Computer Architecture
Verilog – ALU Design

TA: 鄧傑方 (D06943020@ntu.edu.tw)

1. 作業目標

從 ALU 基本運算單元的例子，讓大家熟悉如何完成 RTL level 的電路敘述，並在工作站上利用 simulator (NC-verilog) 配合 testbench 跑 RTL-level simulation。配合 Notes 以及 Appendix 講解設計硬體時，與 Verilog Coding (Coding Style)要注意的地方。

2. ALU 介紹

Arithmetic logic unit (ALU)是在電腦處理器之中其中一個組成單元，ALU 有數學、邏輯、還有一些設計過的運算在電腦之中。請設計下表中 specification **8-bit ALU with I/O register**，**3-bit instruction set** 的電路。

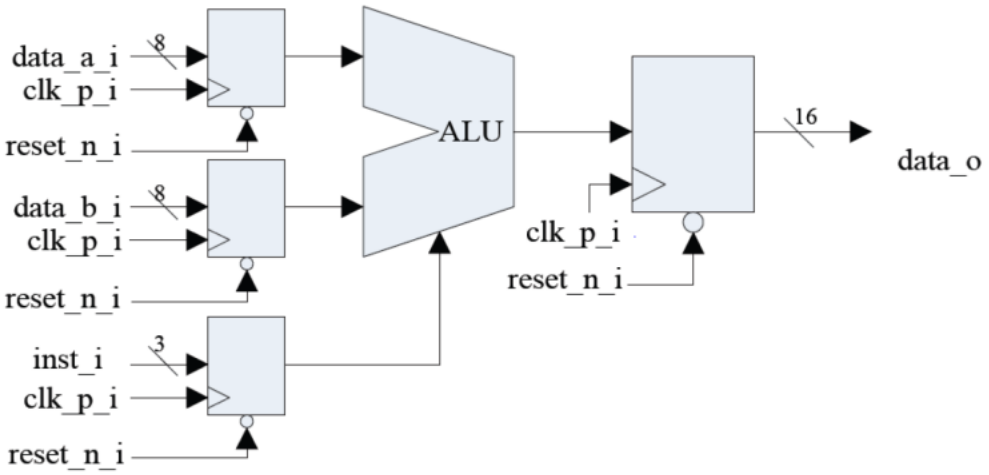
Inst_i [2:0]	Operation	Notes
000	Signed Addition	data_o = data_a_i + data_b_i
001	Signed Subtraction	data_o = data_b_i - data_a_i
010	Unsigned Multiplication	data_o = data_a_i * data_b_i
011	AND	data_o = data_a_i **&** data_b_i
100	XOR	data_o = data_a_i⊕data_b_i
101	Absolute Value	data_o = | data_a_i |
110	Addition & Divide by 2	data_o = (data_b_i + data_a_i) / 2
111	Mod	data_o=(data_b_i **%** data_a_i)

[Notes] 數字系統與訊號之間的注意事項

- Output signal "data_o"有 16bits 並且用 **2's complement** 表示。
- Input data(a & b)對於 instruction 010/011/100/110/111 為 **unsigned**，但對於 instruction 000/001/101 是 **signed**。
- 此次作業不需要考慮 **overflow** 的問題，也就是 output 訊號 16 bit 對於運算結果夠用。
- 對於訊號在特定數字系統(2's complement)下運算前後是需要被注意的
  - 做 zero extension 當 output 永遠為正(instruction 010/101/110/111)
  - 做 **sign-extension** 當 output 有可能為負的(instruction 000/001)

- 做 zero extension 當”AND”和”XOR”運算

- 如果是 fix point 的運算，則需要注意精準度與面積間的拿捏，也就是總 bit 數與小數點的位置(shift bit)選取，可以在整數部分不會 overflow 且小數部分精確度夠情況下使 bit 數最少(面積)。(此次作業無須注意)



1. 作業提供了一個未完成的RTL template program，請依照上述spec.表完成程式，為了加速debug的過程，建議每完成一個instruction就要模擬一次。在testbench之中:
   - If assign “test_all_ins” = 1, 則所有的instruction會被測試(default setting)
   - If assign “test_all_ins” = 0, 則指定特定的"test_instruction"可以測試相對應的instruction

   (ex: test_all_ins = 0, test_instruction = 000 : 只測試 signed addition 功能)

Filename	Description
HW1_alu.v	Unfinished Verilog RTL template
HW1_test_alu.v	Testbench for ALU design

2. 請利用提供的testbench來驗證你的ALU with I/O registers設計，我們會依照testbench對你的design評分。

3. 對於000到001的instruction為基本運算，而101到111為較進階的運算。

3. 繳交要求

請依照下面的檔案命名方式將檔案放在資料夾中：

資料夾名稱: StudentID_HW1_alu

StudentID_HW1_alu.v (*e.g.*, R05943001_HW1_alu.v).

並壓縮成.zip檔 (*e.g.*, R05943001_HW1_alu.zip).

[Notes] 在你的design之中，請依照原本提供Verilog template的port name以及port order，否則你的design可能會在測試中fail進而影響成績。

4. **繳交期限：05/22(三) 22:00**
5. **Appendix**

[Naming Rules]

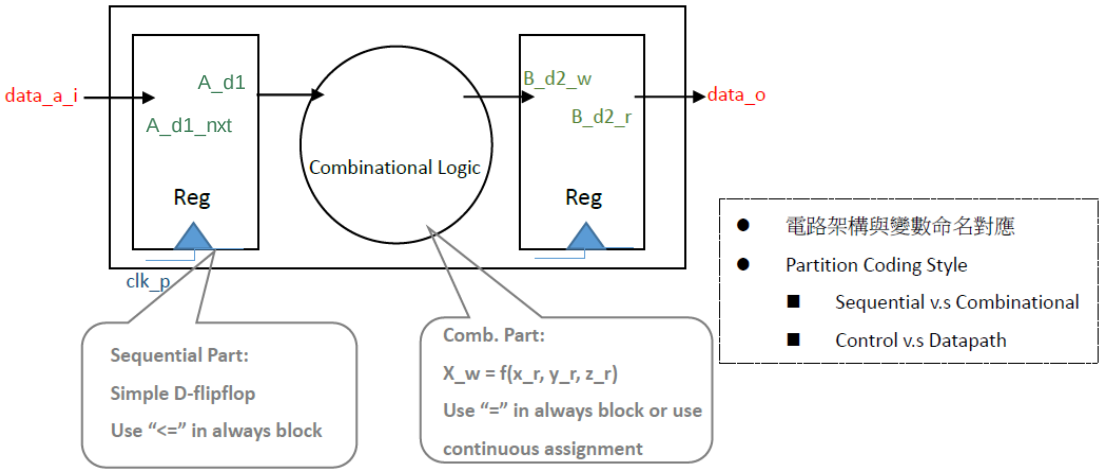
一個好的有共識 Naming Rules 會對 Code 維持性以及可讀性大大提升，最重要的是可以跟硬體架構精準的相對應，可以參考以下所提供的命名規則來對 design 內變數進行命名。

1. "_i"與"_o"應該被加在一個 module 中 input 以及 output 變數的最後面。
2. 如果是 clk 或著 reset signal，則"_p" or "_n"可以用來表示他是正緣觸發還是負緣處發。
3. 如果變數在電路中為 flip-flop，則在 reg 變數名稱後
   - Flip-flop input 加"_w"，表示是 write；Flip-flop output 加"_r"，表示是 read。
   - Flip-flop input: a_nxt；Flip-flop output: a

(注意這裏強調為 flip-flop 不是 reg，因為只要是宣告在 block 內的變數都要是 reg 型態，所以有可能實際上是接線還是宣告為 reg，但 block 外的變數必須宣告為 wire 且一定也是接線)

4. 隨著系統內 pipeline 的層數，訊號會 delayx 個 cycle，可以在之後加"_dx"已表示為在第 x 層的訊號。ex: a_d1_w(第一層的 flip-flop input)
5. 如果是不得已對於 a_w 中間操作要用到 truncate(例如 fix-point operation)，因語法限制無法一行寫完則 combinational circuit 內中間 wire 可命名為 tmp 再接給 a_w，如：

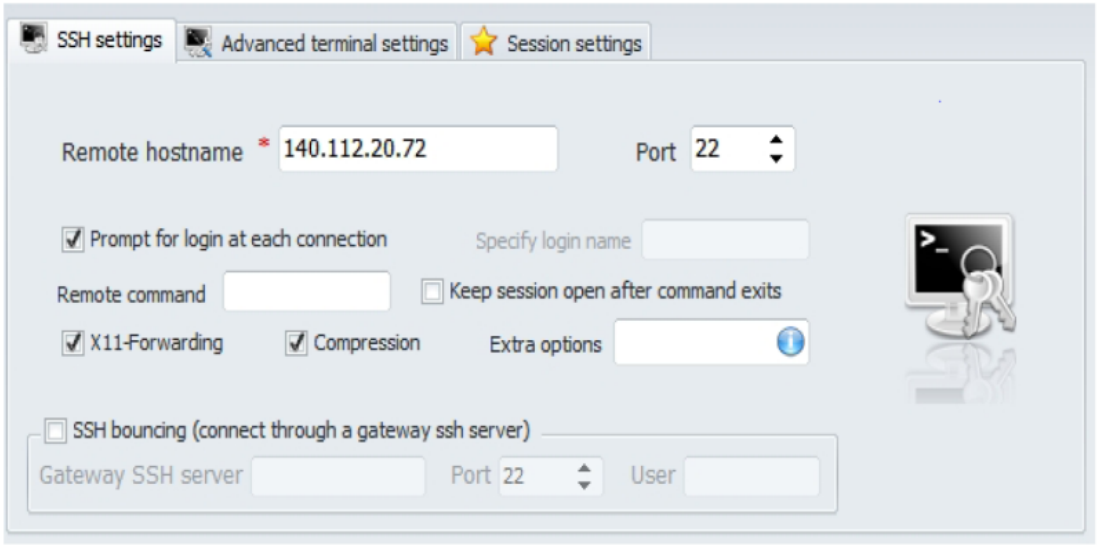
```
a_tmp = b_r + c_r;
a_w = a_tmp[5:3];
```

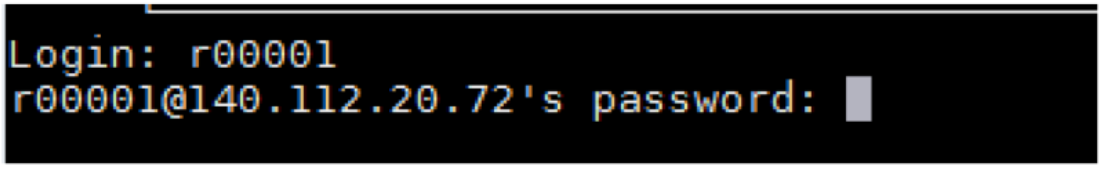


6. [Run NC-Verilog Simulation on GIEE's Workstation]

1. Connect to GIEE's Workstation: [Take MobaXterm for illustration]

Available workstation list: http://cad.ee.ntu.edu.tw/wordpress/?p=33





2. Environment Setup: [type following commands in the console]
   1. Making Directory: mkdir HW1
   2. Enter the Directory: cd HW1
   3. Source the default file: source /usr/cadence/cshrc

3. Check Verilog Code by NC-Verilog
   Type this command: ncverilog [`design_filename]
   Ex: ncverilog HW1_alu.v

4. Run Simulation with a test bench
   Type this command: ncverilog [`testbench_filename] [`design_filename]
   Ex: ncverilog HW1_test_alu.v HW1_alu.v
   PS: 從大 module 依序到小 module，如果未來有 tsmc13.v 則要加在最後面讓 simulator 可以將 foundry standard cell 電路 module 轉成看得懂的 basic behavior block (ex: and gate)，又或著是 include memory、IO pad 額外非 standard cell module 的 external-IP behavior model。

5. If you want to check the waveform, (原則上該次作業不需看波型即可完成)
   Type the command: ncverilog [`testbench_filename] [`design_filename] +access+r
   Ex: ncverilog HW1_test_alu.v HW1_alu.v +access+r
       nWave &

For more GIEE's workstation information, please refer to:
http://cad.ee.ntu.edu.tw/wordpress/
For more workstation commands, please refer to:
http://linux.vbird.org/linux_basic/redhat6.1/linux_06command.php