

Computer Organization

Hardware module analysis(簡短解釋一下每個 module 的功能，以及在 CPU 裡扮演的角色):

Program Counter	存放下個待執行指令的位址
Adder	32-bit 加法，用在 address 的計算
Instr_Memory	input 來自 PC，讀取 instruction 並輸出
MUX	選擇輸出，一在選 rt，一在選 ALU_Src
SE	Extend 16-bit to 32-bit
Shift left	將輸入左移 2bit (還原位址)

Decoder 根據輸入的 Opcode 輸出對應 control	ALU_op	ALU_Src	RegWrite	RegDst	Branch
R-type(op=0)	010	0	1	1	0
addi (op=8)	000	1	1	0	0
slti (op=10)	110	1	1	0	0
beq (op=4)	001	0	0	0	1

	Op	Function	ALU_op	ALU_ctrl 接收 instruction 的 function code 及 Decoder 送出的 ALUOp，輸出控制 信號給 ALU	ALU 根據 ALU Control 對兩個輸入值做 add,sub,and,or,slt 等基本運算
add	0	32(0x20)	010	0010	add
sub		34(0x22)		0110	sub
and		36(0x24)		0000	and
or		37(0x25)		0001	or
slt		42(0x2a)		0111	slt
addi	8	0	000	0010	add
slti	10		110	0111	slt
beq	4		001	0110	sub

Problems you met and solutions:

1.

Decoder 送出的 ALU_op 有三 bits 和課本不一樣，所以本來 addi 跟 slti 的 ALU_op 設錯。不過 ALU_op 也對應正確的 ALU_control 所以好像沒差。

2.

沒注意到要在 Simple_CPU 輸入訊號，然後 simulation 按成 implementation

Summary:

經過這次的作業才比較清楚 top module 在幹嘛，而在裡面宣告 internal signal 可以連接不同 module。另外也再複習一遍 MIPS 指令的各種控制訊號值