UNIVERSITY OF MICHIGAN

JOINT INSTITUTE
交大密西根学院

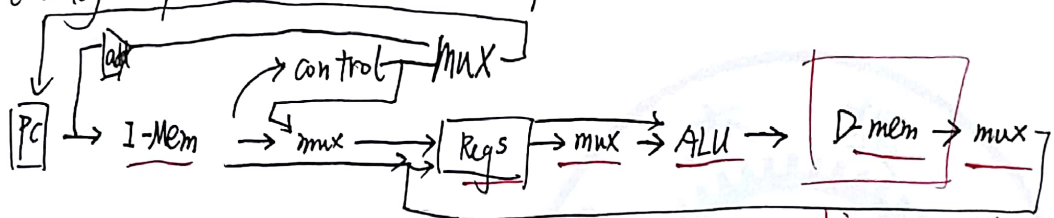
Name: 胡炳城
Student ID: 516021910219
Course Code: VE 370
Date:

Homework 4

4.2.2 a) we need two 2-input ALU or one 3-input ALU
b). we need a new ALU with this shift operation

4.2.3 a). An enable signal for 2nd ALU or tell wether to use 2 or 3 input
b): An signal to tell to perform shift operation.

4.2.4 a). latency depends on the longest path.



critical path is to get the value for load Ins. 注意: SW 和, LW 才用 D-Mem.

I-Mem → Regs → Mux → ALU → D-mem → mux

= 400 + 200 + 30 + 120 + 350 + 30 = 1130ps.

∵ Add units are not in the critical path, don't change

b): Latency for Regs + 100 ⇒ 1130 + 100 = 1230 ps.

4.6.5

a). None. I-Mem is slower than Add, so Add can not be in the critical path.

b). Loads & stores.

4.7.1 (这里没有 sw、lw 等操作)

The longest path for ALU is I-mem → Reg → mux → ALU → Mux → Reg (这儿跳过 D-mem; 另一个Reg)

a). 400 + 200 + 30 + 120 + 30 + 200 = 980

4.7.2

Only LW → Critical path. I-mem → Reg → mux → ALU → D-mem → mux

a). 400 + 200 + 30 + 120 + 350 + 30 = 1130 ps.

4.7.3 add - beq - lw - sw

means: ALU, shift Add, D-mem

And this means D-mem is the largest, so the answer is the same as 4.7.2.

4.8.1

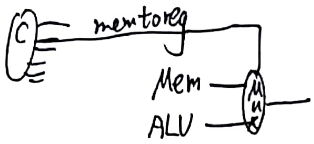
(a) I-mem, out, bit 7. | I | rs | rt | address (15:0) |

So we can test stuck at 0 by make address to be all 0 except bit 7:

addi \$t0, \$t0, 128   (128 is 10000000)

Then we check \$t0 to see wether bit 7 is zero.

(b) Control unit, MemtoReg:



Only when "lw" MemtoReg = 1.

lw \$t0 \$zero, if \$t0 ≠ 0, stuck at zero.

4.8.2

(a): \$t0 origin at 128, then addi \$t0, ~~\$t0~~ \$zero, ~~\$zero~~ 0

(b) No. not relible. If stuck at 1, then random write, may be the same as the value in register.

Because a signal can not be both 0 & 1. we can 4.

4.11.2

(a) 100011 00010 00011 00000 00000 010000

op $= 2^5+2^1+2^0 = 32+2+1 = 35 =$ lw   func

so ALU op = 00 (look up table), Ins (func) = 010000

(b) $000100_2 = 2^2 = 4 =$ beq

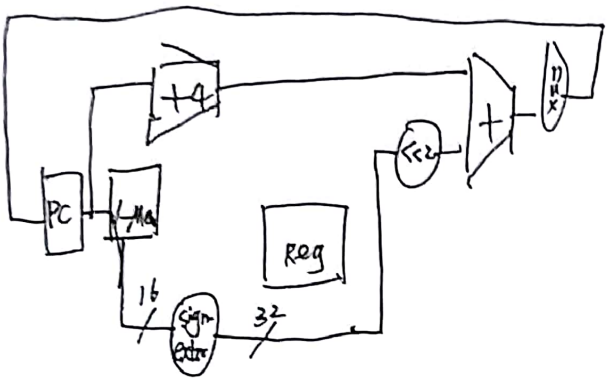
So ALU op = 01, Ins = 001100

4.11.3

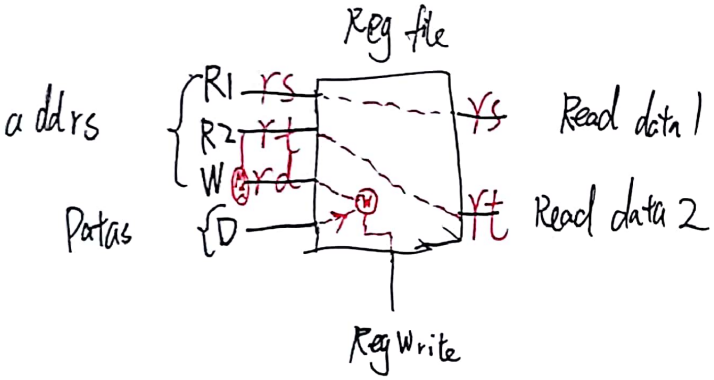
(a) new PC = PC+4

(b) new PC = PC+4 + 12×4 if $1 = $3

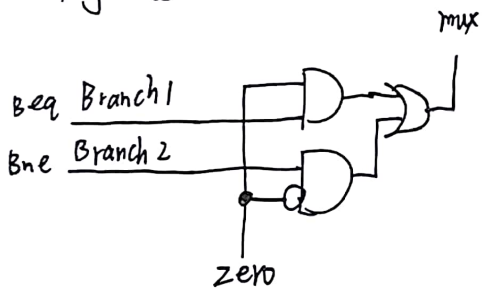
Addr = (001100 = 12)



12. Something important:

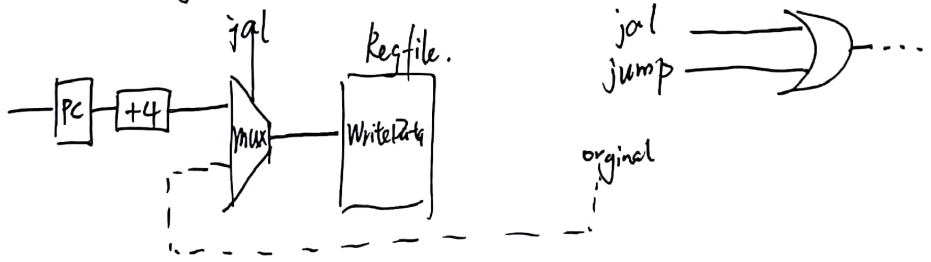


① Bne: Beq Branch1, Bne Branch2



② Jal: 1. make write register at Regfile to be $ra

2.



③ Jr: