National University of Computer and Emerging Sciences, Lahore Campus

	Course Name:	Computer Architecture	Course Code:	EE204
	Program:	BS (Computer Science)	Semester:	Fall2019
	Duration:	180 Minutes	Total Marks:	75
	Paper Date:	18-12-2019	Weight	45
	Exam Type:	Final	Page(s):	8

Student : Name:_______________________________ Roll No._________________ Section:________

Instructions:
1. Attempt all question in the provided space. You can use rough sheets but it should not be attached. Make assumption if something is missing and write it clearly.
2. Answers attempted using pencil will not be considered. Use only black or blue pen.

Question 1 [15] (Marks will not be awarded for cutting and over-writing)

1. The type of hazard that occurs when branch instruction is dependent on previous instruction is called__________.
2. EPC register in case of an exception records__________
3. Decimal representation of 110111.101 is ____________
4. Ideal speedup from pipelining is equal to ____________
5. Simplified expression of A+ A'B + A'B' is __________
6. DRAM is fast but expensive as compared to SRAM. **True/False**
7. Shift Logic Left (sll) is an R-type instruction. **True/False**
8. VLIW is less efficient but require less hardware as compared to Superscalar. **True/False**
9. Spatial locality can be improved by using LRU replacement policy. **True/False**
10. J-type instructions do not have the destination register. **True/False**
11. In case of an exception at decode stage, instructions at Ex, Mem and WB will be
    A. Discarded and PC moves to ISR
    B. completed before PC moves to ISR
    C. PC moves to ISR and instruction complete alongside
    D. None of the above
12. In case of Virtual memory, if address translation is not available in TLB then it is
    A. Page fault
    B. TLB miss
    C. Page table hit
    D. Both A and B
13. To read and write registers to/from Register File we need
    A. Decoders, multiplexers
    B. Encoders, multiplexers
    C. decoders encoder and multiplexers
    D. only decoders
14. Biased value of -5 is
    A. 122
    B. 123
    C. 132
    D. 133
15. Multiplier implemented using adder is a
    A. Combinational circuit
    B. Sequential circuit
    C. Can be implemented as combinational or sequential
    D. None of the above

Question 2 [2+6+2]

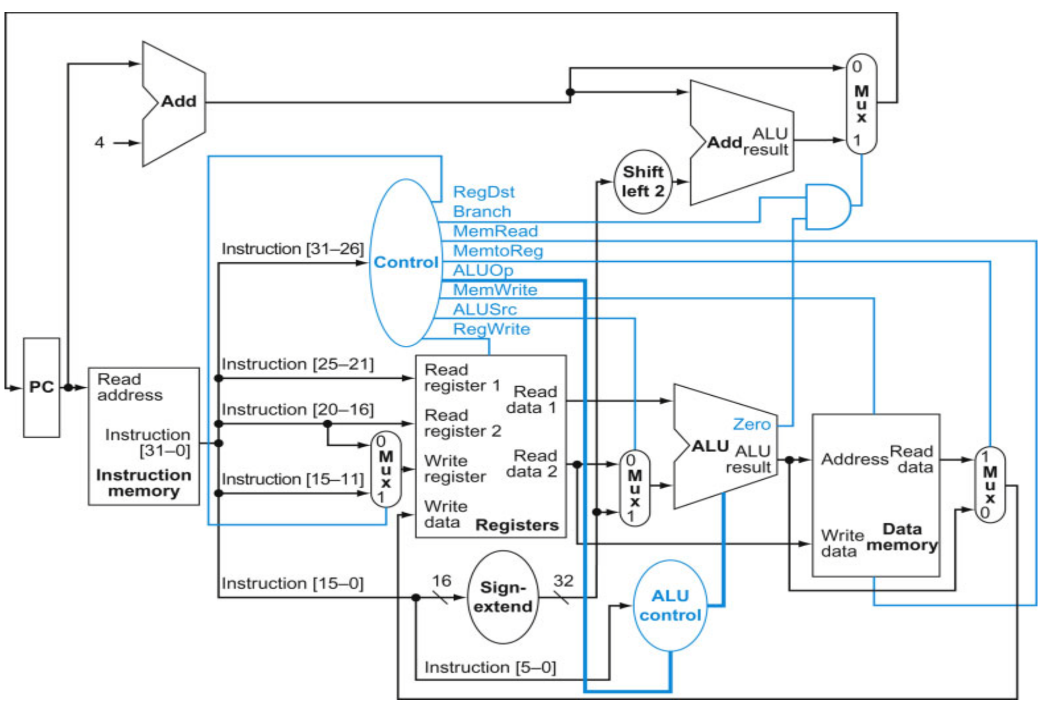
Suppose we want to add support for the following instruction in our MIPS architecture.

Instruction	Operation being performed
BEQM RT, RS, offset	if Rt = =Mem[Rs + offset] then PC = Rt *4

The BEQM instruction is a conditional branch which compares the value of RT register with value at memory location determined by adding RS and offset (Immediate value). If the values are equal then PC register is updated by the value of RT multiplied by 4. Sample instruction is as follows:
BEQM R4, R7, 88

a. For the above instruction, write the MIPS instruction in binary value.

b. What must be changed in the pipelined data-path to add this instruction to the MIPS Instruction Set Architecture? Use space provided above and around the diagram to draw any new required resources/stages. You can draw lines from/to the diagram for input/output of these units. Draw neatly so that it's easier for the instructor to understand your logic.



Brief description of the changes being made:

c. Which of the following hazards (if any) will occur if the above mentioned new instruction is added to the 5-stages data path? If any hazard occurs, then how many stalls will be required to avoid any wrong execution of the code segments? Answer the question by filling the following table.

Hazard Type	**Will the hazard occur if new instruction is added? Yes/No**	**No. of stalls required in case of hazard**	**Brief Description**
Data Hazard			
Control Hazard			
Structural Hazard			

Question 3[5+8+7]

Consider the following program fragment executing on a MIPS pipeline *with multiple execution units*. Execution stage takes a variable number of cycles, depending on the functional unit used.

Functional unit	Number of EX cycles
Integer ALU	1
Floating Point Add	2
Floating Point Multiply	3

Data forwarding is implemented.

```
1.  Loop:  Ld R2, R1, 40
2.         S.D F1, R2, 8
3.         Ld.D F3, R2, 50
4.         MUL.D F4, F1, F3
5.         S.D F4, R2, 16
6.         Subi R1, R1, 8
7.         Beq R1, R0, Loop
```

Note: R registers are for integer and F registers are for floating. Integer ALU is used for integer operations and Floating point units are used if the operands are floating point numbers.

a. For the correct execution of the above program, inserts stall between instructions to avoid data and control hazards. Ignore structural hazards!

b. Unroll the loop for level 2 (two iterations only), add stalls and then reschedule to remove as many stalls as possible. Ignore structural hazards. You have to consider only data and control hazards. Each instructions take multiple cycle in execution as stated in part a. Remove false dependencies for better scheduling.

Code after loop unrolling	**Code with added stalls**	**Optimized schedule of instruction with remaining stalls**

c. Consider the given program (used in part a) to be executed on 2-issue superscalar processor with following specification.

- There are three type of execution units with latency as given in part a, however we have 2 integer ALU, 1 floating Adder and 1 floating Multiplier. Other than execution there are two units in each stage and all instructions take 1 cycle in stages other than execution.
- Full Forwarding is implemented
- Do not rename for this part. In case of false dependencies, write back should be in order. No need to wait in decode stage. Assume buffers available between different stages!

Show two iterations of the code given in part A

Cycle No.	IF		ID		EX				MEM		WB	
					IAdd	IAdd	FADD	FMUL				
1												
2												
.												
.												
.												
.												
.												
.												
.												
.												
.												
.												
.												
.												
.												
.												
.												
.												
.												
.												
.												
.												
.												
.												
.												

Question 4 [20]

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. VPN and PPN stand for Virtual Page Number and Physical Page Number.

- The memory is byte addressable
- Virtual addresses are 24 bits wide. (six hex digits)
- Physical addresses are 16 bits wide. (four hex digits)
- The page size is 2048 bytes.
- The TLB is 4-way set associative with 64 blocks.
- The Data Cache (hereafter called cache) is 2-way set associative, with a 16-byte block size and 32blocks.

In the following tables, all numbers are given in hexadecimal. You are provided below with **initial blocks of TLB**, Page Table entries for selected pages (in table titled "Page Table"), and **contents of initial blocks of cache**:

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

Page Table					
VPN	**PPN**	**Valid**	**VPN**	**PPN**	**Valid**
000	-	0	550	-	0
2AB	D	1	55A	-	0
312	A	1	561	3	1
31B	-	0	5C9	-	0
320	9	1	601	4	1
37A	-	0	699	-	0
393	E	1	6A1	6	1
3AB	-	0	70B	-	0
433	-	0	712	5	1
45C	-	0	72A	1	1
480	F	1	733	2	1
48D	-	0	740	-	0
513	0	1	781	B	1
529	-	0	79B	-	0
532	-	0	872	C	1
54A	8	1	ABC	-	0

2-way Set Associate Cache																		
			Bytes															
Index	**Tag**	**Valid**	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
0	7A	1	09	EE	12	64	99	04	03	48	99	04	03	48	09	EE	12	64
	7F	1	23	54	44	12	55	66	88	99	AA	AB	CA	15	12	45	44	6A
1	07	1	60	17	18	19	FF	BC	0B	37	FF	BC	0B	37	60	17	18	19
	7F	1	32	54	65	76	66	33	55	77	88	34	4D	5F	FF	99	67	23
2	55	1	30	EB	C2	0D	8F	E2	05	BD	8F	E2	05	BD	30	EB	C2	0D
	0B	0	BC	0B	37	FF	8F	E2	05	12	64	99	FF	BC	0B	34	21	69
3	07	1	03	04	05	06	7A	08	03	22	7A	08	03	22	03	04	05	06
	5D	1	98	65	32	65	98	65	34	54	65	76	94	65	77	5B	4D	45

Part 1: (8 marks)

1. How many virtual pages are there?			
2. How many physical pages are there?			
3. How many full Page Table Entries (PTEs) are present in the Page Table for any OS process?			
4. How many sets in TLB and cache?			
5. Calculate the number of bits required for page offset, TLB index, and TLB tag.	Page offset bits		
	TLB index bits		
	TLB tag bits		
6. Calculate the number of bits required for block offset, cache index, and cache tag.	Block offset bits		
	Cache index bits		
	Cache tag bits		

Part 2: (8 marks)

For the following virtual address, write the corresponding physical address. Also, indicate whether the TLB misses and whether a page fault occurs. If there is a page fault, enter "-" for "Physical Address". Write the physical address as a hex value.

Virtual Address	TLB Hit? (Y/N)	Page Fault? (Y/N)	Physical Address (hex value)
0x393A76			
0x8720A8			
0x532CAB			
0x601786			

Part 3: (4 marks)

For the following physical addresses, fill the following table according to cache information given. If there is a cache miss, enter "-" for "Value of Cache Byte Returned".

Physical Address	Cache Hit? (Y/N)	Value of Cache Byte Returned (hex value)
0x7F19		
0x0738		
0x340B		
0x5D30		

Question 5 [4+6]

A. A compiler designer is trying to decide between two code segments for particular machine. There are two classes of available instructions: A and B. CPIs for classes A and B are 2 and 3 respectively. Instruction count for two code segments are provided in the table below:

	Instruction Counts for Instruction Classes	
Code Sequence	A	B
1	300	500
2	700	200

i. How many cycles are required for each code sequence?

Total cycles for code sequence #1:

Total cycles for code sequence #2:

ii. What is the CPI for each code sequence?

CPI for code sequence #1:

CPI for code sequence #2:

B. Consider a processor with a 16 KB L1 on-chip cache. The miss rate for this cache is 3% and the hit time is 2 Clock Cycles (CCs). The processor also has an 8 MB, off-chip L2 cache. 95% of the time, data requests to the L2 cache are found and the hit time for L2 cache is 15 CCs. If the data is not found in the L2 cache, a request is made to a 4 GB main memory. The time to service a memory request is 400 CCs.

i. What is the average memory access time in terms of clock cycles?

ii. Calculate the execution time of a program that has a total of 10,000 instructions? 60% of these instructions are load and store instructions while others are compute instructions. The value of base CPI (Cycles Per Instruction) is 2. The clock speed for the processor is 1 GHz. Assume that all instructions are fetched from L1 cache.