Computer Architecture

Homework - 6:

1.
```
for(i=0; i<8; i++)
    for(j=0; j<8000; j++)
        A[i][j] = B[i][0] + A[i][j]
```

a) Each 32 bit integer requires 4 cache blocks to store, therefore two 32-bit integers can be stored in the 8 byte cache block.

b) variables i, j & B[i][0] are referenced frequently and exhibit temporal locality.

c) Variables A[i][j] & B[i][0] exhibit spatial locality

d) Total number of elements in the array = 8000 × 8 = 64000 - each 32 bit.
~~This means we require 8000 bytes of~~
We can represent the above data using 64000/4 = 16000 bytes.
This would require 16000/8 = 2000 cache blocks.

e) i, j & B[i][0] represent temporal locality

f) A[i][j] & B[i][0] represent spatial locality.

2. a)

Word Address	Binary	Tag	Index	Hit/Miss
3	00000011	0000	0011	M
180	10110100	1011	0100	M
34	00100010	0010	0010	M
2	00000010	0000	0010	M
157	10011101	1001	1101	M
88	01011000	0101	1000	M
190	10111110	1011	1110	M
24	00011000	0001	1000	M
187	10111011	1011	1011	M
34	00100010	0010	0010	H
186	10111010	1011	1010	M
253	11111101	1111	1101	M

b)

Word Address	Binary	Tag	Index	Hit/Miss
3	00000011	0	1	M
180	10110100	11	01	M
34	00100010	2	01	M
2	00000010	0	0	M
157	10011101	9	13	M
88	01011000	5	12	M
190	10111110	11	07	M
24	00011000	1	04	M
187	10111011	11	05	M
34	00100010	2	01	H
186	10111010	11	05	H
253	11111101	15	13	M

3 a) Assuming the stream represents byte addresses the first access will be a miss. The following 31 bits will be hit. Then the next 32nd bit will be a miss and so on:
So there will be a miss rate of 1/16.

b) Cache size of 16 bytes - 1/8.
Cache size of 64 bytes - 1/32
Cache size of 128 bytes - 1/64.
This workload exploits a spatial locality.

4. Input Stream: 4669, 2227, 12916, 34587, 45870, 12608, 45255

We can divide by (4096) 4kb & and take integer

Value for	VPN	TLB	Page Table	Page fault
4669	1	Miss	Miss	Yes
2227	0	Miss	Hit	No
12916	3	Hit	-	No
34587	8	Miss	Miss	Yes
45870	11	Hit	-	No
12608	3	Hit	-	No
45255	11	Hit	-	No

TLB

Valid	Tag	PPN
1	11	12
1	8	14
1	3	6
1	11	13.

Valid	PPN
1	5
1	~~Data~~ 13.
0	disk
1	6
1	9
1	11
0	disk
1	4
0	~~disk~~ 14
0	disk
1	3
1	12

2c) 8 blocks = 3 bit index

decimal	Binary	Tag	Index	hit/miss
3	00000011	00000	011	M
180	10110100	10110	100	M
34	00100010	00100	010	M
2	00000010	00000	010	M
157	10011101	10011	101	M
88	01011000	01011	000	M
190	10111110	10111	110	M
24	00011000	00011	000	M
181	10110101	10110	101	M
34	00100010	00100	010	M

186	1011 1010	10111	010	M
253	1111 1101	11111	101	M

Cache Miss rate = 12/12

- 4blocks = 2 bit index

	Tag	Index	Offset	hit/miss
3	000000	01	1	miss
180	10110	10	0	miss
34	00100	01	0	miss
2	00000	01	0	hit
157	10011	10	1	miss
88	01011	00	0	miss
190	10111	11	0	miss
24	00011	00	0	Miss
187	10110	10	1	hit
34	00100	01	0	hit
186	10111	01	0	miss
253	11111	10	1	miss

Cache miss rate = 9/12

2 blocks w/ 4 words blocks - 2 bit index 2 bit offset

	Tag	index	offset	hit/miss
3	00000	0	11	M
180	10110	1	00	M
34	00010	0	10	M
2	00000	0	10	H
137	10011	1	01	M
88	01011	0	00	M
190	10111	1	10	M
24	00011	0	00	M
181	10110	1	01	M
34	00100	0	10	H
186	10111	0	10	M
253	11111	1	01	M

Cache hit rate = 1/12

Total cycles for C1 = (12×2) + (8×25) = 324 cycles

Total cycles for C2 = (12×3) + (9×25) = 261 cycles

Total cycles for C3 = (10×5) + (10×25) = 310 cycles

Cache miss rate: C1 = 100 %

C2 = 75 % ← Best instruction miss rate

C3 = 83.33 %

C2 also uses least no. of clock cycles & is hence the best design.