

CS2100 Computer Organisation
Tutorial #5: MIPS Processor: Datapath and Control
(Week 7: 2 – 6 March 2020)

LumiNUS Open Discussion Question:

D1. Refer to the following two MIPS instructions:

- i. **add \$t0, \$s0, \$s1**
- ii. **bne \$24, \$25, finish**

Assume that the instruction at label “**finish**” in (ii) is three instructions after **bne**.

For each of the above two instructions, do the following:

- (a) Write out the instruction in hexadecimal representation.
- (b) What are the values of the following control signals: **RegDst**, **RegWrite**, and **Branch**?
- (c) What is the register number supplied to the register file’s “Read Register 1” (RR1) input? How about “Read Register 2” (RR2)? Are these register contents utilized in the the ALU stage?
- (d) What is the register number supplied to the register file’s “Write register” (WR) input? Is this register actually written?

The answers for (i) are given below.

- (a) \$t0 = \$8; \$s0 = \$16; \$s1 = \$17

add \$t0, \$s0, \$s1:

000000 10000 10001 01000 00000 100000 = **0x02114020**

- (b)

Instruction	RegDst	RegWrite	Branch
add	1	1	0

- (c)

Instruction	Read Register 1	Used in ALU?	Read Register 2	Used in ALU?
add	16 (10000 ₂)	Yes	17 (10001 ₂)	Yes

- (d)

Instruction	Write register	Actually written?
add	8 (01000 ₂)	Yes

Tutorial 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, the diagram is also included at the end of this tutorial sheet.

- 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:

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

For each instruction encoding, do the following:

- Fill in the two tables below. The first table contains the various data (information) at 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 data at memory address X.

Registers File				ALU		Data Memory	
RR1	RR2	WR	WD	Opr1	Opr2	Address	Write Data

[Wr = Write; Rd = Read; M = Mem; R = Reg]

RegDst	RegWr	ALUSrc	MRd	MWr	MToR	Brch	ALUop	ALUctrl

- Indicate the value of the PC after the instruction is executed.

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 working more or less simultaneously.

3. [AY2011/12 Semester 2 Exam]

Let us consider the task of incorporating the “**sll**” (shift-left logical) instruction into the MIPS processor. Suppose we have the following new datapath element “Shifter” which has the following functionality:

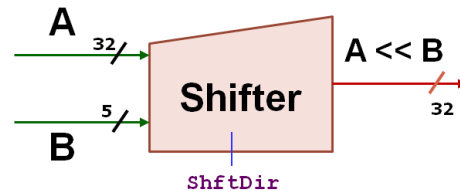
Inputs:

A is a 32-bit value

B is a 5-bit unsigned value

Output:

$A \ll B$, i.e. A left shifted by B positions



Control (ShftDir):

0 for left-shift, 1 for right-shift

Hardwired to 0 for left-shifting for this part

- (a) For the inputs A and B, indicate the correct values to connect to them.

For the output ($A \ll B$), indicate **an existing multiplexer** that it can connect to. Briefly explain how to control that existing multiplexer properly. You do not need to figure out how the control signal(s) is/are generated.

- (b) It is obviously a waste to use the “Shifter” element in (a) for left shifting only. Suppose we want to incorporate the “**srl**” (shift-right logical) instruction into the processor design in (a). Considering only the **ShftDir** control signal, explain how you can generate this signal from the Opcode/Funct Field of “**sll**” and “**srl**” instructions (given below). Give the simplest Boolean expression.

sll: opcode = 0_{16} ; funct = 0_{16}

srl: opcode = 0_{16} ; funct = 2_{16}

You may use Op[5:0] and F[5:0] to indicate the respective bits in the opcode and funct fields.

- (c) How does the processor with the above new functionalities handle non-shift instructions? Briefly explain.

