

Lebanese American University

School of Arts and Sciences

Department of Computer Science and Mathematics

CSC 320 – Computer Organization

---

## Problem Set 5: The Processor

---

### Exercise 1

Different instructions utilize different hardware blocks in the basic single-cycle implementation. The next three problems in this exercise refer to the following instruction:

	Instruction	Interpretation
a.	AND Rd, Rs, Rt	$\text{Reg}[\text{Rd}] = \text{Reg}[\text{Rs}] \text{ AND } \text{Reg}[\text{Rt}]$
b.	SW Rt, Offs(Rs)	$\text{Mem}[\text{Reg}[\text{Rs}] + \text{Offs}] = \text{Reg}[\text{Rt}]$

**1.1** What are the values of control signals generated by the control in Figure 4.2 for this instruction?

Solution:

The values of the signals are as follows:

	RegWrite	MemRead	ALUMux	MemWrite	ALUop	RegMux	Branch
a.	1	0	0(Reg)	0	AND	1(ALU)	0
b.	0	0	1(Imm)	1	ADD	X	0

ALUMux is the control signal that controls the Mux at the ALU input, 0 (Reg) selects the output of the register file and 1 (Imm) selects the immediate from the instruction word as the second input to the ALU.

RegMux is the control signal that controls the Mux at the data input to the register file, 0 (ALU) selects the output of the ALU, and 1 (Mem) selects the output of memory.

A value of X is a “don’t care” (does not matter if signal is 0 or 1).

**1.2** Which resources (blocks) perform a useful function for this instruction?

Solution:

Resources performing a useful function for this instruction are:

a.	All except Data Memory and branch Add unit
b.	All except branch Add unit and write port of the Registers

**1.3** Which resources (blocks) produce outputs, but their outputs, are not used for this instruction? Which resources produce no outputs for this instruction?

Different execution units and blocks of digital logic have different latencies (time needed to do their work). In Figure 4.2 there are seven kinds of major blocks. Latencies of blocks along the critical (longest -latency) path for an instruction determine the minimum latency of that instruction. For the remaining three problems in this exercise, assume the following resource latencies:

	I-Mem	Add	Mux	ALU	Regs	D-Mem	Control
a.	200ps	70ps	20ps	90ps	90ps	250ps	40ps
b.	750ps	200ps	50ps	250ps	300ps	500ps	300ps

Solution:

	Outputs That Are Not Used	No Outputs
a.	Branch Add	Data Memory
b.	Branch Add, write port of Registers	None (all units produce outputs)

**1.4** What is the critical path for an MIPS AND instruction?

Solution:

One long path for AND instruction is to read the instruction, read the registers, go through the ALU Mux, perform the ALU operation, and go through the Mux that controls the write data for Registers (I-Mem, Regs, Mux, ALU, and Mux). The other long path is similar, but goes through

Control while registers are read (I-Mem, Control, Mux, ALU, Mux). There are other paths but they are shorter, such as the PC increment path (only Add and then Mux), the path to prevent branching (I-Mem to Control to Mux, so the Mux can use the Branch signal to select the PC + 4 input as the new value for PC), and the path that prevents a memory write (only I-Mem and then Control, etc.).

<b>a.</b>	Control is faster than registers, so the critical path is I-Mem, Regs, Mux, ALU, Mux.
<b>b.</b>	The two long paths are equal, so both are critical.

### 1.5 What is the critical path for an MIPS load (LD) instruction?

Solution:

One long path is to read the instruction, read registers, use the Mux to select the immediate as the second ALU input, use ALU (compute address), access D-Mem, and use the Mux to select that as register data input, so we have I-Mem, Regs, Mux, ALU, D-Mem, Mux. The other long path is similar, but goes through Control instead of Regs (to generate the control signal for the ALU MUX).

<b>a.</b>	Control is faster than registers, so the critical path is I-Mem, Regs, Mux, ALU, Mux.
<b>b.</b>	The two long paths are equal, so both are critical.

### 1.6 What is the critical path for an MIPS BEQ instruction?

Solution:

This instruction has two kinds of long paths, those that determine the branch condition and those that compute the new PC. To determine the branch condition, we read the instruction, read registers or use the Control unit, then use the ALU Mux and then the ALU to compare the two values, then use the Zero out-put of the ALU to control the Mux that selects the new PC.

<b>a.</b>	The first path (through Regs) is longer.
<b>b.</b>	The two long paths are equal, so both are critical.

To compute the PC, one path is to increment it by 4 (Add), add the offset (Add), and select that value as the new PC (Mux). The other path for computing the PC is to read the instruction (to get the offset) and use the branch Add unit and Mux. Both of the compute-PC paths are shorter than the critical path that determines the branch condition, because I-Mem is slower than the PC + 4 Add unit, and because ALU is slower than the branch Add.

## Exercise 2

Problems in this exercise assume that logic blocks needed to implement a processor's datapath have the following latencies:

	I-Mem	Add	Mux	ALU	Regs	D-Mem	Sign-Extend	Shift-Left-2
a.	200ps	70ps	20ps	90ps	90ps	250ps	15ps	10ps
b.	750ps	200ps	50ps	250ps	300ps	500ps	100ps	0ps

**2.1** If the only thing we need to do in a processor is fetch consecutive instructions (Figure 4.6), what would the cycle time be?

Solution:

I-Mem takes longer than the Add unit, so the clock cycle time is equal to the latency of the I-Mem:

a.	200ps
b.	750ps

**2.2** Consider a datapath similar to the one in Figure 4.11, but for a processor that only processor has one type of instruction: unconditional PC-relative branch, What would the cycle time be for this datapath?

Solution:

The critical path for this instruction is through the instruction memory, Sign-extend and Shift-left-2 to get the offset, Add unit to compute the new PC, and Mux to select that value instead of PC + 4. Note that the path through the other Add unit is shorter, because the latency of I-Mem is longer than the latency of the Add unit. We have:

a.	$200\text{ps} + 15\text{ps} + 10\text{ps} + 70\text{ps} + 20\text{ps} = 315\text{ps}$
b.	$750\text{ps} + 100\text{ps} + 0\text{ps} + 200\text{ps} + 50\text{ps} = 1100\text{ps}$

**2.3** Repeat 2.2, but this time we need to support only conditional PC-relative branches.

Solution:

Conditional branches have the same long-latency path that computes the branch address as unconditional branches do. Additionally, they have a long-latency path that goes through Registers, Mux, and ALU to compute the PCSrc condition. The critical path is the longer of the two, and the path through PCSrc is longer for these latencies:

a.	$200\text{ps} + 90\text{ps} + 20\text{ps} + 90\text{ps} + 20\text{ps} = 420\text{ps}$
b.	$750\text{ps} + 300\text{ps} + 50\text{ps} + 250\text{ps} + 50\text{ps} = 1400\text{ps}$

**Exercise 3**

In this exercise refer to the following logic block (resource) in the datapath:

Resource	
a.	Shift-left-2
b.	Registers

**3.1** Which kinds of instructions require this resource?

Solution:

a.	PC-relative branches.
b.	All instructions except unconditional jumps without a register operand (jal, j).