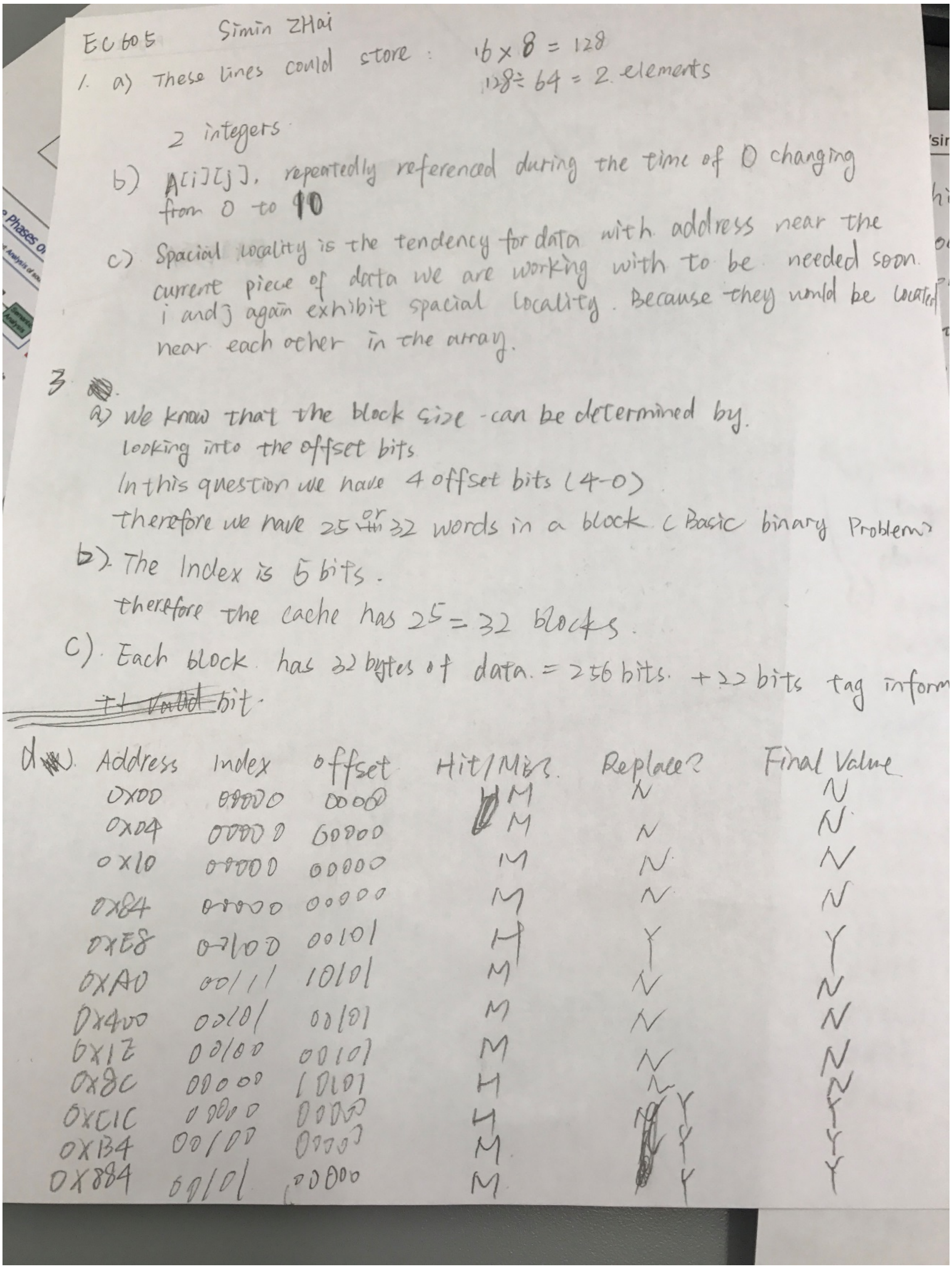


EC605 Simin ZHai

1. a) These lines could store : $16 \times 8 = 128$
$128 \div 64 = 2$ elements

2 integers

b) A[i][j], repeatedly referenced during the time of 0 changing from 0 to 10

c) Spacial locality is the tendency for data with address near the current piece of data we are working with to be needed soon. i and j again exhibit spacial locality. Because they would be located near each other in the array.

3.

a) We know that the block size can be determined by looking into the offset bits

In this question we have 4 offset bits (4-0)

therefore we have 2^5 or 32 words in a block (Basic binary Problem)

b). The Index is 5 bits.

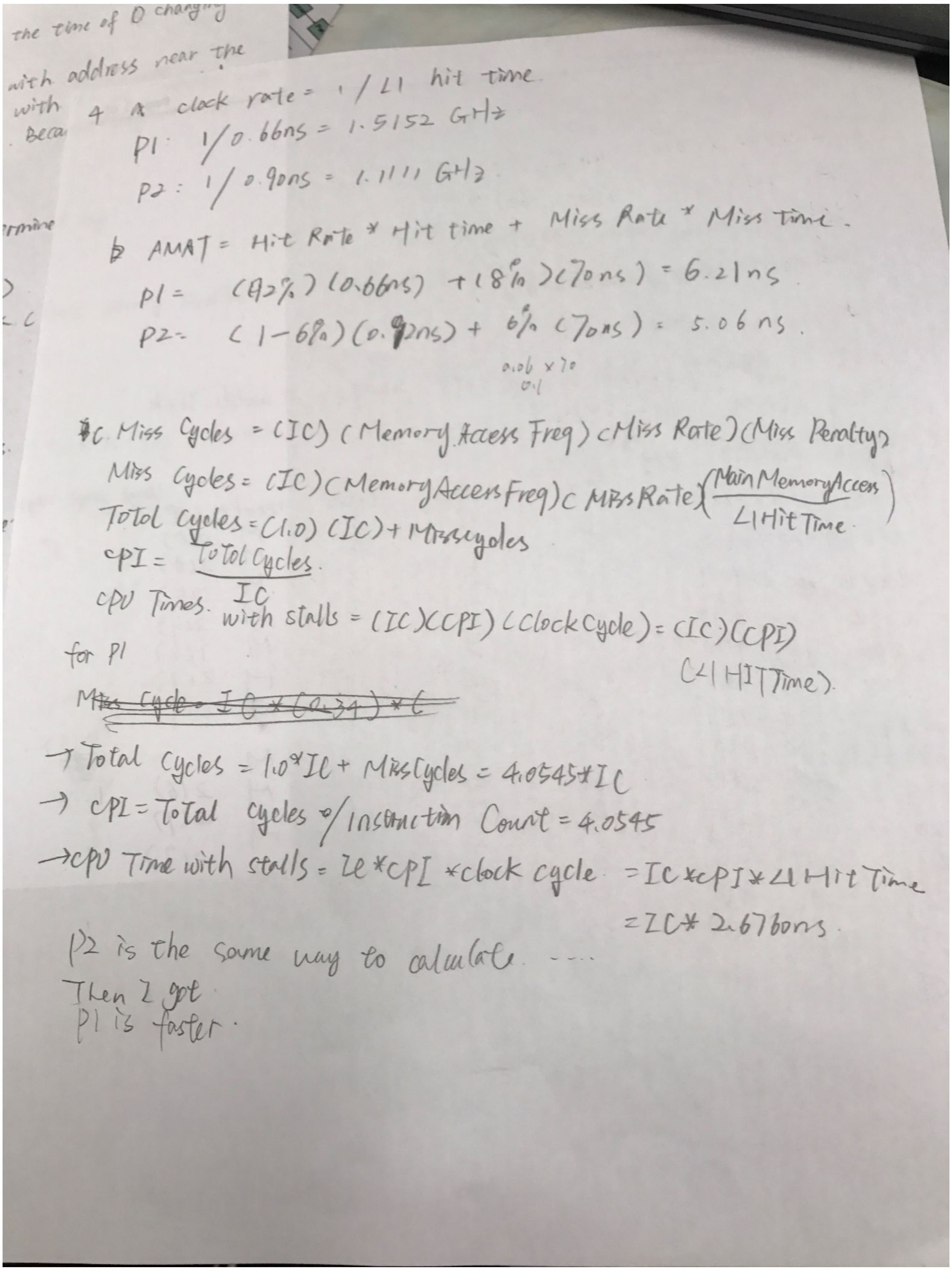
therefore the cache has $2^5 = 32$ blocks.

c). Each block has 32 bytes of data. = 256 bits. + 22 bits tag inform

~~+1 Valid~~ bit.

d).

Address	Index	offset	Hit/Miss.	Replace?	Final Value
0X00	00000	00000	M	N	N
0X04	00000	00000	M	N	N
0X10	00000	00000	M	N	N
0X84	00000	00000	M	N	N
0XE8	00100	00101	H	Y	Y
0XA0	00111	10101	M	N	N
0X400	00101	00101	M	N	N
0X1Z	00100	00101	M	N	N
0X8C	00000	10101	H	N	N
0XC1C	00000	00000	H	Y	Y
0XB34	00100	00000	M	Y	Y
0X884	00101	00000	M	Y	Y



the time of 0 changing

with address near the

with

. Beca

rmine

4 a clock rate = 1 / L1 hit time

P1: 1/0.66ns = 1.5152 GHz

P2: 1/0.90ns = 1.1111 GHz

b AMAT = Hit Rate * Hit time + Miss Rate * Miss Time.

P1 = (92%) (0.66ns) + (8%) (70ns) = 6.21ns

P2 = (1 − 6%) (0.92ns) + 6% (70ns) = 5.06 ns.

0.06 × 70

0.1

c. Miss Cycles = (IC) (Memory Access Freq) (Miss Rate) (Miss Penalty)

Miss Cycles = (IC) (Memory Access Freq) (Miss Rate) $\left(\frac{\text{Main Memory Access}}{\text{L1 Hit Time}}\right)$

Total cycles = (1.0) (IC) + Miss cycles

CPI = $\frac{\text{Total Cycles}}{\text{IC}}$.

CPU Times with stalls = (IC) (CPI) (Clock Cycle) = (IC) (CPI) (L1 HIT Time).

for P1

~~Miss Cycle = IC * (0.34) * C~~

→ Total cycles = 1.0 * IC + Miss Cycles = 4.0545 * IC

→ CPI = Total cycles / Instruction Count = 4.0545

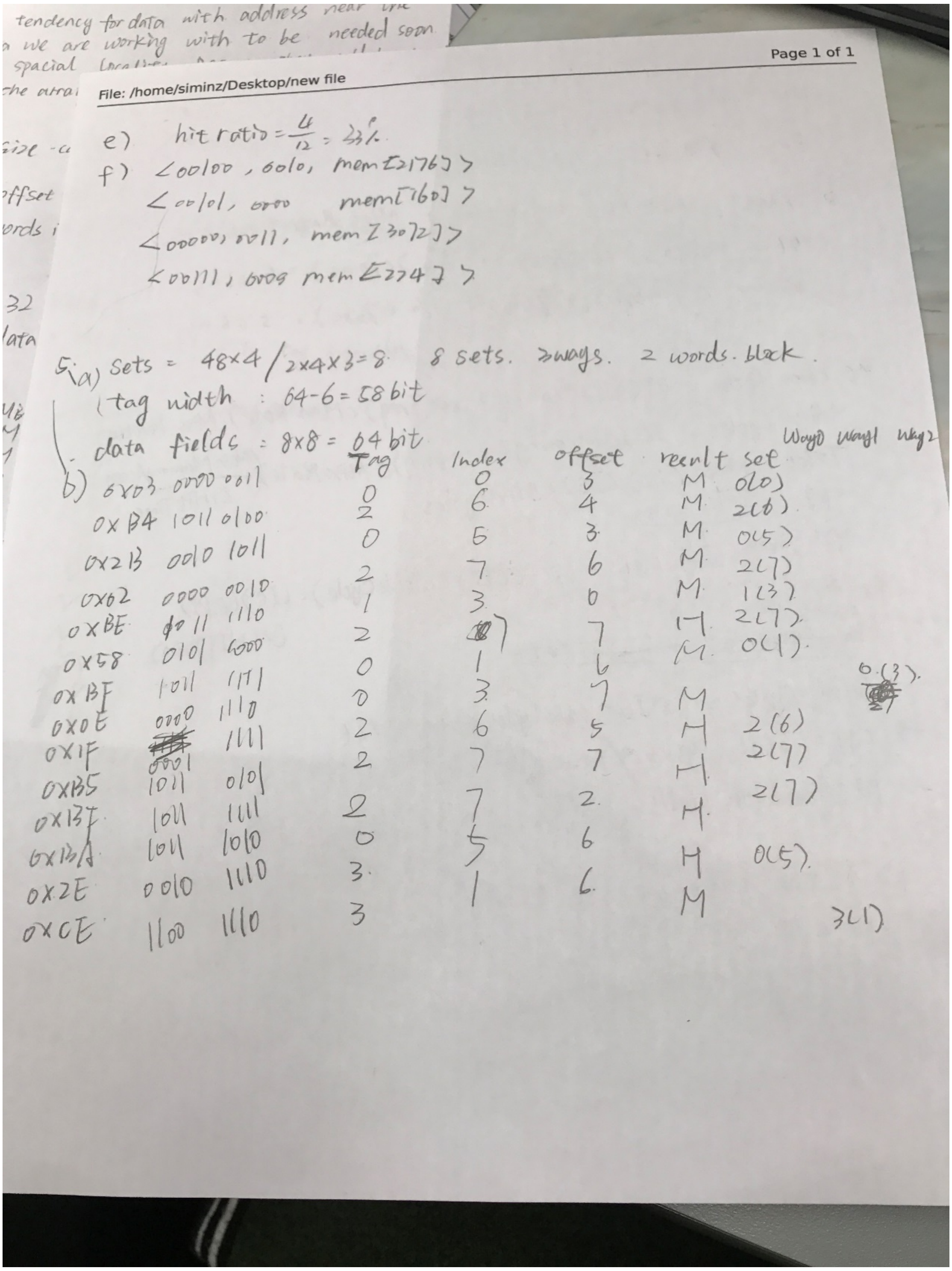
→ CPU Time with stalls = IC * CPI * clock cycle = IC * CPI * L1 Hit Time

= IC * 2.6760ns.

P2 is the same way to calculate ....

Then I got

P1 is faster.



tendency for data with address near the
a we are working with to be needed soon
spacial locali...
the arra

size -c

offset

ords i

32

ata

4B

4

1

File: /home/siminz/Desktop/new file

Page 1 of 1

e) hit ratio = $\frac{4}{12}$ = 33%

f) < 00100 , 0010, mem[2176] >

< 00101, 0000 mem[160] >

< 00000, 0011, mem[3072] >

< 00111, 0009 mem[2274] >

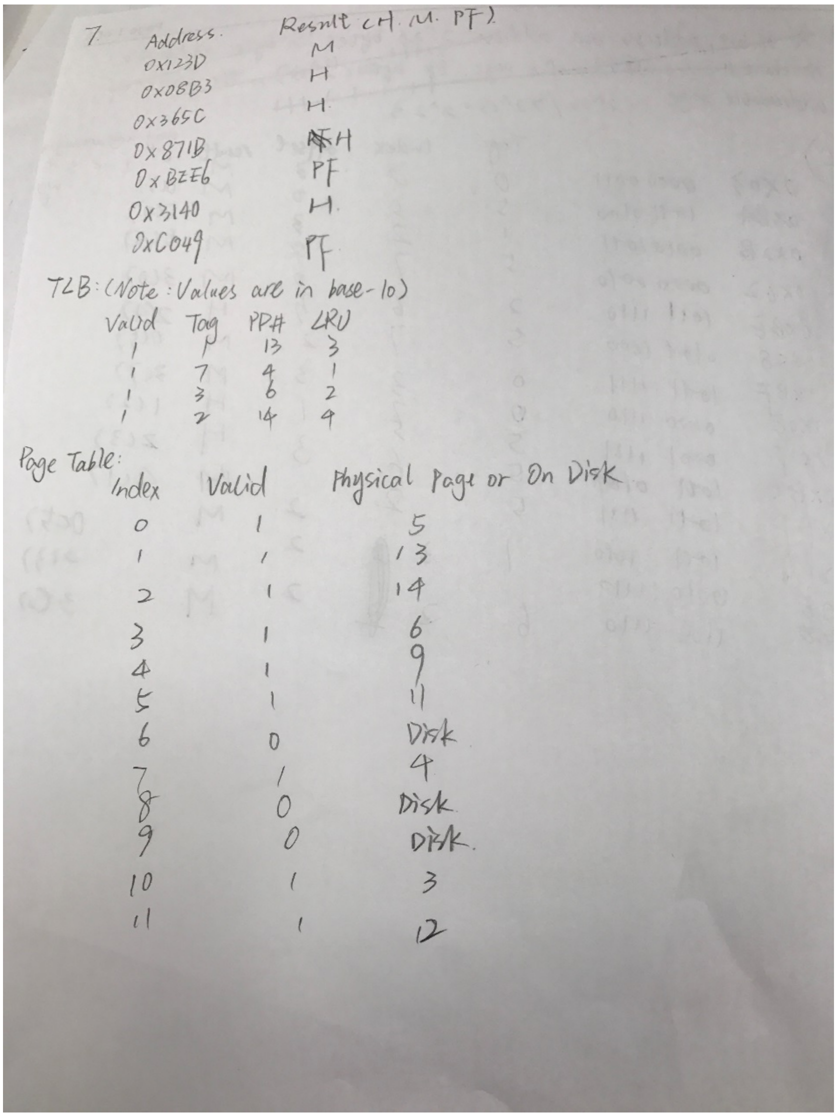
5. (a) Sets = 48×4 / 2×4×3 = 8. 8 sets. 3 ways. 2 words. black.

tag width : 64-6 = 58 bit

data fields : 8×8 = 64 bit

b)

		Tag	Index	offset	result	set	Way0	Way1	Way2
0x03	0000 0011	0	0	3	M	0(0)			
0xB4	1011 0100	2	6	4	M	2(6)			
0x2B	0010 1011	0	5	3	M	0(5)			
0x62	0000 0010	2	7	6	M	2(7)			
0xBE	1011 1110	1	3	0	M	1(3)			
0x58	0101 1000	2	7	7	H	2(7)			
0xBF	1011 1111	0	1	6	M	0(1)			
0x0E	0000 1110	0	3	7	M		0.(3)		
0x1F	~~0001~~ 0001 1111	2	6	5	H	2(6)			
0xB5	1011 0101	2	7	7	H	2(7)			
0xBF	1011 1111	2	7	2	H	2(7)			
0xBA	1011 1010	0	5	6	H	0(5)			
0x2E	0010 1110	3	1	6	M				
0xCE	1100 1110	3					3(1)		



7.

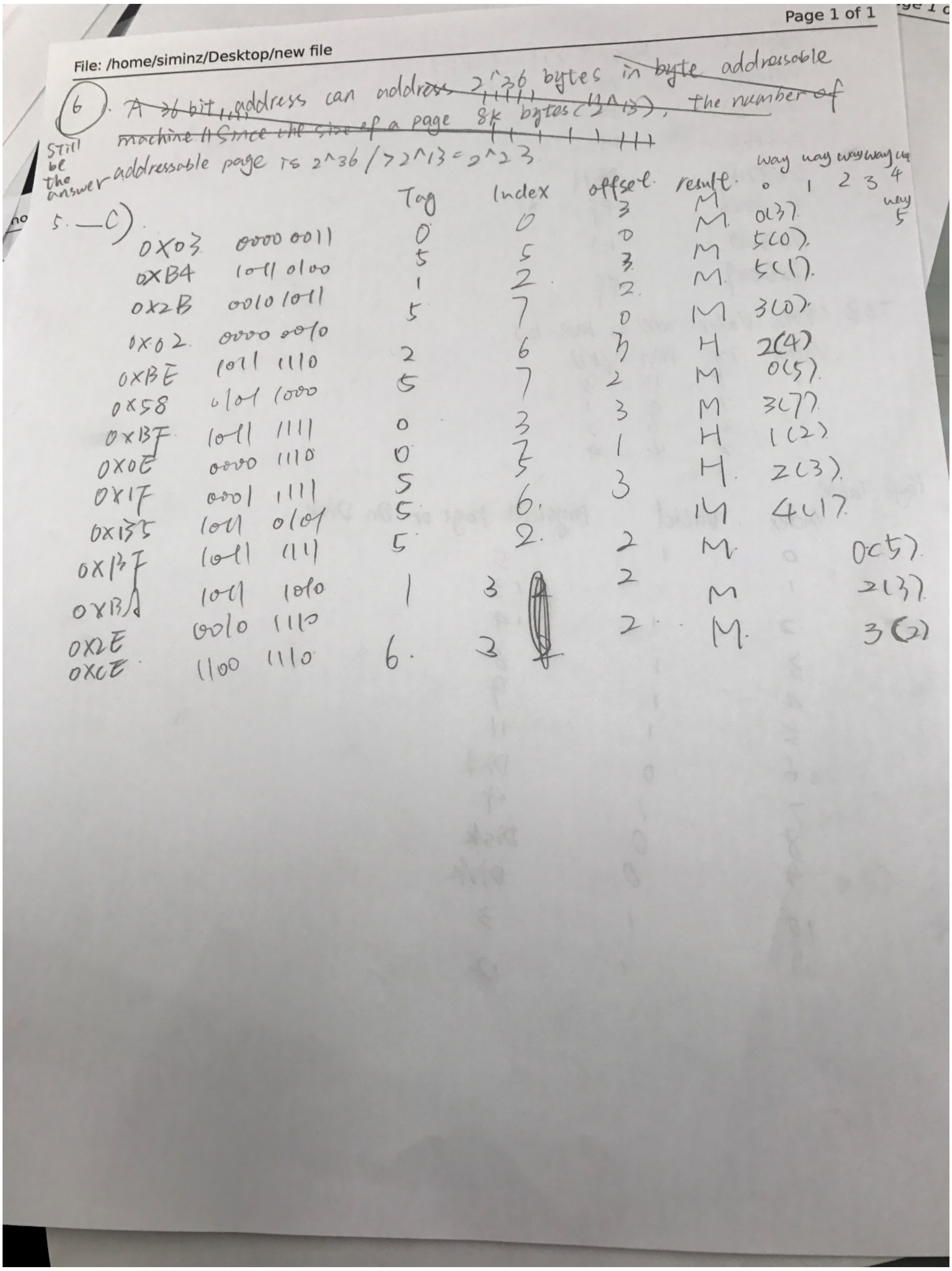
Address.	Result (H. M. PF)
0x123D	M
0x08B3	H
0x365C	H.
0x871B	~~M~~H
0xB2E6	PF
0x3140	H.
0xC049	PF.

TLB: (Note: Values are in base-10)

Valid	Tag	PP#	LRU
1	1	13	3
1	7	4	1
1	3	6	2
1	2	14	4

Page Table:

Index	Valid	Physical Page or On Disk
0	1	5
1	1	13
2	1	14
3	1	6
4	1	9
5	1	11
6	0	Disk.
7	1	4
8	0	Disk.
9	0	Disk.
10	1	3
11	1	12



File: /home/siminz/Desktop/new file

Page 1 of 1

6). ~~A 36 bit~~ address can ~~address 2^36 bytes in byte addressable~~ ~~machine. Since the size of a page 8k bytes (2^13),~~ the number of addressable page is 2^36 / 2^13 = 2^23

Still be the answer

5. —C)

		Tag	Index	offset	result M	way way way way way 0 1 2 3 4	way 5
0X03	0000 0011	0	0	3	M.	0(3).	
0XB4	1011 0100	5	5	0	M	5(0).	
0X2B	0010 1011	1	2.	3.	M.	5(1).	
0X02.	0000 0010	5	7	2.	M	3(0).	
0XBE	1011 1110	2	6	0	H	2(4)	
0X58	0101 1000	5	7	h	M	0(5).	
0XBF	1011 1111	0	3	2	M	3(7).	
0X0E	0000 1110	0	7	3	H	1(2)	
0X1F	0001 1111	5	5	1	H.	2(3).	
0X1B5	1011 0101	5	6.	3	IM	4(1).	
0X1BF	1011 1111	5	2.	2	M.		0(5).
0XBA	1011 1010	1	3	2	M		2(3).
0X2E	0010 1110			2.	M.		3(2)
0XCE	1100 1110	6.	3				