National University of Computer and Emerging Sciences, Lahore Campus

Course:	**Computer Architecture**	**Course Code:**	**EE204**
Program:	**BS(Computer Science)**	**Semester:**	**Fall 2017**
Duration:	**150 Minutes**	**Total Marks:**	**60**
Date:	**26-12-2017**	**Weight**	**30/45**
Section:	**All**	**Page(s):**	**10**
Exam:	**CA Subjective**	**Student Id:**	

NOTE: Answer in the space provided. You can ask for rough sheets but they won't be graded.
If you think some information is missing. Make proper assumption and clearly mention it at the top of your solution.
Pipelined Architecture diagrams have been provided at the end of the exam for your reference.

Question 1 [Marks: 2+2+4+4]
Answer these questions. Show précised but brief calculations if applicable.

1. Suppose each page table entry consists of:

   o A physical page number

   o 1 valid bit

   o 1 dirty bit

   - Virtual addresses are 32 bits - Physical addresses are 26 bits

   - The page size is 8 Kbytes

   What is the size of the page table?

2. Assume that you have the following sequence of pipelined instructions:

```
lwR6, 0(R4)
addR5, R2, R3
sub R7, R6,R5
```

   a. Which pipeline stage will forward data operand R6 during the EX stage of the subtract?

   b. Which pipeline stage will forward data operand R5 during the EX stage of the subtract?

3. A compiler designer is trying to decide between two code segments for a particular machine. The hardware designers have provided the following data about the CPI for each class, and the instruction counts being considered for each code sequence

Instruction Class	CPI for Instructions of this class
A	3
B	2

	Instruction Count for each code Sequence	
Code Sequence	Class A type Instructions	Class B type Instructions
1	4	3
2	2	4

   a. How many cycles are required for code sequence1?

   b. What is the CPI for code sequence 2?

4. Assuming a cache of 4K blocks, a 4-word block size, and a 32‑bit address.

    1. The total number of sets for a direct mapped cache: _____

    2. The total number of tag bits for a direct mapped cache: _____

    3. The total number of sets for a fully associative cache: _____

    4. The total number of tag bits for a fully associative cache: _____

Question No. 2 [Marks: 15]

Consider the following program that will be executed on **2-issue Superscalar MIPS** architecture with **Static Branch Prediction (BACKWARD TAKEN)**
Consider the following specification for our pipelined processor:
• There are 2 units for each stage.
• Five stages MIPS pipeline with **execution stage consuming 2 clock** cycles for all instructions whereas all other stages consume 1 cycle.
• **Forwarding is implemented**
• Computation of PC and TARGET ADDRESS for branch & jump instructions anticipated in the **ID stage**

```
Loop: ld R1,40(R6)
      ld R2,60(R6)
      add R5,R2,R1
      ld R3,0(R5)
      or R3,R5,R3
      st R3,0(R5)
      addi R6,R6,4
      beq R6,R7, Loop
```

a. In-order issue and out-of-order execution and completion rules are applied. For each cycle, show which instructions is at which stage by filling the following table. You don't need to write complete instructions. Just write the instruction numbers from the code above. Remember you cannot re-arrange the instructions for this part of the question. Instructions wait in decode stage if there is some dependency. You can fetch a new instruction as soon as a slot is available in IF stage.

Cycle No.	IF	ID	EX	MEM	WB
1					
2					
.					
.					
.					
.					
.					
.					
.					
.					
.					
.					
.					
.					

b. Now try to remove stalls from the code, unroll the loop for level-2 (2 iterations only) if required and rearrange the instructions to get best execution time with the specifications described above. Also mention the changes required in instructions.

Question No. 3 [Marks: 18]

The following problem concerns the way we access data in the presence of virtual memory. We have to follow complete process of address translation and then access the data.

- The memory is byte addressable
- Virtual addresses are 18 bits wide.
- Physical addresses are 12 bits wide.
- The page size is 512 bytes.
- The TLB is 8-way set associative with 16 blocks.
- The cache is 2-way set associative, with a 4-byte block size and 32 blocks.

In the following tables, all numbers are given in hexadecimal. The contents of the TLB and the page table for the first 32 pages, and initial 4 sets of cache are as follows:

TLB			
Index	Tag	PPN	Valid
0	55	6	0
	48	F	1
	00	A	0
	32	9	1
	6A	3	1
	56	1	0
	60	4	1
	78	9	0
1	71	5	1
	31	A	1
	53	F	0
	87	8	0
	51	D	0
	39	E	1
	43	B	0
	73	2	1

Page Table					
VPN	PPN	Valid	VPN	PPN	Valid
000	7	0	010	1	0
001	5	0	011	3	0
002	1	1	012	3	0
003	5	0	013	0	0
004	0	0	014	6	1
005	5	0	015	5	0
006	2	0	016	7	0
007	4	1	017	2	1
008	7	0	018	0	0
009	2	0	019	2	0
00A	3	0	01A	1	0
00B	0	0	01B	3	0
00C	0	0	01C	2	0
00D	3	0	01D	7	0
00E	4	0	01E	5	1
00F	7	1	01F	0	0

2-way Set Associative Cache												
Index	Tag	Valid	Byte 0	Byte 1	Byte 2	Byte 3	Tag	Valid	Byte 0	Byte 1	Byte 2	Byte 3
0	7A	1	09	EE	12	64	00	0	99	04	03	48
1	02	0	60	17	18	19	7F	1	FF	BC	0B	37
2	55	1	30	EB	C2	0D	0B	0	8F	E2	05	BD
3	07	1	03	04	05	06	5D	1	7A	08	03	22

Acronyms:

VPN Virtual Page Number

PPN Physical Page Number

Part 1:

1. How many pages are there in Virtual Memory?
2. How many pages are available in Physical Memory?
3. How many page table entries are present in Page table?
4. Calculate the number of bits required for page offset, TLB index and TLB tag.
5. Calculate the number of bits required for block offset, cache index and cache tag.

Part 2:

For the following virtual address, fill the table entries by indicating Virtual page number, the TLB entry accessed and the corresponding physical address. Indicate whether the TLB misses and whether a page fault occurs.
" If there is a page fault, enter "-" for "PPN" and leave part 2 blank.

Virtual address: `0x1A9F4`

Virtual address in binary format is

17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	0	1	0	1	0	0	1	1	1	1	1	0	1	0	0

1. Address translation

Parameter	Value
VPN	0x
TLB Index	0x
TLB Tag	0x
TLB Hit? (Y/N)	
Page Fault? (Y/N)	
PPN	0x

2. Physical address in binary format (one bit per box)

11	10	9	8	7	6	5	4	3	2	1	0

3. For the above address, fill the following table according to cache information given. If there is a cache miss, enter "-" for "Cache Byte returned.

Parameter	Value
Block Offset	0x
Cache Index	0x
Cache Tag	0x
Cache Hit? (Y/N)	
Value of Cache Byte Returned	0x

Question No. 4 [Marks: 15]

Consider the execution of the following code snippet on MIPS 5-stage pipelined processor with hazard detection and **forwarding fully implemented**. Static Branch prediction is applied which always assumes branch to be **not taken**.

Instruction 1	Lw R5,0(R6)
Instruction 2	Sub R7,R7,1
Instruction 3	Lw R2, 4(R3)
Instruction 4	Lw R1,0(R3)
Instruction 5	Add R4,R2,R1
Instruction 6	Beq R7,R5,30
Instruction 7	Sw R4, 0(R3)

Initial values for the registers and memory locations are as follows

R6 = 8, R7=3, R3=0

Memory[0] = 10, Memory[4] = 20,Memory [8] = 30

a. How many cycles will it take to complete the correct execution of the above code (including stalls)? _____

b. Consider the execution of the above snippet in the 8th Clock Cycle. Which instructions are executing at each of the pipeline stages? Fill in the following table to answer the question. Write instruction number and if a stall is at some stage then simply write **stall** for that stage**.**

Stage	Executing Instruction
IF	
ID	
EXE	
MEM	
WB	

c. In the **start** of the ***8thClock Cycle***, what will be the contents of each of the pipeline separation registers? Fill in the following tables to answer the question. You need to explicitly state names and values of ***ALL*** the variables in each register. For values that are not known to you, you can write them in terms of English phrases. For example, First entry of the IF/ID register has been filled as a guideline. You don't know the exact value of the Program Counter (PC) yet you can state that it is the value of the PC that points to instruction x plus 4 (Here you should replace "x" by the correct instruction number). All other variables, whose values can be determined, should state exact values. For example, if EXE/MEM register contains the result of the addition of instruction 5, you should state the Variable Name as "Result of ALU" and Value as "30"

IF/ID Register Contents

Variable Name	***Value***
PC	***(Program counter value of the instruction x) + 4***

EXE/MEM Register Contents

Variable Name	***Value***

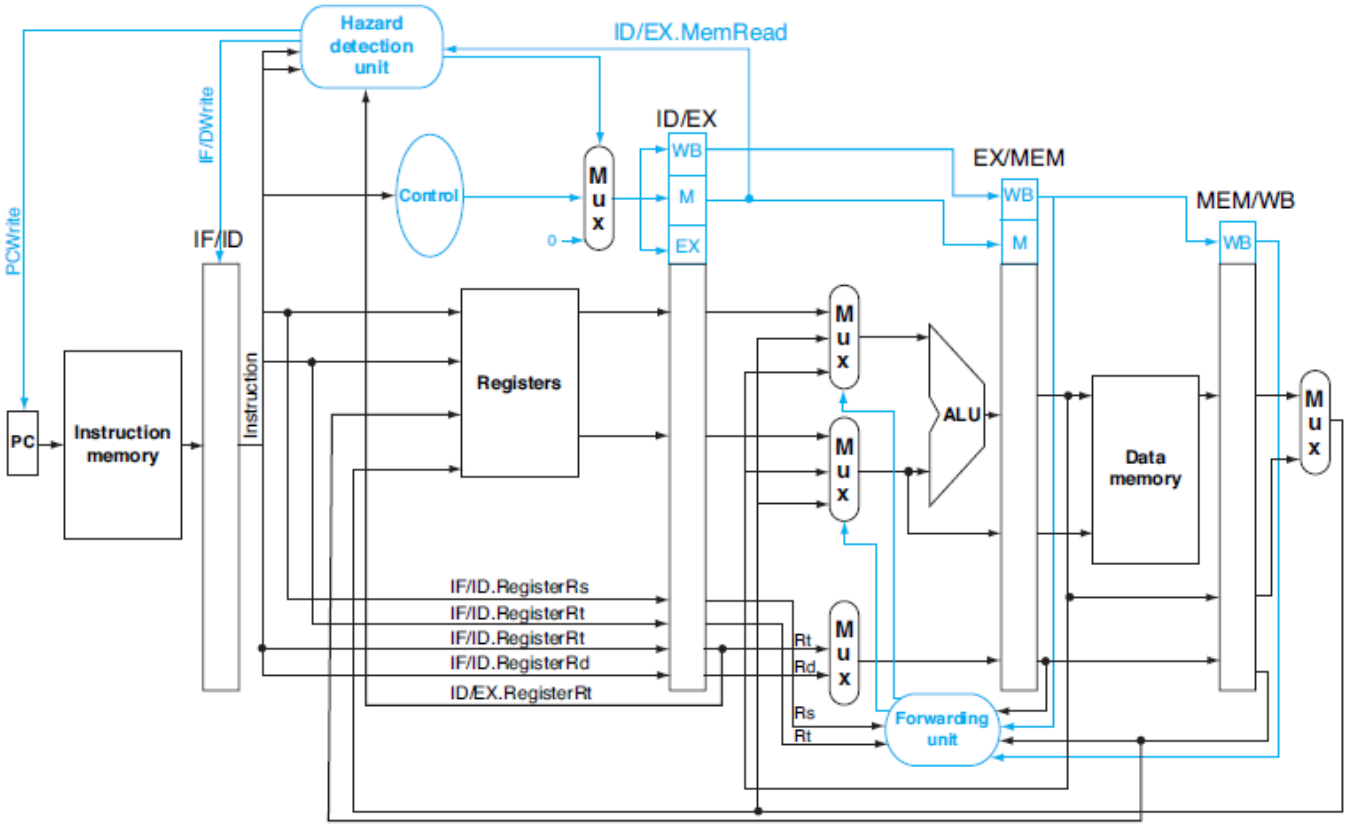
ID/EXE Register Contents

Variable Name	***Value***

MEM/WB Register Contents

Variable Name	***Value***

Pipelined Data Path with Forwarding and Hazard Detection Units



Information of control signals