- **EVE is a high performance vision engine that can accelerate imaging and vision functions.**
- **Multiple Kernels (accelerated Functions) for vision, Imaging and signal Processing domains are provided in this product**
- **High Level applications utilizing single/multiple kernels are also provided in this product**
- **List of kernels and applications available in this product are listed in Performance Summary Table**

Description

This product has three key components

Starterware – This component contains the APIs to program different control modules of EVE subsystem. Data sheet doesn't capture any information regarding this component

Accelerated Functions – It is set of accelerated functions utilizing EVE Vector Co-processor (VCOP) for different applications (vision and imaging). These functions expect input and output in EVE subsystem memory.

Apps – These are high-level applications working on the data in external memory and underneath utilizing starterware and accelerated functions. Example of such applications are resizing of an image, Harris corner detection etc.

Summary of performance

Target Platform Name: Vision28 Super ADAS Applications Processor (Vayu)
CPU Cores: EVE
OPP Table / Frequency: 500 MHz
DDR Frequency : 500 MHz
Tools Versions: Code Composer Studio version 5.4.0.00091, ARP32 Code Generation Tools 1.0.7
Dependent Component Versions: NA

Table 1. Configuration Table

COMPONENT CONFIGURATION DESCRIPTION	ID
Kernels for vision domain	EVE_SW_KERNELS_VISION
Kernels for imaging and signal processing domain	EVE_SW_KERNELS_IMGSIG
Applications on EVE	EVE_SW_APPS

Table 2. Performance Information for EVE_SW_KERNELS_VISION

Function Name	Compute Block Size		Additional Parameters	VCOP Cycles Per Element	Comments
vcop_grayscale_dilation	64	64	SE size = 3x3	1.1	Cycles per output pixel
vcop_grayscale_erosion	64	64	SE size = 3x3	1.1	Cycles per output pixel
vcop_grayscale_opening	64	64	SE size = 3x3	2.17	Cycles per output pixel
vcop_grayscale_closing	64	64	SE size = 3x3	2.17	Cycles per output pixel
vcop_grayscale_tophat	64	64	SE size = 3x3	2.23	Cycles per output pixel
vcop_grayscale_bothat	64	64	SE size = 3x3	2.23	Cycles per output pixel
vcop_grayscale_morp_gradient	64	64	SE size = 3x3	2.27	Cycles per output pixel
vcop_grayscale_dilation_rect	64	64	SE size = 3x3	0.32	Cycles per output pixel
vcop_grayscale_erosion_rect	64	64	SE size = 3x3	0.32	Cycles per output pixel
vcop_grayscale_opening_rect	64	64	SE size = 3x3	0.63	Cycles per output pixel
vcop_grayscale_closing_rect	64	64	SE size = 3x3	0.63	Cycles per output pixel
vcop_grayscale_tophat_rect	64	64	SE size = 3x3	0.7	Cycles per output pixel
vcop_grayscale_bothat_rect	64	64	SE size = 3x3	0.7	Cycles per output pixel
vcop_grayscale_morp_gradient_rect	64	64	SE size = 3x3	0.71	Cycles per output pixel
vcop_grayscale_dilation_cross	64	64	SE size = 3x3	0.35	Cycles per output pixel
vcop_grayscale_erosion_cross	64	64	SE size = 3x3	0.35	Cycles per output pixel
vcop_grayscale_opening_cross	64	64	SE size = 3x3	0.69	Cycles per output pixel
vcop_grayscale_closing_cross	64	64	SE size = 3x3	0.69	Cycles per output pixel
vcop_grayscale_tophat_cross	64	64	SE size = 3x3	0.76	Cycles per output pixel
vcop_grayscale_bothat_cross	64	64	SE size = 3x3	0.76	Cycles per output pixel
vcop_grayscale_morp_gradient_cross	64	64	SE size = 3x3	0.83	Cycles per output pixel
vcop_nonMaxSuppress_mxn_16s	64	32	m = 3, n = 3	0.74	Cycles per output pixel
vcop_nonMaxSuppress_mxn_16s	64	32	m = 5, n = 5	1.02	Cycles per output pixel
vcop_nonMaxSuppress_mxn_16s	64	32	m = 7, n = 7	1.3	Cycles per output pixel
vcop_nonMaxSuppress_mxn_16sbitPack	64	32	m = 3, n = 3	0.7734	Cycles per output pixel
vcop_nonMaxSuppress_mxn_16sbitPack	64	32	m = 5, n = 5	1.005	Cycles per output pixel
vcop_nonMaxSuppress_mxn_16sbitPack	64	32	m = 7, n = 7	1.18	Cycles per output pixel
vcop_nonMaxSuppress_mxn_32s	32	32	m = 3, n = 3	1.26	Cycles per output pixel
vcop_nonMaxSuppress_mxn_32s	32	32	m = 5, n = 5	1.88	Cycles per output pixel
vcop_nonMaxSuppress_mxn_32s	32	32	m = 7, n = 7	2.51	Cycles per output pixel
vcop_nonMaxSuppress_mxn_32s_bitPack	32	32	m = 3, n = 3	1.302	Cycles per output pixel
vcop_nonMaxSuppress_mxn_32s_bitPack	32	32	m = 5, n = 5	1.92	Cycles per output pixel
vcop_nonMaxSuppress_mxn_32s_bitPack	32	32	m = 7, n = 7	2.51	Cycles per output pixel
vcop_vertical_non_max_suppression	2048	1	No. of Corners	2.23	VCOP Cycles = (0.1875 + 2.13) * num_corners
vcop_horizontal_non_max_suppression	2048	1	No. of Corners	1.31	VCOP Cycles = (0.1835 + 1.13) * num_corners
vcop_vec_gradients_xy_and_magnitude	32	32		0.25	30 + 0.25*cycles per output pixel
vcop_gradients_xy_list	48	48		0.43	Cycles/per Input Pixel
vcop_gradients_xy_mag_lut_index_calulcation	48	48		1.304	Cycles/per Input Pixel
vcop_reciprocal_sqrt_lookup	48	48		0.148	Cycles/per Input Pixel
vcop_gradients_xy_unit_vecs	48	48		0.348	Cycles/per Input Pixel
vcop_orientation_binning	32	32	Number of bins = 6	0.31	cycles/pixel/bin

vcop_location_matrix	32	32	Number of bins = 6	0.11	cycles/pixel/bin
vcop_nxn_block_sum	64	64	n=4	0.12	cycles/input pixel
vcop_block_statistics	64	64	Cell Size = 8 x 8	0.33	
vcop_median_filter_mxn_u8 (Large Block Median)	121	71		0.344	Cycles per input pixel for single block
vcop_fast9	64	8		5.3	VCOP Cycles = 230 + 4.875 * in_w * in_h
vcop_fast9_score – SAD based	32	1	No. of Corners	17.5	VCOP Cycles = 145 + 12.9 * num_corners
vcop_fast9_score – Threshold baed	32	1	No. of Corners	32.5	VCOP Cycles = 172 + 27.125 * num_corners
vcop_multipoint_harrisScore_u16	32	1	7x7 Patch Window	38.69	Cyc/Corner
vcop_compute_rBrief	1	1		2150	cycles/point, it includes moments, wedge and descriptor calculation
vcop_hammingDistance	128	24	Two input strings of 128 descriptors of xsize = 24 bytes are processed	2.02	Cycles/byte of input
	160	32	Two input strings of 160 descriptors of xsize = 32 bytes are processed	0.22	
	64	48	Two input strings of 64 descriptors of xsize = 48 bytes are processed	1.59	
vcop_featureMatching	16	128	Descriptor size of 24 bytes	2.69	Cycles per descriptor pair
	16	128	Descriptor size of 32 bytes	2.18	
	16	64	Descriptor size of 48 bytes	5.76	
vcop_featureMatch_initialize	128	1	Distance arrays pMinDist0 & pMinDist1 have 128 entries	0.5625	Cycle/entry
vcop_findTwoBestMatches	128	16	Picking minimum from 128 hamming distances working with a vector of 16 at a time	0.20	Cycles/input
vcop_pickConfidentMatches	128	1	Picking confident matches from an array of 128 minDist measures	2	Cycle/output
vcop_extract_bit_field_u32	2048	1		0.15	
vcop_prune_big_list	1024	1		1.22	
vcop_harrisScore_7x7	32	32		9	VCOP Busy cycles/(number of output pixels)
vcop_harrisScore_u32_7x7	32	32	Window size 5x5	8.48	VCOP Busy cycles/(number of output pixels)
vcop_harrisScore_u32_7x7	32	32	Window size 3x3	7.96	VCOP Busy cycles/(number of output pixels)
vcop_harrisScore_u32_7x7	32	32	Window size 7x7	9	VCOP Busy cycles/(number of output pixels)
vcop_harrisScore_32_methodB	32	32	Window size 3x3	1.65	VCOP Busy cycles/(number of output pixels)

vcop_harrisScore_32_methodB	32	32	Window size 5x5	2.17	VCOP Busy cycles/(number of output pixels)
vcop_harrisScore_32_methodB	32	32	Window size 7x7	2.70	VCOP Busy cycles/(number of output pixels)
vcop_multiblock_gradient_xy	8	1	10x10 block	22.75	Cyc/Corner
vcop_multiblock_bilinear_interp_intlv_7x7_s16	8	1	8x7 Patch Window Per Corner Point	38.75	Cyc/Corner
vcop_multiblock_bilinear_interp_7x7_u8	8	1	8x7 Patch Window Per Corner Point	36.75	Cyc/Corner
vcop_calc_inverse_structure_tensor_2x2	8	1		39.5	Cyc/Corner
vcop_calc_determinant_tensor_matrix	8	1		21.25	Cyc/Corner
vcop_sad_error_measure_lk	16	1	Patch Size = 7x7	14.25	
vcop_weight_computation	8	1		9.25	Cycles per key point
vcop_weight_address_bilinear_interpolation	8	1		14.75	Cycles per key point
vcop_sum_grad_cross_inter_frame_diff_7x7	16	1	Patch Size = 7x7	20.625	Cycles per key point
vcop_tensor_matrix_7x7_s16_grad	16	1	Patch Size = 7x7	19.875	Cycles per key point
vcop_calc_new_lk_xy	16	1		15	Cycles per key point
vcop_initialize_glcm	16	16	numOffsets = 1	0.66	Cycle/output pixel / angle/histogram channel
vcop_glcm_compute_1c	64	64		2.21	Cycles/input pixel
vcop_glcm_compute_8c	64	64		0.47	Cycles/input pixel
vcop_accumulate_8c_glcm	16	16	numOffsets = 1	1.81	Cycles/output pixel
vcop_hough_for_lines	4000	1	# of edge-points = 4000 rhoMaxLength=800	0.9735	Cycles/edge-point/angle
vcop_hough_circle_compute_idx	48	48	List size = 1024	0.4511	Cycles/list element
vcop_hough_circle_init_hough_space	256	256	List size = 2048, Downscale = 2	0.0335	Cycles/Hough space pixel
vcop_hough_circle_vote_to_hough_space	256	256	List size = 2048, Downscale = 2	2.027	Cycles/list element
vcop_hough_for_circle_detect	256	256	List size = 2048, Downscale = 2	0.63	Cycle/Hough space pixel
vcop_bhattacharyaDistance	8	8		6.84	Alpha
vcop_canny_bin_indexing	48	48		0.73	Cycles/output pixel
vcop_canny_nms_max_cases	48	48		0.92	Cycles/output pixel
vcop_canny_nms_double_thresholding	48	48		0.79	Cycles/output pixel
vcop_doublethresholding	56	64		0.74	Alpha
vcop_gradient5x5PyramidKernel_8	16	16		1.02	Alpha
vcop_normalFlow	8	8		2.12	Alpha
vcop_vec_bin_image_dilate_cross	256	30		0.03	Cycles/output pixel
vcop_vec_bin_image_dilate_mask	256	30		0.08	Cycles/output pixel
vcop_vec_bin_image_dilate_square	256	30		0.03	Cycles/output pixel
vcop_vec_bin_image_erode_cross	256	30		0.04	Cycles/output pixel
vcop_vec_bin_image_erode_mask	256	30		0.08	Cycles/output pixel
vcop_vec_bin_image_erode_single_pixel	256	30		0.04	Alpha
vcop_vec_bin_image_erode_square	256	30		0.05	Cycles/output pixel
vcop_vec_bin_image_morph_diff	256	30		0.02	Cycles/output pixel
vcop_vec_update_ewr_mean_s16	32	1		2.94	Alpha
vcop_vec_update_ewr_variance_s16	32	1		4.12	Alpha
vcop_select_list_elements			List Size 4072	0.146	Cycles/point
vcop_vcop_vec_array_l1_distance	64	1		0.125	Alpha

					VCOP Cycles = 51 + 0.125*array length
vcop_census_8bits	32	16	3x3, horzStep=vertStep=1	1.35	Cycles/pix
			5x5, horzStep=vertStep=1	2.87	
			9x9, horzStep=vertStep=2	2.87	
			9x9, horzStep=vertStep=1	8.12	
vcop_census_16bits	32	16	3x3, horzStep=vertStep=1	2.03	Cycles/pix
			5x5, horzStep=vertStep=1	4.56	
			9x9, horzStep=vertStep=2	4.56	
			9x9, horzStep=vertStep=1	13.31	

***Alpha – Not Validated**

Table 3. Performance Information for EVE_SW_KERNELS_IMGSIG

Function Name	Compute Block Size		Additional Parameters	VCOP Cycles Per Element	Comments
vcop_bin_image_to_list	56	56		0.41	
vcop_blockAverage2x2	64	32		0.25	Cycles/output pixel
vcop_filter (non-seperable)	16	12	3x3 filter	1.34	cycles/output pixel
vcop_gauss5x5PyramidKernel_8	16	16		0.4	Cycles/output pixels
vcop_gauss5x5PyramidKernel_16	16	16		0.4	Cycles/output pixels
vcop_alpha_blend_yuv420nv12	64	64	Block is 64x64 YUV 420.	0.4	Cycles/pix
vcop_alpha_blend_yuv422i	64	64	Block is 64x64 YUV 422.	0.52	Cycles/pix
vcop_yuv_420nv12_to_422uyvy	64	64	i/p block is 64x64 YUV 420 NV12	0.15	Cycles/pix
vcop_yuv_422uyvy_to_420nv12	64	64	i/p block is 64x64 YUV 422 UYVY	0.2	Cycles/pix
vcop_contrast_stretching	32	32		0.31	Cycles/pix
vcop_decompand_piecewise_linear	52	52		0.58	Cycles/pix
vcop_black_clamp_c_balance	52	52		0.18	Cycles/pix
vcop_soft_isp_extract_r	48	48		0.05	Cycles/pix
vcop_stats_collector_dense	48	48		0.33	Cycles/pix
vcop_rccc_to_cccc	52	52		1.09	Cycles/pix
vcop_gbce_simple	48	48		0.29	Cycles/pix
vcop_gbce_interp	48	48		0.94	Cycles/pix
vcop_intensity_scaling	64	64		0.14	cycles/pixel
Vcop_integral_image	32	64		0.46	Alpha 28 + 5/16 cycles per pixel
vcop_median3x3	64	62		1.2	Alpha, cycles/output pixel
vcop_median_filter_col	30	30		0.16	Alpha
vcop_median_filter_row	72	72		0.3	Alpha
vcop_BayerCFA_HorzUpsample	64	60		6.02	Alpha
vcop_BayerCFA_interpolate	32	32		0.44	Alpha
vcop_rgb_to_yuv	32	1		2.82	Alpha
vcop_dct8x8col_chen	16	16		0.82	Alpha
vcop_dct8x8col_odd_even	16	16		0.88	Alpha
vcop_dct8x8row_chen	16	16		0.82	Alpha
vcop_dct8x8row_odd_even	16	16		0.88	Alpha
vcop_HorzUpsample	72	72		4.46	Alpha
vcop_matrix_mul	8	5		4.6	Alpha
vcop_Filter_vertical_resampling	72	72		0.82	Alpha
vcop_rgb16bitPack	16	1		11.26	Alpha

vcop_rgb16bitUnPack	16	1		8.76	Alpha
vcop_rotate	68	64		0.26	Alpha
vcop_SAD	32	16		5.16	Alpha
vcop_transparentBlt	16	16		0.34	Alpha
YCbCr422Deinterleave	128	1		1.28	Alpha
vcop_YCbCr444Deinterleave	144	1		1.2	Alpha
YCbCr444Downsample422	144	1		2.18	Alpha
vcop_yuv_scalar	128	64		0.87	Cycles/output pixels
vcop_sobelXY_3x3_separable_uchar	64	64		0.403	Cycles/output pixels
vcop_sobelX_3x3_separable_uchar	64	64		0.3	Cycles/output pixels
vcop_sobelY_3x3_separable_uchar	64	64		0.27	Cycles/output pixels
vcop_sobelXy_3x3_L1_thresholding	64	64		0.26	Cycles/output pixels
vcop_sobelXy_3x3_L1_thresholding_binPack	64	64		0.33	Cycles/output pixels
vcop_binary_masking	64	64		0.39	Cycles/output pixels

***Alpha – Not Validated**

Table 4. Performance Information for EVE_SW_APPS

Applet	Frame Size and Configuration		VCOP Cycles Per Element	Code Size (Bytes)	Comments
YUV Padding	640x360	Padding of 64 on all directions	0.75	518	Cycles considering total amount of pixels being padded
Median Filter	640x480	Block size = 121x71, StepSize of 35 in both x and y.	2.22	2655	Cycle per input pixel
	640x482	Block size = 3x3, StepSize of 1 in both x and y.	1.32		
	640x480	Block size = 3x3, StepSize of 1 in both x and y.	1.42		
Binary Morphology	240 x 1080	Custom SE size = 3x3, Dilation	0.1	15420	Uses generic mask kernels
		Custom SE size = 3x3, Erosion	0.1		
		Custom SE size = 3x3, Opening	0.18		
		Custom SE size = 3x3, Closing	0.18		
		Custom SE size = 3x3, Top Hat	0.2		
		Custom SE size = 3x3, Bot Hat	0.2		
		Custom SE size = 3x3, Gradient	0.19		
		Rect SE size = 3x3, Dilation	0.06		Uses square mask kernels
		Rect SE size = 3x3, Erosion	0.06		
		Rect SE size = 3x3, Opening	0.09		
		Rect SE size = 3x3, Closing	0.09		
		Rect SE size = 3x3, Top Hat	0.11		
		Rect SE size = 3x3, Bot Hat	0.10		
		Rect SE size = 3x3, Gradient	0.10		
		Cross SE size = 3x3, Dilation	0.06		Uses cross mask kernels
		Cross SE size = 3x3, Erosion	0.06		
		Cross SE size = 3x3, Opening	0.08		
		Cross SE size = 3x3, Closing	0.07		
		Cross SE size = 3x3, Top Hat	0.09		
		Cross SE size = 3x3, Bot Hat	0.09		
		Cross SE size = 3x3, Gradient	0.09		
Grayscale Morphology	768 x 512	Custom SE size = 3x3, Dilation	1.23	15420	Uses generic mask kernels
		Custom SE size = 3x3, Erosion	1.23		
		Custom SE size = 3x3, Opening	2.70		
		Custom SE size = 3x3, Closing	2.70		
		Custom SE size = 3x3, Top Hat	2.77		

		Custom SE size = 3x3, Bot Hat	2.77		
		Custom SE size = 3x3, Gradient	2.44		
		Rect SE size = 3x3, Dilation	0.69		Uses rect mask kernels
		Rect SE size = 3x3, Erosion	0.69		
		Rect SE size = 3x3, Opening	0.88		
		Rect SE size = 3x3, Closing	0.88		
		Rect SE size = 3x3, Top Hat	0.95		
		Rect SE size = 3x3, Bot Hat	0.95		
		Rect SE size = 3x3, Gradient	0.96		
		Cross SE size = 3x3, Dilation	0.69		Uses cross mask kernels
		Cross SE size = 3x3, Erosion	0.69		
		Cross SE size = 3x3, Opening	0.91		
		Cross SE size = 3x3, Closing	0.91		
		Cross SE size = 3x3, Top Hat	0.98		
		Cross SE size = 3x3, Bot Hat	0.98		
		Cross SE size = 3x3, Gradient	0.98		
Image Pyramid	768x368	Number of pyramid levels = 1, 2x2 average filter	0.39	3936	Cycles are in unit of base resolution pixels. Create time 3.46 MHz
		Number of pyramid levels = 1, 5x5 Gaussian filter	0.68		Cycles are in unit of base resolution pixels. Create time 4.24 MHz
		Number of pyramid levels = 3, 2x2 average filter	0.43		Cycles are in unit of base resolution pixels. Create time 6.5 MHz
		Number of pyramid levels = 3, 5x5 Gaussian filter	2.01		Cycles are in unit of base resolution pixels. Create time 5.5 MHz
Separable filter for YUV 4:2:0	640x360	3x3 Gaussian filter, contrast stretch = OFF	1.2	2440	Cycles in unit of ouput pixels
		3x3 Gaussian filter, contrast stretch = ON	1.79		
Integral Image	720x480		1.14	2292	Cycles are per input pixel. Create time 2.06 MHz
Harris Corner Detection (16 bit)	640 x 480		16.88	922	Cycles are per corner point.
Harris Corner Detection (32 bit)	288 x 224	Num Corners Detected = 579	18.55	14852	Cycles are per pixel. Incurs one time create graph of 4.02 MHz

Harris Corner Detection (32 bit)	320x240	Harris Score Window : 7x7 NMS Window Size : 7x7 Harris Score Method : A Output Format : List	19.22	14852	Cycles are per pixel. Incurs one time create graph of 4.02 MHz
		Harris Score Window : 7x7 NMS Window Size : 7x7 Harris Score Method : B Output Format : List	8.57	14852	Cycles are per pixel. Incurs one time create graph of 4.02 MHz
		Harris Score Window : 7x7 NMS Window Size : 7x7 Harris Score Method : A Output Format : : Bin Pack	17.79	14852	Cycles are per pixel. Incurs one time create graph of 4.02 MHz
		Harris Score Window : 7x7 NMS Window Size : 7x7 Harris Score Method : B Output Format : Bin Pack	7.24	14852	Cycles are per pixel. Incurs one time create graph of 4.02 MHz
Harris Best Feature To Front		bestN = 1024, Num Levels = 3	157.68	4748	Cycles are per corner point. Incurs one time create graph of 0.3 MHz
FAST9 corner detect	358x358	Number of levels = 3	8.28	4228	Cycles are per corner point. Incurs one time create graph of 0.27 MHz
FAST9 Best Feature to Front	996 x 1	BestN = 100, 4-way NMS	181.8	6703	Cycles are per input corner point. Incurs one time create graph of 0.8 MHz
		BestN = 100, 8-way NMS	340.9		
Block Statistics (Min, Max,Mean,Variance)	640 x 480	Statistics Block size = 8x8	0.57	2612	Cycle per input pixel
Gray Level Co-occurrence Matrix (GLCM)	320x240	Number of Angle = 1, Number of bins = 8	2.18	18896	Cycles in unit of input pixels
Hough for Lines	640x480	ListSize 65536 RhoMaxLength 800 Theta 0 to 180 degrees	1.17	2484	Cycles are per edge pixel/per theta
Pyramid LK Tracker With Error Measure Compute	1226x370	Number of key points = 649, Number of Levels =4, Search Range = 12	1274	9405	Cycles are per corner point. Incurs one time create graph of 0.64 MHz
		Number of key points = 649, Number of Levels =4, Search Range = 18	1619		
RBRIEF	96x48	NumFeatures = 3	16788	4326	Cycles are per corner point. Incurs one time create graph of 0.15 MHz
Remap Luma : Bilinear Chroma: Nearest Neighbor	640x80	SRC FORMAT / DST FORMAT / MERGE		8921	cycles/pixel. Numbers in Bracket are with bounding box based approach. Numbers without bracket are for tile based approach. Create time additional 0.24 MHZ
		U8 / U8 / NO	5.14 (4.69)		
		420 / 422 ILE / NO	6.69 (5.24)		
		420 / 422 ILE / YES	8.59 (5.79)		
		422 ILE / 420 / NO	8.99 (8.2)		
		422 ILE / 420 / YES	11.22 (8.76)		
		420 / 420 / NO	6.69 (4.99)		
		420 / 420 / YES	8.48 (5.57)		
		422 ILE / 422 ILE / NO	8.88 (7.8)		

		422 ILE	422 ILE	YES	11.05 (8.47)		
Remap Luma : Bilinear Chroma: Bilinear	640x80	SRC FORMAT	DST FORMAT	MERGE		8921	cycles/pixel. Numbers in Bracket are with bounding box based approach. Numbers without bracket are for tile based approach. Create time additional 0.24 MHZ
		420	420	NO	7.48 (6.1)		
		422 IBE	422 IBE	NO	11.03 (8.77)		
2D NMS	642x482	Window width =3 Window Height =3			1.31	3360	Cycles are output pixel
Census Transform 8-bits	320x240	9x7 hStep= vStep= 1			6.65	7780	Cycles per output pixels
		9x9 hStep= vStep= 2			3.02		
Census Transform 16-bits	320x240	9x7 hStep= vStep= 1			10.6		
		9x9 hStep= vStep= 2			4.6		
Disparity using hamming dist.	640x480	64 disparities, step=2 -> 32 disparities calculated			50.9	16652	Cycles per output pixels
		64 disparities, step=1 -> 64 disparities calculated			104.4		
		128 disparities, step=2 -> 64 disparities calculated			104.4		
		128 disparities, step=1 -> 128 disparities calculated			215.4		
Feature Matching	1024x1024	1024 descriptor list to 1024 descriptor list for 32 byte descriptors.			2.38	9864	Cycles per input descriptor pair
	1280x1280	1280 descriptor list to 1280 descriptor list for 32 byte descriptors.			2.34		
	800x800	800 descriptor list to 800 descriptor list for 24 byte descriptors.			2.88		
	768x768	768 descriptor list to 768 descriptor list for 48 byte descriptors.			5.97		
	128x128	128 descriptor list to 128 descriptor list for 512 byte descriptors.			49.47		
Hough for Circle	640x480	For 1 radius			0.73	-	Total MHz for 1 frame
	640x480	For 8 radius			3.9		
Yuv scalar	640x360 to 586x 330	scaleRatioQ12 = 4466			1.515	-	Cycles per output pixel
Edge Detector	640x480	Using Sobel with L1 Norm, Output format : byte edgemap			0.936	8524	Cycles per pixel
Edge Detector	640x480	Using Sobel with L1 Norm, Output format : Bin Pack			1.002	8524	Cycles per pixel

Edge Detector	640x480	Using Canny with L1 Norm	3.95	8524	Cycles per pixel
bin_image_to_list	640x480	enableMasking = 1 enableListSuppression = 0	1.04	4276	Cycles per pixel
bin_image_to_list	640x480	enableMasking = 0 enableListSuppression = 0	0.66	4276	Cycles per pixel
bin_image_to_list	640x480	enableMasking = 1 enableListSuppression = 1	1.37	4276	Cycles per pixel

Demo Algorithms – These are high-level algorithms built using multiple apps as per the data flow of the algorithm pursued. Examples of such algorithms are pyramid LK tracker based Sparse Optical Flow (SOF), ORB etc. These are only for demonstration purpose and not intended to be used in production

Table 5. Performance Information for Demo algorithms

Function Name	Frame Size		MHz/Frame	Comments
Sparse Optical Flow (with Harris Corner)	1280x720	5 Pyramid Levels, keyPointDetectInterva =2, Search Range =18,	11MHz	Incurs one time create graph of 30.5 MHz
Sparse Optical Flow (with Harris Corner)	1280x720	5 Pyramid Levels, keyPointDetectInterva =0, Search Range =18,	23.64MHz	Incurs one time create graph of 30.5 MHz
Sparse Optical Flow (with Fast9 corners)	800 x 400	4 Pyramid Levels, 1024 feature points, Search Range = 18	8.86	Fast9 Corner: 2.67 Mega cycles – 8.359 Cyc/Pixel (This includes best feature to front and Q format conversion) Image Pyramid Applet : 0.87 Mega cycles - 4.129 Cyc/Output Pixel Pyramid LK Tracker Applet : 5.31 Mega cycles - 1296.991 Cyc/Corner Point
ORB	400 x 400	Number of levels = 3 Number of targeted features = 500	3.11	Incurs one time create graph of 4.73 MHz
Feature plane computation for Pedestrian detection using HOG	1280x720 (ROI = 1280x144)	contrast stretching, 21 scales, Number of planes =10, Number of angle bins = 6 Number of scales per octave = 6 Complete Cell sum (enableCellSum=1)	16.93	Incurs one time create of 78.672 MHz
	1280x720 (ROI = 1280x144)	contrast stretching, 21 scales, Number of planes =10, Number of angle bins = 6 Number of scales per octave = 6 Partial Cell sum (enableCellSum=0)	14.29	Incurs one time create of 78.672 MHz
	1280x720 (ROI = 1280x180)	contrast stretching, 21 scales, Number of planes =10, Number of angle bins = 6 Number of scales per octave = 6 Complete Cell sum (enableCellSum=1)	18.42	Incurs one time create of 78.672 MHz
	1280x720 (ROI = 1280x180)	contrast stretching, 21 scales, Number of planes =10, Number of angle bins = 6 Number of scales per octave = 6 Partial Cell sum (enableCellSum=0)	15.80	Incurs one time create of 78.672 MHz

	1280x720 (ROI = 1280x216)	contrast stretching, 21 scales, Number of planes =10, Number of angle bins = 6 Number of scales per octave = 6 Complete Cell sum (enableCellSum=1)	20.34	Incurs one time create of 78.672 MHz
	1280x720 (ROI = 1280x216)	contrast stretching, 21 scales, Number of planes =10, Number of angle bins = 6 Number of scales per octave = 6 Partial Cell sum (enableCellSum=0)	17.69	Incurs one time create of 78.672 MHz
Software Image Signal Processing	1280x724	Decompanding - ON, GBCE simple method, Statistics Collection – OFF, Extract R - ON	2.11	Incurs one time create graph of 0.25 MHz

notes

- Total data memory for N non pre-emptive instances = Constants + Scratch + N * (Instance + I/O buffers + Stack)
- Kernels having alpha in their "comments" field has gone through very limited verification. Also the performance number reported might be un-optimal/wrong because of improper block size being used during benchmarking

references

- ADAS Superset 28 Automotive Vision Applications Processor Silicon Revision 1.0 Embedded Vision Engine (literature number SPRUHK5)
- ADAS Superset 28 ADAS Applications Processor Silicon Revision 1.0 (literature number SPRUHK5A)
- Embedded Vector Engine (EVE) Programmer's Guide (literature number SPRUHC1C)
- VisionSurround28 Super/High/Mid Vision28 Super/High/Mid ADAS Applications Processor (SPRS884D)

glossary

Constants — Elements that go into .const memory section

Scratch — Memory space that can be reused across different instances of the algorithm

Shared — Sum of Constants and Scratch

Instance — Persistent-memory that contains persistent information - allocated for each instance of the algorithm

acronyms

EVE	Embedded Vision/Vector Engine

PRODUCT PREVIEW

VCOP	Vector Co-Processor
VLIB	Vision Library
IMGSIGLIB	Image & Signal Processing Library
ARP32	32-bit Application Specific RISC Processor
DMEM	Data Memory in EVE Subsystem
WBUF	DMEM WBUF Work Buffer in EVE Subsystem
IBUF	Image Buffer in EVE Subsystem
EDMA	Enhanced Direct Memory Access

IMPORTANT NOTICE

PRODUCT PREVIEW

Products		**Applications**	
Audio	www.ti.com/audio	Automotive & Transportation	www.ti.com/automotive
Amplifiers	amplifier.ti.com	Communications & Telecom	www.ti.com/communications
Data Converters	dataconverter.ti.com	Computers & Peripherals	www.ti.com/computers
DLP® Products	www.dlp.com	Consumer Electronics	www.ti.com/consumer-apps
DSP	dsp.ti.com	Energy and Lighting	www.ti.com/energyapps
Clocks and Timers	www.ti.com/clocks	Industrial	www.ti.com/industrial
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Security	www.ti.com/security
Power Mgmt	power.ti.com	Space, Avionics & Defense	www.ti.com/space-avionics-defense
Microcontrollers	microcontroller.ti.com	Video & Imaging	www.ti.com/video
RFID	www.ti-rfid.com		
OMAP Applications Processors	www.ti.com/omap	**TI E2E Community**	e2e.ti.com
Wireless Connectivity	www.ti.com/wirelessconnectivity		

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265