CS 3340.004
Drew Pulliam - DTP180003

252, 189, 181, 44, 14, 190, 6, 42, 175, 8, 180, 12

252 - 1111 1100	190 - 1011 1110	8 - 0000 1000
189 - 1011 1101	6 - 0000 0110	180 - 1011 0100
44 - 0010 1100	42 - 0010 1010	12 - 0000 1100
14 - 0000 1110	175 - 1010 1111	181 - 1011 0101

(a)

1 word $= 2^2 = 4$ bytes
2 bits used for offset

total of 8 blocks
$8 = 2^3 =$ 3 bits used for index

remaining $32 - 3 - 2 = 27$ bits used for tag (Only show 3 bits of tag, others are all zero)

Decimal	Binary	Offset	index	tag	Hit/Miss
252	1111 1100	00	111	111	M
189	1011 1101	01	111	101	M (replace)
181	1011 0101	01	101	101	M
44	0010 1100	00	011	001	M
14	0000 1110	10	011	000	M (replace)
190	1011 1110	10	111	101	H
6	0000 0110	10	001	000	M
42	0010 1010	10	010	001	M
175	1010 1111	11	011	101	M (replace)
8	0000 1000	00	010	000	M (replace)
180	1011 0100	00	101	101	H
12	0000 1100	00	011	000	M (replace)

CS 3340
Drew Pulliam DTP180003

(b) 2 word block = $8 = 2^3$ bytes | total 8 blocks | 26 bits tag
3 bits offset | 2^3 = 3 bit index

Decimal	Binary	offset	index	tag	Hit/Miss
252	1111 1100	100	111	11	M
189	1011 1101	101	111	10	M (replace)
181	1011 0101	101	110	10	M
44	0010 1100	100	101	00	M
14	0000 1110	110	001	00	M
190	1011 1110	110	111	10	H
6	0000 0110	110	000	00	M
42	0010 1010	010	101	00	H
175	1010 1111	111	101	10	M (replace)
8	0000 1000	000	001	00	H
180	1011 0100	100	110	10	H
12	0000 1100	100	001	00	H

(c) 2 way associative w/ 8 blocks = $\frac{8}{2} = 4$ sets $= 2^2$ 2 bits index
2 word block = 3 bit offset | 27 bit tag

Decimal	Binary	offset	index	tag	Hit/Miss
252	1111 1100	100	11	111	M
189	1011 1101	101	11	101	M
181	1011 0101	101	10	101	M
44	0010 1100	100	01	001	M
14	0000 1110	110	01	000	M
190	1011 1110	110	11	101	H
6	0000 0110	110	00	000	M
42	0010 1010	010	01	001	H
175	1010 1111	111	01	101	M (replace)
8	0000 1000	000	01	000	H
180	1011 0100	100	10	101	H
12	0000 1100	100	01	000	H

CS 3340
Drew Pulliam DTP180003

(d) 4 way associative $= \frac{8}{4} = 2^1 = 1$ bit index
3 bit offset, 28 bit tag

Dec	Bin	offset	index	tag	H/M
252	1111 1100	100	1	1111	M
189	1011 1101	101	1	1011	M
181	1011 0101	101	0	1011	M
44	0010 1100	100	1	0010	M
14	0000 1110	110	1	0000	M
190	1011 1110	110	1	1011	H
6	0000 0110	110	0	0000	M
42	0010 1010	010	1	0010	H
175	1010 1111	111	1	1010	M (replace)
8	0000 1000	000	1	0000	H
180	1011 0100	100	0	1011	H
12	0000 1100	100	1	0000	H

(e) in terms of miss rate, largest block size possible is usually best
in this example that is C4 with 16 word blocks

C1 - 16 x 2-word blocks - 3 bit offset, 4 bit index, 25 bit tag
C2 - 8 x 4-word blocks - 4 bit offset, 3 bit index, 25 bit tag
C3 - 4 x 8-word blocks - 5 bit offset, 2 bit index, 25 bit tag
C4 - 2 x 16-word blocks - 6 bit offset, 1 bit index, 25 bit tag

(C1)

Binary	tag	index	H/M
1111 1100	1	1111	M
1011 1101	1	0111	M
1011 0101	1	0110	M
0010 1100	0	0101	M
0000 1110	0	0001	M
1011 1110	1	0111	M
0000 0110	0	0000	H
0010 1010	0	0101	M
1010 1111	1	0101	H
0000 1000	0	0001	M
1011 0100	1	0110	H
0000 1100	0	0001	H

7/12 Miss ratio

(C2)

tag	index	H/M
1	111	M
1	011	M
1	011	H
0	010	M
0	000	M
1	011	H
0	000	H
0	010	H
1	010	M
0	000	H
1	011	H
0	000	H

5/12 Miss ratio

(C3)

tag	index	H/M
1	11	M
1	01	M
1	01	H
0	01	M
0	00	M
1	01	M
0	00	H
0	01	M
1	01	M
0	00	H
1	01	H
0	00	H

7/12 Miss ratio

(C4)

tag	index	H/M
1	1	M
1	0	M
1	0	H
0	0	M
0	0	H
1	0	M
0	0	M
0	0	H
1	0	M
0	0	M
1	0	M
0	0	M

9/12 Miss ratio

CS 3340
Drew Pulliam DTP 180003

(e) continued

AMAT = access time + miss penalty • miss rate

$AMAT_1 = 2 + 25 \cdot 7/12 = 16.58$

$AMAT2 = 3 + 25 \cdot 5/12 = 13.42$

$AMAT3 = 5 + 25 \cdot 7/12 = 19.58$

$AMAT4 = 6 + 25 \cdot 9/12 = 24.75$

C2 has smallest AMAT, so C2 is best cache design

(f) $64\ KiB = \dfrac{2^{14} \text{ words}}{16 = 2^4 \text{ words}} = 2^{10}$ blocks

each block has $16 \times 32 = 2^9$ bits of data

offset = 6 bits, index = 10 bits

tag = $32 - 10 - 6 = 16$

each block also has a valid bit

so total bits per block = $2^9 + 16 + 1 = 529$ bits

total cache = 2^{10} blocks • 529 bits = $\boxed{541,696 \text{ bits}}$

next smallest 4-word block cache

total bits per block would be $2^7 + 16 + 1 = 145$ bits

total cache = $2^{12} \times 145 = 593,920$ = TOO BIG

$= 2^{11} \times 145 = \boxed{296,960 \text{ bits}}$

the second cache might have better performance because of a better access time