Homework 5 Solution

1. [Memory Hierarchy]
   a) 64 bit integer = 8 byte, so a16-byte block can store 2 integers.
   b) i, j, and B[i][0].
   c) A[i,j]

2. [Memory Hierarchy]
   a)

Word addr	Binary addr	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

   b)

Word addr	Binary addr	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	7	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

   c)

Word addr	Binary addr	Tag	Cache 1		Cache 2		Cache 3	
			Index	Hit/ Miss	Index	Hit/ Miss	Index	Hit/ Miss
0x03	0000 0011	0x00	3	M	1	M	0	M
0xB4	1011 0100	0x16	4	M	2	M	1	M
0x2B	0010 1011	0x05	3	M	1	M	0	M
0x02	0000 0010	0x00	2	M	1	M	0	M
0xBF	1011 1111	0x17	7	M	3	M	1	M
0x58	0101 1000	0x0B	0	M	0	M	0	M
0xBE	1011 1110	0x17	6	M	3	H	1	H
0x0E	0000 1110	0x01	6	M	3	M	1	M
0xB5	1011 0101	0x16	5	M	2	H	1	M
0x2C	0010 1100	0x05	4	M	2	M	1	M
0xBA	1011 1010	0x17	2	M	1	M	0	M
0xFD	1111 1101	0x1F	5	M	2	M	1	M

So Cache 2 has the best performance.

3. [Memory Hierarchy]

a) The offset is 5 bits. The three least significant bits are byte offset (the offset into each byte of the 8-byte word). The remaining two bits (4-3) are the word offset. Two bits allow us to enumerate $2^2 = 4$ words. So, the block size is 4 8-byte words.

b) There are 5 index bits. Therefore, there are $2^5 = 32$ blocks (lines) in the cache.

c) The cache stores a total of 32 blocks * 4 words/block * 8 bytes/word = 8192 bits of data.
In addition to data, each line contains 54 tag bits and 1 valid bit. Thus, the total bits required is 8192 + 54*32 + 1*32 = 9952.
The ratio is therefore 9952/8192 = 1.21.
(Note that you may also neglect the valid bits, but you must explicitly say so.)

d)

Byte addr	Binary addr	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

e) Hit ratio = 4 hits / 12 accesses = 33%

f) <index, tag, data>:
<0, 3, Mem[0xC00]-Mem[0xC1F]>
<4, 2, Mem[0x880]-Mem[0x89F]>
<5, 0, Mem[0x0A0]-Mem[0x0BF]>
<7, 0, Mem[0x0E0]-Mem[0x0FF]>

4. [Memory Hierarchy]

a) P1: 1/0.66ns = 1.515 GHz
P2: 1/0.90ns = 1.11 GHz

b) P1: All memory accesses require 1 cycle to access L1. In addition, 8% of memory accesses require an additional 70ns to access main memory. 70/0.66=106.06 cycles, which we must round up to the nearest cycle, so 107 cycles. Thus, the Average Memory Access Time is 1 + 0.08*107 = 9.56 cycles.

P2: All memory accesses require 1 cycle to access L1. In addition, 6% of memory accesses require an additional 70ns to access main memory. 70/0.90=77.78 cycles, which we must round up to the nearest cycle, so 78 cycles. Thus, the Average Memory Access Time is 1 + 0.06*78 = 5.68 cycles.

c) For P1, every instruction requires at least one cycle. In addition, 8% of all instructions miss in the instruction cache and incur a 107-cycle delay. Furthermore, 36% of the instructions are data accesses. 8% of these 36% are cache misses, which adds an additional 107 cycles.
1 + 0.08*107 + 0.36*0.08*107 = 12.64
With a clock cycle of 0.66ns, each instruction requires 12.64*0.66 = 8.34 ns.

Using the same logic for P2, we get a CPI of 7.36 and an average of only 6.63 ns per instruction. Therefore, P2 is faster.

5. [Memory Hierarchy]

a) Each line/set has a total of 6 words (two in each of three ways). There will be a total of 48/6 = 8 lines/sets.
The offset is $\log_2 2$ = 1 bit
The index is $\log_2 8$ = 3 bits
The tag is 64 – 3 – 1 = 60 bits
The data is 6 blocks * 8 sets = 48 words

b) T(x) is the tag at index x:

Word addr	Binary addr	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(4)=5 T(5)=2 T(7)=B		
0xBF	1011 1111	0xB	7	1	H	T(1)=0 T(2)=B T(4)=5 T(5)=2 T(7)=B		
0x0E	0000 1110	0x0	7	0	M	T(1)=0 T(2)=B T(4)=5 T(5)=2 T(7)=B	T(7)=0	
0x1F	0001 1111	0x1	7	1	M	T(1)=0 T(2)=B T(4)=5 T(5)=2 T(7)=B	T(7)=0	T(7)=1
0xB5	1011 0101	0xB	2	1	H	T(1)=0 T(2)=B T(4)=5 T(5)=2 T(7)=B	T(7)=0	T(7)=1
0xBF	1011 1111	0xB	7	1	H	T(1)=0 T(2)=B	T(7)=0	T(7)=1

						T(4)=5 T(5)=2 T(7)=B		
0xBA	1011 1010	0xB	5	0	M	T(1)=0 T(2)=B T(4)=5 T(5)=2 T(7)=B	T(5)=B T(7)=0	T(7)=1
0x2E	0010 1110	0x2	7	0	M	T(1)=0 T(2)=B T(4)=5 T(5)=2 T(7)=B	T(5)=B T(7)=2	T(7)=1
0xCE	1100 1110	0xC	7	0	M	T(1)=0 T(2)=B T(4)=5 T(5)=2 T(7)=B	T(5)=B T(7)=2	T(7)=C

c)

Word addr	Binary addr	Tag	Hit/Miss	Contents
0x03	0000 0011	0x03	M	3
0xB4	1011 0100	0xB4	M	3, B4
0x2B	0010 1011	0x2B	M	3, B4, 2B
0x02	0000 0010	0x02	M	3, B4, 2B, 2
0xBE	1011 1110	0xBE	M	3, B4, 2B, 2, BE
0x58	0101 1000	0x58	M	3, B4, 2B, 2, BE, 58
0xBF	1011 1111	0xBF	M	3, B4, 2B, 2, BE, 58, BF
0x0E	0000 1110	0x0E	M	3, B4, 2B, 2, BE, 58, BF, E
0x1F	0001 1111	0x1F	M	B4, 2B, 2, BE, 58, BF, E, 1F
0xB5	1011 0101	0xB5	M	2B, 2, BE, 58, BF, E, 1F, B5
0xBF	1011 1111	0xBF	H	2B, 2, BE, 58, E, 1F, B5, BF
0xBA	1011 1010	0xBA	M	2, BE, 58, E, 1F, B5, BF, BA
0x2E	0010 1110	0x2E	M	BE, 58, E, 1F, B5, BF, BA, 2E
0xCE	1100 1110	0xCE	M	58, E, 1F, B5, BF, BA, 2E, CE

6. [Virtual Memory]
The page size is 8KB, and 8K = 2^{13}, so the page offset is 13 bits.
The number of virtual pages is 32 – 13 = 19 bits.
All five page tables would require a combined size of 5 * (2^{19} * 4) bytes = 10MB.

7. [Virtual Memory]
   4KB pages, 4K=2^{12}, so the 12 least significant bits of the virtual address is the page offset, and the rest are the virtual page number:

Address	Virtual Page	Hit/Miss	TLB Content after access			
			Valid	Tag	Physical Page Num	Time Since Last Access
0x123D	1	TLB miss	1	0xB	12	5
		Page Fault	1	0x7	4	2
			1	0x3	6	4
			1	0x1	13	0
0x08B3	0	TLB miss	1	0x0	5	0
		PT hit	1	0x7	4	3
			1	0x3	6	5
			1	0x1	13	1
0x365C	3	TLB hit	1	0x0	5	1
			1	0x7	4	4
			1	0x3	6	0
			1	0x1	13	2
0x871B	8	TLB miss	1	0x0	5	2
		Page Fault	1	0x8	14	0
			1	0x3	6	1
			1	0x1	13	3
0xBEE6	B	TLB miss	1	0x0	5	3
		PT hit	1	0x8	14	1
			1	0x3	6	2
			1	0xB	12	0
0x3140	3	TLB hit	1	0x0	5	4
			1	0x8	14	2
			1	0x3	6	0
			1	0xB	12	1
0xC049	C	TLB miss	1	0xC	15	0
		Page Fault	1	0x8	14	3
			1	0x3	6	1
			1	0xB	12	2