Advanced SoC Lab2-2

組別:第四組

一、 How you design your work (5 modifications) .

1. **Process four pixels per clock cycle.**

   首先，定義一個具有 4 pixels、eol、sof 的 Data type

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

   Top module 的 Data in 修改成該 Data type (Stream_t)

```
//------------------------------------------------------------------------
// Function: run
//   Top interface for data in/out of class. Combines vertical and
//   horizontal derivative and magnitude/angle computation.
#pragma hls_design interface
void CCS_BLOCK(run)(maxWType             &widthIn,
                    maxHType             &heightIn,
                    bool                 &sw_in,
                    uint32               &crc32_pix_in,
                    uint32               &crc32_dat_out,
                    ac_channel<Stream_t>  &dat_in,
                    ac_channel<Stream_t>  &dat_out)
{
  VerDer_inst.run(dat_in, widthIn, heightIn, pix_chan1, dy_chan);
  HorDer_inst.run(pix_chan1, widthIn, heightIn, pix_chan2, dx_chan);
  MagAng_inst.run(dx_chan, dy_chan, pix_chan2, widthIn, heightIn, sw_in, crc32_pix_in, crc32_dat_out, dat_out);
}
```

   將原本 1 pixel 的運算擴展 4 倍。

2. **Use sum of absolute difference (SAD) for edge magnitude calculation.**

   透過 sign bit 去判斷該數的正負。

   如果 sign bit = 0，維持不變。

   如果 sign bit = 1，則取負數，使其該數變為正數。

```
dx0_abs = (dx0[8]) ? (-dx0).slc<8>(0) : dx0.slc<8>(0);
dy0_abs = (dy0[8]) ? (-dy0).slc<8>(0) : dy0.slc<8>(0);
dx1_abs = (dx1[8]) ? (-dx1).slc<8>(0) : dx1.slc<8>(0);
dy1_abs = (dy1[8]) ? (-dy1).slc<8>(0) : dy1.slc<8>(0);
dx2_abs = (dx2[8]) ? (-dx2).slc<8>(0) : dx2.slc<8>(0);
dy2_abs = (dy2[8]) ? (-dy2).slc<8>(0) : dy2.slc<8>(0);
dx3_abs = (dx3[8]) ? (-dx3).slc<8>(0) : dx3.slc<8>(0);
dy3_abs = (dy3[8]) ? (-dy3).slc<8>(0) : dy3.slc<8>(0);

abs_sum_0 = (dx0_abs + dy0_abs);
abs_sum_1 = (dx1_abs + dy1_abs);
abs_sum_2 = (dx2_abs + dy2_abs);
abs_sum_3 = (dx3_abs + dy3_abs);

sum_0 = (abs_sum_0[8]) ? over_flow : abs_sum_0.slc<8>(0);
sum_1 = (abs_sum_1[8]) ? over_flow : abs_sum_1.slc<8>(0);
sum_2 = (abs_sum_2[8]) ? over_flow : abs_sum_2.slc<8>(0);
sum_3 = (abs_sum_3[8]) ? over_flow : abs_sum_3.slc<8>(0);
```

3. Add two crc32 calculation on image input / output.
   設置初始值為 0xFFFFFFFF

```
crc32_pix_in = 0XFFFFFFFF;
crc32_dat_out = 0XFFFFFFFF;
MROW: for (maxHType y = 0; ;
```

接著，再將輸出結果(sum) 透過 calc_crc32 進行運算。

```
pix2=pix_chan2.read();
crc32_pix_in = calc_crc32(crc32_pix_in, pix2);
if (sw_in == 1) {
  sum.pix.set_slc(0 ,sum_0);
  sum.pix.set_slc(8 ,sum_1);
  sum.pix.set_slc(16,sum_2);
  sum.pix.set_slc(24,sum_3);
  crc32_dat_out = calc_crc32(crc32_dat_out, sum.pix);
} else {
  sum.pix.set_slc(0,pix2);
  crc32_dat_out = calc_crc32(crc32_dat_out, sum.pix);
}
dat_out.write(sum);
```

4. Select the output source from input image or the calculated magnitude.

透過 sw_in 判斷輸出結果(sum)應該為 Edge_map (sum_0,sum_1, sum_2, sum_3) 還是 Orig. Img (pix2)。

```
pix2=pix_chan2.read();
crc32_pix_in = calc_crc32(crc32_pix_in, pix2);
if (sw_in == 1) {
  sum.pix.set_slc(0 ,sum_0);
  sum.pix.set_slc(8 ,sum_1);
  sum.pix.set_slc(16,sum_2);
  sum.pix.set_slc(24,sum_3);
  crc32_dat_out = calc_crc32(crc32_dat_out, sum.pix);
} else {
  sum.pix.set_slc(0,pix2);
  crc32_dat_out = calc_crc32(crc32_dat_out, sum.pix);
}
dat_out.write(sum);
```

5. Remove the angle calculation.

將 angle output 刪除

```
class EdgeDetect_MagAng
{
public:
  EdgeDetect_MagAng() {}

  #pragma hls_design interface
  void CCS_BLOCK(run)(ac_channel<gradType4x>  &dx_chan,
                      ac_channel<gradType4x>  &dy_chan,
                      ac_channel<pixelType4x> &pix_chan2,
                      maxWType                &widthIn,
                      maxHType                &heightIn,
                      bool                    &sw_in,
                      uint32                  &crc32_pix_in,
                      uint32                  &crc32_dat_out,
                      ac_channel<Stream_t>    &dat_out)
  {
```

二、What’ s the test result of catapult design(C design checker, testbench)

C design checker:

FATAL	Violated	Waived	Undecided

ERROR	Violated	Waived	Undecided
ABR - Array Bounds Read	2	0	0
ABW - Array Bounds Write	2	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	8	0	6
RRT - Reset referenced in thread	0	0	0
UMR - Uninitialized Memory Read	5	0	0

WARNING	Violated	Waived	Undecided
ACC - Accumulator of native C type	0	0	0
ACS - Accumulator of saturated type	0	0	0
AIC - Assignment used Instead of Comparison	0	0	0
ALS - Ac_int Left Shift check	0	0	0
AWE - Assignments Without Effect	0	0	0
CBU - Conditional break in Unrolled Loop	0	0	0
CCC - Static constant comparison	0	0	0
CGR - Conditional Guard in Rolled Loop	0	0	0
CIA - Comparison Instead of Assignment	0	0	0
CNS - Constant condition of if/switch	0	0	0
CWB - Case Without Break	0	0	0
DIU - Dynamic Index in Unrolled Loop	0	0	0
FVI - For Loop with Variable Iterations	6	0	0
FXD - Mixed fixed and non-fixed datatypes	0	0	0
MDB - Missing Default Branch	0	0	0
NCO - No Contribution to Output	2	0	0
OSA - Optimal Size Accumulator	0	0	0
PDD - Platform dependent datatype (long)	0	0	0
RIU - Rolled loop Inside Unrolled loop	0	0	0
SAT - Sub-optimal Adder Tree	0	0	0
SUD - Suboptimal Use of Divide and Modulus Operator	0	0	0

INFO	Violated	Waived	Undecided
APT - Array Dimension Power of Two	3	0	0
CMC - C style Memory Check	0	0	0
LRC - Long Reset Cycle	0	0	0
MXS - Mixed signed and unsigned datatypes	0	0	0
STF - Funcs with statics called multiple times	0	0	0

Testbench:

```
! Info: HW reset: TLS_rst active @ 0 s
! Info: HW reset: TLS_arst_n active @ 0 s
! Magnitude: Manhattan norm per pixel 5.384757
! Writing algorithmic bitmap output to: out_algorithm.bmp
! Writing bit-accurate bitmap output to: out_hw.bmp
! sofErr: 0 eolErr: 0
! crc32_alg_pix_in  = ebb44e76  crc32_hw_pix_in  = ebb44e76
! crc32_alg_dat_out = 398625ad  crc32_hw_dat_out = 4c0301be
! Finished
! Info: Execution of user-supplied C++ testbench 'main()' has completed with exit code = 0
!
```

```
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        = 57600
   comparison count     = 57600
   ignore count         = 0
   error count          = 0
   stuck in dut fifo    = 0
   stuck in golden fifo = 0
'dat_out_sof'
   capture count        = 57600
   comparison count     = 57600
   ignore count         = 0
   error count          = 0
   stuck in dut fifo    = 0
   stuck in golden fifo = 0
'dat_out_eol'
   capture count        = 57600
   comparison count     = 57600
   ignore count         = 0
   error count          = 0
   stuck in dut fifo    = 0
   stuck in golden fifo = 0
```

三、How to integrate your design in FSIC

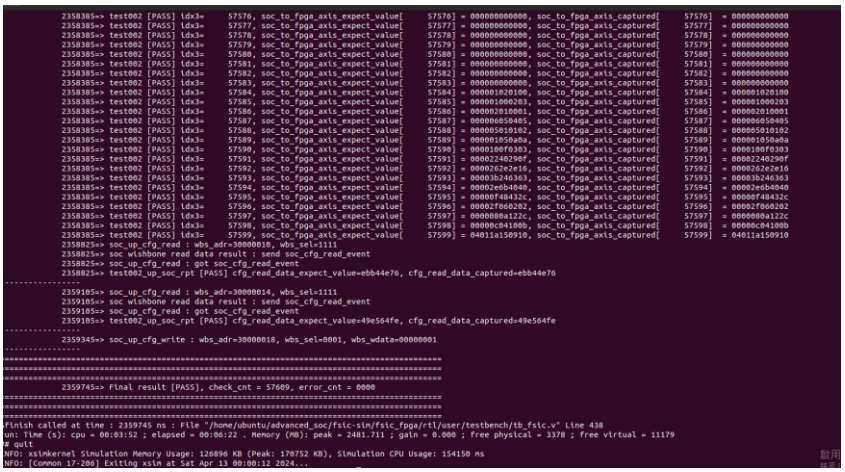
```
module EdgeDetect_IP_EdgeDetect_Top (
  clk, rst, arst_n, widthIn, heightIn, sw_in_rsc_dat, sw_in_triosy_lz, crc32_pix_in_rsc_zout,
      crc32_pix_in_rsc_lzout, crc32_pix_in_rsc_zin, crc32_pix_in_triosy_lz, crc32_dat_out_rsc_zout,
      crc32_dat_out_rsc_lzout, crc32_dat_out_rsc_zin, crc32_dat_out_triosy_lz, dat_in_rsc_dat,
      dat_in_rsc_vld, dat_in_rsc_rdy, dat_out_rsc_dat, dat_out_rsc_vld, dat_out_rsc_rdy,
      line_buf0_rsc_clken, line_buf0_rsc_q, line_buf0_rsc_re, line_buf0_rsc_radr,
      line_buf0_rsc_we, line_buf0_rsc_d, line_buf0_rsc_wadr, line_buf1_rsc_clken,
      line_buf1_rsc_q, line_buf1_rsc_re, line_buf1_rsc_radr, line_buf1_rsc_we, line_buf1_rsc_d,
      line_buf1_rsc_wadr
);
  input clk;
  input rst;
  input arst_n;
  input [9:0] widthIn;
  input [8:0] heightIn;
  input sw_in_rsc_dat;
  output sw_in_triosy_lz;
  output [31:0] crc32_pix_in_rsc_zout;
  output crc32_pix_in_rsc_lzout;
  input [31:0] crc32_pix_in_rsc_zin;
  output crc32_pix_in_triosy_lz;
  output [31:0] crc32_dat_out_rsc_zout;
  output crc32_dat_out_rsc_lzout;
  input [31:0] crc32_dat_out_rsc_zin;
  output crc32_dat_out_triosy_lz;
  input [33:0] dat_in_rsc_dat;
  input dat_in_rsc_vld;
  output dat_in_rsc_rdy;
  output [33:0] dat_out_rsc_dat;
  output dat_out_rsc_vld;
  input dat_out_rsc_rdy;
  output line_buf0_rsc_clken;
  input [63:0] line_buf0_rsc_q;
  output line_buf0_rsc_re;
  output [6:0] line_buf0_rsc_radr;
  output line_buf0_rsc_we;
  output [63:0] line_buf0_rsc_d;
  output [6:0] line_buf0_rsc_wadr;
  output line_buf1_rsc_clken;
  input [63:0] line_buf1_rsc_q;
  output line_buf1_rsc_re;
  output [6:0] line_buf1_rsc_radr;
  output line_buf1_rsc_we;
  output [63:0] line_buf1_rsc_d;
  output [6:0] line_buf1_rsc_wadr;
```

```
EdgeDetect_IP_EdgeDetect_Top U_EdgeDetect (
.clk                    (axi_clk           ), //user_clock2 ?
.rst                    (reg_rst           ),
.arst_n                 (axi_reset_n       ), //~uck2_rst_n ?
.widthIn                (reg_widthIn       ), //I
.heightIn               (reg_heightIn      ), //I
.sw_in_rsc_dat          (reg_sw_in         ), //I
.sw_in_triosy_lz        (),                   //O
.crc32_pix_in_rsc_zout  (crc32_stream_in   ), //O
.crc32_pix_in_rsc_lzout (),                   //O
.crc32_pix_in_rsc_zin   (),                   //O
.crc32_pix_in_triosy_lz (),                   //O, not useful
.crc32_dat_out_rsc_zout (crc32_stream_out  ), //O
.crc32_dat_out_rsc_lzout  (),                  //O
.crc32_dat_out_rsc_zin  (),                  //O
.crc32_dat_out_triosy_lz (edgedetect_done  ), //O
.dat_in_rsc_dat         (dat_in_rsc_dat    ), //I
.dat_in_rsc_vld         (ss_tvalid         ), //I
.dat_in_rsc_rdy         (dat_in_rsc_rdy    ), //O
.dat_out_rsc_dat        (dat_out_rsc_dat   ), //O
.dat_out_rsc_vld        (sm_tvalid         ), //O
.dat_out_rsc_rdy        (sm_tready         ), //I
.line_buf0_rsc_clken       (ram0_en           ), //O
.line_buf0_rsc_q        (ram0_q            ), //I
.line_buf0_rsc_we       (ram0_we           ), //O
.line_buf0_rsc_re       (ran0_we           ),
.line_buf0_rsc_d        (ram0_d            ), //O
.line_buf0_rsc_wadr       (ram0_adr          ), //O
.line_buf0_rsc_radr       (ram0_adr          ), //O
.line_buf1_rsc_clken       (ram1_en           ), //O
.line_buf1_rsc_q        (ram1_q            ), //I
.line_buf1_rsc_we       (ram1_we           ), //O
.line_buf1_rsc_re       (ram1_we           ), //O
.line_buf1_rsc_d        (ram1_d            ), //O
.line_buf1_rsc_wadr       (ram1_adr          ),  //O
.line_buf1_rsc_radr       (ram1_adr          )  //O
);
```

利用 Catapult 合成出 EdgeDetect 的 RTL 電路，並接上 SRAM 及 Control Register，如上圖所示。SRAM 的規格是如題目規定的 640 x 360，Control Register 的部分則是基本參照 Lab1，只有稍微 modify 一些 port 的連接以符合 Catapult 合成的 concat_EdgeDetect.v 的規格。

四、 What’ s the simulation result of FSIC

SW_IN = 1 (Edge Map)



SW_IN = 0 (Orig. Img)