EXERCISES/QUESTIONS TAKEN FROM PREVIOUS ONLINE EXAMS

Question 1 (Open Text Answer) from 01/07/2021

Let's consider a directory-based protocol for a distributed shared memory system with 4 Nodes (N0, N1, N2, N3) where: **Directory N1 Block B1 | State: Uncached | Sharer Bits: ---- |**

*After a **Write Miss on B1 from node N2,** please answer to the following questions:*
a) What are the home node, the local node and the remote node(s)?
b) What are the messages sent among the nodes?
c) Which is the state of the block B1 in the home directory?
d) Which is the state of the block B1 in the local cache?

(OPEN TEXT ANSWER: max 100 words)
2 points

Feedback:

a) **N1** is the home node of B1 and **N2** is the local node (requestor); There are no remote cache(s) because the block in **Uncached**;

b) Being the block B1 in the **Uncached** state, the only copy of the block in the home memory is up to date in N1. Due to the **Write Miss on B1** from the local node N2 to home node N1, the requested data are sent by a **Data Value Reply** from home memory N1 to the local cache of node N2 which becomes the **owner.**

c) The state of the block B1 in the home directory N1 becomes **Modified** and the **owner is** N2: **Directory N1 Block B1 | State: Modified | Sharer Bits: 0010.**

d) The state of the block B1 in the local cache C2 of N2 also becomes **Modified**.

EXERCISES/QUESTIONS TAKEN FROM PREVIOUS ONLINE EXAMS

Question 2 (format Multiple Choice – Single answer) from 01/07/2021

Let's consider the following code executed by a Vector Processor with a Vector Register File composed of 32 vectors of 8 elements per 64 bits/element with 1 Load/Store Vector Unit and 2 ALU Vector Units **with operation chaining:**

```
L.V V1, RX          # load vector from memory address RX into V1
L.V V2, RY          # load vector from memory address RY into V2
MULVV.D V3, V1, V2  # multiply vectors
ADDVV.D V5, V3, V4  # add vectors
S.V V5, RZ          # store vector V5 into memory address RZ
```

How many convoys? How many clock cycles to execute the code?
(SINGLE ANSWER)
2 points

Answer 1: 4 convoys; 32 clock cycles

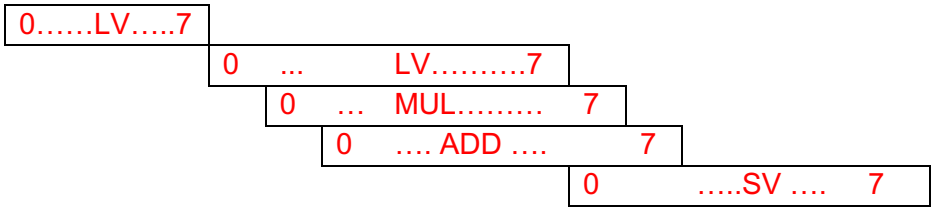
Answer 2: 3 convoys; 24 clock cycles

Answer 3: 2 convoy; 16 clock cycles

Answer 4: 5 convoys; 40 clock cycles

Feedback:

Answer 2: 3 convoys; 24 clock cycles **(TRUE)**



EXERCISES/QUESTIONS TAKEN FROM PREVIOUS ONLINE EXAMS

Question 3 (format Multiple Choice – Multiple answer) from 01/07/2021

*How does a MESI write-invalidate write-back protocol manage a **Write Hit** on an Exclusive cache block?*
(MULTIPLE ANSWERS)
2 points

Answer 1: The status of the cache block becomes Modified

Answer 2: The cache block is retrieved from memory

Answer 3: An invalidate is broadcasted on the bus to the other copies of the block

Answer 4: The cache block is retrieved from another cache

Answer 5: The cache block is overwritten in the processor's cache

Feedback:

Answer 1: The status of the cache block becomes Modified **(TRUE)**

Answer 5: The cache block is overwritten in the processor's cache **(TRUE)**

On a Write Hit on an Exclusive cache block means that the block is clean and there are no other copies of the block in other caches.
Therefore, there is no need to send an invalidate signal on the snooping bus to other caches.

We have that:

- *The processor's cache is updated by writing the new data value.*
- *The status of the cache block becomes Modified*

EXERCISES/QUESTIONS TAKEN FROM PREVIOUS ONLINE EXAMS

Question 3 (format open text)

Please consider the following assembly code:

```
Loop:   LD F0, 0(R1)
        FADD F3, F0, F1
        FMULT F5, F0, F1
        FADD F7, F3, F5
        SD F7, 0(R1)
        LD F2, 4(R1)
        FADD F4, F2, F1
        FMULT F6, F2, F1
        FADD F8, F4, F6
        SD F8, 4(R1)
        ADD R1, R1,8
        BNE R1 R2 Loop
```

Details about the **4-issue VLIW** machine with 4 fully pipelined functional units:

- 1 Integer ALU with 1 cycle latency to next Integer/FP and with 2 cycle latency to next Branch
- 1 Memory Unit with 2 cycle latency
- 1 FP ADDER with 3 cycle latency
- 1 FP MULTIPLIER with 3 cycle latency

The branch is completed with 1 cycle delay slot (branch solved in ID stage).

In the Register File, it is possible to read and write at the same address at the same clock cycle

Please make a **schedule** on the 4-issue VLIW machine **(NOPs are not written).**

	Integer ALU	Mem Unit	FP ADDER	FP MULTIPLIER
C1				
C2				
C3				
C4				
C5				
C6				
C7				
C8				
C9				
C10				
C11				

Open text answer

EXERCISES/QUESTIONS TAKEN FROM PREVIOUS ONLINE EXAMS

Feedback:

The schedule (including the branch delay slot) requires 13 cycles as follows:

	Integer ALU	Mem Unit	FP ADDER	FP MULTIPLIER
C1		`LD F0, 0(R1)`		
C2		`LD F2, 4(R1)`		
C3			`FADD F3, F0, F1`	`FMULT F5, F0, F1`
C4			`FADD F4, F2, F1`	`FMULT F6, F2, F1`
C5				
C6			`FADD F7, F3, F5`	
C7			`FADD F8, F4, F6`	
C8				
C9		`SD F7, 0(R1)`		
C10	`ADD R1,R1,8`	`SD F8, 4(R1)`		
C11				
C12	`BNE R1 R2 Loop`			
C13	`Br. delay slot`			

EXERCISES/QUESTIONS TAKEN FROM PREVIOUS ONLINE EXAMS

Question 6 (format open text) from PART2 from 16/07/2020

Let's consider the following code:

```
LOOP: LD $F0, 0 ($R1)
      ADDD $F4, $F0, $F2
      SD $F4, 0 ($R1)
      ADDDUI $R1, $R1, 8
      BNE $R1, $R2, LOOP
```

Show a software-pipelined version of this loop by omitting the start-up and finish up code
2 points

Open text answer (max 100 words)

Feedback:

```
LOOP: SD $F4, 0 ($R1)          /* store from iteration [i] corr. to index 0 */
      ADDD $F4, $F0, $F2       /* add from iteration [i+1] corr. to index 8  */
      LD $F0, 16 ($R1)         /* load from iteration [i+2] corr. to index 16 */
      ADDDUI $R1, $R1, 8
      BNE $R1, $R2, LOOP
```