```
HW5
5.1
   5.1.1 \quad \frac{1bbyte}{64bit} = \frac{16 \times 8 bit}{64bit} = 2
   5.1.2
           I, J, B[] [] exhibit temporal locality
  5.1.3 ACIJEJJ exhibit spatial locality
  5.1.4
         I, J, B[]] [Dexhibit temporal locality
  1.1.5 A(J,1) exhibit spatial locality
 (5.1.6)
        16 byte block holds 2 words
        for matrix A, whether using C or Matlab to store,
                     the size is 8000 x 8000, thus the block num = \frac{8000 \times 8000}{2} = 3.2 \times 10^7
        for matrix B,
        using Matlab's matrix storage: B(I.O) is contiguous in memory
          for every two I, B(I,O) holds a new block
         thus the block num = 4
       using C's matrix storage: B(I.O) is not contiguous in memory
         for each I, B(I,O) holds a new block
         thus the block num = 8
5.2
   5.21 cache performance: direct-mapped, 16 blocks -> 4 bit index
         block size = 1 word -> block offset = 0 bit
               word address
               tag index
```

0x03 00000011 0000 0011 miss 0x44 1011 0100 1011 0100 miss 0x2b 0010 1011 0010 1011 miss 0x02 0000 0010 0000 0010 miss 0xbf 1011 1111 1011 1111 miss 0x58 0101 1000 0101 1000 miss 0xbe 1011 110 1011 1110 miss 0x00 00001110 0000 1110 miss 0xbt 1011 1010 1011 1010 miss 0xbt 1011 1010 1011 1010 miss 0xbd 1011 1010 1011 1010 miss 5.2.2 Cache performance: direct-mapped , & blocks → 3 bit index block size = 2 word → block offet = 1 bit 1111 111 miss 0x03 00000011 0000 001 1 miss 0xbd 1011 0100 1011 0010 0 miss 0xbd 1011 0100 1011 11 miss 0xbd 0010 1011 0010 101 1 miss 0xbd 1011 111 1011 111 11 1 miss 0xbd 1011 111 1011 111 1 miss 0xbc 1011 1110 1011 111 1 miss 0xbc 1011 1110 1011 111 0 miss 0xbc 10101 1010 0000 110 0 miss 0xbc 10101 1010 0000 110 0 miss 0xbc 10101 1010 1011 111 0 miss 0xbc 10101 1010 1011 111 0 miss 0xbc 10101 1010 1011 111 0 miss 0xbc 10101 1010 1011 1010 0 miss 0xbc 1011 1100 1011 1010 0 miss 0xbc 1011 1100 1011 1010 0 miss 0xbc 1011 1101 1011 111 0 miss 0xbc 1011 1101 1011 111 0 miss	0xb4	0xb4	hex add	r binary addr	tag	index	hit/mis	\$	
0x2b 0010 1011 0010 1011 miss 0x02 0000 0010 0000 0010 miss 0xbf 1011 1111 1011 1111 miss 0xts 01011000 0101 1000 miss 0xbe 1011110 1011 1110 miss 0x0e 00001110 0000 1110 miss 0xbt 10110101 1011 0101 miss 0xbt 1011010 1011 1010 miss 0xba 10111010 1011 1010 miss 0xta 10111010 1011 1010 miss 0xfd 1111101 1111 1101 miss 5.2.2 Cache performance: direct-mapped, & biocks \rightarrow 3 bit index block size = 2 word \rightarrow block offet = 1 bit 111 111 word address 10x03 00000011 0000 001 1 miss 0x04 10110100 1011 0010 101 1 miss 0x02 0000 0010 0000 001 0 miss 0xbf 1011111 1011 111 1 miss 0xbf 1011111 1011 111 1 miss 0xbg 1011000 0101 1001 0 miss 0xbe 10111110 1011 111 0 miss 0xbe 1011110 1011 111 0 miss 0xbt 1010101 0000 111 0 miss 0xbt 1010101 1011 101 0 miss 0xbt 1010101 1011 1011 0 miss 0xbt 1010101 1011 1010 1 miss 0xba 1011000 1011 000 110 0 miss 0xba 1011000 1011 000 110 0 miss 0xba 1011010 1011 1010 0 miss 0xba 1011010 1011 101 0 miss	0x2b 0010 1011 0010 1011 miss 0x0b 0010 0010 0000 0010 miss 0xbf 1011 1111 1011 1111 miss 0xbe 1011110 1011 1110 miss 0x0e 00001110 0000 1110 miss 0xbt 1011010 1011 0110 miss 0xbt 1011010 1011 1110 miss 0xbt 1011010 1011 1010 miss 0xba 10111010 1011 1010 miss 0xba 10111010 1011 1010 miss 0xfd 11111101 1111 1101 miss 5.2.2 Cache performance: direct-mapped, & blocks \rightarrow 3 bit index block size = 2 word \rightarrow block offet = 1 bit	0x1b 0010 [011 0010 1011 miss 0x02 0000 0010 0000 0010 miss 0xbf 1011 111 1011 1111 miss 0xb8 01011000 0101 1000 miss 0xbe 10111110 1011 1110 miss 0x0e 00001110 0000 1110 miss 0xbt 10110101 1011 0101 miss 0xbt 10110101 1011 1010 miss 0xba 10111010 1011 1010 miss 0xfd 1111101 1111 1101 miss 0xfd 1111101 1111 1101 miss 52.2 Cache performance: direct-mapped, & blocks \rightarrow 3 bit index block size = 2 word \rightarrow block offet = 1 bit 11	0x03	00000011	0000	0011	miss		
0x02 0000 0010 0000 0010 miss 0xbf 1011 1111 1011 1111 miss 0xxxxx 01011000 0101 1000 miss 0xbe 1011110 1011 1110 miss 0x0e 00001110 0000 1110 miss 0xbt 1011010 1011 1010 miss 0xbt 1011010 1011 1010 miss 0xba 10111010 1011 1010 miss 0xfd 11111101 1111 1101 miss 5.2.2 cache performance: direct-mapped, & blocks → 3 bit index block size = 2 word → block offet = 1 bit	0x02 0000 0010 0000 0010 miss 0xbf 1011 1111 1011 1111 miss 0xxb8 01011000 0101 1000 miss 0xbe 1011110 1011 1110 miss 0x0e 00001110 0000 1110 miss 0xbt 1010101 1011 0101 miss 0xbt 1010101 1011 1010 miss 0xbd 10111010 1011 1010 miss 0xbd 10111010 1011 1010 miss 0xbd 10111010 1011 1010 miss 0xfd 11111101 1111 1101 miss 5.2.2 cache performance: direct-mapped, 8 biocks -> 3 bit index block size = 2 word -> block offet = 1 bit	0x02 0000 0010 0000 0010 miss 0xbf 1011 1111 1011 1111 miss 0xx8 01011000 0101 1000 miss 0xbe 1011110 1011 1110 miss 0x0e 00001110 0000 1110 miss 0xbt 1010101 1011 0101 miss 0xbt 1010101 1011 1010 miss 0xbd 10111010 1011 1010 miss 0xfd 1111101 1111 1101 miss 0xfd 1111101 1111 1101 miss 0xcd cache performance: direct-mapped, 8 blocks \rightarrow 3 bit index 0xblock size = 2 word \rightarrow block offet = 1 bit 0x03 00000011 0000 001 1 miss 0xb4 1011 0100 1011 0100 0 miss 0xb6 1011 1111 1011 111 1 miss 0xb6 1011 1111 1011 111 1 miss 0xb6 1011 1110 1011 111 0 miss 0xb6 1011 1110 1011 111 0 miss 0xb7 1010101 1011 111 0 miss 0xb8 1010100 0101 1011 111 0 miss 0xb1 1010101 1011 111 0 miss 0xb2 0000110 0000 111 0 miss 0xb1 1010101 1011 111 0 miss 0xb2 0000110 0000 111 0 miss 0xb1 1010101 1011 1010 0 miss 0xb2 0xb1 1010101 1011 1010 0 miss 0xb2 0xb1 1010101 1011 1010 0 miss 0xb2 0xb1 1010101 1011 1010 0 miss	0 x b4	10110100	(01)	0100	miss		
0xbf	0xbf	0xbf 1011 111 1011 1111 miss 0xb8 0101 1000 0101 1000 miss 0xbe 1011 1110 1011 1110 miss 0x0e 00001110 0000 1110 miss 0xbr 1010 1010 1011 0101 miss 0xbr 1011 1010 1011 1010 miss 0xbr 1011 1010 1011 1010 miss 0xbr 1011 1010 1011 1110 miss 0xbr 1011 1010 1111 1101 miss 0xfr 1111 101 1111 1101 miss 0xfr 1111 111 mord address 0xr 1011 1010 1011 1010 miss 0xr 1011 1010 1011 1010 miss 0xbr 1011 1010 1011 1011 111 1 0xbr 1011 1011 111 1 0xbr 1011 111 1011 111 1 0xbr 1011 1000 1011 1010 miss 0xbr 1011 1110 1011 111 111 0 0xbr 1011 110 1011 111 0 0xbr 1011 110 1011 111 0 0xbr 1011 1010 1011 1010 miss 0xbr 1011 1010 1011 1010 0 0xbr 1011 1010 1010 0 0xbr 1011	0×2b	0010 1011	0010	1011	miss		
0x58	0x58	0x58 0 01 1000 0 01 1000 miss 0xbe 10 1 1110 miss 0x0e 0000 1110 miss 0xbr 10 10 0 10 1 10 0 miss 0xbr 10 10 0 00 0 1100 miss 0xbr 10 10 0 10 1 10 0 miss 0xbr 10 10 0 10 1 10 0 miss 0xbr 10 10 0 10 1 10 0 miss 0xbr 10 10 0 10 1 1100 miss 0xfd 11 11 0 11 1 1101 miss 0xfd 11 11 0 11 1 1101 miss 0xfd 11 11 0 10 1 10 0 miss 0xord index block offset bit	0x02	0000 0010	0000	0010	miss		
0x58	0x5-8	0x58 01011000 0101 1000 miss 0xbe	oxbf	1011 1111	1011	[]11	miss		
0x0e 00001110 0000 1110 miss 0xbt 10110101 1011 1010 miss 0xba 1011010 1011 1010 miss 0xfd 11111101 1111 1101 miss 5.2.2 cache performance: direct-mapped, & blocks -> 3 bit index block size = 2 word -> block offret = 1 bit 111 word address 1003 00000011 0000 001 1 miss 0xb4 (0110100 (011) 010 0 miss 0x2b 0010 1011 0010 101 1 miss 0x02 0000 0010 0000 001 0 miss 0xbf 1011111 1011 111 1 miss 0xfs 01011000 0101 1010 0 miss 0xbe 10111110 1011 111 1 miss 0xbe 10111110 1011 111 0 miss 0xbb 1011010 1011 111 0 miss 0xbb 1011010 1011 111 0 miss 0xbb 10111110 1011 111 0 miss 0xbb 10111110 1011 111 0 miss 0xbb 1011110 1011 111 0 miss 0xbb 1011110 1011 111 0 miss	0x0e 00001110 0000 1110 miss 0xbt 10110101 1011 0101 miss 0xbc 0010100 0010 1100 miss 0xba 10111010 1011 1010 miss 0xfd 1111101 1111 1101 miss 5.2.2 Cache performance: direct-mapped, & blocks -> 3 bit index block size = 2 mord -> block offiet = 1 bit 1	0x 0e 0000110 0000 1110 miss 0xbt 10110101 1011 1010 miss 0xbc 0010100 0010 1100 miss 0xba 10111010 1011 1010 miss 0xfa 1111101 1111 1101 miss 5.2.2 Cache performance: direct-mapped, & blocks → 3 bit index block size = 2 voord → block offiet = 1 bit		01011000	0101	1000	miss		
0xbt 10110101 1011 0101 miss 0x2c 00101100 0010 1100 miss 0xba 10111010 1011 1010 miss 0xfd 1111101 1111 1101 miss 5.2.2 cache performance: direct-mapped, 8 blocks -> 3 bit index block size = 2 word -> block officet = 1 bit 1	0xbt 10110101 1011 0101 miss 0x2c 00101100 0010 1100 miss 0xba 10111010 1011 1010 miss 0xfol 1111101 1111 1101 miss 5.2.2 cache performance: clirect-mapped, 8 biocks -> 3 bit index block size = 2 mord -> block offret = 1 bit 1	0xbs 1011010 1011 0101 miss 0x2c 00101100 0010 1100 miss 0xba 10111010 1011 1010 miss 0xfol 1111101 1111 1101 miss 5.2.2 cache performance: direct-mapped, 8 blocks → 3 bit index	oxbe	10111110	loil	1110	miss		
0x2c 00101100 0010 1100 miss 0xba 1011 1010 1011 1010 miss 0xfol 11111101 1111 1101 miss 0xfol 11111101 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 10x0e 0000110 0000 110 0 miss 0xbs 1011010 1011 1010	0x2c 00101100 0010 1100 miss 0xba 10111010 1011 1010 miss 0xfol 11111101 1111 1101 miss 5.2.2 cache performance: direct-mapped, 8 blocks → 3 bit index block size = 2 mord → block offset = 1 bit	0x2c 00101100 0010 1100 miss 0xba 1011010 1011 1010 miss 0xfol 1111101 1111 1101 miss 5.2.2 cache performance: direct-mapped, 8 blocks → 3 bit index block size = 2 word → block offet = 1 bit	ox oe	00001110	0000	1110	miss		
0xba 1011 1010 1011 1010 miss 5.2.2	0xba 1011 1010 1011 1010 miss 0xfol 1111 1101 1111 1101 miss 5.2.2	0xba 1011 1010 1011 1010 miss 0xfol 1111 1101 1111 1101 miss 5.2.2 cache performance: direct-mapped, 8 blocks → 3 bit index block size = 2 word → block offset = 1 bit	0xb5	10110101	1011	0101	miss		
0xfol	0xfol	0xfol	0x2c	00101100	0010	1100	miss		
5.2.2 cache performance: direct-mapped, 8 blocks -> 3 bit index block size = 2 word -> block offret = 1 bit	5.2.2 cache performance: clirect-mapped, 8 blocks \(\rightarrow 3 \) bit index block size = 2 word \(\rightarrow block \) offset = 1 bit \[\begin{arrow} \q	5.2.2 cache performance: direct-mapped, 8 blocks -> 3 bit index block size = 2 mord -> block offeet = 1 bit	oxba	1011 1010	1011	1010	miss		
5.2.2 cache performance: direct-mapped, 8 blocks -> 3 bit index block size = 2 word -> block offet = 1 bit	5.2.2 cache performance: direct-mapped, 8 blocks -> 3 bit index block size = 2 word -> block offret = 1 bit	5.2.2 Cache performance: direct-mapped, 8 blocks -> 3 bit index block size = 2 voord -> block offret = 1 bit	oxfd	111) 1101	Int	(10)	miss		
block size = $2 \text{ word} \rightarrow block \text{ offret} = 1 \text{ bit}$ \[\text{Times } \text{ word} \text{ word} \text{ address} \] \[\text{tag} \text{ index} \text{ block offret} \] \[\text{hex addr} \text{ binary addr} \text{ tag} \text{ index} \text{ offret} \] \[\text{hit/miss} \text{ oxo3} \text{ 000000011 \text{ 0000 0001 0} 000 001 1 \text{ miss} \] \[\text{0xb4} \text{ 1011 0100 } \text{ 1011 } \text{ 0010 } \text{ 101 } \text{ 101 } \text{ 1010 } \text{ 101 } \text{ 1011 } \text{ 111 } \text{ 111 } \text{ miss} \] \[\text{0x02} \text{ 0000 0010 } \text{ 0000 } \text{ 0010 } \text{ 1011 } \text{ 111 } \text{ 101 } \text{ 111 } \text{ 111 } \text{ miss} \] \[\text{0xb4} \text{ 1011 1000 } \text{ 0101 } \text{ 1010 } \text{ 101 } \text{ 111 } \text{ 0 } \text{ miss} \] \[\text{0xb6} \text{ 1011 1010 } \text{ 1011 } \text{ 101 } \text{ 010 } \text{ 10 } \text{ miss} \] \[\text{0xb6} \text{ 1011 1010 } \text{ 1010 } \	block size = 2 mord → block offet = 1 bit	block size = $2 \text{ mord} \rightarrow block \text{ offret} = 1 \text{ bit}$							
0x03 00000011 0000 001 1 miss 0xb4 (0110100 (01) 010 0 miss 0x2b 0010 1011 000 101 1 miss 0x02 0000 0010 0000 001 0 hit 0xbf 1011 1111 1011 111 1 miss 0xs8 0101 1000 0101 100 0 miss 0xbe 1011 110 1011 111 0 hit 0x0e 00001110 0000 111 0 miss 0xbs 1011 0101 1011 010 1 hit 0xbc 0010 1100 0010 110 0 miss 0xbs 1011 1010 1011 100 0 miss 0xba 1011 1010 1011 101 0 miss	0x03 00000011 0000 001 1 miss 0xb4 (011 0100 (01) 010 0 miss 0x2b 0010 1011 000 101 1 miss 0x02 0000 0010 0000 001 0 hit 0xbf 1011 111 1011 111 1 miss 0xs8 0101 1000 0101 100 0 miss 0xbe 1011 110 1011 111 0 hit 0x0e 00001110 0000 111 0 miss 0xbs 1011 0101 1011 00 1 hit 0xba 1011 1010 1011 101 0 miss	0x03 00000011 0000 001 1 miss 0xb4 (011000 (01) 000 0 miss 0x2b 0010 1011 0000 101 1 miss 0x02 0000 0010 0000 001 0 hit 0xbf 1011 1111 1011 111 1 miss 0xs8 0101 1000 0101 100 0 miss 0xbe 1011 1110 1011 111 0 hit 0x0e 00001110 0000 111 0 miss 0xbs 1011 0101 101 00 1 hit 0xba 1011 1010 1011 101 0 miss							
0xb4 (011 0100 (01) 010 0 miss 0x2b 0010 1011 0010 101 1 miss 0x02 0000 0010 0000 001 0 hit 0xbf 1011 1111 1011 111 1 miss 0xbs 1011 1000 0101 100 0 miss 0xbe 1011 1100 1011 111 0 hit 0x0e 00001110 0000 111 0 miss 0xbt 1011 0101 1011 010 1 hit 0xbc 0010 1100 0010 110 0 miss 0xbc 1011 1010 1011 101 0 miss	0xb4 (011 0100 (01) 010 0 miss 0x2b 0010 1011 0010 101 1 miss 0x02 0000 0010 0000 001 0 hit 0xbf 1011 1111 1011 111 1 miss 0xbs 1011 1000 0101 100 0 miss 0xbe 1011 110 1011 111 0 hit 0xbe 1011 1010 1011 010 1 hit 0xbt 1011 1010 1011 010 1 hit 0xba 1011 1010 1011 101 0 miss	0xb4 (011 0100 (01) 010 0 miss 0x2b 0010 1011 0010 101 1 miss 0x02 0000 0010 0000 001 0 hit 0xbf 1011 1111 1011 111 1 miss 0xbs 1011 1000 0101 100 0 miss 0xbe 1011 110 1011 111 0 hit 0xbe 1011 1010 1011 010 1 hit 0xbs 1011 1010 1011 010 1 hit 0xba 1011 1010 1011 101 0 miss	_	toug index b	l l				
0x2b 0010 1011 0010 101 1 miss 0x02 0000 0010 0000 001 0 hit 0xbf 1011 111 1011 111 1 miss 0xs8 0101 1000 0101 100 0 miss 0xbe 1011 1110 1011 111 0 hit 0x0e 00001110 0000 111 0 miss 0xbf 1011 0101 101 010 1 hit 0x2c 0010 1100 0010 110 0 miss 0xba 1011 1010 1011 101 0 miss	0x2b 0010 1011 0010 101 1 miss 0x02 0000 0010 0000 001 0 hit 0xbf 1011 1111 1011 111 1 miss 0xs8 0101 1000 0101 100 0 miss 0xbe 1011 1110 1011 111 0 hit 0x0e 00001110 0000 111 0 miss 0xbf 1010101 1011 010 1 hit 0x2c 00101100 0010 110 0 miss 0xba 1011 1010 1011 101 0 miss	0x2b 0010 1011 0010 101 1 miss 0x02 0000 0010 0000 001 0 hit 0xbf 1011 1111 1011 111 1 miss 0x58 0101 1000 0101 100 0 miss 0xbe 1011 1110 1011 111 0 hit 0x0e 00001110 0000 111 0 miss 0xbs 1011 0101 1011 010 1 hit 0xbc 0010 1100 0010 110 0 miss 0xba 1011 1010 1011 101 0 miss	hex add		l lock offict		offret	hit/miss	
0x02 0000 0010 0000 001 0 hit 0xbf 1011 111 1011 111 1 miss 0x58 0101 1000 0101 100 0 miss 0xbe 1011 1110 1011 111 0 hit 0x0e 00001110 0000 111 0 miss 0xbt 1011 0101 1011 010 1 hit 0xbc 0010 1100 0010 110 0 miss 0xba 1011 1010 1011 101 0 miss	0x02 0000 0010 0000 001 0 hit 0xbf 1011 1111 1011 111 1 miss 0x58 0101 1000 0101 100 0 miss 0xbe 1011 110 1011 111 0 hit 0x0e 00001110 0000 111 0 miss 0xbt 1011 010 101 010 1 hit 0xbc 0010 1100 0010 110 0 miss 0xba 1011 1010 1011 101 0 miss	0x02 0000 0010 0000 001 0 hit 0xbf 1011 1111 1011 111 1 miss 0xs8 0101 1000 0101 100 0 miss 0xbe 1011 110 1011 111 0 hit 0x0e 0000 1110 0000 111 0 miss 0xbs 1011 010 101 010 1 hit 0xba 1011 1010 1011 101 0 miss 0xba 1011 1010 1011 101 0 miss		'r binary addr	l lock offict tag	index	offiet 1		
0xbf 1011 1111 1011 111 1 miss 0x58 0101 1000 0101 100 0 miss 0xbe 1011 1110 1011 111 0 hit 0xbe 00001110 0000 111 0 miss 0xbf 1011 0101 1011 010 1 hit 0x2c 0010 1100 0010 110 0 miss 0xba 1011 1010 1011 101 0 miss	0xbf 1011 111 1011 111 1 miss 0x58 0101 1000 0101 100 0 miss 0xbe 1011 1110 1011 111 0 hit 0xbe 00001110 0000 111 0 miss 0xbs 1011 0101 1011 010 1 hit 0x2c 0010 1100 0010 110 0 miss 0xba 1011 1010 1011 101 0 miss	0xbf 1011 111 1011 111 1 miss 0x58 0101 1000 0101 100 0 miss 0xbe 1011 110 101 111 0 hit 0xbe 00001110 0000 111 0 miss 0xbs 1011 0101 101 010 1 hit 0x2c 0010 1100 0010 110 0 miss 0xba 1011 1010 1011 101 0 miss	0x03	'r binary addr 00000011	lock offet tag 0000	index 001	,	miss	
0x58 01011000 0101 100 0 miss 0xbe 1011110 1011 111 0 hit 0x0e 00001110 0000 111 0 miss 0xbs 10110101 1011 010 1 hit 0x2c 00101100 0010 110 0 miss 0xba 10111010 1011 101 0 miss	0x58 01011000 0101 100 0 miss 0xbe 1011110 1011 111 0 hit 0x0e 00001110 0000 111 0 miss 0xbs 10110101 101 010 1 hit 0x2c 00101100 0010 110 0 miss 0xba 10111010 1011 101 0 miss	0x58 01011000 0101 100 0 miss 0xbe 1011110 101 111 0 hit 0x0e 00001110 0000 111 0 miss 0xbs 10110101 101 010 1 hit 0x2c 00101100 0010 110 0 miss 0xba 1011010 1011 101 0 miss	0 x b4	r binary addr 00000011 (0110100	tag 0000 (01)	index ooi oio	0	miss miss	
0xbe 1011110 1011 111 0 hit 0x0e 00001110 0000 111 0 miss 0xbt 1011010 101 010 1 hit 0x2c 00101100 0010 110 0 miss 0xba 10111010 1011 101 0 miss	0xbe 1011110 101 111 0 hit 0x0e 00001110 0000 111 0 miss 0xbt 1010101 101 010 1 hit 0x2c 00101100 0010 110 0 miss 0xba 10111010 1011 101 0 miss	0xbe 1011110 101 111 0 hit 0x0e 00001110 0000 111 0 miss 0xbt 1011010 101 010 1 hit 0x2c 00101100 0010 110 0 miss 0xba 10111010 1011 101 0 miss	0×03 0×b4 0×2b	r binary addr 00000011 1011 0100 0010 1011	tag 0000 (01)	index 001 010 101	0	miss miss miss	
0x0e 00001110 0000 111 0 miss 0xbs 1011010 0010 110 0 miss 0x2c 00101100 0010 110 0 miss 0xba 10111010 1011 101 0 miss	0x0e 00001110 0000 111 0 miss 0xbs 10110101 101 010 1 hit 0x2c 00101100 0010 110 0 miss 0xba 10111010 1011 101 0 miss	0x0e 00001110 0000 111 0 miss 0xbs 10110101 101 010 1 hit 0x2c 00101100 0010 110 0 miss 0xba 10111010 1011 101 0 miss	0x03 0xb4 0x2b 0x02	r binary addr 00000011 1011 0100 0010 1011 0000 0010	tag 0000 (01) 0010	index 001 010 101	0 1	miss miss miss hit	
0xb5 10110101 1011 010 1 hit 0x2c 00101100 0010 110 0 miss 0xba 10111010 1011 101 0 miss	0xb5 10110101 1011 010 1 hit 0x2c 00101100 0010 110 0 miss 0xba 10111010 1011 101 0 miss	0xb5 10110101 1011 010 1 hit 0x2c 00101100 0010 110 0 miss 0xba 1011 101 0 miss	0x03 0xb4 0x2b 0x02 0xbf	r binary addr 00000011 (011 0100 0010 1011 0000 0010	tag 0000 (01) 0010 0000	index 001 010 101 001	0 1	miss miss miss hit miss	
0x2c 00101100 0010 110 0 miss 0xba 10111010 1011 101 0 miss	0x2c 00101100 0010 110 0 miss 0xba 10111010 1011 101 0 miss	0x2c 00101100 0010 110 0 miss 0xba 10111010 1011 101 0 miss	0x03 0xb4 0x2b 0x02 0xbf 0x58	r binary addr 0000 0011 1011 0100 0010 1011 0000 0010 1011 1111	tag 0000 (01) 0010 0000 1011	index 001 010 101 001 111	0 1 0	miss miss miss hit miss miss	
0xba 1011 1010 1011 101 0 miss	0xba 1011 1010 1011 101 0 miss	0xba 1011 1010 1011 101 0 miss	0x03 0xb4 0x2b 0x02 0xbf 0x58 0xbe	r binary addr 00000011 (011 0100 0010 1011 0000 0010 1011 1111 0101 1000 1011 1110	tag 0000 (01) 0000 1011 0101 1011	index 001 010 101 001 111 100	1 0 1 0 1 0 0	miss miss miss hit miss miss hit	
			0x03 0xb4 0x2b 0x02 0xbf 0x58 0xbe 0x0e	r binary addr 00000011 (011 0100 0010 1011 0000 0010 1011 1111 0101 1000 1011 1110	tag 0000 1011 0000 1011 0101 0000	index 001 010 101 001 111 100		miss miss miss hit miss miss miss miss miss	
0xfd 1111101 1111 110 1 miss	0xfol 11111101 1111 110 1 miss	0xfol 11111101 1111 110 1 miss	0x03 0xb4 0x2b 0x02 0xbf 0x58 0xbe 0x0e 0xb5	r binary addr 00000011 1011 0100 0010 1011 0000 0010 1011 1111 0101 1000 1011 1110 00001110 1011 0101	tag 0000 (01) 0000 1011 0101 1011 0000 1011	index 001 010 101 001 111 100 111		miss miss miss hit miss miss hits hit	
			0x03 0xb4 0x2b 0x02 0xbf 0x58 0xbe 0x0e 0xbt 0x2c	r binary addr 00000011 1011 0100 0010 1011 0000 0010 1011 1111 0101 1000 1011 1110 00001110 1011 0101	tag 0000 (01) 0000 1011 0101 1011 0000 1011	index 001 010 101 001 111 100 111 111		miss miss miss hit miss miss miss miss hit miss hit miss	

5.2.3 C, has the highest miss nate with 1-word block size. Because every word address is different from each other, one-word block size loses spatial locality badly. consider Cz and Cz Cz block size = 2 words -> block offiet = 1 bit C3 block size = 4 words → offset = 2 bit block num = 8/2=4 \rightarrow index=2bit block num = $8/4 = 2 \rightarrow index = 1 bit$ tag = 8-3=5bit tag=5bit Cz 03 hex addr binary addr hit/miss tag offset offet hit/miss index index 0X03 00000011 00000 miss 01 miss 1 0 11 10110100 0 x b4 10110 miss miss 10 0 00 1 0010 1011 OXZb 00101 01 miss miss 1 11 0000 0010 OXOZ 00000 miss 01 0 miss 0 10 1011 1111 oxbf miss 10111 miss 11 11 0 X 5 8 01011000 01011 miss miss 00 0 0 00 10111110 0xbe LOIL 11 hit 0 hit 10 ox oe 00001110 miss miss 00001 11 0 10 0xb5 10110101 miss 10110 hit 10 01 OXZC 0010 1100 miss 00101 10 miss 0 00 1011 1010 oxba miss 10111 01 0 miss 10 oxfd 11111101 10 mu01 miss miss Cz hit rate > Cz hit rate : Cz is the optimized design for the given references.

cache data size = $32KB = 32 \times 2^{10} \times 2^3 bit$ block size = 2 words = 2 x 8 bytes → block offset = 4 bit block num = $\frac{32\times2^{10}\times2^{3}bit}{2\times64bit} = 2^{11} \rightarrow index = 11bit$ tag = 64-11-4 = 49 bit valid = 1 bit : total size = (valid + tag + block size) x block num = (1+49+128) x 2" bit = 364544 bit cache data size = 64KB = 219bit 5.3.2 block size = 16 words = 24x23bytes > block offset = 7 bit block num = $\frac{2^{19}bit}{2^{10}bit} = 2^{9}bit \rightarrow index = 9bit$ tag = 64-7-9 = 48 bit valid = 1 bit total size = (valid+tag+block size) x block num = (1+48+210) x 29 bit = 549376 bit 5.3.3 1 block size is larger, which means a larger multiplexer is needed to select a specified byte from one block. @ number of blocks is smaller, which means the replacement of blocks may be more frequent, increasing miss rate. 5.3.4 In 5.3.1, the cache is direct-mapped, cache data size = 32KB, block size = 2 words data Une valid tag 128 } 2046 lines in 5.3.4, the cache is 2-way set associated, cache data size = 32KB. block size = 2 words block num = 2'', index = $\frac{2''}{2}$ = 2^{10} bit tag = 64 - 4 - 10 = 50 bit

5.3

5.3.1

data line data line valid tag valid tag 1 50 128 128 1024 lines Read requests series example: always keep the index the same, but change tag index=0 tag = 0. address: tag=1, index=0 index=0 tag=0, tag=1, index=0 we will see the direct-mapped cache always misses but the 2-way set associated cache always hit (except first two) thus the I-way set associated cache has a lower miss rate assume I word = 64 bit 5.5 cache block size = 25 bit = 0.5 word 5.5.1 block number = $2^5 = 32$ 5.5.2 cache data size = block size x block number = $2^{5}x2^{5} = 2^{10}bit$ 5.5.3 total coche size = (valid + tag + block size) x block number = (1+54+32) x 25 bit $ratio = \frac{87 \times 2^5 \text{ bit}}{32 \times 2^5 \text{ bit}} = 2.72$

```
5.5.4
                                                                         replaced bytes
 Hex Addr
                                                           hitlmiss
              Binary Addr
                                     index
                                               offset
                             tag
   00
              00000000
                                     00000
                                               00000
   04
               00000100
                                     00000
                                               00100
                                                            h
   10
              00010000
                                     00000
                                                            h
                                               10000
   84
              1000 0100
                                    00100
                                               00100
                                                            m
   E8
              1110 1000
                             0
                                    00111
                                              01000
                                                            M
   A0
             1010 0000
                             0
                                    00101
                                               00000
                                                           m
  400
            0000 0000 0000
                                    00000
                                              00000
                                                           m
                                                                 MemtxO] - MemtxIFJ
   1E
             00011110
                                    00000
                                              11110
                                                                 Mem[x400]-Mem[x41F]
  80
                                              01100
                             0
            10001100
                                    กกเอด
                                              11100
                                                                Mem[XO] - Mem[XIF]
  CIC
          1100 0001 1100
                                   00000
                                                          m
  B4
                            0
                                             10100
                                                          h
            10110100
                                   00101
                                                                Mem[X80] - Mem[X9F]
  884
                           10
                                             00100
          1000100001000
                                   00100
                                                         m
 \zeta.S.\xi hit ratio = \frac{4}{12} = 33.3%
 5.5.6 < 0, 3, Mem[xC00] - Mem[xCIF]>
        <4, Z, Mem[X880] - Mem[x89F] >
         < S, O, Mem[xAO] - Mem[xBf] >
         <7,0, Mem[xEO]-Mem[xFF]>
3.6
        between L1 and L2: need a small buffer, such as a few words
 5.6.1
        between Lz and memory: need a bigger buffer, at least one block.
5.6.2
       handling an LI write-miss:
        if the target block is in L2,
            write around the data in L2
       else if Lz is full, write back a dirty block to memory
            read the target block from memory to L2
            write the data in L2
```

handling an LI write-miss: 5.6.3 if the target block is in L2. write around the data in L2 else if Lz is full, write back a dirty block to memory read the target block from memory to L2 write the data in L2 handling an LI read-miss: if the target block is in L2 if L1 is full, use replace strategy to replace one block read block to LI from LZ else if 12 is full, write back one dirty block to memory read block to LZ from memory if L1 is full, use replace strategy to replace one block read block to LI from LZ