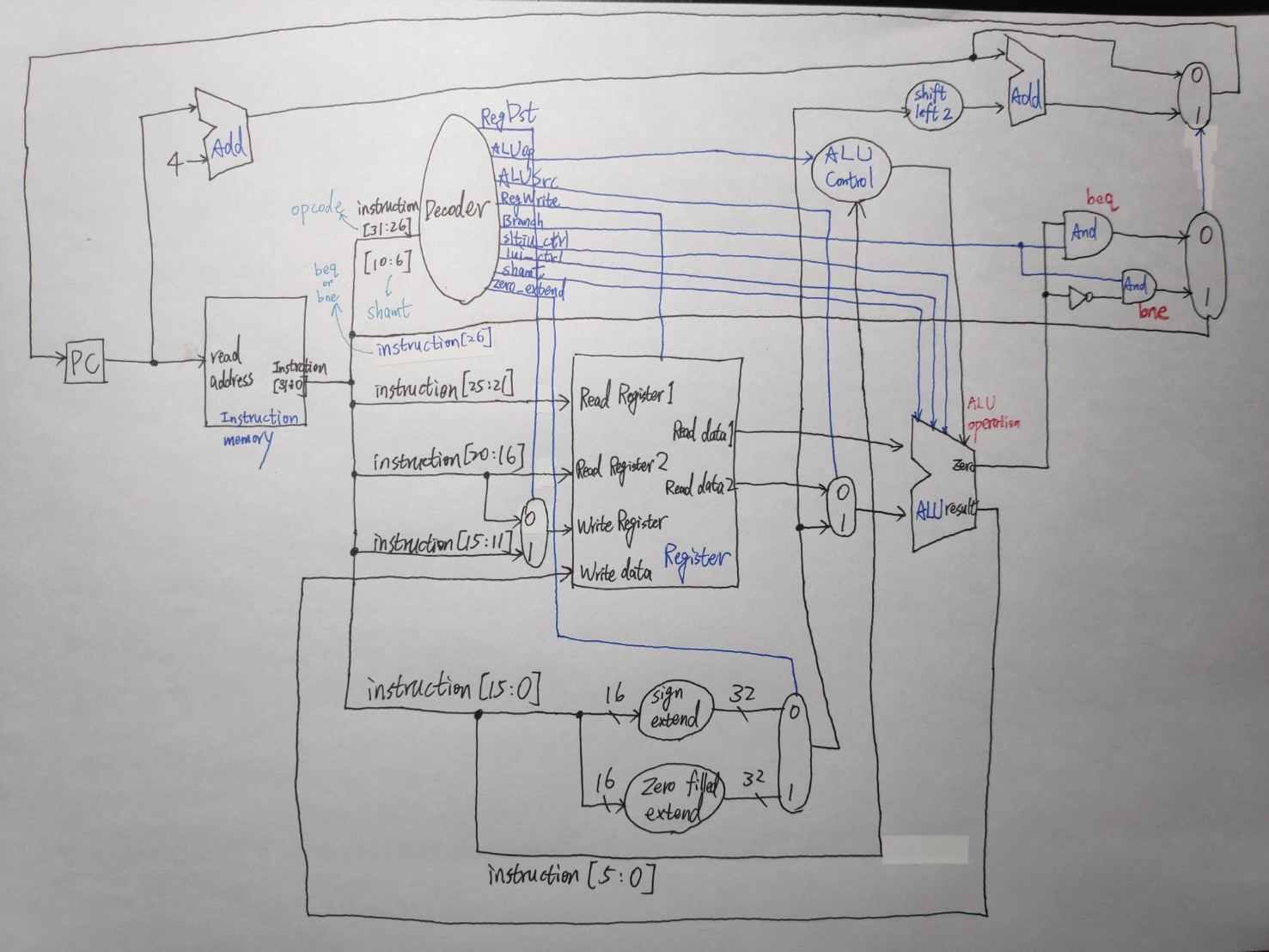
## Term Project - Lab 2(Simplified single cycle CPU) Report

Architecture diagram:



Module description:

Adder: 給兩個input值，並相加出一個output值。用於兩個地方。1. PC每次+4。2. 算出Branch要跳到的address。

ALU (include alu\_top, alu8bits, alu32bits, CLA4bits, CLA8bits): 執行and, or, add, sub, slt, shift等主要運算。主要有兩個input值，Data1和Data2(R-type)或Data1和signed-extension過後的instruction[15:0](I-type)。利用Decoder的各種output決定要執行哪種運算(包含sltiu\_ctrl, lui\_crtl, shamp)。若是其他R-type instructions則用ALU\_Control的output來決定。

ALU\_Ctrl: 輸入ALUop和instruction[5:0](function field)並輸出ALU control input給ALU，讓ALU知道它該做甚麼。

Decoder: 輸入instruction[31:26](opcode)和shamt(right shift bits, sra, srav才會用到)，輸出各個指令需要用到的值。包含RegDst, ALUop, ALUSrc, RegWrite, Branch, sltiu\_ctrl, lui\_ctrl, shamt, zero\_extend。並接到需要用到的地方。

Instr\_Memory: 讀進PC address，輸出32bits instruction。

MUX\_2to1: 兩個input值，透過一個control值來決定要選哪個input值來output。需要選擇時就需要MUX。此lab有五個地方會用到。

ProgramCounter: 輸出PC值。

Reg\_File: 讀取Rs, Rt的值，寫入Rd(有需要的話)。

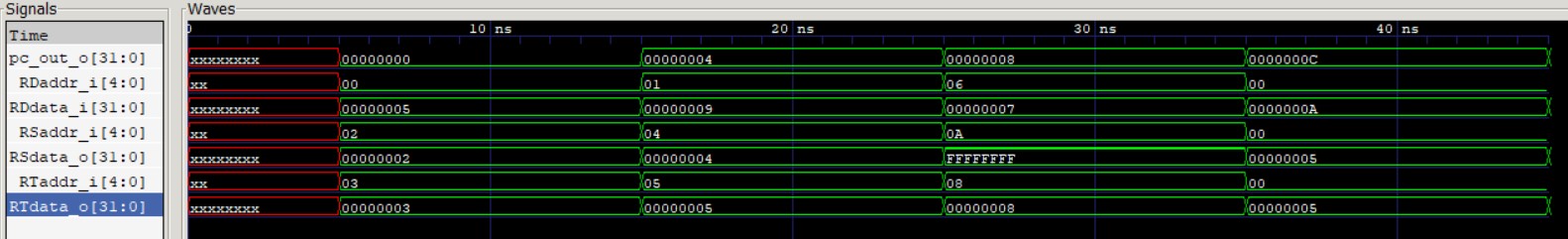
Shift\_Left\_Two\_32: 32bits的input值左移兩位(x4)。

Sign\_Extend: 把16 bit的input 做sign extension成32 bit。

Zero\_Filled\_Extend: 16 bit的input前面填16個0擴充成32 bit。

Simple\_Single\_CPU: 將上述module依照上圖diagram接線。

Waveform:



Questions:

• What is the difference between "input [15:0] input\_0" and "input [0:15] input\_0" inside the module?

Ans: 前者最左邊為 Most significant bit ，後者最右邊為Most significant bit。例如:

output [3:0] A; // A為4bit，A[3]，A[2]，A[1]，A[0]

output [0:3] B; // B為4 bit，B[0]，B[1]，B[2]，B[3]

• What is the meaning of "always" block in Verilog?

Ans: 以always為主的程式區塊，只有每當觸發條件成立時，執行一次，執行完後需要等待下一次的觸發條件成立才會再次執行。

• What are the advantages and disadvantages of port connection by

order and port connection by name in Verilog?

Ans:

Port connection by order:

Advantages: 不需要列出全部port name，只要確認port的順序。

Disadvantage: 可讀性較差，port 順序有變化就容易出錯。

Port connection by name:

Advantage: 可讀性較佳，不需要確認順序，只需確認port name是否正確。

Disadvantage: 需要花費大量時間打出每個port name。

Contribution:

討論各種實作方法、各個module實作、CPU接線、debug … …。每個module都是共同討論並實作，無明顯分工。

Discussion:

建議老師或助教先教一點基本的verilog語法與實作範例。否則有些對verilog不熟的同學會做得蠻辛苦的。另外，實作上我們用decoder輸出額外的控制元(sltiu\_ctrl, lui\_crtl, shamt)來執行sltiu, lui, sra, srav。