

*King Fahd University of Petroleum and Minerals*  
*College of Computer Science and Engineering*  
*Computer Engineering Department*

**COE 301 COMPUTER ORGANIZATION**  
**ICS 233: COMPUTER ARCHITECTURE & ASSEMBLY LANGUAGE**  
**Term 161 (Fall 2016-2017)**  
**Final Exam**  
**Monday Jan. 16, 2017**  
**12:30-3:00 PM**

**Time: 150 minutes, Total Pages: 14**

**Name:** \_\_\_\_\_ **ID:** \_\_\_\_\_ **Section:** \_\_\_\_\_

**Notes:**

- Do not open the exam book until instructed
- Answer all questions
- All steps must be shown
- Any assumptions made must be clearly stated
- Mobile phones must be switched off

<b>Question</b>	<b>Max Points</b>	<b>Score</b>
<b>Q1</b>	<b>18</b>	
<b>Q2</b>	<b>20</b>	
<b>Q3</b>	<b>15</b>	
<b>Q4</b>	<b>22</b>	
<b>Total</b>	<b>75</b>	

Dr. Aiman El-Maleh  
Dr. Marwan Abu-Amara

**[18 Points]****(Q1)**

- (i) Given two different computers, A and B, the following measurements have been made for running a Java program (Program 1) on these two computers:

Program 1	A	B
#Instructions Executed	$5 \times 10^9$	$8 \times 10^9$
CPI	1.2	1.5

Compute the clock rate ratio of computer A to computer B so that the execution time for Program 1 on both computers is the same. **(3 points)**

- (ii) Consider two different implementations, M1 and M2, of the same instruction set. There are three classes of instructions (A, B, and C) in the instruction set. M1 has a clock rate of 4 GHz and M2 has a clock rate of 2 GHz. The CPI for each instruction class on M1 and M2 is given in the following table:

Class	CPI on M1	CPI on M2	C1 Usage	C2 Usage
A	1	2	40%	30%
B	3	2	50%	30%
C	4	2	10%	40%

The table above also contains a summary of the usage of the instruction classes generated by two different compilers, C1 and C2. Assume for any given program that compiler C1 generates 10% more instructions than compiler C2. Which implementation and compiler combination gives the best performance? **(4 points)**

- (iii) 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 = 300 ps      Register write = 100 ps  
ALU delay = 300 ps

Ignore the other delays in the multiplexers, wires, etc. Assume the following instruction mix: 40% ALU, 20% load, 10% store, 20% branch, and 10% jump.

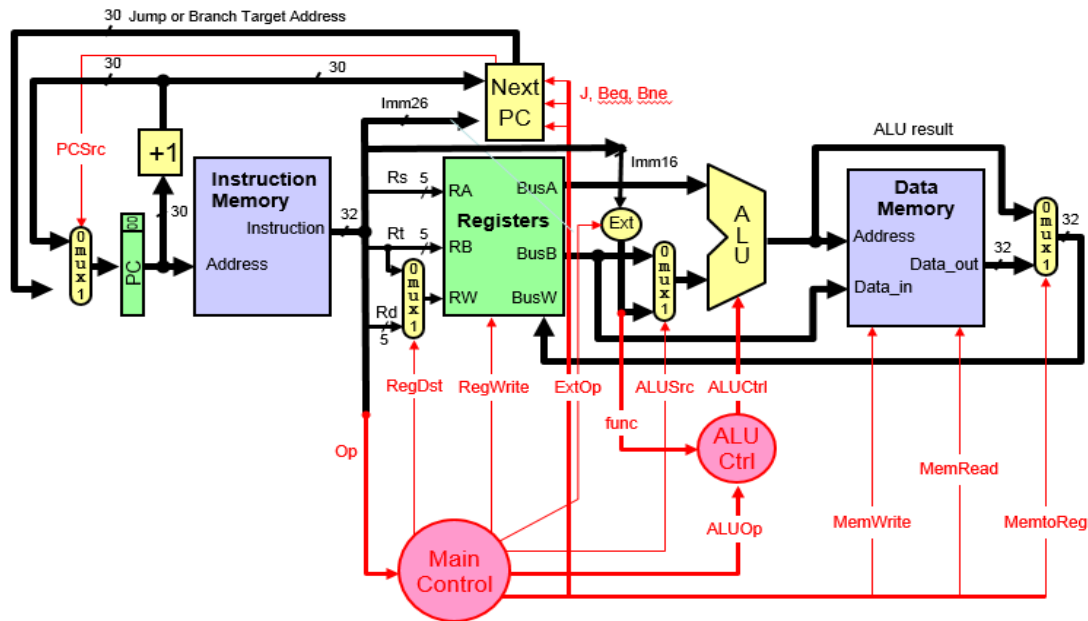
- a) Compute the delay for each instruction class and the clock cycle for the **single-cycle processor**. **(4 points)**

- b) Compute the clock cycle and the average CPI for the **multi-cycle processor**. **(3 points)**

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

- d) If we want to improve the execution time of a program on the **single-cycle processor** by a factor of 2, determine quantitatively whether enhancing the ALU unit can or cannot help achieve an overall speedup of 2 for the program execution time. **(2 points)**

(Q2) Consider the single-cycle CPU design given below:



- (i) Show the necessary modifications in the data path and control unit to implement this CPU as a 5-stage Pipeline (IF, ID, EX, MEM, WB) without consideration of data and control hazards. You can make changes in the given diagram but clarify the changes made and label them clearly. Assume that the NextPC block is independent of the ALU and that it has its own comparator and will be placed in the 2<sup>nd</sup> stage (i.e., ID stage) to reduce branch penalty. (7 Points)
- (ii) Add the necessary changes to perform forwarding due to data hazards. Draw only the modified parts. Show the conditions that will be used for generating the **ForwardA** signals. (6 Points)
- (iii) Add the necessary changes to stall the pipeline due to data hazards. Draw only the modified parts. State the conditions for stalling the pipeline due to Data Hazards. (4 Points)
- (iv) Add the necessary changes to stall the pipeline due to control hazards. Draw only the modified parts. State the conditions for stalling the pipeline due to Control Hazards. (3 Points)











- (ii) A sequence of three branches are shown in the first column of the table below with their respective PC value, branch target address, and next PC. These branches are executed in four passes (pass 1 to 4). The actual branch outcomes of each branch in each pass are shown where T represents a branch taken and NT represents a branch not taken. Assume a Branch Target Buffer (BTB) is used for early prediction of taken branches.

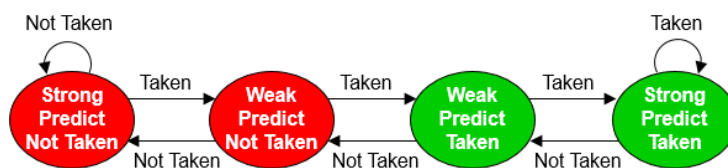
Four Branches with their respective PC values, branch target address, and next PC.	Branch outcomes in four passes			
	Actual Branch outcome In pass 1	Actual Branch outcome In pass 2	Actual Branch outcome In pass 3	Actual Branch outcome In pass 4
00010 Beq --, (target=00100) 00011 .... ..	T	T	T	NT
00101 Beq --, (target=00111) 00110 .... ..	NT	T	NT	T
01011 Beq --, (target=01101) 01100 .... ..	T	NT	T	T

- a) Fill in the BTB entries for PC, target and initial prediction (T for all) for the three branches above. Fill in the prediction in BTB table **after** each pass (1 to 4) by assuming a 1-bit prediction. Then, compute the probability of correct prediction. (3 points)

BTB			BTB Prediction just <b>after</b> pass k (k = 1 to 4)			
PC	Target	Prediction initial	Prediction pass 1	Prediction pass 2	Prediction pass 3	Prediction pass 4

Probability of Correct Prediction =

- b) Repeat the question above by predicting the branch outcome using a 2-bit saturating counter (given below). Denote by NT1 and T1 the weak NT and weak T, respectively. Assume that the predictor is initialized to T1 (weak predict taken). Then, compute the probability of correct prediction. (3 points)



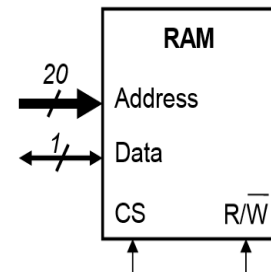
BTB			BTB Prediction just <b>after</b> pass k (1 to 4)			
PC	Target	Prediction initial	Prediction pass 1	Prediction pass 2	Prediction pass 3	Prediction pass 4
Probability of Correct Prediction =						

- c) Assume that a correctly predicted branch incurs zero stalls and a mis-predicted branch incurs 2 stalls. Evaluate the average number of stalls per instruction for using the 1-bit and 2-bit predictors if 15% of the instructions are branches. **(2 points)**

[22 Points]

(Q4)

- (i) Given a 1M x 1 memory block as shown below. Use this block to implement a 4M x 4 memory block. (4 Points)



- (ii) Assume that you have a 32-bit address and a cache with **8K byte data size** (not including tag and valid bits).
- a) Assuming that the cache is organized as **direct-mapped** with a **32-byte block size**, determine the number of bits in the offset, index and tag fields. (3 Points)

b) Assuming that the cache is organized as **four-way set associative** with a **32-byte block size**, determine the number of bits in the offset, index and tag fields. **(2 Points)**

c) Show the organization of a **cache organized as four-way set associative** with a **32-byte block size**. **(4 Points)**

(iii) A processor runs at 2.0 GHz and has a CPI=1.9 for a perfect cache (i.e. without including the stall cycles due to cache misses). Assume that load and store instructions are 20% of the instructions. The processor has an I-cache with a 3% 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 RAM to the cache, i.e. miss penalty, is 25 ns.

a) What is the number of stall cycles per instruction and the overall CPI? **(3 Points)**

b) What is the average memory access time (AMAT) in ns? (4 Points)

c) Consider the given figure below which plots the Miss Rate vs. the Block Size for various cache sizes. Explain why increasing the block size reduces the miss rate initially but after a certain point the miss rate increases by increasing the block size. (2 Points)

