2.1

a. Each element is 8B. Because a 64B cacheline has 8 elements, and each column access will result in fetching a new line for the nonideal matrix, we need a minimum of 8×8 (64 elements) for each matrix. Hence, the minimum cache size is $128 \times 8\text{B} = 1$ KB.

b. The blocked version only has to fetch each input and output element once. The unblocked version will have one cache miss for every $64\text{B}/8\text{B} = 8$ row elements. Each column requires $64\text{B} \times 256$ of storage, or 16 KB. Thus, column elements will be replaced in the cache before they can be used again. Hence, the unblocked version will have 9 misses (1 row and 8 columns) for every 2 in the blocked version.

c.
```
for (i = 0; i < 256; i = i + B) {
    for (j = 0; j < 256; j = j + B) {
        for (m = 0; m < B; m++) {
            for (n = 0; n < B; n++) {
                output[j + n][i + m] = input[i + m][j + n];
            }
        }
    }
}
```

d. 2-way set associative. In a direct-mapped cache, the blocks could be allocated so that they map to overlapping regions in the cache.

e. You should be able to determine the level-1 cache size by varying the block size. The ratio of the blocked and unblocked program speeds for arrays that do not fit in the cache in comparison to blocks that do is a function of the cache block size, whether the machine has out-of-order issue, and the bandwidth provided by the level-2 cache. You may have discrepancies if your machine has a write-through level-1 cache and the write buffer becomes a limiter of performance.

2.2

Because the unblocked version is too large to fit in the cache, processing eight 8B elements requires fetching one 64B row cache block and 8 column cache blocks. Because each iteration requires 2 cycles without misses, prefetches can be initiated every 2 cycles, and the number of prefetches per iteration is more than one, the memory system will be completely saturated with prefetches. Because the latency of a prefetch is 16 cycles, and one will start every 2 cycles, $16/2 = 8$ will be outstanding at a time.

2.18

a. The average memory access time of the current (4-way 64 KB) cache is 1.69 ns. 64 KB direct mapped cache access time = 0.86 ns @ 0.5 ns cycle time = 2 cycles Way-predicted cache has cycle time and access time similar to direct mapped cache and miss rate similar to 4-way cache.

   The AMAT of the way-predicted cache has three components: miss, hit with way prediction correct, and hit with way prediction mispredict: $0.0033 \times (20) + (0.80 \times 2 + (1 - 0.80) \times 3) \times (1 - 0.0033) = 2.26$ cycles = 1.13 ns.

b. The cycle time of the 64 KB 4-way cache is 0.83 ns, while the 64 KB direct-mapped cache can be accessed in 0.5 ns. This provides 0.83/0.5 = 1.66 or 66% faster cache access.

c. With 1 cycle way misprediction penalty, AMAT is 1.13 ns (as per part a), but with a 15 cycle misprediction penalty, the AMAT becomes: $0.0033 \times 20 + (0.80 \times 2 + (1 - 0.80) \times 15) \times (1 - 0.0033) = 4.65$ cycles or 2.3 ns.

d. The serial access is 2.4 ns/1.59 ns = 1.509 or 51% slower.

2.19

a. The access time is 1.12 ns, while the cycle time is 0.51 ns, which could be potentially pipelined as finely as 1.12/0.51 = 2.2 pipestages.

b. The pipelined design (not including latch area and power) has an area of 1.19 mm^2 and energy per access of 0.16 nJ. The banked cache has an area of 1.36 mm^2 and energy per access of 0.13 nJ. The banked design uses slightly more area because it has more sense amps and other circuitry to support the two banks, while the pipelined design burns slightly more power because the memory arrays that are active are larger than in the banked case.

2.20

a. With critical word first, the miss service would require 120 cycles. Without critical word first, it would require 120 cycles for the first 16B and 16 cycles for each of the next 3 16B blocks, or $120 + (3 \times 16) = 168$ cycles.

b. It depends on the contribution to Average Memory Access Time (AMAT) of the level-1 and level-2 cache misses and the percent reduction in miss service times provided by critical word first and early restart. If the percentage reduction in miss service times provided by critical word first and early restart is roughly the same for both level-1 and level-2 miss service, then if level-1 misses contribute more to AMAT, critical word first would likely be more important for level-1 misses.

2.22

In all three cases, the time to look up the L1 cache will be the same. What differs is the time spent servicing L1 misses. In case (a), that time = 100 (L1 misses) × 16 cycles + 10 (L2 misses) × 200 cycles = 3600 cycles. In case (b), that time = 100 × 4 + 50 × 16 + 10 × 200 = 3200 cycles. In case (c), that time = 100 × 2 + 80 × 8 + 40 × 16 + 10 × 200 = 3480 cycles. The best design is case (b) with a 3-level cache. Going to a 2-level cache can result in many long L2 accesses (1600 cycles looking up L2). Going to a 4-level cache can result in many futile look-ups in each level of the hierarchy.

2.36

The cores will be executing 8 cores × 3 GHz/2.0CPI = 12 billion instructions per second. This will generate 12 × 0.00667 = 80 million level-2 misses per second. With the burst length of 8, this would be 80 × 32B = 2560 MB/s. If the memory bandwidth is sometimes 2X this, it would be 5120 MB/s. From Fig. 2.14, this is just barely within the bandwidth provided by DDR2-667 DIMMs, so just one memory channel would suffice.

Fig 2.5 DDR2-667 5336 MiB/S/DIMM

2.40

Hibernating will be useful when the static energy saved in DRAM is at least equal to the energy required to copy from DRAM to Flash and then back to DRAM. DRAM dynamic energy to read/write is negligible compared to Flash and can be ignored.

$$Time = \frac{8 \times 10^9 \times 2 \times 2.56 \times 10^{-6}}{64 \times 1.6} = 400\,seconds$$

The factor 2 in the above equation is because to hibernate and wakeup, both Flash and DRAM have to be read and written once.