CS224
Lab No: 4
Section No: 06
Barış Tan Ünal
22003617

LAB 6 PRELIMINARY REPORT

PART 1

No.	Cache Size KB	N Way Cache	Word Size (no. of bits)	Block Size (no. of words	No. of Sets	Tag Size in Bits	Index Size (Set no.) In bits	Block Offset Size	Byte Offset Size In Bits	Block Replacement Policy Needed
1	64	1	32	4	2^{12}	15	12	2	2	No
2	64	2	32	4	2^{11}	16	11	2	2	Yes
3	64	4	32	8	2^{9}	17	9	3	2	Yes
4	64	Full	32	8	1	26	0	3	2	Yes
9	128	1	16	4	2^{14}	14	14	2	1	No
10	128	2	16	4	2^{13}	15	13	2	1	Yes
11	128	4	16	16	2^{10}	16	10	4	1	Yes
12	128	Full	16	16	1	26	0	4	1	Yes

PART 2.a

Instruction	**Iteration No.**				
	1	**2**	**3**	**4**	**5**
lw $t1, 0x24($0)	Compulsory	Conflict	Conflict	Conflict	Conflict
lw $t2, 0xAC($0)	Compulsory	HIT	HIT	HIT	HIT
lw $t3, 0xC8($0)	Conflict	Conflict	Conflict	Conflict	Conflict

PART 2.b

C (Capacity): 16 words
b (block size): 4 words
B (Number of blocks): 8
S (Number of sets): 2

With the quantities above, the offset and set bits can be calculated.

Number of set bits: $log_2 S = log_2 2 = 1$
Number of block offset bits: $log_2 b = log_2 4 = 2$
Number of byte offset bits: 2

Number of tag bits: 32 – (1 + 2 + 2) = 27

Taking 1-bit valid bit, tag bits and the data field bits into account in the calculations.

Total memory cache size: (1 + 27 + (32 * 4))* 4 = **624 bits**

PART 2.c

(x2) 4:1 mux: for selecting the proper memory location in the block
(x1) 2:1 mux: for selecting from the proper way of the cache since it is 2-way
(x2) equality comparator: for checking if the tag is equal to the tag we are searching for
(x2) AND gate: for generating hit signal, taking the outputs of comparators
(x1) OR gate: for generating hit signal, taking the outputs of the AND gates

PART 3.a

Instruction	**Iteration No.**				
	1	**2**	**3**	**4**	**5**
lw $t1, 0x24($0)	Compulsory	Capacity	Capacity	Capacity	Capacity
lw $t2, 0xAC($0)	Compulsory	Capacity	Capacity	Capacity	Capacity
lw $t3, 0xC8($0)	Capacity	Capacity	Capacity	Capacity	Capacity

PART 3.b

Since there is only one set, there is no need for set bits.
Also, there is no need for block offset too because the block size is only one word.
We need 2-bits of byte offset since number of blocks equals to 2.

Tag bits should be 30-bits (32 – 2 = 30).

One bit is sufficient for LRU policy since cache memory size is 2.

Total cache size: (32 + 1 + 30) * 2 = **126 bits.**
(32 bits for data, 1 bit for valid bit and 30 bits for tag.)

PART 4

AMAT = 2 + 0.1(4 + 0.05 * 20) = **2.5 clock cycles**

2GHz = 2 * 2^{30} Hz → Clock Period (T) = 2^{-31}s
Time Needed = AMAT * Instr. Count * T = **11.64 s**