

NYCU-ECE DCS-2024

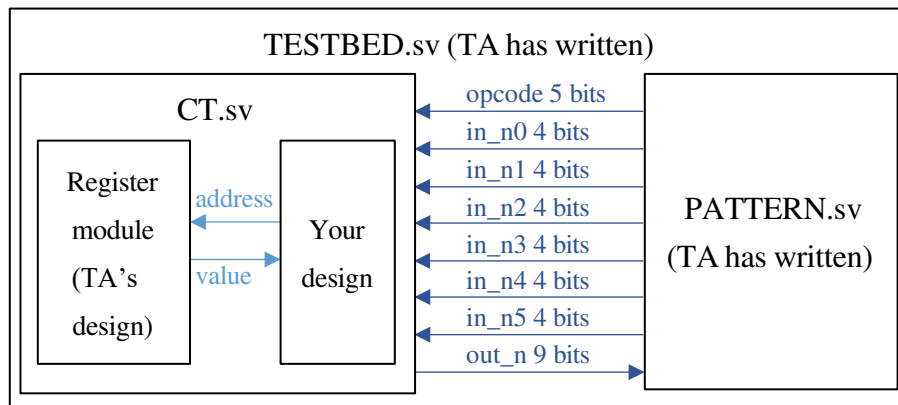
HW01

Design: Calculation Task

資料準備

- 從 TA 目錄資料夾解壓縮:
`% tar -xvf ~dcsTA01/HW01.tar`
- 解壓縮資料夾 hw01 包含以下:
 - 00_TESTBED/
 - 01_RTL/
 - 02_SYN/
 - 03_GATE/
 - 09_UPLOAD/

Block Diagram



設計描述

本次作業目標設計對輸入數字進行解碼、排序並做些指定的運算。

你會接收到一個opcode和六個數字{in_n0, in_n1, in_n2, in_n3, in_n4, in_n5}，opcode會決定對六個數字的運算方式，最後輸出運算後的數字out_n。

首先收到的六個數字{in_n0, in_n1, in_n2, in_n3, in_n4, in_n5}是register(暫存器)的地址，並非資料的值。本次作業助教會提供register file的module，你們必須透過module connection(name mapping)，將暫存器與你的design連接，以取得暫存器內的資料值。

暫存器的位置及內容如下:

Address(4 bits)	0000	0001	0010	0011	0100	0101	0110	0111
Value(5 bits)	9	27	30	3	11	8	26	17
Address(4 bits)	1000	1001	1010	1011	1100	1101	1110	1111
Value(5 bits)	3	12	1	10	15	5	23	20

六個 Address{in_n0, in_n1, in_n2, in_n3, in_n4, in_n5} 會對應到六個數值 {value_0, value_1, value_2, value_3, value_4, value_5}，**注意取出的資料是5 bits**。例如: {10, 9, 15, 2, 12, 0} 會對應到數值 {1, 12, 20, 30, 15, 9}，接下來由 opcode 決定這六個數值要進行的排列與運算。

接下來會先**排序再運算結果**，opcode 為 5 bits 可以分成兩個部分，會決定排序與運算的結果，兩個部分的控制如下：

1. Sorting	<p>如果 opcode[4:3] 為 11，這六個數值由小到大進行排列。 例如: {1, 12, 20, 30, 15, 9} → {1, 9, 12, 15, 20, 30}</p> <p>如果 opcode[4:3] 為 10，這六個數值由大到小進行排列。 例如: {1, 12, 20, 30, 15, 9} → {30, 20, 15, 12, 9, 1}</p> <p>如果 opcode[4:3] 為 01，這六個數值順序為原本的相反。 例如: {1, 12, 20, 30, 15, 9} → {9, 15, 30, 20, 12, 1}</p> <p>如果 opcode[4:3] 為 00，這六個數值維持原本的順序。 例如: {1, 12, 20, 30, 15, 9} → {1, 12, 20, 30, 15, 9}</p>
2. Calculation	<p>假設排序後的數列為 {num0, num1, num2, num3, num4, num5}</p> <p>如果 opcode[2:0] 為 000，進行 $\mu = \sum_{i=0}^5 \frac{num_i}{6}$ 例如: {1, 12, 20, 30, 15, 9} → $\mu = 87 / 6 = 14$ (round down) 並以 μ 當作及格分數判斷及格人數 → out_n = 3</p> <p>如果 opcode[2:0] 為 001，進行 out_n = num0 + num5 例如: {1, 12, 20, 30, 15, 9} → out_n = 1 + 9 = 10</p> <p>如果 opcode[2:0] 為 010，進行 out_n = (num3 * num4) / 2 例如: {1, 12, 20, 30, 15, 9} → out_n = (15 * 20) / 2 = 150</p> <p>如果 opcode[2:0] 為 011，進行 out_n = num0 + (num2 * 2) 例如: {1, 12, 20, 30, 15, 9} → out_n = 1 + (12 * 2) = 25</p>

	<p>如果 opcode[2:0] 為 100，進行 out_n = num1 & num2 例如: {1, 12, 20, 30, 15, 9} → out_n = 01001₍₂₎&01100₍₂₎ $= 01000_{(2)}$</p> <p>如果 opcode[2:0] 為 101，進行 out_n = ~num0 例如: {1, 12, 20, 30, 15, 9} → out_n = ~00001₍₂₎=11110₍₂₎</p> <p>如果 opcode[2:0] 為 110，進行 out_n = num3 ^ num4 例如: {1, 12, 20, 30, 15, 9} → out_n = 01111₍₂₎^10100₍₂₎ $= 11011_{(2)}$</p> <p>如果 opcode[2:0] 為 111，進行 out_n = num1 << 1 例如: {1, 12, 20, 30, 15, 9} → out_n = 01001₍₂₎<<1 $= 10010_{(2)}$</p>
--	---

※當opcode[2:0]=000/010時，除法為無條件捨去，Ex: 15/2 = 7, 391/2 = 195, etc.

最後將會輸出一個9-bit的數字out_n，testbench測試pattern將會在下一組測資進來之前測試這9-bit output。

Inputs

Signal name	Number of bit	Description
opcode	5 bits	<p>opcode[4:3]決定六個數值的排序方式。 opcode[4:3] = 00, 維持原順序， opcode[4:3] = 01, 順序相反 opcode[4:3] = 10, 由大到小 opcode[4:3] = 11, 由小到大</p> <p>假設排序後的數列為 {num0, num1, num2, num3, num4, num5}</p> <p>opcode[2:0]決定排序後數值的運算。 opcode[2:0] = 000, average & compare opcode[2:0] = 001, num0 + num5 opcode[2:0] = 010, (num3 * num4) / 2 opcode[2:0] = 011, num0 + (num2 * 2) opcode[2:0] = 100, num1 & num2 opcode[2:0] = 101, ~num0 opcode[2:0] = 110, num3 ^ num4 opcode[2:0] = 111, num1 << 1</p>

in_n0	4 bits	範圍為 0~15 的隨機正整數。
in_n1	4 bits	
in_n2	4 bits	
in_n3	4 bits	
in_n4	4 bits	
in_n5	4 bits	

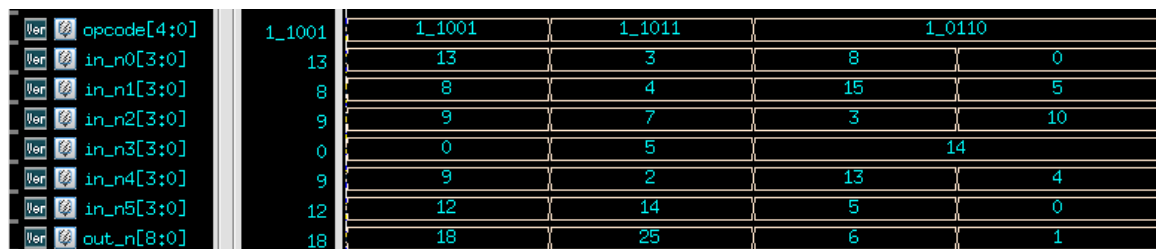
Outputs

Signal name	Number of bit	Description
out_n	9 bits	運算後的答案，範圍為 0~511 的正整數。

Specifications

1. Top module name: **CT**(File name : **CT.sv**)
2. 請用 **Systemverilog** 完成你的作業。
3. 請用 **combination circuit** 完成你的作業。
4. 請用助教給予你的 **register module** 拿資料。
5. 02_SYN result 不行有 **error** 且不能有任何 **latch**。

Example waveform



Signal	Value	Signal	Value	Signal	Value	Signal	Value
opcode[4:0]	1_1001	1_1001	13	1_1011	3	1_0110	8
in_n0[3:0]	13	in_n1[3:0]	8	in_n2[3:0]	9	in_n3[3:0]	0
in_n4[3:0]	9	in_n5[3:0]	12	out_n[8:0]	18		

上傳檔案

1. Code使用09_UPLOAD/01_upload上傳。
2. report_dcsxx.pdf, xx is your server account. 上傳至new E3。
3. 1de 請在 3/21 16:30 pm 之前上傳 / 2de 請在 3/28 16:30 pm 之前上傳

Grading policy

1. Pass the RTL& Synthesis simulation. 70%
2. Area 15%
3. Report 15%

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**
4. 09_UPLOAD/ (upload) → **./01_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。