

# NYCU-ECE DCS-2024

## HW01

### Design: Calculation Task

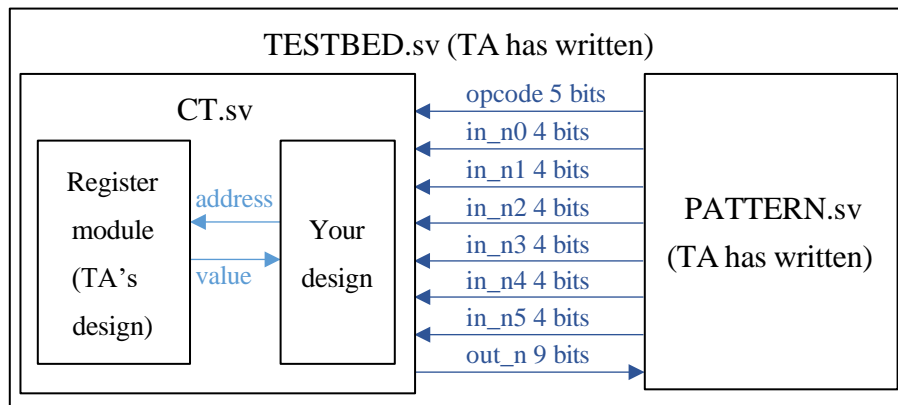
#### 資料準備

---

- 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:

<b>Address(4 bits)</b>	0000	0001	0010	0011	0100	0101	0110	0111
<b>Value(5 bits)</b>	9	27	30	3	11	8	26	17
<b>Address(4 bits)</b>	1000	1001	1010	1011	1100	1101	1110	1111
<b>Value(5 bits)</b>	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 <b>opcode[4:3]</b> 為 <b>11</b>，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 <b>opcode[4:3]</b> 為 <b>10</b>，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 <b>opcode[4:3]</b> 為 <b>01</b>，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 <b>opcode[4:3]</b> 為 <b>00</b>，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 <b>opcode[2:0]</b> 為 <b>000</b>，進行 <math>\mu = \sum_{i=0}^5 \frac{num_i}{6}</math> 例如: {1, 12, 20, 30, 15, 9} → <math>\mu = 87 / 6 = 14</math> (round down) 並以 <math>\mu</math> 當作及格分數判斷及格人數 → out_n = 3</p> <p>如果 <b>opcode[2:0]</b> 為 <b>001</b>，進行 <b>out_n = num0 + num5</b> 例如: {1, 12, 20, 30, 15, 9} → out_n = 1 + 9 = 10</p> <p>如果 <b>opcode[2:0]</b> 為 <b>010</b>，進行 <b>out_n = (num3 * num4) / 2</b> 例如: {1, 12, 20, 30, 15, 9} → out_n = (15 * 20) / 2 = 150</p> <p>如果 <b>opcode[2:0]</b> 為 <b>011</b>，進行 <b>out_n = num0 + (num2 * 2)</b> 例如: {1, 12, 20, 30, 15, 9} → out_n = 1 + (12 * 2) = 25</p>

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

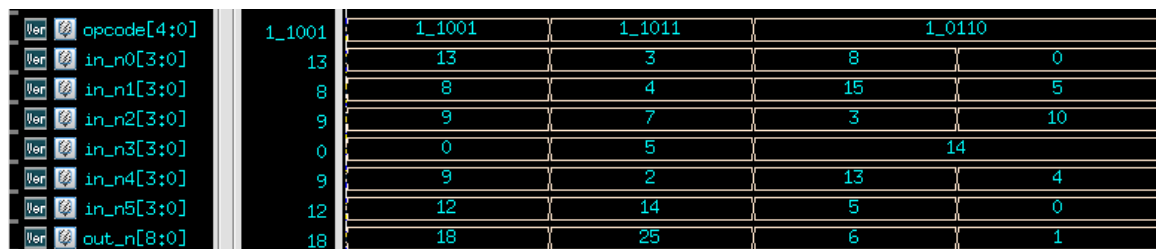
## Outputs

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。