Department of Computer Science & Engineering, IIT Guwahati

CS 223: Computer Architecture & Organization – Quiz #2 (18.04.2023)

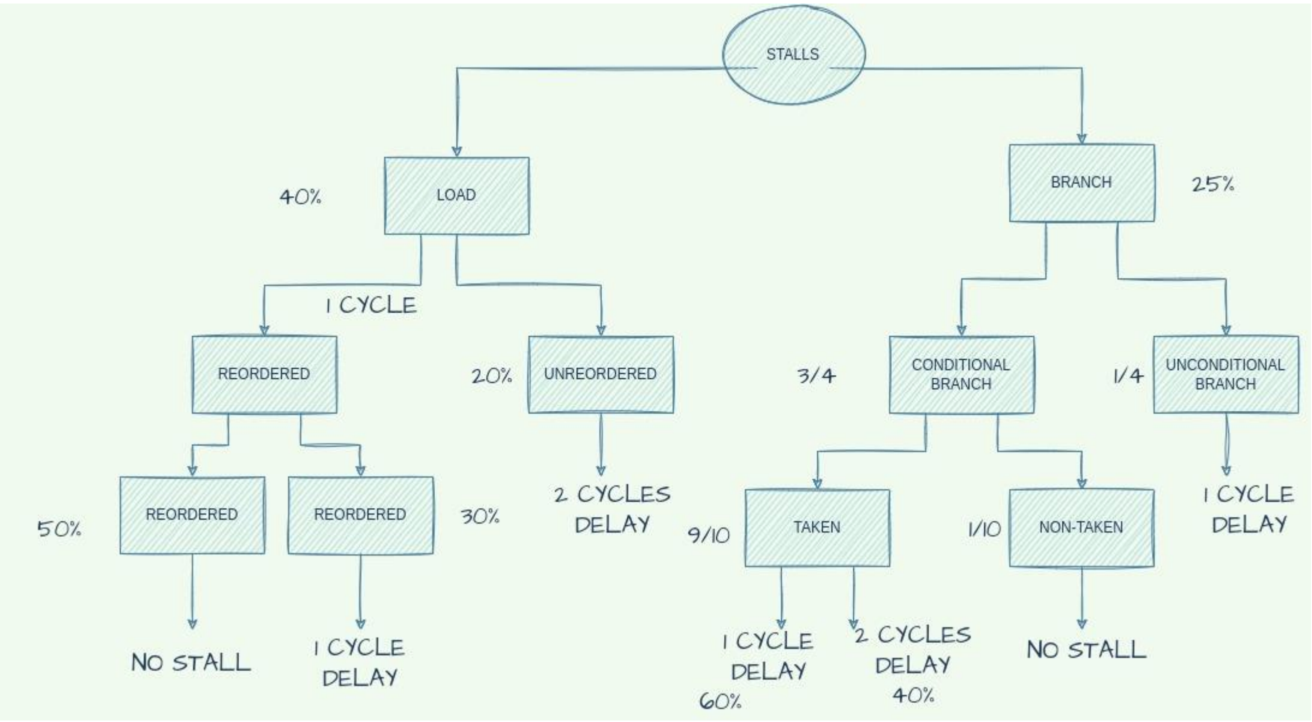
Max. Marks=30 **Time =90 minutes**

Name: ………………………………………………………… Roll Number: ……………….

Answer only in the allotted space. Calculations, wherever applicable has to be provided in the same space itself. Answers and calculation on rough sheet will not be graded.

Sign. Student	Sign. Invigilator	Q1 [4]	Q2 [4]	Q3 [6]	Q4 [6]	Q5 [10]	Total [30]	Sign. Instructor

1. A processor with a base CPI of 1 incurs 2 cycles of stall for every load and 1 cycle of stall for every taken branch instructions. There are 40% load instructions and 25% branch instructions. Out of the branch instructions three fourth are conditional branch instructions and rest are unconditional branch instructions. For conditional branches, taken : non-taken ratio is 9:1. 60% of conditional taken branch instructions are schedulable to fill 1 cycle. All unconditional branch instructions are also schedulable to fill 1 cycle delay slot. A conditional not-taken branch will not incur a stall. 50% of load instructions can be reordered to avoid two cycles delay slots. 20% of load instructions cannot be reordered. Rest of the load instructions can be reordered to fill in 1 delay slot. Under these circumstances, what is the resultant CPI of the machine for the following 2 cases. Show calculations. (Total=4marks)



CPI = Base CPI + Stalls = 1 + Stall for load + Stall for branch

(a) Case A: Without reordering or scheduling of any the instructions. (1+1=2Marks)

Resultant CPI = 1 + 0.4 * 2 + (0.25 * 0.25 * 1) + (0.25 * 0.75 * 0.9) * 1
= 1 + 0.8 + 0.0625 + 0.16875
= 2.03125

The CPI in this case is 2.03125

(b) Case B: With reordering and scheduling the instructions for filling delay slots (1+1=2marks)

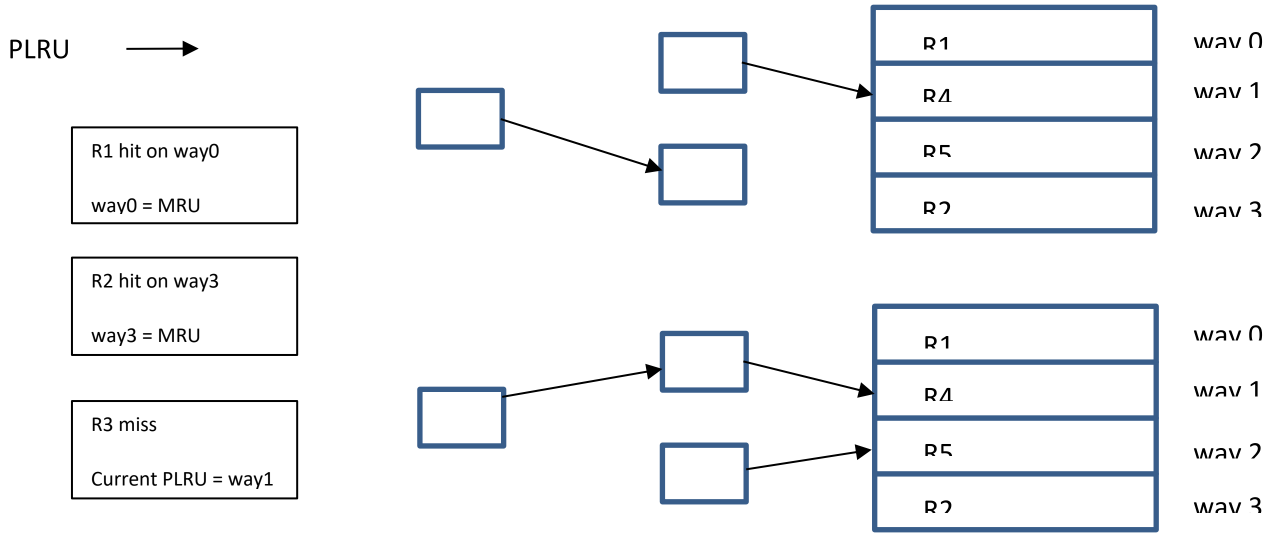
CPI = Base CPI + Stalls = 1 + Stall for load + Stall for branch
= 1 + (0.2 * 0.4 * 2) + (0.3 * 0.4 * 1) + (0.25 * 0.75 * 0.9 * 0.4)
= 1 + 0.28 + 0.0675
= 1.3475

The CPI in this case is 1.3475

2. In a 4-way set associative cache that uses PLRU block replacement strategy, three distinct memory requests R1, R2 and R3 (in the given order) indexed to the same set number which has valid data on all four blocks of the set. Under these conditions, state whether the following statement is True/False. Give necessary justifications. "If R1 is a hit in way-0, followed by R2 which is a hit in way-3; a miss in R3 will evict contents in way-2." (Total=4marks)

Answer: True / False Justification: [Draw the final PLRU tree and explain]

FALSE (1 Mark)



The statement is FALSE, Since the current PLRU is way1 and R3 will evict the content from way1 (3 Marks)

3. Consider a speculative dynamic scheduled instruction pipeline that uses Tomasulo's algorithm with an issue width of 1. There is one FP Mul unit, one FP Add unit, one Integer Add unit, and one Load unit, all connected to a single CDB. All the functional units are internally pipelined. Assume that the FP Mul unit, FP Add unit and Integer Add units take 7, 4, and 1 cycle, respectively for their execution stages. Load unit will take one cycle for address computation followed by writing the accessed word from memory to CDB in the next cycle. This speculative pipeline is implemented with a limited size ROB. An instruction waiting for data on CDB can move to its EX-stage in the next cycle after the CDB write stage. There are five reservation stations in each functional unit. 8 instructions (I1 to I8) as shown below are to be executed. The following table contains the clock cycle number in which various operations are done in the processor with the above specifications. Fill up the table entries for 2 individual cases of ROB size. (Total=6 Marks)

	Case A: ROB Size = 3				Case B: ROB Size = 5			
Instruction	**Issue**	**Start Execute**	**CDB Write**	**Commit**	**Issue**	**Start Execute**	**CDB Write**	**Commit**
I1: LOAD F1, 16(R2)	1	2	3	4	1	2	3	4
I2: FADD F2, F1, F3	2	4	8	9	2	4	8	9
I3: FMUL F5, F6, F1	3	4	11	12	3	4	11	12
I4: ADD R2, R2, R1	5	6	7	13	4	5	6	13
I5: FADD F3, F5, F1	10	12	16	17	5	12	16	17
I6: ADD R3, R2, R1	13	14	15	18	6	7	9* collission	18
I7: FMUL F2, F4, F3	14	17	24	25	10	17	24	25
I8: FADD F5, F2, F1	18	25	29	30	13	25	29	30
	(2+1=3 Marks)				(2+1=3 Marks)			

4. Let us take a branch instruction in a program which is iterated 8 times. A (2, 2) type corelating branch predictor is used by the processor. The outcome of the last two branches is used to index into the BHT. Each entry of BHT is a 2-bit value that is updated by a standard 2-bit finite state machine. The initial entry of the BHT for NN/NT/TN/TT is 00/01/10/11. The BHT is indexed with an NT value initially. Upon execution of the branch using the predictor, the following observations were noticed. (Total = 6Marks)

   1. The branch predictor has an overall accuracy of 0.625 for these 8 iterations.
   2. There were not more than 2 continuous correct predictions.
   3. The actual outcomes were a set of continuous N followed by a set of continuous T.

   (a) Based on the above conditions, illustrate the operation of the predictor by filling up the table.

Branch Iteration	**Last 2 branch Outcomes**	**BHT** **NN / NT / TN / TT**	**Prediction**	**Outcome**	**Remarks**
I1	**NT**	**00 / 01 / 10 / 11**	N	N	correct prediction
I2	TN	**00 / 00 / 10 / 11**	T	N	misprediction
I3	NN	**00 / 00 / 00 / 11**	N	N	correct prediction
I4	NN	**00 / 00 / 00 / 11**	N	N	correct prediction
I5	NN	**00 / 00 / 00 / 11**	N	T	misprediction
I6	NT	**01 / 00 / 00 / 11**	N	T	misprediction
I7	TT	**01/ 01 / 00 / 11**	T	T	correct prediction
I8	TT	**01/ 01 / 00 / 11**	T	T	correct prediction
	Final State	**01/ 01/ 00 / 11**			

(4Marks)

   (b) How many instances of state changes happened in BHT? Mention the iterations where this happened. (1Mark)

   4 state changes: After I1, I2, I5 and I6

   (c) What is the BHT entry after the execution of the 8 iterations of the branch instruction?

   01/ 01/ 00 / 11 (1Mark)

5. A small RISC processor-based system with 32-bit word has a 4 GB main memory. It has a 1 KB direct mapped cache with a block size of 32 B. Assume the cache is initially empty. Consider a small program P that consists of 14 instructions. Instruction size is also same as word length. Execution of P swaps the contents of three arrays; A, B, and C each of size 24 words using a loop that iterate 24 times. A temporary variable T of size one word is used for facilitating the swap. Consider the following physical addresses for the first byte of P and data arrays. P→ 0x20608040 : A→0x320A0580 : B→0x53BC0D24 : C→0xA2B62A48 : T→0x40030948. The high level

language structure of the loop that does the swapping is as follows.

{T=A[i]; A[i]=B[i]; B[i]=C[i]; C[i]=T; }

(a) For the cache, give the split-up of tag, set index and byte offset fields of physical address. (2 Marks)

Given cache size = 1 KB = 2^10 bytes

Block size = 32 bytes = 2^5 bytes

Number of blocks in the cache = (1 KB) / (32 B) = 2^5 = 32 blocks

Thus, we need 22 bits for tag. 5 bits for set index, and 5 bits (3 bit for block and 2 bit for byte offset) for byte offset.

Tag – 22 bits	Set Index – 5 Bits	Byte offset – 5 Bits

(b) How many data memory loads and stores are generated by the processor to swap the data? How many of them are hits and how many of them are misses. Explain. (4 Marks)

This will execute for 24 iterations

LOAD A[i]

STORE T

LOAD B[i]

STORE A[i]

LOAD C[i]

STORE B[i]

LOAD T

STORE C[i]

In one iteration, we have 8 memory references. In which 4 would be LOAD and 4 would be STORE. Total number of memory references = 8*24 = 192 memory references out of which 96 would be LOAD and 96 would be STORE

LOAD = 96

STORE = 96

Total= 192 memory references

Given addresses,

P→ 0x20608040 : A→0x320A0580 : B→0x53BC0D24 : C→0xA2B62A48 : T→0x40030948

In the address split-up, first 22 bits would be used for tag bits, next 5 bits would be used for finding which set it belongs to and last 5 bits for the byte offset.

Blue – Set No

Green – Offset (3 bits for finding the correct word and 2 bits for finding the correct byte in the word)

		Set No	Word No
P->0x20608	0000 0100 0000	2	0
A→0x320A0	0101 1000 0000	12	0
B→0x53BC0	1101 0010 0100	9	1 (001 00) - 3 bits for the word and 2 bits for Bytes in the word
C→0xA2B62	1010 0100 1000	18	2 (010 00)
T→0x40030	1001 0100 1000	10	2 (010 00)

Misses,

A = 3 Misses

Set No 12: A [0] - A [7]

Set No 13: A [8] - A [15]

Set No 14: A [16] - A [23]

B = 11 Misses

Set No 9: B [0] - B [6]

Set No 10: B [7] - B [14]

Set No 11: B [15] - B [22]

Set No 12: B [23]

C = 4 Misses

Set No 18: C [0] - C [5]

Set No 19: C [6] - C [13]

Set No 20: C [15] - C [21]

Set No 21: C [22] - C [23]

T = 9 misses

Set No 10: T

Total = 37 Misses

B and T have conflict on set no 10: After 6th iteration, content of set No 10 is T
7th Iteration: Store T will be hit. Next Load B [7] would be compulsory miss. Load T will be a miss.

This will continue till 14th iteration.

15th Iteration: Store T will be hit. Next Load B [15] will be a compulsory miss and it will be loaded into set no 11 till B [22].

Similarly, B and A have a conflict on set no 12. Contents of A in the set no 12 would be removed and B [23] would be placed in set No 12

So we have A = 45 hits, B = 37 hits, C = 44 hits, T = 39 hits --> Total= 165

(c) List out the cache block numbers (in decimal) that are non-empty after the execution of P. (2 Marks)

There are 14 instructions in the program P, which would be stored in set no 2 and 3

Set No	
0	
1	
2	P0 - P7
3	P8 - P13
:	
:	
9	B [0], B [1], B [2], B [3], B [4], B [5], B [6]
10	T
11	B [15], B [16], B [17], B [18], B [19], B [20], B [21], B [22]
12	B [23]
13	A [8], A [9], A [10], A [11], A [12], A [13], A [14], A [15]
14	A [16], A [17], A [18], A [19], A [20], A [21], A [22], A [23]
:	
:	
18	C [0], C [1], C [2], C [3], C [4], C [5]
19	C [6], C [7], C [8], C [9], C [10], C [11], C [12], C [13]
20	C [14], C [15], C [16], C [17], C [18], C [19], C [20], C [21]
21	C [22], C [23]
:	
31	

Set Number: 2(P0-P7), 3(P8-P13), 9(B), 10(T), 11(B), 12(B), 13(A), 14(A), 18(C), 19(C), 20(C), 21(C) are non-empty.

(d) List the data items that are present in the cache after the execution of P.

(2 Marks)

	Set No
B [0] - B [6]	9
T	10
B [15] - B [22]	11
B [23]	12
A [8] - A [15]	13
A [16] - A [23]	14
C [0] - C [5]	18
C [6] - C [13]	19
C [14] - C [21]	20
C [22] - C [23]	21