Edge detect Report

1. 5 modifications

a. Process four pixels per clock cycle

為了能夠一次傳送四筆 pixel 與 gradient，我們定義了新的 datatype

```
// Define some bit-accurate types to use in this model
typedef uint8                 pixelType;    // input pixel is 0-255
typedef uint16                pixelType2x;  // two pixels packed
typedef uint32                pixelType4x;  // four pixels packed
typedef ac_int<64, false>     pixelType8x;  // eight pixels packed
typedef int9                  gradType;     // Derivative is max range -255 to 255
typedef int36                 gradType4x;   // 4 gradients packed
typedef uint18                sqType;       // Result of 9-bit x 9-bit
typedef ac_fixed<19,19,false> sumType;      // Result of 18-bit + 18-bit fixed pt integer for squareroot
typedef uint9                 magType;      // 9-bit unsigned magnitute result
typedef ac_fixed<8,3,true>    angType;      // 3 integer bit, 5 fractional bits for quantized angle -pi to pi

// Compute number of bits for max image size count, used internally and in testbench
typedef ac_int<ac::nbits<maxImageWidth+1>::val,false> maxWType;
typedef ac_int<ac::nbits<maxImageHeight+1>::val,false> maxHType;

struct  Stream_t{
  pixelType4x pix;
  bool   sof;
  bool   eol;
};
```

其中 stream_t 包含 4 個 pixel 與 tb 會測試的 sof 與 eol。

此外，在 verder 及 horder 中，會需要同時計算四筆 pix 經過 filter 得到的結果，我們採用手動 unroll 的方式，並利用.slc 與.set_slc 來獲取對應的 pix 位置。

```
// Calculate derivative
//pix = pix2*kernel[0] + pix1*kernel[1] + pix0*kernel[2];
g0 = pix0.slc<8>(0) * kernel[2] + pix1.slc<8>(0) * kernel[1] + pix2.slc<8>(0) * kernel[0];
g1 = pix0.slc<8>(8) * kernel[2] + pix1.slc<8>(8) * kernel[1] + pix2.slc<8>(8) * kernel[0];
g2 = pix0.slc<8>(16)* kernel[2] + pix1.slc<8>(16)* kernel[1] + pix2.slc<8>(16)* kernel[0];
g3 = pix0.slc<8>(24)* kernel[2] + pix1.slc<8>(24)* kernel[1] + pix2.slc<8>(24)* kernel[0];
pix.set_slc(0,g0);
pix.set_slc(9,g1);
pix.set_slc(18,g2);
pix.set_slc(27,g3);
```

```
// Calculate derivative
//pix = pix2*kernel[0] + pix1*kernel[1] + pix0*kernel[2];
g0 = pix2.slc<8>(24) * kernel[0] + pix1.slc<8>(0) * kernel[1] + pix1.slc<8>(8) * kernel[2];
g1 = pix1.slc<8>(0) * kernel[0] + pix1.slc<8>(8) * kernel[1] + pix1.slc<8>(16) * kernel[2];
g2 = pix1.slc<8>(8) * kernel[0] + pix1.slc<8>(16) * kernel[1] + pix1.slc<8>(24) * kernel[2];
g3 = pix1.slc<8>(16) * kernel[0] + pix1.slc<8>(24) * kernel[1] + pix0.slc<8>(0) * kernel[2];
pix.set_slc(0,g0);
pix.set_slc(9,g1);
pix.set_slc(18,g2);
pix.set_slc(27,g3);
```

b. Use SAD

分別將四筆 dx 與 dy 取出後，透過手動的方式得到 abs value，再做相加，並考慮 overflow

```
dx0 = dx.slc<9>(0);
dx1 = dx.slc<9>(9);
dx2 = dx.slc<9>(18);
dx3 = dx.slc<9>(27);
dy0 = dy.slc<9>(0);
dy1 = dy.slc<9>(9);
dy2 = dy.slc<9>(18);
dy3 = dy.slc<9>(27);

if (dx0>=0) abs_dx0=dx0; else abs_dx0 = -dx0;
if (dx1>=0) abs_dx1=dx1; else abs_dx1 = -dx1;
if (dx2>=0) abs_dx2=dx2; else abs_dx2 = -dx2;
if (dx3>=0) abs_dx3=dx3; else abs_dx3 = -dx3;
if (dy0>=0) abs_dy0=dy0; else abs_dy0 = -dy0;
if (dy1>=0) abs_dy1=dy1; else abs_dy1 = -dy1;
if (dy2>=0) abs_dy2=dy2; else abs_dy2 = -dy2;
if (dy3>=0) abs_dy3=dy3; else abs_dy3 = -dy3;

if ((abs_dx0 + abs_dy0) >= 255) sum0 = 255; else sum0 = (abs_dx0 + abs_dy0);
if ((abs_dx1 + abs_dy1) >= 255) sum1 = 255; else sum1 = (abs_dx1 + abs_dy1);
if ((abs_dx2 + abs_dy2) >= 255) sum2 = 255; else sum2 = (abs_dx2 + abs_dy2);
if ((abs_dx3 + abs_dy3) >= 255) sum3 = 255; else sum3 = (abs_dx3 + abs_dy3);

sum.set_slc(0, sum0);
sum.set_slc(8, sum1);
sum.set_slc(16, sum2);
sum.set_slc(24, sum3);
```

c. Use crc32

如圖，使用 calc_crc32 這個 function 對 streamin.pix 做一次，透過 sw 選擇 streamout 資料後，再對 streamout.pix 做一次。

```
crc32_pix_in = calc_crc32<32>(crc32_pix_in, streamin.pix);

if (sw_in) streamin.pix = sum;

crc32_dat_out = calc_crc32<32>(crc32_dat_out, streamin.pix);

streamout.pix = streamin.pix;
streamout.sof = (x == 0) && (y == 0);
streamout.eol = (x == maxWType(widthIn-1));

dat_out.write(streamout);
```

d. Select output source

如上圖，使用 sw 選擇 streamout 會是 streamin 或 sum

e. Remove angle

將 angle caculation 相關部分刪除。

2. catapult test result

FATAL	Violated	Waived	Undecided
ERROR	**Violated**	**Waived**	**Undecided**
ABR - Array Bounds Read	0	0	0
ABW - Array Bounds Write	0	0	0
AOB - Arithmetic Operator with Boolean	0	0	0
CAS - Incomplete Switch-Case	0	0	0
DBZ - Divide By Zero	0	0	0
ISE - Illegal Shift Error	0	0	0
OVL - Overflow/Underflow	11	0	2
RRT - Reset referenced in thread	0	0	0
UMR - Uninitialized Memory Read	10	2	0

檢查 OVL 部分都沒有甚麼問題，而 UMR 在講義中也出現，便先無視。

```
# height: 864 width: 1296
# Running
# SCVerify intercepting C++ function 'EdgeDetect_IP::EdgeDetect_Top::run' for RTL block 'EdgeDetect_IP_EdgeDetect_Top'
#                      DUT instance '0x2aaab4cd9b80'
# Info: HW reset: TLS rst active @ 0 s
# Info: HW reset: TLS arst_n active @ 0 s
# Magnitude: Manhattan norm per pixel 3.655830
# Writing algorithmic bitmap output to: out_alg.bmp
# Writing bit-accurate bitmap output to: out_hw.bmp
# sofErr: 0 eolErr: 0
# crc32_alg_pix_in  = 79b62527  crc32_hw_pix_in  = 79b62527
# crc32_alg_dat_out = 427a0911  crc32_hw_dat_out = 3ee1dd06
# Finished
# Info: Execution of user-supplied C++ testbench 'main()' has completed with exit code = 0
#
# Info: Collecting data completed
#    captured 1 values of widthIn
#    captured 1 values of heightIn
#    captured 1 values of sw_in
#    captured 1 values of crc32_pix_in_IN
#    captured 1 values of crc32_pix_in
#    captured 1 values of crc32_dat_out_IN
#    captured 1 values of crc32_dat_out
#    captured 279936 values of dat_in_pix
#    captured 279936 values of dat_in_sof
#    captured 279936 values of dat_in_eol
#    captured 279936 values of dat_out_pix
#    captured 279936 values of dat_out_sof
#    captured 279936 values of dat_out_eol
# Info: scverify_top/user_tb: Simulation completed
#
# Checking results
# 'crc32_pix_in'
#    capture count       = 1
#    comparison count    = 1
#    ignore count        = 0
#    error count         = 0
#    stuck in dut fifo   = 0
#    stuck in golden fifo = 0
# 'crc32_dat_out'
#    capture count       = 1
#    comparison count    = 1
#    ignore count        = 0
#    error count         = 0
#    stuck in dut fifo   = 0
#    stuck in golden fifo = 0
# 'dat_out_pix'
#    capture count       = 279936
#    comparison count    = 279936
#    ignore count        = 0
#    error count         = 0
#    stuck in dut fifo   = 0
#    stuck in golden fifo = 0
# 'dat_out_sof'
#    capture count       = 279936
#    comparison count    = 279936
#    ignore count        = 0
#    error count         = 0
#    stuck in dut fifo   = 0
#    stuck in golden fifo = 0
# 'dat_out_eol'
#    capture count       = 279936
#    comparison count    = 279936
#    ignore count        = 0
#    error count         = 0
#    stuck in dut fifo   = 0
#    stuck in golden fifo = 0
#
# Info: scverify_top/user_tb: Simulation PASSED @ 2837266 ns
# ** Note: (vsim-6574) SystemC simulation stopped by user.
# 1
#
```

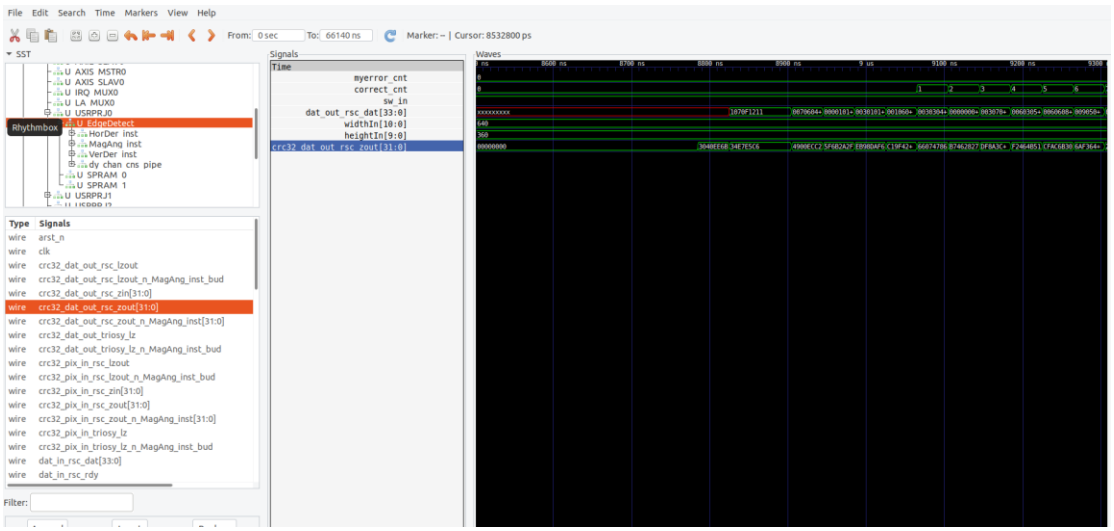
Questa sim 結果與預期也都相同。

3. integrate to FSIC

   使用 lab fsic-sim 的檔案架構，更改 tb。在處理完 reset 之後，使用 soc_cfg_write 將圖片長寬與 sw 餵進去，並使用 fpga_axis_req stream in data，在 is_as_tvalid 時比對 data。

   User_prj0 也需要修改，將 edgedetect 介面 port 都改成與 HLS 產生的 verilog 一致，並注意 bit 數。

4. simulation result of FSIC



比對 golden data 與 user_prj0 產生的 data，可發現都一致。

```
Correct at expect  574630159, get  574630159
             2391625=> fpga_is_as_data_valid wait fpga_is_as_tvalid done, fpga_is_as_tvalid = 1
             2391625=> fpga_is_as_data_valid wait fpga_is_as_tvalid
Correct at expect  640560662, get  640560662
             2391665=> fpga_is_as_data_valid wait fpga_is_as_tvalid done, fpga_is_as_tvalid = 1
             2391665=> fpga_is_as_data_valid wait fpga_is_as_tvalid
Correct at expect  992240483, get  992240483
             2391705=> fpga_is_as_data_valid wait fpga_is_as_tvalid done, fpga_is_as_tvalid = 1
             2391705=> fpga_is_as_data_valid wait fpga_is_as_tvalid
Correct at expect  778780736, get  778780736
             2391745=> fpga_is_as_data_valid wait fpga_is_as_tvalid done, fpga_is_as_tvalid = 1
             2391745=> fpga_is_as_data_valid wait fpga_is_as_tvalid
Correct at expect  256394028, get  256394028
             2391785=> fpga_is_as_data_valid wait fpga_is_as_tvalid done, fpga_is_as_tvalid = 1
             2391785=> fpga_is_as_data_valid wait fpga_is_as_tvalid
Correct at expect  788922882, get  788922882
             2391825=> fpga_is_as_data_valid wait fpga_is_as_tvalid done, fpga_is_as_tvalid = 1
             2391825=> fpga_is_as_data_valid wait fpga_is_as_tvalid
Correct at expect  134877740, get  134877740
             2391865=> fpga_is_as_data_valid wait fpga_is_as_tvalid done, fpga_is_as_tvalid = 1
             2391865=> fpga_is_as_data_valid wait fpga_is_as_tvalid
Correct at expect  201592843, get  201592843
             2391905=> fpga_is_as_data_valid wait fpga_is_as_tvalid done, fpga_is_as_tvalid = 1
             2391905=> fpga_is_as_data_valid wait fpga_is_as_tvalid
Correct at expect  437586192, get  437586192
-------------------------------------------------
correct_cnt =       57600, error_cnt =           0
```