ECE485/585 Homework No. 3 Solution

Problem No. 1

(a) The 7-bit data packet (P1 through P7) could be arranged as follows with the 3 check bits (R1, R2 and R3) placed at the 2^N positions and the data bits (D1, D2, D3 and D4) placed in the remaining positions:

	P1	P2	P3	P4	P5	P6	P7
Bit Pos_2	001	010	011	100	101	110	111
Bit	R1	R2	D1	R3	D2	D3	D4
Rcv'd bits	1	0	1	0	0	0	1

Data word = 1001
ECC check bits = 100

(b) To determine whether the data word was received correctly or not, we need to re-compute the check bits based on the received data. The calculations are as follows:

$R1 = P_{001} = D1 \wedge D2 \wedge D4 = 1 \wedge 0 \wedge 1 = 0$
$R2 = P_{010} = D1 \wedge D3 \wedge D4 = 1 \wedge 0 \wedge 1 = 0$
$R3 = P_{100} = D2 \wedge D3 \wedge D4 = 0 \wedge 0 \wedge 1 = 1$

Since the calculated check bits (001) do not match the received check bits (100), we can conclude that the data word was <u>not received</u> correctly.

(c) To determine which bit is incorrect, we take the XOR of the received check bits and the re-computed check bits. In this case, we calculate 100 ^ 001 = 101 which indicates that P5 (bit#5 in the received packet) is incorrect. Referring to the table above, P5 refers to D2. We can correct the error by inverting D2 (from 0 to 1). Therefore, the correct data word is {D1, D2, D3, D4} = <u>1101</u>.

We can confirm this by re-computing the check bits with what we feel is the correct data (1101):

$R1 = P_{001} = D1 \wedge D2 \wedge D4 = 1 \wedge 1 \wedge 1 = 1$
$R2 = P_{010} = D1 \wedge D3 \wedge D4 = 1 \wedge 0 \wedge 1 = 0$
$R3 = P_{100} = D2 \wedge D3 \wedge D4 = 1 \wedge 0 \wedge 1 = 0$

This matches the received check bits.

Problem No. 2

(a) Loop-A is expected to have more instruction cache hits. This is because loop-A has more iterations and thus the same instructions keep on getting accessed over and over again. Furthermore, loop-A has fewer instructions in each iteration, causing the instruction footprint to easily fit in a small cache.

(b) Increasing the cache block size results in more data being transferred from the main memory to the cache on every cache miss.

Advantage: This can result in a higher cache hit ratio, if the application exhibits high spatial locality.

Disadvantages: (i) This can increase the cache miss latency and put more bandwidth pressure on the next level cache (or memory), (ii) It can result in fragmentation and wastage of capacity, if the program does not exhibit high spatial locality.

Problem No. 3

(a) Block size = 64 bytes = 2^6 bytes
Therefore, Number of bits in the *Byte Select* field = 6
Cache size = 16K-byte = 2^{14} bytes
Total number of cache blocks = Cache size / Block size = 2^{14} / 2^6 = 2^8 = 256
Number of cache blocks per set = 1 (Direct-mapped cache)
Number of sets = 2^8 / 1 = 2^8
Therefore, Number of bits in the *Index* field = 8
Total number of address bits = 32
Therefore, Number of bits in the *Tag* field = 32 - 6 - 8 = 18

(b) Each cache block requires the following SRAM bits:
18 bits for tag, 1 valid bit, 1 dirty bit and 512 data bits
Total SRAM bits in the cache = Total number of blocks * (SRAM bits needed for each block)
= 256 * (18 + 1 + 1 + 512) = 136,192 bits

Problem No. 4

(a) Block size = 128 bytes = 2^7 bytes
Therefore:
Number of bits in the *Byte Select* field = 7
Cache size = 8Mbyte = 2^{23} bytes
Total number of cache blocks = Cache size / Block size = 2^{23} / 2^7 = 2^{16} = 65536
Number of cache blocks per set = 16 = 2^4
Number of sets = 2^{16} / 2^4 = 2^{12}
Therefore, Number of bits in the *Index* field = 12
Total number of address bits = 32
Therefore, Number of bits in the *Tag* field = 32 - 7 - 12 = 13

(b) Each cache block requires the following SRAM bits:
13 bits for tag, 1 valid bit, 1 dirty bit and 1024 data bits
Therefore:
Number of SRAM bits needed for each block = 13 + 1 + 1 + 1024 = 1039
Total SRAM bits in the cache = Total number of blocks * (SRAM bits needed for each block)
= 65,536 * 1039 = 68,091,904 bits

Problem No. 5

Block size = 16 bytes = 2^4 bytes
Therefore, Number of bits in the *Byte Select* field = 4
Total number of cache blocks = 8
Number of cache blocks per set = 2 (2-way set-associative cache)
Number of sets = 8 / 2 = 4 = 2^2
Therefore, Number of bits in the *Index* field = 2
Total number of address bits = 12
Therefore, Number of bits in the *Tag* field = 12 - 4 - 2 = 6

CPU Operation	Tag	Index	Byte Select	H/M	Miss Type	Memory Operation
Read 0x014 000000010100	00	1	4	M	Compulsory	Read 16 bytes from 0x010 Set-1, way-0 marked valid, tag = 00
Read 0x020 000000100000	00	2	0	M	Compulsory	Read 16 bytes from 0x020 Set-2, way-0 marked valid, tag = 00
Write 0x200 001000000000	08	0	0	M	Compulsory	Read 16 bytes from 0x200 Set-0, way-0 marked valid & dirty, tag = 08
Write 0x0A0 000010100000	02	2	0	M	Compulsory	Read 16 bytes from 0x0A0 Set-2, way-1 marked valid & dirty, tag = 02
Read 0x024 000000100100	00	2	4	H		
Read 0x380 001110000000	0E	0	0	M	Compulsory	Read 16 bytes from 0x380 Set-0, way-1 marked valid, tag = 0E
Write 0x388 001110000000	0E	0	0	H		No memory writes since this is a writeback cache, Set-0, way-1 marked dirty
Read 0x100 000100000000	04	0	0	M	Compulsory	Read 16 bytes from 0x100 Evict the LRU line in set-0 (address 0x200 in way-0). Write the evicted line back to memory. New tag for set-0, way-0 = 04
Read 0x0A8 000010101000	02	2	8	H		
Write 0x200 001000000000	08	0	0	M	Conflict	Read 16 bytes from 0x200 Evict the LRU line in set-0 (address 0x380 in way-1). Writeback the evicted line to memory

Problem No. 6

(a) L1 cache miss ratio = 1 – L1 cache hit ratio = 1 – 95% = 5%

(b) L1 cache miss rate = L1 cache access rate * L1 cache miss ratio = 400 * 5% = 20 MPKI

(c) L2 cache miss rate = L2 cache access rate * L2 cache miss ratio
Every time, there is an L1 cache miss, it results in an L2 cache access.
Therefore, L2 cache access rate = 20 Accesses PKI
Hence, L2 cache miss rate = 20 * 30% = 6 MPKI

(d) AMAT of L2 cache = (L2 hit time * L2 hit ratio) + (L2 miss time * L2 miss ratio)
= (15 * (1 – 30%)) + ((15 + 200) * 30%)
= 75 cycles

(e) AMAT of L1 cache = (L1 hit time * L1 hit ratio) + (L1 miss time * L1 miss ratio)
L1 hit time = 3 cycles
L1 hit ratio = 95%
L1 miss ratio = 5%
L1 miss time = 3 + AMAT of L2 cache = 3 + 75 = 78 cycles
Therefore:
AMAT of L1 cache = (3 * 95%) + (78 * 5%) = 6.75 cycles