

Report

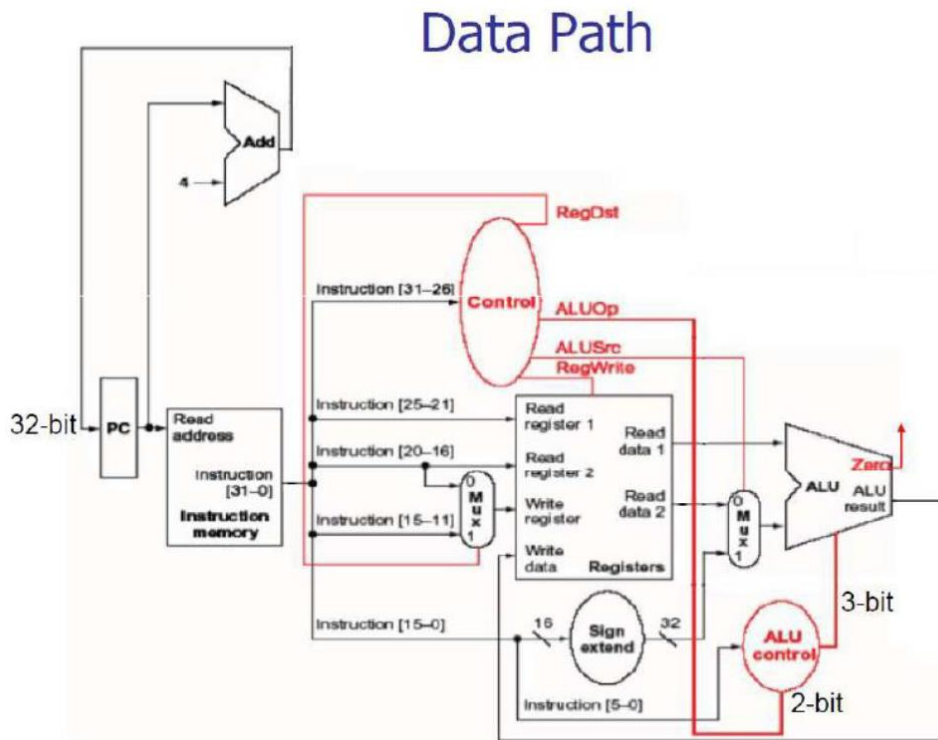
B04902027 資工三 陳昇

1. Coding environment

使用 Windows 的環境來寫，編譯環境則用 Linux。

2. Module implementation explanation

(一)



(二)

OP	RegDst	ALUSrc	RegWrite	ALUOp
000000	1	0	1	11
001000	0	1	1	10

(三)

ALUOp	func	ALUctr
10	x	010 (Add)
11	100000	010 (Add)
11	100010	110 (Sub)
11	100100	000 (And)
11	100101	001 (or)
11	011000	011 (Mul)

從表一的最前面開始實作，首先先寫 PC+4 的部分(Adder)，再來實作以 instaddr 來取 inst 的部分(Instruction_Memory)。有了 inst 後，我們可以分別將

inst 傳入 Control, MUX5, Registers, Sign_Extend, ALU_Control 做運算，其中 Control 以表二的規則來決定 4 個 output，分別傳入對應的目標內(MUX5, MUX32, Registers, ALU_Control)，ALU_Control 以 ALUOp 與 inst 的末 6 碼 (func)以表三的規則來決定 ALUCtr 給 ALU。之後 ALU 便以不同的 ALUCtr 來做不同的運算，並且將 Result 傳回 Registers。