國立清華大學 電機工程系
112 學年度第二學期

Advanced SOC Design Laboratory

Final project report

Team 7

學校系所：清大電子所、成大電機所

學號姓名：110063553 張傑閔

1122029s 蔡宗頴

指導教授：賴瑾 教授 (Prof. Jiin Lai)

中 華 民 國 一 一 三 年 六 月

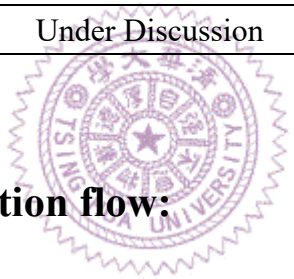
Table of Contents

1. Status

- **We haven't finished our application accelerator, because our target is the HDR ISP, which contains about 18 individual functional modules.**
- **We listed the status (for final project, including Lab 5 and Lab 6) in the table below:**

Identify algorithm C-source code and run Catapult C-sim - 1w		
Tasks	**Person In Charge**	**Status**
Identify test dataset	蔡宗穎	Done. (With the help from 傑閔)
Run Catapult C-sim (pre-HLS verification)	張傑閔	Done.
Kernel HLS implementation, Host implementation - 3w		
Tasks	**Person In Charge**	**Status**
UNPACK	蔡宗穎 (proposal)	Done.
DEPWL		Done.
DPC		Done.
LSC		Done.
BLC		Done by 傑閔.
RNS(N/A)		N/A.
WBGAIN		Done by 傑閔.
DEMOASIC		Done.
CCM		Done.
LTM	張傑閔 (proposal)	N/A.
RGBGAMMA		Done.
RGBYUV		Done.
YGAMMA		Done.
CONTRAST		Done.
SHARPEN		Working on it.
CNS		Done by 宗穎.
SATURATION		Done by 宗穎.

YUV2RGB		Done by 宗穎.
HDR ISP Integration in Catapult/FSIC		
Tasks	**Person In Charge**	**Status**
HDR ISP HLS-C Top Integration & Catapult C/RTL co-sim	Under Discussion	Working on it.
HDR ISP Integration into FSIC	Under Discussion	Working on it.
Individual Kernel FPGA validation/integration test - 2w		
Tasks	**Person In Charge**	**Status**
FSIC Simulation	張傑閔	Stuck.
FSIC Validation	蔡宗穎	Stuck.
Synopsys Flow – 1w		
Tasks	**Person In Charge**	**Status**
Front-end (Logic Synthesis - DFT)	Under Discussion	Stuck.
Back-end (APR - Signoff)	Under Discussion	Stuck.



- **The implementation flow:**
  1. **Transform the original algorithm C codes into classes (transformed algorithm C codes).**
  2. **Understand the algorithm of a certain module, design the micro-architecture of that module, and implement it with HLS C codes.**
  3. **Write one testbench for verifying the HLS module with Makefile.**
  4. **Verifying the HLS module with Catapult C-**

simulation.

5. **Synthesize the HLS C codes with directives.**
6. **Run C/RTL co-simulation for verifying the synthesized modules. (See UNPACK module below to understand the flow from step 2 to step 6.)**
7. **Repeat step 2 to step 6 for 17 times until all modules are finished.**
8. **Integrating all modules into the top level HDR ISP HLS C codes, run step3 to step 6 for this module for top-level verification.**
9. **Integrating the synthesized RTL codes into FSIC.**
10. **FSIC simulation. (Lab 6)**
11. **FSIC validation. (Lab 6)**
12. **Synopsys Flow. (Lab 6)**

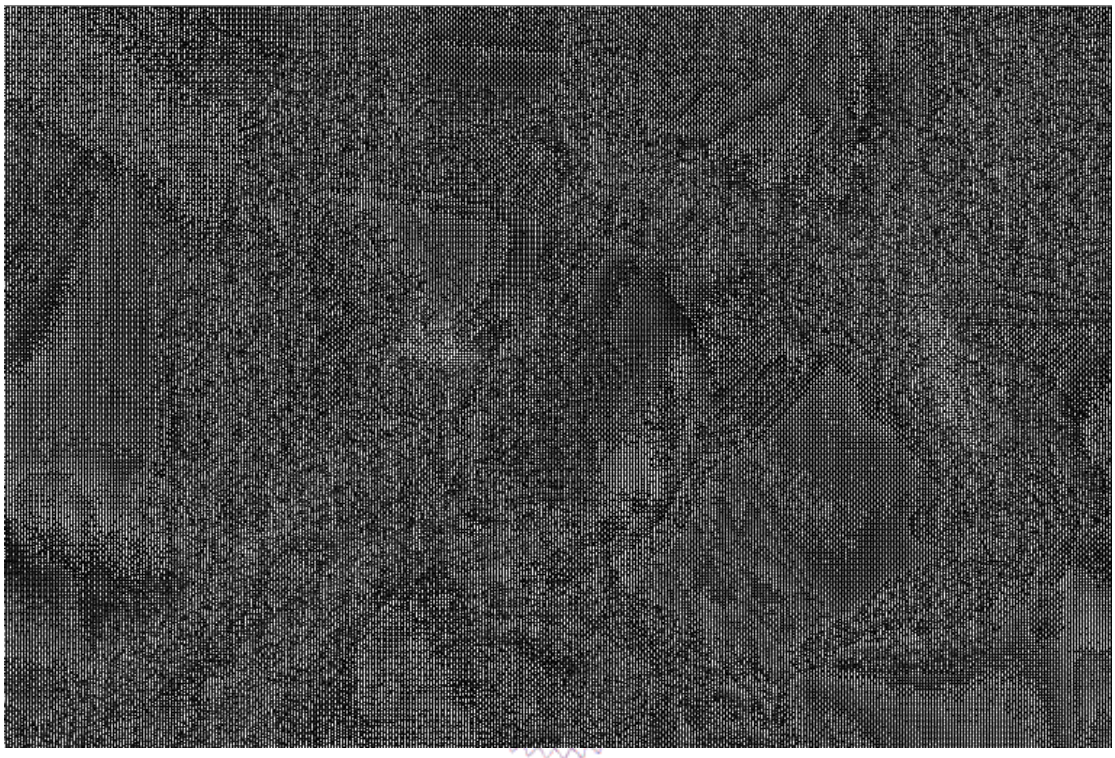
- **The difficulties we faced:**
  1. **The original algorithm C codes were written in separate .cpp files. To simplify the deisgn of HLS C codes, we need to transform the original algorithm C codes into several single files that contain one**

2. **class to represent one module. It takes us one more week.**
3. **The difficulties for writing various testbenches for single module verification, which compares the original algorithm C codes (.cpp) and the transformed algorithm C codes (.h), and the comparison between the transformed algorithm and the HLS C codes. This takes us another one week to understand the memory architecture in the original algorithm, and how to extract the correct values for comparison. This was done by testing the unpack module.**
4. **Since the original C codes were written in algorithm style, some unexpected difficulties would be faced. For example, the pixel used for loop iteration of DPC module (using the last point in the 5*5 region) is different from DPC algorithm (using central point), and the generation of zero pads for the output image is hard.**

2. HDR-ISP module introduction

− Original raw image:

- RAW domain image:



− Module: Unpack

- Function: Transform the data stored in MIPI Raw format into data with uint16 type.
- Transformed algorithm C codes:

-

```
#ifndef HDRISP_UNPACK_H_
#define HDRISP_UNPACK_H_

#include "/home/course/ee525201/final_project/ISP1/cmodel/inc/srcs/include/modules/modules.h"
#define MOD_NAME "unpack"

template <int Width, int Height>
class unpack_algorithm
{
public:
  unpack_algorithm(){}

  static void UnpackRaw10ToRaw16(uint8_t *raw,
                                 uint16_t *unpack_raw16,
                int width,
                int height)
  {
      uint8_t *raw10_packed_in = raw;
      uint16_t *raw16_unpacked_out = unpack_raw16;
      //int pixel_idx = 0;

      for (int p1 = 0, p2 = 0; p2 < width * height; p1 += 5, p2 += 4)
      {
          raw10_packed_in = raw + p1;
          raw16_unpacked_out = unpack_raw16 + p2;
          raw16_unpacked_out[0] = ((uint16_t)raw10_packed_in[0] << 2) | (((uint16_t)raw10_packed_in[5] >> 6) & 0x03);
          raw16_unpacked_out[1] = ((uint16_t)raw10_packed_in[1] << 2) | (((uint16_t)raw10_packed_in[5] >> 4) & 0x03);
          raw16_unpacked_out[2] = ((uint16_t)raw10_packed_in[2] << 2) | (((uint16_t)raw10_packed_in[5] >> 2) & 0x03);
          raw16_unpacked_out[3] = ((uint16_t)raw10_packed_in[3] << 2) | (((uint16_t)raw10_packed_in[5] >> 0) & 0x03);
      }
  }
  /**
   * @brief  mipi raw12 to raw16
   * [p1 11:4][p2 11:4][(p1 3:0)(p1 3:0)]
   * @param raw
   * @param unpack_raw16
   * @param width
   * @param height
   */
  static void UnpackRaw12ToRaw16(uint8_t *raw,
                                 uint16_t *unpack_raw16,
                int width,
                int height)
  {
      uint8_t *raw12_packed_in = raw;
      uint16_t *raw16_unpacked_out = unpack_raw16;

      for (int p1 = 0, p2 = 0; p2 < width * height; p1 += 3, p2 += 2)
      {
          raw12_packed_in = raw + p1;
          raw16_unpacked_out = unpack_raw16 + p2;
          raw16_unpacked_out[0] = ((uint16_t)raw12_packed_in[0] << 4) | (((raw12_packed_in[2] >> 4) & 0x0F);
```

- HLS C codes:

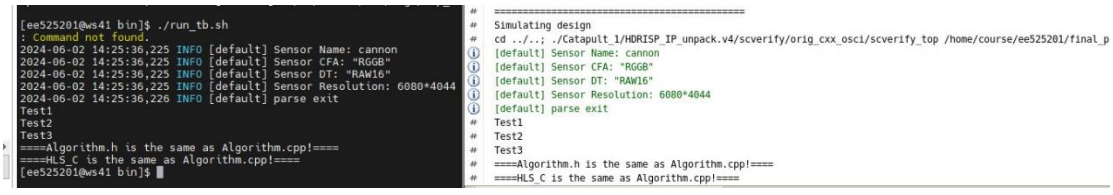
```
class unpack
{
private:
public:
unpack() {}

#pragma hls_design interface
  void CCS_BLOCK(run)(int          &mode,
          stream_u8    &dat_in,
                       maxWType    &widthIn,
                       maxHType    &heightIn,
                       stream_u16  &dat_out)
  {
  uint16 raw16_in = 0;
  //uint16 raw16_out = 0;

  //uint16 raw16_out = 0;
  uint8 raw16_in0;
  uint8 raw16_in1;
  #pragma hls_pipeline_init_interval 2
  RAW16:for(int i = 0;;i=i+1){
    if(dat_in.available(2)){

      raw16_in0 = dat_in.read();
      raw16_in1 = dat_in.read();
      raw16_in = ((uint16)raw16_in1 << 8) | ((uint16)raw16_in0 & 0xff);
      /*
      if(i <= 50){
      std::cout << "i = " << i
              << ", raw16_in0 = " << (int)raw16_in0
          << ", raw16_in1 = " << (int)raw16_in1
          << ", raw16_in = " << (int)raw16_in << std::endl;
      }*/

      dat_out.write(raw16_in);
      //raw16_out = raw16_in;
    }/*
    if(i <= 50){
      std::cout << "out: i = " << i
              << ", raw16_in0 = " << raw16_in0
          << ", raw16_in1 = " << raw16_in1
          << ", raw16_in = " << raw16_in << std::endl;
    }*/
    //dat_out.write(raw16_in);
    //if(i%2 == 0){
    //dat_out.write(raw16_in);
    //}

    if(i == (widthIn + 8) * (widthIn + 8) * 2 -1){
      break;
    }
  }
```

- Testbench:

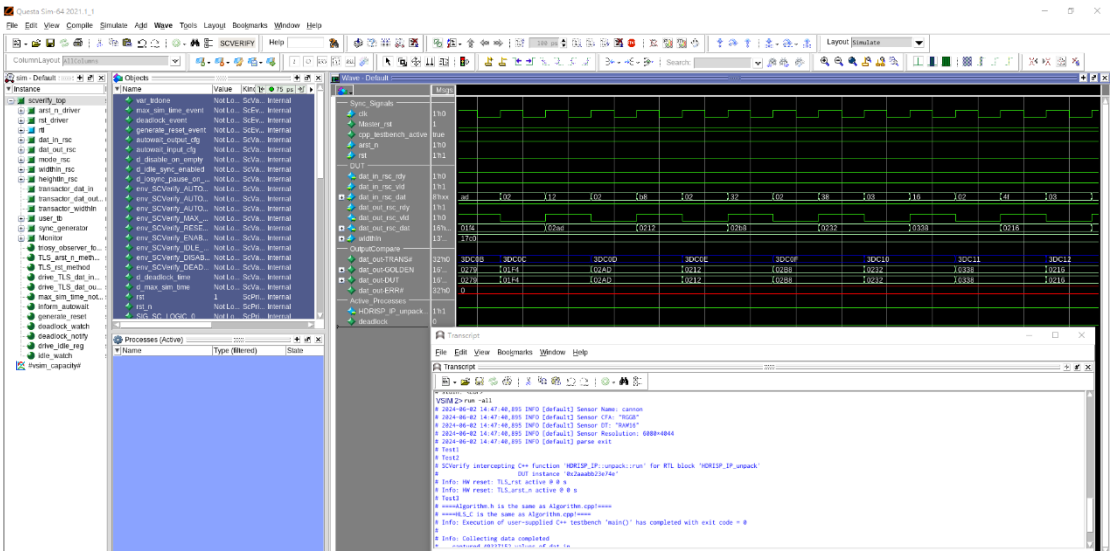
```
//unpack.h
Frame frame(isp_prms.info);
  frame.ReadFileToFrame(isp_prms.raw_file, width * height * isp_prms.info.bpp / 8);
ref_inst.MipiDataUnpack(&frame, &isp_prms);

//unpack.cpp
Frame frame2(isp_prms.info);
frame2.ReadFileToFrame(isp_prms.raw_file, width * height * isp_prms.info.bpp / 8);
MipiDataUnpack(&frame2, &isp_prms);

//hls_c
Frame frame3(isp_prms.info);
frame3.ReadFileToFrame(isp_prms.raw_file, width * height * isp_prms.info.bpp / 8);

uint8_t * dataIn_ptr = (uint8_t *)frame3.data.raw_u8_i;
uint8 data_hls = 0;
HDRISP_IP::stream_u8 dataIn_chn;
HDRISP_IP::stream_u16 dataOut_chn;

HDRISP_IP::maxWType widthIn = 6080;
  HDRISP_IP::maxHType heightIn = 4044;

int mode_fake = 0;
std::cout << "Test1" << std::endl;
for(int i = 0; i < (iW + 8) * (iH + 8) * 2; i++){
  data_hls = (uint8)dataIn_ptr[i];
  if(i <= 100){
    //std::cout << "i = " << i << ", data_hls = " << data_hls << std::endl;
  }
  dataIn_chn.write(data_hls);

}
std::cout << "Test2" << std::endl;
u_unpack.run(mode_fake, dataIn_chn, widthIn, heightIn, dataOut_chn);
std::cout << "Test3" << std::endl;
```

```
//compare unpack_algorithm (data1) and unpack.cpp (data2)
int err_num = 0;
uint16_t* data1;
uint16_t* data2;
data1 = (uint16_t*)frame.data.raw_u16_i;
data2 = (uint16_t*)frame2.data.raw_u16_i;

for(int i = 0; i < iH + 8; i++){
  for(int j = 0; j < iW + 8; j++){
    if((int)data1[i * iW + j] != (int)data2[i * iW + j]){
      std::cout << "No." << i * iW + j << " = " << (int)data1[i * iW + j]
                << ", data2 = " << (int)data2[i * iW + j] << std::endl;
      err_num += 1;
    }
  }
}

if(err_num == 0){
  std::cout << "====Algorithm.h is the same as Algorithm.cpp!====" << std::endl;
}

//compare unpack_hls (data_hls_out) and unpack.cpp (data2)
int hls_err_num = 0;
uint16 data_hls_out = 0;
for(int i = 0; i < (iW + 8) * (iH + 8); i++){
  data_hls_out = dataOut_chn.read();
  if((int)data_hls_out != (int)data2[i]){
    if(i <= 100){
      //std::cout << "No." << i << " = " << data_hls_out
      //          << ", data2 = " << (int)data2[i] << std::endl;
    }
    hls_err_num += 1;
  }
}

if(hls_err_num == 0){
  std::cout << "====HLS_C is the same as Algorithm.cpp!====" << std::endl;
}
```

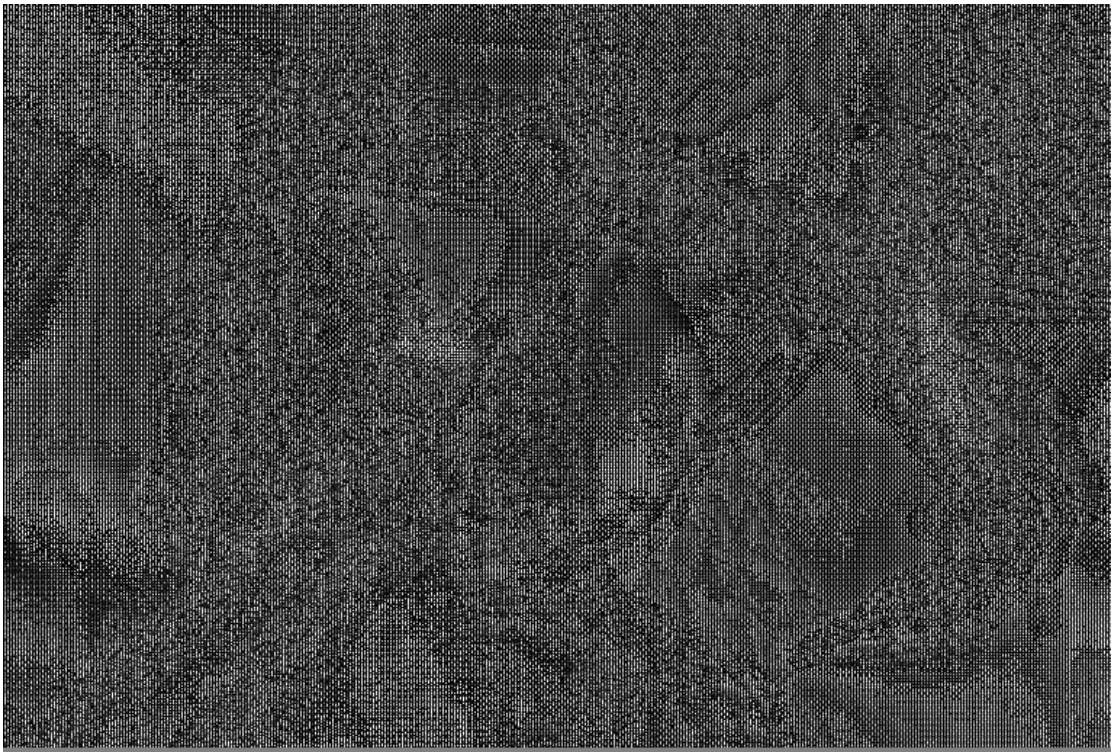
- Makefile C-sim & Catapult C-sim:



- Catapult C/RTL co-simulation:

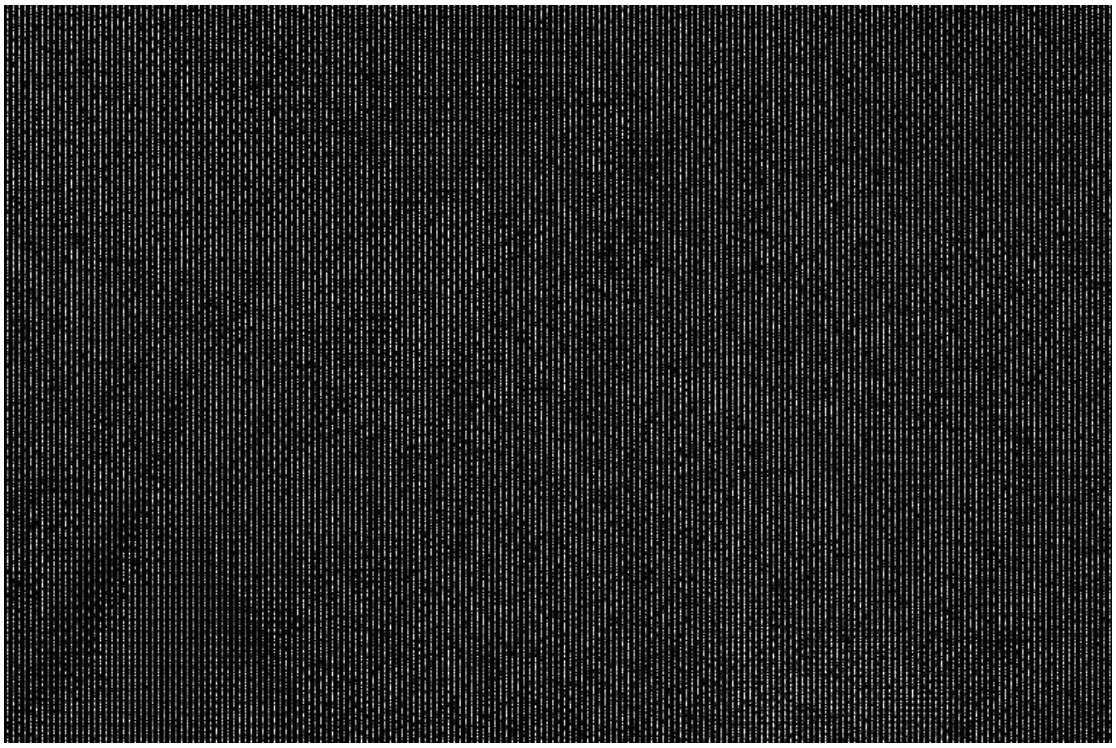


- RAW domain image:



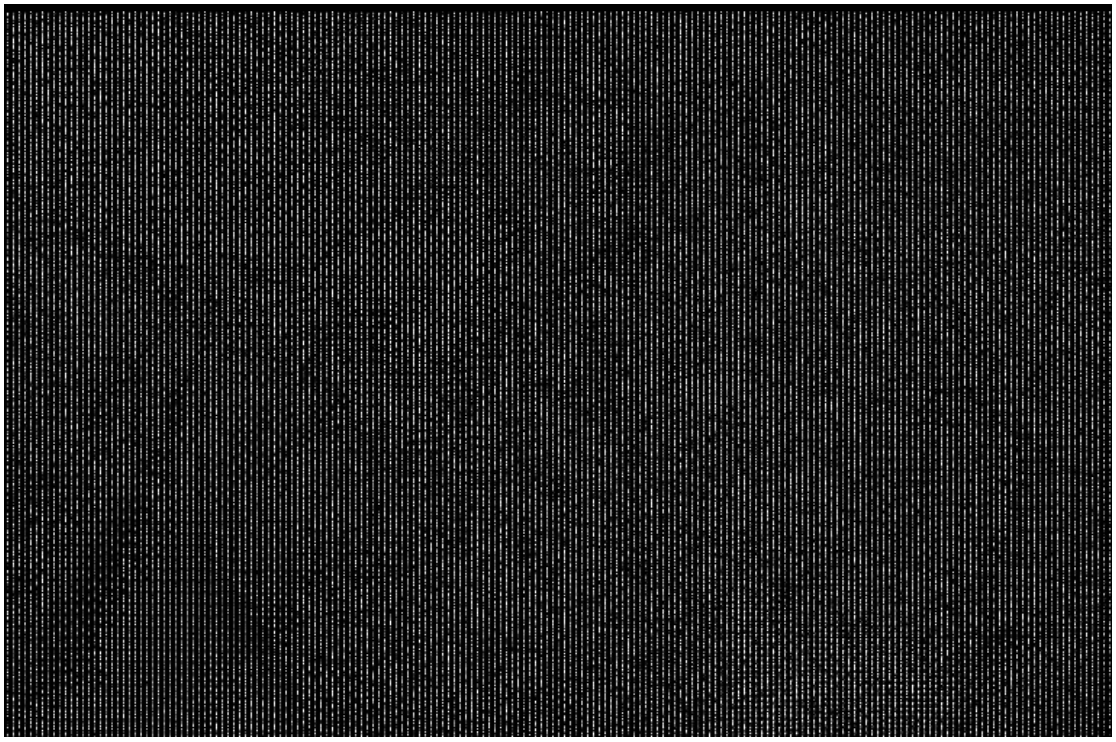
– Module: DEPWL

- Function: Decode the pixel values from the PWL curve.
- RAW domain image:

- 

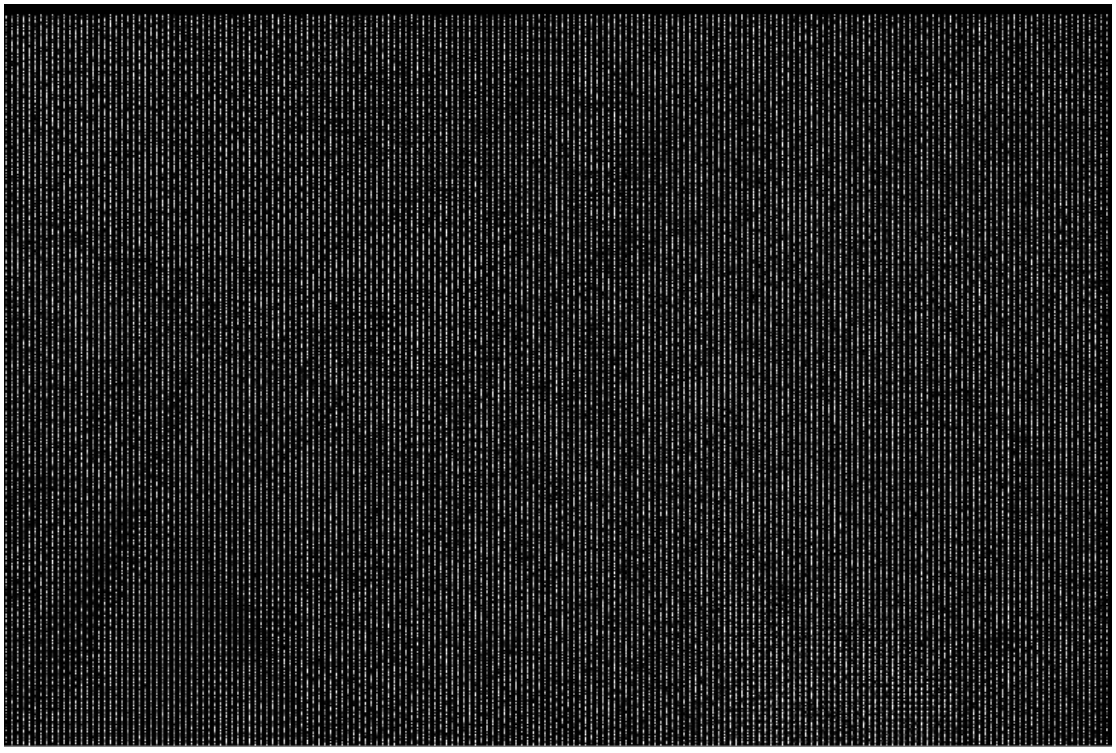
– Module: DPC

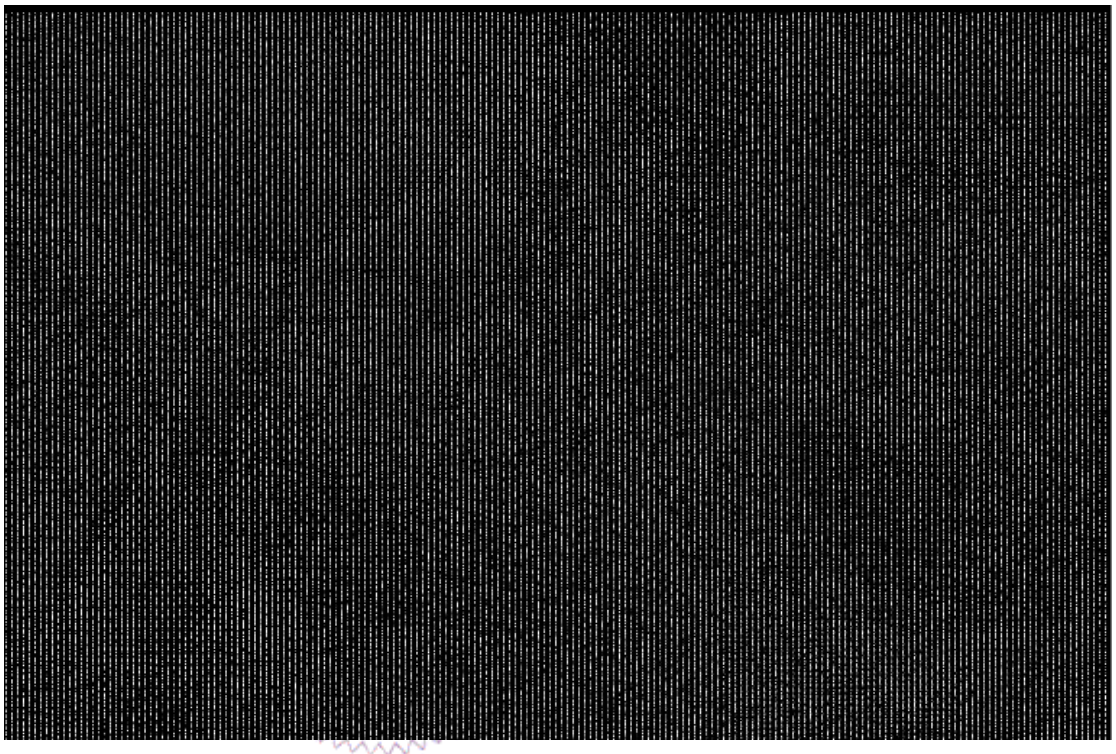
- Function: Dynamic pixel correction. The pixel values are compared with the 8 pixels near the central pixel value, and the pixel values will be assigned to the calculated value according to the conditions this pixel satisfied.
- RAW domain image:

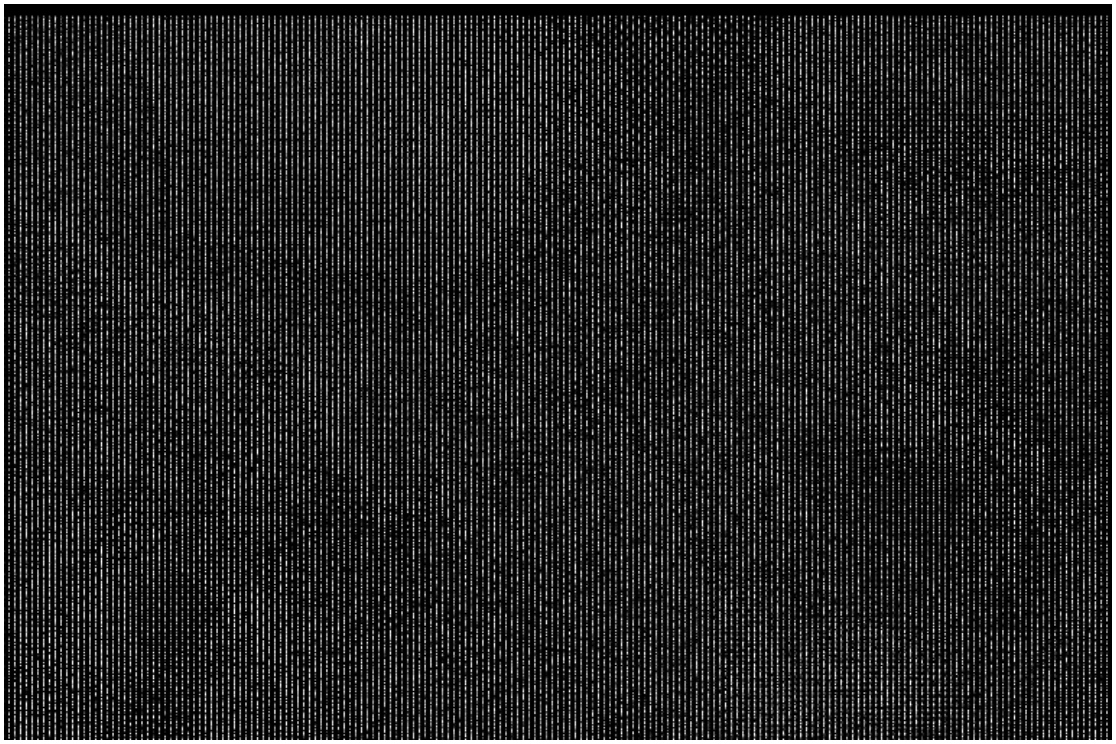
- 

– Module: LSC

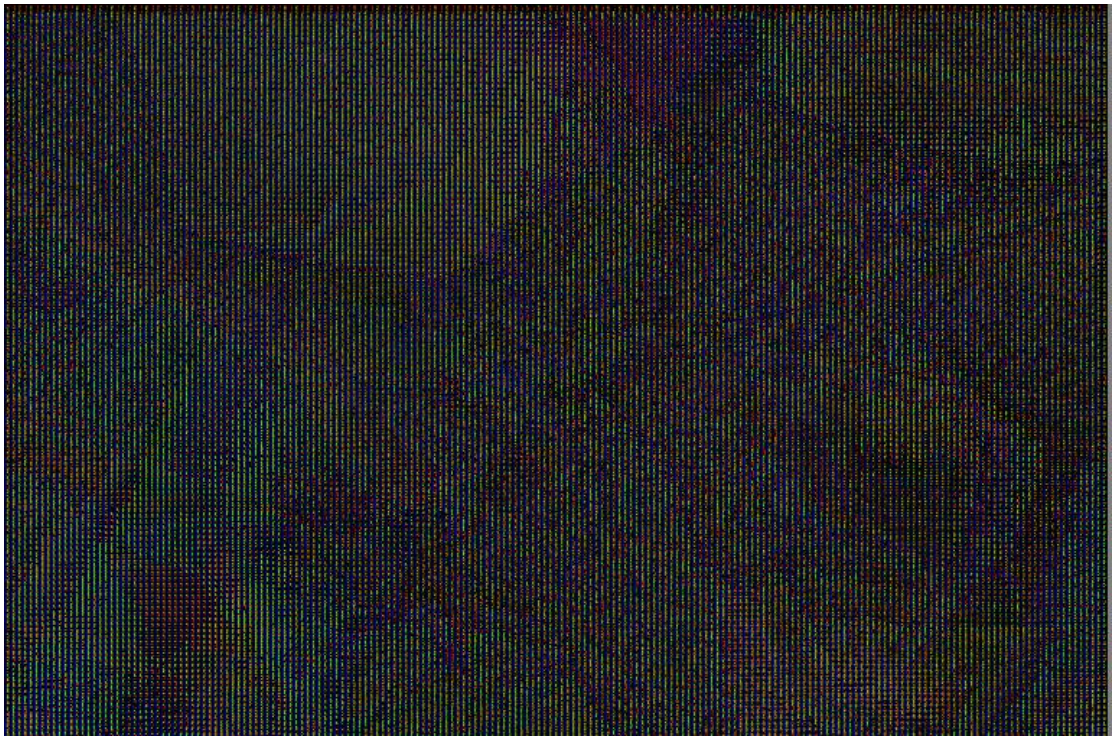
- Function: Lens shading correction.
- RAW domain image:

- 

- **Module: BLC**
  - Function: Black level correction.
  - RAW domain image:
  - 
- **Module: WBGAIN**
  - Function: White balance gain. Adjusting the pixel values by multiplying color gains calculated from the original pixels.
  - RAW domain image:

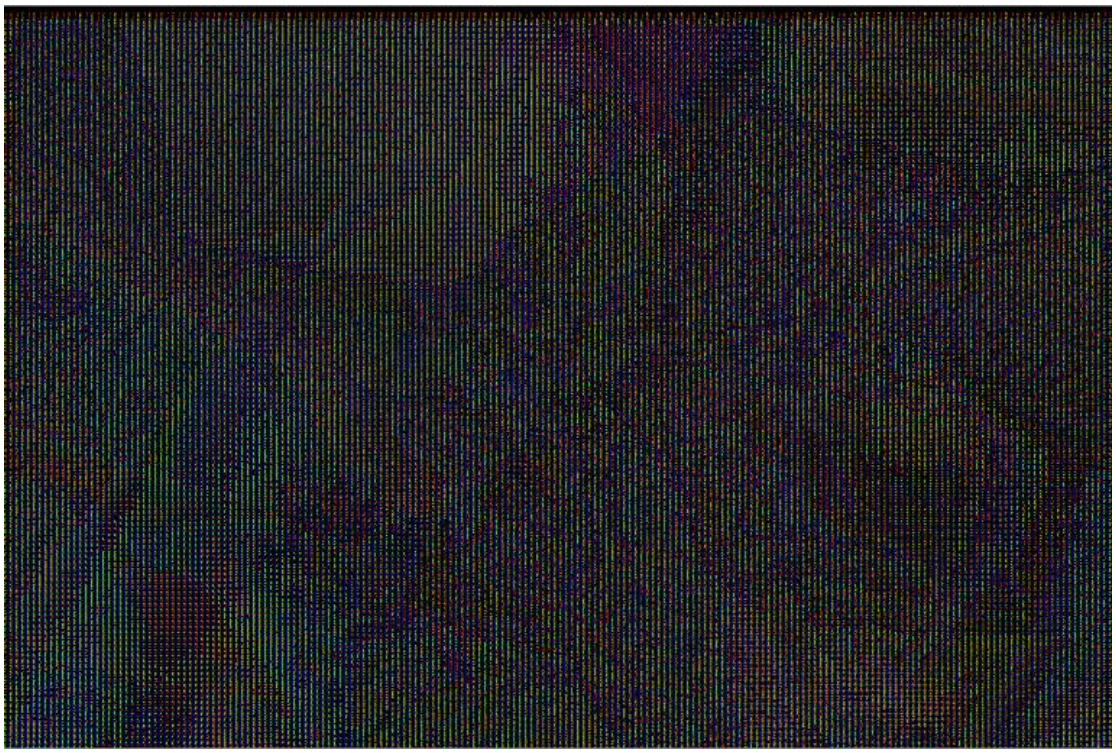
- 

- **Module: DEMOASIC**
- Function: Converting a single channel Bayer RAW image into a three-color image.
  - RGB domain image:

- 

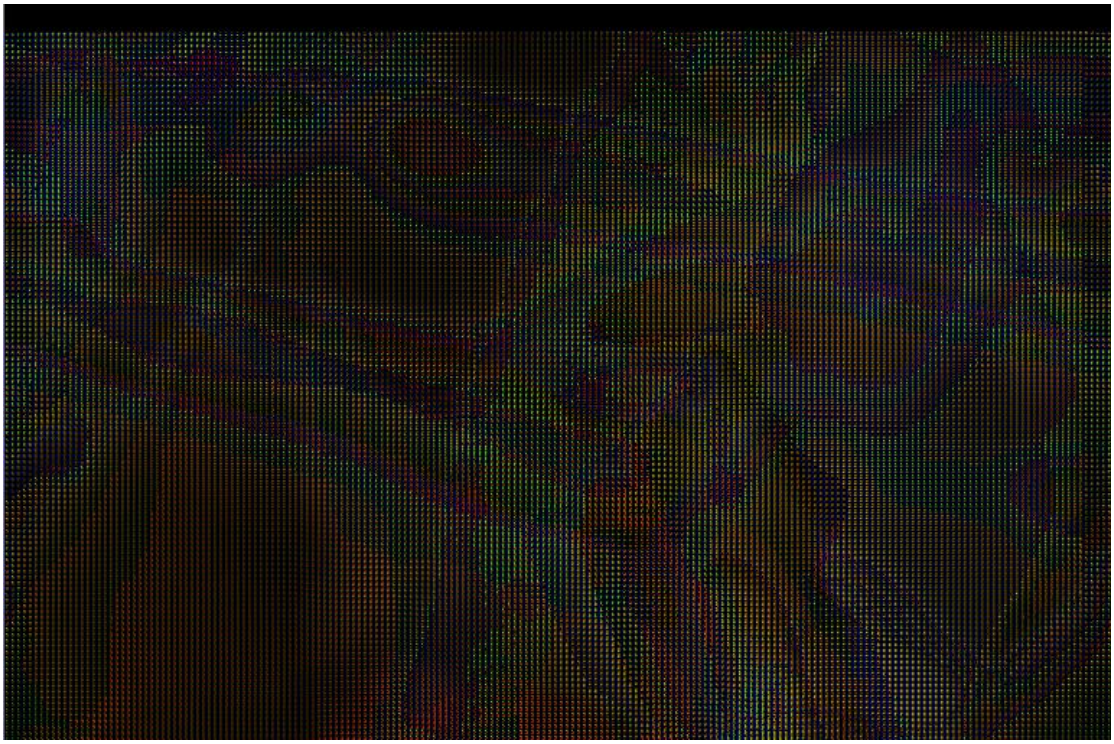
− Module: CCM

- Function: Color correct matrix. Using this matrix to correct the pixel values and results in a color picture that approaches the color seem by humans. This is a method to enhance white balance.
- RGB domain image:

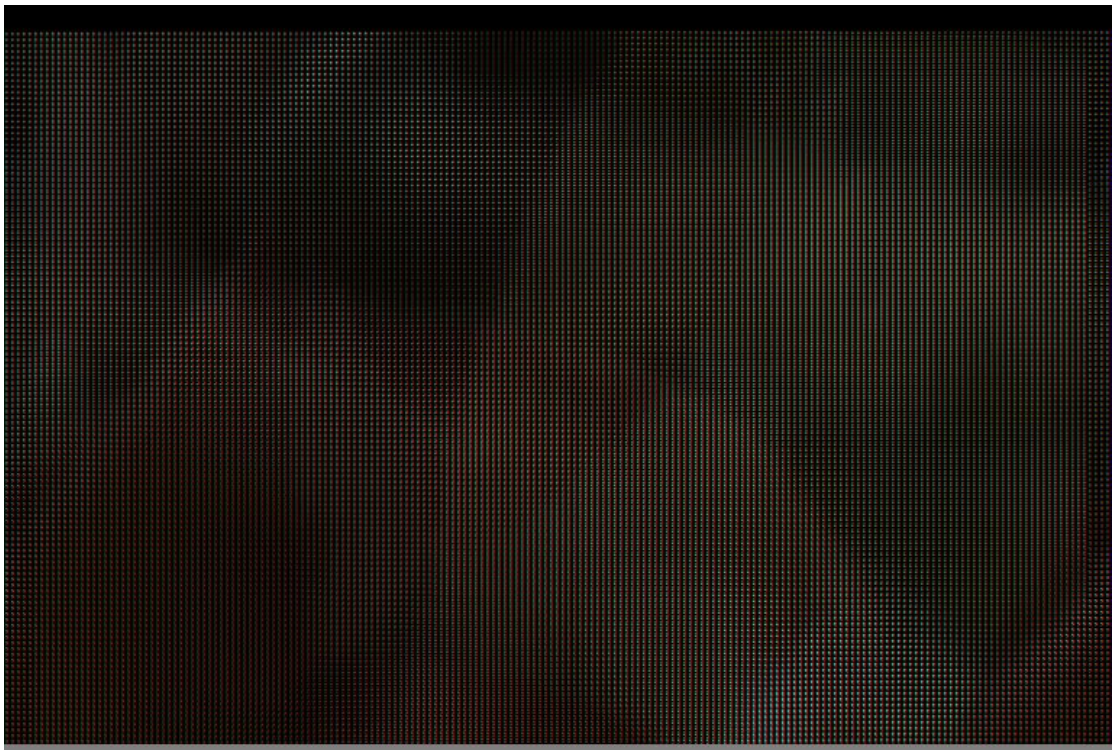
- 

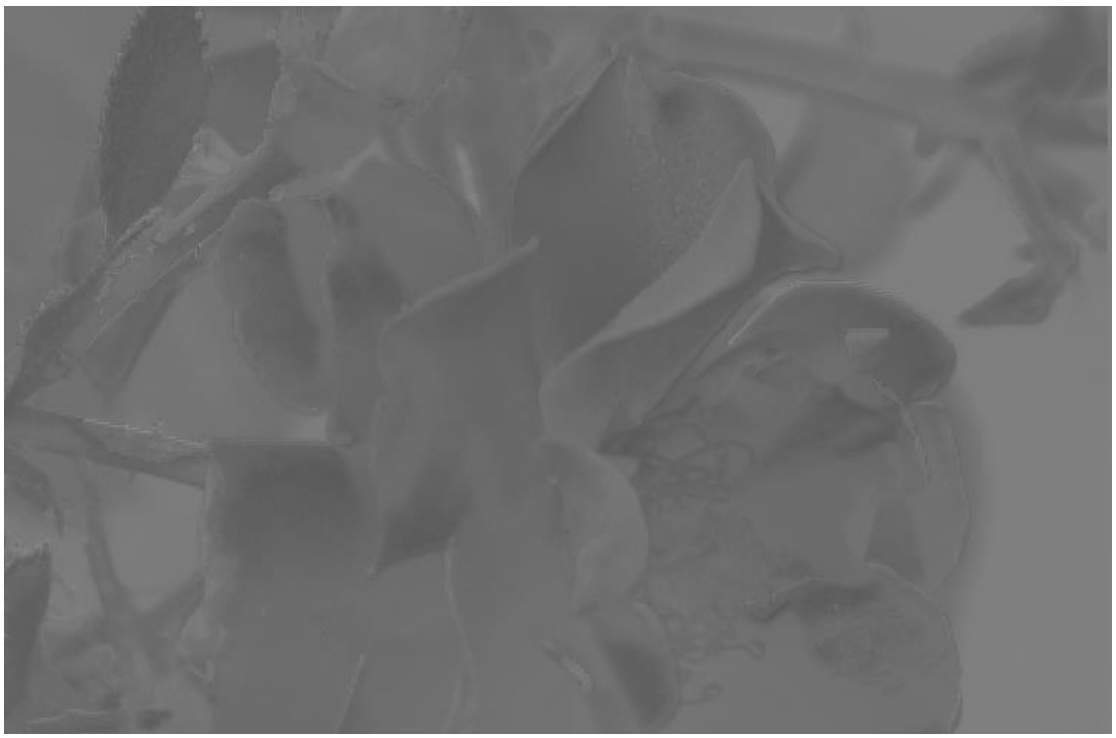
− Module: LTM

- Function: Local Tone Mapping. Local contrast enhancement based on brightness domain.
- RGB domain image:

- 

- **Module: RGBGAMMA**
  - Function: Gamma correction is to perform non-linear processing on the gray scale of the image.
  - RGB domain image:

- 

- **Module: RGB2YUV**
  - Function: Convert RGB domain to YUV domain.
  - U domain image:
    - 

- V domain image:
- 
- Y domain image:
- 

– Module: YGAMMA

- Function: gamma correction. Like the human eye increases sensitivity in dark areas and decreases sensitivity in highlights.
- Y domain image:
- 

− Module: CONTRAST

- Function: Control the brightness to change the picture effect.
- Y domain image:

- 

– Module: SHARPEN

- Function: Enhance the contrast between light and dark pixels.
- Y domain image:

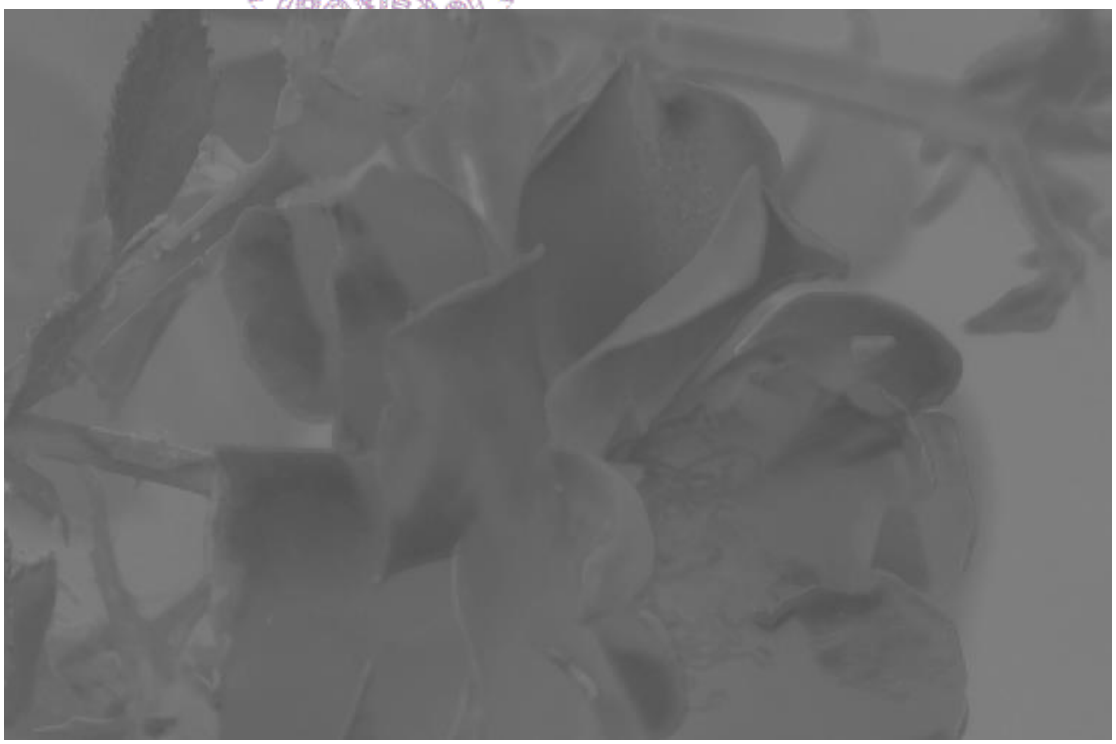
- 

– Module: CNS

- Function: Noise Reduction for Chroma.
- V domain image:

- U domain image:
- 

- **Module: SATURATION**
  - Function: Adjust color saturation.
  - V domain image:

- 
- U domain image:
- 

– Module: YUV2RGB

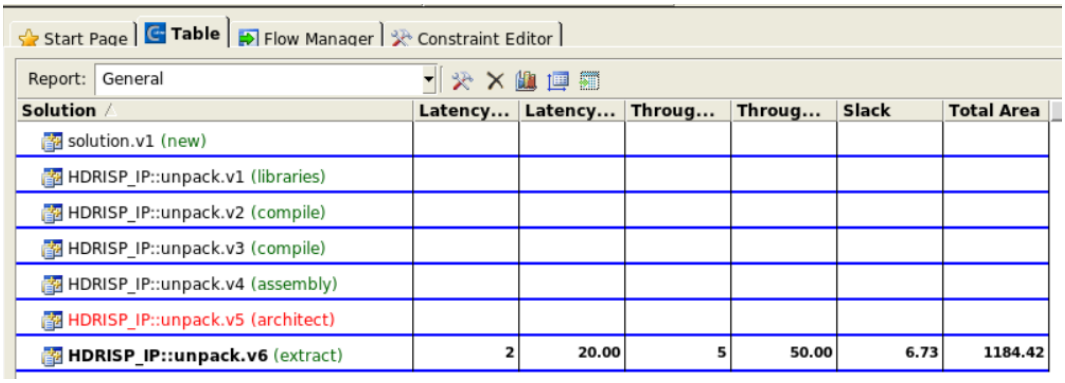
- Function: Convert image from YUV domain to RGB

domain.

- RGB domain image:

3. Reports:

- UNPACK:

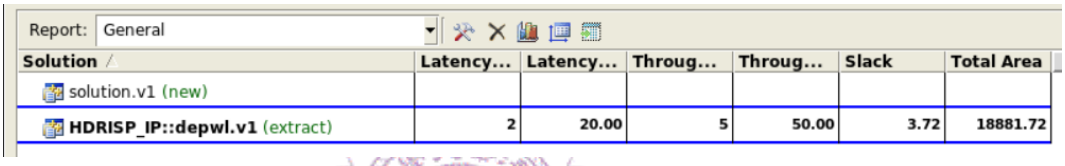


Start Page | Table | Flow Manager | Constraint Editor

Report: General

Solution	Latency...	Latency...	Throug...	Throug...	Slack	Total Area
solution.v1 (new)						
HDRISP_IP::unpack.v1 (libraries)						
HDRISP_IP::unpack.v2 (compile)						
HDRISP_IP::unpack.v3 (compile)						
HDRISP_IP::unpack.v4 (assembly)						
HDRISP_IP::unpack.v5 (architect)						
HDRISP_IP::unpack.v6 (extract)	2	20.00	5	50.00	6.73	1184.42

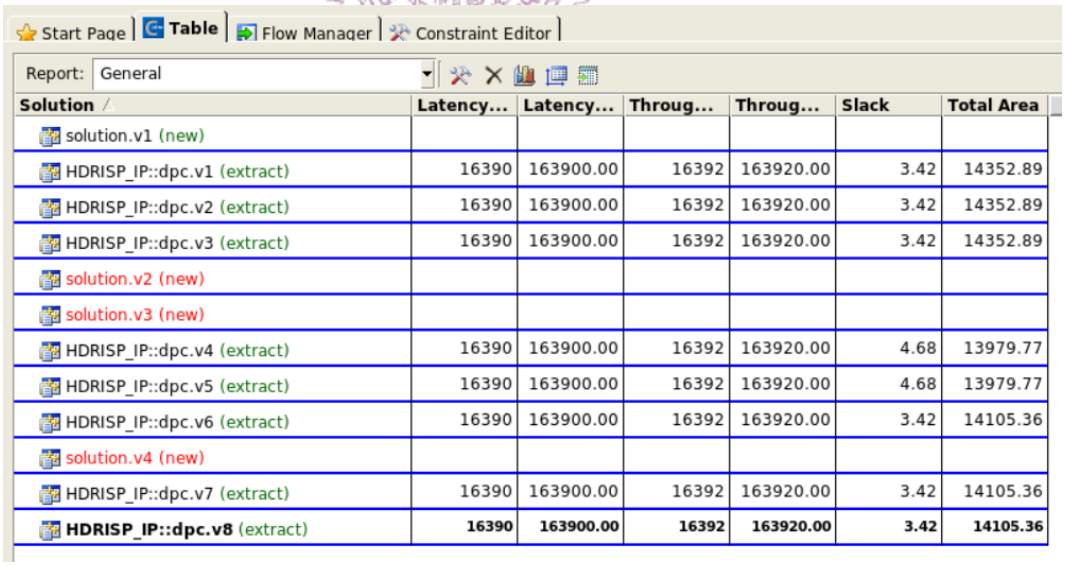
- DEPWL:



Report: General

Solution	Latency...	Latency...	Throug...	Throug...	Slack	Total Area
solution.v1 (new)						
HDRISP_IP::depwl.v1 (extract)	2	20.00	5	50.00	3.72	18881.72

- DPC:



Start Page | Table | Flow Manager | Constraint Editor

Report: General

Solution	Latency...	Latency...	Throug...	Throug...	Slack	Total Area
solution.v1 (new)						
HDRISP_IP::dpc.v1 (extract)	16390	163900.00	16392	163920.00	3.42	14352.89
HDRISP_IP::dpc.v2 (extract)	16390	163900.00	16392	163920.00	3.42	14352.89
HDRISP_IP::dpc.v3 (extract)	16390	163900.00	16392	163920.00	3.42	14352.89
solution.v2 (new)						
solution.v3 (new)						
HDRISP_IP::dpc.v4 (extract)	16390	163900.00	16392	163920.00	4.68	13979.77
HDRISP_IP::dpc.v5 (extract)	16390	163900.00	16392	163920.00	4.68	13979.77
HDRISP_IP::dpc.v6 (extract)	16390	163900.00	16392	163920.00	3.42	14105.36
solution.v4 (new)						
HDRISP_IP::dpc.v7 (extract)	16390	163900.00	16392	163920.00	3.42	14105.36
HDRISP_IP::dpc.v8 (extract)	16390	163900.00	16392	163920.00	3.42	14105.36

- LSC:

Report: General

Solution	Latency...	Latency...	Throug...	Throug...	Slack	Total Area
solution.v1 (new)						
HDRISP_IP::lsc.v1 (compile)						
HDRISP_IP::lsc.v2 (compile)						
solution.v2 (analyze)						
HDRISP_IP::lsc.v3 (extract)	6	60.00	9	90.00	0.74	207986.50

- BLC:

Solution	Latency...	Latency...	Throug...	Throug...	Slack	Total Area
solution.v1 (new)						
HDRISP_IP::blc.v1 (extract)	2	20.00	5	50.00	7.15	1018.01

- WBGAIN:

- DEMOSAIC:

Start Page | Table | Flow Manager | Constraint Editor

Report: General

Solution	Latency...	Latency...	Throug...	Throug...	Slack	Total Area
solution.v1 (new)						
HDRISP_IP::demosaic.v1 (extract)	3	30.00	7	70.00	6.16	6066.69

- CCM:

Start Page | Table | Flow Manager | Constraint Editor

Report: General

Solution	Latency...	Latency...	Throug...	Throug...	Slack	Total Area
solution.v1 (new)						
HDRISP_IP::ccm.v1 (extract)	4	40.00	8	80.00	5.56	27674.11

- LTM:

- RGBGAMMA:

Solution	Latency...	Latency...	Throug...	Throug...	Slack	Total Area
solution.v1 (new)						
HDRISP_IP::rgbgamma.v1 (extract)	4	40.00	7	70.00	1.00	19941.27

- RGB2YUV:

Solution	Latency...	Latency...	Throug...	Throug...	Slack	Total Area
solution.v1 (new)						
HDRISP_IP::rgb2yuv.v1 (extract)	3	30.00	8	80.00	6.65	5351.60

- YGAMMA:

Start Page | Table | Flow Manager | Constraint Editor

Report: General

Solution	Latency...	Latency...	Throug...	Throug...	Slack	Total Area
solution.v1 (new)						
HDRISP_IP::ygamma.v1 (extract)	2	20.00	5	50.00	0.02	15435.53

- CONTRAST:

Report: General

Solution	Latency...	Latency...	Throug...	Throug...	Slack	Total Area
solution.v1 (new)						
HDRISP_IP::contrast.v1 (extract)	2	20.00	5	50.00	6.48	2018.24

- SHARPEN:
- CNS:

Start Page | Flow Manager | Table | Constraint Editor

Report: General

Solution	Latency...	Latency...	Throug...	Throug...	Slack	Total Area
solution.v1 (new)						
HDRISP_IP::cns.v1 (extract)	3	30.00	7	70.00	6.18	6012.74

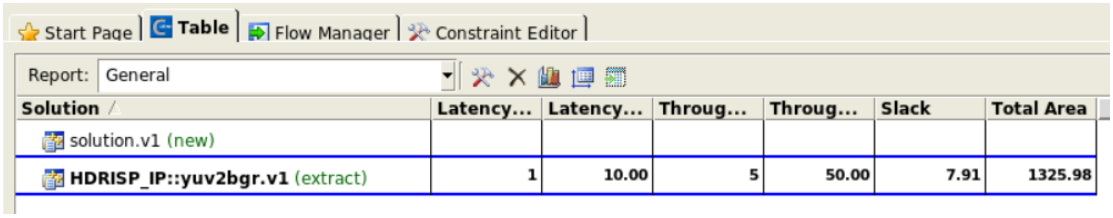
- SATURATION:

Start Page | Table | Flow Manager | Constraint Editor

Report: General

Solution	Latency...	Latency...	Throug...	Throug...	Slack	Total Area
solution.v1 (new)						
HDRISP_IP::saturation.v1 (extract)	201	2010.00	204	2040.00	5.61	7824.46

- YUV2RGB:



4. Distribution of work:

- HDR-ISP Module: 蔡宗穎
- HDR-ISP Module: 張傑閔
- Discussion with each other when we were faced with some problems.