

CS2100 Computer Organization
Tutorial 5
Suggested answers for selected questions

Questions 1 and 2 refer to the complete datapath and control design covered in lectures #11 and #12. Please use the diagram in Lecture #12 slide 29 or in the COD MIPS 4th edition textbook, Figure 4.17. For your convenience, Lecture #12 slide 29 is also included at the end of this tutorial sheet.

1. Let us perform a complete trace to understand the working of the complete datapath and control implementation. Given the following three hexadecimal representations of MIPS instructions:

- i. **0x8df80000: lw \$24, 0(\$15)**
- ii. **0x1023000C: beq \$1, \$3, 12**
- iii. **0x0285c822: sub \$25, \$20, \$5**

For each instruction encoding, do the following:

- a. Fill in the tables below. The first table concerns with the various data (information) at each of the datapath elements, while the second table records the control signals generated. Use the notation \$8 to represent register number 8, [\$8] to represent the content of register number 8 and Mem(X) to represent the memory data at address X.
- b. Indicate the value of the PC after the instruction is executed.

Answers:

Only values in **RED** and **BOLD** font are actually utilized in the execution.

- i. **0x8df80000 = lw \$24, 0(\$15);** next PC = PC+4

Registers File				ALU		Data Memory	
RR1	RR2	WR	WD	Opr1	Opr2	Address	Write Data
\$15	\$24	\$24	MEM([\$15]+0)	[\$15]	0	[\$15]+0	[\$24]

RegDst	RegWr	ALUSrc	MRd	MWr	MToR	Brch	ALUop	ALUctrl
0	1	1	1	0	1	0	00	0010

- ii. **0x1023000C = beq \$1, \$3, 12;** next PC = PC+4 or (PC+4)+(12×4)

Registers File				ALU		Data Memory	
RR1	RR2	WR	WD	Opr1	Opr2	Address	Write Data
\$1	\$3	\$3 or \$0	[\$1]-[\$3] or random value	[\$1]	[\$3]	[\$1]-[\$3]	[\$3]

RegDst	RegWr	ALUSrc	MRd	MWr	MToR	Brch	ALUop	ALUctrl
X	0	0	0	0	X	1	01	0110

iii. `0x0285c822 = sub $25, $20, $5;` next PC = PC+4

Registers File				ALU		Data Memory	
RR1	RR2	WR	WD	Opr1	Opr2	Address	Write Data
\$20	\$5	\$25	[\$20]-[\$5]	[\$20]	[\$5]	[\$20]-[\$5]	[\$5]

RegDst	RegWr	ALUSrc	MRd	MWr	MToR	Brch	ALUop	ALUctrl
1	1	0	0	0	0	0	10	0110

2. With the complete datapath and control design, it is now possible to estimate the latency (time needed for a task) for the various type of instructions. Given below are the resource latencies of the various hardware components (ps = picoseconds = 10^{-12} second):

Inst-Mem	Adder	MUX	ALU	Reg-File	Data-Mem	Control/ ALUControl	Left-shift/ Sign- Extend/ AND
400ps	100ps	30ps	120ps	200ps	350ps	100ps	20ps

Give the estimated latencies for the following MIPS instructions:

- (a) “SUB” instruction (e.g. `sub $25, $20, $5`)
- (b) “LW” instruction (e.g. `lw $24, 0($15)`)
- (c) “BEQ” instruction (e.g. `beq $1, $3, 12`)

What do you think the **cycle time** should be for this particular processor implementation?

Hint: First, you need to find out the **critical path** of an instruction, i.e. the path that takes the longest time to complete. Note that there could be several parallel paths that work more or less simultaneously.

Answers:

Refer to Powerpoint slides.

- (a) SUB instruction (R-type):

Critical Path:

I-Mem → Reg.File → MUX(ALUSrc) → ALU → MUX(MemToReg) → Reg.File

Note: I-MEM → Control is a parallel path, the earliest signal needed is the ALUSrc. So, as long as the Control latency is lesser than Reg.File access latency, then it will not be in the critical path. Once the signal is generated, the Control latency will no longer affect the overall delays.

Similarly, there is another path to calculate the next PC (I-MEM → Control → AND → MUX(PCSrc) which is again not critical to the overall latency.

Latency = 400 + 200 + 30 + 120 + 30 + 200 = 980ps

(b) LW instruction:

Critical Path:

I-Mem \rightarrow Reg.File \rightarrow ALU \rightarrow DataMem \rightarrow MUX(MemToReg) \rightarrow Reg.File

$$\text{Latency} = 400 + 200 + 120 + 350 + 30 + 200 = 1300\text{ps}$$

Note: The path I-Mem \rightarrow Immediate \rightarrow MUX(ALUSrc) occurs simultaneously with the above.

(c) BEQ instruction:

Critical Path:

I-Mem \rightarrow Reg.File \rightarrow MUX(ALUSrc) \rightarrow ALU \rightarrow AND \rightarrow MUX(PCSrc)

$$\text{Latency} = 400 + 200 + 30 + 120 + 20 + 30 = 800\text{ps}$$

Since LW has the longest latency. The overall cycle time of the whole machine is determined by LW, i.e. at least 1300ps.

