

NYCU-ECE DCS-2024

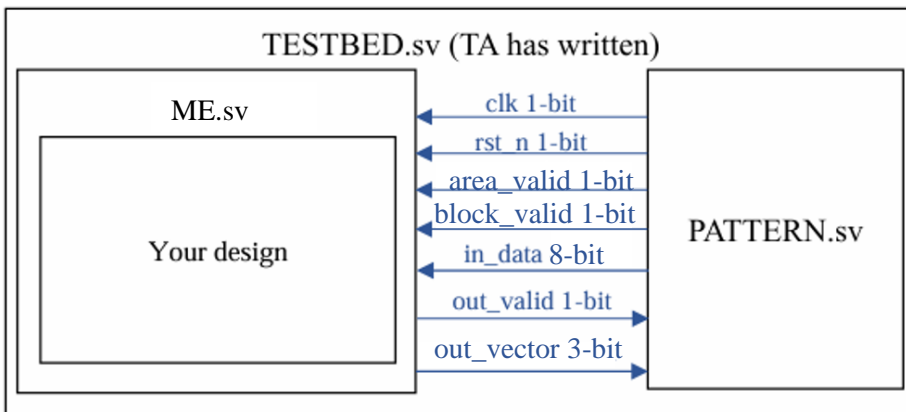
HW04

Design: Motion Estimation

資料準備

1. 從 TA 目錄資料夾解壓縮:
% tar -xvf ~dcsTA01/HW04.tar
2. 解壓縮資料夾 hw04 包含以下:
 - a. 00_TESTBED/
 - b. 01_RTL/
 - c. 02_SYN/
 - d. 03_GATE/

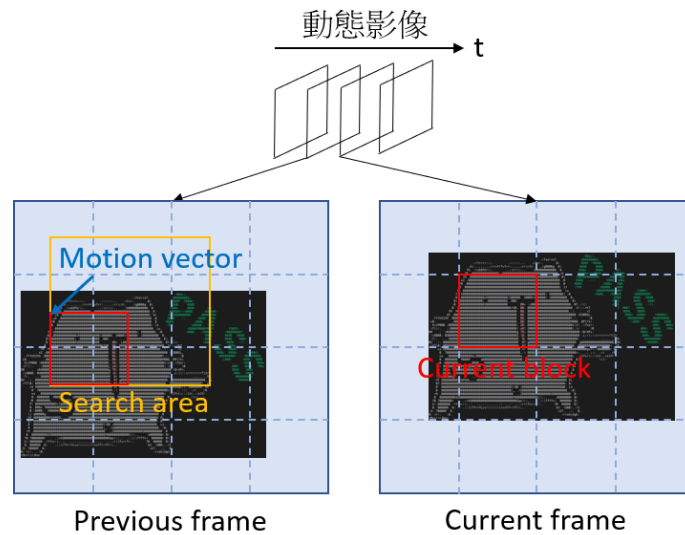
Block Diagram



設計描述

本次作業可以自己調整clock period，詳情請看Specification。

移動估測(motion estimation)是一種動態影像壓縮的技術。以下圖為例，可以看到pass貓貓往右上移動了一小段距離，而背景幾乎沒有變化，這時我們只需要紀錄其相對位置的移動向量，就可以利用前一張畫面來重建當前畫面。



至於如何計算移動向量?本次作業採用的是全域搜尋區塊比對演算法，並利用絕對誤差和(Sum of Absolute Differences, SAD)從Search area找到與Current block最相似的區塊。

絕對誤差和(Sum of Absolute Differences, SAD)的計算公式如下:

$$SAD(u, v) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |a(i, j) - b(i+u, j+v)|$$

這邊我們直接以實例來說明如何找到移動向量:

1. 本次作業的Search area是8*8的二維矩陣，數值範圍0~255。假設如下:

204	208	90	136	237	247	159	219
97	235	245	231	57	226	98	117
77	188	90	241	166	60	249	64
135	71	238	3	17	130	70	92
136	49	103	12	206	83	22	93
46	88	230	32	137	182	44	133
174	4	221	119	178	108	165	27
19	20	17	150	142	58	191	81

2. 本次作業的Current block是4*4的二維矩陣，數值範圍0~255。假設如下:

221	85	103	87
81	181	158	243
68	36	46	179
121	30	127	106

3. 利用SAD做全域搜尋(此動作與convolution中的stride=1動作相同，但運算方式為SAD)後，可以得到5*5的二維矩陣如下:

907	1531	1552	1497	1336
1263	1460	1034	1141	1328
1513	1209	1478	1482	1176
1383	1424	1129	1356	1311
1246	1184	1404	1295	976

4. 左上角的907為最小值，代表與之最接近的區塊，輸出向量(-2, 2)。
5. 各種情況應輸出的向量值如下表:

(-2, 2)	(-1, 2)	(0, 2)	(1, 2)	(2, 2)
(-2, 1)	(-1, 1)	(0, 1)	(1, 1)	(2, 1)
(-2, 0)	(-1, 0)	(0, 0)	(1, 0)	(2, 0)
(-2, -1)	(-1, -1)	(0, -1)	(1, -1)	(2, -1)
(-2, -2)	(-1, -2)	(0, -2)	(1, -2)	(2, -2)

6. 範例中的pass貓貓被切成16個block，以上述方法找到16個移動向量再加上前一張圖片後就可以用以估計當前的圖片。(本次作業僅會給予一組search area和current block計算移動向量)

Pattern:

本次Pattern僅會提供十組測資。Demo測資會是1000筆。均不會存在SAD有複數個最小值的情形。

Inputs

Signal name	Number of bit	Description
clk	1-bit	Clock
rst_n	1-bit	Asynchronous reset
area_valid	1-bit	為1時代表in_data給search area資料，連續給64cycles
block_valid	1-bit	為1時代表in_data給current block資料，連續給16個cycles
in_data	8-bit	連續給80個cycles，先給block再給area，數值範圍: 0~255

Outputs

Signal name	Number of bit	Description
out_valid	1 bits	必須在area_valid變0後300 cycles內拉起，並持續2 cycles
out_vector	3 bits	第一個cycle輸出x向量，第二個cycle輸出y向量，是signed，數值範圍: -2~2

Specifications

1. Top module name: **ME**(File name : **ME.sv**)
2. 請用 **Systemverilog** 完成你的作業。
3. 在非同步負準位reset後，所有的output訊號必須全部歸零。
4. In_valid(包含area_valid和block_valid)和out_valid不可重疊。
5. Out_valid要連續輸出兩個cycle，不能多不能少，且輸出答案要正確。
6. 所有output訊號要在輸出結束後全部歸零。
7. 下一筆測資會在out_valid拉下後一到三個cycle內給予。
8. 測資的輸入方式為raster scan order。
9. 02_SYN result 不行有 **error** 且不能有任何 **latch**也不能有**timing violation**。
10. 03_GATE不能有任何 **timing violation**。
11. 03_GATE的Latency要與01_RTL相同。
12. Input delay = 0.5 * clock period. Output delay = 0.5 * clock period。
13. Clock period 最大10.0ns，以 0.1ns 為單位，例如5.1ns，不要有5.17ns。要更改clock period 要修改兩個地方再去跑02，一個是PATTERN.sv第1行，另一個是syn.tcl 第26行，如下圖所示:

HW04/00_TESTBED/PATTERN.sv 第1行的10.0

```
1  `define CYCLE_TIME 10.0
2  module PATTERN(
3      // Input signals
```

HW04/02_SYN/syn.tcl 第26行的10.0

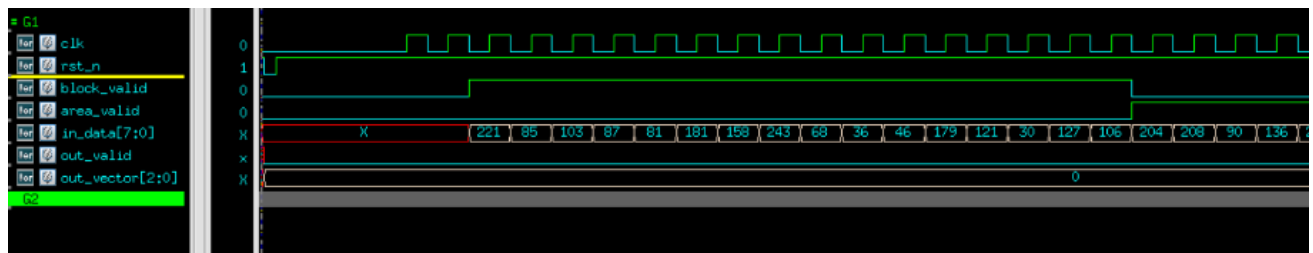
```
24 set DESIGN "ME"
25 #set MAX_Delay 10
26 set clk_period 10.0
27 set IN_DLY [expr 0.5*$clk_period]
28 set OUT_DLY [expr 0.5*$clk_period]
```

14. 若有reset矩陣的需求可使用for loop。

Example waveform

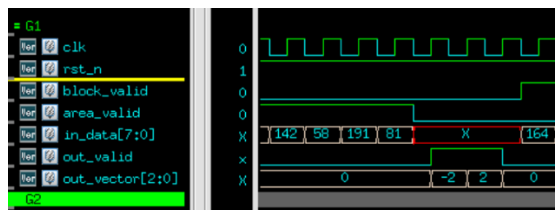
Input

先輸入16個cycle的block data再輸入64個cycle的area data。



Output

out_valid拉起時先輸出x向量再輸出y向量，out_valid拉下時out_vector歸零。



上傳檔案

1. 請將HW04/01_RTL裡的ME.sv依以下命名規則重新命名後上傳至E3。
命名規則：ME_{clock cycle time}_dcsxxx.sv，xxx為工作站帳號號碼, clock cycle time請取到小數第一位。
例如：某同學的工作站帳號為dcs230，clock cycle time為6ns，他的檔名應為ME_6.0_dcs230.sv。命名錯誤扣5分
2. report_dcsxx.pdf, xx is your server account. 上傳至 E3。
3. 1 demo請在 5/2 23:59:59 之前上傳
4. 2 demo請在 5/9 23:59:59 之前上傳
5. Report請在 5/9 23:59:59 之前上傳

Grading policy

1. Pass the RTL& Synthesis simulation. (60%)
2. Performance = Area × Total cycles × Clock period (30%)
3. Report (10%)

Note

Template folders and reference commands:

1. 01_RTL/ (RTL simulation) → **.01_run**
2. 02_SYN/ (synthesis) → **.01_run_dc**
3. 03_GATE/ (GATE simulation) → **.01_run**

報告請簡單且重點撰寫，**不超過兩頁A4**，並包括以下內容

1. 描述你的設計方法，包含但不限於如何加速(減少critical path)或降低面積。
2. 基於以上，畫出你的架構圖(Block diagram)
3. 心得報告，不侷限於此次作業，對於作業或上課內容都可以寫下。