description

main leon file

Copyright

All code copyright Movidius Ltd 2012, all rights reserved. For License Warranty see: common/license.txt

7.488.2 Function Documentation

`void mvcvScharr_fp16 (half ** in, half ** out, UInt32 width)`

This kernel performs scharr edge detection operator

Parameters

<code>in</code>	<code>in</code>	- Input lines
<code>out</code>	<code>out</code>	- Output line
<code>in</code>	<code>width</code>	- Width of input line

Referenced by `svuScharr_fp16()`.

7.489 svusLaplacian3x3Fp16ToFp16.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/sLaplacian3x3Fp16ToFp16/sLaplacian3x3Fp16ToFp16.h>
```

Functions

- void `sLaplacian3x3Fp16ToFp16` (half **in, half **out, UInt32 inWidth)
- void `svusLaplacian3x3Fp16ToFp16` (SippFilter *fptr)
Shave function of the [Laplacian 3x3 Fp16 To Fp16](#) filter.

7.489.1 Function Documentation

`void sLaplacian3x3Fp16ToFp16 (half ** in, half ** out, UInt32 inWidth)`

Laplacian filter - applies a Laplacian filter with custom size (see http://en.wikipedia.org/wiki/Discrete_Laplace_operator)

Parameters

<code>in</code>	<code>in</code>	- array of pointers to input lines
<code>out</code>	<code>out</code>	- pointer to output line
<code>in</code>	<code>width</code>	- width of input line

Referenced by `svusLaplacian3x3Fp16ToFp16()`.

7.490 svuSobel.c File Reference

main leon file

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/sobel/sobel.h>
#include <math.h>
```

Functions

- void [sobel](#) ([UInt8](#) **in, [UInt8](#) **out, [UInt32](#) width)
- void [svuSobel](#) ([SippFilter](#) *fptr)

Shave function of the [Sobel](#) filter.

7.490.1 Detailed Description

main leon file

Copyright

All code copyright Movidius Ltd 2012, all rights reserved. For License Warranty see: common/license.txt

7.490.2 Function Documentation

void [sobel](#) ([UInt8](#) ** in, [UInt8](#) ** out, [UInt32](#) width)

[sobel](#) filter - Filter, calculates magnitude

Parameters

in	<i>in</i>	- array of pointers to input lines
out		- array of pointers for output lines
in	<i>width</i>	- width of input line

Referenced by [svuSobel](#)().

7.491 svuSSD11x11.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/ssd11x11/ssd11x11.h>
```

Functions

- void [sumOfSquaredDiff11x11](#) ([UInt8](#) **in1, [UInt8](#) **in2, [UInt8](#) **out, [UInt32](#) width)
- void [svuSSD11x11](#) ([SippFilter](#) *fptr)

Shave function of the [Sum of Squared Differences 11x11](#) filter.

7.491.1 Function Documentation

void [sumOfSquaredDiff11x11](#) ([UInt8](#) ** in1, [UInt8](#) ** in2, [UInt8](#) ** out, [UInt32](#) width)

SSD (Sum of Squared differences) 11x11

Parameters

in	<i>in1</i>	- array of pointers to input lines from the first image
in	<i>in2</i>	- array of pointers to input lines from the second image
out	<i>out</i>	- array of pointers for output line
in	<i>width</i>	- width of input line

Referenced by svuSSD11x11().

7.492 svuSSD5x5.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/ssd5x5/ssd5x5.h>
```

Functions

- void [sumOfSquaredDiff5x5](#) (UInt8 **in1, UInt8 **in2, UInt8 **out, UInt32 width)
- void [svuSSD5x5](#) (SippFilter *fptr)

Shave function of the [Sum of Squared Differences 5x5](#) filter.

7.492.1 Function Documentation

```
void sumOfSquaredDiff5x5 ( UInt8 ** in1, UInt8 ** in2, UInt8 ** out, UInt32 width )
```

SSD (Sum of Squared differences) 5x5

Parameters

in	<i>in1</i>	- array of pointers to input lines from the first image
in	<i>in2</i>	- array of pointers to input lines from the second image
out	<i>out</i>	- array of pointers for output line
in	<i>width</i>	- width of input line

Referenced by svuSSD5x5().

7.493 svuSSD7x7U8ToU32.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/ssd7x7U8ToU32/ssd7x7U8ToU32.h>
```

Functions

- void [sumOfSquaredDiff7x7U8ToU32](#) (UInt8 **in1, UInt8 **in2, UInt32 **out, UInt32 width)
- void [svuSSD7x7U8ToU32](#) (SippFilter *fptr)

Shave function of the [Sum of Squared Differences 7x7 \(U8 to U32\)](#) filter.

7.493.1 Function Documentation

`void sumOfSquaredDiff7x7U8ToU32 (UInt8 ** in1, UInt8 ** in2, UInt32 ** out, UInt32 width)`

SSD (Sum of Squared differences) 7x7U8ToU32

Parameters

in	<i>in1</i>	- array of pointers to input lines from the first image
in	<i>in2</i>	- array of pointers to input lines from the second image
out	<i>out</i>	- array of pointers for output line
in	<i>width</i>	- width of input line

Referenced by `svuSSD7x7U8ToU32()`.

7.494 svuSsdPointLine7x7U8U32.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/ssdPointLine7x7U8U32/ssdPointLine7x7U8U32.h>
```

Functions

- void `mvsvSsdPointLine7x7U8U32` (UInt8 **in1, UInt8 **in2, UInt32 **out, UInt32 width)
- void `svuSsdPointLine7x7U8U32` (SippFilter *fptr)

Shave function of the [Sum of Squared Differences 7x7](#) filter.

7.494.1 Function Documentation

`void mvsvSsdPointLine7x7U8U32 (UInt8 ** in1, UInt8 ** in2, UInt32 ** out, UInt32 width)`

SSD (Sum of Squared differences) (7x7 in this case)

Parameters

in	<i>in1</i>	- Input lines from the first image, 8-bits unsigned integer
in	<i>in2</i>	- Input lines from the second image, 8-bits unsigned integer
out	<i>out</i>	- Output line, 32-bits unsigned integer
in	<i>width</i>	- Width of input line

Referenced by `svuSsdPointLine7x7U8U32()`.

7.495 svuStartBicubic.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/startBicubic/startBicubic.h>
#include <sippBicubic.h>
```

Macros

- #define `COMMAND_SIZEOF` 0x40

Functions

- void `configureBicubicHWblock` (`StartBicubicParam` *p, `UInt64` *input)
- void `svuStartBicubic` (`SippFilter` *fptr)

7.495.1 Macro Definition Documentation

```
#define COMMAND_SIZEOF 0x40
```

Referenced by `configureBicubicHWblock()`.

7.495.2 Function Documentation

```
void configureBicubicHWblock ( StartBicubicParam * p, UInt64 * input )
```

Referenced by `svuStartBicubic()`.

```
void svuStartBicubic ( SippFilter * fptr )
```

7.496 svuStatsAwbSatPixels.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/statsAwbSatPixels/statsAwbSatPixels.h>
```

Macros

- #define `CLAMPZ255`(X) ((X)>255 ? 255 : ((X)<0 ? 0 : (X)))

Functions

- void `statsAWBSatPixels` (`UInt8` *output, `UInt16` *iline, `UInt32` nPaxelX, `UInt32` gapPaxelX, `UInt32` widthPaxelX, `UInt32` satPaxelX)
- void `svuStatsAwbSatPixels` (`SippFilter` *fptr)

7.496.1 Macro Definition Documentation

```
#define CLAMPZ255( X ) ((X)>255 ? 255 : ((X)<0 ? 0 : (X)))
```

Referenced by `statsAWBSatPixels()`.

7.496.2 Function Documentation

```
void statsAWBSatPixels ( UInt8 * output, UInt16 * iline, UInt32 nPaxelX, UInt32 gapPaxelX,
    UInt32 widthPaxelX, UInt32 satPaxelX )
```

Referenced by svuStatsAwbSatPixels().

```
void svuStatsAwbSatPixels ( SippFilter * fptr )
```

7.497 svuStatsAwbSatPixelsU32.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/statsAwbSatPixelsU32/statsAwbSatPixelsU32.h>
```

Functions

- void [statsAWBSatPixelsU32](#) (UInt32 *output, UInt16 *iline, UInt32 nPaxelX, UInt32 gapPaxelX, UInt32 widthPaxelX, UInt32 satPaxelX)
- void [svuStatsAwbSatPixelsU32](#) (SippFilter *fptr)

7.497.1 Function Documentation

```
void statsAWBSatPixelsU32 ( UInt32 * output, UInt16 * iline, UInt32 nPaxelX, UInt32 gapPaxelX,
    UInt32 widthPaxelX, UInt32 satPaxelX )
```

Referenced by svuStatsAwbSatPixelsU32().

```
void svuStatsAwbSatPixelsU32 ( SippFilter * fptr )
```

7.498 svuSubpixelFilter.c File Reference

```
#include <sipp.h>
#include <sippInternal.h>
#include <sippShaveMacros.h>
#include <filters/subpixelFilter/subpixelFilter.h>
```

Functions

- void [mvcvSubPixelFilter_asm](#) (u8 *inputDisparityMap, u8 *inputCostVolume, u16 *outputDisparityMap, u32 width, u8 maxDisp, u8 numFractionalBit, u8 *lutToUse)
- void [svuSubpixelFilter](#) (SippFilter *fptr)

7.498.1 Function Documentation

```
void mvcvSubPixelFilter_asm ( u8 * inputDisparityMap, u8 * inputCostVolume, u16 *
outputDisparityMap, u32 width, u8 maxDisp, u8 numFractionalBit, u8 * lutToUse )
```

Referenced by svuSubpixelFilter().

```
void svuSubpixelFilter ( SippFilter * fptr )
```

7.499 svuThreshold.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/threshold/threshold.h>
```

Functions

- void thresholdKernel (UInt8 **in, UInt8 **out, UInt32 width, UInt32 height, UInt8 thresh, UInt32 thresh_type)
- void svuThreshold (SippFilter *fptr)
Shave function of the Threshold filter.

7.499.1 Function Documentation

```
void thresholdKernel ( UInt8 ** in, UInt8 ** out, UInt32 width, UInt32 height, UInt8 thresh,
UInt32 thresh_type )
```

threshold kernel computes the output image based on a threshold value and a threshold type

Parameters

in	in	- array of pointers to input lines
out	out	- array of pointers to output lines
in	width	- width of the input line
in	height	- height of the input line
in	thresh	- threshold value
in	thresh_type	- one of the 5 available thresholding types: <ul style="list-style-type: none">• Thresh_To_Zero: values below threshold are zeroed• Thresh_To_Zero_Inv: opposite of Thresh_To_Zero• Thresh_To_Binary: values below threshold are zeroed and all others are saturated to pixel max value• Thresh_To_Binary_Inv: opposite of Thresh_To_Binary• Thresh_Trunc: values above threshold are given threshold value• default mode: Thresh_Trunc

Returns

Nothing

Referenced by svuThreshold().

7.500 svuThresholdBinaryRange.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/thresholdBinaryRange/thresholdBinaryRange.h>
```

Functions

- void [thresholdBinaryRange](#) (UInt8 **in, UInt8 **out, UInt8 lowerValue, UInt8 upperValue, UInt32 width)
- void [svuThresholdBinaryRange](#) (SippFilter *fptr)
Shave function of the [Threshold Binary Range](#) filter.

7.500.1 Function Documentation

[void thresholdBinaryRange](#) (UInt8 ** in, UInt8 ** out, UInt8 lowerValue, UInt8 upperValue, UInt32 width)

[thresholdBinaryRange](#) kernel -this kernel set output to 0xFF if source(x,y) is in specified range, otherwise output = 0.

Parameters

in	<i>in</i>	- array of pointers to input lines
out	<i>out</i>	- array of pointers for output lines
in	<i>lowerValue</i>	- lowerValue
in	<i>upperValue</i>	- upperValue
in	<i>width</i>	- width of input line

Referenced by svuThresholdBinaryRange().

7.501 svuThresholdBinaryU8.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/thresholdBinaryU8/thresholdBinaryU8.h>
```

Functions

- void [thresholdBinaryU8](#) (UInt8 **in, UInt8 **out, UInt8 threshold, UInt32 width)
- void [svuThresholdBinaryU8](#) (SippFilter *fptr)
Shave function of the [Threshold Binary U8](#) filter.

7.501.1 Function Documentation

`void thresholdBinaryU8 (UInt8 ** in, UInt8 ** out, UInt8 threshold, UInt32 width)`

thresholdBinaryU8 kernel -this kernel set output to 0 if threshold value is less then input value and to 0xFF if threshold value is greater then input value

Parameters

in	<i>in</i>	- array of pointers to input lines
out		- array of pointers for output lines
	<i>threshold</i>	- threshold value
in	<i>width</i>	- width of input line

Referenced by svuThresholdBinaryU8().

7.502 svuThresholdFilter.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/thresholdFilter/thresholdFilter.h>
```

Functions

- void [mvcvThresholdFilter](#) (float *cornerScores, float threshold, UInt32 width, UInt32 posOffset, UInt16 *candPos, UInt32 *candCount)
- void [svuThresholdFilter](#) (SippFilter *fptr)
Shave function of the [ThresholdFilter](#) filter.

7.502.1 Function Documentation

`void mvcvThresholdFilter (float * cornerScores, float threshold, UInt32 width, UInt32 posOffset, UInt16 * candPos, UInt32 * candCount)`

Parameters

in	<i>cornerScores</i>	- pointer to the input line
in	<i>threshold</i>	- threshold value
in	<i>width</i>	- input line width
in	<i>posOffset</i>	- offset value
out	<i>candPos</i>	- pointer to the output line
out	<i>candCount</i>	- pointer to the width of the output line

Referenced by svuThresholdFilter().

7.503 svuUndistortBrown.c File Reference

```
#include <sipp.h>
```

```
#include <sippShaveMacros.h>
#include <filters/undistortBrown/undistortBrown.h>
```

Functions

- **UInt32** **getInPtrAbs** (**SippFilter** *fptr, **UInt32** parNo, **UInt32** lineNo, **UInt32** planeNo, **UInt32** targetSlice)
- void **svuUndistortBrown** (**SippFilter** *fptr)
*Shave function of the **Undistort** filter.*

7.503.1 Function Documentation

UInt32 **getInPtrAbs** (**SippFilter** * fptr, **UInt32** parNo, **UInt32** lineNo, **UInt32** planeNo, **UInt32** targetSlice)

7.504 svuWhiteBalanceBayerGBRG.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/whiteBalanceBayerGBRG/whiteBalanceBayerGBRG.h>
```

Functions

- void **whiteBalanceBayerGBRG** (**UInt16** **inGBRG, **UInt16** awbCoef[3], **UInt16** **outGBRG, **UInt16** clamp[1], **UInt32** width, **UInt32** line)
- void **svuWhiteBalanceBayerGBRG** (**SippFilter** *fptr)
*Shave function of the **White Balance Bayer GBRG** filter.*

7.504.1 Function Documentation

void **whiteBalanceBayerGBRG** (**UInt16** ** inGBRG, **UInt16** awbCoef[3], **UInt16** ** outGBRG, **UInt16** clamp[1], **UInt32** width, **UInt32** line)

White Balance Gains for BayerGBRG input

Parameters

in	<i>inGBRG</i>	- pointer to the first line from input
	<i>Out</i>	outGBRG - pointer to the first line from output R plane
in	<i>awbCoef</i>	- awb gains coefs
in	<i>clamp</i>	- clamp value
in	<i>line</i>	- line's number

<code>in</code>	<code>width</code>	- line width in pixels
-----------------	--------------------	------------------------

Returns

Nothing

Referenced by `svuWhiteBalanceBayerGBRG()`.

7.505 `svuWhiteBalanceRGB.c` File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/whiteBalanceRGB/whiteBalanceRGB.h>
```

Functions

- void `whiteBalanceRGB` (`UInt16 **inR`, `UInt16 **inG`, `UInt16 **inB`, `UInt16 awbCoef[3]`, `UInt16 **outR`, `UInt16 **outG`, `UInt16 **outB`, `UInt16 clamp[1]`, `UInt32 width`)
- void `svuWhiteBalanceRGB` (`SippFilter *fptr`)
Shave function of the [White Balance RGB](#) filter.

7.505.1 Function Documentation

```
void whiteBalanceRGB ( UInt16 ** inR, UInt16 ** inG, UInt16 ** inB, UInt16 awbCoef[3],  

UInt16 ** outR, UInt16 ** outG, UInt16 ** outB, UInt16 clamp[1], UInt32 width )
```

White Balance Gains for RGB input

Parameters

<code>in</code>	<code>inR</code>	- pointer to the first line from input R plane
<code>in</code>	<code>inG</code>	- pointer to the first line from input G plane
<code>in</code>	<code>inB</code>	- pointer to the first line from input B plane
	<code>Out]</code>	outR - pointer to the first line from output R plane
	<code>Out]</code>	outG - pointer to the first line from output G plane
	<code>Out]</code>	outB - pointer to the first line from output B plane
<code>in</code>	<code>awbCoefs</code>	- awb gains coefs
<code>in</code>	<code>clamp</code>	- clamp value
<code>in</code>	<code>width</code>	- line width in pixels

Returns

Nothing

Referenced by `svuWhiteBalanceRGB()`.

7.506 svuXYgen.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/xyGen/xyGen.h>
```

Functions

- void [genXYlist](#) (float *warp_matrix, [UInt64](#) *output, [UInt32](#) width, [UInt32](#) lineNo, [UInt32](#) firstShave, [UInt32](#) lastShave)
- void [svuXYgen](#) ([SippFilter](#) *fptr)
Shave function of the [XY Generator](#) filter.

7.506.1 Function Documentation

void [genXYlist](#) (float * warp_matrix, [UInt64](#) * output, [UInt32](#) width, [UInt32](#) lineNo, [UInt32](#) firstShave, [UInt32](#) lastShave)

Referenced by [svuXYgen\(\)](#).

7.507 threshold.h File Reference

This file contains the declaration of the [Threshold](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [ThresholdParam](#)
Parameter structure of the [Threshold](#) filter.

Enumerations

- enum {
 [Thresh_To_Zero](#) = 0, [Thresh_To_Zero_Inv](#) = 1, [Thresh_To_Binary](#) = 2, [Thresh_To_Binary_Inv](#) = 3,
 [Thresh_Trunc](#) = 4 }

Functions

- void [SVU_SYM\(\)](#) [svuThreshold](#) ([SippFilter](#) *fptr)
Shave function of the [Threshold](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuThreshold](#))

7.507.1 Detailed Description

This file contains the declaration of the [Threshold](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.508 thresholdBinaryRange.h File Reference

This file contains the declaration of the [Threshold Binary Range](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [ThresholdBinaryRangeParam](#)
Parameter structure of the [Threshold Binary Range](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuThresholdBinaryRange](#) ([SippFilter](#) *fptr)
Shave function of the [Threshold Binary Range](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuThresholdBinaryRange](#))

7.508.1 Detailed Description

This file contains the declaration of the [Threshold Binary Range](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.509 thresholdBinaryU8.h File Reference

This file contains the declaration of the [Threshold Binary U8](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [ThresholdBinaryU8Param](#)
Parameter structure of the [Threshold Binary U8](#) filter.

Functions

- void **SVU_SYM()** **svuThresholdBinaryU8** (SippFilter *fptr)
*Shave function of the **Threshold Binary U8** filter.*
- **SHAVE_SYM_EXPORT** (svuThresholdBinaryU8)

7.509.1 Detailed Description

This file contains the declaration of the **Threshold Binary U8** SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.510 thresholdFilter.h File Reference

This file contains the declaration of the **ThresholdFilter** SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct **ThresholdFilterParam**
*Parameter structure of the **ThresholdFilter** filter.*

Functions

- void **SVU_SYM()** **svuThresholdFilter** (SippFilter *fptr)
*Shave function of the **ThresholdFilter** filter.*
- **SHAVE_SYM_EXPORT** (svuThresholdFilter)

7.510.1 Detailed Description

This file contains the declaration of the **ThresholdFilter** SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.511 tripleConv3x3.c File Reference

```
#include <sipp.h>
#include <sippShaveMacros.h>
#include <filters/conv3x3/conv3x3.h>
#include <filters/tripleConv3x3/tripleConv3x3.h>
```


Functions

- `TripleConv3x3 * createTripleConv3x3 (SippPipeline *pl, UInt32 width, UInt32 height)`

7.511.1 Function Documentation

TripleConv3x3* createTripleConv3x3 (**SippPipeline** * pl, **UInt32** width, **UInt32** height)

7.512 tripleConv3x3.h File Reference

```
#include <sipp.h>
```

Data Structures

- struct **TripleConv3x3**

Functions

- `TripleConv3x3 * createTripleConv3x3 (SippPipeline *pl, UInt32 width, UInt32 height)`

7.512.1 Function Documentation

TripleConv3x3* createTripleConv3x3 (**SippPipeline** * pl, **UInt32** width, **UInt32** height)

7.513 undistortBrown.h File Reference

This file contains the declaration of the **Undistort** SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct **UndistortBParam**
*Parameter structure of the **Undistort** filter.*

Functions

- void **SVU_SYM**() **svuUndistortBrown** (**SippFilter** *fptr)
*Shave function of the **Undistort** filter.*
- **SHAVE_SYM_EXPORT** (**svuUndistortBrown**)

7.513.1 Detailed Description

This file contains the declaration of the **Undistort** SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.514 whiteBalanceBayerGBRG.h File Reference

This file contains the declaration of the [White Balance Bayer GBRG](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [WhiteBalanceBayerGBRGParam](#)
Parameter structure of the [White Balance Bayer GBRG](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuWhiteBalanceBayerGBRG](#) ([SippFilter](#) *fptr)
Shave function of the [White Balance Bayer GBRG](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuWhiteBalanceBayerGBRG](#))

7.514.1 Detailed Description

This file contains the declaration of the [White Balance Bayer GBRG](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: common/license.txt

7.515 whiteBalanceRGB.h File Reference

This file contains the declaration of the [White Balance RGB](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [WhiteBalanceRGBParam](#)
Parameter structure of the [White Balance RGB](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuWhiteBalanceRGB](#) ([SippFilter](#) *fptr)
Shave function of the [White Balance RGB](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuWhiteBalanceRGB](#))

7.515.1 Detailed Description

This file contains the declaration of the [White Balance RGB](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: [common/license.txt](#)

7.516 wrapperSem.cpp File Reference

```
#include "wrapperSem.h"
```

7.517 wrapperSem.cpp File Reference

```
#include "wrapperSem.h"
```

7.518 wrapperSem.h File Reference

```
#include <semaphore.h>
#include <memory>
```

Data Structures

- class [Semaphore](#)

7.519 wrapperSem.h File Reference

```
#include <semaphore.h>
#include <memory>
```

Data Structures

- class [Semaphore](#)

7.520 xyGen.h File Reference

This file contains the declaration of the [XY Generator](#) SIPP filter API.

```
#include <sipp.h>
```

Data Structures

- struct [XYGenParam](#)

Parameter structure of the [XY Generator](#) filter.

Functions

- void [SVU_SYM\(\)](#) [svuXYgen](#) ([SippFilter](#) *fptr)
Shave function of the [XY Generator](#) filter.
- [SHAVE_SYM_EXPORT](#) ([svuXYgen](#))

7.520.1 Detailed Description

This file contains the declaration of the [XY Generator](#) SIPP filter API.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see:
common/license.txt

Index

- ~Semaphore
 - Semaphore, [422](#), [423](#)
- _16bitmode
 - Disp2depthParam, [373](#)
- 12Bpp to 8Bpp conversion, [185](#)
 - SHAVE_SYM_EXPORT, [185](#)
 - svuConvertFrom12BppTo8Bpp, [185](#)
- a_f1f2
 - PpAf, [415](#)
- ACTIVE
 - sippHwCommon_ma2x5x.h, [729](#)
- AF Stats, [233](#)
 - SHAVE_SYM_EXPORT, [233](#)
 - svuExtAfStats, [233](#)
- AF_paxel_statistics, [331](#)
 - filter1_number_of_used_pixels_green, [331](#)
 - filter1_sum_green, [331](#)
 - filter1_sum_max_green, [331](#)
 - filter2_number_of_used_pixels_green, [331](#)
 - filter2_sum_green, [331](#)
 - filter2_sum_max_green, [331](#)
 - sum_all_green, [331](#)
 - UNDEFINED, [331](#)
- ALIGNED
 - Disp2depthParam, [373](#)
 - sippMem.c, [772](#)
 - sippPlatform_ma2x5x.h, [809](#)
- AMC_WIDTH
 - sippHwCommon_ma2x5x.h, [729](#)
- absdiff.h, [493](#)
- Absolute difference, [115](#)
 - SHAVE_SYM_EXPORT, [115](#)
 - svuAbsdiff, [115](#)
- AbsoluteDiff
 - svuAbsdiff.c, [842](#)
- accum
 - ae_patch_stats, [330](#)
- Accumulate Square, [116](#)
 - SHAVE_SYM_EXPORT, [116](#)
 - svuAccumulateSquare, [116](#)
- Accumulate Weighted, [117](#)
 - SHAVE_SYM_EXPORT, [117](#)
 - svuAccumulateWeighted, [117](#)
- AccumulateSquare
 - svuAccumulateSquare.c, [843](#)
- accumulateSquare.h, [493](#)
- AccumulateWeighted
 - svuAccumulateWeighted.c, [843](#)
- accumulateWeighted.h, [494](#)
- AccumulateWeightedParam, [329](#)
 - alpha, [329](#)
- addr
 - sSippCdmaQuEntryS, [468](#)
 - sSippCMDQuEntryS, [469](#)
- AddressType
 - sippHwCommon_ma2x5x.h, [734](#)
- adiff
 - svuFast9M2.c, [909](#)
 - svuFast9ScoreCv.c, [911](#)
- adjustNodesRecursively
 - sippCoreUtils.c, [626](#)
- ae_patch_stats, [329](#)
 - accum, [330](#)
 - alt_accum, [330](#)
 - count, [330](#)
- AeAwbStatsCfg, [330](#)
 - firstPatchX, [330](#)
 - firstPatchY, [330](#)
 - nPatchesX, [330](#)
 - nPatchesY, [330](#)
 - patchGapX, [330](#)
 - patchGapY, [330](#)
 - patchHeight, [330](#)
 - patchWidth, [330](#)
 - satThresh, [330](#)
 - StatsAwbSatPixelsParam, [471](#)
 - StatsAwbSatPixelsParamU32, [473](#)
- afConfig
 - PpAf, [415](#)
- afF1coefs
 - RawParam, [419](#)
- afF2coefs
 - RawParam, [419](#)
- afMinThresh
 - RawParam, [419](#)
- afPatchCfg

- RawParam, [419](#)
- afPatchStart
 - RawParam, [419](#)
- afStatsBase
 - RawParam, [420](#)
- afSubtract
 - RawParam, [420](#)
- allDoneMask
 - SippPipelineS, [451](#)
- allMask
 - SchedInfoS, [421](#)
- allocReq
 - sSchLineBufferS, [466](#)
- alpha
 - AccumulateWeightedParam, [329](#)
- alphaBadPixel
 - Low Level Correction, [267](#)
 - LowLvlCorrNPIParam, [391](#)
- alt_accum
 - ae_patch_stats, [330](#)
- Arbitrary Downscale, [311](#)
 - SHAVE_SYM_EXPORT, [311](#)
 - svuScdBilinArb, [311](#)
- args
 - SIPP_PAL_THREAD, [430](#)
- Arithmetic, [121](#)
 - SHAVE_SYM_EXPORT, [126](#)
 - svuArithmeticSub, [126](#)
- Arithmetic addition, [118](#)
 - SHAVE_SYM_EXPORT, [118](#)
 - svuArithmeticAdd, [118](#)
- Arithmetic addition with mask, [119](#)
 - SHAVE_SYM_EXPORT, [119](#)
 - svuArithmeticAddmask, [119](#)
- Arithmetic subtraction, [120](#)
- Arithmetic subtraction fp16, [127](#)
 - SHAVE_SYM_EXPORT, [127](#)
 - svuArithmeticSubFp16ToFp16, [127](#)
- Arithmetic subtraction with mask, [128](#)
 - SHAVE_SYM_EXPORT, [128](#)
 - svuArithmeticSubmask, [128](#)
- arithmeticAdd.h, [494](#)
- arithmeticAddImplementation
 - svuArithmeticAdd.c, [844](#)
- arithmeticAddmask.h, [495](#)
- arithmeticAddmaskImplementation
 - svuArithmeticAddmask.c, [845](#)
- arithmeticSub.h, [495](#)
- arithmeticSubFp16ToFp16
 - svuArithmeticSubFp16ToFp16.c, [846](#)
- arithmeticSubFp16ToFp16.h, [496](#)
- arithmeticSubImplementation
 - svuArithmeticSub.c, [845](#)
- arithmeticSubmask.h, [496](#)
- arithmeticSubmaskImplementation
 - svuArithmeticSubmask.c, [846](#)
- askChromaGenDownsizer
 - sippGenericSchReq.c, [646](#)
- askCrop
 - sippGenericSchReq.c, [646](#)
- askCropLatency
 - sippGenericSchReq.c, [646](#)
- askHwColorCombChroma
 - sippGenericSchReq.c, [646](#)
- askHwMedLumaLatency
 - sippGenericSchReq.c, [646](#)
- askPolyFirResizer
 - sippGenericSchReq.c, [646](#)
- askRegular
 - sippGenericSchReq.c, [646](#)
- askRegularLatency
 - sippGenericSchReq.c, [646](#)
- askResizer
 - sippGenericSchReq.c, [646](#)
- askResizerLatency
 - sippGenericSchReq.c, [646](#)
- assigned
 - tsSippHeap, [482](#)
- autoType
 - PolyFirParam, [412](#)
- Average, [129](#)
 - SHAVE_SYM_EXPORT, [129](#)
 - svuAvg, [129](#)
- avg.h, [497](#)
- awbCoef
 - PixelUnpackerWBParam, [411](#)
 - WhiteBalanceBayerGBRGParam, [486](#)
 - WhiteBalanceRGBParam, [487](#)
- BAYER_FORMAT_BGGR
 - combDecimDemosaicAwbGains.h, [513](#)
- BAYER_FORMAT_GBRG
 - combDecimDemosaicAwbGains.h, [513](#)
- BAYER_FORMAT_GRBG
 - combDecimDemosaicAwbGains.h, [513](#)
- BAYER_FORMAT_RGGB
 - combDecimDemosaicAwbGains.h, [513](#)
- b_f1f2
 - PpAf, [415](#)
- BAYER
 - sippHwCommon_ma2x5x.h, [729](#)
- BAYER_ORDER_BGGR
 - sippHwBitfieldDefs.h, [675](#)

BAYER_ORDER_GBRG
 sippHwBitfieldDefs.h, 675
 BAYER_ORDER_GRBG
 sippHwBitfieldDefs.h, 675
 BAYER_ORDER_RGGG
 sippHwBitfieldDefs.h, 675
 bChunked
 DmaParam, 375
 BGEN_KERNEL_SIZE
 sippHwCommon_ma2x5x.h, 729
 BGGR
 sippHwCommon_ma2x5x.h, 729
 bInit
 tSippFramework, 479
 BM/src/leon/sippPalTypes.h
 SIPP_TRACE_ANY, 790
 SIPP_TRACE_API, 790
 SIPP_TRACE_CORE, 790
 SIPP_TRACE_HEAP, 790
 SIPP_TRACE_MAX, 790
 SIPP_TRACE_PAL, 791
 BM/src/leon/sippPlatformAbstractionLayer.c
 sippPalCriticalSectionBegin, 811
 sippPalCriticalSectionEnd, 811
 sippPalFindHighestBit, 811
 sippPalMemCompare, 811
 sippPalMemcpy, 811
 sippPalMemset, 811
 sippPalPrintInt, 812
 sippPalTrace, 812
 sippPalTraceInit, 812
 BOX_FILT
 svuBoxFilter.c, 852
 BOX_MAX_V_SZ
 boxFilter.h, 501
 BPP
 sipp_ma2x5x.h, 597
 bSVUOnly
 tSippPipelineSuper, 481
 BUFF_HUGE_SZ
 sippCfg.h, 617
 bWaitLongest
 SIPP_ACCESS_SCHEDULER, 425
 badPixCfg
 RawParam, 420
 base
 ExtStatsSatPixelsU32Param, 381
 SippHwBufS, 444
 SippHwIOBuf, 445
 BayerFormat
 combDecimDemosaicAwbGains.h, 513
 bayerOrder
 CombDecimAwbGainsParam, 343
 bayerPattern
 SigmaParam, 424
 bilateral5x5, 130
 SHAVE_SYM_EXPORT, 130
 svuBilateral5x5, 130
 bilateral5x5.h, 497
 Bilateral5x5Param, 331
 sigma, 331
 bilateralVal
 svuBilateral5x5.c, 847
 Bitwise And, 131
 SHAVE_SYM_EXPORT, 131
 svuBitwiseAnd, 131
 Bitwise And with mask, 132
 SHAVE_SYM_EXPORT, 132
 svubitwiseAndMask, 132
 Bitwise Not, 133
 SHAVE_SYM_EXPORT, 133
 svuBitwiseNot, 133
 Bitwise Or, 134
 SHAVE_SYM_EXPORT, 134
 svuBitwiseOr, 134
 Bitwise Or with mask, 135
 SHAVE_SYM_EXPORT, 135
 svuBitwiseOrMask, 135
 Bitwise Xor, 136
 SHAVE_SYM_EXPORT, 136
 svuBitwiseXor, 136
 Bitwise Xor with mask, 137
 SHAVE_SYM_EXPORT, 137
 svuBitwiseXorMask, 137
 bitwiseAnd
 svuBitwiseAnd.c, 848
 bitwiseAnd.h, 498
 bitwiseAndMask
 svuBitwiseAndMask.c, 849
 bitwiseAndMask.h, 498
 bitwiseNot
 svuBitwiseNot.c, 849
 bitwiseNot.h, 498
 bitwiseOr
 svuBitwiseOr.c, 850
 bitwiseOr.h, 499
 bitwiseOrMask
 svuBitwiseOrMask.c, 851
 bitwiseOrMask.h, 499
 bitwiseXor
 svuBitwiseXor.c, 851
 bitwiseXor.h, 500

- bitwiseXorMask
 - svuBitwiseXorMask.c, [852](#)
- bitwiseXorMask.h, [500](#)
- black01
 - MipiRxParam, [403](#)
- black23
 - MipiRxParam, [403](#)
- blackLevel
 - Low Level Correction, [267](#)
 - LowLvlCorrNPIParam, [391](#)
- blcB
 - SigmaParam, [424](#)
- blcGB
 - SigmaParam, [424](#)
- blcGR
 - SigmaParam, [424](#)
- blcR
 - SigmaParam, [424](#)
- bool
 - sippBaseTypes.h, [616](#)
- bottomLineReplication
 - sSchLineBufferS, [466](#)
- Box Filter 11x11, [140](#)
 - SHAVE_SYM_EXPORT, [140](#)
 - svuBoxFilter11x11, [140](#)
- Box Filter 13x13, [141](#)
 - SHAVE_SYM_EXPORT, [141](#)
 - svuBoxFilter13x13, [141](#)
- Box Filter 15x15, [142](#)
 - SHAVE_SYM_EXPORT, [142](#)
 - svuBoxFilter15x15, [142](#)
- Box Filter 3x3, [143](#)
 - SHAVE_SYM_EXPORT, [143](#)
 - svuBoxFilter3x3, [143](#)
- Box Filter 5x5, [144](#)
 - SHAVE_SYM_EXPORT, [144](#)
 - svuBoxFilter5x5, [144](#)
- Box Filter 7x7, [145](#)
 - SHAVE_SYM_EXPORT, [145](#)
 - svuBoxFilter7x7, [145](#)
- Box Filter 9x9, [146](#)
 - SHAVE_SYM_EXPORT, [146](#)
 - svuBoxFilter9x9, [146](#)
- boxDataFmt
 - Generic Box Filter, [138](#)
- boxFilter.h, [501](#)
 - BOX_MAX_V_SZ, [501](#)
- boxFilter11x11.h, [502](#)
- BoxFilter11x11Param, [332](#)
 - normalize, [332](#)
- boxFilter13x13.h, [502](#)
 - BoxFilter13x13Param, [332](#)
 - normalize, [332](#)
- boxFilter15x15.h, [503](#)
- BoxFilter15x15Param, [333](#)
 - normalize, [333](#)
- boxFilter3x3.h, [503](#)
- BoxFilter3x3Param, [333](#)
 - normalize, [333](#)
- boxFilter5x5.h, [504](#)
- BoxFilter5x5Param, [334](#)
 - normalize, [334](#)
- boxFilter7x7.h, [504](#)
- BoxFilter7x7Param, [334](#)
 - normalize, [334](#)
- boxFilter9x9.h, [505](#)
- BoxFilter9x9Param, [335](#)
 - normalize, [335](#)
- BoxFilterParam, [335](#)
 - dataFormat, [336](#)
 - filterSizeH, [336](#)
 - filterSizeV, [336](#)
 - normalize, [336](#)
- boxfilter
 - svuBoxFilter.c, [853](#)
- boxfilter11x11
 - svuBoxFilter11x11.c, [853](#)
- boxfilter13x13
 - svuBoxFilter13x13.c, [854](#)
- boxfilter15x15
 - svuBoxFilter15x15.c, [854](#)
- boxfilter3x3
 - svuBoxFilter3x3.c, [855](#)
- boxfilter5x5
 - svuBoxFilter5x5.c, [856](#)
- boxfilter7x7
 - svuBoxFilter7x7.c, [856](#)
- boxfilter9x9
 - svuBoxFilter9x9.c, [857](#)
- bpp
 - MipiRxLoopbackParam, [402](#)
 - MipiTxLoopbackParam, [404](#)
 - SippFilterS, [433](#)
 - StartBicubicParam, [470](#)
- buffer
 - svuCannyEdgeDetection.c, [858](#)
 - svuCornerMinEigenVal.c, [891](#)
- bulkBuff
 - svuFast9ScoreCv.c, [912](#)
- bytesPerPix
 - SippFilterS, [433](#)
- c1

- TripleConv3x3, [478](#)
- c2
 - TripleConv3x3, [478](#)
- c3
 - TripleConv3x3, [478](#)
- CMX_ADDRESS
 - sippHwCommon_ma2x5x.h, [734](#)
- CMX_MIRRORED_ADDRESS
 - sippHwCommon_ma2x5x.h, [734](#)
- CC_CFG
 - sippHwBitfieldDefs.h, [675](#)
- CC_CFG_FORCE_LUMA
 - sippHwBitfieldDefs.h, [676](#)
- CCM_COEF
 - sippUtils.c, [836](#)
- CGEN_CFG_BYPASS
 - sippHwBitfieldDefs.h, [676](#)
- CGEN_CFG_SET
 - sippHwBitfieldDefs.h, [677](#)
- CHK_HW_PEND_LIST
 - sippAccessSchedulerTypes.h, [610](#)
- CHROMA_DNS_CFG
 - sippHwBitfieldDefs.h, [679](#)
- CHROMA_DNS_HOR_T1
 - sippHwBitfieldDefs.h, [681](#)
- CHROMA_DNS_HOR_T2
 - sippHwBitfieldDefs.h, [681](#)
- CHROMA_DNS_HOR_T3
 - sippHwBitfieldDefs.h, [681](#)
- CHROMA_DNS_LIMIT
 - sippHwBitfieldDefs.h, [682](#)
- CHROMA_DNS_VER_T1
 - sippHwBitfieldDefs.h, [682](#)
- CHROMA_DNS_VER_T2
 - sippHwBitfieldDefs.h, [682](#)
- CHROMA_DNS_VER_T3
 - sippHwBitfieldDefs.h, [682](#)
- CLAMPU8
 - svuConvert16bppTo8bpp.c, [880](#)
- CLAMPZ255
 - svuCombDecimDemosaicAwbGains.c, [866](#)
 - svuStatsAwbSatPixels.c, [967](#)
- CLR_HW_PEND_LIST
 - sippAccessSchedulerTypes.h, [610](#)
- CLR_HW_WAIT_IVAL
 - sippAccessSchedulerTypes.h, [610](#)
- CLR_SW_PEND_LIST
 - sippAccessSchedulerTypes.h, [610](#)
- CMD_EXIT
 - sippDefines.h, [631](#)
- CMD_H_PAD
 - sippDefines.h, [631](#)
- CMD_RUN
 - sippDefines.h, [631](#)
- CMX_BSS
 - sippPlatform_ma2x5x.h, [809](#)
- CMX_DATA
 - sippPlatform_ma2x5x.h, [809](#)
- CMX_NSLICES
 - sippHwCommon_ma2x5x.h, [729](#)
- CMX_RODATA
 - sippPlatform_ma2x5x.h, [809](#)
- CMX_TEXT
 - sippPlatform_ma2x5x.h, [809](#)
- cMat
 - Conv11x11Param, [346](#)
 - Conv15x1Param, [347](#)
 - Conv1x15Param, [348](#)
 - Conv1x5Fp16ToFp16Param, [348](#)
 - Conv1x5Param, [349](#)
 - Conv1x7Fp16ToFp16Param, [350](#)
 - Conv1x7Param, [350](#)
 - Conv1x9Param, [351](#)
 - Conv3x3Fp16ToFp16Param, [351](#)
 - Conv3x3Param, [352](#)
 - Conv5x1Fp16ToFp16Param, [353](#)
 - Conv5x1Param, [353](#)
 - Conv5x5Fp16ToFp16Param, [354](#)
 - Conv5x5Param, [355](#)
 - Conv7x1Fp16ToFp16Param, [355](#)
 - Conv7x1Param, [356](#)
 - Conv7x7Param, [356](#)
 - Conv7x7ParamFp16ToFp16, [357](#)
 - Conv7x7ParamFp16ToU8, [358](#)
 - Conv9x1Param, [358](#)
 - Conv9x9Fp16ToFp16Param, [359](#)
 - Conv9x9Param, [360](#)
 - ConvGenericParam, [360](#)
 - ConvSeparable11x11Fp16ToFp16Param, [362](#)
 - ConvSeparable11x11Param, [362](#)
 - ConvSeparable3x3Fp16ToFp16Param, [363](#)
 - ConvSeparable3x3Param, [364](#)
 - ConvSeparable5x5Fp16ToFp16Param, [364](#)
 - ConvSeparable5x5Param, [365](#)
 - ConvSeparable7x7Fp16ToFp16Param, [365](#)
 - ConvSeparable7x7Param, [366](#)
 - ConvSeparable9x9Fp16ToFp16Param, [366](#)
 - ConvSeparable9x9Param, [367](#)
- COMMAND_SIZEOF
 - svuStartBicubic.c, [967](#)
- CONCAT3
 - sippDefines.h, [632](#)

- CONCAT5
 - sippHwCommon_ma2x5x.h, [729](#)
- CONCAT7
 - sippHwCommon_ma2x5x.h, [729](#)
- CONV_CFG
 - sippHwBitfieldDefs.h, [682](#)
- CONV_EVENODD_LINE
 - sippHwBitfieldDefs.h, [684](#)
- CONV_EVENODD_PIX
 - sippHwBitfieldDefs.h, [684](#)
- CONV_KERNEL_SIZE
 - sippHwBitfieldDefs.h, [684](#)
- CONV_KSZ_3x3
 - sippHwBitfieldDefs.h, [684](#)
- CONV_KSZ_5x5
 - sippHwBitfieldDefs.h, [684](#)
- CONV_OUTPUT_CLAMP
 - sippHwBitfieldDefs.h, [684](#)
- CONV_OUTPUT_SQUARE
 - sippHwBitfieldDefs.h, [684](#)
- CT1
 - HPadInfoS, [390](#)
- CV, [206](#)
 - SHAVE_SYM_EXPORT, [208](#)
 - svuCrop, [208](#)
 - svuCropCvtPlaneMode, [208](#)
- canConsume
 - SippSchEntS, [461](#)
- canRunC
 - SippSchEntS, [461](#)
- canRunMask
 - SippPipelineS, [451](#)
- canRunP
 - SippSchEntS, [461](#)
- candPos
 - nonMax3x3Fp32Param, [409](#)
- canny
 - svuCannyEdgeDetection.c, [858](#)
- Canny Edge Detection, [147](#)
 - SHAVE_SYM_EXPORT, [147](#)
 - svuCannyEdgeDetection, [147](#)
- cannyEdgeDetection.h, [506](#)
- cannyEdgeDetectionParam, [336](#)
 - threshold1, [336](#)
 - threshold2, [336](#)
- ccOffs
 - ColCombParam, [342](#)
- ccm
 - ChromaBlkParam, [341](#)
 - ColCombParam, [342](#)
- censusMatching16, [148](#)
 - SHAVE_SYM_EXPORT, [148](#)
 - svuCensusMatching16, [148](#)
- censusMatching16.h, [506](#)
- censusMatching32, [149](#)
 - SHAVE_SYM_EXPORT, [149](#)
 - svuCensusMatching32, [149](#)
- censusMatching32.h, [507](#)
- CensusMatching32Param, [337](#)
 - flag, [337](#)
- censusMatching64, [150](#)
 - SHAVE_SYM_EXPORT, [150](#)
 - svuCensusMatching64, [150](#)
- censusMatching64.h, [507](#)
- CensusMatching64Param, [337](#)
 - flag, [337](#)
- censusMatching65, [151](#)
 - SHAVE_SYM_EXPORT, [151](#)
 - svuCensusMatching65, [151](#)
- censusMatching65.h, [508](#)
- censusMatchingPyr, [152](#)
 - SHAVE_SYM_EXPORT, [152](#)
 - svuCensusMatchingPyr, [152](#)
- censusMatchingPyr.h, [508](#)
- CensusMatchingPyrParam, [338](#)
 - predicted, [338](#)
- censusMin16, [153](#)
 - SHAVE_SYM_EXPORT, [153](#)
 - svuCensusMin16, [153](#)
- censusMin16.h, [509](#)
- censusMin64, [154](#)
 - SHAVE_SYM_EXPORT, [154](#)
 - svuCensusMin64, [154](#)
- censusMin64.h, [509](#)
- censusMin65, [155](#)
 - SHAVE_SYM_EXPORT, [155](#)
 - svuCensusMin65, [155](#)
- censusMin65.h, [510](#)
- censusMin7, [156](#)
 - SHAVE_SYM_EXPORT, [156](#)
 - svuCensusMin7, [156](#)
- censusMin7.h, [510](#)
- CensusTransform5x5, [157](#)
 - SHAVE_SYM_EXPORT, [157](#)
 - svuCensusTransform5x5, [157](#)
- censusTransform5x5.h, [511](#)
- cfg
 - ChrDnsParam, [339](#)
 - ColCombParam, [342](#)
 - ConvParam, [361](#)
 - DbyrParam, [370](#)
 - DogLtmParam, [377](#)

- EdgeParam, 378
- ExtStatsSatPixelsU32Param, 381
- GenChrParam, 384
- HarrisParam, 387
- LscParam, 393
- LutParam, 397
- MedParam, 399
- MipiRxParam, 403
- MipiTxParam, 406
- RawParam, 420
- SigmaParam, 424
- SippHwBufS, 444
- SippHwIOBuf, 445
- UsmParam, 485
- YDnsParam, 489
- cfgMipiRxLoopback
 - sippHwDefs_ma2x5x.h, 737
- cfgMipiTxLoopback
 - sippHwDefs_ma2x5x.h, 737
- cfgReg
 - PolyFirParam, 412
- chainLinked
 - SippMemRegionListNode, 447
- chainStart
 - SippMemRegionListNode, 447
- channelExtract, 158
 - SHAVE_SYM_EXPORT, 158
 - svuChannelExtract, 158
 - svuChannelExtract.c, 865
- channelExtract.h, 511
- ChannelExtractParam, 338
 - plane, 338
- chrCoefs
 - ChrDnsParam, 339
 - GenChrParam, 384
- ChrDnsParam, 339
 - cfg, 339
 - chrCoefs, 339
 - frmDim, 339
 - greyPt, 339
 - thr, 339
- ChrGenParam, 340
 - epsilon, 340
- ChrGenSSParam, 340
 - epsilon, 340
 - scale, 340
- Chroma Block, 159
 - SHAVE_SYM_EXPORT, 159
 - svuChromaBlock, 159
- Chroma Denoise, 102
- Chroma Generation, 111
- ChromaBlkParam, 341
 - ccm, 341
 - rangeLut, 341
- chromaBlock.h, 512
- clamp
 - PolyFirParam, 412
 - WhiteBalanceBayerGBRGParam, 486
 - WhiteBalanceRGBParam, 487
- clampInLines
 - padBayer5.h, 581
 - padBayer5Frame, 286
- clip
 - UsmParam, 485
- cmd
 - StartBicubicParam, 470
- cmxMapResult
 - SippPipelineS, 451
- cmxRegUsage, 341
 - cmxRegionIdx, 341
 - fullCmxSlice, 341
 - usedCount, 342
- cmxRegionIdx
 - cmxRegUsage, 341
- cmxSliceUsageBitMask
 - SippCmxBufferMapS, 431
- coef01
 - UsmParam, 485
- coef23
 - UsmParam, 485
- coefMat
 - XYGenParam, 488
- coefs
 - GenLumaU8Fp16Param, 385
 - PixelUnpackerMipi10bParam, 410
- coefsMat
 - cvtColorChromaNV12Param, 368
 - cvtColorLumaNV12Param, 369
- ColCombParam, 342
 - ccOffs, 342
 - ccm, 342
 - cfg, 342
 - frmDim, 342
 - krgb, 343
 - lutFormat, 343
 - threeDLut, 343
- Color Combination, 104
- combDecimDemosaicAwbGains.h
 - BAYER_FORMAT_BGGR, 513
 - BAYER_FORMAT_GBRG, 513
 - BAYER_FORMAT_GRBG, 513
 - BAYER_FORMAT_RGGB, 513

- CombDecimAwbGainsParam, [343](#)
 - bayerOrder, [343](#)
 - gains, [343](#)
- combDecimDemosaicAwbGains.h, [512](#)
 - BayerFormat, [513](#)
 - svuCombDecimDemosaicAwbGains, [513](#)
- combDecimDemosaicAwbGainsBG
 - svuCombDecimDemosaicAwbGains.c, [866](#)
- combDecimDemosaicAwbGainsGB
 - svuCombDecimDemosaicAwbGains.c, [866](#)
- combDecimDemosaicAwbGainsGR
 - svuCombDecimDemosaicAwbGains.c, [866](#)
- combDecimDemosaicAwbGainsRG
 - svuCombDecimDemosaicAwbGains.c, [866](#)
- combDecimDemosaicAwbGainsStats.c, [513](#)
 - combDecimDemosaicFinal, [513](#)
 - DisablePaxelSumMacro, [514](#)
 - emptyBuf, [514](#)
 - hzInterval, [514](#)
 - svuCombDecimDemosaicAwbGainsStats, [514](#)
- combDecimDemosaicAwbGainsStats.h, [514](#)
 - svuCombDecimDemosaicAwbGainsStats, [514](#)
- combDecimDemosaicFinal
 - combDecimDemosaicAwbGainsStats.c, [513](#)
- CombDecimStatsGainsParam, [343](#)
 - crtPaxelLine, [344](#)
 - gains, [344](#)
 - paxelsIntervalsHz, [344](#)
 - paxelsIntervalsVert, [344](#)
 - runNr, [344](#)
 - satPixelsStats, [344](#)
 - statsOutput, [344](#)
- CommInfo, [344](#)
 - curFrame, [345](#)
 - numShaves, [345](#)
 - pl, [345](#)
 - sliceFirst, [345](#)
 - sliceLast, [345](#)
 - sliceSize, [345](#)
- configureBicubicHWblock
 - svuStartBicubic.c, [967](#)
- cons
 - SippFilterS, [433](#)
 - SippSchEntS, [461](#)
- consIbufIdx
 - SippFilterS, [433](#)
 - SippOseS, [449](#)
 - SippSchEntS, [461](#)
- consSE
 - SippOseS, [449](#)
- consumptionLatency
 - sSchedIbufUsageInfoS, [465](#)
- Contrast, [160](#)
 - SHAVE_SYM_EXPORT, [160](#)
 - svuContrast, [160](#)
- contrast.h, [514](#)
- ContrastParam, [345](#)
 - idxLow, [346](#)
 - scale, [346](#)
- conv11x11.h, [515](#)
- Conv11x11Param, [346](#)
 - cMat, [346](#)
- conv15x1.h, [516](#)
- Conv15x1Param, [347](#)
 - cMat, [347](#)
- conv1x15.h, [516](#)
- Conv1x15Param, [347](#)
 - cMat, [348](#)
- conv1x5.h, [517](#)
- conv1x5Fp16ToFp16.h, [517](#)
- Conv1x5Fp16ToFp16Param, [348](#)
 - cMat, [348](#)
- Conv1x5Param, [349](#)
 - cMat, [349](#)
- conv1x7.h, [518](#)
- conv1x7Fp16ToFp16.h, [518](#)
- Conv1x7Fp16ToFp16Param, [349](#)
 - cMat, [350](#)
- Conv1x7Param, [350](#)
 - cMat, [350](#)
- conv1x9.h, [519](#)
- Conv1x9Param, [350](#)
 - cMat, [351](#)
- conv3x3.h, [520](#)
- conv3x3FilterImplementation
 - svuConv3x3.c, [872](#)
- conv3x3Fp16ToFp16.h, [520](#)
- Conv3x3Fp16ToFp16Param, [351](#)
 - cMat, [351](#)
- Conv3x3Param, [352](#)
 - cMat, [352](#)
- conv5x1.h, [521](#)
- conv5x1Fp16ToFp16.h, [521](#)
- Conv5x1Fp16ToFp16Param, [352](#)
 - cMat, [353](#)
- Conv5x1Param, [353](#)
 - cMat, [353](#)
- conv5x5.h, [522](#)
- conv5x5Fp16ToFp16.h, [522](#)
- Conv5x5Fp16ToFp16Param, [354](#)

- cMat, [354](#)
- Conv5x5Param, [354](#)
 - cMat, [355](#)
- conv7x1.h, [523](#)
- conv7x1Fp16ToFp16.h, [524](#)
- Conv7x1Fp16ToFp16Param, [355](#)
 - cMat, [355](#)
- Conv7x1Param, [355](#)
 - cMat, [356](#)
- conv7x7.h, [524](#)
- conv7x7Fp16ToFp16.h, [525](#)
- conv7x7Fp16ToU8.h, [525](#)
- Conv7x7Param, [356](#)
 - cMat, [356](#)
- Conv7x7ParamFp16ToFp16, [357](#)
 - cMat, [357](#)
- Conv7x7ParamFp16ToU8, [357](#)
 - cMat, [358](#)
- conv9x1.h, [526](#)
- Conv9x1Param, [358](#)
 - cMat, [358](#)
- conv9x9.h, [526](#)
- conv9x9Fp16ToFp16.h, [527](#)
- Conv9x9Fp16ToFp16Param, [359](#)
 - cMat, [359](#)
- Conv9x9Param, [359](#)
 - cMat, [360](#)
- convGeneric.h, [531](#)
- ConvGenericParam, [360](#)
 - cMat, [360](#)
 - filterSize, [360](#)
- ConvParam, [361](#)
 - cfg, [361](#)
 - frmDim, [361](#)
 - kernel, [361](#)
 - shadowKernel, [361](#)
- convSeparable11x11
 - svuConvSeparable11x11.c, [884](#)
- convSeparable11x11.h, [531](#)
- convSeparable11x11Fp16ToFp16
 - svuConvSeparable11x11Fp16ToFp16.c, [885](#)
- convSeparable11x11Fp16ToFp16.h, [532](#)
- ConvSeparable11x11Fp16ToFp16Param, [362](#)
 - cMat, [362](#)
- ConvSeparable11x11Param, [362](#)
 - cMat, [362](#)
- convSeparable3x3
 - svuConvSeparable3x3.c, [885](#)
- convSeparable3x3.h, [532](#)
- convSeparable3x3Fp16ToFp16
 - svuConvSeparable3x3Fp16ToFp16.c, [886](#)
- convSeparable3x3Fp16ToFp16.h, [533](#)
- ConvSeparable3x3Fp16ToFp16Param, [363](#)
 - cMat, [363](#)
- ConvSeparable3x3Param, [363](#)
 - cMat, [364](#)
- convSeparable5x5
 - svuConvSeparable5x5.c, [886](#)
- convSeparable5x5.h, [533](#)
- convSeparable5x5Fp16ToFp16
 - svuConvSeparable5x5Fp16ToFp16.c, [887](#)
- convSeparable5x5Fp16ToFp16.h, [534](#)
- ConvSeparable5x5Fp16ToFp16Param, [364](#)
 - cMat, [364](#)
- ConvSeparable5x5Param, [364](#)
 - cMat, [365](#)
- convSeparable7x7
 - svuConvSeparable7x7.c, [888](#)
- convSeparable7x7.h, [535](#)
- convSeparable7x7Fp16ToFp16
 - svuConvSeparable7x7Fp16ToFp16.c, [888](#)
- convSeparable7x7Fp16ToFp16.h, [535](#)
- ConvSeparable7x7Fp16ToFp16Param, [365](#)
 - cMat, [365](#)
- ConvSeparable7x7Param, [365](#)
 - cMat, [366](#)
- convSeparable9x9
 - svuConvSeparable9x9.c, [889](#)
- convSeparable9x9.h, [536](#)
- convSeparable9x9Fp16ToFp16
 - svuConvSeparable9x9Fp16ToFp16.c, [889](#)
- convSeparable9x9Fp16ToFp16.h, [536](#)
- ConvSeparable9x9Fp16ToFp16Param, [366](#)
 - cMat, [366](#)
- ConvSeparable9x9Param, [366](#)
 - cMat, [367](#)
- convYuv444.h, [537](#)
- Convert 16bpp To 8bpp, [183](#)
 - SHAVE_SYM_EXPORT, [183](#)
 - svuConvert16bppTo8bpp, [183](#)
- Convert F16 To U8, [184](#)
 - SHAVE_SYM_EXPORT, [184](#)
 - svuConvertF16ToU8, [184](#)
- Convert Fp16 to U16, [186](#)
 - SHAVE_SYM_EXPORT, [186](#)
 - svuConvertPFp16U16, [186](#)
- Convert to YUV444, [201](#)
 - SHAVE_SYM_EXPORT, [201](#)
 - svuRgbYuv444, [201](#)
- Convert U16 to Fp16, [187](#)
 - SHAVE_SYM_EXPORT, [187](#)
 - svuConvertPU16Fp16, [187](#)

Convert U8 To F16, [188](#)
 SHAVE_SYM_EXPORT, [188](#)
 svuConvertU8ToF16, [188](#)
 convert16bppTo8bpp.h, [528](#)
 convertF16ToU8
 svuConvertF16ToU8.c, [880](#)
 convertF16ToU8.h, [528](#)
 convertFrom12BppTo8Bpp.h, [528](#)
 convertPFp16U16.h, [529](#)
 convertPU16Fp16.h, [529](#)
 convertU8ToF16
 svuConvertU8ToF16.c, [882](#)
 convertU8ToF16.h, [530](#)
 ConvertYUV400ToYUV422
 svuConvertYUV400ToYUV422.c, [883](#)
 convertYUV400ToYUV422.h, [530](#)
 Convolution, [105](#)
 svuConvGeneric.c, [884](#)
 Convolution 11x11, [161](#)
 SHAVE_SYM_EXPORT, [161](#)
 svuConv11x11, [161](#)
 Convolution 15x1, [162](#)
 SHAVE_SYM_EXPORT, [162](#)
 svuConv15x1, [162](#)
 Convolution 1x15, [163](#)
 SHAVE_SYM_EXPORT, [163](#)
 svuConv1x15, [163](#)
 Convolution 1x5, [164](#)
 SHAVE_SYM_EXPORT, [164](#)
 svuConv1x5, [164](#)
 Convolution 1x5 Fp16ToFp16, [165](#)
 SHAVE_SYM_EXPORT, [165](#)
 svuConv1x5Fp16ToFp16, [165](#)
 Convolution 1x7, [166](#)
 SHAVE_SYM_EXPORT, [166](#)
 svuConv1x7, [166](#)
 Convolution 1x7 Fp16ToFp16, [167](#)
 SHAVE_SYM_EXPORT, [167](#)
 svuConv1x7Fp16ToFp16, [167](#)
 Convolution 1x9, [168](#)
 SHAVE_SYM_EXPORT, [168](#)
 svuConv1x9, [168](#)
 Convolution 3x3, [169](#)
 SHAVE_SYM_EXPORT, [169](#)
 svuConv3x3, [169](#)
 Convolution 3x3 Fp16ToFp16, [170](#)
 SHAVE_SYM_EXPORT, [170](#)
 svuConv3x3Fp16ToFp16, [170](#)
 Convolution 5x1, [171](#)
 SHAVE_SYM_EXPORT, [171](#)
 svuConv5x1, [171](#)
 Convolution 5x1 Fp16ToFp16, [172](#)
 SHAVE_SYM_EXPORT, [172](#)
 svuConv5x1Fp16ToFp16, [172](#)
 Convolution 5x5, [173](#)
 SHAVE_SYM_EXPORT, [173](#)
 svuConv5x5, [173](#)
 Convolution 5x5 Fp16ToFp16, [174](#)
 SHAVE_SYM_EXPORT, [174](#)
 svuConv5x5Fp16ToFp16, [174](#)
 Convolution 7x1, [175](#)
 SHAVE_SYM_EXPORT, [175](#)
 svuConv7x1, [175](#)
 Convolution 7x1 Fp16ToFp16, [176](#)
 SHAVE_SYM_EXPORT, [176](#)
 svuConv7x1Fp16ToFp16, [176](#)
 Convolution 7x7, [177](#)
 SHAVE_SYM_EXPORT, [177](#)
 svuConv7x7, [177](#)
 Convolution 7x7 Fp16ToFp16, [178](#)
 SHAVE_SYM_EXPORT, [178](#)
 svuConv7x7Fp16ToFp16, [178](#)
 Convolution 7x7 Fp16ToU8, [179](#)
 SHAVE_SYM_EXPORT, [179](#)
 svuConv7x7Fp16ToU8, [179](#)
 Convolution 9x1, [180](#)
 SHAVE_SYM_EXPORT, [180](#)
 svuConv9x1, [180](#)
 Convolution 9x9, [181](#)
 SHAVE_SYM_EXPORT, [181](#)
 svuConv9x9, [181](#)
 Convolution 9x9 Fp16ToFp16, [182](#)
 SHAVE_SYM_EXPORT, [182](#)
 svuConv9x9Fp16ToFp16, [182](#)
 Convolution Separable 11x11, [191](#)
 SHAVE_SYM_EXPORT, [191](#)
 svuConvSeparable11x11, [191](#)
 Convolution Separable 11x11 Fp16ToFp16, [192](#)
 SHAVE_SYM_EXPORT, [192](#)
 svuConvSeparable11x11Fp16ToFp16, [192](#)
 Convolution Separable 3x3, [193](#)
 SHAVE_SYM_EXPORT, [193](#)
 svuConvSeparable3x3, [193](#)
 Convolution Separable 3x3 Fp16ToFp16, [194](#)
 SHAVE_SYM_EXPORT, [194](#)
 svuConvSeparable3x3Fp16ToFp16, [194](#)
 Convolution Separable 5x5, [195](#)
 SHAVE_SYM_EXPORT, [195](#)
 svuConvSeparable5x5, [195](#)
 Convolution Separable 5x5 Fp16ToFp16, [196](#)
 SHAVE_SYM_EXPORT, [196](#)
 svuConvSeparable5x5Fp16ToFp16, [196](#)

- Convolution Separable 7x7, [197](#)
 - SHAVE_SYM_EXPORT, [197](#)
 - svuConvSeparable7x7, [197](#)
- Convolution Separable 7x7 Fp16ToFp16, [198](#)
 - SHAVE_SYM_EXPORT, [198](#)
 - svuConvSeparable7x7Fp16ToFp16, [198](#)
- Convolution Separable 9x9, [199](#)
 - SHAVE_SYM_EXPORT, [199](#)
 - svuConvSeparable9x9, [199](#)
- Convolution Separable 9x9 Fp16ToFp16, [200](#)
 - SHAVE_SYM_EXPORT, [200](#)
 - svuConvSeparable9x9Fp16ToFp16, [200](#)
- Convolution11x11
 - svuConv11x11.c, [867](#)
- Convolution15x1
 - svuConv15x1.c, [868](#)
- Convolution1x15
 - svuConv1x15.c, [868](#)
- Convolution1x5
 - svuConv1x5.c, [869](#)
- Convolution1x5Fp16ToFp16
 - svuConv1x5Fp16ToFp16.c, [870](#)
- Convolution1x7
 - svuConv1x7.c, [870](#)
- Convolution1x7Fp16ToFp16
 - svuConv1x7Fp16ToFp16.c, [871](#)
- Convolution1x9
 - svuConv1x9.c, [871](#)
- convolution3x3Fp16ToFp16Implementation
 - svuConv3x3Fp16ToFp16.c, [872](#)
- Convolution5x1
 - svuConv5x1.c, [873](#)
- Convolution5x1Fp16ToFp16
 - svuConv5x1Fp16ToFp16.c, [874](#)
- Convolution5x5
 - svuConv5x5.c, [874](#)
- Convolution5x5Fp16ToFp16
 - svuConv5x5Fp16ToFp16.c, [875](#)
- Convolution7x1
 - svuConv7x1.c, [875](#)
- convolution7x1Fp16ToFp16
 - svuConv7x1Fp16ToFp16.c, [876](#)
- convolution7x7
 - svuConv7x7.c, [876](#)
- convolution7x7Fp16ToFp16
 - svuConv7x7Fp16ToFp16.c, [877](#)
- convolution7x7Fp16ToU8
 - svuConv7x7Fp16ToU8.c, [878](#)
- Convolution9x1
 - svuConv9x1.c, [878](#)
- Convolution9x9
 - svuConv9x9.c, [879](#)
- Convolution9x9Fp16ToFp16
 - svuConv9x9Fp16ToFp16.c, [879](#)
- Copy, [202](#)
 - SHAVE_SYM_EXPORT, [202](#)
 - svuCopy, [202](#)
- copy.h, [537](#)
- Corner Min Eigenvalue, [203](#)
 - SHAVE_SYM_EXPORT, [203](#)
 - svuCornerMinEigenVal, [203](#)
- Corner Min Eigenvalue Patched, [204](#)
 - SHAVE_SYM_EXPORT, [204](#)
 - svuCornerMinEigenValpatched, [204](#)
- CornerMinEigenVal
 - svuCornerMinEigenVal.c, [891](#)
- cornerMinEigenVal.h, [538](#)
- CornerMinEigenVal_patched
 - svuCornerMinEigenValpatched.c, [892](#)
- cornerMinEigenValpatched.h, [538](#)
- count
 - ae_patch_stats, [330](#)
- countBit
 - svuHammingDistance.c, [918](#)
- createTripleConv3x3
 - tripleConv3x3.c, [977](#)
 - tripleConv3x3.h, [977](#)
- Crop, [205](#)
- crop.h, [539](#)
- cropCvtPlaneMode
 - svuCropCvtPlaneMode.c, [893](#)
- cropCvtPlaneMode.h, [539](#)
- CropParam, [367](#)
 - st_Y, [367](#)
- crtPaxel
 - ExtStatsSatPixelsU32Param, [381](#)
 - StatsAwbSatPixelsParam, [471](#)
 - StatsAwbSatPixelsParamU32, [473](#)
- crtPaxelLine
 - CombDecimStatsGainsParam, [344](#)
- crtPosInPaxel
 - ExtStatsSatPixelsU32Param, [381](#)
 - StatsAwbSatPixelsParam, [471](#)
 - StatsAwbSatPixelsParamU32, [473](#)
- ctx
 - ExtStatsSatPixelsU32Param, [381](#)
 - SippHwBufS, [444](#)
 - SippHwIOBuf, [445](#)
- ctxSwitchChromaDns
 - sippInternal.h, [757](#)
- ctxSwitchColComb
 - sippInternal.h, [757](#)

- ctxSwitchLut
 - sippInternal.h, [757](#)
- ctxSwitchMipiRx
 - sippInternal.h, [757](#)
- ctxSwitchMipiTx
 - sippInternal.h, [757](#)
- ctxSwitchOnePar
 - sippInternal.h, [757](#)
- ctxSwitchPoly
 - sippInternal.h, [757](#)
- ctxSwitchTwoPar
 - sippInternal.h, [757](#)
- cum_hist
 - EqualizeHistParam, [379](#)
- curFrame
 - CommInfo, [345](#)
- currKSOffset
 - SippSchEntS, [461](#)
- curves
 - LocalTMPParam, [391](#)
- cvtColorChromaNV12Param, [367](#)
 - coefsMat, [368](#)
 - offset, [368](#)
- cvtColorChromaYUV420ToNV12
 - svuCvtColorChromaYUVToNV12.c, [893](#)
- cvtColorChromaYUV444ToNV12
 - svuCvtColorChromaYUVToNV12.c, [893](#)
- cvtColorChromaYUVToNV12.h, [540](#)
 - svuCvtColorChromaYUVToNV12, [540](#)
- CvtColorChromaYUVToNV12Param, [368](#)
 - inputFrameType, [368](#)
 - needs2Parents, [368](#)
- cvtColorKernelRGBToYUV422
 - svuCvtColorRGBToYUV422.c, [900](#)
- cvtColorKernelYUV422ToRGB
 - svuCvtColorYUV422ToRGB.c, [901](#)
- cvtColorKernelYUVToRGB
 - svuCvtColorYUVToRGB.c, [901](#)
- cvtColorLumaNV12Param, [368](#)
 - coefsMat, [369](#)
 - offset, [369](#)
- cvtColorNV21toRGB.h, [541](#)
- cvtColorNV21toRGBImplementation
 - svucvtColorNV21toRGB.c, [894](#)
- cvtColorRGBToYUV422.h, [544](#)
- cvtColorRGBfp16ToLumaU8
 - svuCvtColorRGBfp16ToLumaU8.c, [895](#)
- cvtColorRGBfp16ToLumaU8.h, [541](#)
- cvtColorRGBfp16ToUV420U8
 - svuCvtColorRGBfp16ToUV420U8.c, [895](#)
- cvtColorRGBfp16ToUV420U8.h, [541](#)
- cvtColorRGBtoChromaNV12
 - svuCvtColorRGBtoChromaNV12.c, [896](#)
- cvtColorRGBtoChromaNV12.h, [542](#)
- cvtColorRGBtoLuma
 - svuCvtColorRGBtoLuma.c, [897](#)
- cvtColorRGBtoLuma.h, [542](#)
- cvtColorRGBtoLumaNV12
 - svuCvtColorRGBtoLumaNV12.c, [897](#)
- cvtColorRGBtoLumaNV12.h, [543](#)
- cvtColorRGBtoUV
 - svuCvtColorRGBtoUV.c, [898](#)
- cvtColorRGBtoUV.h, [543](#)
- cvtColorRGBtoUV420
 - svuCvtColorRGBtoUV420.c, [898](#)
- cvtColorRGBtoUV420.h, [544](#)
- cvtColorYUV422ToRGB.h, [545](#)
- cvtColorYUVToRGB.h, [545](#)
- cx
 - UndistortBParam, [483](#)
- cy
 - UndistortBParam, [483](#)
- DDR_ADDRESS
 - sippHwCommon_ma2x5x.h, [734](#)
- DDR_MIRRORED_ADDRESS
 - sippHwCommon_ma2x5x.h, [734](#)
- DMA_MODE_PARTIAL_LINE
 - sippTypes.h, [830](#)
- DMA_MODE_STANDARD
 - sippTypes.h, [830](#)
- DBG_PRINT
 - sippPlatform_ma2x5x.h, [809](#)
- DBYR_KERNEL_SIZE
 - sippHwCommon_ma2x5x.h, [729](#)
- DDR_BSS
 - sippPlatform_ma2x5x.h, [809](#)
- DDR_DATA
 - sippPlatform_ma2x5x.h, [809](#)
- DDR_RODATA
 - sippPlatform_ma2x5x.h, [809](#)
- DDR_TEXT
 - sippPlatform_ma2x5x.h, [809](#)
- DEBAYER_ABS_THRESH1
 - sippHwBitfieldDefs.h, [684](#)
- DEBAYER_ABS_THRESH2
 - sippHwBitfieldDefs.h, [684](#)
- DEBAYER_CFG
 - sippHwBitfieldDefs.h, [684](#)
- DEBAYER_DEWORM
 - sippHwBitfieldDefs.h, [685](#)
- DEBAYER_LUMA_EN
 - sippHwBitfieldDefs.h, [686](#)

- DEBAYER_LUMA_ONLY
 - sippHwBitfieldDefs.h, [686](#)
- DEBAYER_OFFSET
 - sippHwBitfieldDefs.h, [686](#)
- DEBAYER_PREVIEW_EN
 - sippHwBitfieldDefs.h, [687](#)
- DEBAYER_RGB_EN
 - sippHwBitfieldDefs.h, [687](#)
- DEBAYER_SLOPE
 - sippHwBitfieldDefs.h, [687](#)
- DEBAYER_THRESH
 - sippHwBitfieldDefs.h, [687](#)
- DEF_SLICE_SIZE
 - sippHwCommon_ma2x5x.h, [730](#)
- DEFAULT
 - sippHwCommon_ma2x5x.h, [730](#)
- DISABLED
 - sippHwCommon_ma2x5x.h, [730](#)
- DMA, [328](#)
- DMA_TASK_LIST_SZ
 - sippCmxDmaIf.c, [621](#)
- DMACfgPlaceholder
 - DMAExtCfg, [374](#)
- DMAExtCfg, [374](#)
 - DMACfgPlaceholder, [374](#)
 - tPartialCfg, [374](#)
- DMAPartialCfg, [376](#)
 - ddrLineStride, [376](#)
- DMAWaitCycles
 - tRTStats, [479](#)
- dMat
 - Dilate3x3Param, [370](#)
 - Dilate5x5Param, [371](#)
 - Dilate7x7Param, [371](#)
 - DilateGenericParam, [372](#)
- DOGL_CFG_MODE
 - sippHwBitfieldDefs.h, [688](#)
- DOGL_CFG_SET
 - sippHwBitfieldDefs.h, [688](#)
- DOGL_MODE_DOG_LTM
 - sippHwBitfieldDefs.h, [689](#)
- DYNAMIC_IRQ_SIPP_0
 - sippCfg.h, [617](#)
- DYNAMIC_IRQ_SIPP_1
 - sippCfg.h, [618](#)
- DYNAMIC_IRQ_SIPP_2
 - sippCfg.h, [618](#)
- data
 - SippHeapCB, [444](#)
- dataFormat
 - BoxFilterParam, [336](#)
 - LscParam, [393](#)
- dataWidth
 - LscParam, [393](#)
- dbLineOut
 - SippFilterS, [433](#)
- dbLinesIn
 - SippFilterS, [433](#)
- dbg_svu_no
 - sippHpad.c, [650](#)
 - sippIoPtrs.c, [769](#)
 - sippShave.c, [819](#)
- dbgJustRoll
 - SippSchEntS, [462](#)
- dbgLevel
 - SippPipelineS, [452](#)
- DbyrParam, [369](#)
 - cfg, [370](#)
 - dewormCfg, [370](#)
 - frmDim, [370](#)
 - lumaWeight, [370](#)
 - thresh, [370](#)
- ddrAddr
 - DmaParam, [375](#)
- ddrCmxBackupAdr
 - SippPipelineS, [452](#)
- ddrCmxBackupLen
 - SippPipelineS, [452](#)
- ddrLineStride
 - DMAPartialCfg, [376](#)
- Debayer, [99](#)
- descriptor_size
 - HammingDistanceParam, [386](#)
- dewormCfg
 - DbyrParam, [370](#)
- Dilate 3x3, [225](#)
 - SHAVE_SYM_EXPORT, [225](#)
 - svuDilate3x3, [225](#)
- Dilate 5x5, [226](#)
 - SHAVE_SYM_EXPORT, [226](#)
 - svuDilate5x5, [226](#)
- Dilate 7x7, [227](#)
 - SHAVE_SYM_EXPORT, [227](#)
 - svuDilate7x7, [227](#)
- Dilate3x3
 - svuDilate3x3.c, [902](#)
- dilate3x3.h, [546](#)
- Dilate3x3Param, [370](#)
 - dMat, [370](#)
- Dilate5x5
 - svuDilate5x5.c, [902](#)
- dilate5x5.h, [546](#)

- Dilate5x5Param, [371](#)
 - dMat, [371](#)
- Dilate7x7
 - svuDilate7x7.c, [903](#)
- dilate7x7.h, [547](#)
- Dilate7x7Param, [371](#)
 - dMat, [371](#)
- DilateGeneric
 - svuDilateGeneric.c, [903](#)
- dilateGeneric.h, [547](#)
- DilateGenericParam, [372](#)
 - dMat, [372](#)
 - kernelSize, [372](#)
- DisablePaxelSumMacro
 - combDecimDemosaicAwbGainsStats.c, [514](#)
- disp2depth.h, [548](#)
 - SHAVE_SYM_EXPORT, [548](#)
 - svudisp2depth, [548](#)
- disp2depth_exec.c, [549](#)
 - dsp2depth_explic_vect, [549](#)
 - flip_dsp2depth_explic_vect, [549](#)
 - flip_mvcvDisp2depth, [549](#)
 - flip_mvcvDisp2depth16bit, [549](#)
 - mvcvDisp2depth, [549](#)
 - mvcvDisp2depth16bit, [549](#)
- disp2depth_exec.h, [549](#)
 - dsp2depth_explic_vect, [550](#)
 - flip_dsp2depth_explic_vect, [550](#)
 - flip_mvcvDisp2depth, [550](#)
 - flip_mvcvDisp2depth16bit, [550](#)
 - mvcvDisp2depth, [550](#)
 - mvcvDisp2depth16bit, [550](#)
- Disp2depthParam, [372](#)
 - _16bitmode, [373](#)
 - ALIGNED, [373](#)
 - flip_disp2depth, [373](#)
- distCfg
 - YDnsParam, [489](#)
- distOffsets
 - YDnsParam, [489](#)
- dmaCmxPop
 - SippPipelineS, [452](#)
- dmaCmxPush
 - SippPipelineS, [452](#)
- DmaDesc, [373](#)
 - dscCtrlLinkAddr, [373](#)
 - dscDstSrcAddr, [373](#)
 - dscDstStrdWidth, [373](#)
 - dscPIStrides, [373](#)
 - dscPlanesLen, [373](#)
 - dscSrcStrdWidth, [374](#)
- dmaDsc
 - DmaParam, [375](#)
- dmaIdle
 - sippCmxDmaIf.c, [623](#)
- dmaKickSequence
 - sippCmxDmaIf.c, [621](#)
- dmaKickSequenceCQ
 - sippCmxDmaIf.c, [621](#)
- dmaKickSequenceConcurrent
 - sippCmxDmaIf.c, [621](#)
 - sippInternal.h, [757](#)
- dmaMask
 - SchedInfoS, [421](#)
- dmaMode
 - DmaParam, [375](#)
- DmaParam, [374](#)
 - bChunked, [375](#)
 - ddrAddr, [375](#)
 - dmaDsc, [375](#)
 - dmaMode, [375](#)
 - dstLnS, [375](#)
 - dstPLS, [375](#)
 - extCfg, [375](#)
 - pLineDesList, [376](#)
 - srcLnS, [376](#)
- DmaTaskList, [376](#)
 - rPtr, [376](#)
 - taskPI, [376](#)
 - wPtr, [377](#)
- dmaTaskList
 - sippCmxDmaIf.c, [623](#)
 - SippPipelineS, [452](#)
- DoG LTM, [112](#)
- dogCoeffs11
 - DogLtmParam, [377](#)
- dogCoeffs15
 - DogLtmParam, [377](#)
- DogLtmParam, [377](#)
 - cfg, [377](#)
 - dogCoeffs11, [377](#)
 - dogCoeffs15, [377](#)
 - dogStrength, [377](#)
 - frmDim, [377](#)
 - ltmCurves, [377](#)
- dogStrength
 - DogLtmParam, [377](#)
- Downscale by 2, [298](#)
 - SHAVE_SYM_EXPORT, [298](#)
 - svuScI05BilinHV, [298](#)
- Downscale by 2 (fp16/fp16), [300](#)
 - SHAVE_SYM_EXPORT, [300](#)

svuScale05BilinHVFp16, 300
 Downscale by 2 (fp16/u8), 299
 SHAVE_SYM_EXPORT, 299
 svuScale05BilinHV_Fp16U8, 299
 dscCtrlLinkAddr
 DmaDesc, 373
 dscDstSrcAddr
 DmaDesc, 373
 dscDstStrdWidth
 DmaDesc, 373
 dscPlStrides
 DmaDesc, 373
 dscPlanesLen
 DmaDesc, 373
 dscSrcStrdWidth
 DmaDesc, 374
 dsp2depth_explic_vect
 disp2depth_exec.c, 549
 disp2depth_exec.h, 550
 svudisp2depth.c, 904
 dstLeftO
 HorizPaddingOffS, 389
 dstLnS
 DmaParam, 375
 dstPlS
 DmaParam, 375
 dstRightO
 HorizPaddingOffS, 389
 dynIrqSipp0
 tSippFramework, 479
 dynIrqSipp1
 tSippFramework, 479
 dynIrqSipp2
 tSippFramework, 479

 E_BLOCK_CALL_REJECTED
 sippTypes.h, 829
 E_CANNOT_FINISH_FILTER
 sippTypes.h, 829
 E_CDMA_QU_OVERFLOW
 sippTypes.h, 829
 E_DATA_ALIGN
 sippTypes.h, 829
 E_DATA_NOT_FOUND
 sippTypes.h, 829
 E_FINALISE_FAIL
 sippTypes.h, 830
 E_HEAP_CREATION_FAIL
 sippTypes.h, 830
 E_INVALID_HW_PARAM
 sippTypes.h, 829
 E_INVALID_MEM_P
 sippTypes.h, 829
 E_INVLD_FILT_FIRST_SLICE
 sippTypes.h, 829
 E_INVLD_FILT_LAST_SLICE
 sippTypes.h, 829
 E_INVLD_HW_ID
 sippTypes.h, 829
 E_INVLD_MIPI_RX_LOOPBACK
 sippTypes.h, 829
 E_INVLD_MULTI_INSTANCE
 sippTypes.h, 829
 E_INVLD_SLICE_WIDTH
 sippTypes.h, 829
 E_LAST
 sippTypes.h, 830
 E_MISSING_SHAVE_IMAGE
 sippTypes.h, 829
 E_OPT_EXEC_NUM
 sippTypes.h, 829
 E_OSE_CREATION_ERROR
 sippTypes.h, 829
 E_OUT_OF_MEM
 sippTypes.h, 829
 E_PAR_NOT_FOUND
 sippTypes.h, 829
 E_PC_CMX_MEM_ALLOC_ERR
 sippTypes.h, 829
 E_PC_RUNTIME_FAILURE
 sippTypes.h, 829
 E_PRECOMP_SCHED
 sippTypes.h, 829
 E_RUN_DON_T_KNOW
 sippTypes.h, 829
 E_RUNS_ITER_GROUPS
 sippTypes.h, 829
 E_SCHEDULING_OVF
 sippTypes.h, 829
 E_SUCCESS
 sippTypes.h, 829
 E_TOO_MANY_CONSUMERS
 sippTypes.h, 829
 E_TOO_MANY_DMAS
 sippTypes.h, 829
 E_TOO_MANY_FILTERS
 sippTypes.h, 829
 E_TOO_MANY_PARENTS
 sippTypes.h, 829
 E_UNIMPLEMENTED_FEAT
 sippTypes.h, 829
 eBayerOrderBGGR
 sippHwBitfieldDefs.h, 723

eBayerOrderGBRG
 sippHwBitfieldDefs.h, 723
 eBayerOrderGRBG
 sippHwBitfieldDefs.h, 723
 eBayerOrderRGGB
 sippHwBitfieldDefs.h, 723
 eRawFormatBayer
 sippHwBitfieldDefs.h, 724
 eRawFormatPlanar
 sippHwBitfieldDefs.h, 724
 eSIPP_ACCESS_SCHEDULER_CMD_FINALI-
 SE_PIPE
 sippAccessSchedulerTypes.h, 610
 eSIPP_ACCESS_SCHEDULER_CMD_HWINIT
 sippAccessSchedulerTypes.h, 610
 eSIPP_ACCESS_SCHEDULER_CMD_PROCE-
 SS_ITERS
 sippAccessSchedulerTypes.h, 610
 eSIPP_ACCESS_SCHEDULER_CMD_RESCH-
 EDULE_PIPE
 sippAccessSchedulerTypes.h, 610
 eSIPP_ACCESS_SCHEDULER_CMD_RESET
 sippAccessSchedulerTypes.h, 610
 eSIPP_ACCESS_SCHEDULER_CMD_TEARD-
 OWN_PIPE
 sippAccessSchedulerTypes.h, 610
 eSIPP_ACCESS_SCHEDULER_NULL
 sippAccessSchedulerTypes.h, 610
 eSIPP_ACCESS_SCHEDULER_PIPE_UPDAT-
 E_STATUS
 sippAccessSchedulerTypes.h, 610
 eSIPP_CMD_DELETE_PIPE_SW
 sippTypesPrivate.h, 834
 eSIPP_CMD_FINALISE_PIPE_SW
 sippTypesPrivate.h, 834
 eSIPP_CMD_INIT_HW
 sippTypesPrivate.h, 834
 eSIPP_CMD_PROCESS_ITERS_HW
 sippTypesPrivate.h, 835
 eSIPP_CMD_RESCHEDULE_PIPE_SW
 sippTypesPrivate.h, 834
 eSIPP_HW_ACTIVE
 sippTypesPrivate.h, 834
 eSIPP_HW_INACTIVE
 sippTypesPrivate.h, 834
 eSIPP_PIPE_ACTIVE
 sippTypesPrivate.h, 835
 eSIPP_PIPE_END_SESSION
 sippTypesPrivate.h, 835
 eSIPP_PIPE_WAIT_FINALISE
 sippTypesPrivate.h, 835
 eSIPP_PIPE_WAIT_INIT
 sippTypesPrivate.h, 835
 eSIPP_PIPELINE_FINALISED
 sippEvents.h, 637
 eSIPP_PIPELINE_FRAME_DONE
 sippEvents.h, 637
 eSIPP_PIPELINE_ITERS_DONE
 sippEvents.h, 637
 eSIPP_PIPELINE_RESCHEDULED
 sippEvents.h, 637
 eSIPP_PIPELINE_STARTED
 sippEvents.h, 637
 eSIPP_PIPELINE_SYNC_OP_DONE
 sippEvents.h, 637
 eSIPP_SET_OBUF_SPACE_EMPTY
 sippTypesPrivate.h, 835
 eSIPP_SET_OBUF_SPACE_FULL
 sippTypesPrivate.h, 835
 eSIPP_SET_OBUF_SPACE_ITER
 sippTypesPrivate.h, 835
 eSIPP_STATUS_ALREADY_ATTACHED
 sippTypes.h, 831
 eSIPP_STATUS_ALREADY_DONE
 sippTypes.h, 831
 eSIPP_STATUS_ALREADY_INIT
 sippTypes.h, 830
 eSIPP_STATUS_ASLEEP
 sippTypes.h, 831
 eSIPP_STATUS_BAD_ATTACHMENT
 sippTypes.h, 831
 eSIPP_STATUS_BAD_COMMAND
 sippTypes.h, 831
 eSIPP_STATUS_BAD_HANDLE
 sippTypes.h, 830
 eSIPP_STATUS_BAD_LENGTH
 sippTypes.h, 830
 eSIPP_STATUS_BAD_PARAMETER
 sippTypes.h, 830
 eSIPP_STATUS_BAD_UNIT
 sippTypes.h, 830
 eSIPP_STATUS_BUSY
 sippTypes.h, 830
 eSIPP_STATUS_CANCELLED
 sippTypes.h, 830
 eSIPP_STATUS_CLOSED_HANDLE
 sippTypes.h, 830
 eSIPP_STATUS_CODE_LAST
 sippTypes.h, 831
 eSIPP_STATUS_COMPLETE
 sippTypes.h, 831
 eSIPP_STATUS_DESTROYED

[sippTypes.h, 830](#)
[eSIPP_STATUS_DISCONNECTED](#)
[sippTypes.h, 830](#)
[eSIPP_STATUS_DUPLICATE_NODE](#)
[sippTypes.h, 831](#)
[eSIPP_STATUS_EMPTY](#)
[sippTypes.h, 830](#)
[eSIPP_STATUS_FAILURE](#)
[sippTypes.h, 831](#)
[eSIPP_STATUS_FULL](#)
[sippTypes.h, 831](#)
[eSIPP_STATUS_HARDWARE_NOT_FOUND](#)
[sippTypes.h, 831](#)
[eSIPP_STATUS_ILLEGAL_OPERATION](#)
[sippTypes.h, 831](#)
[eSIPP_STATUS_IN_USE](#)
[sippTypes.h, 830](#)
[eSIPP_STATUS_INCOMPATIBLE_FORMATS](#)
[sippTypes.h, 831](#)
[eSIPP_STATUS_INT_HANDLED](#)
[sippTypes.h, 831](#)
[eSIPP_STATUS_INT_NOT_HANDLED](#)
[sippTypes.h, 831](#)
[eSIPP_STATUS_INTERNAL_ERROR](#)
[sippTypes.h, 830](#)
[eSIPP_STATUS_INVALID](#)
[sippTypes.h, 830](#)
[eSIPP_STATUS_INVALID_DEVICE](#)
[sippTypes.h, 831](#)
[eSIPP_STATUS_INVALID_EDGE](#)
[sippTypes.h, 831](#)
[eSIPP_STATUS_INVALID_NODE](#)
[sippTypes.h, 831](#)
[eSIPP_STATUS_INVALID_NUMBER](#)
[sippTypes.h, 831](#)
[eSIPP_STATUS_INVALID_STATE](#)
[sippTypes.h, 831](#)
[eSIPP_STATUS_INVALID_TYPE](#)
[sippTypes.h, 831](#)
[eSIPP_STATUS_NOT_ATTACHED](#)
[sippTypes.h, 830](#)
[eSIPP_STATUS_NOT_AVAILABLE](#)
[sippTypes.h, 830](#)
[eSIPP_STATUS_NOT_COMPATIBLE](#)
[sippTypes.h, 830](#)
[eSIPP_STATUS_NOT_FOUND](#)
[sippTypes.h, 830](#)
[eSIPP_STATUS_NOT_HOOKED](#)
[sippTypes.h, 831](#)
[eSIPP_STATUS_NOT_IMPLEMENTED](#)
[sippTypes.h, 830](#)
[eSIPP_STATUS_NOT_INIT](#)
[sippTypes.h, 830](#)
[eSIPP_STATUS_NOT_SET](#)
[sippTypes.h, 831](#)
[eSIPP_STATUS_NOT_SUPPORTED](#)
[sippTypes.h, 830](#)
[eSIPP_STATUS_OK](#)
[sippTypes.h, 830](#)
[eSIPP_STATUS_REOPENED_HANDLE](#)
[sippTypes.h, 830](#)
[eSIPP_STATUS_RESOURCE_ERROR](#)
[sippTypes.h, 830](#)
[eSIPP_STATUS_STOPPED](#)
[sippTypes.h, 831](#)
[eSIPP_STATUS_SUSPENDED](#)
[sippTypes.h, 831](#)
[eSIPP_STATUS_TERMINATED](#)
[sippTypes.h, 831](#)
[eSIPP_STATUS_TIMEOUT](#)
[sippTypes.h, 830](#)
[eSIPP_STATUS_UNDEFINED](#)
[sippTypes.h, 830](#)
[eSIPP_STATUS_UNKNOWN](#)
[sippTypes.h, 830](#)
[eBayerOrder](#)
[sippHwBitfieldDefs.h, 723](#)
[eControlPipeAction](#)
[SIPP_ACCESS_SCHEDULER, 425](#)
[EDGE_OPERATOR_CFG](#)
[sippHwBitfieldDefs.h, 689](#)
[eDmaMode](#)
[sippTypes.h, 830](#)
[eEvent](#)
[SIPP_ACCESS_SCHEDULER_QU_ENTR-](#)
[Y, 427](#)
[eMat](#)
[Erode3x3Param, 379](#)
[Erode5x5Param, 380](#)
[Erode7x7Param, 380](#)
[ENABLED](#)
[sippHwCommon_ma2x5x.h, 730](#)
[ePendActions](#)
[SIPP_ACCESS_SCHEDULER, 425](#)
[eRawInputFmt](#)
[sippHwBitfieldDefs.h, 723](#)
[eSIPP_HW_STATE](#)
[sippTypesPrivate.h, 834](#)
[eSIPP_STATUS](#)
[sippTypes.h, 830](#)
[eSippCommand](#)
[sippTypesPrivate.h, 834](#)

- eSippObufControl
 - sippTypesPrivate.h, [835](#)
- eSippPipeState
 - sippTypesPrivate.h, [835](#)
- eState
 - SIPP_HW_SESSION, [428](#)
 - tSippPipelineSuper, [481](#)
- EXE_NUM
 - sippHwIds.h, [741](#)
- Edge operator, [109](#)
- EdgeParam, [377](#)
 - cfg, [378](#)
 - frmDim, [378](#)
 - xCoeff, [378](#)
 - yCoeff, [378](#)
- elementsInQ
 - SIPP_PAL_QU, [429](#)
- emptyBuf
 - combDecimDemosaicAwbGainsStats.c, [514](#)
- end
 - tSippPhysicalPool, [481](#)
- endIter
 - SippPipelineS, [452](#)
- entry
 - sSippCdmaQuS, [468](#)
- epsilon
 - ChrGenParam, [340](#)
 - ChrGenSSParam, [340](#)
- Equalize Histogram, [229](#)
 - SHAVE_SYM_EXPORT, [229](#)
 - svuEqualizeHist, [229](#)
- equalizeHist
 - svuEqualizeHist.c, [904](#)
- equalizeHist.h, [550](#)
- EqualizeHistParam, [378](#)
 - cum_hist, [379](#)
- Erode 3x3, [230](#)
 - SHAVE_SYM_EXPORT, [230](#)
 - svuErode3x3, [230](#)
- Erode 5x5, [231](#)
 - SHAVE_SYM_EXPORT, [231](#)
 - svuErode5x5, [231](#)
- Erode 7x7, [232](#)
 - SHAVE_SYM_EXPORT, [232](#)
 - svuErode7x7, [232](#)
- Erode3x3
 - svuErode3x3.c, [905](#)
- erode3x3.h, [551](#)
- Erode3x3Param, [379](#)
 - eMat, [379](#)
- Erode5x5
 - svuErode5x5.c, [905](#)
- erode5x5.h, [551](#)
- Erode5x5Param, [379](#)
 - eMat, [380](#)
- Erode7x7
 - svuErode7x7.c, [906](#)
- erode7x7.h, [552](#)
- Erode7x7Param, [380](#)
 - eMat, [380](#)
- errorStatus
 - SippFilterS, [433](#)
 - SippPipelineS, [452](#)
- exeNo
 - SippFilterS, [433](#)
- extAfStats.h, [552](#)
- extCfg
 - DmaParam, [375](#)
- extStatsSatPixelsU32.h, [553](#)
 - svuExtStatsSatPixelsU32, [553](#)
- ExtStatsSatPixelsU32Param, [380](#)
 - base, [381](#)
 - cfg, [381](#)
 - crtPaxel, [381](#)
 - crtPosInPaxel, [381](#)
 - ctx, [381](#)
 - fillCtrl, [381](#)
 - firstPatchX, [381](#)
 - firstPatchY, [381](#)
 - in, [381](#)
 - irqRate, [381](#)
 - ls, [382](#)
 - mask, [382](#)
 - nPatchesX, [382](#)
 - nPatchesY, [382](#)
 - nextVerticalStartPos, [382](#)
 - outStatsBuffer, [382](#)
 - outStatsBufferOutside, [382](#)
 - patchGapX, [382](#)
 - patchGapY, [382](#)
 - patchHeight, [382](#)
 - patchWidth, [382](#)
 - ps, [382](#)
 - runNr, [382](#)
 - satThresh, [382](#)
 - shaveNr, [382](#)
 - stackPointer, [382](#)
 - statsSaturatePxl, [382](#)
- f1Coeffs
 - PpAf, [415](#)
- f1Threshold
 - PpAf, [415](#)

- f2
 - YDnsParam, [489](#)
- f2Coeffs
 - PpAf, [416](#)
- f2Threshold
 - PpAf, [416](#)
- FMT_F16
 - Generic Box Filter, [138](#)
- FMT_F32
 - Generic Box Filter, [138](#)
- FMT_U16
 - Generic Box Filter, [138](#)
- FMT_U32
 - Generic Box Filter, [138](#)
- FMT_U8
 - Generic Box Filter, [138](#)
- FALSE
 - PC/sippPlatformAbstractionLayer.c, [812](#)
 - sippTypes.h, [826](#)
- false
 - sippBaseTypes.h, [616](#)
- Fast9M2, [234](#)
 - SHAVE_SYM_EXPORT, [234](#)
 - svuFast9M2, [234](#)
- fast9M2
 - svuFast9M2.c, [909](#)
- fast9M2.h, [553](#)
- Fast9M2Param, [382](#)
 - frameSliceWidth, [383](#)
 - threshold, [383](#)
- Fast9ScoreCv, [235](#)
 - SHAVE_SYM_EXPORT, [235](#)
 - svuFast9ScoreCv, [235](#)
- fast9ScoreCv.h, [554](#)
- Fast9ScoreCvParam, [383](#)
 - frameSliceWidth, [383](#)
 - threshold, [383](#)
- fastBitFlag
 - svuFast9M2.c, [909](#)
- fastExclude
 - svuFast9M2.c, [910](#)
- fastExcludePos
 - svuFast9ScoreCv.c, [911](#)
- fastFlagBit
 - svuFast9ScoreCv.c, [911](#)
- fastScore
 - svuFast9ScoreCv.c, [911](#)
- fillCtrl
 - ExtStatsSatPixelsU32Param, [381](#)
 - SippHwBufS, [444](#)
 - SippHwIOBuf, [445](#)
- filtOutHeight
 - sSchLineBufferS, [466](#)
- filter1_number_of_used_pixels_green
 - AF_paxel_statistics, [331](#)
- filter1_sum_green
 - AF_paxel_statistics, [331](#)
- filter1_sum_max_green
 - AF_paxel_statistics, [331](#)
- filter2_number_of_used_pixels_green
 - AF_paxel_statistics, [331](#)
- filter2_sum_green
 - AF_paxel_statistics, [331](#)
- filter2_sum_max_green
 - AF_paxel_statistics, [331](#)
- filterSize
 - ConvGenericParam, [360](#)
- filterSizeH
 - BoxFilterParam, [336](#)
- filterSizeV
 - BoxFilterParam, [336](#)
- filters
 - SippPipelineS, [452](#)
- filtersDMA
 - SippPipelineS, [453](#)
- filtersHw
 - SippPipelineS, [453](#)
- filtersSvu
 - SippPipelineS, [453](#)
- filts
 - SippOseS, [449](#)
- firstIbflUseMask
 - SippPipelineS, [453](#)
- firstIterLines
 - SippFilterS, [434](#)
- firstOutSlc
 - SippFilterS, [434](#)
- firstPatchX
 - AeAwbStatsCfg, [330](#)
 - ExtStatsSatPixelsU32Param, [381](#)
 - PpAf, [416](#)
 - StatsAwbSatPixelsParam, [471](#)
 - StatsAwbSatPixelsParamU32, [473](#)
- firstPatchY
 - AeAwbStatsCfg, [330](#)
 - ExtStatsSatPixelsU32Param, [381](#)
 - PpAf, [416](#)
 - StatsAwbSatPixelsParam, [471](#)
 - StatsAwbSatPixelsParamU32, [473](#)
- firstRunFilts
 - SippPipelineS, [453](#)
 - tMLPIStartCQCtrl, [478](#)

- firstRunNumLines
 - SippFilterS, [434](#)
- firstRunRollLines
 - SippFilterS, [434](#)
- firstShave
 - StartBicubicParam, [470](#)
 - XYGenParam, [488](#)
- firstUseInProg
 - SippPipelineS, [453](#)
 - tMLPIStartCQCtrl, [478](#)
- firstUseMask
 - SippPipelineS, [453](#)
- flag
 - CensusMatching32Param, [337](#)
 - CensusMatching64Param, [337](#)
- Flags
 - SIPP_TRACE_FLAGS, [430](#)
- flags
 - SippFilterS, [434](#)
 - SippOseS, [449](#)
 - SippPipelineS, [453](#)
- flip_disp2depth
 - Disp2depthParam, [373](#)
- flip_dsp2depth_explic_vect
 - disp2depth_exec.c, [549](#)
 - disp2depth_exec.h, [550](#)
 - svudisp2depth.c, [904](#)
- flip_mvcvDisp2depth
 - disp2depth_exec.c, [549](#)
 - disp2depth_exec.h, [550](#)
- flip_mvcvDisp2depth16bit
 - disp2depth_exec.c, [549](#)
 - disp2depth_exec.h, [550](#)
- FnCheckOpipeCons
 - sippTypesPrivate.h, [834](#)
- FnGetIBufCtx
 - sippTypesPrivate.h, [834](#)
- FnGetIBufIds
 - sippTypesPrivate.h, [834](#)
- FnGetOBufIds
 - sippTypesPrivate.h, [834](#)
- FnHwFltInit
 - sippTypesPrivate.h, [834](#)
- FnHwFltLoad
 - sippTypesPrivate.h, [834](#)
- FnHwSetOBufLatencies
 - sippTypesPrivate.h, [834](#)
- FnHwSetupUpdate
 - sippTypesPrivate.h, [834](#)
- FnSetOBufs
 - sippTypesPrivate.h, [834](#)
- FnSvuRun
 - sippTypes.h, [827](#)
- fp32
 - sippBaseTypes.h, [616](#)
- fptr
 - HPadInfoS, [390](#)
- fraction
 - LscParam, [393](#)
- frameSliceWidth
 - Fast9M2Param, [383](#)
 - Fast9ScoreCvParam, [383](#)
- freeBase
 - memRegDescriptor, [399](#)
- freeCounter
 - tsSippHeap, [482](#)
- freePtr
 - tsSippHeap, [482](#)
- frmDim
 - ChrDnsParam, [339](#)
 - ColCombParam, [342](#)
 - ConvParam, [361](#)
 - DbyrParam, [370](#)
 - DogLtmParam, [377](#)
 - EdgeParam, [378](#)
 - GenChrParam, [384](#)
 - HarrisParam, [387](#)
 - LscParam, [393](#)
 - LutParam, [397](#)
 - MedParam, [399](#)
 - MipiRxParam, [403](#)
 - MipiTxParam, [406](#)
 - RawParam, [420](#)
 - SigmaParam, [424](#)
 - UsmParam, [485](#)
 - YDnsParam, [489](#)
- frmDimFlt
 - PolyFirParam, [412](#)
- frmDimPar
 - PolyFirParam, [412](#)
- fullCmxSlice
 - cmxRegUsage, [341](#)
- fullFrmDim
 - YDnsParam, [489](#)
- funcAsk
 - sSchedIBufUsageInfoS, [465](#)
- funcSvuRun
 - SippFilterS, [434](#)
- GBRG
 - sippHwCommon_ma2x5x.h, [730](#)
- GET_REG_WORD_VAL
 - svuExtAfStats.c, [907](#)

- gFakeFptr
 - sippCoreApi.c, [624](#)
- GRBG
 - sippHwCommon_ma2x5x.h, [730](#)
- gSipp
 - sippTypesPrivate.h, [835](#)
- gSippErrCode
 - sippCoreApi.c, [624](#)
 - sippError.c, [636](#)
- gSippErrRdIdx
 - sippError.c, [636](#)
- gSippErrWrIdx
 - sippError.c, [636](#)
- gSippFatalErrList
 - sippError.c, [637](#)
- gSippFramework
 - sippAccessScheduler.c, [607](#)
 - sippApi.c, [613](#)
 - sippApiInternal.c, [615](#)
 - sippGenericRuntime.c, [642](#)
 - sippIsr.c, [770](#)
 - sippManagerApi.c, [771](#)
 - sippPipeSessionControl.c, [807](#)
- gSippInitMask
 - sippApi.c, [613](#)
 - sippCmxDmaIf.c, [623](#)
 - sippCoreApi.c, [625](#)
- gSippSliceSz
 - sippApi.c, [613](#)
 - sippCmxDmaIf.c, [623](#)
 - sippCoreApi.c, [625](#)
 - sippCoreUtils.c, [628](#)
 - sippMem.c, [774](#)
- gainSat
 - RawParam, [420](#)
- gains
 - CombDecimAwbGainsParam, [343](#)
 - CombDecimStatsGainsParam, [344](#)
- gammaLut
 - YDnsParam, [489](#)
- gauss
 - svuGauss.c, [912](#)
- Gauss Blur, [236](#)
 - SHAVE_SYM_EXPORT, [236](#)
 - svuGauss, [236](#)
- gauss.h, [555](#)
- GaussHx2, [237](#)
 - SHAVE_SYM_EXPORT, [237](#)
 - svuGaussHx2, [237](#)
- gaussHx2.h, [555](#)
- GaussHx2_fp16, [238](#)
 - SHAVE_SYM_EXPORT, [238](#)
 - svuGaussHx2_fp16, [238](#)
 - svuGaussHx2_fp16.c, [914](#)
- gaussHx2_fp16.h, [555](#)
- gaussLut
 - YDnsParam, [489](#)
- GaussVx2, [239](#)
 - SHAVE_SYM_EXPORT, [239](#)
 - svuGaussVx2, [239](#)
- gaussVx2.h, [556](#)
- GaussVx2_fp16, [240](#)
 - SHAVE_SYM_EXPORT, [240](#)
 - svuGaussVx2_fp16, [240](#)
 - svuGaussVx2_fp16.c, [915](#)
- gaussVx2_fp16.h, [556](#)
- GenChrParam, [384](#)
 - cfg, [384](#)
 - chrCoefs, [384](#)
 - frmDim, [384](#)
 - yCoefs, [384](#)
- genChroma
 - svuGenChroma.c, [915](#)
- genChroma.h, [557](#)
- GenChromaSS
 - svuGenChromaSS.c, [916](#)
- genChromaSS.h, [557](#)
- genDnsRef
 - svuGenDnsRef.c, [916](#)
- genDnsRef.h, [558](#)
- genDnsRefFp16
 - svuGenDnsRefFp16.c, [917](#)
- genDnsRefFp16.h, [558](#)
- genLuma
 - svuGenLuma.c, [917](#)
- genLuma.h, [559](#)
- genLumaU8Fp16
 - svuGenLumaU8Fp16.c, [917](#)
- genLumaU8Fp16.h, [559](#)
- GenLumaU8Fp16Param, [385](#)
 - coefs, [385](#)
- genRand
 - svuRandNoiseFp16.c, [952](#)
- genXYlist
 - svuXYgen.c, [974](#)
- Generate Chroma, [241](#)
 - SHAVE_SYM_EXPORT, [241](#)
 - svuGenChroma, [241](#)
- Generate Chroma with subsampling, [242](#)
 - SHAVE_SYM_EXPORT, [242](#)
 - svuGenChromaSS, [242](#)
- Generate Luma U8 to Fp16, [246](#)

- SHAVE_SYM_EXPORT, [246](#)
- svuGenLumaU8Fp16, [246](#)
- Generate Reference for Luma Denoise, [243](#)
 - SHAVE_SYM_EXPORT, [243](#)
 - svuGenDnsRef, [243](#)
- Generate Reference for Luma Denoise(fp16 input), [244](#)
 - SHAVE_SYM_EXPORT, [244](#)
 - svuGenDnsRefFp16, [244](#)
- Generic Box Filter, [138](#)
 - boxDataFmt, [138](#)
 - FMT_F16, [138](#)
 - FMT_F32, [138](#)
 - FMT_U16, [138](#)
 - FMT_U32, [138](#)
 - FMT_U8, [138](#)
 - SHAVE_SYM_EXPORT, [139](#)
 - svuBoxFilter, [139](#)
- Generic Convolution, [190](#)
 - SHAVE_SYM_EXPORT, [190](#)
 - svuConvGeneric, [190](#)
- Generic Dilate, [228](#)
 - SHAVE_SYM_EXPORT, [228](#)
 - svuDilateGeneric, [228](#)
- getIn3PlanePtr
 - sipp_ma2x5x.h, [599](#)
 - sippIoPtrs.c, [767](#)
- getInPtr
 - sipp_ma2x5x.h, [599](#)
 - sippIoPtrs.c, [767](#)
- getInPtrAbs
 - sippIoPtrs.c, [768](#)
 - svuHomography.c, [921](#)
 - svuUndistortBrown.c, [972](#)
- getOutPtr
 - sipp_ma2x5x.h, [600](#)
 - sippIoPtrs.c, [768](#)
- getPlaneIoPtrs
 - sipp_ma2x5x.h, [600](#)
 - sippIoPtrs.c, [768](#)
- getRandom
 - svuRandNoise.c, [951](#)
- gi
 - SippFilterS, [434](#)
 - SippPipelineS, [453](#)
- gmBase
 - LscParam, [393](#)
- gmDim
 - LscParam, [393](#)
- gmHeight
 - LscParam, [393](#)
- gmWidth
 - LscParam, [393](#)
- grey
 - GreyDesatParam, [385](#)
- greyDesat, [247](#)
 - SHAVE_SYM_EXPORT, [247](#)
 - svuGreyDesat, [247](#)
 - svuGreyDesat.c, [918](#)
- greyDesat.h, [560](#)
- GreyDesatParam, [385](#)
 - grey, [385](#)
 - offset, [385](#)
 - slope, [386](#)
- greyPt
 - ChrDnsParam, [339](#)
- grgbDecay
 - RawParam, [420](#)
- grgbPlat
 - RawParam, [420](#)
- gsSippMemMap
 - sippMem.c, [774](#)
- HARRIS_CORNER_CFG
 - sippHwBitfieldDefs.h, [692](#)
- HARRIS_SW_RADIUS
 - svuHarrisResponse.c, [919](#)
- hActiveWidth
 - MipiTxParam, [406](#)
- hBackPorch
 - MipiTxParam, [406](#)
- hFrontPorch
 - MipiTxParam, [406](#)
- HIST_KERNEL_SIZE
 - sippHwCommon_ma2x5x.h, [730](#)
- hKerSz
 - sSchedIBufUsageInfoS, [465](#)
- HPadInfo
 - sippTypes.h, [827](#)
- HPadInfoS, [389](#)
 - CT1, [390](#)
 - fptr, [390](#)
 - oBufId, [390](#)
 - padOffsets, [390](#)
 - svuPadFunc, [390](#)
- hPadding
 - SippFilterS, [435](#)
 - sSchLineBufferS, [466](#)
- hSyncWidth
 - MipiTxParam, [406](#)
- HWPipeID
 - SippPipelineS, [454](#)
- HWWaitCycles

- tRTStats, [479](#)
- hammingDistance, [248](#)
 - SHAVE_SYM_EXPORT, [248](#)
 - svuHammingDistance, [248](#)
- hammingDistance.h, [561](#)
- HammingDistanceParam, [386](#)
 - descriptor_size, [386](#)
- Harris Corner Detector, [106](#)
- HarrisParam, [386](#)
 - cfg, [387](#)
 - frmDim, [387](#)
 - kValue, [387](#)
- HarrisResponse
 - svuHarrisResponse.c, [919](#)
- harrisResponse, [249](#)
 - SHAVE_SYM_EXPORT, [249](#)
 - svuHarrisResponse, [249](#)
- harrisResponse.h, [561](#)
- HarrisSwParam, [387](#)
 - k, [387](#)
- hasBuffers
 - SippFilterSchedInfoS, [443](#)
- hbp
 - MipiTxLoopbackParam, [404](#)
- height
 - StartBicubicParam, [470](#)
- hfp
 - MipiTxLoopbackParam, [404](#)
- hist
 - HistogramParam, [388](#)
- histLumaBase
 - RawParam, [420](#)
- histRgbBase
 - RawParam, [420](#)
- Histogram, [250](#)
 - SHAVE_SYM_EXPORT, [250](#)
 - svuHistogram, [250](#)
- histogram
 - svuHistogram.c, [920](#)
- histogram.h, [562](#)
- HistogramParam, [387](#)
 - hist, [388](#)
- histogramStat, [251](#)
 - SHAVE_SYM_EXPORT, [251](#)
 - svuHistogramStat, [251](#)
- histogramStat.h, [562](#)
- HistogramStatParam, [388](#)
 - step, [388](#)
- homoMat3x3
 - HomographyParam, [389](#)
- Homography, [252](#)
 - SHAVE_SYM_EXPORT, [252](#)
 - svuHomography, [252](#)
- homography.h, [563](#)
- HomographyParam, [388](#)
 - homoMat3x3, [389](#)
- HorizPaddingOff
 - sippTypes.h, [827](#)
- HorizPaddingOffS, [389](#)
 - dstLeftO, [389](#)
 - dstRightO, [389](#)
 - srcLeftO, [389](#)
 - srcRightO, [389](#)
- horzCoefs
 - PolyFirParam, [413](#)
- horzD
 - PolyFirParam, [413](#)
- horzN
 - PolyFirParam, [413](#)
- hsync
 - MipiTxLoopbackParam, [404](#)
- hwFnInit
 - UnitInfo, [484](#)
- hwFnLoad
 - UnitInfo, [484](#)
- hwInputBufId
 - sSchLineBufferS, [466](#)
- hwOutputBufId
 - sSchLineBufferS, [467](#)
- hwSippCtxSwMask
 - SippPipelineS, [454](#)
- hwSippFirst
 - SippPipelineS, [454](#)
- hwSippFltCnt
 - SippPipelineS, [454](#)
- hzInterval
 - combDecimDemosaicAwbGainsStats.c, [514](#)
- I_BASE
 - sippDefines.h, [632](#)
- I_CFG
 - sippDefines.h, [632](#)
- I_CTX
 - sippDefines.h, [632](#)
- I_FC
 - sippDefines.h, [632](#)
- I_LS
 - sippDefines.h, [632](#)
- I_PS
 - sippDefines.h, [632](#)
- I_SHADOW_BASE
 - sippDefines.h, [632](#)
- I_SHADOW_CFG

- sippDefines.h, [632](#)
- I_SHADOW_LS
 - sippDefines.h, [632](#)
- I_SHADOW_PS
 - sippDefines.h, [632](#)
- iBuf
 - SippFilterS, [435](#)
- iBufCtx
 - SippFilterS, [435](#)
 - SippOseS, [449](#)
 - SippSchEntS, [462](#)
- iBufs
 - SippFilterS, [435](#)
 - SippOseS, [449](#)
 - SippSchEntS, [462](#)
- INLINE
 - sippPlatform_ma2x5x.h, [809](#)
- iPadLines
 - sSchedIBufUsageInfoS, [465](#)
- IRF_BASE
 - svuExtAfStats.c, [907](#)
 - svuExtStatsSatPixelsU32.c, [908](#)
- ISP, [215](#)
 - SHAVE_SYM_EXPORT, [218](#)
 - svuCvtColorRGBtoLuma, [218](#)
- ibflIncDelta
 - SippPipelineS, [454](#)
- id
 - SippFilterS, [435](#)
- idxLow
 - ContrastParam, [346](#)
- imgAddr
 - MipiRxLoopbackParam, [402](#)
 - MipiTxLoopbackParam, [404](#)
- imgH
 - MipiRxLoopbackParam, [402](#)
 - MipiTxLoopbackParam, [404](#)
- imgW
 - MipiRxLoopbackParam, [402](#)
 - MipiTxLoopbackParam, [404](#)
- in
 - ExtStatsSatPixelsU32Param, [381](#)
 - PpAf, [416](#)
- inBufferCandidates
 - MaxTest3x3fp16Param, [398](#)
 - MinTest3x3fp16Param, [401](#)
- initialSubtractionValue
 - PpAf, [416](#)
- input
 - StartBicubicParam, [470](#)
- inputBits
 - MonoImbalanceParam, [408](#)
- inputFrameType
 - CvtColorChromaYUVToNV12Param, [368](#)
- Int16
 - sippBaseTypes.h, [616](#)
- Int32
 - sippBaseTypes.h, [616](#)
- Int8
 - sippBaseTypes.h, [616](#)
- Integral Image Square Sum(U32), [254](#)
 - SHAVE_SYM_EXPORT, [254](#)
 - svuIntegralImageSqSumU32M2, [254](#)
- Integral Image Square Sum(f32), [253](#)
 - SHAVE_SYM_EXPORT, [253](#)
 - svuIntegralImageSqSumF32M2, [253](#)
- Integral Image Sum(U16toU32), [256](#)
 - SHAVE_SYM_EXPORT, [256](#)
 - svuIntegralImageSumU16U32, [256](#)
- Integral Image Sum(U32), [257](#)
 - SHAVE_SYM_EXPORT, [257](#)
 - svuIntegralImageSumU32M2, [257](#)
- Integral Image Sum(f32), [255](#)
 - SHAVE_SYM_EXPORT, [255](#)
 - svuIntegralImageSumF32M2, [255](#)
- integralImageSqSumF32M2.h, [563](#)
- integralImageSqSumU32M2.h, [564](#)
- integralImageSumF32M2.h, [564](#)
- integralImageSumU16U32
 - svuIntegralImageSumU16U32.c, [924](#)
- integralImageSumU16U32.h, [565](#)
- integralImageSumU32M2.h, [565](#)
- integralimage_sqsum_f32_M2
 - svuIntegralImageSqSumF32M2.c, [922](#)
- integralimage_sqsum_u32M2
 - svuIntegralImageSqSumU32M2.c, [923](#)
- integralimage_sum_f32M2
 - svuIntegralImageSumF32M2.c, [924](#)
- integralimage_sum_u32M2
 - svuIntegralImageSumU32M2.c, [926](#)
- internalFillLevel
 - sSchLineBufferS, [467](#)
- interpolatePixelBilinear, [258](#)
 - SHAVE_SYM_EXPORT, [258](#)
 - svuInterpolatePixelBilinear, [258](#)
- interpolatePixelBilinear.h, [566](#)
- InterpolatePixelBilinearParam, [390](#)
 - x, [390](#)
 - y, [390](#)
- irqRate
 - ExtStatsSatPixelsU32Param, [381](#)
 - SippHwBufS, [445](#)

- SippHwIOBuf, 445
- isrFlags
 - SippPipelineS, 454
- isrSetup
 - SIPP_HW_SESSION, 428
- iterTime
 - tRTStats, 479
- iteration
 - SippPipelineS, 454
- itersLeft
 - SippPipelineS, 454
- k
 - HarrisSwParam, 387
- k1
 - UndistortBParam, 483
- k2
 - UndistortBParam, 483
- KSDelta
 - sSchedIBufUsageInfoS, 465
- KSIterList
 - SippFilterS, 435
 - SippSchEntS, 462
- kValue
 - HarrisParam, 387
- kerSz
 - PolyFirParam, 413
- kern
 - svuScale2xLancHV.c, 961
- kernel
 - ConvParam, 361
- kernelSize
 - DilateGenericParam, 372
- krgb
 - ColCombParam, 343
- LSC_KERNEL_SIZE
 - sippHwCommon_ma2x5x.h, 730
- LUMA_DNS_CFG
 - sippHwBitfieldDefs.h, 693
- LUMA_KERNEL_SIZE
 - sippHwCommon_ma2x5x.h, 730
- LUT
 - SubpixelFilterParam, 474
- LUT 10 to 16, 270
 - SHAVE_SYM_EXPORT, 270
 - svuLut10to16, 270
- LUT 10 to 8, 271
 - SHAVE_SYM_EXPORT, 271
 - svuLut10to8, 271
- LUT 12 to 16, 272
 - SHAVE_SYM_EXPORT, 272
- svuLut12to16, 272
- LUT 12 to 8, 273
 - SHAVE_SYM_EXPORT, 273
 - svuLut12to8, 273
- LUT 16 to 8, 275
 - SHAVE_SYM_EXPORT, 275
 - svuLutP10BppU16inU8out, 275
- LUT 8 to 8, 274
 - SHAVE_SYM_EXPORT, 274
 - svuLut8to8, 274
- LUT10to16
 - svuLut10to16.c, 934
- LUT10to8
 - svuLut10to8.c, 935
- LUT12to16
 - svuLut12to16.c, 936
- LUT12to8
 - svuLut12to8.c, 937
- LUT8to8
 - svuLut8to8.c, 937
- LUT_CFG
 - sippHwBitfieldDefs.h, 694
- LUT_CFG_NUM_LUTS
 - sippHwBitfieldDefs.h, 695
- LUT_KERNEL_SIZE
 - sippHwCommon_ma2x5x.h, 730
- Lanczos Downscale by 2 (6 taps), 301
 - SHAVE_SYM_EXPORT, 301
 - svuScI05Lanc6, 301
- Lanczos Downscale by 2 (7 taps), 302
 - SHAVE_SYM_EXPORT, 302
 - svuScI05Lanc7, 302
- Lanczos Horizontal Upscale by 2, 308
 - SHAVE_SYM_EXPORT, 308
 - svuScI2xLancH, 308
- Lanczos Upscale by 2, 309
 - SHAVE_SYM_EXPORT, 309
 - svuScI2xLancHV, 309
- Lanczos Vertical Upscale by 2, 310
 - SHAVE_SYM_EXPORT, 310
 - svuScI2xLancV, 310
- Laplacian 3x3, 259
 - SHAVE_SYM_EXPORT, 259
 - svuLaplacian3x3, 259
- Laplacian 3x3 Fp16 To Fp16, 313
 - SHAVE_SYM_EXPORT, 313
 - svusLaplacian3x3Fp16ToFp16, 313
- Laplacian 5x5, 260
 - SHAVE_SYM_EXPORT, 260
 - svuLaplacian5x5, 260
- Laplacian 5x5 Fp16 To Fp16, 261

- SHAVE_SYM_EXPORT, [261](#)
- svuLaplacian5x5Fp16ToFp16, [261](#)
- Laplacian 7x7, [262](#)
 - SHAVE_SYM_EXPORT, [262](#)
 - svuLaplacian7x7, [262](#)
- Laplacian 7x7 Fp16 To Fp16, [263](#)
 - SHAVE_SYM_EXPORT, [263](#)
 - svuLaplacian7x7Fp16ToFp16, [263](#)
- laplacian3x3.h, [566](#)
- Laplacian3x3Implementation
 - svuLaplacian3x3.c, [928](#)
- laplacian5x5.h, [567](#)
- laplacian5x5Fp16ToFp16.h, [567](#)
- laplacian7x7.h, [568](#)
- laplacian7x7Fp16ToFp16.h, [568](#)
- lastAskLineNo
 - sSchedIBufUsageInfoS, [465](#)
- lastShave
 - StartBicubicParam, [470](#)
 - XYGenParam, [488](#)
- latenciesKnown
 - SippFilterSchedInfoS, [443](#)
- latency
 - sSchLineBufferS, [467](#)
- Lens Shading Correction, [97](#)
- limit
 - UsmParam, [485](#)
- lineCompare
 - MipiTxParam, [406](#)
- lineMemPoolBase
 - SippPipelineS, [454](#)
- lineNo
 - PixelUnpackerMipi10bParam, [410](#)
- linePtrs
 - SippFilterS, [435](#)
- linePtrs1stBase
 - SippFilterS, [435](#)
- linePtrs2ndBase
 - SippFilterS, [435](#)
- linePtrs3rdBase
 - SippFilterS, [436](#)
- linePtrs4thBase
 - SippFilterS, [436](#)
- linePtrs5thBase
 - SippFilterS, [436](#)
- lineStride
 - SippFilterS, [436](#)
- linesPerIter
 - SippFilterS, [436](#)
 - SippOseS, [449](#)
 - SippPipelineS, [455](#)
 - SippSchEntS, [462](#)
- linesPerIterShift
 - SippFilterS, [436](#)
- linesThisIter
 - SippFilterS, [436](#)
- lnToPad
 - SippFilterS, [436](#)
- localMaxMin3x3_fp16, [264](#)
 - SHAVE_SYM_EXPORT, [264](#)
 - svuLocalMaxMin3x3_fp16, [264](#)
- localMaxMin3x3_fp16.h, [569](#)
- localTM, [265](#)
 - SHAVE_SYM_EXPORT, [265](#)
 - svuLocalTM, [265](#)
 - svuLocalTM.c, [932](#)
- localTM.h, [569](#)
- LocalTMParam, [391](#)
 - curves, [391](#)
- Look-up table, [103](#)
- Low Level Correction, [266](#)
 - alphaBadPixel, [267](#)
 - blackLevel, [267](#)
 - SHAVE_SYM_EXPORT, [266](#)
 - svuLowLvlCorr, [266](#)
- Low Level Correction on Multiple Planes, [268](#)
 - SHAVE_SYM_EXPORT, [268](#)
 - svulowLvlCorrMultiplePlanes, [268](#)
- lowLvlCorr.h, [570](#)
- lowLvlCorrMultiplePlanes.c, [570](#)
 - SUB_SATURATE, [571](#)
- lowLvlCorrMultiplePlanes.h, [571](#)
- LowLvlCorrNPiParam, [391](#)
 - alphaBadPixel, [391](#)
 - blackLevel, [391](#)
- LowLvlCorrParam, [392](#)
- lowerValue
 - ThresholdBinaryRangeParam, [475](#)
- ls
 - ExtStatsSatPixelsU32Param, [382](#)
 - SippHwBufS, [445](#)
 - SippHwIOBuf, [446](#)
- LscParam, [392](#)
 - cfg, [393](#)
 - dataFormat, [393](#)
 - dataWidth, [393](#)
 - fraction, [393](#)
 - frmDim, [393](#)
 - gmBase, [393](#)
 - gmDim, [393](#)
 - gmHeight, [393](#)
 - gmWidth, [393](#)

- ltnCurves
 - DogLtnParam, 377
- Luma Blur, 245, 269
 - SHAVE_SYM_EXPORT, 245, 269
 - svuGenLuma, 245
 - svuLumaBlur, 269
- Luma Denoise, 101
- lumaAlpha
 - MedParam, 399
- lumaBlur
 - svuLumaBlur.c, 933
- lumaBlur.h, 571
- lumaGenLut
 - sipp_ma2x5x.h, 601
 - sippUtils.c, 836
- lumaWeight
 - DbyrParam, 370
- lut
 - LutParam, 397
 - YDnsRefLut10bppParam, 491
- lut10to16.h, 572
- Lut10to16Param, 394
 - lutValue, 394
- lut10to8.h, 572
- Lut10to8Param, 394
 - lutValue, 394
- lut12to16.h, 573
- Lut12to16Param, 395
 - lutValue, 395
- lut12to8.h, 573
- Lut12to8Param, 395
 - lutValue, 395
- lut8to8.h, 574
- Lut8to8Param, 396
 - lutValue, 396
- lutDist
 - YDnsRefFp16Param, 490
 - YDnsRefParam, 491
- lutFormat
 - ColCombParam, 343
 - LutParam, 397
- lutGamma
 - YDnsRefFp16Param, 490
 - YDnsRefParam, 491
- lutP10BppU16inU8out.h, 575
- LutParam, 396
 - cfg, 397
 - frmDim, 397
 - lut, 397
 - lutFormat, 397
 - mat, 397
 - offset, 397
 - sizeA, 397
 - sizeB, 397
- lutValue
 - Lut10to16Param, 394
 - Lut10to8Param, 394
 - Lut12to16Param, 395
 - Lut12to8Param, 395
 - Lut8to8Param, 396
- MAGN_ORIENT_16BIT
 - sippHwCommon_ma2x5x.h, 730
- MARGIN
 - svuScale2xLancHV.c, 961
- MAX
 - svuPurpleFlare.c, 950
- MAX_LUT_IDX
 - svuLutP10BppU16inU8out.c, 938
- MAX_PLANES
 - sippHwCommon_ma2x5x.h, 730
- MAX_STATS_SIZE
 - combDecimDemosaicAwbGainsStats.c, 513
- MAX_U16_VAL
 - svuConvertPFp16U16.c, 882
 - svuConvertPU16Fp16.c, 882
- MAX_WIDTH
 - svuCannyEdgeDetection.c, 858
 - svuCornerMinEigenVal.c, 891
- MCB_MAGIC
 - sippHeap.c, 648
- MDKdox-Sipp-intro.txt, 576
- MED_CFG
 - sippHwBitfieldDefs.h, 698
- MED_CFG_GO_FAST
 - sippHwBitfieldDefs.h, 698
- MED_CFG_THRESHOLD
 - sippHwBitfieldDefs.h, 699
- MED_KERNEL_SIZE
 - sippHwCommon_ma2x5x.h, 730
- MED_LUMA_ALPHA
 - sippHwBitfieldDefs.h, 699
- MIN
 - svuPurpleFlare.c, 950
- MIPI Rx, 113
- MIPI Tx, 114
- MIPI_RX_CFG
 - sippHwBitfieldDefs.h, 699
- MIPI_TX_FRM_CFG
 - sippHwBitfieldDefs.h, 704
- MIPI_TX_FRM_WIDTH
 - sippHwBitfieldDefs.h, 705
- MIPI_TX_INT_SEL

- sippHwBitfieldDefs.h, 705
- MIPI_TX_ONESHOT
 - sippHwBitfieldDefs.h, 705
- MIPI_TX_SCAN_MODE
 - sippHwBitfieldDefs.h, 705
- MIPI_TX_SYNC_IDLE
 - sippHwBitfieldDefs.h, 705
- manageReq
 - sSchLineBufferS, 467
- Mask
 - MinMaxPosParam, 400
- mask
 - ExtStatsSatPixelsU32Param, 382
 - PpAf, 416
- maskAddr
 - minMaxValParam, 400
 - positionKernelParam, 414
- mat
 - LutParam, 397
- max
 - svuBilateral5x5.c, 848
- maxCountIn
 - MaxTest3x3fp16Param, 398
- maxLineRequired
 - SippSchEntS, 462
- maxLocationsIn
 - MaxTest3x3fp16Param, 398
- maxOutputH
 - SippSchEntS, 462
- maxQElements
 - SIPP_PAL_QU, 429
- maxTest3x3_fp16, 276
 - SHAVE_SYM_EXPORT, 276
 - svuMaxTest3x3_fp16, 276
- maxTest3x3_fp16.h, 575
- MaxTest3x3fp16Param, 397
 - inBufferCandidates, 398
 - maxCountIn, 398
 - maxLocationsIn, 398
- maxVal
 - minMaxValParam, 400
- mbinImg
 - SippPipelineS, 455
- mbinImgSipp
 - sippTestCommon_ma2x5x.h, 823
- meanStdDev, 277
 - SHAVE_SYM_EXPORT, 277
 - svuMeanStdDev, 277
- meanStdDev.h, 576
- MedParam, 398
 - cfg, 399
- frmDim, 399
- lumaAlpha, 399
- Median, 96
- memRegDescriptor, 399
 - freeBase, 399
 - sizeRemaining, 399
- memsetBpp
 - sippHpad.c, 650
- Min/Max Value, 279
 - SHAVE_SYM_EXPORT, 279
 - svuMinMaxValue, 279
- Min/Max Value Position, 278
 - SHAVE_SYM_EXPORT, 278
 - svuMinMaxPos, 278
- minCountIn
 - MinTest3x3fp16Param, 401
- minLineRequired
 - SippSchEntS, 462
- minLocationsIn
 - MinTest3x3fp16Param, 401
- minMaxKernel
 - svuMinMaxValue.c, 941
- minMaxPos
 - svuMinMaxPos.c, 940
- minMaxPos.h, 576
- MinMaxPosParam, 399
 - Mask, 400
- minMaxValParam, 400
 - maskAddr, 400
 - maxVal, 400
 - minVal, 400
- minMaxValue.h, 577
- minTest3x3_fp16, 280
 - SHAVE_SYM_EXPORT, 280
 - svuMinTest3x3_fp16, 280
- minTest3x3_fp16.h, 577
- MinTest3x3fp16Param, 401
 - inBufferCandidates, 401
 - minCountIn, 401
 - minLocationsIn, 401
- minVal
 - minMaxValParam, 400
- minimumCalc
 - svuFast9ScoreCv.c, 911
- MipiRxLoopbackParam, 401
 - bpp, 402
 - imgAddr, 402
 - imgH, 402
 - imgW, 402
 - rxID, 402
- MipiRxParam, 402

- black01, [403](#)
- black23, [403](#)
- cfg, [403](#)
- frmDim, [403](#)
- sel01, [403](#)
- sel23, [403](#)
- selMask, [403](#)
- vbp, [403](#)
- winX, [403](#)
- winY, [404](#)
- MipiTxLoopbackParam, [404](#)
 - bpp, [404](#)
 - hbp, [404](#)
 - hfp, [404](#)
 - hsync, [404](#)
 - imgAddr, [404](#)
 - imgH, [404](#)
 - imgW, [404](#)
 - txID, [404](#)
 - vsync, [404](#)
- MipiTxParam, [405](#)
 - cfg, [406](#)
 - frmDim, [406](#)
 - hActiveWidth, [406](#)
 - hBackPorch, [406](#)
 - hFrontPorch, [406](#)
 - hSyncWidth, [406](#)
 - lineCompare, [406](#)
 - vActiveHeight, [406](#)
 - vBackPorch, [406](#)
 - vCompare, [407](#)
 - vFrontPorch, [407](#)
 - vSyncEndOff, [407](#)
 - vSyncStartOff, [407](#)
 - vSyncWidth, [407](#)
- Mix Median, [281](#)
 - SHAVE_SYM_EXPORT, [281](#)
 - svuMixMedian, [281](#)
- mixMedian
 - svuMixMedian.c, [942](#)
- mixMedian.h, [578](#)
- MixMedianParam, [407](#)
 - offset, [408](#)
 - slope, [408](#)
- mode
 - PolyFirParam, [413](#)
- MonoImbalance, [282](#)
- monoImbalance.h, [579](#)
 - SHAVE_SYM_EXPORT, [579](#)
 - svuMonoImbalance, [579](#)
- MonoImbalanceParam, [408](#)
- inputBits, [408](#)
- thrBright, [408](#)
- thrDark, [408](#)
- mostRecentLine
 - sSchLineBufferS, [467](#)
- mskBit
 - svuRandNoiseFp16.c, [953](#)
- multiHwCtx
 - SippPipelineS, [455](#)
- mvcvBilateral5x5
 - svuBilateral5x5.c, [848](#)
- mvcvCensusMatching16
 - svuCensusMatching16.c, [859](#)
- mvcvCensusMatching32
 - svuCensusMatching32.c, [859](#)
- mvcvCensusMatching64
 - svuCensusMatching64.c, [860](#)
- mvcvCensusMatching65
 - svuCensusMatching65.c, [861](#)
- mvcvCensusMatchingPyr
 - svuCensusMatchingPyr.c, [861](#)
- mvcvCensusMin16
 - svuCensusMin16.c, [862](#)
- mvcvCensusMin64
 - svuCensusMin64.c, [862](#)
- mvcvCensusMin65
 - svuCensusMin65.c, [863](#)
- mvcvCensusMin7
 - svuCensusMin7.c, [864](#)
- mvcvCensusTransform5x5
 - svuCensusTransform5x5.c, [864](#)
- mvcvConvert12BppTo8Bpp
 - svuConvertFrom12BppTo8Bpp.c, [881](#)
- mvcvDisp2depth
 - disp2depth_exec.c, [549](#)
 - disp2depth_exec.h, [550](#)
- mvcvDisp2depth16bit
 - disp2depth_exec.c, [549](#)
 - disp2depth_exec.h, [550](#)
- mvcvGaussHx2
 - svuGaussHx2.c, [913](#)
- mvcvGaussVx2
 - svuGaussVx2.c, [914](#)
- mvcvHammingDistance
 - svuHammingDistance.c, [919](#)
- mvcvInterpolatePixelBilinear
 - svuInterpolatePixelBilinear.c, [927](#)
- mvcvLocalMaxMin3x3_fp16
 - svuLocalMaxMin3x3_fp16.c, [931](#)
- mvcvMaxTest3x3_fp16
 - svuMaxTest3x3_fp16.c, [939](#)

- mv cvMeanStddev
 - svuMeanStdDev.c, [939](#)
- mv cvMinTest3x3_fp16
 - svuMinTest3x3_fp16.c, [942](#)
- mv cvMonoImbalance
 - svuMonoImbalance.c, [943](#)
- mv cvNonMax3x3_fp32
 - svuNonMax3x3Fp32.c, [944](#)
- mv cvNonMax3x3_u8
 - svuNonMax3x3U8.c, [945](#)
- mv cvScharr_fp16
 - svuScharr_fp16.c, [963](#)
- mv cvSsdPointLine7x7U8U32
 - svuSsdPointLine7x7U8U32.c, [966](#)
- mv cvSubPixelFilter_asm
 - svuSubpixelFilter.c, [969](#)
- mv cvThresholdFilter
 - svuThresholdFilter.c, [971](#)
- mv cvfast9ScoreCv
 - svuFast9ScoreCv.c, [911](#)
- mvispHistogramStat
 - svuHistogramStat.c, [921](#)
- myriad2SippDefs.inc, [579](#)
- N_PL
 - sipp_ma2x5x.h, [597](#)
- nCons
 - SippFilterS, [436](#)
- nCtxLoads
 - SippFilterS, [437](#)
- nFilters
 - SippPipelineS, [455](#)
- nFiltersDMA
 - SippPipelineS, [455](#)
- nFiltersHw
 - SippPipelineS, [455](#)
- nFiltersSvu
 - SippPipelineS, [455](#)
- nFirstRunFilts
 - SippPipelineS, [455](#)
 - tMLPIStartCQCtrl, [478](#)
- nIter
 - SippPipelineS, [456](#)
- nLines
 - SippFilterS, [437](#)
- nLinesUsed
 - SippFilterS, [437](#)
- NOP
 - sippPlatform_ma2x5x.h, [809](#)
- NORMAL_MODE
 - sippHwCommon_ma2x5x.h, [731](#)
- NORMAL_THETA
 - sippHwCommon_ma2x5x.h, [731](#)
- nPadFilters
 - SippPipelineS, [456](#)
- nParents
 - SippFilterS, [437](#)
- nPatchesX
 - AeAwbStatsCfg, [330](#)
 - ExtStatsSatPixelsU32Param, [382](#)
 - PpAf, [416](#)
 - StatsAwbSatPixelsParam, [471](#)
 - StatsAwbSatPixelsParamU32, [473](#)
- nPatchesY
 - AeAwbStatsCfg, [330](#)
 - ExtStatsSatPixelsU32Param, [382](#)
 - PpAf, [416](#)
 - StatsAwbSatPixelsParam, [471](#)
 - StatsAwbSatPixelsParamU32, [473](#)
- nPlanes
 - ParentInfoS, [409](#)
 - SippFilterS, [437](#)
- nSkipRows
 - PpAf, [416](#)
- NULL
 - sippTypes.h, [826](#)
- NV21 to RGB conversion, [210](#)
 - SHAVE_SYM_EXPORT, [210](#)
 - svucvtColorNV21toRGB, [210](#)
- name
 - tSippPhysicalPool, [481](#)
- needs2Parents
 - CvtColorChromaYUVToNV12Param, [368](#)
- Negative, [283](#)
 - SHAVE_SYM_EXPORT, [283](#)
 - svuNegative, [283](#)
- negative.h, [579](#)
- negativeFilterImplementation
 - svuNegative.c, [944](#)
- NextVerticalStartPos
 - StatsAwbSatPixelsParam, [471](#)
 - StatsAwbSatPixelsParamU32, [473](#)
- nextVerticalStartPos
 - ExtStatsSatPixelsU32Param, [382](#)
- nlinesUsedParent
 - SippFilterSchedInfoS, [443](#)
- noIntBarSwitchPend
 - SIPP_HW_SESSION, [428](#)
- noMipiRxWorkaround
 - PixelUnpackerMipi10bParam, [410](#)
- nonMax3x3Fp32.h, [580](#)
- nonMax3x3Fp32Param, [409](#)
 - candPos, [409](#)

- nonMax3x3U8, [285](#)
 - SHAVE_SYM_EXPORT, [285](#)
 - svuNonMax3x3U8, [285](#)
- nonMax3x3U8.h, [580](#)
- nonMaxFp32, [284](#)
 - SHAVE_SYM_EXPORT, [284](#)
 - svuNonMax3x3Fp32, [284](#)
- normValue
 - svuRandNoiseFp16.c, [953](#)
- normalize
 - BoxFilter11x11Param, [332](#)
 - BoxFilter13x13Param, [332](#)
 - BoxFilter15x15Param, [333](#)
 - BoxFilter3x3Param, [333](#)
 - BoxFilter5x5Param, [334](#)
 - BoxFilter7x7Param, [334](#)
 - BoxFilter9x9Param, [335](#)
 - BoxFilterParam, [336](#)
- numCmxSlicesAvail
 - SippCmxBufferMapS, [431](#)
- numConsumers
 - SippOseS, [449](#)
 - SippSchEntS, [462](#)
- numDisp
 - SubpixelFilterParam, [474](#)
- numEntries
 - sSippCdmaQuS, [468](#)
- numFiltS
 - SippOseS, [449](#)
- numFractionalBits
 - SubpixelFilterParam, [475](#)
- numIBufs
 - SippFilterS, [437](#)
 - SippOseS, [449](#)
 - SippSchEntS, [462](#)
- numLineRuns
 - SippSchEntS, [463](#)
- numLines
 - sSchLineBufferS, [467](#)
- numLinesUsed
 - sSchedIBufUsageInfoS, [465](#)
- numMemRegions
 - SippPipelineS, [456](#)
- numOBufs
 - SippFilterS, [437](#)
 - SippOseS, [449](#)
 - SippSchEntS, [463](#)
- numSE
 - SippPipelineS, [456](#)
- numSWConsumers
 - sSchLineBufferS, [467](#)
- numShaves
 - CommInfo, [345](#)
- numVPools
 - tSippMCB, [480](#)
- nxtExeNo
 - SippPipelineS, [456](#)
- OTHER
 - sippHwCommon_ma2x5x.h, [734](#)
- O_BASE
 - sippDefines.h, [632](#)
- O_CFG
 - sippDefines.h, [632](#)
- O_CTX
 - sippDefines.h, [632](#)
- O_FC
 - sippDefines.h, [632](#)
- O_LS
 - sippDefines.h, [632](#)
- O_SHADOW_BASE
 - sippDefines.h, [632](#)
- oBuf
 - SippFilterS, [438](#)
- oBufAlloc
 - SippFilterS, [438](#)
- oBufId
 - HPadInfoS, [390](#)
- oBufIdx
 - sippOpipBufInfo, [448](#)
- oBufLatency
 - SippFilterSchedInfoS, [443](#)
- oBufs
 - SippFilterS, [438](#)
 - SippOseS, [449](#)
 - SippSchEntS, [463](#)
- OCR_STOP_GO
 - svuExtAfStats.c, [907](#)
 - svuExtStatsSatPixelsU32.c, [908](#)
- ONE_OVER_UINT32_MAX
 - svuRandNoise.c, [951](#)
- ORIENT_8BIT
 - sippHwCommon_ma2x5x.h, [731](#)
- OSEFiltCfg
 - SippOseS, [449](#)
- OSR_SWI_HALT
 - svuExtAfStats.c, [907](#)
 - svuExtStatsSatPixelsU32.c, [908](#)
- OUTPLANE_ORDER_BGR
 - sippHwBitfieldDefs.h, [705](#)
- OUTPLANE_ORDER_BRG
 - sippHwBitfieldDefs.h, [705](#)
- OUTPLANE_ORDER_GBR

- sippHwBitfieldDefs.h, 705
- OUTPLANE_ORDER_GRB
 - sippHwBitfieldDefs.h, 705
- OUTPLANE_ORDER_RGB
 - sippHwBitfieldDefs.h, 706
- OUTPLANE_ORDER_RGB
 - sippHwBitfieldDefs.h, 706
- offset
 - cvtColorChromaNV12Param, 368
 - cvtColorLumaNV12Param, 369
 - GreyDesatParam, 385
 - LutParam, 397
 - MixMedianParam, 408
- oldRunMask
 - SippPipelineS, 456
- ones
 - svuFast9ScoreCv.c, 911
- outLineDeficit
 - SippFilterS, 438
- outLineOffset
 - SippFilterS, 438
- outLinePtr
 - SippFilterS, 438
- outOff
 - ParentInfoS, 409
 - SippFilterS, 438
- outStatsBuffer
 - ExtStatsSatPixelsU32Param, 382
 - PpAf, 416
- outStatsBufferOutside
 - ExtStatsSatPixelsU32Param, 382
 - PpAf, 416
- output
 - StartBicubicParam, 470
- outputBuffer
 - SippFilterS, 438
- outputH
 - SippFilterS, 439
- outputHeight
 - sSchedIBufUsageInfoS, 465
- outputW
 - SippFilterS, 439
- p1
 - UndistortBParam, 483
- p2
 - UndistortBParam, 484
- POLY_BICUBIC
 - Polyphase FIR Scaler, 108
- POLY_BILINEAR
 - Polyphase FIR Scaler, 108
- POLY_LANCZOS
 - Polyphase FIR Scaler, 108
- POLY_MODE_ADVANCE
 - Polyphase FIR Scaler, 108
- POLY_MODE_AUTO
 - Polyphase FIR Scaler, 108
- POLY_PLANE_ALL
 - Polyphase FIR Scaler, 108
- POLY_PLANE_U
 - Polyphase FIR Scaler, 108
- POLY_PLANE_UV
 - Polyphase FIR Scaler, 108
- POLY_PLANE_V
 - Polyphase FIR Scaler, 108
- POLY_PLANE_Y
 - Polyphase FIR Scaler, 108
- P_BGR
 - sippHwCommon_ma2x5x.h, 731
- P_BRG
 - sippHwCommon_ma2x5x.h, 731
- P_GBR
 - sippHwCommon_ma2x5x.h, 731
- P_GRB
 - sippHwCommon_ma2x5x.h, 731
- P_RGB
 - sippHwCommon_ma2x5x.h, 731
- P_RGB
 - sippHwCommon_ma2x5x.h, 731
- PADDING
 - svuCannyEdgeDetection.c, 858
 - svuCornerMinEigenVal.c, 891
- PC/sippPalTypes.h
 - SIPP_ERROR_FATAL, 795
 - SIPP_TRACE_ANY, 797
 - SIPP_TRACE_API, 797
 - SIPP_TRACE_CORE, 797
 - SIPP_TRACE_HEAP, 797
 - SIPP_TRACE_MAX, 798
 - SIPP_TRACE_PAL, 798
- PC/sippPlatformAbstractionLayer.c
 - FALSE, 812
 - palTraceSerialiseSem, 813
 - sippPalCriticalSectionBegin, 813
 - sippPalCriticalSectionEnd, 813
 - sippPalFindHighestBit, 813
 - sippPalMemCompare, 813
 - sippPalMemcpy, 813
 - sippPalMemset, 813
 - sippPalTrace, 813
 - sippPalTraceInit, 813
 - TRUE, 812
- pCmxMap

SippPipelineS, [456](#)
 pCmxSliceRegionList
 SippCmxBufferMapS, [431](#)
 pControlPipePendData
 SIPP_ACCESS_SCHEDULER, [425](#)
 pEventData
 SIPP_ACCESS_SCHEDULER_QU_ENTR-
 Y, [427](#)
 PFL_DMA_DONE
 sippDefines.h, [632](#)
 PFL_SIPP_DONE
 sippDefines.h, [632](#)
 PFL_SIPP_EOF_DONE
 sippDefines.h, [633](#)
 PFL_SVU_DONE
 sippDefines.h, [633](#)
 pHeap
 tSippPhysicalPool, [481](#)
 PLANAR
 sippHwCommon_ma2x5x.h, [731](#)
 PLF_IS_FINALIZED
 sipp_ma2x5x.h, [597](#)
 pLineDesList
 DmaParam, [376](#)
 pNext
 SippHeapCB, [444](#)
 SippMemRegionListNode, [447](#)
 pNextChunkReg
 SippMemRegionListNode, [447](#)
 POLY_CFG
 sippHwBitfieldDefs.h, [706](#)
 POLY_CFG_DIM_IN
 sippHwBitfieldDefs.h, [706](#)
 POLY_CFG_DIM_OUT
 sippHwBitfieldDefs.h, [706](#)
 POLY_FRM_WIDTH_IN
 sippHwBitfieldDefs.h, [708](#)
 pOpipesch
 SippFilterS, [440](#)
 pPendData
 SIPP_ACCESS_SCHEDULER, [425](#)
 pPipe
 SippFilterS, [440](#)
 pPipelines
 tSippFramework, [479](#)
 PRE_FP16_GRAD
 sippHwCommon_ma2x5x.h, [731](#)
 PRE_U8_GRAD
 sippHwCommon_ma2x5x.h, [731](#)
 PROG_IO_BUFF
 sippDefines.h, [633](#)
 pRunAdditionalParam
 SippSchEntS, [463](#)
 pSE
 SippPipelineS, [457](#)
 pSIPP_HW_SESSION
 sippTypesPrivate.h, [834](#)
 pSippCmxBufferMap
 sippTypes.h, [827](#)
 pSippCurrHWPipe
 SIPP_HW_SESSION, [428](#)
 pSippCurrSvuPipe
 SIPP_HW_SESSION, [428](#)
 pSippFilter
 sippTypes.h, [827](#)
 pSippFilterSchedInfo
 sippTypes.h, [827](#)
 pSippHeapCB
 sippTypesPrivate.h, [834](#)
 pSippLoadedHWPipe
 SIPP_HW_SESSION, [428](#)
 pSippManagedBufSchedInfo
 sippSchTypes.h, [817](#)
 pSippMemRegionListNode
 sippTypesPrivate.h, [834](#)
 pSippPipeline
 sippTypes.h, [827](#)
 pSippPrevSvuPipe
 SIPP_HW_SESSION, [428](#)
 pSippVPhysMap
 sippTypes.h, [827](#)
 pVPoolListEnd
 tSippMCB, [480](#)
 pVPoolListStart
 tSippMCB, [480](#)
 pVirtPhysMap
 tSippMCB, [480](#)
 packColCombCCM
 sippHwDefs_ma2x5x.h, [737](#)
 sippUtils.c, [836](#)
 packConv3x3CCM
 sippHwDefs_ma2x5x.h, [737](#)
 sippUtils.c, [836](#)
 packConv5x5CCM
 sippHwDefs_ma2x5x.h, [737](#)
 sippUtils.c, [836](#)
 packLumaDnsGaussLut
 sippHwDefs_ma2x5x.h, [737](#)
 sippUtils.c, [836](#)
 padBayer5.h, [581](#)
 clampInLines, [581](#)
 SHAVE_SYM_EXPORT, [581](#)

- svuPadBayer5, [581](#)
- padBayer5Frame, [286](#)
 - clampInLines, [286](#)
 - SHAVE_SYM_EXPORT, [286](#)
 - svuPadBayer5Frame, [286](#)
- padBayer5Frame.h, [581](#)
- padBayer5Reference
 - svuPadBayer5.c, [946](#)
 - svuPadBayer5Frame.c, [946](#)
- padList
 - SippPipelineS, [456](#)
- padOffsets
 - HPadInfoS, [390](#)
- palTraceSerialiseSem
 - PC/sippPlatformAbstractionLayer.c, [813](#)
- parInfo
 - SippFilterS, [440](#)
- paramSz
 - UnitInfo, [484](#)
- params
 - SippFilterS, [439](#)
 - sSchedIBufUsageInfoS, [465](#)
- ParentInfo
 - sippTypes.h, [827](#)
- ParentInfoS, [409](#)
 - nPlanes, [409](#)
 - outOff, [409](#)
- parentKS
 - SippSchEntS, [463](#)
- parentKSMin
 - SippSchEntS, [463](#)
- parentLatenciesKnown
 - SippFilterSchedInfoS, [443](#)
- parentOBufIdx
 - SippFilterS, [440](#)
- parents
 - SippFilterS, [440](#)
- parentsKS
 - SippFilterS, [440](#)
- patchGapX
 - AeAwbStatsCfg, [330](#)
 - ExtStatsSatPixelsU32Param, [382](#)
 - StatsAwbSatPixelsParam, [472](#)
 - StatsAwbSatPixelsParamU32, [473](#)
- patchGapY
 - AeAwbStatsCfg, [330](#)
 - ExtStatsSatPixelsU32Param, [382](#)
 - StatsAwbSatPixelsParam, [472](#)
 - StatsAwbSatPixelsParamU32, [473](#)
- patchHeight
 - AeAwbStatsCfg, [330](#)
- ExtStatsSatPixelsU32Param, [382](#)
- PpAf, [416](#)
- StatsAwbSatPixelsParam, [472](#)
- StatsAwbSatPixelsParamU32, [474](#)
- patchWidth
 - AeAwbStatsCfg, [330](#)
 - ExtStatsSatPixelsU32Param, [382](#)
 - PpAf, [416](#)
 - StatsAwbSatPixelsParam, [472](#)
 - StatsAwbSatPixelsParamU32, [474](#)
- paxelNr
 - PpAf, [416](#)
- paxelsIntervalsHz
 - CombDecimStatsGainsParam, [344](#)
- paxelsIntervalsVert
 - CombDecimStatsGainsParam, [344](#)
- pfCallback
 - SippPipelineS, [456](#)
- pfRunAdditionalCheck
 - SippSchEntS, [463](#)
- pfnSippRunFrameReset
 - SippPipelineS, [456](#)
- pfnSippRunIterDone
 - SippPipelineS, [457](#)
- pfnSippRunNextIter
 - SippPipelineS, [457](#)
- pfnSippRuntime
 - SippPipelineS, [457](#)
- pfnSippRuntimeClaimHWResource
 - SippPipelineS, [457](#)
- pfnSippScheSetBufConsModels
 - SippPipelineS, [457](#)
- pfnSippSchedule
 - SippPipelineS, [457](#)
- pgSippHW
 - sippAccessScheduler.c, [607](#)
 - sippApi.c, [613](#)
 - sippHWSessionControl.c, [752](#)
 - sippIsr.c, [770](#)
 - sippPipeSessionControl.c, [807](#)
 - sippScheduleIsr.c, [817](#)
- physPoolMap
 - SippVPhysMapS, [464](#)
- pipeIdx
 - tSippMCB, [480](#)
- pipeSEId
 - SippSchEntS, [463](#)
- Pixel packer, [287](#)
 - SHAVE_SYM_EXPORT, [287](#)
 - svuPixelPacker10b, [287](#)
- Pixel Position, [291](#)

- SHAVE_SYM_EXPORT, 291
- svuPositionKernel, 291
- Pixel Unpacker, 288
 - SHAVE_SYM_EXPORT, 288
 - svuPixelUnpacker, 288
- Pixel Unpacker Mipi 10b, 289
 - SHAVE_SYM_EXPORT, 289
 - svuPixelUnpackerMipi10b, 289
- Pixel Unpacker WB, 290
 - SHAVE_SYM_EXPORT, 290
 - svuPixelUnpackerWB, 290
- pixelPacker10b
 - svuPixelPacker10b.c, 947
- pixelPacker10b.h, 582
- pixelPos
 - svuPositionKernel.c, 949
- pixelPosition
 - positionKernelParam, 414
- pixelUnpacker
 - svuPixelUnpacker.c, 947
- pixelUnpacker.h, 582
- pixelUnpackerMipi10b
 - svuPixelUnpackerMipi10b.c, 948
- pixelUnpackerMipi10b.h, 583
- PixelUnpackerMipi10bParam, 409
 - coefs, 410
 - lineNo, 410
 - noMipiRxWorkaround, 410
- PixelUnpackerParam, 410
 - shift, 410
- pixelUnpackerWB
 - svuPixelUnpackerWB.c, 948
- pixelUnpackerWB.h, 583
- PixelUnpackerWBParam, 411
 - awbCoef, 411
 - shift, 411
- pixelValue
 - positionKernelParam, 414
- pl
 - CommInfo, 345
- placeholder
 - SippManagedBufSchedInfo, 446
- plane
 - ChannelExtractParam, 338
- planeMode
 - PolyFirParam, 413
- planeStride
 - SippFilterS, 440
- pmemRegDescriptor
 - sippTypesPrivate.h, 834
- PolyFirParam, 411
 - autoType, 412
 - cfgReg, 412
 - clamp, 412
 - frmDimFlt, 412
 - frmDimPar, 412
 - horzCoefs, 413
 - horzD, 413
 - horzN, 413
 - kerSz, 413
 - mode, 413
 - planeMode, 413
 - vertCoefs, 413
 - vertD, 413
 - vertN, 413
- PolyModes
 - Polyphase FIR Scaler, 108
- PolyPlaneMode
 - Polyphase FIR Scaler, 108
- PolyScalerType
 - Polyphase FIR Scaler, 108
- Polyphase FIR Scaler
 - POLY_BICUBIC, 108
 - POLY_BILINEAR, 108
 - POLY_LANCZOS, 108
 - POLY_MODE_ADVANCE, 108
 - POLY_MODE_AUTO, 108
 - POLY_PLANE_ALL, 108
 - POLY_PLANE_U, 108
 - POLY_PLANE_UV, 108
 - POLY_PLANE_V, 108
 - POLY_PLANE_Y, 108
- Polyphase FIR Scaler, 107
 - PolyModes, 108
 - PolyPlaneMode, 108
 - PolyScalerType, 108
- popCmd
 - SippPipelineS, 457
- pos
 - tSippPhysicalPool, 481
- posInPaxel
 - PpAf, 416
- posOffset
 - ThresholdFilterParam, 476
- positionKernel.h, 584
- positionKernelParam, 414
 - maskAddr, 414
 - pixelPosition, 414
 - pixelValue, 414
 - status, 414
- posix_sem
 - Semaphore, 423

- Post
 - Semaphore, [423](#)
- PpAf, [415](#)
 - a_f1f2, [415](#)
 - afConfig, [415](#)
 - b_f1f2, [415](#)
 - f1Coeffs, [415](#)
 - f1Threshold, [415](#)
 - f2Coeffs, [416](#)
 - f2Threshold, [416](#)
 - firstPatchX, [416](#)
 - firstPatchY, [416](#)
 - in, [416](#)
 - initialSubtractionValue, [416](#)
 - mask, [416](#)
 - nPatchesX, [416](#)
 - nPatchesY, [416](#)
 - nSkipRows, [416](#)
 - outStatsBuffer, [416](#)
 - outStatsBufferOutside, [416](#)
 - patchHeight, [416](#)
 - patchWidth, [416](#)
 - paxelNr, [416](#)
 - posInPaxel, [416](#)
 - runNr, [416](#)
 - shaveNr, [416](#)
 - stackPointer, [416](#)
 - statsAf0, [416](#)
- predicted
 - CensusMatchingPyrParam, [338](#)
- priority
 - SIPP_PAL_THREAD, [430](#)
- ps
 - ExtStatsSatPixelsU32Param, [382](#)
 - SippHwBufS, [445](#)
 - SippHwIOBuf, [446](#)
- psSchLineBuffer
 - sippTypes.h, [827](#)
- psSippCMDQu
 - sippTypes.h, [828](#)
- psSippCMDQuEntry
 - sippTypes.h, [828](#)
- psSippCdmaQu
 - sippTypes.h, [827](#)
- psSippCdmaQuEntry
 - sippTypes.h, [828](#)
- ptMLPISStartCQCtrl
 - sippTypesPrivate.h, [834](#)
- ptRTStats
 - sippTypes.h, [828](#)
- ptSippMCB
 - sippTypes.h, [828](#)
- ptSippPhysicalPool
 - sippTypesPrivate.h, [834](#)
- ptSippPipelineSuper
 - sippTypesPrivate.h, [834](#)
- ptrChunkPos
 - SippFilterS, [441](#)
- ptrFilt
 - sippOpipeBufInfo, [448](#)
- ptrFiltKS
 - SippOseS, [449](#)
 - SippSchEntS, [463](#)
- purpleFlare, [292](#)
 - SHAVE_SYM_EXPORT, [292](#)
 - svuPurpleFlare, [292](#)
 - svuPurpleFlare.c, [950](#)
- purpleFlare.h, [584](#)
- PurpleFlareParam, [417](#)
 - strength, [417](#)
- pushCmd
 - SippPipelineS, [457](#)
- pyrDown.h, [585](#)
- Pyramid Downscale, [293](#)
 - SHAVE_SYM_EXPORT, [293](#)
 - svuPyrDown, [293](#)
- pyrdown
 - svuPyrDown.c, [951](#)
- qFlags
 - SIPP_PAL_QU, [429](#)
- Qu
 - SIPP_ACCESS_SCHEDULER_QU, [426](#)
- quEntry
 - sSippCMDQuS, [469](#)
- quNum
 - sSippCMDQuS, [469](#)
- quSize
 - sSippCMDQuS, [469](#)
- REV1
 - sippHwCommon_ma2x5x.h, [734](#)
- REV2
 - sippHwCommon_ma2x5x.h, [734](#)
- RS_CAN_RUN
 - sippSchTypes.h, [817](#)
- RS_CANNOT
 - sippSchTypes.h, [817](#)
- RS_DONT_KNOW
 - sippSchTypes.h, [817](#)
- RAW_BAD_PIX_CFG
 - sippHwBitfieldDefs.h, [710](#)
- RAW_CFG

- sippHwBitfieldDefs.h, [711](#)
- RAW_CFG_FORMAT
 - sippHwBitfieldDefs.h, [712](#)
- RAW_CFG_GAIN_MODE
 - sippHwBitfieldDefs.h, [712](#)
- RAW_CFG_SDC_EN
 - sippHwBitfieldDefs.h, [713](#)
- RAW_GRGB_DECAY
 - sippHwBitfieldDefs.h, [713](#)
- RAW_GRGB_PLATO
 - sippHwBitfieldDefs.h, [713](#)
- RAW_KERNEL_SIZE
 - sippHwCommon_ma2x5x.h, [731](#)
- RAW_PATCH_CFG
 - sippHwBitfieldDefs.h, [714](#)
- RAW_PATCH_START
 - sippHwBitfieldDefs.h, [715](#)
- RAW_STATS_PLANES
 - sippHwBitfieldDefs.h, [715](#)
- REV1_DEF
 - sippHwCommon_ma2x5x.h, [731](#)
- REV2_DEF
 - sippHwCommon_ma2x5x.h, [731](#)
- RGB to Chroma NV12 conversion, [213](#)
 - SHAVE_SYM_EXPORT, [213](#)
 - svuCvtColorRGBtoChromaNV12, [213](#)
- RGB to Luma conversion, [214](#)
- RGB to Luma NV12 conversion, [219](#)
 - SHAVE_SYM_EXPORT, [219](#)
 - svuCvtColorRGBtoLumaNV12, [219](#)
- RGB to UV conversion, [220](#)
 - SHAVE_SYM_EXPORT, [220](#)
 - svuCvtColorRGBtoUV, [220](#)
- RGB to UV420 conversion, [221](#)
 - SHAVE_SYM_EXPORT, [221](#)
 - svuCvtColorRGBtoUV420, [221](#)
- RGB to YUV422 conversion, [222](#)
 - svuCvtColorRGBToYUV422, [222](#)
- RGB(fp16) to Luma(u8) conversion, [211](#)
 - SHAVE_SYM_EXPORT, [211](#)
 - svuCvtColorRGBfp16ToLumaU8, [211](#)
- RGB(fp16) to UV420(u8) conversion, [212](#)
 - SHAVE_SYM_EXPORT, [212](#)
 - svuCvtColorRGBfp16ToUV420U8, [212](#)
- RGG
 - sippHwCommon_ma2x5x.h, [731](#)
- rPtr
 - DmaTaskList, [376](#)
- RTEMS/src/leon/sippPlatformAbstractionLayer.c
 - sippPalCriticalSectionBegin, [814](#)
 - sippPalCriticalSectionEnd, [814](#)
 - sippPalFindHighestBit, [814](#)
 - sippPalMemCompare, [814](#)
 - sippPalMemcpy, [814](#)
 - sippPalMemset, [814](#)
 - sippPalPrintInt, [815](#)
 - sippPalQuAttach, [815](#)
 - sippPalQuCreate, [815](#)
 - sippPalQuDestroy, [815](#)
 - sippPalQuPost, [815](#)
 - sippPalQuReceive, [815](#)
 - sippPalThreadCreate, [815](#)
 - sippPalThreadTerminate, [815](#)
 - sippPalTrace, [815](#)
 - sippPalTraceInit, [815](#)
- randNoise.h, [585](#)
- randNoiseFp16
 - svuRandNoiseFp16.c, [952](#)
- randNoiseFp16.h, [586](#)
- RandNoiseFp16Param, [417](#)
 - strength, [417](#)
- RandNoiseParam, [417](#)
 - strength, [418](#)
- Random Noise, [294](#)
 - SHAVE_SYM_EXPORT, [294](#)
 - svuGenNoise, [294](#)
- Random Noise (high speed), [295](#)
 - SHAVE_SYM_EXPORT, [295](#)
 - svuGenNoiseFp16, [295](#)
- rangeLut
 - ChromaBlkParam, [341](#)
- Raw, [98](#)
- RawParam, [418](#)
 - afF1coefs, [419](#)
 - afF2coefs, [419](#)
 - afMinThresh, [419](#)
 - afPatchCfg, [419](#)
 - afPatchStart, [419](#)
 - afStatsBase, [420](#)
 - afSubtract, [420](#)
 - badPixCfg, [420](#)
 - cfg, [420](#)
 - frmDim, [420](#)
 - gainSat, [420](#)
 - grgbDecay, [420](#)
 - grgbPlat, [420](#)
 - histLumaBase, [420](#)
 - histRgbBase, [420](#)
 - statsBase, [420](#)
 - statsFrmDim, [421](#)
 - statsPatchCfg, [421](#)
 - statsPatchSkip, [421](#)

- statsPatchStart, [421](#)
- statsPlanes, [421](#)
- statsThresh, [421](#)
- recordParentKS
 - SippSchEntS, [464](#)
- regionAddr
 - SippMemRegionListNode, [447](#)
- regionOffset
 - SippMemRegion, [446](#)
- regionSize
 - SippMemRegion, [446](#)
 - SippMemRegionListNode, [447](#)
- regionUsed
 - SippMemRegion, [446](#)
 - SippMemRegionListNode, [447](#)
- regionUsedPtr
 - SippMemRegionListNode, [448](#)
- Revision
 - sippHwCommon_ma2x5x.h, [734](#)
- rgnStop01
 - UsmParam, [485](#)
- rgnStop23
 - UsmParam, [486](#)
- rightrot
 - svuFast9ScoreCv.c, [912](#)
- rtemsQuId
 - SIPP_PAL_QU, [429](#)
- rtemsRWQuId
 - SIPP_PAL_QU, [429](#)
- rtemsThread
 - SIPP_PAL_THREAD, [430](#)
- runAddSchedCheck
 - SippPipelineS, [457](#)
- runFullSearch
 - sSchedIBufUsageInfoS, [465](#)
- runNr
 - CombDecimStatsGainsParam, [344](#)
 - ExtStatsSatPixelsU32Param, [382](#)
 - PpAf, [416](#)
 - StatsAwbSatPixelsParam, [472](#)
 - StatsAwbSatPixelsParamU32, [474](#)
- runPadCheck
 - SippSchEntS, [464](#)
- RunStatus
 - sippSchTypes.h, [817](#)
- rxID
 - MipiRxLoopbackParam, [402](#)
- s16
 - sippBaseTypes.h, [616](#)
- s32
 - sippBaseTypes.h, [616](#)

- s64
 - sippBaseTypes.h, [616](#)
- s8
 - sippBaseTypes.h, [616](#)
- SIPP_PIPE_ACTIVE
 - sippSessionControl.h, [818](#)
- SIPP_PIPE_END_SESSION
 - sippSessionControl.h, [818](#)
- SIPP_PIPE_WAIT_FINALISE
 - sippSessionControl.h, [818](#)
- SIPP_PIPE_WAIT_INIT
 - sippSessionControl.h, [818](#)
- SCALE
 - svuConvYuv444.c, [890](#)
- SCALED_MAGN_16BIT
 - sippHwCommon_ma2x5x.h, [731](#)
- SCALED_MAGN_8BIT
 - sippHwCommon_ma2x5x.h, [731](#)
- SECTION
 - sippPlatform_ma2x5x.h, [809](#)
- SET_HW_PEND_LIST
 - sippAccessSchedulerTypes.h, [610](#)
- SET_REG_WORD
 - svuExtAfStats.c, [907](#)
 - svuExtStatsSatPixelsU32.c, [908](#)
- SET_SW_PEND_LIST
 - sippAccessSchedulerTypes.h, [610](#)
- SHADOW
 - sippHwCommon_ma2x5x.h, [731](#)
- SHARPEN_CFG
 - sippHwBitfieldDefs.h, [716](#)
- SHARPEN_CFG_CLAMP
 - sippHwBitfieldDefs.h, [716](#)
- SHARPEN_CFG_MODE
 - sippHwBitfieldDefs.h, [717](#)
- SHARPEN_COEF0_CFG
 - sippHwBitfieldDefs.h, [717](#)
- SHARPEN_COEF0_MASK
 - sippHwBitfieldDefs.h, [717](#)
- SHARPEN_COEF0_SHIFT
 - sippHwBitfieldDefs.h, [717](#)
- SHARPEN_COEF1_CFG
 - sippHwBitfieldDefs.h, [717](#)
- SHARPEN_COEF1_MASK
 - sippHwBitfieldDefs.h, [717](#)
- SHARPEN_COEF1_SHIFT
 - sippHwBitfieldDefs.h, [717](#)
- SHARPEN_COEF2_CFG
 - sippHwBitfieldDefs.h, [717](#)
- SHARPEN_COEF2_MASK
 - sippHwBitfieldDefs.h, [717](#)

- SHARPEN_COEF2_SHIFT
 - sippHwBitfieldDefs.h, [717](#)
- SHARPEN_COEF3_CFG
 - sippHwBitfieldDefs.h, [717](#)
- SHARPEN_COEF3_MASK
 - sippHwBitfieldDefs.h, [717](#)
- SHARPEN_COEF3_SHIFT
 - sippHwBitfieldDefs.h, [717](#)
- SHARPEN_COEFS01_CFG
 - sippHwBitfieldDefs.h, [717](#)
- SHARPEN_COEFS23_CFG
 - sippHwBitfieldDefs.h, [718](#)
- SHARPEN_LIMITS_CFG
 - sippHwBitfieldDefs.h, [718](#)
- SHAVE_MAIN
 - sippCoreApi.c, [624](#)
 - sippPlatform_ma2x5x.h, [809](#)
 - sippShave.c, [819](#)
 - sippShaveIf.c, [820](#)
- SHAVE_SYM_EXPORT
 - 12Bpp to 8Bpp conversion, [185](#)
 - Absolute difference, [115](#)
 - Accumulate Square, [116](#)
 - Accumulate Weighted, [117](#)
 - AF Stats, [233](#)
 - Arbitrary Downscale, [311](#)
 - Arithmetic, [126](#)
 - Arithmetic addition, [118](#)
 - Arithmetic addition with mask, [119](#)
 - Arithmetic subtraction fp16, [127](#)
 - Arithmetic subtraction with mask, [128](#)
 - Average, [129](#)
 - bilateral5x5, [130](#)
 - Bitwise And, [131](#)
 - Bitwise And with mask, [132](#)
 - Bitwise Not, [133](#)
 - Bitwise Or, [134](#)
 - Bitwise Or with mask, [135](#)
 - Bitwise Xor, [136](#)
 - Bitwise Xor with mask, [137](#)
 - Box Filter 11x11, [140](#)
 - Box Filter 13x13, [141](#)
 - Box Filter 15x15, [142](#)
 - Box Filter 3x3, [143](#)
 - Box Filter 5x5, [144](#)
 - Box Filter 7x7, [145](#)
 - Box Filter 9x9, [146](#)
 - Canny Edge Detection, [147](#)
 - censusMatching16, [148](#)
 - censusMatching32, [149](#)
 - censusMatching64, [150](#)
 - censusMatching65, [151](#)
 - censusMatchingPyr, [152](#)
 - censusMin16, [153](#)
 - censusMin64, [154](#)
 - censusMin65, [155](#)
 - censusMin7, [156](#)
 - CensusTransform5x5, [157](#)
 - channelExtract, [158](#)
 - Chroma Block, [159](#)
 - combDecimDemosaicAwbGains.h, [513](#)
 - combDecimDemosaicAwbGainsStats.h, [514](#)
 - Contrast, [160](#)
 - Convert 16bpp To 8bpp, [183](#)
 - Convert F16 To U8, [184](#)
 - Convert Fp16 to U16, [186](#)
 - Convert to YUV444, [201](#)
 - Convert U16 to Fp16, [187](#)
 - Convert U8 To F16, [188](#)
 - Convolution 11x11, [161](#)
 - Convolution 15x1, [162](#)
 - Convolution 1x15, [163](#)
 - Convolution 1x5, [164](#)
 - Convolution 1x5 Fp16ToFp16, [165](#)
 - Convolution 1x7, [166](#)
 - Convolution 1x7 Fp16ToFp16, [167](#)
 - Convolution 1x9, [168](#)
 - Convolution 3x3, [169](#)
 - Convolution 3x3 Fp16ToFp16, [170](#)
 - Convolution 5x1, [171](#)
 - Convolution 5x1 Fp16ToFp16, [172](#)
 - Convolution 5x5, [173](#)
 - Convolution 5x5 Fp16ToFp16, [174](#)
 - Convolution 7x1, [175](#)
 - Convolution 7x1 Fp16ToFp16, [176](#)
 - Convolution 7x7, [177](#)
 - Convolution 7x7 Fp16ToFp16, [178](#)
 - Convolution 7x7 Fp16ToU8, [179](#)
 - Convolution 9x1, [180](#)
 - Convolution 9x9, [181](#)
 - Convolution 9x9 Fp16ToFp16, [182](#)
 - Convolution Separable 11x11, [191](#)
 - Convolution Separable 11x11 Fp16ToFp16, [192](#)
 - Convolution Separable 3x3, [193](#)
 - Convolution Separable 3x3 Fp16ToFp16, [194](#)
 - Convolution Separable 5x5, [195](#)
 - Convolution Separable 5x5 Fp16ToFp16, [196](#)
 - Convolution Separable 7x7, [197](#)
 - Convolution Separable 7x7 Fp16ToFp16, [198](#)
 - Convolution Separable 9x9, [199](#)
 - Convolution Separable 9x9 Fp16ToFp16, [200](#)

Copy, [202](#)
 Corner Min Eigenvalue, [203](#)
 Corner Min Eigenvalue Patched, [204](#)
 CV, [208](#)
 Dilate 3x3, [225](#)
 Dilate 5x5, [226](#)
 Dilate 7x7, [227](#)
 disp2depth.h, [548](#)
 Downscale by 2, [298](#)
 Downscale by 2 (fp16/fp16), [300](#)
 Downscale by 2 (fp16/u8), [299](#)
 Equalize Histogram, [229](#)
 Erode 3x3, [230](#)
 Erode 5x5, [231](#)
 Erode 7x7, [232](#)
 extStatsSatPixelsU32.h, [553](#)
 Fast9M2, [234](#)
 Fast9ScoreCv, [235](#)
 Gauss Blur, [236](#)
 GaussHx2, [237](#)
 GaussHx2_fp16, [238](#)
 GaussVx2, [239](#)
 GaussVx2_fp16, [240](#)
 Generate Chroma, [241](#)
 Generate Chroma with subsampling, [242](#)
 Generate Luma U8 to Fp16, [246](#)
 Generate Reference for Luma Denoise, [243](#)
 Generate Reference for Luma Denoise(fp16 input), [244](#)
 Generic Box Filter, [139](#)
 Generic Convolution, [190](#)
 Generic Dilate, [228](#)
 greyDesat, [247](#)
 hammingDistance, [248](#)
 harrisResponse, [249](#)
 Histogram, [250](#)
 histogramStat, [251](#)
 Homography, [252](#)
 Integral Image Square Sum(f32), [253](#)
 Integral Image Square Sum(U32), [254](#)
 Integral Image Sum(f32), [255](#)
 Integral Image Sum(U16toU32), [256](#)
 Integral Image Sum(U32), [257](#)
 interpolatePixelBilinear, [258](#)
 ISP, [218](#)
 Lanczos Downscale by 2 (6 taps), [301](#)
 Lanczos Downscale by 2 (7 taps), [302](#)
 Lanczos Horizontal Upscale by 2, [308](#)
 Lanczos Upscale by 2, [309](#)
 Lanczos Vertical Upscale by 2, [310](#)
 Laplacian 3x3, [259](#)
 Laplacian 3x3 Fp16 To Fp16, [313](#)
 Laplacian 5x5, [260](#)
 Laplacian 5x5 Fp16 To Fp16, [261](#)
 Laplacian 7x7, [262](#)
 Laplacian 7x7 Fp16 To Fp16, [263](#)
 localMaxMin3x3_fp16, [264](#)
 localTM, [265](#)
 Low Level Correction, [266](#)
 Low Level Correction on Multiple Planes, [268](#)
 Luma Blur, [245](#), [269](#)
 LUT 10 to 16, [270](#)
 LUT 10 to 8, [271](#)
 LUT 12 to 16, [272](#)
 LUT 12 to 8, [273](#)
 LUT 16 to 8, [275](#)
 LUT 8 to 8, [274](#)
 maxTest3x3_fp16, [276](#)
 meanStdDev, [277](#)
 Min/Max Value, [279](#)
 Min/Max Value Position, [278](#)
 minTest3x3_fp16, [280](#)
 Mix Median, [281](#)
 monoImbalance.h, [579](#)
 Negative, [283](#)
 nonMax3x3U8, [285](#)
 nonMaxFp32, [284](#)
 NV21 to RGB conversion, [210](#)
 padBayer5.h, [581](#)
 padBayer5Frame, [286](#)
 Pixel packer, [287](#)
 Pixel Position, [291](#)
 Pixel Unpacker, [288](#)
 Pixel Unpacker Mipi 10b, [289](#)
 Pixel Unpacker WB, [290](#)
 purpleFlare, [292](#)
 Pyramid Downscale, [293](#)
 Random Noise, [294](#)
 Random Noise (high speed), [295](#)
 RGB to Chroma NV12 conversion, [213](#)
 RGB to Luma NV12 conversion, [219](#)
 RGB to UV conversion, [220](#)
 RGB to UV420 conversion, [221](#)
 RGB to YUV422 conversion, [222](#)
 RGB(fp16) to Luma(u8) conversion, [211](#)
 RGB(fp16) to UV420(u8) conversion, [212](#)
 scharr_fp16, [312](#)
 Sobel, [314](#)
 startBicubic.h, [840](#)
 statsAwbSatPixels.h, [840](#)
 statsAwbSatPixelsU32.h, [841](#)
 Sum of Absolute Differences 11x11, [296](#)

Sum of Absolute Differences 5x5, [297](#)
 Sum of Squared Differences 11x11, [315](#)
 Sum of Squared Differences 5x5, [316](#)
 Sum of Squared Differences 7x7, [318](#)
 Sum of Squared Differences 7x7 (U8 to U32), [317](#)
 Threshold, [320](#)
 Threshold Binary Range, [321](#)
 Threshold Binary U8, [322](#)
 ThresholdFilter, [323](#)
 Undistort, [324](#)
 Upscale by 2, [303](#)
 Upscale by 2 with phases 0.25 and 0.75 fp16 to fp16, [304](#)
 Upscale by 2 with phases 0.25 and 0.75 fp16 to u8, [306](#)
 Upscale by 2 with phases 0.25 and 0.75 u16 to u16, [305](#)
 Upscale by 2 with phases 0.25 and 0.75 u8 to u8, [307](#)
 White Balance Bayer GBRG, [325](#)
 White Balance RGB, [326](#)
 XY Generator, [327](#)
 YUV to RGB conversion, [224](#)
 YUV400 to YUV422 conversion, [189](#)
 YUV422 to RGB conversion, [223](#)
 SIGMA_DNS_CFG
 sippHwBitfieldDefs.h, [719](#)
 SIGMA_DNS_CFG_NF
 sippHwBitfieldDefs.h, [720](#)
 SIGMA_KERNEL_SIZE
 sippHwCommon_ma2x5x.h, [732](#)
 SIPP_AF_STATS_ID
 sippHwIds.h, [741](#)
 SIPP_API_TL_DEBUG
 PC/sippPalTypes.h, [794](#)
 SIPP_API_TL_ERROR
 PC/sippPalTypes.h, [794](#)
 SIPP_API_TL_FUNC
 BM/src/leon/sippPalTypes.h, [787](#)
 PC/sippPalTypes.h, [794](#)
 SIPP_API_TL_INFO
 BM/src/leon/sippPalTypes.h, [787](#)
 PC/sippPalTypes.h, [794](#)
 SIPP_AUTO
 sipp_ma2x5x.h, [598](#)
 SIPP_CBL_OFFSET
 sippHwCommon_ma2x5x.h, [732](#)
 SIPP_CC_3DLUT_ID
 sippHwIds.h, [741](#)
 SIPP_CC_CHROMA_ID
 sippHwIds.h, [741](#)
 SIPP_CC_ID
 sippHwIds.h, [741](#)
 SIPP_CC_ID_MASK
 sippHwIds.h, [741](#)
 SIPP_CC_NLBRC
 sippHwCommon_ma2x5x.h, [732](#)
 SIPP_CDMA_AGENT_NO
 sippCfg.h, [618](#)
 SIPP_CDMA_INT_NO
 sippCfg.h, [618](#)
 SIPP_CGEN_ID
 sippHwIds.h, [741](#)
 SIPP_CGEN_ID_MASK
 sippHwIds.h, [741](#)
 SIPP_CHROMA_ID
 sippHwIds.h, [742](#)
 SIPP_CHROMA_NLBRC
 sippHwCommon_ma2x5x.h, [732](#)
 SIPP_CMX_POOL_SZ
 sippCfg.h, [618](#)
 SIPP_CONV_ID
 sippHwIds.h, [742](#)
 SIPP_CONV_ID_MASK
 sippHwIds.h, [742](#)
 SIPP_CONV_NLBRC
 sippHwCommon_ma2x5x.h, [732](#)
 SIPP_CORE_TL_FUNC
 PC/sippPalTypes.h, [795](#)
 SIPP_CORE_TL_INFO
 PC/sippPalTypes.h, [795](#)
 SIPP_CQ_ADD_WRITE
 sippHwCommon.c, [725](#)
 SIPP_CROP
 sipp_ma2x5x.h, [598](#)
 SIPP_CS_MASK
 sippHwCommon_ma2x5x.h, [732](#)
 SIPP_CS_OFFSET
 sippHwCommon_ma2x5x.h, [732](#)
 SIPP_CTXUP_BIT
 sippHwCommon_ma2x5x.h, [732](#)
 SIPP_DBYR_ID
 sippHwIds.h, [742](#)
 SIPP_DBYR_ID_MASK
 sippHwIds.h, [742](#)
 SIPP_DBYR_LUMA_ID
 sippHwIds.h, [742](#)
 SIPP_DBYR_NLBRC
 sippHwCommon_ma2x5x.h, [732](#)
 SIPP_DDR_POOL_SZ
 sippCfg.h, [618](#)

SIPP_DMA_ID
 sippHwIds.h, [742](#)
 SIPP_DOG_NLBRC
 sippHwCommon_ma2x5x.h, [732](#)
 SIPP_DOGL_ID
 sippHwIds.h, [742](#)
 SIPP_DOGL_ID_MASK
 sippHwIds.h, [742](#)
 SIPP_DOGL_NLBRC
 sippHwCommon_ma2x5x.h, [732](#)
 SIPP_EDGE_OP_ID
 sippHwIds.h, [742](#)
 SIPP_ERROR_FATAL
 BM/src/leon/sippPalTypes.h, [787](#)
 PC/sippPalTypes.h, [795](#)
 RTEMS/src/leon/sippPalTypes.h, [802](#)
 SIPP_ERROR_WARNING
 PC/sippPalTypes.h, [795](#)
 SIPP_FAKE_ID
 sippHwIds.h, [742](#)
 SIPP_FO_MASK
 sippHwCommon_ma2x5x.h, [732](#)
 SIPP_FO_OFFSET
 sippHwCommon_ma2x5x.h, [732](#)
 SIPP_GCHR_NLBRC
 sippHwCommon_ma2x5x.h, [732](#)
 SIPP_HARRIS_ID
 sippHwIds.h, [743](#)
 SIPP_HARRIS_NLBRC
 sippHwCommon_ma2x5x.h, [732](#)
 SIPP_HCB_SIZE
 sippTypesPrivate.h, [833](#)
 SIPP_HEAP_TL_FUNC
 PC/sippPalTypes.h, [796](#)
 SIPP_HEAP_TL_INFO
 PC/sippPalTypes.h, [796](#)
 SIPP_HW_SESSION, [427](#)
 eState, [428](#)
 isrSetup, [428](#)
 noIntBarSwitchPend, [428](#)
 pSippCurrHWPipe, [428](#)
 pSippCurrSvuPipe, [428](#)
 pSippLoadedHWPipe, [428](#)
 pSippPrevSvuPipe, [428](#)
 uHWFeatures, [428](#)
 uNumCurrHwPipes, [428](#)
 useIntBar, [428](#)
 SIPP_IC_MASK
 sippHwCommon_ma2x5x.h, [732](#)
 SIPP_IC_OFFSET
 sippHwCommon_ma2x5x.h, [732](#)

SIPP_IMGDIM_MASK
 sippHwCommon_ma2x5x.h, [732](#)
 SIPP_IMGDIM_SIZE
 sippHwCommon_ma2x5x.h, [732](#)
 SIPP_INCDEC_BIT
 sippHwCommon_ma2x5x.h, [732](#)
 SIPP_IR_MASK
 sippHwCommon_ma2x5x.h, [733](#)
 SIPP_IR_OFFSET
 sippHwCommon_ma2x5x.h, [733](#)
 SIPP_KL_MASK
 sippHwCommon_ma2x5x.h, [733](#)
 SIPP_LS_MASK
 sippHwCommon_ma2x5x.h, [733](#)
 SIPP_LSC_GM_ID
 sippHwIds.h, [743](#)
 SIPP_LSC_ID
 sippHwIds.h, [743](#)
 SIPP_LSC_ID_MASK
 sippHwIds.h, [743](#)
 SIPP_LSC_NLBRC
 sippHwCommon_ma2x5x.h, [733](#)
 SIPP_LUMA_C4LUT_ID
 sippHwIds.h, [743](#)
 SIPP_LUMA_HIST_ID
 sippHwIds.h, [743](#)
 SIPP_LUMA_ID
 sippHwIds.h, [743](#)
 SIPP_LUMA_ID_MASK
 sippHwIds.h, [743](#)
 SIPP_LUMA_NLBRC
 sippHwCommon_ma2x5x.h, [733](#)
 SIPP_LUMA_REF_ID
 sippHwIds.h, [744](#)
 SIPP_LUT_ID
 sippHwIds.h, [744](#)
 SIPP_LUT_ID_MASK
 sippHwIds.h, [744](#)
 SIPP_LUT_LOAD_ID
 sippHwIds.h, [744](#)
 SIPP_LUT_NLBRC
 sippHwCommon_ma2x5x.h, [733](#)
 SIPP_MAX_ID
 sippHwIds.h, [744](#)
 SIPP_MBIN
 sippPlatform_ma2x5x.h, [809](#)
 SIPP_MED_ID
 sippHwIds.h, [744](#)
 SIPP_MED_ID_MASK
 sippHwIds.h, [744](#)
 SIPP_MED_LUMA_ID

- sippHwIds.h, 744
- SIPP_MED_NLBRC
 - sippHwCommon_ma2x5x.h, 733
- SIPP_MEMPOOL_CMx
 - sippMem.c, 772
- SIPP_MEMPOOL_DDR
 - sippMem.c, 772
- SIPP_MIPI_RX0_ID
 - sippHwIds.h, 744
- SIPP_MIPI_RX1_ID
 - sippHwIds.h, 744
- SIPP_MIPI_RX2_ID
 - sippHwIds.h, 745
- SIPP_MIPI_RX3_ID
 - sippHwIds.h, 745
- SIPP_MIPI_TX0_ID
 - sippHwIds.h, 745
- SIPP_MIPI_TX1_ID
 - sippHwIds.h, 745
- SIPP_NL_MASK
 - sippHwCommon_ma2x5x.h, 733
- SIPP_NL_OFFSET
 - sippHwCommon_ma2x5x.h, 733
- SIPP_NLBWC
 - sippHwCommon_ma2x5x.h, 733
- SIPP_NP_MASK
 - sippHwCommon_ma2x5x.h, 733
- SIPP_NP_OFFSET
 - sippHwCommon_ma2x5x.h, 733
- SIPP_NUM_SVUS
 - sippCfg.h, 619
- SIPP_OCTOPUS_MASK
 - sippOPipeSchedulingEntity.c, 779
- SIPP_OF_MASK
 - sippHwCommon_ma2x5x.h, 733
- SIPP_OF_OFFSET
 - sippHwCommon_ma2x5x.h, 733
- SIPP_OPIPE_ID
 - sippHwIds.h, 745
- SIPP_PAL_QU, 429
 - elementsInQ, 429
 - maxQElements, 429
 - qFlags, 429
 - rtemsQuId, 429
 - rtemsRWQuId, 429
- SIPP_PAL_QU_ID
 - RTEMS/src/leon/sippPalTypes.h, 806
- SIPP_PAL_QU_MSG
 - RTEMS/src/leon/sippPalTypes.h, 806
- SIPP_PAL_THREAD, 429
 - args, 430

- priority, 430
- rtemsThread, 430
- stackSize, 430
- SIPP_PAL_TIMER_ID
 - PC/sippPalTypes.h, 798
- SIPP_PAL_TL_DEBUG
 - PC/sippPalTypes.h, 796
- SIPP_PAL_TL_ERROR
 - PC/sippPalTypes.h, 796
- SIPP_PAL_TL_FUNC
 - BM/src/leon/sippPalTypes.h, 788
 - PC/sippPalTypes.h, 796
- SIPP_PAL_TL_INFO
 - BM/src/leon/sippPalTypes.h, 789
 - PC/sippPalTypes.h, 796
- SIPP_PC
 - sippPlatform_ma2x5x.h, 810
- SIPP_PIPE_STATE
 - sippSessionControl.h, 818
- SIPP_PS_MASK
 - sippHwCommon_ma2x5x.h, 733
- SIPP_RAW_ID
 - sippHwIds.h, 745
- SIPP_RAW_ID_MASK
 - sippHwIds.h, 745
- SIPP_RAW_NLBRC
 - sippHwCommon_ma2x5x.h, 733
- SIPP_REQ_SW_VIEW
 - sipp_ma2x5x.h, 598
- SIPP_RESERVED_ID
 - sippHwIds.h, 745
- SIPP_RESIZE
 - sipp_ma2x5x.h, 598
- SIPP_RGB_HIST_ID
 - sippHwIds.h, 745
- SIPP_SB_MASK
 - sippHwCommon_ma2x5x.h, 733
- SIPP_SC_MASK
 - sippHwCommon_ma2x5x.h, 733
- SIPP_SC_OFFSET
 - sippHwCommon_ma2x5x.h, 733
- SIPP_SHARPEN_ID
 - sippHwIds.h, 745
- SIPP_SHIFT_PLANES
 - sipp_ma2x5x.h, 598
- SIPP_SIGMA_ID
 - sippHwIds.h, 746
- SIPP_SIGMA_NLBRC
 - sippHwCommon_ma2x5x.h, 734
- SIPP_SL_OFFSET
 - sippHwCommon_ma2x5x.h, 734

SIPP_SS_MASK
 sippHwCommon_ma2x5x.h, [734](#)
 SIPP_SS_OFFSET
 sippHwCommon_ma2x5x.h, [734](#)
 SIPP_START_BIT
 sippHwCommon_ma2x5x.h, [734](#)
 SIPP_STATS_ID
 sippHwIds.h, [746](#)
 SIPP_STATS_MASK
 sippHwIds.h, [746](#)
 SIPP_SVU_ID
 sippHwIds.h, [746](#)
 SIPP_THREAD_NAME
 sippCfg.h, [619](#)
 SIPP_TRACE_ANY
 BM/src/leon/sippPalTypes.h, [790](#)
 PC/sippPalTypes.h, [797](#)
 RTEMS/src/leon/sippPalTypes.h, [805](#)
 SIPP_TRACE_API
 BM/src/leon/sippPalTypes.h, [790](#)
 PC/sippPalTypes.h, [797](#)
 RTEMS/src/leon/sippPalTypes.h, [805](#)
 SIPP_TRACE_CORE
 BM/src/leon/sippPalTypes.h, [790](#)
 PC/sippPalTypes.h, [797](#)
 RTEMS/src/leon/sippPalTypes.h, [805](#)
 SIPP_TRACE_FLAGS, [430](#)
 Flags, [430](#)
 SIPP_TRACE_HEAP
 BM/src/leon/sippPalTypes.h, [790](#)
 PC/sippPalTypes.h, [797](#)
 RTEMS/src/leon/sippPalTypes.h, [805](#)
 SIPP_TRACE_LEVEL_1
 PC/sippPalTypes.h, [797](#)
 SIPP_TRACE_LEVEL_2
 PC/sippPalTypes.h, [797](#)
 SIPP_TRACE_LEVEL_3
 PC/sippPalTypes.h, [797](#)
 SIPP_TRACE_LEVEL_4
 PC/sippPalTypes.h, [797](#)
 SIPP_TRACE_LEVEL_5
 PC/sippPalTypes.h, [797](#)
 SIPP_TRACE_LEVEL_6
 PC/sippPalTypes.h, [797](#)
 SIPP_TRACE_MAX
 BM/src/leon/sippPalTypes.h, [790](#)
 PC/sippPalTypes.h, [798](#)
 RTEMS/src/leon/sippPalTypes.h, [805](#)
 SIPP_TRACE_PAL
 BM/src/leon/sippPalTypes.h, [791](#)
 PC/sippPalTypes.h, [798](#)
 RTEMS/src/leon/sippPalTypes.h, [805](#)
 SIPP_TRACE_RUNTIME
 PC/sippPalTypes.h, [798](#)
 SIPP_UPFIRDN0_ID
 sippHwIds.h, [746](#)
 SIPP_UPFIRDN1_ID
 sippHwIds.h, [746](#)
 SIPP_UPFIRDN2_ID
 sippHwIds.h, [746](#)
 SIPP_UPFIRDN_ID
 sippHwBitFields.h, [723](#)
 sLaplacian3x3Fp16ToFp16
 svuLaplacian3x3Fp16ToFp16.c, [963](#)
 sLaplacian3x3Fp16ToFp16.h, [837](#)
 sLaplacian5x5Fp16ToFp16Implementation
 svuLaplacian5x5Fp16ToFp16.c, [930](#)
 sLaplacian5x5Implementation
 svuLaplacian5x5.c, [929](#)
 sLaplacian7x7Fp16ToFp16Implementation
 svuLaplacian7x7Fp16ToFp16.c, [931](#)
 sLaplacian7x7Implementation
 svuLaplacian7x7.c, [930](#)
 sSchLineBuffer
 sippTypes.h, [828](#)
 sSchLineBufferS, [466](#)
 allocReq, [466](#)
 bottomLineReplication, [466](#)
 filtOutHeight, [466](#)
 hPadding, [466](#)
 hwInputBufId, [466](#)
 hwOutputBufId, [467](#)
 internalFillLevel, [467](#)
 latency, [467](#)
 manageReq, [467](#)
 mostRecentLine, [467](#)
 numLines, [467](#)
 numSWConsumers, [467](#)
 sSchedIBufUsageInfoS, [464](#)
 consumptionLatency, [465](#)
 funcAsk, [465](#)
 hKerSz, [465](#)
 iPadLines, [465](#)
 KSDelta, [465](#)
 lastAskLineNo, [465](#)
 numLinesUsed, [465](#)
 outputHeight, [465](#)
 params, [465](#)
 runFullSearch, [465](#)
 sSippCMDQu
 sippTypes.h, [829](#)
 sSippCMDQuEntry

- sippTypes.h, [829](#)
- sSippCMDQuEntryS, [468](#)
 - addr, [469](#)
 - value, [469](#)
- sSippCMDQuS, [469](#)
 - quEntry, [469](#)
 - quNum, [469](#)
 - quSize, [469](#)
- sSippCdmaQu
 - sippTypes.h, [828](#)
- sSippCdmaQuEntry
 - sippTypes.h, [829](#)
- sSippCdmaQuEntryS, [467](#)
 - addr, [468](#)
 - value, [468](#)
- sSippCdmaQuS, [468](#)
 - entry, [468](#)
 - numEntries, [468](#)
 - size, [468](#)
- SUB_SATURATE
 - lowLvlCorrMultiplePlanes.c, [571](#)
 - svuLowLvlCorr.c, [933](#)
- SVU_IRR
 - svuExtAfStats.c, [907](#)
 - svuExtStatsSatPixelsU32.c, [908](#)
- SVU_OCR
 - svuExtAfStats.c, [907](#)
 - svuExtStatsSatPixelsU32.c, [908](#)
- SVU_OSR
 - svuExtAfStats.c, [907](#)
 - svuExtStatsSatPixelsU32.c, [908](#)
- SVU_PTR
 - svuExtAfStats.c, [907](#)
 - svuExtStatsSatPixelsU32.c, [908](#)
- SVU_SYM
 - sippInternal.h, [766](#)
 - sippPlatform_ma2x5x.h, [810](#)
- SZ
 - sipp_ma2x5x.h, [599](#)
- sad11x11.h, [587](#)
- sad5x5.h, [587](#)
- satPixelsStats
 - CombDecimStatsGainsParam, [344](#)
- satThresh
 - AeAwbStatsCfg, [330](#)
 - ExtStatsSatPixelsU32Param, [382](#)
 - StatsAwbSatPixelsParam, [472](#)
 - StatsAwbSatPixelsParamU32, [474](#)
- satUInt8add
 - svuFast9ScoreCv.c, [912](#)
- satUInt8sub
 - svuFast9ScoreCv.c, [912](#)
- satFast9ScoreCv.c, [912](#)
- svuFast9M2.c, [910](#)
- svuFast9M2.c, [910](#)
- svuRandNoiseFp16.c, [953](#)
- svuGetShaveNumber
 - sippHpad.c, [650](#)
 - sippIoPtrs.c, [769](#)
 - sippPlatform_ma2x5x.h, [810](#)
 - sippShave.c, [819](#)
- svuScale
 - ChrGenSSParam, [340](#)
 - ContrastParam, [346](#)
- svuScale05BilinHV.h, [587](#)
- svuScale05BilinHV_Fp16U8
 - svuScale05BilinHV_Fp16U8.c, [955](#)
- svuScale05BilinHV_Fp16U8.h, [588](#)
- svuScale05BilinHV_U16ToU16
 - svuScale05BilinHV.c, [955](#)
- svuScale05BilinHV_U8ToU8
 - svuScale05BilinHV.c, [955](#)
- svuScale05BilinHVFp16
 - svuScale05BilinHVFp16.c, [956](#)
- svuScale05BilinHVFp16.h, [588](#)
- svuScale05Lanc6HV
 - svuScale05Lanc6HV.c, [957](#)
- svuScale05Lanc6HV.h, [589](#)
- svuScale05Lanc7HV.h, [589](#)
- svuScale2xBilinHV.h, [590](#)
- svuScale2xBilinHV025_Fp16ToFp16
 - svuScale2xBilinHV_025_075_Fp16ToFp16.c, [958](#)
- svuScale2xBilinHV025_Fp16U8
 - svuScale2xBilinHV_Fp16U8_phase025_075.c, [959](#)
- svuScale2xBilinHV025_U16ToU16
 - svuScale2xBilinHV_025_075_U16ToU16.c, [959](#)
- svuScale2xBilinHV025_U8ToU8
 - svuScale2xBilinHV_U8ToU8_phase025_075.c, [960](#)
- svuScale2xBilinHV075_Fp16ToFp16
 - svuScale2xBilinHV_025_075_Fp16ToFp16.c, [958](#)
- svuScale2xBilinHV075_Fp16U8
 - svuScale2xBilinHV_Fp16U8_phase025_075.c, [959](#)
- svuScale2xBilinHV075_U16ToU16
 - svuScale2xBilinHV_025_075_U16ToU16.c,

[959](#)
 scale2xBilinHV075_U8ToU8
 svuScale2xBilinHV_U8ToU8_phase025_-
 075.c, [960](#)
 scale2xBilinHV_025_075_Fp16ToFp16.h, [590](#)
 scale2xBilinHV_025_075_U16ToU16.h, [591](#)
 scale2xBilinHV_Fp16U8_phase025_075.h, [591](#)
 scale2xBilinHV_U8ToU8_phase025_075.h, [592](#)
 scale2xLancH.h, [592](#)
 scale2xLancHV.h, [592](#)
 scale2xLancV.h, [593](#)
 scaleBilinArb.h, [593](#)
 sch
 SippFilterS, [441](#)
 SippOseS, [449](#)
 schNo
 SippFilterS, [441](#)
 scharr_fp16, [312](#)
 SHAVE_SYM_EXPORT, [312](#)
 svuScharr_fp16, [312](#)
 scharr_fp16.h, [594](#)
 SchedAddCheck
 sippSchTypes.h, [817](#)
 schedConsumeMask
 SippPipelineS, [457](#)
 SchedFuncAsk
 sippTypes.h, [828](#)
 SchedIBufUsageInfo
 sippTypes.h, [828](#)
 SchedInfo
 sippTypes.h, [828](#)
 schedInfo
 SippPipelineS, [457](#)
 schedInfoCtx
 SippPipelineS, [458](#)
 schedInfoCtxSz
 SippPipelineS, [458](#)
 schedInfoEntries
 SippPipelineS, [458](#)
 SchedInfoS, [421](#)
 allMask, [421](#)
 dmaMask, [421](#)
 shaveMask, [422](#)
 sippHwWaitMask, [422](#)
 ScheduleCycles
 tRTStats, [479](#)
 seedlist
 svuRandNoiseFp16.c, [953](#)
 seeds
 svuRandNoise.c, [952](#)
 sel01
 MipiRxParam, [403](#)
 sel23
 MipiRxParam, [403](#)
 selMask
 MipiRxParam, [403](#)
 Semaphore, [422](#)
 ~Semaphore, [422](#), [423](#)
 posix_sem, [423](#)
 Post, [423](#)
 Semaphore, [422](#), [423](#)
 Wait, [423](#)
 setIterMask
 SippPipelineS, [458](#)
 shadowKernel
 ConvParam, [361](#)
 shadowSelect
 SippPipelineS, [458](#)
 Sharpen, [100](#)
 sharpenSigmaToCoefficients
 sipp_ma2x5x.h, [601](#)
 sippUtils.c, [836](#)
 shaveCount
 StartBicubicParam, [470](#)
 shaveMask
 SchedInfoS, [422](#)
 shaveNr
 ExtStatsSatPixelsU32Param, [382](#)
 PpAf, [416](#)
 ShaveWaitCycles
 tRTStats, [479](#)
 shift
 PixelUnpackerParam, [410](#)
 PixelUnpackerWBParam, [411](#)
 YDnsRefFp16Param, [490](#)
 YDnsRefParam, [491](#)
 sigma
 Bilateral5x5Param, [331](#)
 Sigma Denoise, [110](#)
 SigmaParam, [423](#)
 bayerPattern, [424](#)
 blcB, [424](#)
 blcGB, [424](#)
 blcGR, [424](#)
 blcR, [424](#)
 cfg, [424](#)
 frmDim, [424](#)
 thresh, [425](#)
 sipp.h, [594](#)
 sippAccessSchedulerTypes.h
 eSIPP_ACCESS_SCHEDULER_CMD_FIN-
 ALISE_PIPE, [610](#)

- eSIPP_ACCESS_SCHEDULER_CMD_HW-INIT, 610
- eSIPP_ACCESS_SCHEDULER_CMD_PROCESS_ITERS, 610
- eSIPP_ACCESS_SCHEDULER_CMD_RE-SCHEDULE_PIPE, 610
- eSIPP_ACCESS_SCHEDULER_CMD_RE-SET, 610
- eSIPP_ACCESS_SCHEDULER_CMD_TE-ARDOWN_PIPE, 610
- eSIPP_ACCESS_SCHEDULER_NULL, 610
- eSIPP_ACCESS_SCHEDULER_PIPE_UPD-ATE_STATUS, 610
- sippEvents.h
 - eSIPP_PIPELINE_FINALISED, 637
 - eSIPP_PIPELINE_FRAME_DONE, 637
 - eSIPP_PIPELINE_ITERS_DONE, 637
 - eSIPP_PIPELINE_RESCHEDULED, 637
 - eSIPP_PIPELINE_STARTED, 637
 - eSIPP_PIPELINE_SYNC_OP_DONE, 637
- sippHwBitfieldDefs.h
 - eBayerOrderBGGR, 723
 - eBayerOrderGBRG, 723
 - eBayerOrderGRBG, 723
 - eBayerOrderRGGG, 723
 - eRawFormatBayer, 724
 - eRawFormatPlanar, 724
- sippHwCommon_ma2x5x.h
 - CMX_ADDRESS, 734
 - CMX_MIRRORED_ADDRESS, 734
 - DDR_ADDRESS, 734
 - DDR_MIRRORED_ADDRESS, 734
 - OTHER, 734
 - REV1, 734
 - REV2, 734
- sippSchTypes.h
 - RS_CAN_RUN, 817
 - RS_CANNOT, 817
 - RS_DONT_KNOW, 817
- sippSessionControl.h
 - SIPP_PIPE_ACTIVE, 818
 - SIPP_PIPE_END_SESSION, 818
 - SIPP_PIPE_WAIT_FINALISE, 818
 - SIPP_PIPE_WAIT_INIT, 818
- sippTypes.h
 - DMA_MODE_PARTIAL_LINE, 830
 - DMA_MODE_STANDARD, 830
 - E_BLOCK_CALL_REJECTED, 829
 - E_CANNOT_FINISH_FILTER, 829
 - E_CDMA_QU_OVERFLOW, 829
 - E_DATA_ALIGN, 829
 - E_DATA_NOT_FOUND, 829
 - E_FINALISE_FAIL, 830
 - E_HEAP_CREATION_FAIL, 830
 - E_INVALID_HW_PARAM, 829
 - E_INVALID_MEM_P, 829
 - E_INVLD_FILT_FIRST_SLICE, 829
 - E_INVLD_FILT_LAST_SLICE, 829
 - E_INVLD_HW_ID, 829
 - E_INVLD_MIPI_RX_LOOPBACK, 829
 - E_INVLD_MULTI_INSTANCE, 829
 - E_INVLD_SLICE_WIDTH, 829
 - E_LAST, 830
 - E_MISSING_SHAVE_IMAGE, 829
 - E_OPT_EXEC_NUM, 829
 - E_OSE_CREATION_ERROR, 829
 - E_OUT_OF_MEM, 829
 - E_PAR_NOT_FOUND, 829
 - E_PC_CMX_MEM_ALLOC_ERR, 829
 - E_PC_RUNTIME_FAILURE, 829
 - E_PRECOMP_SCHED, 829
 - E_RUN_DONT_KNOW, 829
 - E_RUNS_ITER_GROUPS, 829
 - E_SCHEDULING_OVF, 829
 - E_SUCCESS, 829
 - E_TOO_MANY_CONSUMERS, 829
 - E_TOO_MANY_DMAS, 829
 - E_TOO_MANY_FILTERS, 829
 - E_TOO_MANY_PARENTS, 829
 - E_UNIMPLEMENTED_FEAT, 829
 - eSIPP_STATUS_ALREADY_ATTACHED, 831
 - eSIPP_STATUS_ALREADY_DONE, 831
 - eSIPP_STATUS_ALREADY_INIT, 830
 - eSIPP_STATUS_ASLEEP, 831
 - eSIPP_STATUS_BAD_ATTACHMENT, 831
 - eSIPP_STATUS_BAD_COMMAND, 831
 - eSIPP_STATUS_BAD_HANDLE, 830
 - eSIPP_STATUS_BAD_LENGTH, 830
 - eSIPP_STATUS_BAD_PARAMETER, 830
 - eSIPP_STATUS_BAD_UNIT, 830
 - eSIPP_STATUS_BUSY, 830
 - eSIPP_STATUS_CANCELLED, 830
 - eSIPP_STATUS_CLOSED_HANDLE, 830
 - eSIPP_STATUS_CODE_LAST, 831
 - eSIPP_STATUS_COMPLETE, 831
 - eSIPP_STATUS_DESTROYED, 830
 - eSIPP_STATUS_DISCONNECTED, 830
 - eSIPP_STATUS_DUPLICATE_NODE, 831
 - eSIPP_STATUS_EMPTY, 830
 - eSIPP_STATUS_FAILURE, 831

eSIPP_STATUS_FULL, 831
 eSIPP_STATUS_HARDWARE_NOT_FOU-
 ND, 831
 eSIPP_STATUS_ILLEGAL_OPERATION,
 831
 eSIPP_STATUS_IN_USE, 830
 eSIPP_STATUS_INCOMPATIBLE_FORM-
 ATS, 831
 eSIPP_STATUS_INT_HANDLED, 831
 eSIPP_STATUS_INT_NOT_HANDLED,
 831
 eSIPP_STATUS_INTERNAL_ERROR, 830
 eSIPP_STATUS_INVALID, 830
 eSIPP_STATUS_INVALID_DEVICE, 831
 eSIPP_STATUS_INVALID_EDGE, 831
 eSIPP_STATUS_INVALID_NODE, 831
 eSIPP_STATUS_INVALID_NUMBER, 831
 eSIPP_STATUS_INVALID_STATE, 831
 eSIPP_STATUS_INVALID_TYPE, 831
 eSIPP_STATUS_NOT_ATTACHED, 830
 eSIPP_STATUS_NOT_AVAILABLE, 830
 eSIPP_STATUS_NOT_COMPATIBLE, 830
 eSIPP_STATUS_NOT_FOUND, 830
 eSIPP_STATUS_NOT_HOOKED, 831
 eSIPP_STATUS_NOT_IMPLEMENTED,
 830
 eSIPP_STATUS_NOT_INIT, 830
 eSIPP_STATUS_NOT_SET, 831
 eSIPP_STATUS_NOT_SUPPORTED, 830
 eSIPP_STATUS_OK, 830
 eSIPP_STATUS_REOPENED_HANDLE,
 830
 eSIPP_STATUS_RESOURCE_ERROR, 830
 eSIPP_STATUS_STOPPED, 831
 eSIPP_STATUS_SUSPENDED, 831
 eSIPP_STATUS_TERMINATED, 831
 eSIPP_STATUS_TIMEOUT, 830
 eSIPP_STATUS_UNDEFINED, 830
 eSIPP_STATUS_UNKNOWN, 830
 vPoolCMXDMADesc, 831
 vPoolFilterLineBuf, 831
 vPoolFilterLineBuf0, 831
 vPoolFilterLineBuf1, 831
 vPoolFilterLineBuf10, 832
 vPoolFilterLineBuf11, 832
 vPoolFilterLineBuf2, 831
 vPoolFilterLineBuf3, 831
 vPoolFilterLineBuf4, 832
 vPoolFilterLineBuf5, 832
 vPoolFilterLineBuf6, 832
 vPoolFilterLineBuf7, 832
 vPoolFilterLineBuf8, 832
 vPoolFilterLineBuf9, 832
 vPoolGeneral, 831
 vPoolLast, 832
 vPoolPipeStructs, 831
 vPoolSchedule, 831
 vPoolScheduleTemp, 831
 sippTypesPrivate.h
 eSIPP_CMD_DELETE_PIPE_SW, 834
 eSIPP_CMD_FINALISE_PIPE_SW, 834
 eSIPP_CMD_INIT_HW, 834
 eSIPP_CMD_PROCESS_ITERS_HW, 835
 eSIPP_CMD_RESCHEDULE_PIPE_SW,
 834
 eSIPP_HW_ACTIVE, 834
 eSIPP_HW_INACTIVE, 834
 eSIPP_PIPE_ACTIVE, 835
 eSIPP_PIPE_END_SESSION, 835
 eSIPP_PIPE_WAIT_FINALISE, 835
 eSIPP_PIPE_WAIT_INIT, 835
 eSIPP_SET_OBUF_SPACE_EMPTY, 835
 eSIPP_SET_OBUF_SPACE_FULL, 835
 eSIPP_SET_OBUF_SPACE_ITER, 835
 sipp_ma2x5x.h, 594
 BPP, 597
 getIn3PlanePtr, 599
 getInPtr, 599
 getOutPtr, 600
 getPlaneIoPtrs, 600
 lumaGenLut, 601
 N_PL, 597
 PLF_IS_FINALIZED, 597
 SIPP_AUTO, 598
 SIPP_CROP, 598
 SIPP_REQ_SW_VIEW, 598
 SIPP_RESIZE, 598
 SIPP_SHIFT_PLANES, 598
 SZ, 599
 sharpenSigmaToCoefficients, 601
 sippAllocCmxMemRegion, 601
 sippChooseMemPool, 601
 sippCreateFilter, 601
 sippCreatePipeline, 601
 sippDbgCompareDeltaU8, 601
 sippDbgCompareU16, 601
 sippDbgCompareU32, 601
 sippDbgCompareU8, 601
 sippDbgLevel, 601
 sippDeletePipeline, 601
 sippDynRouteIrq, 601
 sippError, 601

- sippErrorSetFatal, 602
- sippFilterAddOBuf, 602
- sippFilterGetLinesPerIter, 602
- sippFilterGetLinesThisIter, 602
- sippFilterGetNumOutPlanes, 603
- sippFilterGetOutputBpp, 603
- sippFilterGetParentInputLines, 603
- sippFilterGetParentOutputHeight, 603
- sippFilterGetParentOutputWidth, 604
- sippFilterGetParentPlaneStride, 604
- sippFilterGetParentSliceWidth, 604
- sippFilterGetPlaneStride, 604
- sippFilterSetBufBitsPP, 604
- sippFinalizePipeline, 604
- sippFrameworkRun, 604
- sippGetErrorHistory, 604
- sippGetLastError, 604
- sippHwChromaDnsCfgReg, 604
- sippHwConvolutionCfgReg, 604
- sippHwHarrisCornerCfgReg, 604
- sippHwLutCfgReg, 604
- sippHwMedianCfgReg, 604
- sippHwUnsharpCfgReg, 604
- sippInitialize, 604
- sippLinkFilter, 605
- sippLinkFilterSetOBuf, 605
- sippMemCheck, 605
- sippMemStatus, 605
- sippPipeGetErrorStatus, 605
- sippPipeSetLinesPerIter, 605
- sippPipeSetNumLinesPerBuf, 605
- sippProcessFrame, 605
- sippProcessFrameNB, 605
- sippProcessIters, 605
- sippProcessItersNB, 605
- sippRdFileU8, 605
- sippRdFileU8toF16, 605
- sippRegisterEventCallback, 605
- sippReschedulePipeline, 605
- sippSetInitMask, 605
- sippSetSliceSize, 605
- sippStopSvus, 605
- sippTerm, 605
- sippTestCrcCheck, 606
- sippUtilComputeFp16Lut, 606
- sippUtilComputeFp16LutChannelMode, 606
- sippUtilPrintFp16Lut, 606
- sippWrFileF16toU8, 606
- sippWrFileU8, 606
- sipp_pl
 - sippHpad.c, 650
 - sippIoPtrs.c, 769
 - sippShave.c, 819
 - sippShvDbg.c, 821
- sippAccessScheduleCheckPending
 - sippAccessScheduler.c, 607
 - sippAccessScheduler.h, 608
- sippAccessScheduler.c, 606
 - gSippFramework, 607
 - pgSippHW, 607
 - sippAccessScheduleCheckPending, 607
 - sippAccessSchedulerControl, 607
 - sippAccessSchedulerInit, 607
 - sippAccessSchedulerQuPush, 607
 - UPDATE_PEND_LISTS, 607
- sippAccessScheduler.h, 608
 - sippAccessScheduleCheckPending, 608
 - sippAccessSchedulerControl, 608
 - sippAccessSchedulerInit, 608
 - sippAccessSchedulerQuPush, 608
- sippAccessSchedulerControl
 - sippAccessScheduler.c, 607
 - sippAccessScheduler.h, 608
- sippAccessSchedulerInit
 - sippAccessScheduler.c, 607
 - sippAccessScheduler.h, 608
- sippAccessSchedulerQuPush
 - sippAccessScheduler.c, 607
 - sippAccessScheduler.h, 608
- sippAccessSchedulerTypes.h, 608
- sippAddFilterToPipe
 - sippApiInternal.c, 614
 - sippInternal.h, 757
- sippAlloc
 - sippHeap.c, 648
 - sippInternal.h, 757
- sippAllocCmxLineBuffers
 - sippInternal.h, 757
 - sippMemLineBuffer.c, 775
- sippAllocCmxLineBuffersOPipe
 - sippInternal.h, 757
 - sippMemLineBuffer.c, 775
- sippAllocCmxMemRegion
 - sipp_ma2x5x.h, 601
 - sippMemLineBuffer.c, 775
- sippAnalysePipe
 - sippPipelineValidate.c, 806
- sippAnalysePipe2x5x
 - sippAnalysePipema2x5x.c, 611
- sippAnalysePipema2x5x.c, 610
 - sippAnalysePipe2x5x, 611
- sippApi.c, 611

- gSippFramework, 613
- gSippInitMask, 613
- gSippSliceSz, 613
- pgSippHW, 613
- sippCreateFilter, 612
- sippCreatePipeline, 612
- sippDeletePipeline, 613
- sippDynRouteIrq, 613
- sippFilterAddOBuf, 613
- sippFilterSetBufBitsPP, 613
- sippFinalizePipeline, 613
- sippFrameworkRun, 613
- sippInitialize, 613
- sippLinkFilter, 613
- sippLinkFilterSetOBuf, 613
- sippPipeSetLinesPerIter, 613
- sippPipeSetNumLinesPerBuf, 613
- sippProcessFrame, 613
- sippProcessFrameNB, 613
- sippRegisterEventCallback, 613
- sippReschedulePipeline, 613
- sippTerm, 613
- sippApiInternal.c, 614
 - gSippFramework, 615
 - sippAddFilterToPipe, 614
 - sippElaboratePipeline, 614
 - sippFreePipeResource, 614
 - sippHWInit, 614
 - sippInitPipeline, 614
 - sippRescheduleRequest, 615
 - sippResetFilterVariables, 615
 - sippRunItersRequest, 615
 - sippSWInit, 615
 - sippTermInternal, 615
- sippAsmOptSetup
 - sippCoreUtils.c, 626
 - sippInternal.h, 757
- sippAssert
 - sippError.c, 636
 - sippInternal.h, 757
- sippAssignCmxMemRegion
 - sippMemLineBuffer.c, 775
- sippBaseTypes.h, 615
 - bool, 616
 - false, 616
 - fp32, 616
 - Int16, 616
 - Int32, 616
 - Int8, 616
 - s16, 616
 - s32, 616
 - s64, 616
 - s8, 616
 - true, 616
 - u16, 616
 - u32, 616
 - u64, 616
 - u8, 616
 - UInt16, 616
 - UInt32, 616
 - UInt64, 616
 - UInt8, 617
- sippBufGetOBufCtx
 - sippHwCommon.c, 725
 - sippInternal.h, 757
- sippBufSetupIrqRate
 - sippHwCommon.c, 725
 - sippInternal.h, 757
- sippBufSetupIrqRateCQ
 - sippHwCommon.c, 725
 - sippInternal.h, 757
- sippBuildLnBufs
 - sippCoreUtils.c, 627
 - sippInternal.h, 757
- sippCQInit
 - sippCoreUtils.c, 627
 - sippInternal.h, 759
- SippCallback
 - sippTypes.h, 828
- sippCallbackInit
 - sippInternal.h, 757
 - sippManagerApi.c, 771
- sippCfg.h, 617
 - BUFF_HUGE_SZ, 617
 - DYNAMIC_IRQ_SIPP_0, 617
 - DYNAMIC_IRQ_SIPP_1, 618
 - DYNAMIC_IRQ_SIPP_2, 618
 - SIPP_CDMA_INT_NO, 618
 - SIPP_CMX_POOL_SZ, 618
 - SIPP_DDR_POOL_SZ, 618
 - SIPP_NUM_SVUS, 619
 - SIPP_THREAD_NAME, 619
- sippChainDmaDesc
 - sippCmxDmaIf.c, 621
 - sippInternal.h, 758
- sippCheckIterComplete
 - sippInternal.h, 758
 - sippScheduleIsr.c, 816
- sippCheckOPipeConnectionChrDns
 - sippOPipeSchedulingEntity.c, 780
- sippCheckOPipeConnectionColourComb
 - sippOPipeSchedulingEntity.c, 780

- sippCheckOPipeConnectionDbyr
 - sippOPipeSchedulingEntity.c, [780](#)
- sippCheckOPipeConnectionDefault
 - sippOPipeSchedulingEntity.c, [780](#)
- sippCheckOPipeConnectionDoGLTM
 - sippOPipeSchedulingEntity.c, [780](#)
- sippCheckOPipeConnectionGenChroma
 - sippOPipeSchedulingEntity.c, [780](#)
- sippCheckOPipeConnectionLsc
 - sippOPipeSchedulingEntity.c, [780](#)
- sippCheckOPipeConnectionLuma
 - sippOPipeSchedulingEntity.c, [780](#)
- sippCheckOPipeConnectionLut
 - sippOPipeSchedulingEntity.c, [780](#)
- sippCheckOPipeConnectionMedian
 - sippOPipeSchedulingEntity.c, [780](#)
- sippCheckOPipeConnectionPoly
 - sippOPipeSchedulingEntity.c, [780](#)
- sippCheckOPipeConnectionRaw
 - sippOPipeSchedulingEntity.c, [780](#)
- sippCheckOPipeConnectionSharpen
 - sippOPipeSchedulingEntity.c, [780](#)
- sippCheckOPipeConnectionSigma
 - sippOPipeSchedulingEntity.c, [780](#)
- sippChooseMemPool
 - sipp_ma2x5x.h, [601](#)
 - sippMem.c, [772](#)
- sippCmxBase
 - sippDbg.c, [630](#)
 - sippMem.c, [774](#)
 - sippMemLineBuffer.c, [776](#)
- SippCmxBufferMap
 - sippTypes.h, [828](#)
- SippCmxBufferMapS, [430](#)
 - cmxSliceUsageBitMask, [431](#)
 - numCmxSlicesAvail, [431](#)
 - pCmxSliceRegionList, [431](#)
 - totalMem, [431](#)
- sippCmxDmaDoneIrqHandler
 - sippCmxDmaIf.c, [622](#)
 - sippInternal.h, [758](#)
- sippCmxDmaIf.c, [620](#)
 - dmaIdle, [623](#)
 - dmaKickSequence, [621](#)
 - dmaKickSequenceCQ, [621](#)
 - dmaKickSequenceConcurrent, [621](#)
 - dmaTaskList, [623](#)
 - gSippInitMask, [623](#)
 - gSippSliceSz, [623](#)
 - sippChainDmaDesc, [621](#)
 - sippCmxDmaDoneIrqHandler, [622](#)
 - sippCmxDmaInit, [622](#)
 - sippCmxDmaInitAsync, [622](#)
 - sippDmaCQInit, [622](#)
 - sippInitDma, [622](#)
 - sippKickDma, [622](#)
 - sippKickDmaCQ, [622](#)
 - sippRunDmaCQ, [622](#)
 - sippRunDmaCQDrain, [622](#)
 - sippUpdateDmaAddr, [622](#)
 - sippUpdateDmaAddrCQ, [622](#)
 - sippWaitDma, [622](#)
 - topLevelCmxDmaIrqHandler, [623](#)
- sippCmxDmaInit
 - sippCmxDmaIf.c, [622](#)
 - sippInternal.h, [758](#)
- sippCmxDmaInitAsync
 - sippCmxDmaIf.c, [622](#)
 - sippInternal.h, [758](#)
- SippCommandData
 - sippTypesPrivate.h, [834](#)
- sippComputeBufferProps
 - sippCoreUtils.c, [627](#)
 - sippInternal.h, [758](#)
- sippComputeChunkWidths
 - sippCoreUtils.c, [627](#)
- sippComputeChunkWidthsSW
 - sippCoreUtils.c, [627](#)
- sippComputeHwCtxChg
 - sippInternal.h, [758](#)
- sippComputePaddingOffsets
 - sippCoreUtils.c, [627](#)
 - sippInternal.h, [758](#)
- sippComputeSliceLayout
 - sippCoreUtils.c, [627](#)
 - sippInternal.h, [758](#)
- sippComputeSwOutCt
 - sippCoreUtils.c, [627](#)
 - sippInternal.h, [758](#)
- sippConfigSvus
 - sippInternal.h, [758](#)
- sippConfirmChunkChain
 - sippMemLineBuffer.c, [775](#)
- sippCoreApi.c, [623](#)
 - gFakeFptr, [624](#)
 - gSippErrCode, [624](#)
 - gSippInitMask, [625](#)
 - gSippSliceSz, [625](#)
 - SHAVE_MAIN, [624](#)
 - sippCoreFinalisePipeline, [624](#)
 - sippCoreReInitLineBuffers, [624](#)
 - sippCoreReschedulePipeline, [624](#)

- sippCoreResourceInit, [624](#)
- sippCoreSetPaddingReqs, [624](#)
- sippProcessSchedData, [624](#)
- sippSetInitMask, [624](#)
- sippSetSliceSize, [624](#)
- sippCoreFinalisePipeline
 - sippCoreApi.c, [624](#)
 - sippInternal.h, [758](#)
- sippCoreGlobals.c, [625](#)
- sippCoreHw.c, [625](#)
 - sippCoreHwInitialLoad, [625](#)
 - sippCoreHwInitialSave, [625](#)
 - sippCoreUnitLoad, [625](#)
- sippCoreHwInitialLoad
 - sippCoreHw.c, [625](#)
 - sippInternal.h, [758](#)
- sippCoreHwInitialSave
 - sippCoreHw.c, [625](#)
 - sippInternal.h, [758](#)
- sippCoreReInitLineBuffers
 - sippCoreApi.c, [624](#)
- sippCoreReschedulePipeline
 - sippCoreApi.c, [624](#)
 - sippInternal.h, [758](#)
- sippCoreResourceInit
 - sippCoreApi.c, [624](#)
 - sippInternal.h, [758](#)
- sippCoreSetPaddingReqs
 - sippCoreApi.c, [624](#)
- sippCoreUnitLoad
 - sippCoreHw.c, [625](#)
- sippCoreUtils.c, [626](#)
 - adjustNodesRecursively, [626](#)
 - gSippSliceSz, [628](#)
 - sippAsmOptSetup, [626](#)
 - sippBuildLnBufs, [627](#)
 - sippCQInit, [627](#)
 - sippComputeBufferProps, [627](#)
 - sippComputeChunkWidths, [627](#)
 - sippComputeChunkWidthsSW, [627](#)
 - sippComputePaddingOffsets, [627](#)
 - sippComputeSliceLayout, [627](#)
 - sippComputeSwOutCt, [627](#)
 - sippFiltersResetSliceWidths, [627](#)
 - sippGetCoord2, [627](#)
 - sippGetCtxOrder, [627](#)
 - sippGetFirstHwFiltIdx, [628](#)
 - sippIncrementOutBufs, [628](#)
 - sippIniHwFilters, [628](#)
 - sippInitBufferLnPointers, [628](#)
 - sippInitSyncMutexes, [628](#)
 - sippUsingPrecompSched, [628](#)
- sippCreateFilter
 - sipp_ma2x5x.h, [601](#)
 - sippApi.c, [612](#)
- sippCreatePipeline
 - sipp_ma2x5x.h, [601](#)
 - sippApi.c, [612](#)
- sippDataSectAction
 - sippInternal.h, [759](#)
- sippDbg.c, [628](#)
 - sippCmxBase, [630](#)
 - sippDbgCompareDeltaU8, [629](#)
 - sippDbgCompareU16, [629](#)
 - sippDbgCompareU32, [629](#)
 - sippDbgCompareU8, [629](#)
 - sippDbgCreateDumpFiles, [629](#)
 - sippDbgDumpAsmOffsets, [629](#)
 - sippDbgDumpFilterOuts, [629](#)
 - sippDbgDumpGraph, [629](#)
 - sippDbgDumpRunMask, [630](#)
 - sippDbgDumpSchedForVcsCArr, [630](#)
 - sippDbgLevel, [630](#)
 - sippDbgShowBuffPtr, [630](#)
 - sippDumpHtmlMap, [630](#)
 - sippPrintSliceWidth, [630](#)
 - sippRdFileU8, [630](#)
 - sippRdFileU8toF16, [630](#)
 - sippTestCrcCheck, [630](#)
 - sippWrFileF16toU8, [630](#)
 - sippWrFileU8, [630](#)
- sippDbgCompareDeltaU8
 - sipp_ma2x5x.h, [601](#)
 - sippDbg.c, [629](#)
- sippDbgCompareU16
 - sipp_ma2x5x.h, [601](#)
 - sippDbg.c, [629](#)
- sippDbgCompareU32
 - sipp_ma2x5x.h, [601](#)
 - sippDbg.c, [629](#)
- sippDbgCompareU8
 - sipp_ma2x5x.h, [601](#)
 - sippDbg.c, [629](#)
- sippDbgCreateDumpFiles
 - sippDbg.c, [629](#)
 - sippInternal.h, [759](#)
- sippDbgDumpAsmOffsets
 - sippDbg.c, [629](#)
 - sippInternal.h, [759](#)
- sippDbgDumpFilterOuts
 - sippDbg.c, [629](#)
 - sippInternal.h, [759](#)

- sippDbgDumpGraph
 - sippDbg.c, [629](#)
 - sippInternal.h, [759](#)
- sippDbgDumpRunMask
 - sippDbg.c, [630](#)
 - sippInternal.h, [759](#)
- sippDbgDumpSchedForVcs
 - sippInternal.h, [759](#)
- sippDbgDumpSchedForVcsCArr
 - sippDbg.c, [630](#)
 - sippInternal.h, [759](#)
- sippDbgFrameCheck
 - sippInternal.h, [759](#)
- sippDbgLevel
 - sipp_ma2x5x.h, [601](#)
 - sippDbg.c, [630](#)
- sippDbgPrintNumPar
 - sippInternal.h, [759](#)
- sippDbgShowBuffPtr
 - sippDbg.c, [630](#)
 - sippInternal.h, [759](#)
- sippDefines.h, [630](#)
 - CMD_EXIT, [631](#)
 - CMD_H_PAD, [631](#)
 - CMD_RUN, [631](#)
 - CONCAT3, [632](#)
 - I_BASE, [632](#)
 - I_CFG, [632](#)
 - I_CTX, [632](#)
 - I_FC, [632](#)
 - I_LS, [632](#)
 - I_PS, [632](#)
 - I_SHADOW_BASE, [632](#)
 - I_SHADOW_CFG, [632](#)
 - I_SHADOW_LS, [632](#)
 - I_SHADOW_PS, [632](#)
 - O_BASE, [632](#)
 - O_CFG, [632](#)
 - O_CTX, [632](#)
 - O_FC, [632](#)
 - O_LS, [632](#)
 - O_SHADOW_BASE, [632](#)
 - PFL_DMA_DONE, [632](#)
 - PFL_SIPP_DONE, [632](#)
 - PFL_SIPP_EOF_DONE, [633](#)
 - PFL_SVU_DONE, [633](#)
 - PROG_IO_BUFF, [633](#)
- sippDeletePipeline
 - sipp_ma2x5x.h, [601](#)
 - sippApi.c, [613](#)
- sippDmaCQInit
 - sippCmxDmaIf.c, [622](#)
 - sippInternal.h, [759](#)
- sippDriverCmxDmaIf.c, [634](#)
- sippDumpHtmlMap
 - sippDbg.c, [630](#)
 - sippInternal.h, [759](#)
- sippDynRouteIrq
 - sipp_ma2x5x.h, [601](#)
 - sippApi.c, [613](#)
- sippElaboratePipeline
 - sippApiInternal.c, [614](#)
 - sippInternal.h, [760](#)
- sippError
 - sipp_ma2x5x.h, [601](#)
 - sippError.c, [636](#)
- sippError.c, [635](#)
 - gSippErrCode, [636](#)
 - gSippErrRdIdx, [636](#)
 - gSippErrWrIdx, [636](#)
 - gSippFatalErrList, [637](#)
 - sippAssert, [636](#)
 - sippError, [636](#)
 - sippErrorInit, [636](#)
 - sippErrorSetFatal, [636](#)
 - sippGetErrorHistory, [636](#)
 - sippGetLastError, [636](#)
 - sippPipeGetErrorStatus, [636](#)
- sippErrorInit
 - sippError.c, [636](#)
 - sippInternal.h, [760](#)
- sippErrorSetFatal
 - sipp_ma2x5x.h, [602](#)
 - sippError.c, [636](#)
- sippEventCallback_t
 - sippTypes.h, [828](#)
- sippEventNotify
 - sippInternal.h, [760](#)
 - sippManagerApi.c, [771](#)
- sippEvents.h, [637](#)
- sippFastExeUpd.h, [638](#)
- SippFilter
 - sippTypes.h, [828](#)
- sippFilterAccesors.c, [638](#)
 - sippFilterGetLinesPerIter, [638](#)
 - sippFilterGetLinesThisIter, [638](#)
 - sippFilterGetNumOutPlanes, [639](#)
 - sippFilterGetOutputBpp, [640](#)
 - sippFilterGetParentInputLines, [640](#)
 - sippFilterGetParentOutputHeight, [640](#)
 - sippFilterGetParentOutputWidth, [640](#)
 - sippFilterGetParentPlaneStride, [640](#)

- sippFilterGetParentSliceWidth, 640
- sippFilterGetPlaneStride, 640
- sippFilterAddOBuf
 - sipp_ma2x5x.h, 602
 - sippApi.c, 613
- sippFilterGetLinesPerIter
 - sipp_ma2x5x.h, 602
 - sippFilterAccesors.c, 638
- sippFilterGetLinesThisIter
 - sipp_ma2x5x.h, 602
 - sippFilterAccesors.c, 638
- sippFilterGetNumOutPlanes
 - sipp_ma2x5x.h, 603
 - sippFilterAccesors.c, 639
- sippFilterGetOutputBpp
 - sipp_ma2x5x.h, 603
 - sippFilterAccesors.c, 640
- sippFilterGetParentInputLines
 - sipp_ma2x5x.h, 603
 - sippFilterAccesors.c, 640
- sippFilterGetParentOutputHeight
 - sipp_ma2x5x.h, 603
 - sippFilterAccesors.c, 640
- sippFilterGetParentOutputWidth
 - sipp_ma2x5x.h, 604
 - sippFilterAccesors.c, 640
- sippFilterGetParentPlaneStride
 - sipp_ma2x5x.h, 604
 - sippFilterAccesors.c, 640
- sippFilterGetParentSliceWidth
 - sipp_ma2x5x.h, 604
 - sippFilterAccesors.c, 640
- sippFilterGetPlaneStride
 - sipp_ma2x5x.h, 604
 - sippFilterAccesors.c, 640
- SippFilterS, 431
 - bpp, 433
 - bytesPerPix, 433
 - cons, 433
 - consIbufIdx, 433
 - dbLineOut, 433
 - dbLinesIn, 433
 - errorStatus, 433
 - exeNo, 433
 - firstIterLines, 434
 - firstOutSlc, 434
 - firstRunNumLines, 434
 - firstRunRollLines, 434
 - flags, 434
 - funcSvuRun, 434
 - gi, 434
 - hPadding, 435
 - iBuf, 435
 - iBufCtx, 435
 - iBufs, 435
 - id, 435
 - KSIterList, 435
 - linePtrs, 435
 - linePtrs1stBase, 435
 - linePtrs2ndBase, 435
 - linePtrs3rdBase, 436
 - linePtrs4thBase, 436
 - linePtrs5thBase, 436
 - lineStride, 436
 - linesPerIter, 436
 - linesPerIterShift, 436
 - linesThisIter, 436
 - lnToPad, 436
 - nCons, 436
 - nCtxLoads, 437
 - nLines, 437
 - nLinesUsed, 437
 - nParents, 437
 - nPlanes, 437
 - numIBufs, 437
 - numOBufs, 437
 - oBuf, 438
 - oBufAlloc, 438
 - oBufs, 438
 - outLineDeficit, 438
 - outLineOffset, 438
 - outLinePtr, 438
 - outOff, 438
 - outputBuffer, 438
 - outputH, 439
 - outputW, 439
 - pOpipeSch, 440
 - pPipe, 440
 - parInfo, 440
 - params, 439
 - parentOBufIdx, 440
 - parents, 440
 - parentsKS, 440
 - planeStride, 440
 - ptrChunkPos, 441
 - sch, 441
 - schNo, 441
 - sliceWidth, 441
 - sliceWidthLastSvu, 442
 - unit, 442
- SippFilterSchedInfo
 - sippTypes.h, 828

- SippFilterSchedInfoS, 442
 - hasBuffers, 443
 - latenciesKnown, 443
 - nlinesUsedParent, 443
 - oBufLatency, 443
 - parentLatenciesKnown, 443
 - subSampleScale, 443
- sippFilterSetBufBitsPP
 - sipp_ma2x5x.h, 604
 - sippApi.c, 613
- sippFiltersResetSliceWidths
 - sippCoreUtils.c, 627
- sippFinalizePipeline
 - sipp_ma2x5x.h, 604
 - sippApi.c, 613
- sippFindConnectionsListRecursive
 - sippOPipeSchedulingEntity.c, 781
- sippFindInList
 - sippInternal.h, 760
 - sippUtils.c, 836
- sippFrameworkRun
 - sipp_ma2x5x.h, 604
 - sippApi.c, 613
- sippFree
 - sippHeap.c, 648
 - sippInternal.h, 760
- sippFreeList
 - sippHeap.c, 648
 - sippInternal.h, 760
- sippFreePipeResource
 - sippApiInternal.c, 614
 - sippInternal.h, 760
- sippGenericAllocRuntimeSched
 - sippGenericSchApi.c, 644
- sippGenericBlockHWUnits2x5x
 - sippGenericRuntimeMa2x5x.c, 643
- sippGenericCheckSERunParents
 - sippGenericSchApi.c, 644
- sippGenericColorCombChromaCheck
 - sippGenericSchReq.c, 646
- sippGenericDMACheck
 - sippGenericSchReq.c, 647
- sippGenericDbgDumpBuffState
 - sippInternal.h, 760
- sippGenericDbgPrintRunnable
 - sippGenericSchDebug.c, 645
 - sippInternal.h, 760
- sippGenericDbgShowBufferReq
 - sippGenericSchDebug.c, 645
 - sippInternal.h, 760
- sippGenericGetBufferEntry
 - sippGenericSchApi.c, 644
- sippGenericLinePrepare
 - sippGenericRuntime.c, 641
 - sippInternal.h, 760
- sippGenericRecordParentKS
 - sippGenericSchApi.c, 644
- sippGenericRunIterDone
 - sippGenericRuntime.c, 641
 - sippInternal.h, 760
- sippGenericRunNextIter
 - sippGenericRuntime.c, 641
 - sippInternal.h, 760
- sippGenericRuntime
 - sippGenericRuntime.c, 641
 - sippInternal.h, 760
- sippGenericRuntime.c, 641
 - gSippFramework, 642
 - sippGenericLinePrepare, 641
 - sippGenericRunIterDone, 641
 - sippGenericRunNextIter, 641
 - sippGenericRuntime, 641
 - sippGenericRuntimeClaimHWResource, 642
 - sippGenericRuntimeFrameReset, 642
 - sippGenericRuntimeLoadPipeline, 642
 - sippGenericRuntimeProcessIters, 642
 - sippGenericStartHWUnits2x5x, 642
 - sippGenericStartUnits, 642
 - sippGenericUpdateExecNums, 642
 - sippGenericWaitUnits, 642
 - sippGlobalOBFLIncStatus, 642
- sippGenericRuntimeClaimHWResource
 - sippGenericRuntime.c, 642
 - sippInternal.h, 761
- sippGenericRuntimeFrameReset
 - sippGenericRuntime.c, 642
 - sippInternal.h, 761
- sippGenericRuntimeHWProcessIters
 - sippGenericRuntimeMa2x5x.c, 643
 - sippInternal.h, 761
- sippGenericRuntimeLoadPipeline
 - sippGenericRuntime.c, 642
- sippGenericRuntimeProcessIters
 - sippGenericRuntime.c, 642
- sippGenericRuntimeMa2x5x.c, 642
 - sippGenericBlockHWUnits2x5x, 643
 - sippGenericRuntimeHWProcessIters, 643
 - sippGenericStartHWUnits2x5x, 643
 - sippGenericUpdateHWUnits2x5x, 643
 - sippGenericWaitUnits, 643
- sippGenericSchApi.c, 643
 - sippGenericAllocRuntimeSched, 644

- sippGenericCheckSERunParents, [644](#)
- sippGenericGetBufferEntry, [644](#)
- sippGenericRecordParentKS, [644](#)
- sippGenericSchCreateSEFromFilter, [644](#)
- sippGenericSchCreateSEFromOSE, [644](#)
- sippGenericSchedAllocTempStorage, [644](#)
- sippGenericSchedInitTempStorage, [644](#)
- sippGenericSchedPipeInit, [644](#)
- sippGenericSchedule, [644](#)
- sippGenericScheduleSetBufConsModels, [645](#)
- sippGenericSearchBuffer, [645](#)
- sippGenericSchCreateSEFromFilter
 - sippGenericSchApi.c, [644](#)
- sippGenericSchCreateSEFromOSE
 - sippGenericSchApi.c, [644](#)
- sippGenericSchDebug.c, [645](#)
 - sippGenericDbgPrintRunnable, [645](#)
 - sippGenericDbgShowBufferReq, [645](#)
- sippGenericSchReq.c, [645](#)
 - askChromaGenDownsizer, [646](#)
 - askCrop, [646](#)
 - askCropLatency, [646](#)
 - askHwColorCombChroma, [646](#)
 - askHwMedLumaLatency, [646](#)
 - askPolyFirResizer, [646](#)
 - askRegular, [646](#)
 - askRegularLatency, [646](#)
 - askResizer, [646](#)
 - askResizerLatency, [646](#)
 - sippGenericColorCombChromaCheck, [646](#)
 - sippGenericDMACheck, [647](#)
- sippGenericSchWrite.c, [647](#)
 - sippGenericSchedWr, [647](#)
- sippGenericSchedAllocTempStorage
 - sippGenericSchApi.c, [644](#)
- sippGenericSchedInitTempStorage
 - sippGenericSchApi.c, [644](#)
- sippGenericSchedPipeInit
 - sippGenericSchApi.c, [644](#)
- sippGenericSchedWr
 - sippGenericSchWrite.c, [647](#)
 - sippInternal.h, [761](#)
- sippGenericSchedule
 - sippGenericSchApi.c, [644](#)
 - sippInternal.h, [761](#)
- sippGenericScheduleSetBufConsModels
 - sippGenericSchApi.c, [645](#)
 - sippInternal.h, [761](#)
- sippGenericSearchBuffer
 - sippGenericSchApi.c, [645](#)
- sippGenericStartHWUnits2x5x
 - sippGenericRuntime.c, [642](#)
 - sippGenericRuntimema2x5x.c, [643](#)
- sippGenericStartUnits
 - sippGenericRuntime.c, [642](#)
 - sippInternal.h, [761](#)
- sippGenericUpdateExecNums
 - sippGenericRuntime.c, [642](#)
 - sippInternal.h, [761](#)
- sippGenericUpdateHWUnits2x5x
 - sippGenericRuntimema2x5x.c, [643](#)
 - sippInternal.h, [761](#)
- sippGenericWaitUnits
 - sippGenericRuntime.c, [642](#)
 - sippGenericRuntimema2x5x.c, [643](#)
- sippGetChunkStartPixelPos
 - sippShave.c, [819](#)
- sippGetCoord2
 - sippCoreUtils.c, [627](#)
 - sippInternal.h, [761](#)
- sippGetCtxOrder
 - sippCoreUtils.c, [627](#)
 - sippInternal.h, [761](#)
- sippGetErrorHistory
 - sipp_ma2x5x.h, [604](#)
 - sippError.c, [636](#)
- sippGetFirstHwFiltIdx
 - sippCoreUtils.c, [628](#)
 - sippInternal.h, [761](#)
- sippGetIBufCtxSigma
 - sippHwSigma.c, [752](#)
- sippGetIBufIdsSigma
 - sippHwSigma.c, [752](#)
- sippGetLastError
 - sipp_ma2x5x.h, [604](#)
 - sippError.c, [636](#)
- sippGetNextMemRegion
 - sippMemLineBuffer.c, [775](#)
- sippGetOBufIdsMipiRx0
 - sippHwMipiRx.c, [749](#)
- sippGetOBufIdsMipiRx1
 - sippHwMipiRx.c, [749](#)
- sippGetOBufIdsMipiRx2
 - sippHwMipiRx.c, [749](#)
- sippGetOBufIdsMipiRx3
 - sippHwMipiRx.c, [749](#)
- sippGetOBufIdsMipiTx0
 - sippHwMipiTx.c, [750](#)
- sippGetOBufIdsMipiTx1
 - sippHwMipiTx.c, [750](#)
- sippGetOBufIdsSigma
 - sippHwSigma.c, [752](#)

- sippGlobalOBFLIncStatus
 - sippGenericRuntime.c, [642](#)
 - sippHWSessionControl.c, [752](#)
 - sippIsr.c, [770](#)
 - sippScheduleIsr.c, [817](#)
- SippGlobals, [443](#)
 - uInfo, [443](#)
- sippHWInit
 - sippApiInternal.c, [614](#)
 - sippInternal.h, [761](#)
- sippHWSessionAddActiveLists
 - sippHWSessionControl.c, [751](#)
- sippHWSessionCommand
 - sippHWSessionControl.c, [751](#)
 - sippInternal.h, [762](#)
- sippHWSessionControl.c, [751](#)
 - pgSippHW, [752](#)
 - sippGlobalOBFLIncStatus, [752](#)
 - sippHWSessionAddActiveLists, [751](#)
 - sippHWSessionCommand, [751](#)
 - sippHWSessionInit, [751](#)
 - sippHWSessionRemoveActiveLists, [751](#)
 - sippHWSessionRemoveLoadedPipe, [752](#)
- sippHWSessionInit
 - sippHWSessionControl.c, [751](#)
 - sippInternal.h, [762](#)
- sippHWSessionRemoveActiveLists
 - sippHWSessionControl.c, [751](#)
 - sippInternal.h, [762](#)
- sippHWSessionRemoveLoadedPipe
 - sippHWSessionControl.c, [752](#)
 - sippInternal.h, [762](#)
- sippHandleCtxSwitch
 - sippInternal.h, [761](#)
- sippHeap.c, [647](#)
 - MCB_MAGIC, [648](#)
 - sippAlloc, [648](#)
 - sippFree, [648](#)
 - sippFreeList, [648](#)
 - sippHeapAlloc, [649](#)
 - sippHeapCheck, [649](#)
 - sippHeapCreate, [649](#)
 - sippHeapDefrag, [649](#)
 - sippHeapInit, [649](#)
 - sippHeaps, [649](#)
 - sippSizeList, [649](#)
- sippHeap_end
 - tsSippHeap, [482](#)
- sippHeap_size
 - tsSippHeap, [482](#)
- sippHeap_start
 - tsSippHeap, [482](#)
- sippHeapAlloc
 - sippHeap.c, [649](#)
- SippHeapCB, [443](#)
 - data, [444](#)
 - pNext, [444](#)
 - used_size, [444](#)
- sippHeapCheck
 - sippHeap.c, [649](#)
 - sippInternal.h, [761](#)
- sippHeapCreate
 - sippHeap.c, [649](#)
 - sippInternal.h, [761](#)
- sippHeapDefrag
 - sippHeap.c, [649](#)
- sippHeapInit
 - sippHeap.c, [649](#)
- sippHeaps
 - sippHeap.c, [649](#)
- sippHorizontalPadding
 - sippHpad.c, [650](#)
 - sippShave.c, [819](#)
- sippHpad.c, [649](#)
 - dbg_svu_no, [650](#)
 - memsetBpp, [650](#)
 - scGetShaveNumber, [650](#)
 - sipp_pl, [650](#)
 - sippHorizontalPadding, [650](#)
- sippHwBitfieldDefs.h, [650](#)
 - BAYER_ORDER_BGGR, [675](#)
 - BAYER_ORDER_GBRG, [675](#)
 - BAYER_ORDER_GRBG, [675](#)
 - BAYER_ORDER_RGGB, [675](#)
 - CC_CFG, [675](#)
 - CGEN_CFG_BYPASS, [676](#)
 - CGEN_CFG_SET, [677](#)
 - CHROMA_DNS_CFG, [679](#)
 - CHROMA_DNS_LIMIT, [682](#)
 - CONV_CFG, [682](#)
 - CONV_EVENODD_PIX, [684](#)
 - CONV_KERNEL_SIZE, [684](#)
 - CONV_KSZ_3x3, [684](#)
 - CONV_KSZ_5x5, [684](#)
 - DEBAYER_CFG, [684](#)
 - DEBAYER_DEWORM, [685](#)
 - DEBAYER_LUMA_EN, [686](#)
 - DEBAYER_OFFSET, [686](#)
 - DEBAYER_RGB_EN, [687](#)
 - DEBAYER_SLOPE, [687](#)
 - DEBAYER_THRESH, [687](#)
 - DOGL_CFG_MODE, [688](#)

- DOGL_CFG_SET, 688
- eBayerOrder, 723
- eRawInputFmt, 723
- LUMA_DNS_CFG, 693
- LUT_CFG, 694
- MED_CFG, 698
- MED_CFG_GO_FAST, 698
- MED_LUMA_ALPHA, 699
- MIPI_RX_CFG, 699
- MIPI_TX_FRM_CFG, 704
- MIPI_TX_INT_SEL, 705
- MIPI_TX_ONESHOT, 705
- POLY_CFG, 706
- POLY_CFG_DIM_IN, 706
- RAW_BAD_PIX_CFG, 710
- RAW_CFG, 711
- RAW_CFG_FORMAT, 712
- RAW_CFG_SDC_EN, 713
- RAW_GRGB_DECAY, 713
- RAW_GRGB_PLATO, 713
- RAW_PATCH_CFG, 714
- RAW_PATCH_START, 715
- RAW_STATS_PLANES, 715
- SHARPEN_CFG, 716
- SHARPEN_CFG_MODE, 717
- SHARPEN_COEF0_CFG, 717
- SHARPEN_COEF1_CFG, 717
- SHARPEN_COEF2_CFG, 717
- SHARPEN_COEF3_CFG, 717
- SIGMA_DNS_CFG, 719
- SIPP_UPFIRDN_ID, 723
- SippHwBuf
 - sippTypes.h, 828
- SippHwBufS, 444
 - base, 444
 - cfg, 444
 - ctx, 444
 - fillCtrl, 444
 - irqRate, 445
 - ls, 445
 - ps, 445
- sippHwChromaDns.c, 724
- sippHwChromaDnsCfgReg
 - sipp_ma2x5x.h, 604
- sippHwColComb.c, 724
- sippHwCommon.c, 724
 - sippBufGetObufCtx, 725
 - sippBufSetupIrqRate, 725
 - sippBufSetupIrqRateCQ, 725
 - sippIbufSetup, 725
 - sippObufSetup, 725
- sippHwCommon_ma2x5x.h, 726
 - ACTIVE, 729
 - AMC_WIDTH, 729
 - AddressType, 734
 - BAYER, 729
 - BGGR, 729
 - CMX_NSLICES, 729
 - CONCAT5, 729
 - CONCAT7, 729
 - DEF_SLICE_SIZE, 730
 - DEFAULT, 730
 - DISABLED, 730
 - ENABLED, 730
 - GBRG, 730
 - GRBG, 730
 - LSC_KERNEL_SIZE, 730
 - LUT_KERNEL_SIZE, 730
 - MAGN_ORIENT_16BIT, 730
 - MAX_PLANES, 730
 - MED_KERNEL_SIZE, 730
 - NORMAL_MODE, 731
 - NORMAL_THETA, 731
 - ORIENT_8BIT, 731
 - P_BGR, 731
 - P_BRG, 731
 - P_GBR, 731
 - P_GRB, 731
 - P_RBG, 731
 - P_RGB, 731
 - PLANAR, 731
 - PRE_FP16_GRAD, 731
 - PRE_U8_GRAD, 731
 - RAW_KERNEL_SIZE, 731
 - REV1_DEF, 731
 - REV2_DEF, 731
 - RGGG, 731
 - Revision, 734
 - SCALED_MAGN_16BIT, 731
 - SCALED_MAGN_8BIT, 731
 - SHADOW, 731
 - SIPP_CBL_OFFSET, 732
 - SIPP_CC_NLBRC, 732
 - SIPP_CONV_NLBRC, 732
 - SIPP_CS_MASK, 732
 - SIPP_CS_OFFSET, 732
 - SIPP_CTXUP_BIT, 732
 - SIPP_DBYR_NLBRC, 732
 - SIPP_DOG_NLBRC, 732
 - SIPP_DOGL_NLBRC, 732
 - SIPP_FO_MASK, 732
 - SIPP_FO_OFFSET, 732

[SIPP_GCHR_NLBRC, 732](#)
[SIPP_IC_MASK, 732](#)
[SIPP_IC_OFFSET, 732](#)
[SIPP_INCDEC_BIT, 732](#)
[SIPP_IR_MASK, 733](#)
[SIPP_IR_OFFSET, 733](#)
[SIPP_KL_MASK, 733](#)
[SIPP_LS_MASK, 733](#)
[SIPP_LSC_NLBRC, 733](#)
[SIPP_LUMA_NLBRC, 733](#)
[SIPP_LUT_NLBRC, 733](#)
[SIPP_MED_NLBRC, 733](#)
[SIPP_NL_MASK, 733](#)
[SIPP_NL_OFFSET, 733](#)
[SIPP_NLBWC, 733](#)
[SIPP_NP_MASK, 733](#)
[SIPP_NP_OFFSET, 733](#)
[SIPP_OF_MASK, 733](#)
[SIPP_OF_OFFSET, 733](#)
[SIPP_PS_MASK, 733](#)
[SIPP_RAW_NLBRC, 733](#)
[SIPP_SB_MASK, 733](#)
[SIPP_SC_MASK, 733](#)
[SIPP_SC_OFFSET, 733](#)
[SIPP_SL_OFFSET, 734](#)
[SIPP_SS_MASK, 734](#)
[SIPP_SS_OFFSET, 734](#)
[SIPP_START_BIT, 734](#)
[X_AXIS_REFL, 734](#)
[XY_AXIS_REFL, 734](#)
[sippHwConv.c, 735](#)
[sippHwConvolutionCfgReg](#)
 [sipp_ma2x5x.h, 604](#)
[sippHwDebayer.c, 735](#)
[sippHwDefs.h, 735](#)
[sippHwDefs_ma2x5x.h, 736](#)
 [cfgMipiRxLoopback, 737](#)
 [cfgMipiTxLoopback, 737](#)
 [packColCombCCM, 737](#)
 [packConv3x3CCM, 737](#)
 [packConv5x5CCM, 737](#)
 [packLumaDnsGaussLut, 737](#)
 [startMipiTxLoopback, 738](#)
[sippHwDogLtm.c, 738](#)
[sippHwEdge.c, 738](#)
[sippHwGenChroma.c, 738](#)
[sippHwHarris.c, 739](#)
[sippHwHarrisCornerCfgReg](#)
 [sipp_ma2x5x.h, 604](#)
[SippHwIOBuf, 445](#)
 [base, 445](#)
 [cfg, 445](#)
 [ctx, 445](#)
 [fillCtrl, 445](#)
 [irqRate, 445](#)
 [ls, 446](#)
 [ps, 446](#)
[sippHwIds.h, 739](#)
 [EXE_NUM, 741](#)
 [SIPP_AF_STATS_ID, 741](#)
 [SIPP_CC_3DLUT_ID, 741](#)
 [SIPP_CC_ID, 741](#)
 [SIPP_CC_ID_MASK, 741](#)
 [SIPP_CGEN_ID, 741](#)
 [SIPP_CHROMA_ID, 742](#)
 [SIPP_CONV_ID, 742](#)
 [SIPP_DBYR_ID, 742](#)
 [SIPP_DMA_ID, 742](#)
 [SIPP_DOGL_ID, 742](#)
 [SIPP_EDGE_OP_ID, 742](#)
 [SIPP_FAKE_ID, 742](#)
 [SIPP_HARRIS_ID, 743](#)
 [SIPP_LSC_GM_ID, 743](#)
 [SIPP_LSC_ID, 743](#)
 [SIPP_LSC_ID_MASK, 743](#)
 [SIPP_LUMA_ID, 743](#)
 [SIPP_LUMA_REF_ID, 744](#)
 [SIPP_LUT_ID, 744](#)
 [SIPP_LUT_ID_MASK, 744](#)
 [SIPP_LUT_LOAD_ID, 744](#)
 [SIPP_MAX_ID, 744](#)
 [SIPP_MED_ID, 744](#)
 [SIPP_MED_ID_MASK, 744](#)
 [SIPP_MED_LUMA_ID, 744](#)
 [SIPP_MIPI_RX0_ID, 744](#)
 [SIPP_MIPI_RX1_ID, 744](#)
 [SIPP_MIPI_RX2_ID, 745](#)
 [SIPP_MIPI_RX3_ID, 745](#)
 [SIPP_MIPI_TX0_ID, 745](#)
 [SIPP_MIPI_TX1_ID, 745](#)
 [SIPP_OPIPE_ID, 745](#)
 [SIPP_RAW_ID, 745](#)
 [SIPP_RAW_ID_MASK, 745](#)
 [SIPP_RESERVED_ID, 745](#)
 [SIPP_RGB_HIST_ID, 745](#)
 [SIPP_SHARPEN_ID, 745](#)
 [SIPP_SIGMA_ID, 746](#)
 [SIPP_STATS_ID, 746](#)
 [SIPP_STATS_MASK, 746](#)
 [SIPP_SVU_ID, 746](#)
 [SIPP_UPFIRDN0_ID, 746](#)
 [SIPP_UPFIRDN1_ID, 746](#)

[SIPP_UPFIRDN2_ID, 746](#)
[sippHwLsc.c, 746](#)
[sippHwLumaDns.c, 747](#)
[sippHwLut.c, 747](#)
[sippHwLutCfgReg](#)
 [sipp_ma2x5x.h, 604](#)
[sippHwMedian.c, 748](#)
[sippHwMedianCfgReg](#)
 [sipp_ma2x5x.h, 604](#)
[sippHwMipiRx.c, 748](#)
 [sippGetOBufIdsMipiRx0, 749](#)
 [sippGetOBufIdsMipiRx1, 749](#)
 [sippGetOBufIdsMipiRx2, 749](#)
 [sippGetOBufIdsMipiRx3, 749](#)
 [sippInitMipiRx, 749](#)
 [sippLoadMipiRx, 749](#)
 [sippSetBufLatenciesMipiRx, 749](#)
 [sippSetOBufLevelsMipiRx0, 749](#)
 [sippSetOBufLevelsMipiRx1, 749](#)
 [sippSetOBufLevelsMipiRx2, 749](#)
 [sippSetOBufLevelsMipiRx3, 749](#)
[sippHwMipiTx.c, 749](#)
 [sippGetOBufIdsMipiTx0, 750](#)
 [sippGetOBufIdsMipiTx1, 750](#)
 [sippInitMipiTx, 750](#)
 [sippLoadMipiTx, 750](#)
 [sippSetBufLatenciesMipiTx, 750](#)
 [sippSetOBufLevelsMipiTx0, 750](#)
 [sippSetOBufLevelsMipiTx1, 750](#)
[sippHwPolyFir.c, 750](#)
[sippHwRaw.c, 750](#)
[sippHwSigma.c, 752](#)
 [sippGetIBufCtxSigma, 752](#)
 [sippGetIBufIdsSigma, 752](#)
 [sippGetOBufIdsSigma, 752](#)
 [sippInitSigma, 752](#)
 [sippLoadSigma, 753](#)
 [sippSetBufLatenciesSigma, 753](#)
 [sippSetOBufLevelsSigma, 753](#)
[sippHwUnsharp.c, 753](#)
[sippHwUnsharpCfgReg](#)
 [sipp_ma2x5x.h, 604](#)
[sippHwWaitMask](#)
 [SchedInfoS, 422](#)
[sippIbflDecHandler](#)
 [sippInternal.h, 762](#)
 [sippScheduleIsr.c, 816](#)
[sippIbufSetup](#)
 [sippHwCommon.c, 725](#)
 [sippInternal.h, 762](#)
[sippIdentifyOPipeSchedulingEntity](#)
 [sippOPipeSchedulingEntity.c, 781](#)
[sippIncrementOutBufs](#)
 [sippCoreUtils.c, 628](#)
 [sippInternal.h, 762](#)
[sippIniHwFilters](#)
 [sippCoreUtils.c, 628](#)
 [sippInternal.h, 762](#)
[sippInitBufferLnPointers](#)
 [sippCoreUtils.c, 628](#)
 [sippInternal.h, 762](#)
[sippInitDma](#)
 [sippCmxDmaIf.c, 622](#)
 [sippInternal.h, 762](#)
[sippInitLnMemPool](#)
 [sippInternal.h, 762](#)
 [sippMem.c, 773](#)
[sippInitLnMemPoolSlices](#)
 [sippInternal.h, 762](#)
 [sippMem.c, 773](#)
[sippInitMipiRx](#)
 [sippHwMipiRx.c, 749](#)
[sippInitMipiTx](#)
 [sippHwMipiTx.c, 750](#)
[sippInitPhysicalPoolGlobal](#)
 [sippInternal.h, 762](#)
 [sippMem.c, 773](#)
[sippInitPhysicalPoolPipe](#)
 [sippInternal.h, 763](#)
 [sippMem.c, 773](#)
[sippInitPipeline](#)
 [sippApiInternal.c, 614](#)
 [sippInternal.h, 763](#)
[sippInitSchedPool](#)
 [sippInternal.h, 763](#)
 [sippMem.c, 773](#)
[sippInitSigma](#)
 [sippHwSigma.c, 752](#)
[sippInitSyncMutexes](#)
 [sippCoreUtils.c, 628](#)
 [sippInternal.h, 763](#)
[sippInitialize](#)
 [sipp_ma2x5x.h, 604](#)
 [sippApi.c, 613](#)
[sippIntBarrierSetup](#)
 [sippInternal.h, 763](#)
 [sippIsr.c, 770](#)
[sippInternal.h, 753](#)
 [ctxSwitchChromaDns, 757](#)
 [ctxSwitchColComb, 757](#)
 [ctxSwitchLut, 757](#)
 [ctxSwitchMipiRx, 757](#)

[ctxSwitchMipiTx, 757](#)
[ctxSwitchOnePar, 757](#)
[ctxSwitchPoly, 757](#)
[ctxSwitchTwoPar, 757](#)
[dmaKickSequenceConcurrent, 757](#)
[SVU_SYM, 766](#)
[sippAddFilterToPipe, 757](#)
[sippAlloc, 757](#)
[sippAllocCmxLineBuffers, 757](#)
[sippAllocCmxLineBuffersOPipe, 757](#)
[sippAsmOptSetup, 757](#)
[sippAssert, 757](#)
[sippBufGetObufCtx, 757](#)
[sippBufSetupIrqRate, 757](#)
[sippBufSetupIrqRateCQ, 757](#)
[sippBuildLnBufs, 757](#)
[sippCQInit, 759](#)
[sippCallbackInit, 757](#)
[sippChainDmaDesc, 758](#)
[sippCheckIterComplete, 758](#)
[sippCmxDmaDoneIrqHandler, 758](#)
[sippCmxDmaInit, 758](#)
[sippCmxDmaInitAsync, 758](#)
[sippComputeBufferProps, 758](#)
[sippComputeHwCtxChg, 758](#)
[sippComputePaddingOffsets, 758](#)
[sippComputeSliceLayout, 758](#)
[sippComputeSwOutCt, 758](#)
[sippConfigSvus, 758](#)
[sippCoreFinalisePipeline, 758](#)
[sippCoreHwInitialLoad, 758](#)
[sippCoreHwInitialSave, 758](#)
[sippCoreReschedulePipeline, 758](#)
[sippCoreResourceInit, 758](#)
[sippDataSectAction, 759](#)
[sippDbgCreateDumpFiles, 759](#)
[sippDbgDumpAsmOffsets, 759](#)
[sippDbgDumpFilterOuts, 759](#)
[sippDbgDumpGraph, 759](#)
[sippDbgDumpRunMask, 759](#)
[sippDbgDumpSchedForVcs, 759](#)
[sippDbgDumpSchedForVcsCArr, 759](#)
[sippDbgFrameCheck, 759](#)
[sippDbgPrintNumPar, 759](#)
[sippDbgShowBuffPtr, 759](#)
[sippDmaCQInit, 759](#)
[sippDumpHtmlMap, 759](#)
[sippElaboratePipeline, 760](#)
[sippErrorInit, 760](#)
[sippEventNotify, 760](#)
[sippFindInList, 760](#)
[sippFree, 760](#)
[sippFreeList, 760](#)
[sippFreePipeResource, 760](#)
[sippGenericDbgDumpBuffState, 760](#)
[sippGenericDbgPrintRunnable, 760](#)
[sippGenericDbgShowBufferReq, 760](#)
[sippGenericLinePrepare, 760](#)
[sippGenericRunIterDone, 760](#)
[sippGenericRunNextIter, 760](#)
[sippGenericRuntime, 760](#)
[sippGenericRuntimeClaimHWResource, 761](#)
[sippGenericRuntimeFrameReset, 761](#)
[sippGenericRuntimeHWProcessIters, 761](#)
[sippGenericSchedWr, 761](#)
[sippGenericSchedule, 761](#)
[sippGenericScheduleSetBufConsModels, 761](#)
[sippGenericStartUnits, 761](#)
[sippGenericUpdateExecNums, 761](#)
[sippGenericUpdateHWUnits2x5x, 761](#)
[sippGetCoord2, 761](#)
[sippGetCtxOrder, 761](#)
[sippGetFirstHwFiltIdx, 761](#)
[sippHWInit, 761](#)
[sippHWSessionCommand, 762](#)
[sippHWSessionInit, 762](#)
[sippHWSessionRemoveActiveLists, 762](#)
[sippHWSessionRemoveLoadedPipe, 762](#)
[sippHandleCtxSwitch, 761](#)
[sippHeapCheck, 761](#)
[sippHeapCreate, 761](#)
[sippIbflDecHandler, 762](#)
[sippIbufSetup, 762](#)
[sippIncrementOutBufs, 762](#)
[sippIniHwFilters, 762](#)
[sippInitBufferLnPointers, 762](#)
[sippInitDma, 762](#)
[sippInitLnMemPool, 762](#)
[sippInitLnMemPoolSlices, 762](#)
[sippInitPhysicalPoolGlobal, 762](#)
[sippInitPhysicalPoolPipe, 763](#)
[sippInitPipeline, 763](#)
[sippInitSchedPool, 763](#)
[sippInitSyncMutexes, 763](#)
[sippIntBarrierSetup, 763](#)
[sippIsrSetup, 763](#)
[sippIssueCommand, 763](#)
[sippKickDma, 763](#)
[sippKickDmaCQ, 763](#)
[sippKickShaveM1PC, 763](#)
[sippKickSvus, 763](#)
[sippListSort, 764](#)

- sippMemAlloc, [764](#)
- sippMemFindMaxLnMemPoolFree, [764](#)
- sippMemFree, [764](#)
- sippMemFreeList, [764](#)
- sippMemInitVirtPhysMaps, [764](#)
- sippMemLBConsolidateRegions, [764](#)
- sippMemLBMatchRegionsToChunks, [764](#)
- sippObflIncHandler, [764](#)
- sippObufSetup, [764](#)
- sippPipeSessionControl, [764](#)
- sippPipeSessionControlInit, [765](#)
- sippPrintSliceWidth, [765](#)
- sippRescheduleRequest, [765](#)
- sippResetFilterVariables, [765](#)
- sippRunDmaCQ, [765](#)
- sippRunItersRequest, [765](#)
- sippSWInit, [765](#)
- sippSetupSvus, [765](#)
- sippStopSvus, [765](#)
- sippSvuDoneIrqHandler, [765](#)
- sippTermInternal, [765](#)
- sippUsingPrecompSched, [765](#)
- sippUtilOrderPixels, [765](#)
- sippValidatePipe, [765](#)
- sippWaitDma, [766](#)
- sippWaitShave, [766](#)
- topLevelCmxDmaIrqHandler, [766](#)
- sippIoPtrs.c, [766](#)
 - dbg_svu_no, [769](#)
 - getIn3PlanePtr, [767](#)
 - getInPtr, [767](#)
 - getInPtrAbs, [768](#)
 - getOutPtr, [768](#)
 - getPlaneIoPtrs, [768](#)
 - scGetShaveNumber, [769](#)
 - sipp_pl, [769](#)
- sippIsr.c, [769](#)
 - gSippFramework, [770](#)
 - pgSippHW, [770](#)
 - sippGlobalOBFLIncStatus, [770](#)
 - sippIntBarrierSetup, [770](#)
 - sippIsrSetup, [770](#)
- sippIsrSetup
 - sippInternal.h, [763](#)
 - sippIsr.c, [770](#)
- sippIssueCommand
 - sippInternal.h, [763](#)
 - sippManagerApi.c, [771](#)
- sippKickDma
 - sippCmxDmaIf.c, [622](#)
 - sippInternal.h, [763](#)
- sippKickDmaCQ
 - sippCmxDmaIf.c, [622](#)
 - sippInternal.h, [763](#)
- sippKickShaveM1PC
 - sippInternal.h, [763](#)
 - sippShaveIf.c, [820](#)
- sippKickSvus
 - sippInternal.h, [763](#)
 - sippShaveIf.c, [820](#)
- sippLinkFilter
 - sipp_ma2x5x.h, [605](#)
 - sippApi.c, [613](#)
- sippLinkFilterSetOBuf
 - sipp_ma2x5x.h, [605](#)
 - sippApi.c, [613](#)
- sippListSort
 - sippInternal.h, [764](#)
 - sippUtils.c, [836](#)
- sippLoadMipiRx
 - sippHwMipiRx.c, [749](#)
- sippLoadMipiTx
 - sippHwMipiTx.c, [750](#)
- sippLoadSigma
 - sippHwSigma.c, [753](#)
- sippMLPIRuntime.c, [776](#)
- sippMLPIRuntimeema2x5x.c, [777](#)
- sippMLPISchApi.c, [777](#)
- sippMLPISchDebug.c, [777](#)
- sippMLPISchReq.c, [777](#)
- sippMLPISchWrite.c, [777](#)
- SippManagedBufSchedInfo, [446](#)
 - placeholder, [446](#)
- sippManagerApi.c, [770](#)
 - gSippFramework, [771](#)
 - sippCallbackInit, [771](#)
 - sippEventNotify, [771](#)
 - sippIssueCommand, [771](#)
- sippMapRegionMapAddrToSliceZero
 - sippMemLineBuffer.c, [775](#)
- sippMapRegionToCmx
 - sippMemLineBuffer.c, [775](#)
- sippMem.c, [771](#)
 - ALIGNED, [772](#)
 - gSippSliceSz, [774](#)
 - gsSippMemMap, [774](#)
 - SIPP_MEMPOOL_CMx, [772](#)
 - SIPP_MEMPOOL_DDR, [772](#)
 - sippChooseMemPool, [772](#)
 - sippCmxBase, [774](#)
 - sippInitLnMemPool, [773](#)
 - sippInitLnMemPoolSlices, [773](#)

- sippInitPhysicalPoolGlobal, 773
- sippInitPhysicalPoolPipe, 773
- sippInitSchedPool, 773
- sippMemAlloc, 773
- sippMemCheck, 773
- sippMemFindMaxLnMemPoolFree, 773
- sippMemFree, 773
- sippMemFreeList, 773
- sippMemInitVirtPhysMaps, 773
- sippMemStatus, 774
- sippPoolsUsage, 774
- sippSetCircularLnBuffBase, 774
- sippMemAlloc
 - sippInternal.h, 764
 - sippMem.c, 773
- sippMemAllocChainChunk
 - sippMemLineBuffer.c, 776
- sippMemCheck
 - sipp_ma2x5x.h, 605
 - sippMem.c, 773
- sippMemFindMaxLnMemPoolFree
 - sippInternal.h, 764
 - sippMem.c, 773
- sippMemFree
 - sippInternal.h, 764
 - sippMem.c, 773
- sippMemFreeList
 - sippInternal.h, 764
 - sippMem.c, 773
- sippMemInitVirtPhysMaps
 - sippInternal.h, 764
 - sippMem.c, 773
- sippMemLBConsolidateRegions
 - sippInternal.h, 764
 - sippMemLineBuffer.c, 776
- sippMemLBMatchRegionsToChunks
 - sippInternal.h, 764
 - sippMemLineBuffer.c, 776
- sippMemLBRemoveNode
 - sippMemLineBuffer.c, 776
- sippMemLineBuffer.c, 774
 - sippAllocCmxLineBuffers, 775
 - sippAllocCmxLineBuffersOPipe, 775
 - sippAllocCmxMemRegion, 775
 - sippAssignCmxMemRegion, 775
 - sippCmxBase, 776
 - sippConfirmChunkChain, 775
 - sippGetNextMemRegion, 775
 - sippMapRegionMapAddrToSliceZero, 775
 - sippMapRegionToCmx, 775
 - sippMemAllocChainChunk, 776
 - sippMemLBConsolidateRegions, 776
 - sippMemLBMatchRegionsToChunks, 776
 - sippMemLBRemoveNode, 776
 - sippMemRegionAllocLineBuffer, 776
- SippMemRegion, 446
 - regionOffset, 446
 - regionSize, 446
 - regionUsed, 446
- sippMemRegionAllocLineBuffer
 - sippMemLineBuffer.c, 776
- SippMemRegionListNode, 447
 - chainLinked, 447
 - chainStart, 447
 - pNext, 447
 - pNextChunkReg, 447
 - regionAddr, 447
 - regionSize, 447
 - regionUsed, 447
 - regionUsedPtr, 448
 - slice0Addr, 448
- sippMemStatus
 - sipp_ma2x5x.h, 605
 - sippMem.c, 774
- sippOPipeRuntime.c, 777
- sippOPipeSchApi.c, 778
- sippOPipeSchedulingEntity.c, 778
 - sippCheckOPipeConnectionChrDns, 780
 - sippCheckOPipeConnectionColourComb, 780
 - sippCheckOPipeConnectionDbyr, 780
 - sippCheckOPipeConnectionDefault, 780
 - sippCheckOPipeConnectionDoGLTM, 780
 - sippCheckOPipeConnectionGenChroma, 780
 - sippCheckOPipeConnectionLsc, 780
 - sippCheckOPipeConnectionLuma, 780
 - sippCheckOPipeConnectionLut, 780
 - sippCheckOPipeConnectionMedian, 780
 - sippCheckOPipeConnectionPoly, 780
 - sippCheckOPipeConnectionRaw, 780
 - sippCheckOPipeConnectionSharpen, 780
 - sippCheckOPipeConnectionSigma, 780
 - sippFindConnectionsListRecursive, 781
 - sippIdentifyOPipeSchedulingEntity, 781
 - sippOSEAddFilter, 781
 - sippOSEComplete, 781
 - sippOSECreate, 781
 - sippOSEProcessFilter, 781
 - sippOSEProcessFilterIBufs, 781
 - sippOSEProcessFilterOBufs, 781
 - sippRemoveEntriesList, 781
- sippOSEAddFilter

- sippOPipeSchedulingEntity.c, [781](#)
- sippOSEComplete
 - sippOPipeSchedulingEntity.c, [781](#)
- sippOSECreate
 - sippOPipeSchedulingEntity.c, [781](#)
- sippOSEProcessFilter
 - sippOPipeSchedulingEntity.c, [781](#)
- sippOSEProcessFilterIBufs
 - sippOPipeSchedulingEntity.c, [781](#)
- sippOSEProcessFilterOBufs
 - sippOPipeSchedulingEntity.c, [781](#)
- sippObflIncHandler
 - sippInternal.h, [764](#)
 - sippScheduleIsr.c, [816](#)
- sippObufSetup
 - sippHwCommon.c, [725](#)
 - sippInternal.h, [764](#)
- sippOpipBufInfo, [448](#)
 - oBufIdx, [448](#)
 - ptrFilt, [448](#)
- SippOseS, [448](#)
 - consIbufIdx, [449](#)
 - consSE, [449](#)
 - filts, [449](#)
 - flags, [449](#)
 - iBufCtx, [449](#)
 - iBufs, [449](#)
 - linesPerIter, [449](#)
 - numConsumers, [449](#)
 - numFilts, [449](#)
 - numIBufs, [449](#)
 - numOBufs, [449](#)
 - oBufs, [449](#)
 - OSEFiltCfg, [449](#)
 - ptrFiltKS, [449](#)
 - sch, [449](#)
- sippPal.h, [781](#)
 - sippPalCriticalSectionBegin, [782](#)
 - sippPalCriticalSectionEnd, [782](#)
 - sippPalFindHighestBit, [782](#)
 - sippPalMemCompare, [782](#)
 - sippPalMemcpy, [782](#)
 - sippPalMemset, [782](#)
 - sippPalPrintInt, [782](#)
 - sippPalTrace, [782](#)
 - sippPalTraceInit, [783](#)
- sippPalCriticalSectionBegin
 - BM/src/leon/sippPlatformAbstractionLayer.c, [811](#)
 - PC/sippPlatformAbstractionLayer.c, [813](#)
 - RTEMS/src/leon/sippPlatformAbstraction-
- Layer.c, [814](#)
- sippPal.h, [782](#)
- sippPalCriticalSectionEnd
 - BM/src/leon/sippPlatformAbstractionLayer.c, [811](#)
 - PC/sippPlatformAbstractionLayer.c, [813](#)
 - RTEMS/src/leon/sippPlatformAbstraction-
- Layer.c, [814](#)
- sippPal.h, [782](#)
- sippPalFindHighestBit
 - BM/src/leon/sippPlatformAbstractionLayer.c, [811](#)
 - PC/sippPlatformAbstractionLayer.c, [813](#)
 - RTEMS/src/leon/sippPlatformAbstraction-
- Layer.c, [814](#)
- sippPal.h, [782](#)
- sippPalMemCompare
 - BM/src/leon/sippPlatformAbstractionLayer.c, [811](#)
 - PC/sippPlatformAbstractionLayer.c, [813](#)
 - RTEMS/src/leon/sippPlatformAbstraction-
- Layer.c, [814](#)
- sippPal.h, [782](#)
- sippPalMemcpy
 - BM/src/leon/sippPlatformAbstractionLayer.c, [811](#)
 - PC/sippPlatformAbstractionLayer.c, [813](#)
 - RTEMS/src/leon/sippPlatformAbstraction-
- Layer.c, [814](#)
- sippPal.h, [782](#)
- sippPalMemset
 - BM/src/leon/sippPlatformAbstractionLayer.c, [811](#)
 - PC/sippPlatformAbstractionLayer.c, [813](#)
 - RTEMS/src/leon/sippPlatformAbstraction-
- Layer.c, [814](#)
- sippPal.h, [782](#)
- sippPalPrintInt
 - BM/src/leon/sippPlatformAbstractionLayer.c, [812](#)
 - RTEMS/src/leon/sippPlatformAbstraction-
- Layer.c, [815](#)
- sippPal.h, [782](#)
- sippPalQuAttach
 - RTEMS/src/leon/sippPlatformAbstraction-
- Layer.c, [815](#)
- sippPalQuCreate
 - RTEMS/src/leon/sippPlatformAbstraction-
- Layer.c, [815](#)
- sippPalQuDestroy
 - RTEMS/src/leon/sippPlatformAbstraction-

- Layer.c, [815](#)
- sippPalQuPost
 - RTEMS/src/leon/sippPlatformAbstraction-Layer.c, [815](#)
- sippPalQuReceive
 - RTEMS/src/leon/sippPlatformAbstraction-Layer.c, [815](#)
- sippPalThreadCreate
 - RTEMS/src/leon/sippPlatformAbstraction-Layer.c, [815](#)
- sippPalThreadTerminate
 - RTEMS/src/leon/sippPlatformAbstraction-Layer.c, [815](#)
- sippPalTrace
 - BM/src/leon/sippPlatformAbstractionLayer.c, [812](#)
 - PC/sippPlatformAbstractionLayer.c, [813](#)
 - RTEMS/src/leon/sippPlatformAbstraction-Layer.c, [815](#)
 - sippPal.h, [782](#)
- sippPalTraceInit
 - BM/src/leon/sippPlatformAbstractionLayer.c, [812](#)
 - PC/sippPlatformAbstractionLayer.c, [813](#)
 - RTEMS/src/leon/sippPlatformAbstraction-Layer.c, [815](#)
 - sippPal.h, [783](#)
- sippPalTypes.h, [783](#), [791](#), [798](#)
- sippPipeGetErrorStatus
 - sipp_ma2x5x.h, [605](#)
 - sippError.c, [636](#)
- sippPipeSessionControl
 - sippInternal.h, [764](#)
 - sippPipeSessionControl.c, [807](#)
- sippPipeSessionControl.c, [807](#)
 - gSippFramework, [807](#)
 - pgSippHW, [807](#)
 - sippPipeSessionControl, [807](#)
 - sippPipeSessionControlInit, [807](#)
- sippPipeSessionControlInit
 - sippInternal.h, [765](#)
 - sippPipeSessionControl.c, [807](#)
- sippPipeSetLinesPerIter
 - sipp_ma2x5x.h, [605](#)
 - sippApi.c, [613](#)
- sippPipeSetNumLinesPerBuf
 - sipp_ma2x5x.h, [605](#)
 - sippApi.c, [613](#)
- SippPipeline
 - sippTypes.h, [828](#)
- SippPipelineS, [449](#)
- allDoneMask, [451](#)
- canRunMask, [451](#)
- cmxMapResult, [451](#)
- dbgLevel, [452](#)
- ddrCmxBackupAdr, [452](#)
- ddrCmxBackupLen, [452](#)
- dmaCmxPop, [452](#)
- dmaCmxPush, [452](#)
- dmaTaskList, [452](#)
- endIter, [452](#)
- errorStatus, [452](#)
- filters, [452](#)
- filtersDMA, [453](#)
- filtersHw, [453](#)
- filtersSvu, [453](#)
- firstIbflUseMask, [453](#)
- firstRunFilts, [453](#)
- firstUseInProg, [453](#)
- firstUseMask, [453](#)
- flags, [453](#)
- gi, [453](#)
- HWPipeID, [454](#)
- hwSippCtxSwMask, [454](#)
- hwSippFirst, [454](#)
- hwSippFltCnt, [454](#)
- ibflIncDelta, [454](#)
- isrFlags, [454](#)
- iteration, [454](#)
- itersLeft, [454](#)
- lineMemPoolBase, [454](#)
- linesPerIter, [455](#)
- mbinImg, [455](#)
- multiHwCtx, [455](#)
- nFilters, [455](#)
- nFiltersDMA, [455](#)
- nFiltersHw, [455](#)
- nFiltersSvu, [455](#)
- nFirstRunFilts, [455](#)
- nIter, [456](#)
- nPadFilters, [456](#)
- numMemRegions, [456](#)
- numSE, [456](#)
- nxtExeNo, [456](#)
- oldRunMask, [456](#)
- pCmxMap, [456](#)
- pSE, [457](#)
- padList, [456](#)
- pfCallback, [456](#)
- pfnSippRunFrameReset, [456](#)
- pfnSippRunIterDone, [457](#)
- pfnSippRunNextIter, [457](#)

- pfnSippRuntime, [457](#)
- pfnSippRuntimeClaimHWResource, [457](#)
- pfnSippScheSetBufConsModels, [457](#)
- pfnSippSchedule, [457](#)
- popCmd, [457](#)
- pushCmd, [457](#)
- runAddSchedCheck, [457](#)
- schedConsumeMask, [457](#)
- schedInfo, [457](#)
- schedInfoCtx, [458](#)
- schedInfoCtxSz, [458](#)
- schedInfoEntries, [458](#)
- setIterMask, [458](#)
- shadowSelect, [458](#)
- sliceSz, [458](#)
- startDelta, [458](#)
- svuCmd, [458](#)
- svuStack, [458](#)
- svuSyncMtx, [458](#)
- svuSyncMtxAddr, [458](#)
- svuSyncMtxParity, [458](#)
- svuSyncSem, [459](#)
- svuWinRegs, [459](#)
- tCMDStartQu, [459](#)
- tCMDUpdateQu, [459](#)
- tCmxDmaQu, [459](#)
- tHeapMCB, [459](#)
- uHWFilterIBufUsageMask, [459](#)
- uHWFilterOBufUsageMask, [459](#)
- uHWFilterUsageBitMask, [459](#)
- useCmxRegMap, [459](#)
- useSyncRuntime, [460](#)
- sippPipelineValidate.c, [806](#)
 - sippAnalysePipe, [806](#)
 - sippValidatePipe, [806](#)
- SippPixelChunkPos, [460](#)
 - XPos, [460](#)
 - YPos, [460](#)
- sippPlatform.h, [807](#)
- sippPlatform_ma2x5x.h, [808](#)
 - ALIGNED, [809](#)
 - CMX_BSS, [809](#)
 - CMX_DATA, [809](#)
 - CMX_RODATA, [809](#)
 - CMX_TEXT, [809](#)
 - DBG_PRINT, [809](#)
 - DDR_BSS, [809](#)
 - DDR_DATA, [809](#)
 - DDR_RODATA, [809](#)
 - DDR_TEXT, [809](#)
 - INLINE, [809](#)
 - NOP, [809](#)
 - SECTION, [809](#)
 - SHAVE_MAIN, [809](#)
 - SIPP_MBIN, [809](#)
 - SIPP_PC, [810](#)
 - SVU_SYM, [810](#)
 - scGetShaveNumber, [810](#)
 - U32_MEMCPY, [810](#)
 - U32_MEMSET, [810](#)
 - UNUSED, [810](#)
 - unitTestFinalReport, [810](#)
 - VCS_PRINT_INT, [810](#)
- sippPlatformAbstractionLayer.c, [811–813](#)
- sippPlatformInit
 - sippTestCommon.c, [822](#)
 - sippTestCommon_ma2x5x.h, [823](#)
- sippPlatformInitAsync
 - sippTestCommon.c, [822](#)
 - sippTestCommon_ma2x5x.h, [823](#)
- sippPoolsUsage
 - sippMem.c, [774](#)
- sippPrintSliceWidth
 - sippDbg.c, [630](#)
 - sippInternal.h, [765](#)
- sippProcessFrame
 - sipp_ma2x5x.h, [605](#)
 - sippApi.c, [613](#)
- sippProcessFrameNB
 - sipp_ma2x5x.h, [605](#)
 - sippApi.c, [613](#)
- sippProcessIters
 - sipp_ma2x5x.h, [605](#)
- sippProcessItersNB
 - sipp_ma2x5x.h, [605](#)
- sippProcessSchedData
 - sippCoreApi.c, [624](#)
- SippQu
 - SIPP_ACCESS_SCHEDULER, [426](#)
- sippRdFileU8
 - sipp_ma2x5x.h, [605](#)
 - sippDbg.c, [630](#)
- sippRdFileU8toF16
 - sipp_ma2x5x.h, [605](#)
 - sippDbg.c, [630](#)
- sippRegisterEventCallback
 - sipp_ma2x5x.h, [605](#)
 - sippApi.c, [613](#)
- sippRemoveEntriesList
 - sippOPipeSchedulingEntity.c, [781](#)
- sippReschedulePipeline
 - sipp_ma2x5x.h, [605](#)

- sippApi.c, [613](#)
- sippRescheduleRequest
 - sippApiInternal.c, [615](#)
 - sippInternal.h, [765](#)
- sippResetFilterVariables
 - sippApiInternal.c, [615](#)
 - sippInternal.h, [765](#)
- sippRunDmaCQ
 - sippCmxDmaIf.c, [622](#)
 - sippInternal.h, [765](#)
- sippRunDmaCQDrain
 - sippCmxDmaIf.c, [622](#)
- sippRunFrameReset
 - sippTypes.h, [828](#)
- sippRunIterDoneFunc
 - sippTypes.h, [828](#)
- sippRunItersRequest
 - sippApiInternal.c, [615](#)
 - sippInternal.h, [765](#)
- sippRunNextIterFunc
 - sippTypes.h, [828](#)
- sippRuntimeClaimHWResourceFunc
 - sippTypes.h, [828](#)
- sippRuntimeFunc
 - sippTypes.h, [828](#)
- sippSWInit
 - sippApiInternal.c, [615](#)
 - sippInternal.h, [765](#)
- SippSchEnt
 - sippTypes.h, [828](#)
- SippSchEntS, [460](#)
 - canConsume, [461](#)
 - canRunC, [461](#)
 - canRunP, [461](#)
 - cons, [461](#)
 - consIbufIdx, [461](#)
 - currKSOffset, [461](#)
 - dbgJustRoll, [462](#)
 - iBufCtx, [462](#)
 - iBufs, [462](#)
 - KSIterList, [462](#)
 - linesPerIter, [462](#)
 - maxLineRequired, [462](#)
 - maxOutputH, [462](#)
 - minLineRequired, [462](#)
 - numConsumers, [462](#)
 - numIBufs, [462](#)
 - numLineRuns, [463](#)
 - numOBufs, [463](#)
 - oBufs, [463](#)
 - pRunAdditionalParam, [463](#)

- parentKS, [463](#)
- parentKSMin, [463](#)
- pfRunAdditionalCheck, [463](#)
- pipeSEId, [463](#)
- ptrFiltKS, [463](#)
- recordParentKS, [464](#)
- runPadCheck, [464](#)
- sippSchTypes.h, [817](#)
 - pSippManagedBufSchedInfo, [817](#)
 - RunStatus, [817](#)
 - SchedAddCheck, [817](#)
- sippSchedFunc
 - sippTypes.h, [828](#)
- sippSchedSetBufConsModels
 - sippTypes.h, [828](#)
- sippScheduleIsr.c, [815](#)
 - pgSippHW, [817](#)
 - sippCheckIterComplete, [816](#)
 - sippGlobalOBFLIncStatus, [817](#)
 - sippIbflDecHandler, [816](#)
 - sippObflIncHandler, [816](#)
 - sippSvuDoneIrqHandler, [816](#)
- sippSessionControl.h, [818](#)
 - SIPP_PIPE_STATE, [818](#)
- sippSetBufLatenciesMipiRx
 - sippHwMipiRx.c, [749](#)
- sippSetBufLatenciesMipiTx
 - sippHwMipiTx.c, [750](#)
- sippSetBufLatenciesSigma
 - sippHwSigma.c, [753](#)
- sippSetCircularLnBuffBase
 - sippMem.c, [774](#)
- sippSetInitMask
 - sipp_ma2x5x.h, [605](#)
 - sippCoreApi.c, [624](#)
- sippSetOBufLevelsMipiRx0
 - sippHwMipiRx.c, [749](#)
- sippSetOBufLevelsMipiRx1
 - sippHwMipiRx.c, [749](#)
- sippSetOBufLevelsMipiRx2
 - sippHwMipiRx.c, [749](#)
- sippSetOBufLevelsMipiRx3
 - sippHwMipiRx.c, [749](#)
- sippSetOBufLevelsMipiTx0
 - sippHwMipiTx.c, [750](#)
- sippSetOBufLevelsMipiTx1
 - sippHwMipiTx.c, [750](#)
- sippSetOBufLevelsSigma
 - sippHwSigma.c, [753](#)
- sippSetSliceSize
 - sipp_ma2x5x.h, [605](#)

- sippCoreApi.c, 624
- sippSetupSvus
 - sippInternal.h, 765
 - sippShaveIf.c, 820
- sippShave.c, 818
 - dbg_svu_no, 819
 - SHAVE_MAIN, 819
 - scGetShaveNumber, 819
 - sipp_pl, 819
 - sippGetChunkStartPixelPos, 819
 - sippHorizontalPadding, 819
- sippShaveDebug
 - sippShvDbg.c, 821
- sippShaveIf.c, 820
 - SHAVE_MAIN, 820
 - sippKickShaveM1PC, 820
 - sippKickSvus, 820
 - sippSetupSvus, 820
 - sippStopSvus, 820
 - sippWaitShave, 820
- sippShaveMacros.h, 820
- sippShaveSym.h, 821
- sippShvDbg.c, 821
 - sipp_pl, 821
 - sippShaveDebug, 821
- sippSizeList
 - sippHeap.c, 649
- sippStopSvus
 - sipp_ma2x5x.h, 605
 - sippInternal.h, 765
 - sippShaveIf.c, 820
- sippSvuDoneIrqHandler
 - sippInternal.h, 765
 - sippScheduleIsr.c, 816
- sippTerm
 - sipp_ma2x5x.h, 605
 - sippApi.c, 613
- sippTermInternal
 - sippApiInternal.c, 615
 - sippInternal.h, 765
- sippTestCommon.c, 821
 - sippPlatformInit, 822
 - sippPlatformInitAsync, 822
- sippTestCommon.h, 822
- sippTestCommon_ma2x5x.h, 822
 - mbinImgSipp, 823
 - sippPlatformInit, 823
 - sippPlatformInitAsync, 823
- sippTestCrcCheck
 - sipp_ma2x5x.h, 606
 - sippDbg.c, 630
- sippThread.c, 823
- sippTypes.h, 823
 - eDmaMode, 830
 - eSIPP_STATUS, 830
 - FALSE, 826
 - FnSvuRun, 827
 - HPadInfo, 827
 - HorizPaddingOff, 827
 - NULL, 826
 - pSippCmxBufferMap, 827
 - pSippFilter, 827
 - pSippFilterSchedInfo, 827
 - pSippPipeline, 827
 - pSippVPhysMap, 827
 - ParentInfo, 827
 - psSchLineBuffer, 827
 - psSippCMDQu, 828
 - psSippCMDQuEntry, 828
 - psSippCdmaQu, 827
 - psSippCdmaQuEntry, 828
 - ptRTStats, 828
 - ptSippMCB, 828
 - sSchLineBuffer, 828
 - sSippCMDQu, 829
 - sSippCMDQuEntry, 829
 - sSippCdmaQu, 828
 - sSippCdmaQuEntry, 829
 - SchedFuncAsk, 828
 - SchedIBufUsageInfo, 828
 - SchedInfo, 828
 - SippCallback, 828
 - SippCmxBufferMap, 828
 - sippEventCallback_t, 828
 - SippFilter, 828
 - SippFilterSchedInfo, 828
 - SippHwBuf, 828
 - SippPipeline, 828
 - sippRunFrameReset, 828
 - sippRunIterDoneFunc, 828
 - sippRunNextIterFunc, 828
 - sippRuntimeClaimHWResourceFunc, 828
 - sippRuntimeFunc, 828
 - SippSchEnt, 828
 - sippSchedFunc, 828
 - sippSchedSetBufConsModels, 828
 - SippVirtualPool, 831
 - TRUE, 827
 - tSippVPhysMap, 829
- sippTypesPrivate.h, 832
 - eSIPP_HW_STATE, 834
 - eSippCommand, 834

- eSippObufControl, [835](#)
- eSippPipeState, [835](#)
- FnCheckOpipeCons, [834](#)
- FnGetIBufCtx, [834](#)
- FnGetIBufIds, [834](#)
- FnGetOBufIds, [834](#)
- FnHwFltInit, [834](#)
- FnHwFltLoad, [834](#)
- FnHwSetObufLatencies, [834](#)
- FnHwSetupUpdate, [834](#)
- FnSetOBufs, [834](#)
- gSipp, [835](#)
- pSIPP_HW_SESSION, [834](#)
- pSippHeapCB, [834](#)
- pSippMemRegionListNode, [834](#)
- pmemRegDescriptor, [834](#)
- ptMLPIStartCQCtrl, [834](#)
- ptSippPhysicalPool, [834](#)
- ptSippPipelineSuper, [834](#)
- SIPP_HCB_SIZE, [833](#)
- SippCommandData, [834](#)
- sippUpdateDmaAddr
 - sippCmxDmaIf.c, [622](#)
- sippUpdateDmaAddrCQ
 - sippCmxDmaIf.c, [622](#)
- sippUsingPrecompSched
 - sippCoreUtils.c, [628](#)
 - sippInternal.h, [765](#)
- sippUtilComputeFp16Lut
 - sipp_ma2x5x.h, [606](#)
 - sippUtils.c, [836](#)
- sippUtilComputeFp16LutChannelMode
 - sipp_ma2x5x.h, [606](#)
 - sippUtils.c, [837](#)
- sippUtilOrderPixels
 - sippInternal.h, [765](#)
 - sippUtils.c, [837](#)
- sippUtilPrintFp16Lut
 - sipp_ma2x5x.h, [606](#)
 - sippUtils.c, [837](#)
- sippUtils.c, [835](#)
- CCM_COEF, [836](#)
- lumaGenLut, [836](#)
- packColCombCCM, [836](#)
- packConv3x3CCM, [836](#)
- packConv5x5CCM, [836](#)
- packLumaDnsGaussLut, [836](#)
- sharpenSigmaToCoefficients, [836](#)
- sippFindInList, [836](#)
- sippListSort, [836](#)
- sippUtilComputeFp16Lut, [836](#)
- sippUtilComputeFp16LutChannelMode, [837](#)
- sippUtilOrderPixels, [837](#)
- sippUtilPrintFp16Lut, [837](#)
- sippWait, [837](#)
- SippVPhysMapS, [464](#)
 - physPoolMap, [464](#)
- sippValidatePipe
 - sippInternal.h, [765](#)
 - sippPipelineValidate.c, [806](#)
- SippVirtualPool
 - sippTypes.h, [831](#)
- sippWait
 - sippUtils.c, [837](#)
- sippWaitDma
 - sippCmxDmaIf.c, [622](#)
 - sippInternal.h, [766](#)
- sippWaitShave
 - sippInternal.h, [766](#)
 - sippShaveIf.c, [820](#)
- sippWrFileF16toU8
 - sipp_ma2x5x.h, [606](#)
 - sippDbg.c, [630](#)
- sippWrFileU8
 - sipp_ma2x5x.h, [606](#)
 - sippDbg.c, [630](#)
- size
 - sSippCdmaQuS, [468](#)
- sizeA
 - LutParam, [397](#)
- sizeB
 - LutParam, [397](#)
- sizeRemaining
 - memRegDescriptor, [399](#)
- slice0Addr
 - SippMemRegionListNode, [448](#)
- sliceFirst
 - CommInfo, [345](#)
- sliceLast
 - CommInfo, [345](#)
- sliceSize
 - CommInfo, [345](#)
- sliceSz
 - SippPipelineS, [458](#)
- sliceWidth
 - SippFilterS, [441](#)
- sliceWidthLastSvu
 - SippFilterS, [442](#)
- slope
 - GreyDesatParam, [386](#)
 - MixMedianParam, [408](#)
- Sobel, [314](#)

- SHAVE_SYM_EXPORT, 314
- svuSobel, 314
- sobel
 - svuSobel.c, 964
- sobel.h, 837
- srcLeftO
 - HorizPaddingOffS, 389
- srcLnS
 - DmaParam, 376
- srcRightO
 - HorizPaddingOffS, 389
- ssd11x11.h, 838
- ssd5x5.h, 838
- ssd7x7U8ToU32.h, 839
- ssdPointLine7x7U8U32.h, 839
- st_Y
 - CropParam, 367
- stackPointer
 - ExtStatsSatPixelsU32Param, 382
 - PpAf, 416
- stackSize
 - SIPP_PAL_THREAD, 430
- start
 - tSippPhysicalPool, 481
- startBicubic.h, 840
 - SHAVE_SYM_EXPORT, 840
 - svuStartBicubic, 840
- StartBicubicParam, 469
 - bpp, 470
 - cmd, 470
 - firstShave, 470
 - height, 470
 - input, 470
 - lastShave, 470
 - output, 470
 - shaveCount, 470
 - width, 470
- startDelta
 - SippPipelineS, 458
- startMipiTxLoopback
 - sippHwDefs_ma2x5x.h, 738
- statsAWBSatPixels
 - svuStatsAwbSatPixels.c, 968
- statsAWBSatPixelsU32
 - svuStatsAwbSatPixelsU32.c, 968
- statsAf0
 - PpAf, 416
- statsAwbSatPixels.h, 840
 - SHAVE_SYM_EXPORT, 840
 - svuStatsAwbSatPixels, 840
- StatsAwbSatPixelsParam, 470

- AeAwbStatsCfg, 471
- crtPaxel, 471
- crtPosInPaxel, 471
- firstPatchX, 471
- firstPatchY, 471
- nPatchesX, 471
- nPatchesY, 471
- NextVerticalStartPos, 471
- patchGapX, 472
- patchGapY, 472
- patchHeight, 472
- patchWidth, 472
- runNr, 472
- satThresh, 472
- statsOutput, 472
- StatsAwbSatPixelsParamU32, 472
 - AeAwbStatsCfg, 473
 - crtPaxel, 473
 - crtPosInPaxel, 473
 - firstPatchX, 473
 - firstPatchY, 473
 - nPatchesX, 473
 - nPatchesY, 473
 - NextVerticalStartPos, 473
 - patchGapX, 473
 - patchGapY, 473
 - patchHeight, 474
 - patchWidth, 474
 - runNr, 474
 - satThresh, 474
 - statsOutput, 474
- statsAwbSatPixelsU32.h, 840
 - svuStatsAwbSatPixelsU32, 841
- statsBase
 - RawParam, 420
- statsFrmDim
 - RawParam, 421
- statsOutput
 - CombDecimStatsGainsParam, 344
 - StatsAwbSatPixelsParam, 472
 - StatsAwbSatPixelsParamU32, 474
- statsPatchCfg
 - RawParam, 421
- statsPatchSkip
 - RawParam, 421
- statsPatchStart
 - RawParam, 421
- statsPlanes
 - RawParam, 421
- statsSaturatePxI
 - ExtStatsSatPixelsU32Param, 382

- statsThresh
 - RawParam, [421](#)
- status
 - positionKernelParam, [414](#)
- step
 - HistogramStatParam, [388](#)
- strength
 - PurpleFlareParam, [417](#)
 - RandNoiseFp16Param, [417](#)
 - RandNoiseParam, [418](#)
 - UsmParam, [486](#)
- subSampleScale
 - SippFilterSchedInfoS, [443](#)
- subpixelFilter.h, [841](#)
 - SubpixelFilterParam, [842](#)
 - svuSubpixelFilter, [842](#)
- SubpixelFilterParam, [474](#)
 - LUT, [474](#)
 - numDisp, [474](#)
 - numFractionalBits, [475](#)
 - subpixelFilter.h, [842](#)
- subs05sync7
 - svuScale05Lanc7HV.c, [957](#)
- Sum of Absolute Differences 11x11, [296](#)
 - SHAVE_SYM_EXPORT, [296](#)
 - svuSAD11x11, [296](#)
- Sum of Absolute Differences 5x5, [297](#)
 - SHAVE_SYM_EXPORT, [297](#)
 - svuSAD5x5, [297](#)
- Sum of Squared Differences 11x11, [315](#)
 - SHAVE_SYM_EXPORT, [315](#)
 - svuSSD11x11, [315](#)
- Sum of Squared Differences 5x5, [316](#)
 - SHAVE_SYM_EXPORT, [316](#)
 - svuSSD5x5, [316](#)
- Sum of Squared Differences 7x7, [318](#)
 - SHAVE_SYM_EXPORT, [318](#)
 - svuSsdPointLine7x7U8U32, [318](#)
- Sum of Squared Differences 7x7 (U8 to U32), [317](#)
 - SHAVE_SYM_EXPORT, [317](#)
 - svuSSD7x7U8ToU32, [317](#)
- sum_all_green
 - AF_paxel_statistics, [331](#)
- sumOfAbsDiff11x11
 - svuSAD11x11.c, [954](#)
- sumOfAbsDiff5x5
 - svuSAD5x5.c, [954](#)
- sumOfSquaredDiff11x11
 - svuSSD11x11.c, [964](#)
- sumOfSquaredDiff5x5
 - svuSSD5x5.c, [965](#)
- sumOfSquaredDiff7x7U8ToU32
 - svuSSD7x7U8ToU32.c, [966](#)
- svuAbsdiff
 - Absolute difference, [115](#)
- svuAbsdiff.c, [842](#)
 - AbsoluteDiff, [842](#)
- svuAccumulateSquare
 - Accumulate Square, [116](#)
- svuAccumulateSquare.c, [842](#)
 - AccumulateSquare, [843](#)
- svuAccumulateWeighted
 - Accumulate Weighted, [117](#)
- svuAccumulateWeighted.c, [843](#)
 - AccumulateWeighted, [843](#)
- svuArithmeticAdd
 - Arithmetic addition, [118](#)
- svuArithmeticAdd.c, [844](#)
 - arithmeticAddImplementation, [844](#)
- svuArithmeticAddmask
 - Arithmetic addition with mask, [119](#)
- svuArithmeticAddmask.c, [844](#)
 - arithmeticAddmaskImplementation, [845](#)
- svuArithmeticSub
 - Arithmetic, [126](#)
- svuArithmeticSub.c, [845](#)
 - arithmeticSubImplementation, [845](#)
- svuArithmeticSubFp16ToFp16
 - Arithmetic subtraction fp16, [127](#)
- svuArithmeticSubFp16ToFp16.c, [845](#)
 - arithmeticSubFp16ToFp16, [846](#)
- svuArithmeticSubmask
 - Arithmetic subtraction with mask, [128](#)
- svuArithmeticSubmask.c, [846](#)
 - arithmeticSubmaskImplementation, [846](#)
- svuAvg
 - Average, [129](#)
- svuAvg.c, [847](#)
- svuBilateral5x5
 - bilateral5x5, [130](#)
- svuBilateral5x5.c, [847](#)
 - bilateralVal, [847](#)
 - max, [848](#)
 - mvcvBilateral5x5, [848](#)
- svuBitwiseAnd
 - Bitwise And, [131](#)
- svuBitwiseAnd.c, [848](#)
 - bitwiseAnd, [848](#)
- svuBitwiseAndMask.c, [848](#)
 - bitwiseAndMask, [849](#)
- svuBitwiseNot
 - Bitwise Not, [133](#)

- svuBitwiseNot.c, [849](#)
 - bitwiseNot, [849](#)
- svuBitwiseOr
 - Bitwise Or, [134](#)
- svuBitwiseOr.c, [850](#)
 - bitwiseOr, [850](#)
- svuBitwiseOrMask
 - Bitwise Or with mask, [135](#)
- svuBitwiseOrMask.c, [850](#)
 - bitwiseOrMask, [851](#)
- svuBitwiseXor
 - Bitwise Xor, [136](#)
- svuBitwiseXor.c, [851](#)
 - bitwiseXor, [851](#)
- svuBitwiseXorMask
 - Bitwise Xor with mask, [137](#)
- svuBitwiseXorMask.c, [851](#)
 - bitwiseXorMask, [852](#)
- svuBoxFilter
 - Generic Box Filter, [139](#)
- svuBoxFilter.c, [852](#)
 - BOX_FILT, [852](#)
 - boxfilter, [853](#)
- svuBoxFilter11x11
 - Box Filter 11x11, [140](#)
- svuBoxFilter11x11.c, [853](#)
 - boxfilter11x11, [853](#)
- svuBoxFilter13x13
 - Box Filter 13x13, [141](#)
- svuBoxFilter13x13.c, [854](#)
 - boxfilter13x13, [854](#)
- svuBoxFilter15x15
 - Box Filter 15x15, [142](#)
- svuBoxFilter15x15.c, [854](#)
 - boxfilter15x15, [854](#)
- svuBoxFilter3x3
 - Box Filter 3x3, [143](#)
- svuBoxFilter3x3.c, [855](#)
 - boxfilter3x3, [855](#)
- svuBoxFilter5x5
 - Box Filter 5x5, [144](#)
- svuBoxFilter5x5.c, [855](#)
 - boxfilter5x5, [856](#)
- svuBoxFilter7x7
 - Box Filter 7x7, [145](#)
- svuBoxFilter7x7.c, [856](#)
 - boxfilter7x7, [856](#)
- svuBoxFilter9x9
 - Box Filter 9x9, [146](#)
- svuBoxFilter9x9.c, [857](#)
 - boxfilter9x9, [857](#)
- svuCannyEdgeDetection
 - Canny Edge Detection, [147](#)
- svuCannyEdgeDetection.c, [857](#)
 - buffer, [858](#)
 - canny, [858](#)
 - MAX_WIDTH, [858](#)
 - PADDING, [858](#)
- svuCensusMatching16
 - censusMatching16, [148](#)
- svuCensusMatching16.c, [858](#)
 - mvcvCensusMatching16, [859](#)
- svuCensusMatching32
 - censusMatching32, [149](#)
- svuCensusMatching32.c, [859](#)
 - mvcvCensusMatching32, [859](#)
- svuCensusMatching64
 - censusMatching64, [150](#)
- svuCensusMatching64.c, [860](#)
 - mvcvCensusMatching64, [860](#)
- svuCensusMatching65
 - censusMatching65, [151](#)
- svuCensusMatching65.c, [860](#)
 - mvcvCensusMatching65, [861](#)
- svuCensusMatchingPyr
 - censusMatchingPyr, [152](#)
- svuCensusMatchingPyr.c, [861](#)
 - mvcvCensusMatchingPyr, [861](#)
- svuCensusMin16
 - censusMin16, [153](#)
- svuCensusMin16.c, [862](#)
 - mvcvCensusMin16, [862](#)
- svuCensusMin64
 - censusMin64, [154](#)
- svuCensusMin64.c, [862](#)
 - mvcvCensusMin64, [862](#)
- svuCensusMin65
 - censusMin65, [155](#)
- svuCensusMin65.c, [863](#)
 - mvcvCensusMin65, [863](#)
- svuCensusMin7
 - censusMin7, [156](#)
- svuCensusMin7.c, [863](#)
 - mvcvCensusMin7, [864](#)
- svuCensusTransform5x5
 - CensusTransform5x5, [157](#)
- svuCensusTransform5x5.c, [864](#)
 - mvcvCensusTransform5x5, [864](#)
- svuChannelExtract
 - channelExtract, [158](#)
- svuChannelExtract.c, [865](#)
 - channelExtract, [865](#)

- svuChromaBlock
 - Chroma Block, [159](#)
- svuChromaBlock.c, [865](#)
- svuCmd
 - SippPipelineS, [458](#)
- svuCombDecimDemosaicAwbGains
 - combDecimDemosaicAwbGains.h, [513](#)
 - svuCombDecimDemosaicAwbGains.c, [867](#)
- svuCombDecimDemosaicAwbGains.c, [866](#)
 - CLAMPZ255, [866](#)
 - combDecimDemosaicAwbGainsBG, [866](#)
 - combDecimDemosaicAwbGainsGB, [866](#)
 - combDecimDemosaicAwbGainsGR, [866](#)
 - combDecimDemosaicAwbGainsRG, [866](#)
 - svuCombDecimDemosaicAwbGains, [867](#)
- svuCombDecimDemosaicAwbGainsStats
 - combDecimDemosaicAwbGainsStats.c, [514](#)
 - combDecimDemosaicAwbGainsStats.h, [514](#)
- svuContrast
 - Contrast, [160](#)
- svuContrast.c, [867](#)
- svuConv11x11
 - Convolution 11x11, [161](#)
- svuConv11x11.c, [867](#)
 - Convolution11x11, [867](#)
- svuConv15x1
 - Convolution 15x1, [162](#)
- svuConv15x1.c, [868](#)
 - Convolution15x1, [868](#)
- svuConv1x15
 - Convolution 1x15, [163](#)
- svuConv1x15.c, [868](#)
 - Convolution1x15, [868](#)
- svuConv1x5
 - Convolution 1x5, [164](#)
- svuConv1x5.c, [869](#)
 - Convolution1x5, [869](#)
- svuConv1x5Fp16ToFp16
 - Convolution 1x5 Fp16ToFp16, [165](#)
- svuConv1x5Fp16ToFp16.c, [869](#)
 - Convolution1x5Fp16ToFp16, [870](#)
- svuConv1x7
 - Convolution 1x7, [166](#)
- svuConv1x7.c, [870](#)
 - Convolution1x7, [870](#)
- svuConv1x7Fp16ToFp16
 - Convolution 1x7 Fp16ToFp16, [167](#)
- svuConv1x7Fp16ToFp16.c, [870](#)
 - Convolution1x7Fp16ToFp16, [871](#)
- svuConv1x9
 - Convolution 1x9, [168](#)
- svuConv1x9.c, [871](#)
 - Convolution1x9, [871](#)
- svuConv3x3
 - Convolution 3x3, [169](#)
- svuConv3x3.c, [872](#)
 - conv3x3FilterImplementation, [872](#)
- svuConv3x3Fp16ToFp16
 - Convolution 3x3 Fp16ToFp16, [170](#)
- svuConv3x3Fp16ToFp16.c, [872](#)
 - convolution3x3Fp16ToFp16Implementation, [872](#)
- svuConv5x1
 - Convolution 5x1, [171](#)
- svuConv5x1.c, [873](#)
 - Convolution5x1, [873](#)
- svuConv5x1Fp16ToFp16
 - Convolution 5x1 Fp16ToFp16, [172](#)
- svuConv5x1Fp16ToFp16.c, [873](#)
 - Convolution5x1Fp16ToFp16, [874](#)
- svuConv5x5
 - Convolution 5x5, [173](#)
- svuConv5x5.c, [874](#)
 - Convolution5x5, [874](#)
- svuConv5x5Fp16ToFp16
 - Convolution 5x5 Fp16ToFp16, [174](#)
- svuConv5x5Fp16ToFp16.c, [874](#)
 - Convolution5x5Fp16ToFp16, [875](#)
- svuConv7x1
 - Convolution 7x1, [175](#)
- svuConv7x1.c, [875](#)
 - Convolution7x1, [875](#)
- svuConv7x1Fp16ToFp16
 - Convolution 7x1 Fp16ToFp16, [176](#)
- svuConv7x1Fp16ToFp16.c, [876](#)
 - convolution7x1Fp16ToFp16, [876](#)
- svuConv7x7
 - Convolution 7x7, [177](#)
- svuConv7x7.c, [876](#)
 - convolution7x7, [876](#)
- svuConv7x7Fp16ToFp16
 - Convolution 7x7 Fp16ToFp16, [178](#)
- svuConv7x7Fp16ToFp16.c, [877](#)
 - convolution7x7Fp16ToFp16, [877](#)
- svuConv7x7Fp16ToU8
 - Convolution 7x7 Fp16ToU8, [179](#)
- svuConv7x7Fp16ToU8.c, [877](#)
 - convolution7x7Fp16ToU8, [878](#)
- svuConv9x1
 - Convolution 9x1, [180](#)
- svuConv9x1.c, [878](#)
 - Convolution9x1, [878](#)

- svuConv9x9
 - Convolution 9x9, [181](#)
- svuConv9x9.c, [878](#)
 - Convolution9x9, [879](#)
- svuConv9x9Fp16ToFp16
 - Convolution 9x9 Fp16ToFp16, [182](#)
- svuConv9x9Fp16ToFp16.c, [879](#)
 - Convolution9x9Fp16ToFp16, [879](#)
- svuConvGeneric
 - Generic Convolution, [190](#)
- svuConvGeneric.c, [883](#)
 - Convolution, [884](#)
- svuConvSeparable11x11
 - Convolution Separable 11x11, [191](#)
- svuConvSeparable11x11.c, [884](#)
 - convSeparable11x11, [884](#)
- svuConvSeparable11x11Fp16ToFp16
 - Convolution Separable 11x11 Fp16ToFp16, [192](#)
- svuConvSeparable11x11Fp16ToFp16.c, [884](#)
 - convSeparable11x11Fp16ToFp16, [885](#)
- svuConvSeparable3x3
 - Convolution Separable 3x3, [193](#)
- svuConvSeparable3x3.c, [885](#)
 - convSeparable3x3, [885](#)
- svuConvSeparable3x3Fp16ToFp16
 - Convolution Separable 3x3 Fp16ToFp16, [194](#)
- svuConvSeparable3x3Fp16ToFp16.c, [885](#)
 - convSeparable3x3Fp16ToFp16, [886](#)
- svuConvSeparable5x5
 - Convolution Separable 5x5, [195](#)
- svuConvSeparable5x5.c, [886](#)
 - convSeparable5x5, [886](#)
- svuConvSeparable5x5Fp16ToFp16
 - Convolution Separable 5x5 Fp16ToFp16, [196](#)
- svuConvSeparable5x5Fp16ToFp16.c, [887](#)
 - convSeparable5x5Fp16ToFp16, [887](#)
- svuConvSeparable7x7
 - Convolution Separable 7x7, [197](#)
- svuConvSeparable7x7.c, [887](#)
 - convSeparable7x7, [888](#)
- svuConvSeparable7x7Fp16ToFp16
 - Convolution Separable 7x7 Fp16ToFp16, [198](#)
- svuConvSeparable7x7Fp16ToFp16.c, [888](#)
 - convSeparable7x7Fp16ToFp16, [888](#)
- svuConvSeparable9x9
 - Convolution Separable 9x9, [199](#)
- svuConvSeparable9x9.c, [888](#)
 - convSeparable9x9, [889](#)
- svuConvSeparable9x9Fp16ToFp16
 - Convolution Separable 9x9 Fp16ToFp16, [200](#)
- svuConvSeparable9x9Fp16ToFp16.c, [889](#)
 - convSeparable9x9Fp16ToFp16, [889](#)
- svuConvYuv444.c, [890](#)
 - SCALE, [890](#)
- svuConvert16bppTo8bpp
 - Convert 16bpp To 8bpp, [183](#)
- svuConvert16bppTo8bpp.c, [880](#)
 - CLAMPU8, [880](#)
- svuConvertF16ToU8
 - Convert F16 To U8, [184](#)
- svuConvertF16ToU8.c, [880](#)
 - convertF16ToU8, [880](#)
- svuConvertFrom12BppTo8Bpp
 - 12Bpp to 8Bpp conversion, [185](#)
- svuConvertFrom12BppTo8Bpp.c, [881](#)
 - mvcvConvert12BppTo8Bpp, [881](#)
- svuConvertPFp16U16
 - Convert Fp16 to U16, [186](#)
- svuConvertPFp16U16.c, [881](#)
 - MAX_U16_VAL, [882](#)
- svuConvertPU16Fp16
 - Convert U16 to Fp16, [187](#)
- svuConvertPU16Fp16.c, [882](#)
 - MAX_U16_VAL, [882](#)
- svuConvertU8ToF16
 - Convert U8 To F16, [188](#)
- svuConvertU8ToF16.c, [882](#)
 - convertU8ToF16, [882](#)
- svuConvertYUV400ToYUV422
 - YUV400 to YUV422 conversion, [189](#)
- svuConvertYUV400ToYUV422.c, [883](#)
 - ConvertYUV400ToYUV422, [883](#)
- svuCopy
 - Copy, [202](#)
- svuCopy.c, [890](#)
- svuCornerMinEigenVal
 - Corner Min Eigenvalue, [203](#)
- svuCornerMinEigenVal.c, [890](#)
 - buffer, [891](#)
 - CornerMinEigenVal, [891](#)
 - MAX_WIDTH, [891](#)
 - PADDING, [891](#)
- svuCornerMinEigenValpatched
 - Corner Min Eigenvalue Patched, [204](#)
- svuCornerMinEigenValpatched.c, [892](#)
 - CornerMinEigenVal_patched, [892](#)
- svuCrop
 - CV, [208](#)
- svuCrop.c, [892](#)
- svuCropCvtPlaneMode
 - CV, [208](#)

- svuCvtColorChromaYUVToNV12
 - cropCvtPlaneMode, [893](#)
- svuCvtColorChromaYUVToNV12.h, [540](#)
- svuCvtColorChromaYUVToNV12.c, [894](#)
- svuCvtColorChromaYUVToNV12.c, [893](#)
 - cvtColorChromaYUV420ToNV12, [893](#)
 - cvtColorChromaYUV444ToNV12, [893](#)
 - svuCvtColorChromaYUVToNV12, [894](#)
- svuCvtColorRGBToYUV422
 - RGB to YUV422 conversion, [222](#)
- svuCvtColorRGBToYUV422.c, [900](#)
 - cvtColorKernelRGBToYUV422, [900](#)
- svuCvtColorRGBfp16ToLumaU8
 - RGB(fp16) to Luma(u8) conversion, [211](#)
- svuCvtColorRGBfp16ToLumaU8.c, [895](#)
 - cvtColorRGBfp16ToLumaU8, [895](#)
- svuCvtColorRGBfp16ToUV420U8
 - RGB(fp16) to UV420(u8) conversion, [212](#)
- svuCvtColorRGBfp16ToUV420U8.c, [895](#)
 - cvtColorRGBfp16ToUV420U8, [895](#)
- svuCvtColorRGBtoChromaNV12
 - RGB to Chroma NV12 conversion, [213](#)
- svuCvtColorRGBtoChromaNV12.c, [896](#)
 - cvtColorRGBtoChromaNV12, [896](#)
- svuCvtColorRGBtoLuma
 - ISP, [218](#)
- svuCvtColorRGBtoLuma.c, [896](#)
 - cvtColorRGBtoLuma, [897](#)
- svuCvtColorRGBtoLumaNV12
 - RGB to Luma NV12 conversion, [219](#)
- svuCvtColorRGBtoLumaNV12.c, [897](#)
 - cvtColorRGBtoLumaNV12, [897](#)
- svuCvtColorRGBtoUV
 - RGB to UV conversion, [220](#)
- svuCvtColorRGBtoUV.c, [898](#)
 - cvtColorRGBtoUV, [898](#)
- svuCvtColorRGBtoUV420
 - RGB to UV420 conversion, [221](#)
- svuCvtColorRGBtoUV420.c, [898](#)
 - cvtColorRGBtoUV420, [898](#)
- svuCvtColorYUV422ToRGB
 - YUV422 to RGB conversion, [223](#)
- svuCvtColorYUV422ToRGB.c, [900](#)
 - cvtColorKernelYUV422ToRGB, [901](#)
- svuCvtColorYUVToRGB
 - YUV to RGB conversion, [224](#)
- svuCvtColorYUVToRGB.c, [901](#)
 - cvtColorKernelYUVToRGB, [901](#)
- svuDilate3x3
 - Dilate 3x3, [225](#)
- svuDilate3x3.c, [901](#)
 - Dilate3x3, [902](#)
- svuDilate5x5
 - Dilate 5x5, [226](#)
- svuDilate5x5.c, [902](#)
 - Dilate5x5, [902](#)
- svuDilate7x7
 - Dilate 7x7, [227](#)
- svuDilate7x7.c, [902](#)
 - Dilate7x7, [903](#)
- svuDilateGeneric
 - Generic Dilate, [228](#)
- svuDilateGeneric.c, [903](#)
 - DilateGeneric, [903](#)
- svuEqualizeHist
 - Equalize Histogram, [229](#)
- svuEqualizeHist.c, [904](#)
 - equalizeHist, [904](#)
- svuErode3x3
 - Erode 3x3, [230](#)
- svuErode3x3.c, [904](#)
 - Erode3x3, [905](#)
- svuErode5x5
 - Erode 5x5, [231](#)
- svuErode5x5.c, [905](#)
 - Erode5x5, [905](#)
- svuErode7x7
 - Erode 7x7, [232](#)
- svuErode7x7.c, [905](#)
 - Erode7x7, [906](#)
- svuExtAfStats
 - AF Stats, [233](#)
- svuExtAfStats.c, [906](#)
 - IRF_BASE, [907](#)
 - OCR_STOP_GO, [907](#)
 - OSR_SWI_HALT, [907](#)
 - SET_REG_WORD, [907](#)
 - SVU_IRR, [907](#)
 - SVU_OCR, [907](#)
 - SVU_OSR, [907](#)
 - SVU_PTR, [907](#)
- svuExtStatsSatPixelsU32
 - extStatsSatPixelsU32.h, [553](#)
 - svuExtStatsSatPixelsU32.c, [909](#)
- svuExtStatsSatPixelsU32.c, [908](#)
 - IRF_BASE, [908](#)
 - OCR_STOP_GO, [908](#)
 - OSR_SWI_HALT, [908](#)
 - SET_REG_WORD, [908](#)
 - SVU_IRR, [908](#)
 - SVU_OCR, [908](#)

- SVU_OSR, [908](#)
- SVU_PTR, [908](#)
- svuExtStatsSatPixelsU32, [909](#)
- svuFast9M2
 - Fast9M2, [234](#)
- svuFast9M2.c, [909](#)
 - adiff, [909](#)
 - fast9M2, [909](#)
 - fastBitFlag, [909](#)
 - fastExclude, [910](#)
 - sat8add, [910](#)
 - sat8sub, [910](#)
- svuFast9ScoreCv
 - Fast9ScoreCv, [235](#)
- svuFast9ScoreCv.c, [910](#)
 - adiff, [911](#)
 - bulkBuff, [912](#)
 - fastExcludePos, [911](#)
 - fastFlagBit, [911](#)
 - fastScore, [911](#)
 - minimumCalc, [911](#)
 - mvcvfast9ScoreCv, [911](#)
 - ones, [911](#)
 - rightrot, [912](#)
 - satUInt8add, [912](#)
 - satUInt8sub, [912](#)
 - vectorRotate, [912](#)
 - vectorShift, [912](#)
- svuGauss
 - Gauss Blur, [236](#)
- svuGauss.c, [912](#)
 - gauss, [912](#)
- svuGaussHx2
 - GaussHx2, [237](#)
- svuGaussHx2.c, [913](#)
 - mvcvGaussHx2, [913](#)
- svuGaussHx2_fp16
 - GaussHx2_fp16, [238](#)
- svuGaussHx2_fp16.c, [913](#)
 - GaussHx2_fp16, [914](#)
- svuGaussVx2
 - GaussVx2, [239](#)
- svuGaussVx2.c, [914](#)
 - mvcvGaussVx2, [914](#)
- svuGaussVx2_fp16
 - GaussVx2_fp16, [240](#)
- svuGaussVx2_fp16.c, [914](#)
 - GaussVx2_fp16, [915](#)
- svuGenChroma
 - Generate Chroma, [241](#)
- svuGenChroma.c, [915](#)
 - genChroma, [915](#)
- svuGenChromaSS
 - Generate Chroma with subsampling, [242](#)
- svuGenChromaSS.c, [915](#)
 - GenChromaSS, [916](#)
- svuGenDnsRef
 - Generate Reference for Luma Denoise, [243](#)
- svuGenDnsRef.c, [916](#)
 - genDnsRef, [916](#)
- svuGenDnsRefFp16
 - Generate Reference for Luma Denoise(fp16 input), [244](#)
- svuGenDnsRefFp16.c, [916](#)
 - genDnsRefFp16, [917](#)
- svuGenLuma
 - Luma Blur, [245](#)
- svuGenLuma.c, [917](#)
 - genLuma, [917](#)
- svuGenLumaU8Fp16
 - Generate Luma U8 to Fp16, [246](#)
- svuGenLumaU8Fp16.c, [917](#)
 - genLumaU8Fp16, [917](#)
- svuGenNoise
 - Random Noise, [294](#)
- svuGenNoiseFp16
 - Random Noise (high speed), [295](#)
- svuGreyDesat
 - greyDesat, [247](#)
- svuGreyDesat.c, [918](#)
 - greyDesat, [918](#)
- svuHammingDistance
 - hammingDistance, [248](#)
- svuHammingDistance.c, [918](#)
 - countBit, [918](#)
 - mvcvHammingDistance, [919](#)
- svuHarrisResponse
 - harrisResponse, [249](#)
- svuHarrisResponse.c, [919](#)
 - HARRIS_SW_RADIUS, [919](#)
 - HarrisResponse, [919](#)
- svuHistogram
 - Histogram, [250](#)
- svuHistogram.c, [920](#)
 - histogram, [920](#)
- svuHistogramStat
 - histogramStat, [251](#)
- svuHistogramStat.c, [920](#)
 - mvispHistogramStat, [921](#)
- svuHomography
 - Homography, [252](#)
- svuHomography.c, [921](#)

- getInPtrAbs, [921](#)
- svuIntegralImageSqSumF32M2
 - Integral Image Square Sum(f32), [253](#)
- svuIntegralImageSqSumF32M2.c, [922](#)
 - integralimage_sqsum_f32_M2, [922](#)
- svuIntegralImageSqSumU32M2
 - Integral Image Square Sum(U32), [254](#)
- svuIntegralImageSqSumU32M2.c, [922](#)
 - integralimage_sqsum_u32M2, [923](#)
- svuIntegralImageSumF32M2
 - Integral Image Sum(f32), [255](#)
- svuIntegralImageSumF32M2.c, [923](#)
 - integralimage_sum_f32M2, [924](#)
- svuIntegralImageSumU16U32
 - Integral Image Sum(U16toU32), [256](#)
- svuIntegralImageSumU16U32.c, [924](#)
 - integralImageSumU16U32, [924](#)
- svuIntegralImageSumU32M2
 - Integral Image Sum(U32), [257](#)
- svuIntegralImageSumU32M2.c, [926](#)
 - integralimage_sum_u32M2, [926](#)
- svuInterpolatePixelBilinear
 - interpolatePixelBilinear, [258](#)
- svuInterpolatePixelBilinear.c, [927](#)
 - mvcvInterpolatePixelBilinear, [927](#)
- svuLaplacian3x3
 - Laplacian 3x3, [259](#)
- svuLaplacian3x3.c, [927](#)
 - Laplacian3x3Implementation, [928](#)
- svuLaplacian5x5
 - Laplacian 5x5, [260](#)
- svuLaplacian5x5.c, [929](#)
 - sLaplacian5x5Implementation, [929](#)
- svuLaplacian5x5Fp16ToFp16
 - Laplacian 5x5 Fp16 To Fp16, [261](#)
- svuLaplacian5x5Fp16ToFp16.c, [929](#)
 - sLaplacian5x5Fp16ToFp16Implementation, [930](#)
- svuLaplacian7x7
 - Laplacian 7x7, [262](#)
- svuLaplacian7x7.c, [930](#)
 - sLaplacian7x7Implementation, [930](#)
- svuLaplacian7x7Fp16ToFp16
 - Laplacian 7x7 Fp16 To Fp16, [263](#)
- svuLaplacian7x7Fp16ToFp16.c, [930](#)
 - sLaplacian7x7Fp16ToFp16Implementation, [931](#)
- svuLocalMaxMin3x3_fp16
 - localMaxMin3x3_fp16, [264](#)
- svuLocalMaxMin3x3_fp16.c, [931](#)
 - mvcvLocalMaxMin3x3_fp16, [931](#)
- svuLocalTM
 - localTM, [265](#)
- svuLocalTM.c, [932](#)
 - localTM, [932](#)
- svuLowLvlCorr
 - Low Level Correction, [266](#)
- svuLowLvlCorr.c, [932](#)
 - SUB_SATURATE, [933](#)
- svuLumaBlur
 - Luma Blur, [269](#)
- svuLumaBlur.c, [933](#)
 - lumaBlur, [933](#)
- svuLut10to16
 - LUT 10 to 16, [270](#)
- svuLut10to16.c, [933](#)
 - LUT10to16, [934](#)
- svuLut10to8
 - LUT 10 to 8, [271](#)
- svuLut10to8.c, [935](#)
 - LUT10to8, [935](#)
- svuLut12to16
 - LUT 12 to 16, [272](#)
- svuLut12to16.c, [936](#)
 - LUT12to16, [936](#)
- svuLut12to8
 - LUT 12 to 8, [273](#)
- svuLut12to8.c, [936](#)
 - LUT12to8, [937](#)
- svuLut8to8
 - LUT 8 to 8, [274](#)
- svuLut8to8.c, [937](#)
 - LUT8to8, [937](#)
- svuLutP10BppU16inU8out
 - LUT 16 to 8, [275](#)
- svuLutP10BppU16inU8out.c, [938](#)
 - MAX_LUT_IDX, [938](#)
- svuMaxTest3x3_fp16
 - maxTest3x3_fp16, [276](#)
- svuMaxTest3x3_fp16.c, [938](#)
 - mvcvMaxTest3x3_fp16, [939](#)
- svuMeanStdDev
 - meanStdDev, [277](#)
- svuMeanStdDev.c, [939](#)
 - mvcvMeanstddev, [939](#)
- svuMinMaxPos
 - Min/Max Value Position, [278](#)
- svuMinMaxPos.c, [940](#)
 - minMaxPos, [940](#)
- svuMinMaxValue
 - Min/Max Value, [279](#)
- svuMinMaxValue.c, [941](#)

- minMaxKernel, [941](#)
- svuMinTest3x3_fp16
 - minTest3x3_fp16, [280](#)
- svuMinTest3x3_fp16.c, [941](#)
 - mvcvMinTest3x3_fp16, [942](#)
- svuMixMedian
 - Mix Median, [281](#)
- svuMixMedian.c, [942](#)
 - mixMedian, [942](#)
- svuMonoImbalance
 - monoImbalance.h, [579](#)
 - svuMonoImbalance.c, [943](#)
- svuMonoImbalance.c, [943](#)
 - mvcvMonoImbalance, [943](#)
 - svuMonoImbalance, [943](#)
- svuNegative
 - Negative, [283](#)
- svuNegative.c, [943](#)
 - negativeFilterImplementation, [944](#)
- svuNonMax3x3Fp32
 - nonMaxFp32, [284](#)
- svuNonMax3x3Fp32.c, [944](#)
 - mvcvNonMax3x3_fp32, [944](#)
- svuNonMax3x3U8
 - nonMax3x3U8, [285](#)
- svuNonMax3x3U8.c, [945](#)
 - mvcvNonMax3x3_u8, [945](#)
- svuPadBayer5
 - padBayer5.h, [581](#)
 - svuPadBayer5.c, [946](#)
- svuPadBayer5.c, [945](#)
 - padBayer5Reference, [946](#)
 - svuPadBayer5, [946](#)
- svuPadBayer5Frame
 - padBayer5Frame, [286](#)
- svuPadBayer5Frame.c, [946](#)
 - padBayer5Reference, [946](#)
- svuPadFunc
 - HPadInfoS, [390](#)
- svuPixelPacker10b
 - Pixel packer, [287](#)
- svuPixelPacker10b.c, [946](#)
 - pixelPacker10b, [947](#)
- svuPixelUnpacker
 - Pixel Unpacker, [288](#)
- svuPixelUnpacker.c, [947](#)
 - pixelUnpacker, [947](#)
- svuPixelUnpackerMipi10b
 - Pixel Unpacker Mipi 10b, [289](#)
- svuPixelUnpackerMipi10b.c, [947](#)
 - pixelUnpackerMipi10b, [948](#)
- svuPixelUnpackerWB
 - Pixel Unpacker WB, [290](#)
- svuPixelUnpackerWB.c, [948](#)
 - pixelUnpackerWB, [948](#)
- svuPositionKernel
 - Pixel Position, [291](#)
- svuPositionKernel.c, [949](#)
 - pixelPos, [949](#)
- svuPurpleFlare
 - purpleFlare, [292](#)
- svuPurpleFlare.c, [950](#)
 - MAX, [950](#)
 - MIN, [950](#)
 - purpleFlare, [950](#)
- svuPyrDown
 - Pyramid Downscale, [293](#)
- svuPyrDown.c, [950](#)
 - pyrdown, [951](#)
- svuRandNoise.c, [951](#)
 - getRandom, [951](#)
 - seeds, [952](#)
- svuRandNoiseFp16.c, [952](#)
 - genRand, [952](#)
 - mskBit, [953](#)
 - normValue, [953](#)
 - randNoiseFp16, [952](#)
 - sauOnesX16, [953](#)
 - seedlist, [953](#)
- svuRgbYuv444
 - Convert to YUV444, [201](#)
- svuSAD11x11
 - Sum of Absolute Differences 11x11, [296](#)
- svuSAD11x11.c, [953](#)
 - sumOfAbsDiff11x11, [954](#)
- svuSAD5x5
 - Sum of Absolute Differences 5x5, [297](#)
- svuSAD5x5.c, [954](#)
 - sumOfAbsDiff5x5, [954](#)
- svuSSD11x11
 - Sum of Squared Differences 11x11, [315](#)
- svuSSD11x11.c, [964](#)
 - sumOfSquaredDiff11x11, [964](#)
- svuSSD5x5
 - Sum of Squared Differences 5x5, [316](#)
- svuSSD5x5.c, [965](#)
 - sumOfSquaredDiff5x5, [965](#)
- svuSSD7x7U8ToU32
 - Sum of Squared Differences 7x7 (U8 to U32), [317](#)
- svuSSD7x7U8ToU32.c, [965](#)
 - sumOfSquaredDiff7x7U8ToU32, [966](#)

svuScale05BilinHV.c, [954](#)
 scale05BilinHV_U16ToU16, [955](#)
 scale05BilinHV_U8ToU8, [955](#)
 svuScale05BilinHV_Fp16U8
 Downscale by 2 (fp16/u8), [299](#)
 svuScale05BilinHV_Fp16U8.c, [955](#)
 scale05BilinHV_Fp16U8, [955](#)
 svuScale05BilinHVFP16
 Downscale by 2 (fp16/fp16), [300](#)
 svuScale05BilinHVFP16.c, [956](#)
 scale05BilinHVFP16, [956](#)
 svuScale05Lanc6HV.c, [956](#)
 scale05Lanc6HV, [957](#)
 vStep, [957](#)
 svuScale05Lanc7HV.c, [957](#)
 subs05sync7, [957](#)
 vStep, [957](#)
 svuScale2xBilinHV.c, [957](#)
 svuScale2xBilinHV_025_075_Fp16ToFp16
 Upscale by 2 with phases 0.25 and 0.75 fp16 to fp16, [304](#)
 svuScale2xBilinHV_025_075_Fp16ToFp16.c, [958](#)
 svuScale2xBilinHV_025_075_U16ToU16
 Upscale by 2 with phases 0.25 and 0.75 u16 to u16, [305](#)
 svuScale2xBilinHV_025_075_U16ToU16.c, [958](#)
 svuScale2xBilinHV_Fp16U8_phase025_075
 Upscale by 2 with phases 0.25 and 0.75 fp16 to u8, [306](#)
 svuScale2xBilinHV_Fp16U8_phase025_075.c, [959](#)
 svuScale2xBilinHV_U8ToU8_phase025_075
 Upscale by 2 with phases 0.25 and 0.75 u8 to u8, [307](#)
 svuScale2xBilinHV_U8ToU8_phase025_075.c, [960](#)
 svuScale2xLancH.c, [960](#)
 svuScale2xLancHV.c, [960](#)
 kern, [961](#)
 MARGIN, [961](#)
 upscale2xH, [961](#)
 upscale2xV, [961](#)
 svuScale2xLancV.c, [962](#)
 svuScaleBilinArb.c, [962](#)
 svuScharr_fp16
 scharr_fp16, [312](#)
 svuScharr_fp16.c, [962](#)
 mvcvScharr_fp16, [963](#)
 svuScl05BilinHV
 Downscale by 2, [298](#)
 svuScl05Lanc6
 Lanczos Downscale by 2 (6 taps), [301](#)
 svuScl05Lanc7
 Lanczos Downscale by 2 (7 taps), [302](#)
 svuScl2xBilinHV
 Upscale by 2, [303](#)
 svuScl2xLancH
 Lanczos Horizontal Upscale by 2, [308](#)
 svuScl2xLancHV
 Lanczos Upscale by 2, [309](#)
 svuScl2xLancV
 Lanczos Vertical Upscale by 2, [310](#)
 svuSclBilinArb
 Arbitrary Downscale, [311](#)
 svuSobel
 Sobel, [314](#)
 svuSobel.c, [963](#)
 sobel, [964](#)
 svuSsdPointLine7x7U8U32
 Sum of Squared Differences 7x7, [318](#)
 svuSsdPointLine7x7U8U32.c, [966](#)
 mvcvSsdPointLine7x7U8U32, [966](#)
 svuStack
 SippPipelineS, [458](#)
 svuStartBicubic
 startBicubic.h, [840](#)
 svuStartBicubic.c, [967](#)
 svuStartBicubic.c, [966](#)
 COMMAND_SIZEOF, [967](#)
 configureBicubicHWblock, [967](#)
 svuStartBicubic, [967](#)
 svuStatsAwbSatPixels
 statsAwbSatPixels.h, [840](#)
 svuStatsAwbSatPixels.c, [968](#)
 svuStatsAwbSatPixels.c, [967](#)
 CLAMPZ255, [967](#)
 statsAWBSatPixels, [968](#)
 svuStatsAwbSatPixels, [968](#)
 svuStatsAwbSatPixelsU32
 statsAwbSatPixelsU32.h, [841](#)
 svuStatsAwbSatPixelsU32.c, [968](#)
 svuStatsAwbSatPixelsU32.c, [968](#)
 statsAWBSatPixelsU32, [968](#)
 svuStatsAwbSatPixelsU32, [968](#)
 svuSubpixelFilter
 subpixelFilter.h, [842](#)
 svuSubpixelFilter.c, [969](#)
 svuSubpixelFilter.c, [968](#)
 mvcvSubPixelFilter_asm, [969](#)
 svuSubpixelFilter, [969](#)
 svuSyncMtx
 SippPipelineS, [458](#)

- svuSyncMtxAddr
 - SippPipelineS, [458](#)
- svuSyncMtxParity
 - SippPipelineS, [458](#)
- svuSyncSem
 - SippPipelineS, [459](#)
- svuThreshold
 - Threshold, [320](#)
- svuThreshold.c, [969](#)
 - thresholdKernel, [969](#)
- svuThresholdBinaryRange
 - Threshold Binary Range, [321](#)
- svuThresholdBinaryRange.c, [970](#)
 - thresholdBinaryRange, [970](#)
- svuThresholdBinaryU8
 - Threshold Binary U8, [322](#)
- svuThresholdBinaryU8.c, [970](#)
 - thresholdBinaryU8, [971](#)
- svuThresholdFilter
 - ThresholdFilter, [323](#)
- svuThresholdFilter.c, [971](#)
 - mvcvThresholdFilter, [971](#)
- svuUndistortBrown
 - Undistort, [324](#)
- svuUndistortBrown.c, [971](#)
 - getInPtrAbs, [972](#)
- svuWhiteBalanceBayerGBRG
 - White Balance Bayer GBRG, [325](#)
- svuWhiteBalanceBayerGBRG.c, [972](#)
 - whiteBalanceBayerGBRG, [972](#)
- svuWhiteBalanceRGB
 - White Balance RGB, [326](#)
- svuWhiteBalanceRGB.c, [973](#)
 - whiteBalanceRGB, [973](#)
- svuWinRegs
 - SippPipelineS, [459](#)
- svuXYgen
 - XY Generator, [327](#)
- svuXYgen.c, [974](#)
 - genXYlist, [974](#)
- svubitwiseAndMask
 - Bitwise And with mask, [132](#)
- svucvtColorNV21toRGB
 - NV21 to RGB conversion, [210](#)
- svucvtColorNV21toRGB.c, [894](#)
 - cvtColorNV21toRGBImplementation, [894](#)
- svudisp2depth
 - disp2depth.h, [548](#)
 - svudisp2depth.c, [904](#)
- svudisp2depth.c, [903](#)
 - dsp2depth_explic_vect, [904](#)
 - flip_dsp2depth_explic_vect, [904](#)
 - svudisp2depth, [904](#)
- svulowLvlCorrMultiplePlanes
 - Low Level Correction on Multiple Planes, [268](#)
- svusLaplacian3x3Fp16ToFp16
 - Laplacian 3x3 Fp16 To Fp16, [313](#)
- svusLaplacian3x3Fp16ToFp16.c, [963](#)
 - sLaplacian3x3Fp16ToFp16, [963](#)
- tCMDStartQu
 - SippPipelineS, [459](#)
- tCMDUpdateQu
 - SippPipelineS, [459](#)
- tCmxDmaQu
 - SippPipelineS, [459](#)
- tHeapMCB
 - SippPipelineS, [459](#)
- tMLPISStartCQCtrl, [477](#)
 - firstRunFilt, [478](#)
 - firstUseInProg, [478](#)
 - nFirstRunFilt, [478](#)
- tPartialCfg
 - DMAExtCfg, [374](#)
- tPublicPipe
 - tSippPipelineSuper, [481](#)
- tRTStats, [478](#)
 - DMAWaitCycles, [479](#)
 - HWWaitCycles, [479](#)
 - iterTime, [479](#)
 - ScheduleCycles, [479](#)
 - ShaveWaitCycles, [479](#)
- TRUE
 - PC/sippPlatformAbstractionLayer.c, [812](#)
 - sippTypes.h, [827](#)
- tSippFramework, [479](#)
 - bInit, [479](#)
 - dynIrqSipp0, [479](#)
 - dynIrqSipp1, [479](#)
 - dynIrqSipp2, [479](#)
 - pPipelines, [479](#)
- tSippMCB, [480](#)
 - numVPools, [480](#)
 - pVPoolListEnd, [480](#)
 - pVPoolListStart, [480](#)
 - pVirtPhysMap, [480](#)
 - pipeIdx, [480](#)
- tSippPhysicalPool, [480](#)
 - end, [481](#)
 - name, [481](#)
 - pHeap, [481](#)
 - pos, [481](#)
 - start, [481](#)

- tSippPipelineSuper, 481
 - bSVUOnly, 481
 - eState, 481
 - tPublicPipe, 481
 - uHWPipeID, 481
 - uPipeIdx, 482
- tSippVPhysMap
 - sippTypes.h, 829
- taskPl
 - DmaTaskList, 376
- thr
 - ChrDnsParam, 339
- thrBright
 - MonoImbalanceParam, 408
- thrDark
 - MonoImbalanceParam, 408
- threeDLut
 - ColCombParam, 343
- thresh
 - DbyrParam, 370
 - SigmaParam, 425
- Thresh_To_Binary
 - Threshold, 319
- Thresh_To_Binary_Inv
 - Threshold, 320
- Thresh_To_Zero
 - Threshold, 319
- Thresh_To_Zero_Inv
 - Threshold, 319
- Thresh_Trunc
 - Threshold, 320
- threshType
 - ThresholdParam, 477
- Threshold, 319
 - SHAVE_SYM_EXPORT, 320
 - svuThreshold, 320
 - Thresh_To_Binary, 319
 - Thresh_To_Binary_Inv, 320
 - Thresh_To_Zero, 319
 - Thresh_To_Zero_Inv, 319
 - Thresh_Trunc, 320
- threshold
 - Fast9M2Param, 383
 - Fast9ScoreCvParam, 383
 - ThresholdBinaryU8Param, 476
 - ThresholdFilterParam, 476
- Threshold Binary Range, 321
 - SHAVE_SYM_EXPORT, 321
 - svuThresholdBinaryRange, 321
- Threshold Binary U8, 322
 - SHAVE_SYM_EXPORT, 322
 - svuThresholdBinaryU8, 322
- threshold.h, 974
- threshold1
 - cannyEdgeDetectionParam, 336
- threshold2
 - cannyEdgeDetectionParam, 336
- thresholdBinaryRange
 - svuThresholdBinaryRange.c, 970
- thresholdBinaryRange.h, 975
- ThresholdBinaryRangeParam, 475
 - lowerValue, 475
 - upperValue, 475
- thresholdBinaryU8
 - svuThresholdBinaryU8.c, 971
- thresholdBinaryU8.h, 975
- ThresholdBinaryU8Param, 475
 - threshold, 476
- ThresholdFilter, 323
 - SHAVE_SYM_EXPORT, 323
 - svuThresholdFilter, 323
- thresholdFilter.h, 976
- ThresholdFilterParam, 476
 - posOffset, 476
 - threshold, 476
- thresholdKernel
 - svuThreshold.c, 969
- ThresholdParam, 476
 - threshType, 477
 - thresholdValue, 477
- thresholdValue
 - ThresholdParam, 477
- topLevelCmxDmaIrqHandler
 - sippCmxDmaIf.c, 623
 - sippInternal.h, 766
- totalMem
 - SippCmxBufferMapS, 431
- TripleConv3x3, 478
 - c1, 478
 - c2, 478
 - c3, 478
- tripleConv3x3.c, 976
 - createTripleConv3x3, 977
- tripleConv3x3.h, 977
 - createTripleConv3x3, 977
- true
 - sippBaseTypes.h, 616
- tsSippHeap, 482
 - assigned, 482
 - freeCounter, 482
 - freePtr, 482
 - sippHeap_end, 482

- sippHeap_size, [482](#)
- sippHeap_start, [482](#)
- txID
 - MipiTxLoopbackParam, [404](#)
- u16
 - sippBaseTypes.h, [616](#)
- u32
 - sippBaseTypes.h, [616](#)
- U32_MEMCPY
 - sippPlatform_ma2x5x.h, [810](#)
- U32_MEMSET
 - sippPlatform_ma2x5x.h, [810](#)
- u64
 - sippBaseTypes.h, [616](#)
- u8
 - sippBaseTypes.h, [616](#)
- uDummy
 - SIPP_PIPELINE_FINALISED_DATA, [430](#)
- uHWFeatures
 - SIPP_HW_SESSION, [428](#)
- uHWFilterIBufUsageMask
 - SippPipelineS, [459](#)
- uHWFilterOBufUsageMask
 - SippPipelineS, [459](#)
- uHWFilterUsageBitMask
 - SIPP_HW_SESSION, [428](#)
 - SippPipelineS, [459](#)
- uHWPendList
 - SIPP_ACCESS_SCHEDULER, [426](#)
- uHWPipeID
 - tSippPipelineSuper, [481](#)
- uHWWaitList
 - SIPP_ACCESS_SCHEDULER, [426](#)
- uInfo
 - SippGlobals, [443](#)
- UInt16
 - sippBaseTypes.h, [616](#)
- UInt32
 - sippBaseTypes.h, [616](#)
- UInt64
 - sippBaseTypes.h, [616](#)
- UInt8
 - sippBaseTypes.h, [617](#)
- UNDEFINED
 - AF_paxel_statistics, [331](#)
- UNUSED
 - sippPlatform_ma2x5x.h, [810](#)
- uNumCurrHwPipes
 - SIPP_HW_SESSION, [428](#)
- UPDATE_PEND_LISTS
 - sippAccessScheduler.c, [607](#)
- uPipeEventRdIndex
 - SIPP_ACCESS_SCHEDULER, [426](#)
- uPipeEventWrIndex
 - SIPP_ACCESS_SCHEDULER, [426](#)
- uPipeHWEventCount
 - SIPP_ACCESS_SCHEDULER, [426](#)
- uPipeIdx
 - SIPP_ACCESS_SCHEDULER_QU_ENTR-
Y, [427](#)
 - tSippPipelineSuper, [482](#)
- uRdIdx
 - SIPP_ACCESS_SCHEDULER_QU, [426](#)
- uSWPendList
 - SIPP_ACCESS_SCHEDULER, [426](#)
- uSize
 - SIPP_ACCESS_SCHEDULER_QU, [426](#)
- uWrIdx
 - SIPP_ACCESS_SCHEDULER_QU, [426](#)
- Undistort, [324](#)
 - SHAVE_SYM_EXPORT, [324](#)
 - svuUndistortBrown, [324](#)
- UndistortBParam, [483](#)
 - cx, [483](#)
 - cy, [483](#)
 - k1, [483](#)
 - k2, [483](#)
 - p1, [483](#)
 - p2, [484](#)
- undistortBrown.h, [977](#)
- unit
 - SippFilterS, [442](#)
- UnitInfo, [484](#)
 - hwFnInit, [484](#)
 - hwFnLoad, [484](#)
 - paramSz, [484](#)
- unitTestFinalReport
 - sippPlatform_ma2x5x.h, [810](#)
- upperValue
 - ThresholdBinaryRangeParam, [475](#)
- Upscale by 2, [303](#)
 - SHAVE_SYM_EXPORT, [303](#)
 - svuSc12xBilinHV, [303](#)
- Upscale by 2 with phases 0.25 and 0.75 fp16 to
fp16, [304](#)
 - SHAVE_SYM_EXPORT, [304](#)
 - svuScale2xBilinHV_025_075_Fp16ToFp16,
[304](#)
- Upscale by 2 with phases 0.25 and 0.75 fp16 to u8,
[306](#)
 - SHAVE_SYM_EXPORT, [306](#)
 - svuScale2xBilinHV_Fp16U8_phase025_075,

[306](#)
 Upscale by 2 with phases 0.25 and 0.75 u16 to u16,
[305](#)
 SHAVE_SYM_EXPORT, [305](#)
 svuScale2xBilinHV_025_075_U16ToU16,
[305](#)
 Upscale by 2 with phases 0.25 and 0.75 u8 to u8,
[307](#)
 SHAVE_SYM_EXPORT, [307](#)
 svuScale2xBilinHV_U8ToU8_phase025_075,
[307](#)
 upscale2xH
 svuScale2xLancHV.c, [961](#)
 upscale2xV
 svuScale2xLancHV.c, [961](#)
 useCmxRegMap
 SippPipelineS, [459](#)
 useIntBar
 SIPP_HW_SESSION, [428](#)
 useSyncRuntime
 SippPipelineS, [460](#)
 used_size
 SippHeapCB, [444](#)
 usedCount
 cmxRegUsage, [342](#)
 UsmParam, [484](#)
 cfg, [485](#)
 clip, [485](#)
 coef01, [485](#)
 coef23, [485](#)
 frmDim, [485](#)
 limit, [485](#)
 rgnStop01, [485](#)
 rgnStop23, [486](#)
 strength, [486](#)

 vPoolCMXDMADesc
 sippTypes.h, [831](#)
 vPoolFilterLineBuf
 sippTypes.h, [831](#)
 vPoolFilterLineBuf0
 sippTypes.h, [831](#)
 vPoolFilterLineBuf1
 sippTypes.h, [831](#)
 vPoolFilterLineBuf10
 sippTypes.h, [832](#)
 vPoolFilterLineBuf11
 sippTypes.h, [832](#)
 vPoolFilterLineBuf2
 sippTypes.h, [831](#)
 vPoolFilterLineBuf3
 sippTypes.h, [831](#)

 vPoolFilterLineBuf4
 sippTypes.h, [832](#)
 vPoolFilterLineBuf5
 sippTypes.h, [832](#)
 vPoolFilterLineBuf6
 sippTypes.h, [832](#)
 vPoolFilterLineBuf7
 sippTypes.h, [832](#)
 vPoolFilterLineBuf8
 sippTypes.h, [832](#)
 vPoolFilterLineBuf9
 sippTypes.h, [832](#)
 vPoolGeneral
 sippTypes.h, [831](#)
 vPoolLast
 sippTypes.h, [832](#)
 vPoolPipeStructs
 sippTypes.h, [831](#)
 vPoolSchedule
 sippTypes.h, [831](#)
 vPoolScheduleTemp
 sippTypes.h, [831](#)
 vActiveHeight
 MipiTxParam, [406](#)
 vBackPorch
 MipiTxParam, [406](#)
 VCS_PRINT_INT
 sippPlatform_ma2x5x.h, [810](#)
 vCompare
 MipiTxParam, [407](#)
 vFrontPorch
 MipiTxParam, [407](#)
 vStep
 svuScale05Lanc6HV.c, [957](#)
 svuScale05Lanc7HV.c, [957](#)
 vSyncEndOff
 MipiTxParam, [407](#)
 vSyncStartOff
 MipiTxParam, [407](#)
 vSyncWidth
 MipiTxParam, [407](#)
 value
 sSippCdmaQuEntryS, [468](#)
 sSippCMDQuEntryS, [469](#)
 vbp
 MipiRxParam, [403](#)
 vectorRotate
 svuFast9ScoreCv.c, [912](#)
 vectorShift
 svuFast9ScoreCv.c, [912](#)
 vertCoefs

- PolyFirParam, [413](#)
- vertD
 - PolyFirParam, [413](#)
- vertN
 - PolyFirParam, [413](#)
- vsync
 - MipiTxLoopbackParam, [404](#)
- wPtr
 - DmaTaskList, [377](#)
- Wait
 - Semaphore, [423](#)
- White Balance Bayer GBRG, [325](#)
 - SHAVE_SYM_EXPORT, [325](#)
 - svuWhiteBalanceBayerGBRG, [325](#)
- White Balance RGB, [326](#)
 - SHAVE_SYM_EXPORT, [326](#)
 - svuWhiteBalanceRGB, [326](#)
- whiteBalanceBayerGBRG
 - svuWhiteBalanceBayerGBRG.c, [972](#)
- whiteBalanceBayerGBRG.h, [978](#)
- WhiteBalanceBayerGBRGParam, [486](#)
 - awbCoef, [486](#)
 - clamp, [486](#)
- whiteBalanceRGB
 - svuWhiteBalanceRGB.c, [973](#)
- whiteBalanceRGB.h, [978](#)
- WhiteBalanceRGBParam, [486](#)
 - awbCoef, [487](#)
 - clamp, [487](#)
- width
 - StartBicubicParam, [470](#)
- winX
 - MipiRxParam, [403](#)
- winY
 - MipiRxParam, [404](#)
- wrapperSem.cpp, [979](#)
- wrapperSem.h, [979](#)
- x
 - InterpolatePixelBilinearParam, [390](#)
- X_AXIS_REFL
 - sippHwCommon_ma2x5x.h, [734](#)
- xCoeff
 - EdgeParam, [378](#)
- xOffset
 - YDnsRefFp16Param, [490](#)
- XPos
 - SippPixelChunkPos, [460](#)
- XY Generator, [327](#)
 - SHAVE_SYM_EXPORT, [327](#)
 - svuXYgen, [327](#)
- XY_AXIS_REFL
 - sippHwCommon_ma2x5x.h, [734](#)
- XYGenParam, [487](#)
 - coefMat, [488](#)
 - firstShave, [488](#)
 - lastShave, [488](#)
- xyGen.h, [979](#)
- y
 - InterpolatePixelBilinearParam, [390](#)
- yCoeff
 - EdgeParam, [378](#)
- yCoefs
 - GenChrParam, [384](#)
- YDnsParam, [488](#)
 - cfg, [489](#)
 - distCfg, [489](#)
 - distOffsets, [489](#)
 - f2, [489](#)
 - frmDim, [489](#)
 - fullFrmDim, [489](#)
 - gammaLut, [489](#)
 - gaussLut, [489](#)
- YDnsRefFp16Param, [489](#)
 - lutDist, [490](#)
 - lutGamma, [490](#)
 - shift, [490](#)
 - xOffset, [490](#)
- YDnsRefLut10bppParam, [490](#)
 - lut, [491](#)
- YDnsRefParam, [491](#)
 - lutDist, [491](#)
 - lutGamma, [491](#)
 - shift, [491](#)
- YPos
 - SippPixelChunkPos, [460](#)
- YUV to NV12 chroma conversion, [209](#)
- YUV to RGB conversion, [224](#)
 - SHAVE_SYM_EXPORT, [224](#)
 - svuCvtColorYUVToRGB, [224](#)
- YUV400 to YUV422 conversion, [189](#)
 - svuConvertYUV400ToYUV422, [189](#)
- YUV422 to RGB conversion, [223](#)
 - svuCvtColorYUV422ToRGB, [223](#)