NYCU-ECE DCS-2023

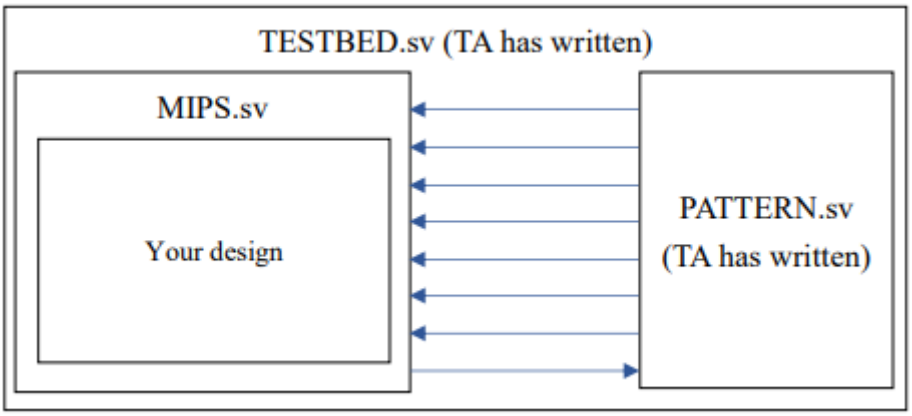
HW05

Design: MIPS CPU

資料準備

1. 從 TA 目錄資料夾解壓縮:
   % tar -xvf ~dcsta01/ HW05.tar
2. 解壓縮資料夾 hw05 包含以下:
   a. 00_TESTBED/
   b. 01_RTL/
   c. 02_SYN/
   d. 03_GATE/
   e. 09_UPLOAD/

Block Diagram



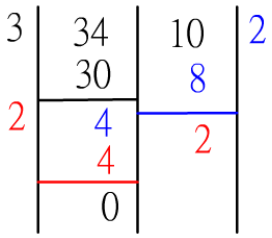
設計描述

此次作業目的是設計一個CPU執行簡單運算並使用pipeline，可以連續執行序列 的Instruction。MIPS是一種指令集，內含有許多指令，此次作業以MIPS指令集做稍微修改。

Type	Instruction 32 bits					
R	Opcode (6 bits)	Rs (5 bits)	Rt (5 bits)	Rd (5 bits)	Shamt (4 bits)	Funct (7 bits)
	Opcode	Funct		Operation		
	000000	0100000		Rs + Rt		
		0100100		Rs and Rt		
		0100101		Rs or Rt		
		0100111		Rs nor Rt		
		0000000		Rt shift 向左 shamt bits		
		0000010		Rt shift 向右 shamt bits		
		1111000		GCD (Rs, Rt) (Rs, Rt 的最大公因數)		
Type	Instruction 32 bits					
I	Opcode (6 bits)	Rs (5 bits)	Rt (5 bits)	Immediate (16 bits)		
	Opcode	Operation				
	001000	Rs + imm (16bits)				

Hint: 求最大公因數可以使用輾轉相除法，原理是兩個數字互相減來減去，最後就會剩下構成兩個數字的共通單位，也就是最大公因數。

Ex. GCD (34,10) = 2



Rs: source register address, Rt: target register address

Rd: destination register address, imm: value

當遇到以下情況時會發生instruction fail:

(1) Opcode不在列表中

(2) R type指令的Funct不在列表中

(3) R type指令的Rs, Rt, Rd不存在暫存器的Address列表中 (雖然shift指令沒有使用到Rs，但是仍要檢查)

(4) I type指令的Rs, Rt不存在暫存器的Address列表中

(5) 在進行最大公因數的計算時，若Rs, Rt的value任一個為0，無法計算

當instruction fail發生時，out_1、out_2、out_3、out_4皆須設置為零，並且將intruction_fail 設為1。

除了Instruction外，還有兩個input : in_valid、output_reg

in_valid為high時代表instruction與output_reg開始給值，design要開始取值。output_reg代表最後output的四個signal需要取的是哪四個暫存器，給的值是address， 並且分別將暫存器的value給out_1、out_2、out_3、out_4，如下表所示。

[19:0]output_reg	[19:15]	[14:10]	[9:5]	[4:0]
	out_4	out_3	out_2	out_1

暫存器需自行宣告，各暫存器的位址如下表。

Address(5 bits)	Value(16 bits) 初始值皆為零
10001	0
10010	0
01000	0
10111	0
11111	0
10000	0

R、I type有寫回功能，分別用Rd/Rt表示寫回位址，須將值存起來進行下次計算。以下為一些指令例子：

1. Instruction : 001000-10001-01000-0000000011001000

Opcode-rs-rt-imm

Addi instruction, reg(10001) = 0，imm = 200, reg(01000) = 0 + 200 = 200

2. Instruction : 001100-10001-10010-0000000000000100

   NO opcode 001100 -> instruction fail !

3. Instruction : 000000-10001-01000-10000-0100-0000010

   Opcode-rs-rt-rd-shamt-funct

   Srl instruction, reg(01000) = 200, shamt = 4, reg(10000) = 12

4. Instruction : 000000-01000-10000-10001-1111-1111000

   Opcode-rs-rt-rd-shamt-funct

   GCD instruction, reg(01000)=200, reg(10000)=12, reg(10001)=GCD(200,12)=4

可以從以上範例來得知，當第一行指令執行完後，reg(01000)值變為200，而第三行指令又拿取reg(01000)的值，這次會延續第一行的結果拿到200。

這次作業助教只會提供十筆測資，同學們需要自行產生測資以驗證自己的設計，可 以更改input.txt/output.txt的檔案或改寫PATTERN.sv，demo時助教會使用不只十筆測資。

input.txt 範例

```
10
00100010001010000000000011001000 10001100100100010001
00110010001100100000000000000100 10001010001111110000
00000010001010001000001000000010 10001100101111101000
00000001000100001000111111111000 10010101111111110000
00000011111110111011110101000000 10010111111011111111
00000010001100101011110111111000 10001100101011110000
00000001000101111111110010100100 10001010001011111111
00100010001100100000000000001001 10010010001011111111
00000010111111111000000111111111 10010010001011111111
00000001000101111111110000100101 10001100011000110000

```

第一行是測資數量，第二行開始依序為instruction、output_reg

output.txt範例

```
0       0       200     0       0
1       0       0       0       0
0       200     0       0       0
0       12      0       0       0
1       0       0       0       0
1       0       0       0       0
0       0       0       200     4
0       0       0       200     13
1       0       0       0       0
0       12      4       4       4
```

順序為：instruction_fail、out_1、out_2、out_3、out_4

Inputs

Signal name	Number of bit	Description
clk	1 bit	Clock
rst_n	1 bit	Asynchronous active-low reset
in_valid	1 bit	When getting high, means start giving instruction
instruction	32 bits	In_valid 為 1 的時候給值
output_reg	20 bits	指定輸出的四個 output signal 分別為哪四個 register

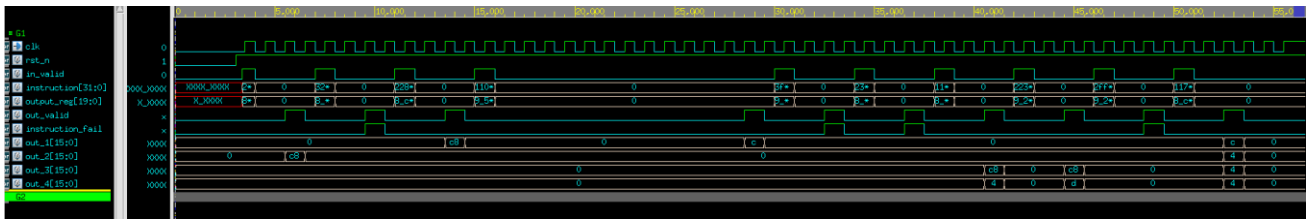
Outputs

Signal name	Number of bit	Description
out_valid	1 bit	輸出有值的時候為 high
out_1	16 bits	Address 是 output_reg[4:0]的暫存器數值，instruction 無效時為零。
out_2	16 bits	Address 是 output_reg[9:5]的暫存器數值，instruction 無效時為零。
out_3	16 bits	Address 是 output_reg[14:10]的暫存器數值，instruction 無效時為零。
out_4	16 bits	Address 是 output_reg[19:15]的暫存器數值，instruction 無效時為零。
instruction_fail	1 bits	instruction 無效時為 1，有效為 0。

Specifications

1. Top module name: **MIPS**(File name : **MIPS.sv**)
2. 在非同步負準位 reset 後，所有的 output 訊號必須全部歸零。
3. Output 要在 Input 結束後的 100 cycles 內輸出。
4. 所有 Output 訊號要在輸出結束後全部歸零。
5. 02_SYN result 不行有 error 且不能有任何 latch。
6. Clock period 10 ns。
7. Input delay = 0.5 * clock period; Output delay = 0.5 * clock period

Example waveform



上傳檔案

1. Code使用09_upload上傳。
2. report_dcsxx.pdf, xx is your server account. 上傳至new E3。
3. 1 demo請在 5/11 23:59:59 之前上傳
4. 2 demo請在 5/18 23:59:59 之前上傳

Grading policy

1. Pass the RTL& Synthesis simulation: 70%
2. Performance: 20%
   Ranking formula: total latency * area
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-level simulation) → **./01_run**
4. 09_UPLOAD/ (upload) → **./09_upload**

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。
4. 遇到的困難與如何解決。