Department of Computer Science & Engineering, IIT Guwahati

CS 223: Computer Architecture & Organization - End Semester Exam (08.05.2023)

Maximum Marks: 70 **Time: 180 minutes**

Common Instructions

1. *The question paper has 11 questions (six 5 marks questions and five 8 marks questions).*
2. *Wherever applicable, show the necessary calculations/diagrams used for obtaining the final answer.*
3. *Fill the facing page of answer booklet with question number (in increasing order from 1, 2, .., 10, 11) and corresponding page numbers. If you are not attempting a question, write the question number and put a cross [x] under page number.*

SECTION-A (6x5 = 30 marks)

Q1: (i) Choose the most appropriate matching for each element in Set A to Set B.

Set A	**Set B**
(a) Row Buffer	1. Prefetched blocks
(b) Victim Buffer	2. Flow control
(c) Virtual Channel Buffer	3. Evicted clean blocks
(d) Re-Order Buffer	4. Dirty blocks
(e) Stream Buffer	5. DRAM
(f) Write Buffer	6. Speculative execution
	7. Multithreaded execution

a->5, b->3, c->2, d->6, e->1, f->4 **[2 Marks]**

1 mark for 3 correct , 2 mark for 6 correct ,

(ii) For each element in Set X, complete the sentence by the most appropriate phrase in Set Y.

Set X	**Set Y**
(g) No-write allocate cache	1. exploits sequential interleaving.
(h) Non-blocking cache	2. allows faster clock cycle time.
(i) Pipelined cache	3. refers to dirty bit during eviction.
(j) Multi-banked cache	4. makes use of loop unrolling.
(k) Compiler assisted pre-fetching	5. does not replace cache block on a write miss.
(l) Victim cache	6. reduces miss penalty.
(m)Way prediction	7. implementation needs MSHR.
(n) Multi level cache	8. reduces hit time.
(o) Write back cache	9. reduces miss rate.

g->5, h->7, i ->2, j->1/8, k->4/9, l->6, m->8, n->6, o->3 **[3 Marks]**

1 mark for 3 correct , 2 mark for 6 correct , 3 mark for 9 correct

Q2: Consider a 16-tile TCMP system with a total of 1 MB shared distributed L2 cache that is 16-way associative having a block size of 16 B. Generally, the 16 ways of a given set are placed in the same tile and the set numbers are partitioned sequentially across tiles. What are the pros and cons of a modified design where the ways are partitioned sequentially across tiles, way 0 of all sets placed in tile 0, way 1 of all the sets placed in tile 1 etc.?

[5 Marks]

Here, ways are partitioned sequentially across tiles. Hence L2 search of a given set sending packet to all tiles and L2 tag comparison in all tiles. High traffic will be generated, and how to identify Miss/Replacement algorithm need to be handle across, need for a centralized one.

Impractical Design

- L2 way distributed packet to all tiles
- Hit/ Miss check (need a centralized one)
- Replacement logic overhead

Q3:(a) Consider an instruction ADD Rz, Rx, Ry (Rz<-Rx+Ry) that has to be executed in a dynamically scheduled speculative processor that follows Tomasulo's algorithm. When the instruction is issued, the seven-tuple entry {Op, Qj, Qk, Vj, Vk, A, Busy} in the reservation station for this instruction is {ADD, 12, 0, 20, 15, 100, 1} and the RSI value of Rz is 16. What is the interpretation of this reservation station entry and the RSI value of Rz?

- Rx is available in ROB #12 **[3 Marks]**
- Ry value is 15 (taken from the register file)
- result to be stored in ROB #16

(b) In a dynamically scheduled speculative processor, an ADD instruction that is residing is the instruction queue is not issued. There are sufficient free entries both in the ROB and in the reservation station of the adder. What could be the potential reason that this instruction is not issued? Assume that none of the instruction in the queue are branch instructions.

- ROB is free.
- RS is free; there is no branch **[2 Marks]**

The issue queue is stalled. The front of the queue has an instruction that is stalled due to the structural hazard at its RS.

Q4: (a) Imagine that memory requests are coming to a particular bank of DRAM at regular intervals such that a new request reaches the scheduler when the processing of the previous request is in progress. The scheduler uses FR-FCFS policy. Under this context, whether open row buffer management or closed row buffer management will give lesser average waiting time? Why?

Both give the same performance **[3 Marks]**

- Always queue is Non-empty
- As per FR-FCFS choose the request from queue in same row or else precharge,
- There is no difference between open row and closed row here.

(b) Consider a single channel 4 GB DRAM system organized in 16 banks (B0, B1,.., B15). Each row has 1024 columns and the system uses a 1 MB, 8-way set associative last level cache having a block size of 32 words. The word length as well as data bus that connect DRAM to last level cache is 32 bit. If we use cache block interleaving, which bank stores the address 0x22446688?

[2 Marks]

LLC is 32 words = 32x4B = 128 B → 7 lsb for locality

16	5	4 (bank)	5	2
Row	High. Column	Bank	Low. Column	Byte in Bus

0x22446688 = 0010 0010 0100 0100 0110 0110 1000 1000

0110 0**110** **1**000 1000

1101 is **bank,** which is **13 (B13)**

Q5: A 16-bit word processor is connected to the following memory hierarchy. L1 cache: 8 KB, 4-way set associative and 32 B block size. L2 cache: 64 KB, 8-way set associative and 128 B block size. Main memory: 16 MB, 8 banks and one channel. When an L1 cache miss occurs, it takes 30 cycles to fetch the first word of a block from L2 cache and 2 cycles for each subsequent word in the block. Assume that processor is stalled due to an L1 cache miss occurred on a word whose first byte address is 0x415ACE. Assume that the word is a hit in L2 cache. How many cycles will the processor stall before it resumes execution under each of the following 3 cases?

(a) Early restart optimization done on L1 cache.

(b) Critical word first optimization done on L1 and L2 caches.

(c) None of the above 2 optimizations are done on caches.

Main Memory = 16MB = 2^{24} => 24 bits

L1 Cache:-

Cache Size = 8KB

4 Way Set Associative

Block Size = 32B

words/blocks = 32/2 = 16 => 4 bits

sets = $2^{13}/(2^5 * 2^2) = 2^6$ => 6 bits

13	6	5
Tag	Set	Block Offset

L2 Cache:-

Cache Size = 64KB

8 Way Set Associative

Block Size = 128B

words/blocks = 128/2 = 64 => 6 bits

sets = $2^{16}/(2^7 * 2^3) = 2^6$ => 6 bits

11	6	7
Tag	Set	Block Offset

	Set	Word No
L1 Cache Miss - 0x41 0101 1010 1100 1110	22	7 (8th word)
L2 Cache Block - 0x41 0101 1010 1100 1110	53	39

a) Early Restart = 30 + 7*2 = 44 cycles **[2 Marks]**

b) Critical Word First = 30 cycles **[1 Marks]**

c) No Optimization = 30 + 15*2 = 60 cycles **[2 Marks]**

Q6: A program is stored in a 16 MB DRAM that is attached to a 4 KB direct mapped cache with a block size of 16 B. The program reads 4 data words A, B, C and D in that order 25 times (total 100 memory references). Let the physical addresses of A, B, C and D are 0x420424, 0x74066A, 0x420420, 0x720660, respectively. Assume the caches are empty initially and one word is 16 bits.

(a) What is the hit rate of the cache considering only the data word accesses?

(b) At the end of program execution, which all data words are residing inside the cache?

a) # words/ blocks = 16/2 = 8 => 3 bits

lines = $2^{12}/2^4 = 2^8$ => 8 bits

12	8	4
Tag	Set	Block Offset

	Set	Word No
A => 0x420424 = 0100 0010 0100	66	2
B => 0x74066A= 0110 0110 1010	102	5
C => 0x420420 = 0100 0010 0000	66	0
D => 0x720660 = 0110 0110 0000	102	0

Data words A and C have the same tag (420), and they map to set 66. A will have a compulsory miss upon which both A and C are loaded into set 66. Thereafter, A will not have any misses. C will also have 0 misses.

B and D are mapped to same set 102, but their tag values are different. So they would be conflict miss in every access of B and D. Since there are 25 access of B as well as D, there would be 25 misses for B and 25 misses for D.

Misses,

A = 1, B = 25, C = 0, D = 25 => Total = 51

Total No of Hits = 100 - 51 = 49

Hit rate = 49/100 = 0.49 **[3 Marks]**

b) At the end of program execution, A, C and D will be in cache

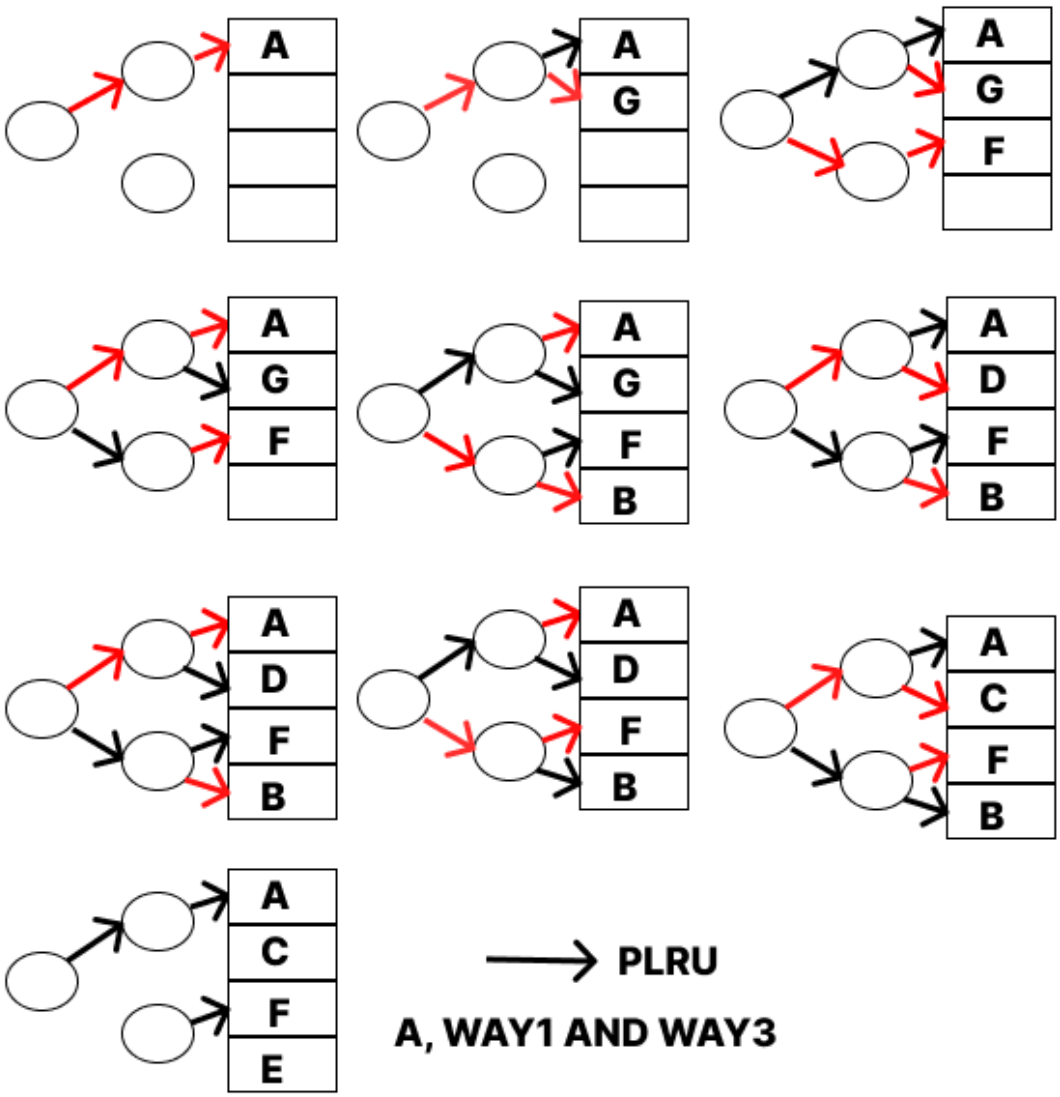
Set no 66 - A and C

Set No 102 - D **[2 Marks]**

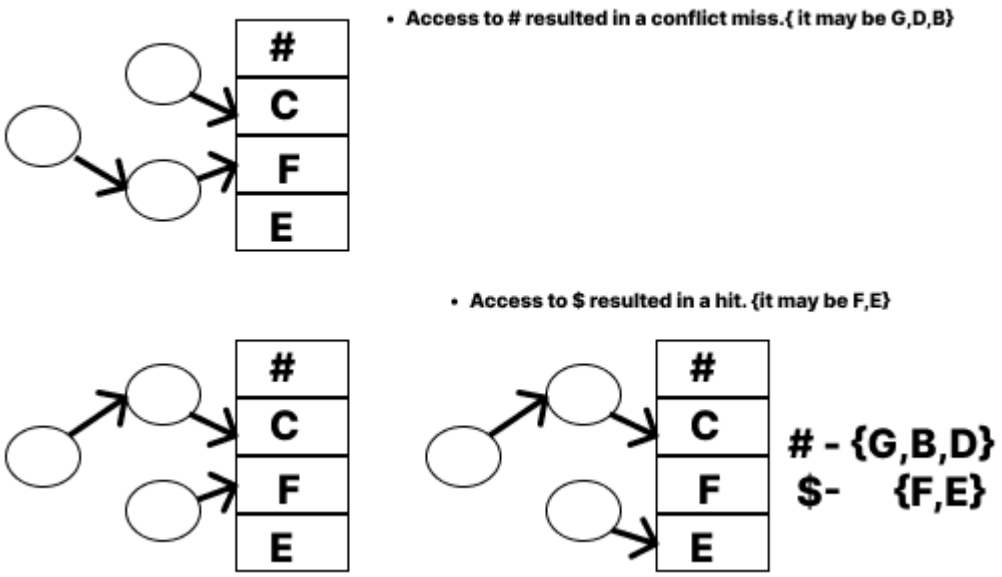
SECTION-B (5x8 = 40 marks)

Q7: Consider a 4-way set associative cache that uses pseudo LRU block replacement policy. Assume all the blocks are initially empty and filling up of empty blocks in a given cache set is done from way-0 to way-3. Consider the following 10 block numbers (excluding #, $ and &), all mapped to a particular cache set n, given in the order of arrival. A, G, F, A, B, D, A, F, C, E, #, $, &.

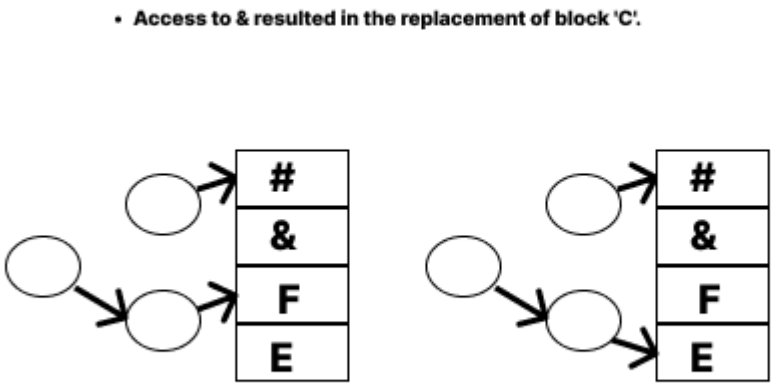
(a) For the first 10 requests, out of the 4 ways, which ways have not experienced a hit? **[2 Marks]**



(b) Consider the requests #, $ and &, given above for set n. If access to # resulted in a conflict miss, access to $ resulted in a hit and access to & resulted in replacement of block 'C', give the set of all possible values for # and $. **[4 Marks]**



(c) Draw the PLRU tree for set n after processing the above 13 requests. **[2 Marks]**



Q8: Consider a 5x5 mesh TCMP that uses MinBD routers in its NoC framework. Consider 5 flits P1, P2, P3, P4, and P5 that reach router 12 at clock cycle T. Flit details (packet number, arrival input port, source, destination) are (P1, N, 14, 2), (P2, E, 19, 7), (P3, S, 7, 24), (P4, W, 10, 12) and (P5, L, 12, 14). At T, P2 is the golden flit in the NoC and P3 is silver flit at router 12. XY routing is used to find productive output port in MinBD routers. Whenever there is an arbitration between two flits which are neither golden nor silver, preference is given to a flit that has travelled longer distance in the past. With respect to router 12, based on operational logic of MinBD routers, answer the following.

Packet	IP Port	Src	Dest	Desired OP	OP allocation	Remarks
P1	N	14	2	S	W	To side buffer
P2- Go	E	19	7	S	S	To south neighbour
P3 - Si	S	7	24	E	E	To east neighbour
P4	W	10	12	L	L	To local tile
P5	L	12	14	E	N	To north neighbour

(a) What are the output port assignments to each of the flits? **[2 Marks]**

P1-Side buffer, P2-South, P3-East, P4- Local P5: North

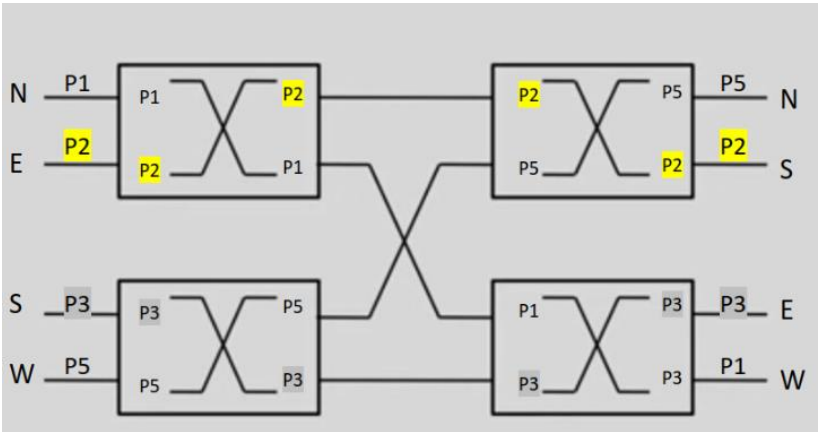
(b) Which flit suffered from injection suppression? Why? **[1 Mark]**

No packet will suffer from injection suppression. P5, the locally injected packet will occupancy the flit channel vacated by P4 which got ejected.

(c) Which all flits got side buffered? Why? **[1 Mark]**

Only P1, P1 wins over P5 due to higher distance travelled in the past

(d) Draw a neat labelled diagram of the PDN logic and its internal 4 arbiters of MinBD router and mark the flits that reach the input of each of these arbiters. **[4 Marks]**



Q9: Consider a TCMP system with an 4x4 mesh NoC where each tile consists of a 64-bit word processor, a private L1 cache, and a shared distributed L2 cache. The TCMP is connected to a main memory of 64 GB using 4 memory channels. Each channel support 16 GB of continuous memory. Each tile has an 8 KB, 2-way set-associative unified L1 cache. Total on-chip L2 cache is 16 MB and it is 16-way set-associative. L1 and L2 caches have 64 B block size each. NoC router uses XY routing. Inter-router link is 128 bits wide. Packet latency from source router to destination router will take 4n+2 cycles, where n is the number of intermediate routers (excluding source and destination). It takes an additional 2 cycles each for packet injection into the network from the tile and packet ejection from the network to the tile.

(a) Identify the tag, set index and offset split up of L1 and L2 cache.

L1 cache Total 36 bit physical address **[2 Marks]**

Tag (24)	Set Index (6)	Byte Offset (6) 3 bit word and 3 bit byte

L2 Cache

Tag (16)	Set Index (14) 4 bit tile and 10 bit set	Offset (6) 3 bit word and 3 bit byte

(b) If tile T3 generates an L1 cache miss on address 0x588687070, which tile has the mapping for corresponding L2 cache? How many cycles are needed for the cache miss request to travel from L1 tile to L2 tile?

0x588687070 = 0101 1000 1000 0110 1000 0111 0000 0111 0000

Set Index Bits (14) = Tile Bits (4) + Set Bits (10)

1000 0111 0000 01

Destination Tile is **8** **[1 mark]**

Src = T3, Dst = T8, XY path = **3→2→1→0→4→8**

No of Cycles: = 4n + 2, here n = 4 (4 intermediate routers in the XY path),

Injection + Ejection = 2+2 =4

Latency of cache miss request packet. = (4x4 + 2) + 4 = 22 cycles **[1 mark]**

(c) If a tile T0 generates an L1 cache miss on address 0xC40856080, which in turn results in an L2 cache miss to a main memory channel. List the set of routers through which L1 and L2 miss request and reply packets are traveling.

0xC40856080 = 1100 0100 0000 1000 0101 0110 0000 1000 0000

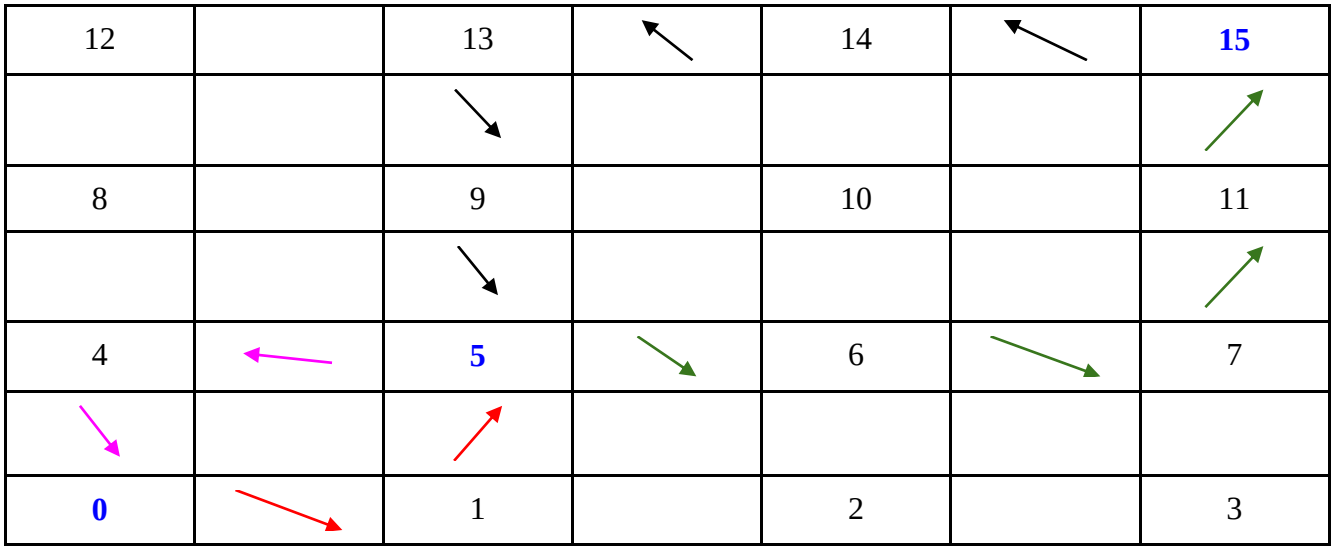
Set Index Bits(14) = Tile Bits(4) + Set Bits(10)

0101 0110 0000 10

Destination Tile is T5 (L2 Mapped)

Msb 2 bits determine channel = 1100 0100 0000 1000

11 indicates Memory channel 3 connected to Tile 15 (Memory Channel)



L1 cache miss request to L2 tile path = **0→1→5** **[4 Marks]**

L2 cache miss request to Memory Controller path = **5 →6-→7→11→15**

L2 cache miss reply from Memory Controller to L2 tile path = **15→14→13→9→5**

L1 cache miss reply from L2 tile to L1 tile path = **5→4→0**

Q10: Compare pros and cons and its impact on IPC for the following techniques.

(a) Hyper threading vs multithreading in hardware. **[2 marks each]**

Hyper threading issue of instruction from different application in same clock cycle.

Reduce empty slots in issue

Exploit task level parallelism at same cycle (Diagram)

Multithreading issue of instruction from different application in same clock cycle.

Can have empty slots in issue.

Exploit task level parallelism at finer or coarser granularity (Diagram)

(b) Software scheduling vs Operand forwarding. **[2 marks each]**

Software scheduling - compiler reorganization based on instruction level dependency.

Overhead at compiler level. Hardware will not do dynamic scheduling.

Higher IPC than operand forwarding.

Operand forwarding - hardware level operand by pass paths.

Can have only limited level impact as instruction order cannot be changed.

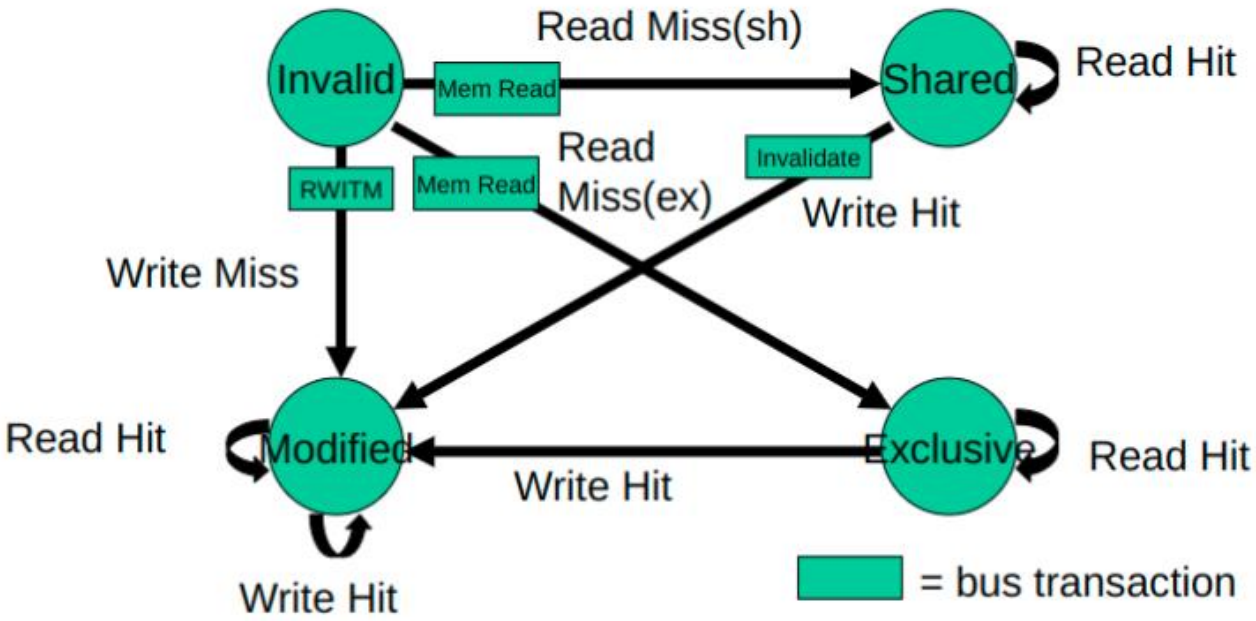
Q11: Draw a neat labelled state transition diagram of MESI cache coherence protocol. Consider a multi-processing system with two cores A and B with their own private caches and a single shared main memory. The following 5 lines of code is running in each of the two cores. Since both A and B are using their private caches, MESI cache coherence protocol is used for operation on shared variables.

1. LW R1, M1
2. LW R2, M2
3. SW R3, M2
4. SW R2, M1
5. LW R1, M2

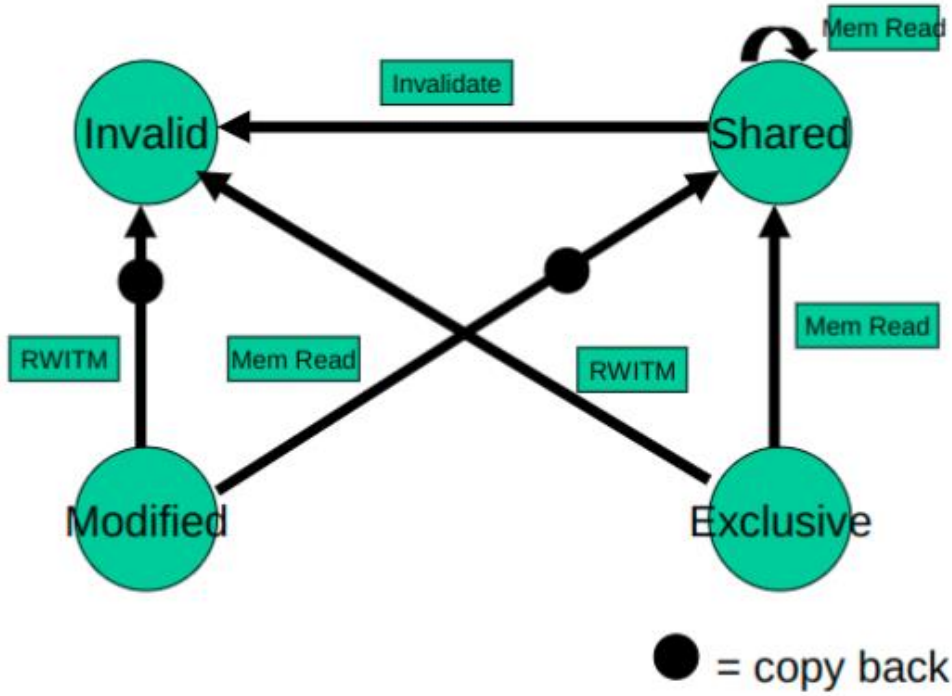
LW indicates Load and SW indicates Store operation. Ri and Mj indicates Register and Memory operands, respectively. Assume that the addresses pointed by M1 and M2 map to different cache blocks. Consider the following execution sequence in the format; Core Name-Instruction Number. A-1, A-2, B-1, B-2, A-3, B-3, B-4, A-4, A-5, B-5 (total 10 instruction instances).

The initial state of all cache blocks is invalid. Refer to the following table that shows the cache block state in the private caches of A and B. Each table entry can be one among M/E/S/I. Fill up the table after the execution of each entry in the execution sequence given above. [There should be total 10 rows in the table after initial, one per instruction instance shown in the sequence above.]

Locally initiated accesses **[2 marks]**



Remotely initiated accesses **[1 mark]**



[5 Marks, 1 mark each for 2 adjacent correct rows]

Core- Instruction	State of M1 in A	State of M2 in A	State of M1 in B	State of M2 in B
initial	I	I	I	I
A-1	E	I	I	I
A-2	E	E	I	I
B-1	S	E	S	I
B-2	S	S	S	S
A-3	S	M	S	I
B-3	S	I	S	M
B-4	I	I	M	M
A-4	M	I	I	M
A-5	M	S	I	S
B-5	M	S	I	S

****