11259022

Computer Organization HW04-1

1.1 180ps + 80ps = 260 ps ✱

1.2 SW ⇒ I-Mem + Ctrl. Unit + ALU-Ctrl. + ALU + Data-Mem = 740 ps

beq ⇒ I-Mem + Ctrl. Unit + ALU-Ctrl. + ALU + AND + MUX = 630 ps

add ⇒ I-Mem + Ctrl. Unit + ALU-Ctrl. + ALU + MUX + Regs. Access = 670 ps

Max (740, 630, 670) = 740 (ps) ✱

1.3 180 + 200 + 60 + 100 + 200 + 30 + 100 = 870 (ps) ✱

2.1 sign-extend : 0000 0000 0000 0000 0101 0000 0010 0010

shift left : 0101 1010 0101 0100 0000 1000 1000

2.2 instruction [31~26] = 000000 ⇒ R-format OPcode [1~0] = 10 ✱, instruction [5~0] = 100010 ✱

2.3 PC+=4 , PC → Add4 → branch MUX → jump MUX ✱

2.4
(1) RegDst MUX : 01010 ⇒ 10 ✱

(2) ALUsrc MUX : 3 ✱

(3) PCsrc MUX : PC+4 ✱

(4) MemtoReg MUX : 1 ✱ ($\frac{1}{rs}$ - $\frac{3}{rt}$)

2.5
(1) ALU : 4 , 3 ✱

(2) Add : PC , 4 ✱

(3) Add (ALU result) : PC+4 , 0000 0000 0000 0001 0100 0000 1000 1000 ✱

shift left 2 (×4)

2.6
(1) Read Register 1 : 11 ✱ (3) Write Register : 10 ✱ (5) Reg Write : 1 ✱

(2) Read Register 2 : 9 ✱ (4) Write Data : 1 ✱

3.1 ALUOP or ALUsrc, they need to generated during Reg. Access (350ps).

In this way, ALUsrc can enable Read Data to pass through Mux (100ps) in the shortest time.

As for ALUop. ALU control (100ps) signal can be generated in the shortest time.

Keep the ALU input delay at 100ps.

3.2

R-format: I-Mem. + RegDst + MUX + Regs-Access = 2250

lw: I-Mem. + ALUop + ALUctrl + ALU + Data-Mem + MUX + Regs-Access = 1950

sw: I-Mem. + MemWrite + Data-Mem = 2000

branch: I-Mem. + Branch + AND + MUX = 1600

MAX(2250, 1950, 2000, 1600)

= 2250 #

~~I-Mem + I-Mem~~

I-Mem + MAX(MAX(MAX(MAX(MAX(Regs.Access, ALUsrc) +MUX, ALUop + ALUctrl) + ALU, MemtoReg), RegWrite, RegDst + MUX)

I-Mem + MAX(MAX(MAX(MAX(Regs.Access, ALUsrc) +MUX, ALUop + ALUctrl) + ALU, MemRead) + Data-Mem, MemtoReg) + MUX, RegDst + MUX, RegWrite)

I-Mem + MAX(MAX(MAX(Regs.Access, ALUsrc) + MUX, ALUop + ALUctrl) + ALU, MemWrite) + Data-Mem

MAX(Add, I-Mem + MAX(Regs.Access, ALUsrc) + MUX, ALUop + ALUctrl) + ALU, Branch)

not support

4.1

I-Mem	+ Registers	+ ALU	+ Data-Mem	+ MUX	+ Registers	
500	+ 250	+ 150	+ 400	+ 50	+ 250	= 1600 ps
500	+ 250	+ 450	+ 400	+ 50	+ 250	= 1900 ps

4.2 $\frac{1600}{0.95 \times 1900} = 0.886$ #

$\frac{cost_1}{P_1} \Big/ \frac{cost_2}{P_2} = \frac{cost_1/cost_2}{P_1/P_2}$

4.3 1100 + 2×30 + 3×20 + 80 + 200 + 1600 + 500 = 3600

1100 + 60 + 60 + 580 + 200 + 1600 + 500 = 4100

$\frac{3600/4100}{0.886} = 0.991$ #