

# COE 301 / ICS 233

## Computer Organization

### Final Exam – Term 172

Tuesday, May 15, 2018

8:00 am – 10:30 am

Computer Engineering Department  
College of Computer Sciences & Engineering  
King Fahd University of Petroleum & Minerals

Student Name: \_\_\_\_\_ ID: \_\_\_\_\_

<input type="checkbox"/> Dr. Aiman El-Maleh	<input type="checkbox"/> COE 301	<input type="checkbox"/> ICS 233
<input type="checkbox"/> Dr. Marwan Abu-Amara	<input type="checkbox"/> COE 301	<input type="checkbox"/> ICS 233
<input type="checkbox"/> Dr. Muhamed Mudawar		

Q1	/ 17	Q2	/ 8
Q3	/ 14	Q4	/ 19
Q5	/ 10	Q6	/ 12
Total	/ 80		

#### Important Reminder on Academic Honesty

Using unauthorized information or notes on an exam, peeking at others work, or altering graded exams to claim more credit are severe violations of academic honesty. Detected cases will receive a failing grade in the course.

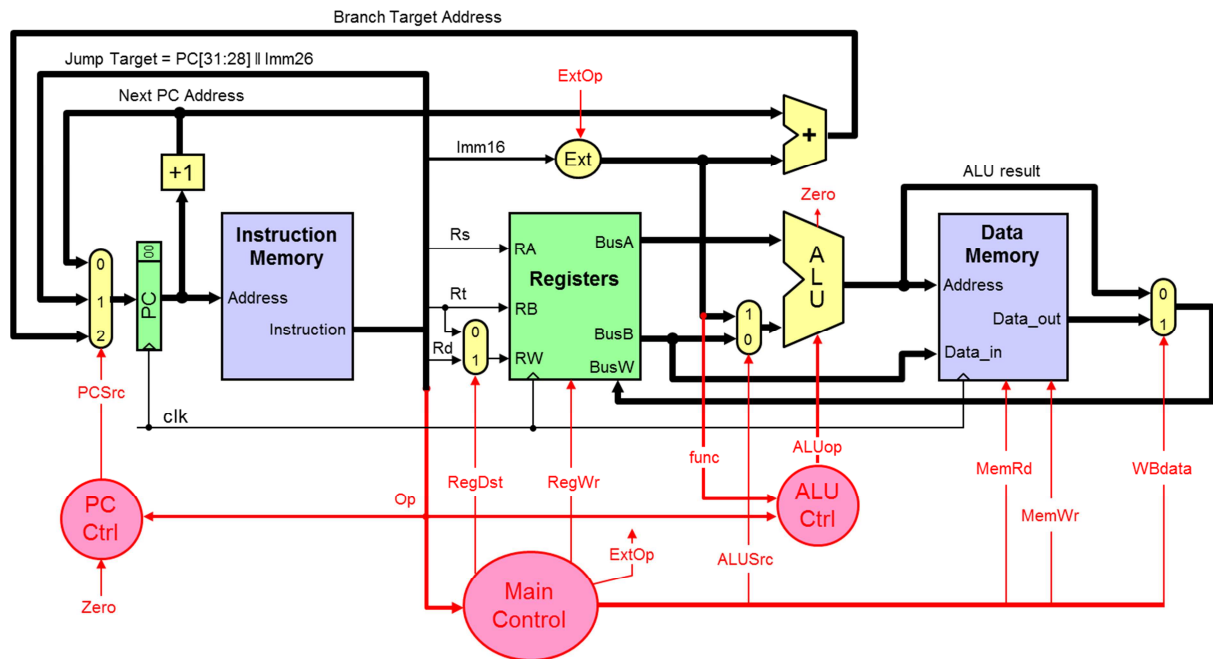
### Q1. [17 points] General Understanding of Topics

- a) **(1 point)** Does pipelining improve the latency of individual instructions? Explain.
- b) **(2 points)** What causes control hazards in a pipelined datapath and how control hazards can be eliminated?
- c) **(2 points)** Explain the difference between static RAM and dynamic RAM.
- d) **(2 points)** Is it possible to use only one memory for both instructions and data in the single-cycle datapath? Explain why or why not. Is it possible to use only one memory for both instructions and data in a multi-cycle datapath? Explain.
- e) **(2 points)** Why do we need cache memory, and why do we have two separate cache memories (I-cache and D-cache) in a pipelined processor?

- f) **(2 points)** Explain the concepts of temporal locality and spatial locality of reference in cache memory.
  
  
  
  
  
  
  
  
  
  
- g) **(2 points)** What needs to be stored inside a cache for block identification? How does a cache know whether there is a cache hit or miss?
  
  
  
  
  
  
  
  
  
  
- h) **(2 points)** Suppose a 4-way set-associative cache has a capacity of 32 KiB (1 KiB = 1024 bytes) and each block consists of 64 Bytes. What is the total number of blocks in the cache? What is the number of sets?
  
  
  
  
  
  
  
  
  
  
- i) **(2 points)** Explain the difference between a write-through and a write-back cache.

## Q2. [8 points] Single-Cycle Processor

The single-cycle datapath and control of a MIPS-like processor is shown below. However, this datapath and control lacks the implementation of many important instructions.



Consider adding the following two new instructions to the above datapath: **JLR** and **LWI**. The **JLR** instruction is I-type and has a unique opcode. The **LWI** instruction is R-type and has a unique function code. The least-significant 2 bits of register **PC** are hardwired to **00**, and not stored in **PC**. Therefore, it is sufficient to increment **PC** by **1** to point to the next instruction in memory.

Instruction	Format	Meaning
Jump and Link Register Op = JLR	Op, Rs, Rt, Imm16	$\text{Reg[Rt]} = \text{PC} + 1$ $\text{PC} = \text{Reg[Rs]} + \text{Imm16}$
Load Word Indexed Op = R-type, Func = LWI	Op, Rs, Rt, Rd, Func	$\text{Reg[Rd]} = \text{Mem}[\text{Reg[Rs]} + \text{Reg[Rt]}]$

- (4 points) Redraw the necessary changes to the above datapath to implement the above two instructions. Draw only the modified parts and explain why they are needed.
- (4 points) Identify any new control signal needed to implement the above two instructions. Draw a table showing the values of all control signals to implement the above two instructions.

## Q2 Solution

**Q3. [14 points] Performance of Single-Cycle, Multi-Cycle, and Pipelined CPU**

Compare the performance of a **single-cycle processor** and a **multi-cycle processor**. The delay times are as follows:

Instruction memory access time = 500 ps

Data memory access time = 500 ps

Instruction Decode and Register read = 200 ps

Register write = 200 ps

ALU delay = 100 ps

Ignore the other delays in the multiplexers, wires, etc. Assume a program has the following instruction mix: **40% ALU, 5% load, 5% store, 30% branch, and 20% jump**.

**a) (6 points)** Compute the delay for each instruction class and the clock cycle for the **single-cycle processor**.

Instruction Class	Instruction Memory	Decode and Register Read	ALU	Data Memory	Write Back	Total Delay
ALU						
Load						
Store						
Branch						
Jump						

Clock cycle for the single-cycle processor =

**b) (2 points)** Compute the **clock cycle** and the **average CPI** for the **multi-cycle processor**.

Clock cycle for the multi-cycle processor =

Average CPI for the multi-cycle processor =

**c) (2 points)** Determine **quantitatively** if there is a speedup when using the multi-cycle processor with respect to the single-cycle.

Speedup =

- d) (2 points)** Assume that the processor is **pipelined**. Furthermore, assume that a program has the following instruction mix: **40%** ALU, **5%** load, **5%** store, **30%** branch, and **20%** jump. Moreover, assume that **90%** of the branches will be **taken**. The CPU stalls **1** cycle for each jump and **2** cycles for each taken branch. Compute the **average CPI** for the **pipelined** processor due to control hazards only.

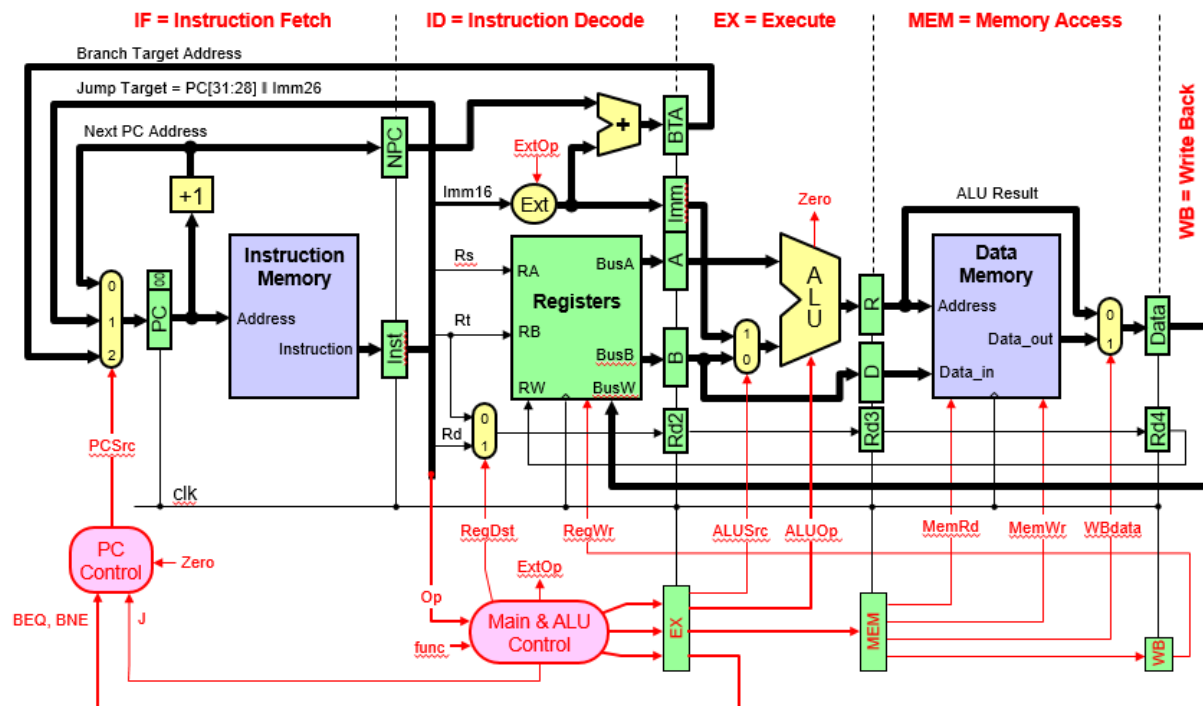
Average CPI for the **pipelined** processor for control hazards =

- e) (2 points)** Assume that the processor is **pipelined** and that load instructions are **5%** of the instruction count and store instructions are also **5%** as given above. However, the program spends **30%** of its execution time executing load instructions and **15%** of its execution time executing store instructions. The designers discovered that the Data cache is producing many cache misses causing the CPU to stall. They decided to improve the design of the data cache and improve the execution time of the load instructions by a factor of **3x** (3 times faster) and the store instructions by a factor of **2x**. Determine the overall speedup of the program due to the improvements done to the data cache.

Speedup due to data cache improvement =

#### Q4. [19 points] Pipelined CPU Design

I. Consider the 5-stage pipelined CPU design given below.



- a) (5 points) Show the design changes needed for handling data hazards using forwarding including a block diagram for data hazard detection and forwarding unit.



- b) (4 points)** Show the control signals that will be used for stalling the pipeline for data hazards due to load instructions along with their conditions. Show the necessary changes that need to be done to the design.

- c) **(2 points)** Show the control signals that will be used for handling control hazards. Show the necessary changes that need to be done to the design.

- d) **(3 points)** Show the design of the PC control logic that includes handling of control hazards assuming that only BEQ, BNE, and J instructions are implemented.

**II. (5 Points)** Consider the following MIPS assembly language code:

```

I1:  ORI  $s0, $0, 5
I2:  ADDI $s1, $0, 10
I3:  ADD  $s1, $s0, $s1
I4:  LW   $s0, -4($s1)
I5:  ADD  $s0, $s0, $s0
I6:  SW   $s0, -4($s1)

```

Complete the following table showing the timing of the above code on the 5-stage pipeline given in part (i) (IF, ID, EX, MEM, WB) supporting **forwarding** and **pipeline stall**. Draw an arrow showing forwarding between the stage that provides the data and the stage that receives the data. Show all stall cycles (draw an X in the box to represent a stall cycle). Determine the number of clock cycles to execute this code.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
I1: ORI	<b>IF</b>	<b>ID</b>	<b>EX</b>	-	<b>WB</b>										
I2: ADDI															
I3: ADD															
I4: LW															
I5: ADD															
I6: SW															

Total number of clock cycles to execute the above code =

**Q5. [10 points] Cache Memory**

- a) (3 points) Given that the memory address consists of 64 bits, consider a 64 KiB **fully associative** cache (1 KiB = 1024 bytes) with 64-byte cache blocks and a write back policy is used. Compute the **total number of bits** required to store the **valid**, **modified**, and **tag** bits in the cache.

Total Valid bits =

Total Modified bits =

Total Tag bits =

- b) (3 points) Assume that the memory address consists of 64 bits, and a 64 KiB **4-way set associative** cache with 64-byte cache blocks is used. Find the number of **tag** bits, **index** bits, and **offset** bits needed.

Offset bits =

Index bits =

Tab bits =

- c) (4 points) Given a 2-way set-associative cache that uses 32-bit memory addresses divided into 4 bits of offset, 12 bits of index, and 16 bits of tag. Starting with an empty cache, show the **tag**, **index**, and **way** (block 0 or 1) for each of the following sequentially referenced addresses and indicate whether the reference resulted in a **hit** or a **miss**. The replacement policy used is **FIFO**.

Address	Tag	Index	Way	Hit / Miss
0x00553F0F				
0x00773F01				
0x00553F02				
0x005530AC				
0x00773F07				
0x005530AA				
0x009930AB				
0x00993F05				

### Q6. [12 points] Cache Performance

A processor runs at 2.5 GHz and has a CPI=1.7 for a perfect cache (i.e. without including the stall cycles due to cache misses). Assume that load and store instructions are 15% of the instructions. The processor has an I-cache with a 4% miss rate and a D-cache with 6% miss rate. The hit time is 1 clock cycle for both caches. Assume that the time required to transfer a block of data from the main memory to the cache, i.e. miss penalty, is 40 ns.

- a) **(4 Points)** Compute the number of stall cycles per instruction and the overall CPI.
- b) **(4 Points)** Compute the average memory access time (AMAT) in ns.
- c) **(4 Points)** Discuss how the average memory access time (AMAT) can be reduced by mentioning all the factors that could reduce it and for each factor explaining how it can be done.