HW3 Solutions

1. (P5.31, but trace in reverse order) A victim cache is used to augment a direct-mapped cache to reduce conflict misses. For additional background on this problem, read Jouppi's paper on victim caches [Jouppi, 1990]. Please fill in the following table to reflect the state of each cache line in a 4-entry direct-mapped cache and a 2-entry fully associative victim cache following each memory reference shown. Also, record whether the reference was a cache hit or a cache miss. The reference addresses are shown in hexadecimal format. Assume the direct-mapped cache is indexed with the low-order bits above the 16-byte line offset (e.g. address 40 maps to set 0, address 50 maps to set 1, etc.). Use '-' to indicate an invalid line and the address of the line to indicate a valid line. Assume LRU policy for the victim cache and mark the LRU line as such in the table.

Reference Address	Hit/Miss	Direct-Mapped Cache				Victim Cache	
		Line 0	Line 1	Line 2	Line 3	Line 0	Line 1
[init state]		-	110	-	FF0	1F0	210/LRU
200	M	200					
80	M			80			
200	H	200					
E0	M			E0		1F0/LRU	80
B0	M				B0	FF0	80/LRU
80	H-victim	80		E0		FF0/LRU	200
200	H-victim	200				FF0/LRU	80
A0	M			A0		E0	80/LRU
80	H-victim			80		E0/LRU	A0

2. (P3.28)Assume a synchronous front-side processor-memory bus that operates at 100MHz and has an 8-byte data bus. Arbitration for the bus takes one bus cycle (10ns), issuing a cache line read command for 64 bytes of data takes one cycle, memory controller latency (including DRAM access) is 60ns, after which data doublewords are returned in back-to-back cycles. Further assume the bus is blocking or circuit-switched. Compute the latency to fill a single 64 byte cache line. Then compute the peak read bandwidth for this processor-memory bus, assuming the processor arbitrates for the bus for a new read in the bus cycle following completion of the last read.

   Fill latency = arb + cmd + mem + #transfers = 1 + 1 + 60/10 + (64/8) = 16 bus cycles or 160ns

   Peak read bandwidth = 64B/160ns = 0.4B/ns = 400 million bytes/sec.

3. (P3.31) Consider finite DRAM bandwidth at a memory controller, as follows. Assume double-data-rate DRAM operating at 100 MHz in a parallel non-interleaved organization, with an 8 byte interface to the DRAM chips. Further assume that each cache line read results in a DRAM row miss, requiring a precharge and RAS cycle, followed by row-hit CAS cycles for each of the doublewords in the cache line. Assuming memory controller overhead of one cycle (10ns) to initiate a read operation, and one cycle latency to transfer data from the DRAM data bus to the processor-memory bus, compute the latency for reading one 64 byte cache block. Now compute the peak data bandwidth for the memory interface, ignoring DRAM refresh cycles.

Memory latency is measured from when the memory controller sees the command to when it places the last doubleword on the processor bus:

Latency = precharge + RAS + overhead + 64Bx(1 xfer/8B)x(1 cycle)/(2 xfer)) = 7 cycles = 70ns

Peak data bandwidth = 64B/70ns = 0.914B/ns = 914 million bytes/sec

4. Simulation problems may have variance in results due to simulation setup issues.

   Part a: bar chart with C/C/C components derived from appropriate simulations (very large cache for cold misses, fully-associative or nearly-FA cache for capacity).

   Part b: Comparison of LRU and OPT for capacity and conflict misses

   Part c: Compare multilevel simulation vs. flat simulation in sim-cheetah.

   Part d: Compare cache in OOO context vs. in-order trace-like simulation