BITS-Pilani, Hyderabad Campus
Computer Architecture (CS F342), Second Semester 2020-21
Comprehensive Exam

Section 1/A

Question1

Variant 1

The different stages of a 7-stage processor have a delay of 5ns, 4ns, 3ns, 3ns, 2ns, 4ns and 3ns. The pipelined version of the processor uses pipeline registers with delay of 0.2ns. What is the speed up of a 7-stage pipelined processor when compared to the 7-stage single cycle processor for executing 100 instructions? (Assume that there are no stalls)

Variant 2

The different stages of a 7-stage processor have a delay of 5ns, 4ns, 3ns, 3ns, 2ns, 4ns and 3ns. The pipelined version of the processor uses pipeline registers with delay of 0.3ns. What is the speed up of a 7-stage pipelined processor when compared to the 7-stage single cycle processor for executing 100 instructions? (Assume that there are no stalls)

Question2

Variant 1

A program with large number of instructions (ideal CPI without any stalls is 1) is to be executed on a 5-stage RISC-V like pipelined processor with forwarding (from EX/MEM and MEM/WB stages to input of EX stage). This program execution leads to execution of **30% load instructions**, **10% branch instructions**, **25% store instructions** and **35% arithmetic instructions**. The 5-stage pipelined processor uses static branch NOT TAKEN predictor which is **20% accurate**. Out of all load instructions, If **20% of them are followed by instruction with data dependency** (i.e. destination register in load instruction is same as source register in immediate next instruction), Determine the CPI. (Assume ideal CPI is 1 and branch condition is evaluated in EX stage).

Variant 2

A program with large number of instructions (ideal CPI without any stalls is 1) is to be executed on a 5-stage RISC-V like pipelined processor with forwarding (from EX/MEM and MEM/WB stages to input of EX stage). This program execution leads to execution of **30% load instructions**, **20% branch instructions**, **20% store instructions** and **30% arithmetic instructions**. The 5-stage pipelined processor uses static branch **NOT TAKEN** predictor which is **20% accurate**. Out of all load instructions, If **20% of them are followed by instruction with data dependency** (i.e. destination register in load instruction is same as source register in immediate next instruction), Determine the CPI. (Assume ideal CPI is 1 and branch condition is evaluated in EX stage).

Question 3

Variant 1

Refer to the RISC V code below.

lw t1, 4(t2);
add t2,t3,t1;
add t3,t2,t1;
sw t3, 0(t2);
For a 5-stage pipeline with forwarding (only from EX/MEM pipeline register to input of EX stage), how many (overall) nops should be added by the complier to avoid hazards? (Assume that a write and read to a single register can complete in one cycle.)

Variant 2

Refer to the RISC V code below.
lw s1, 4(s2);
add s2,s3,s1;
add s3,s2,s1;
sw s3, 0(s2);
For a 5-stage pipeline with forwarding (only from EX/MEM pipeline register to input of EX stage), how many (overall) nops should be added by the complier to avoid hazards? (Assume that a write and read to a single register can complete in one cycle.)

Question 4

Variant 1

If a 4-way associative cache has 2K blocks, with 4 words in each block, then the total no of index bits used to address the cache is equal to

Variant 2

If a 4-way associative cache has 1K blocks, with 4 words in each block, then the total no of index bits used to address the cache is equal to

Question 5

Variant 1

Given 32-bit address, byte addressable memory, word size of 16 bits, 2-way associative cache with 512 blocks and block size as 1 word; The total size of cache (including the data, tag and valid) is ________bytes

Variant 2

Given 32-bit address, byte addressable memory, word size of 16 bits, 4-way associative cache with 512 blocks and block size as 1 word; The total size of cache (including the data, tag and valid) is ________bytes

Question 6

Variant 1

A DRAM needs 1 clock cycle to read the address, 15 clock cycles for enabling the corresponding memory and 1 clock cycle to send/write the data on the bus. Assume one word can be accessed from the DRAM memory at the same time, the bus width between cache and main memory is 2 words and the block size of cache is four words. The miss penalty for the above specifications is ____ clock cycles

Variant 2

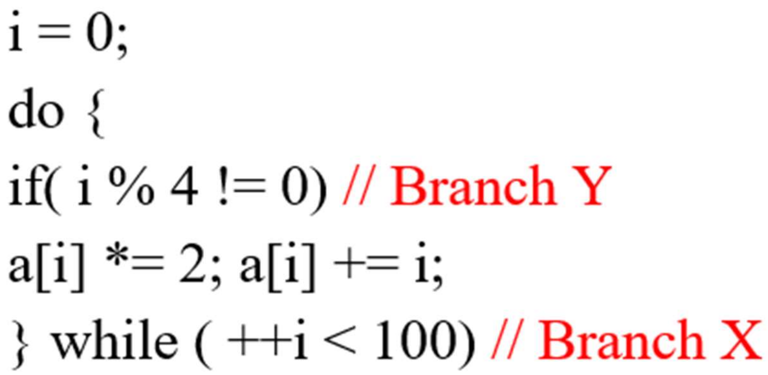
A DRAM needs 1 clock cycle to read the address, 15 clock cycles for enabling the corresponding memory and 1 clock cycle to send/write the data on the bus. Assume two words can be accessed from the DRAM memory at the same time, the bus width between cache and main memory is 1 word and the block size of cache is four words. The miss penalty for the above specifications is ____ clock cycles

Section 2/B

Question 1

Variant 1

Consider the code shown in figure below.



```
i = 0;
do {
if( i % 4 != 0) // Branch Y
a[i] *= 2; a[i] += i;
} while ( ++i < 100) // Branch X
```

When this code is translated to assembly, the branch Y will be taken if (i%4 ! = 0) else branch will not be taken. The designer of the processor on which the assembly code is to be executed has two predictors to choose from

(a) 1-bit dynamic branch predictor which is initialized to Taken state.

(b) 2-bit dynamic branch predictor (saturation counter) initialized to Strongly Taken State

Determine the prediction accuracies for both the predictors.

(Assume unlimited entries for the branch history table) **[6 Marks]**

Question 2

Variant 1

A processor follows a memory hierarchy involving TLB, page table and two levels of Cache. Assume that (i) the cache uses physical addresses, (ii) the CPU stalls until the data is delivered, (iii) everything fits into the main memory. (iv) the hardware accesses the page table and updates TLB in case of TLB Miss. Assume page fault never occurs.

The access latency and the local hit rate for different units in the hierarchy is shown below:

Unit	Additional Access Latency	Local Hit rate
TLB	1 Clock cycle	95%
L1 Cache	1 Clock cycle	95%
L2 Cache	20 Clock Cycles	60%
Main Memory Access (for Data)	100 Clock Cycles	100%
Page Table Access and TLB Update	120 Clock Cycles	100%

Based on the above data, determine the average time required for the address translation alone, the average memory access time (for data) once the physical address is available and overall Average memory access time (address translation and data access). (Round up your answer to the nearest integer) [6 Marks]

Variant 2

A processor follows a memory hierarchy involving TLB, page table and two levels of Cache. Assume that (i) the cache uses physical addresses, (ii) the CPU stalls until the data is delivered, (iii) everything fits into the main memory. (iv) the hardware accesses the page table and updates TLB in case of TLB Miss. Assume page fault never occurs.
The access latency and the local hit rate for different units in the hierarchy is shown below:

Unit	Additional Access Latency	Local Hit rate
TLB	1 Clock cycle	96%
L1 Cache	1 Clock cycle	95%
L2 Cache	20 Clock Cycles	80%
Main Memory Access (for Data)	100 Clock Cycles	100%
Page Table Access and TLB Update	100 Clock Cycles	100%

Based on the above data, determine the average time required for the address translation alone, the average memory access time (for data) once the physical address is available and overall Average memory access time (address translation and data access). (Round up your answer to the nearest integer) [6 Marks]

Question 3
Variant 1
Consider the sequence of instructions below:

```
lw t0, 8(s0)
add t4, t2, t0
add t5, t1, t4
sw t4, 8(s0)
```

The above instructions are to be executed on 3 variants of a RISC V pipelined processor:
(a) Processor A: A processor with no forwarding with clock cycle time as 300ps.
(b) Processor B: A processor which allows forwarding from EX/MEM pipeline register (to input of EX stage) only, with clock cycle time as 310ps
(c) Processor C: A processor which allows forwarding from EX/MEM pipeline register as well as MEM/WB pipeline register (to input of EX stage) with clock cycle time of 350ps.
(Assume Register file can be written in first half and read in the second half of same clock cycle)

For executing the above instruction sequence, answer the following (fill in the answers in the space provided), [3 x 4 = 12Marks]

Variant 4

Consider the sequence of instructions below:

lw t0, 8(s0)
add t4, t2, t0
add t5, t1, t4
sw t4, 8(s0)

The above instructions are to be executed on 3 variants of a RISC V pipelined processor:

(a) Processor A: A processor with no forwarding with clock cycle time as 200ps.

(b) Processor B: A processor which allows forwarding from EX/MEM pipeline register (to input of EX stage) only, with clock cycle time as 210ps

(c) Processor C: A processor which allows forwarding from EX/MEM pipeline register as well as MEM/WB pipeline register (to input of EX stage) with clock cycle time of 250ps.

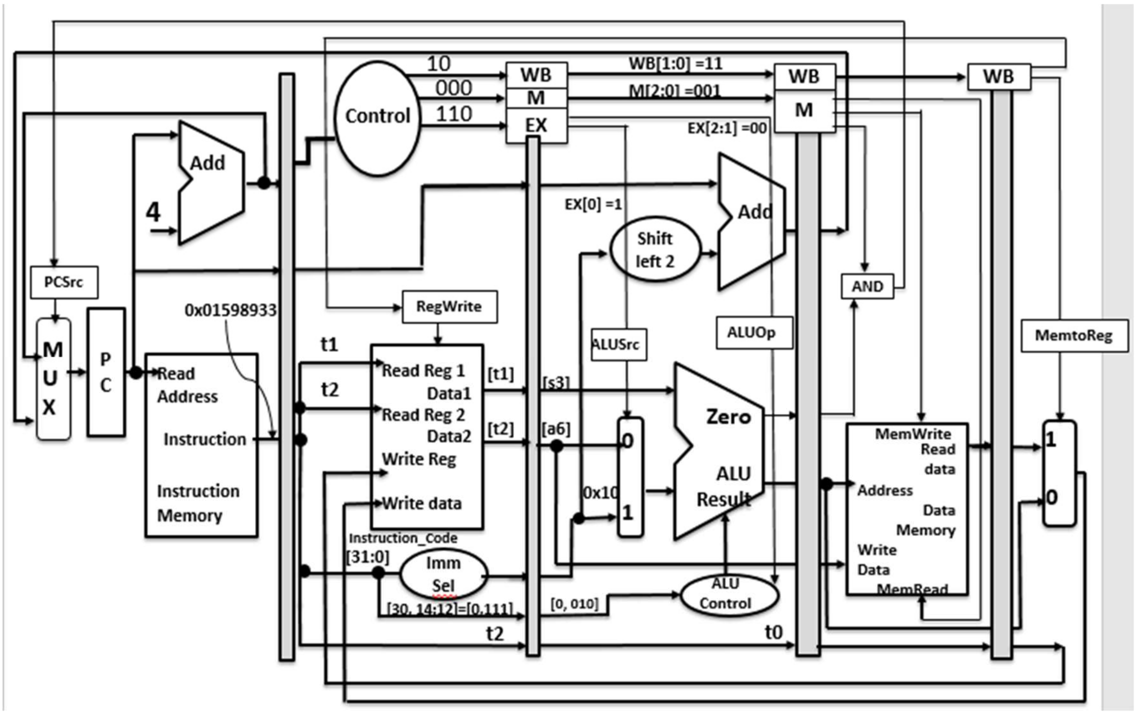
(Assume Register file can be written in first half and read in the second half of same clock cycle)

For executing the above instruction sequence, answer the following (fill in the answers in the space provided), [3 x 4 = 12Marks]

Question 4

Variant 1

Consider the partial datapath of the RISC V pipelined processor (which supports lw, sw, R-type and Beq instructions only) as shown in below figure.



The control signal mapping, ALUOp and function codes for instructions are also shown below:

EX[2:1	**EX[0]**	**M[2]**	**M[1]**	**M[0]**	**WB[1]**	**WB[0]**
ALUop[1:0]	ALUSrc	Branch	MemWrite	MemRead	RegWrite	MemtoReg

Instruction Opcode	**Operation**	**ALUOp[1:0]**	**Func7**	**Func3**
0000011	Load Word	00	xxxxxxx	010
0100011	Store Word	00	xxxxxxx	010
1100011	Branch on equal	01	xxxxxxx	000
0110011	Add	1x	0000000	000
0110011	Sub	1x	0100000	000
0110011	And	1x	0000000	111
0110011	Or	1x	0000000	110

There are five instructions at different stages of the pipeline out of which instructions in IF, ID and EX stage are unidentified. Based Determine the instructions in IF, ID and EX pipeline stages. **[12 Marks]**

Stage	**write the complete instruction in that pipeline stage in the space provided below**
Instruction Fetch (**IF**)	
Instruction Decode (**ID**)	
Execute (**EX**)	

<u>Question 5</u>
<u>Variant 1</u>

A processor has the following specifications:
Word size is 16-bits;
No. of blocks in cache is 8;
Cache block size is 1 word;
Cache uses LRU replacement policy and Write back scheme;
Main memory is byte addressable;
Cache is 4-way associative cache.
For the given sequence of operations, complete the table with the final contents of cache (by specifying the address, data, Valid bit and Dirty bit values) after all operations have been complete.

Sequence of operations:
Read 0xAAAA from address 0xF224
Read 0x4949 from address 0x9D96
Read 0x1188 from address 0x2AE6
Read 0x5432 from address 0xEB34
Read 0x2435 from address 0x48D0
Read 0x4444 from address 0x6B02
Read 0x2018 from address 0xB3A6
Read 0x1777 from address 0x61D4
Write 0x9432 to address 0x1900)

Read 0x2435 from address 0x48D0
Read 0x8204 from address 0xFAE6
Write 0xACEF to address 0x1604

Write the full memory address (instead pf just the tag bits so you can easily write it in hexadecimal.), data, valid bit and dirty bit in each cache entry corresponding to the LRU count (the lowest LRU count value represents the most recently used (MRU) entry, and highest LRU count value represents the least recently used (LRU) entry). Assume that all cache lines are invalid when the sequence of operations starts. **[16 Marks]**

Set	Address (in hexadecimal)	Data in hexadecimal	Valid bit (1 or 0)	Dirty Bit (1 or 0)	LRU count
0					
0					
0					
0					
1					
1					
1					
1					

<u>Variant 2</u>

A processor has the following specifications:
Word size is 16-bits;
No. of blocks in cache is 8;
Cache block size is 1 word;
Cache uses LRU replacement policy and Write back scheme;
Main memory is byte addressable;
Cache is 4-way associative cache.
For the given sequence of operations, complete the table with the final contents of cache (by specifying the address, data, Valid bit and Dirty bit values) after all operations have been complete.

Sequence of operations:
Read 0xAAAA from address 0xF22C
Read 0x4949 from address 0x9D96
Read 0x1188 from address 0x2AEE
Read 0x5432 from address 0xEB34
Read 0x2435 from address 0x48D8
Read 0x4444 from address 0x6B0A
Read 0x2018 from address 0xB3AE
Read 0x1777 from address 0x61DC
Write 0x9432 to address 0x1900
Read 0x2435 from address 0x48D8
Read 0x8204 from address 0xFAE6
Write 0xACEF to address 0x1604

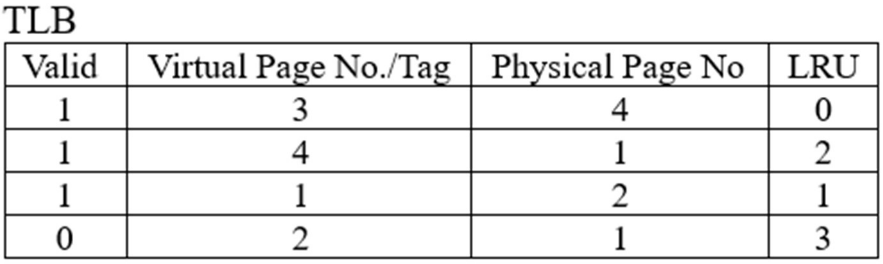
Write the full memory address (instead pf just the tag bits so you can easily write it in hexadecimal.), data, valid bit and dirty bit in each cache entry corresponding to the LRU count (the lowest LRU count value represents the most recently used (MRU) entry, and highest LRU count value represents the least recently used (LRU) entry). Assume that all cache lines are invalid when the sequence of operations starts. **[16 Marks]**

Set	**Address (in hexadecimal)**	**Data in hexadecimal**	**Valid bit (1or 0)**	**Dirty Bit (1 or 0)**	**LRU count**
0					
0					
0					
0					
1					
1					
1					
1					

Question 6
Variant 1

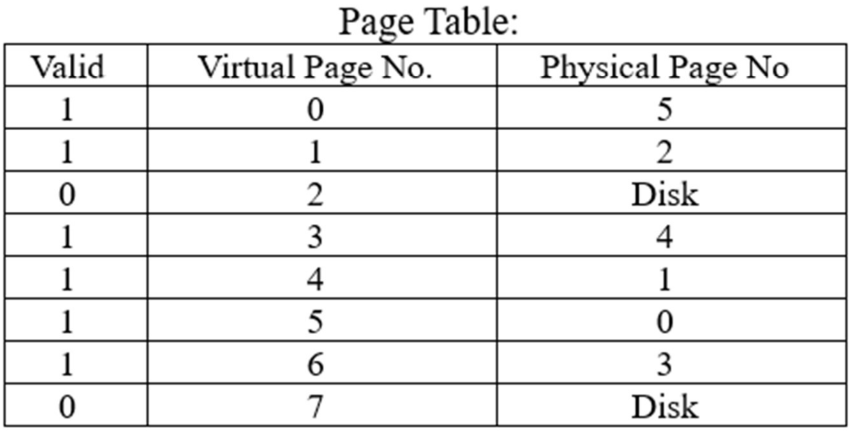
A program is running on a computer with four-entry fully associative translation lookaside buffer (TLB) with true LRU replacement (LRU count 0 indicates most recently used. LRU Count 3 indicates least recently used). Assume page size as 8KB and byte addressable main memory. If pages must be brought in from the disk, give them the next largest unused physical page number.

The initial status of TLB and Page Table are given below:



TLB

Valid	Virtual Page No./Tag	Physical Page No	LRU
1	3	4	0
1	4	1	2
1	1	2	1
0	2	1	3



Page Table:

Valid	Virtual Page No.	Physical Page No
1	0	5
1	1	2
0	2	Disk
1	3	4
1	4	1
1	5	0
1	6	3
0	7	Disk

Assume that the following stream of addresses (represented in hexadecimal) are accessed by the processor in the same order:

0x8DAC
0xAA98
0x4C23
0x4234

For each of the above memory access determine the virtual page no, whether the TLB access will result in Hit or Miss and whether the page resides in main memory (by indicating Hit or fault). **[10 Marks]**

Address	**Virtual Page No/Tag in decimal**	**TLB Access (Write Hit or Miss as your answer in the blanks provided below)**	**Page availability in Main memory (Write Hit or Fault as your Answer in the blanks provided below)**
0x8DAC			
0xAA98			
0x4C23			
0x4234			

Also determine the final TLB status (Corresponding to the LRU count) after all the three memory accesses.

Final TLB Status: [Marking scheme: 1 Marks for each row, partial marks may be given for each row only if the virtual page no of that row is correct.]

Valid	**Virtual Page No/Tag in decimal**	**Physical Page No. in Decimal**	**LRU**
1			0
1			1
1			2
1			3

Detailed Solution

(13-bits for page offset)

Address	**Virtual Page number/Tag**	**TLB Hit/Miss**	**Page Hit/fault**		**TLB status after access**			
					Valid	**Tag**	**PPN**	**LRU**

<u>Variant 2</u>

A program is running on a computer with four-entry fully associative translation lookaside buffer (TLB) with true LRU replacement (LRU count 0 indicates most recently used. LRU Count 3 indicates least recently used). Assume page size as 8KB and byte addressable main memory. If pages must be brought in from the disk, give them the next largest unused physical page number.

The initial status of TLB and Page Table are given below:

TLB

Valid	Virtual Page No./Tag	Physical Page No	LRU
1	3	4	0
1	4	1	2
1	1	2	1
0	2	1	3

Page Table:

Valid	Virtual Page No.	Physical Page No
1	0	5
1	1	2
0	2	Disk
1	3	4
1	4	1
1	5	0
1	6	3
0	7	Disk

Assume that the following stream of addresses (represented in hexadecimal) are accessed by the processor in the same order:

0x3DAC
0xAA98
0x4C1C
0x4234

For each of the above memory access determine the virtual page no, whether the TLB access will result in Hit or Miss and whether the page resides in main memory (by indicating Hit or fault). **[10 Marks]**

Address	**Virtual Page No/Tag in decimal**	**TLB Access (Write Hit or Miss as your answer in the blanks provided below)**	**Page availability in Main memory (Write Hit or Fault as your Answer in the blanks provided below)**
0x3DAC			
0xAA98			
0x4C1C			
0x4234			

Also determine the final TLB status (Corresponding to the LRU count) after all the three memory accesses.

Final TLB Status:

Valid	**Virtual Page No/Tag in decimal**	**Physical Page No. in Decimal**	**LRU**
1	2	6	0
1	5	0	1
1	1	2	2
1	3	4	3

Address	**Virtual Page number/Tag**	**TLB Hit/Miss**	**Page Hit/fault**		**TLB status after access**			
					Valid	**Tag**	**PPN**	**LRU**