CS224 Lab No: 6
Section No: 1
Halil Arda Özongun
22202709

1

No.	Cache Size KB	N way cache	Word Size	Block size (no. of words)	No. of Sets	Tag Size in bits	Index Size (Set No.) in bits	Word Block Offset Size in bits[1]	Byte Offset Size in bits[2]	Block Replacement Policy Needed (Yes/No)
1	64	1	32 bits	4	4096	16	12	2	2	No
2	64	2	32 bits	4	2048	17	11	2	2	Yes
3	64	4	32 bits	8	512	18	9	3	2	Yes
4	64	Full	32 bits	8	1	27	0	3	2	Yes
9	128	1	16 bits	4	16384	15	14	2	1	No
10	128	2	16 bits	4	8192	16	13	2	1	Yes
11	128	4	16 bits	16	1024	17	10	4	1	Yes
12	128	Full	16 bits	16	1	27	0	4	1	Yes

2

Cache capacity is 8 words, Block size: 2 words, N= 1.

```
        addi    $t0, $0, 5
loop:   beq     $t0, $0, done
        lw      $t1, 0x4($0)
        lw      $t2, 0xC($0)
        lw      $t3, 0x8($0)
        addi    $t0, $t0, -1
        j       loop
done:
```

a.

Instruction	Iteration No.				
	1	2	3	4	5
lw $t1, 0x4($0)	Compulsary				
lw $t2, 0xC($0)	Compulsary				
lw $t3, 0x8($0)					

b.

Cache has 4 set, and per set there are two words. So for each set we need to hold 32*2= 64 bit data. We also need to hold a valid bit, to see if data is valid or not, +1 bit. Lastly, we need to check if the data in cache is the data we are looking for, therefore we need to hold tag. Adresses are 32 bit, 2 bit is byte offset, 2 bit is set, and 1 bit for block offset. So we need to check remaining 27 bits.
27+64+1 = 92 per set. There are 4 sets:

4*92 = 368 bit.

c.

Equality Comparator = 1
And Gates = 1
Or Gates = 0
Multiplexers = 1 (2:1 mux)

3

a.

Instruction	Iteration No.				
	1	2	3	4	5
lw $t1, 0x4($0)	Compulsory	Capacity	Capacity	Capacity	Capacity
lw $t2, 0xC($0)	Compulsory	Capacity	Capacity	Capacity	Capacity
lw $t3, 0x8($0)	Compulsory	Capacity	Capacity	Capacity	Capacity

b.

How many bits are needed for the implementation of LRU policy? What is the total cache memory size in number of bits? Include the V bit and the bit(s) used for LRU in your calculations. Show the details of your calculation.

For LRU, we need to hold using order. Since there are only two words, we can specify them with one bit, which shows last recently used. Therefore we can understand it with "1" bit.
Cache size is 2 word, and block size is 1 word, therefore 2 block should exist.
Total bits per block: 1 bit valid bit. Data part 32 bit. 30 bit tag (since last two bit is byte offset, no need to hold them). Per block 63 bits.

Number of blocks * block size + lru bits = 63 * 2 + 1 = 127 bits

c.

Equality Comparator = 2
And Gates = 2
Or Gates = 1
Multiplexers = 1 (2:1)

4

tL1 = 1
MRL1 = 0.2
tL2 =1 + 4(1) = 5
MRL2 = 0.05
tMM = 5 + 10(5) = 55

Amat = tL1 + MRL1(tL2 + MRL2 (tMM))
= 1 + 0.2(5 + 0.05(55))
= 2.55

Execution Time(s) = Cycles / Clock Rate (1/s)

Total Cycles = Number of Instructions × AMAT
= 10^12 × 2.55 = 255×10^10

Clock Rate = 4 × 10^9

Execution Time = (255 × 10^10)/ (4 × 10^9)
= 2550 / 4 = 637.5 s