Pledge: *I pledge my honor that I have abided by the Stevens Honor System.* - **Eric Altenburg**

5.1: Page 497

(5.1.1)

2

(5.1.2)

I, J, B[I][0]

(5.1.3)

A[I][J]

5.2: Page 498

(5.2.1)

Word Address	Binary Address	Tag	Index	Hit/Miss
0x03	0000 0011	0	3	M
0xb4	1011 0100	b	4	M
0x2b	0010 1011	2	b	M
0x02	0000 0010	0	2	M
0xbf	1011 1111	b	f	M
0x58	0101 1000	5	8	M
0xbe	1011 1110	b	e	M
0x0e	0000 1110	0	e	M
0xb5	1011 0101	b	5	M
0x2c	0010 1100	2	c	M
0xba	1011 1010	b	a	M
0xfd	1111 1101	f	d	M

(5.2.2)

Word Address	Binary Address	Tag	Index	Offset	Hit/Miss
0x03	0000 0011	0	1	1	M
0xb4	1011 0100	b	2	0	M
0x2b	0010 1011	2	5	1	M
0x02	0000 0010	0	1	0	H
0xbf	1011 1111	b	7	1	M
0x58	0101 1000	5	4	0	M
0xbe	1011 1110	b	6	0	H
0x0e	0000 1110	0	7	0	M
0xb5	1011 0101	b	2	1	H
0x2c	0010 1100	2	6	0	M
0xba	1011 1010	b	5	0	M
0xfd	1111 1101	f	6	1	M

(5.2.3)

Just wanted to preface this by saying that if there are any discrepancies between the Word Address, Binary Address, and Tag across the tables, then it is simply a mistake. All three tables should have the same numbers/values for the aforementioned columns.

Cache 1				
Word Address	Binary Address	Tag	Index	Hit/Miss
0x03	0000 0011	0x00	3	M
0xb4	1011 0100	0x16	4	M
0x2b	0010 1011	0x05	3	M
0x02	0000 0010	0x00	2	M
0xbf	1011 1111	0x17	7	M
0x58	0101 1000	0x0b	0	M
0xbe	1011 1110	0x17	6	M
0x0e	0000 1110	0x01	6	M
0xb5	1011 0101	0x16	5	M
0x2c	0010 1100	0x05	4	M
0xba	1011 1010	0x17	2	M
0xfd	1111 1101	0x1f	5	M

Cache 1 Miss Rate = **100%**

Cache 1 Total Cycles = **324**

Cache 2				
Word Address	Binary Address	Tag	Index	Hit/Miss
0x03	0000 0011	0x00	1	M
0xb4	1011 0100	0x16	2	M
0x2b	0010 1011	0x05	1	M
0x02	0000 0010	0x00	1	M
0xbf	1011 1111	0x17	3	M
0x58	0101 1000	0x0b	0	M
0xbe	1011 1110	0x17	3	H
0x0e	0000 1110	0x01	3	M
0xb5	1011 0101	0x16	2	H
0x2c	0010 1100	0x05	2	M
0xba	1011 1010	0x17	1	M
0xfd	1111 1101	0x1f	2	M

Cache 2 Miss Rate = **83%**

Cache 2 Total Cycles = **286**

This is the cache that has the best performance.

Cache 3				
Word Address	Binary Address	Tag	Index	Hit/Miss
0x03	0000 0011	0x00	0	M
0xb4	1011 0100	0x16	1	M
0x2b	0010 1011	0x05	0	M
0x02	0000 0010	0x00	0	M
0xbf	1011 1111	0x17	1	M
0x58	0101 1000	0x0b	0	M
0xbe	1011 1110	0x17	1	H
0x0e	0000 1110	0x01	1	M
0xb5	1011 0101	0x16	1	M
0x2c	0010 1100	0x05	1	M
0xba	1011 1010	0x17	0	M
0xfd	1111 1101	0x1F	1	M

Cache 3 Miss Rate = **92%**

Cache 3 Total Cycles = **335**

As mentioned above, cache 2 has the best performance.

5.5: Page 499

(5.5.1)

All of the cache blocks have 4 8-byte words (assuming each word is 8-bytes, but if they were to be 4-bytes per word, then there would be 8 4-byte words).

Offset total = 5 bits with 3 of the 5 being the word offset and the last 2 being for the block offset. The 2 bits enumerate $2^2 = 4$ words.

(5.5.2)

With 5 index bits, you can have $2^5 = 32$ lines in the cache.

(5.5.3)

$32 * 4 * 8 = 8192$ bits.

$8192 + (54 * 32) + (1 * 32) = 9952$ bits.

$\frac{9952}{8192} = \mathbf{1.21}$

(5.5.4)

Byte Address	Binary Address	Tag	Index	Offset	Hit/Miss	Bytes Replaced
0x00	0000 0000 0000	0x0	0x00	0x00	M	
0x04	0000 0000 0100	0x0	0x00	0x04	H	
0x10	0000 0001 0000	0x0	0x00	0x10	H	
0x84	0000 1000 0100	0x0	0x04	0x04	M	
0xe8	0000 1110 1000	0x0	0x07	0x08	M	
0xa0	0000 1010 0000	0x0	0x05	0x00	M	
0x400	0100 0000 0000	0x1	0x00	0x00	M	0x00-0x1F
0x1e	0000 0001 1110	0x0	0x00	0x1e	M	0x400-0x41F
0x8c	0000 1000 1100	0x0	0x04	0x0c	H	
0xc1c	1100 0001 1100	0x3	0x00	0x1c	M	0x00-0x1F
0xb4	0000 1011 0100	0x0	0x05	0x14	H	
0x884	1000 1000 0100	0x2	0x04	0x04	M	0x80-0x9f

(5.5.5)

$\frac{4}{12} = \mathbf{33\%}$

(5.5.6)

<0, 3, Mem[0xC00] - Mem[0xC1F]>
<4, 2, Mem[0x880] - Mem[0x89f]>
<5, 0, Mem[0x0A0] - Mem[0x0Bf]>
<7, 0, Mem[0x0e0] - Mem[0x0ff]>

5.9: Page 501

(5.9.1)

AMAT for B = 8: $0.040 * (20 * 8) = \mathbf{6.4}$

AMAT for B = 16: $0.030 * (20 * 16) = \mathbf{9.6}$

AMAT for B = 32: $0.020 * (20 * 32) = \mathbf{12.8}$

AMAT for B = 64: $0.015 * (20 * 64) = \mathbf{19.2}$

AMAT for B = 128: $0.010 * (20 * 128) = \mathbf{25.6}$

A block size of B = 8 is optimal.

(5.9.2)

AMAT for B = 8: $0.040 * (24 + 8) = \mathbf{1.28}$

AMAT for B = 16: $0.030 * (24 + 16) = \mathbf{1.20}$

AMAT for B = 32: $0.020 * (24 + 32) = \mathbf{1.12}$

AMAT for B = 64: $0.015 * (24 + 64) = \mathbf{1.32}$

AMAT for B = 128: $0.010 * (24 + 128) = \mathbf{1.52}$

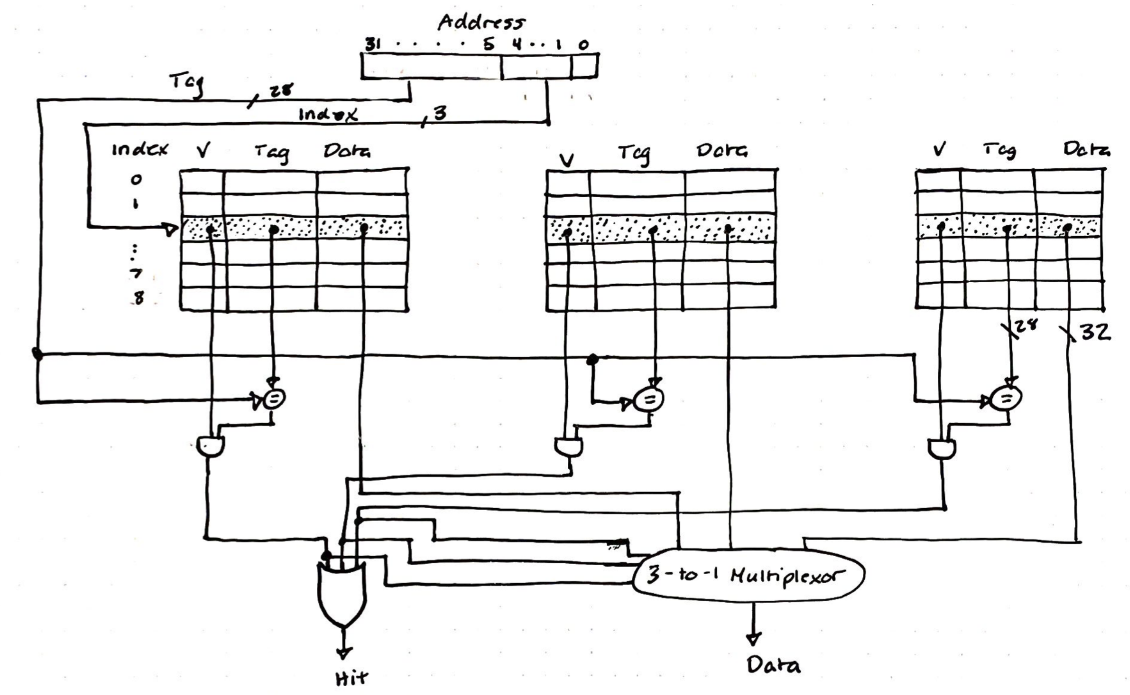
A block size of B = 32 is optimal.

(5.9.3)

A block size of B = 128 is optimal because when you minimize the miss rate, the total miss latency is also reduced.

5.11: Page 502

(5.11.1)



Since each block can hold 2 words, the word offset bits = $log_2 2 = 1$ bit

Number of cache slots = $\frac{cache\ size}{block\ size} = \frac{48}{2} = 24$ slots

Number of sets = $\frac{24}{3} = 8$

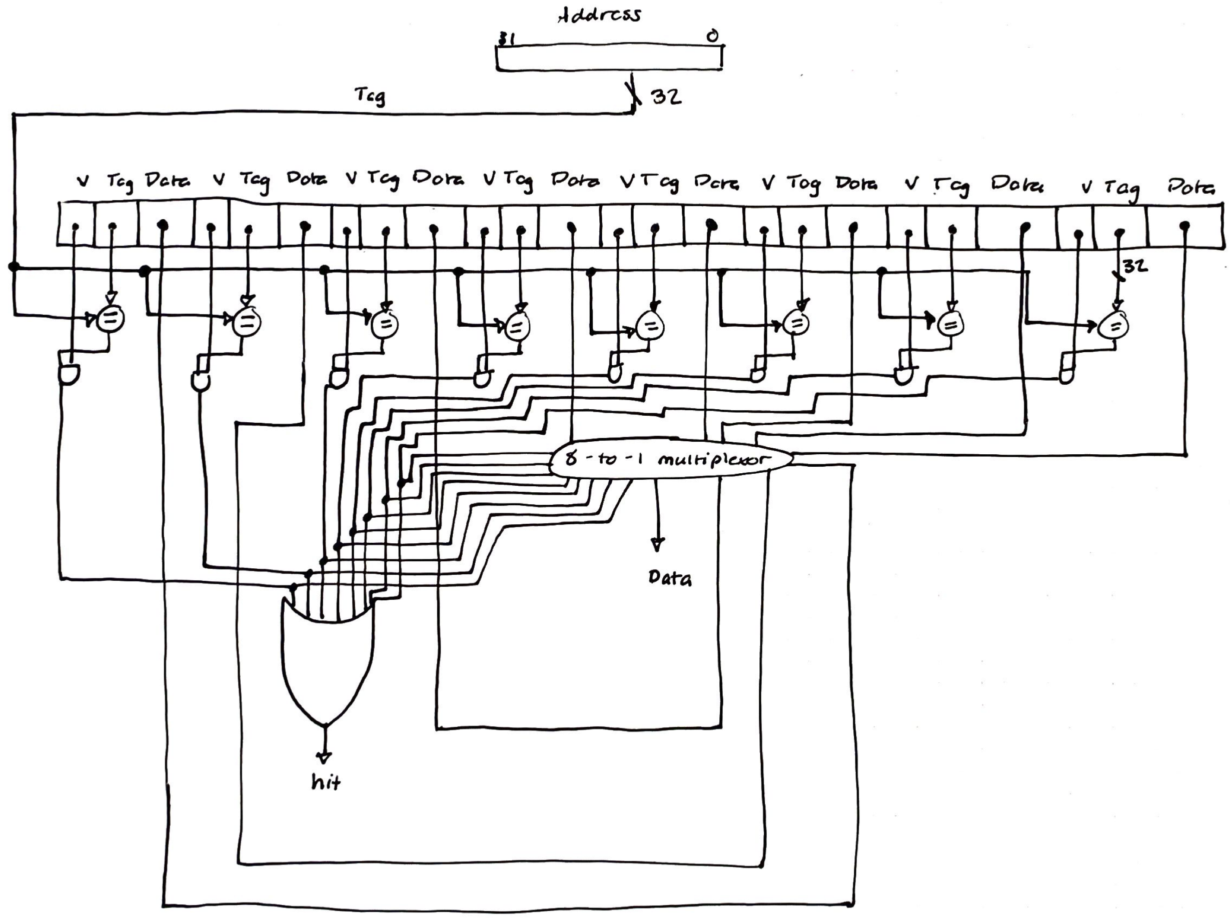
Set Offset Bits = $log_2 8 = 3$ bits

$32 - 3 - 1$= 28 bits for the tag

(5.11.2)

Word Address	Binary Address	Tag	Index	Offset	Hit/Miss	Way 0	Way 1	Way 2
0x03	0000 0011	0x0	1	1	M	T(1)=0		
0xb4	1011 0100	0xb	2	0	M	T(1)=0 T(2)=b		
0x2b	0010 1011	0x2	5	1	M	T(1)=0 T(2)=b T(5)=2		
0x02	0000 0010	0x0	1	0	H	T(1)=0 T(2)=b T(5)=2		
0xbe	1011 1110	0xb	7	0	M	T(1)=0 T(2)=b T(5)=2 T(7)=b		
0x58	0101 1000	0x5	4	0	M	T(1)=0 T(2)=b T(5)=2 T(7)=b T(4)=5		
0xbf	1011 1111	0xb	7	1	H	T(1)=0 T(2)=b T(5)=2 T(7)=b T(4)=5		
0x0e	0000 1110	0x0	7	0	M	T(1)=0 T(2)=b T(5)=2 T(7)=b T(4)=5	T(7)=0	
0x1f	0001 1111	0x1	7	1	M	T(1)=0 T(2)=b T(5)=2 T(7)=b T(4)=5	T(7)=0	T(7)=1
0xb5	1011 0101	0xb	2	1	H	T(1)=0 T(2)=b T(5)=2 T(7)=b T(4)=5	T(7)=0	T(7)=1
0xbf	1011 1111	0xb	7	1	H	T(1)=0 T(2)=b T(5)=2 T(7)=b T(4)=5	T(7)=0	T(7)=1
0xba	1011 1010	0xb	5	0	M	T(1)=0 T(2)=b T(5)=2 T(7)=b T(4)=5	T(7)=2 T(5)=b	T(7)=1
0x2e	0010 1110	0x2	7	0	M	T(1)=0 T(2)=b T(5)=2 T(7)=b T(4)=5	T(7)=2 T(5)=b	T(7)=1
0xce	1100 1110	0xc	7	0	M	T(1)=0 T(2)=b T(5)=2 T(7)=b T(4)=5	T(7)=2 T(5)=b	T(7)=c

(5.11.3)


Address
31
0
Tag
32
V Tag Data V Tag Data V Tag Data V Tag Data V Tag Data V Tag Data V Tag Data V Tag Data
32
=
8-to-1 multiplexor
Data
hit

Since each block can hold 1 word, the word offset bits = $log_2 1 = 0$ bits

Number of cache slots = $\frac{cache\ size}{block\ size} = \frac{8}{1} = 8$ slots

32 – 0= 32 bits for the tag

5.13: Page 504

(5.13.1)

For a MTTF of 3 years and a MTTR of 1 day, the MTBF is **1096 days or 26304 hours**.

(5.13.2)

$\frac{1095}{1096}$ = **99.90875%**.

(5.13.3)

As the MTTR approaches 0, the **availability will approach 1**. Since drives are becoming more and more inexpensive, the possibility of having a MTTR of 0 for hardware is becoming more likely, but replacing the data would be very time consuming and this is often overlooked.

(5.13.4)

Availability would be a high number if MTTR were to vastly increase as it plays a significant role in calculations, but if MTTF were to grow considerably, then the availability would be high as well. It all depends on the ratio of the values of MTTR to MTTF as if MTTF is a large number in comparison to MTTR, MTTR's values will not be meaningful.