CS 201 Section 1 Homework 9 – Suggested Solutions

1. (2.11). (a) and (b) are reasonable. (c) is impossible because there aren't enough address bits to distinguish the cells. (d),(e),(f) are unreasonable, because they waste address bits.

2. (2.18) A cylinder can be read in 6 rotations. One track is read during each of the first five rotations; during the sixth rotation, a seek is done to the next cylinder (which takes half a rotation), and then the system must wait for sector 0 to come around again. Reading 800 cylinders therefore takes 4800 rotations; at 20 msec/rotation, this requires 96 seconds. (The worst-case seek time is irrelevant.)

4. (2.19) With 4096 bits/sector and 64 sectors/track, each track holds 262,144 bits. At 7200 RPM, each rotation takes 1/120 sec. So peak data rate is 31,457,280 bits/sec = 3.9 MBytes/sec. Of course, this would mean repeatedly reading/writing the same data bits 120 times/sec, so it is not too meaningful a number!

5. (3.22) The design uses two AND gates for chip enable logic plus two AND gates per word select line plus one AND gate per data bit. For a 256×8 memory this comes to $2 + 256 \times 2 + 2048 = 2562$ AND gates. The circuit also uses one OR gate for each bit in the word; hence eight of them would be needed.

6.(4.26) The mean access time is 0.8×5 nsec $+ 0.15 \times 20$ nsec $+ 0.05 \times 80$ nsec $= 11$ nsec.

7. (4.29) A conditional branch takes 4 cycles, while all other instructions take just 1. The average number of cycles per instruction is therefore $0.8 \times 1 + 0.2 \times 4 = 1.6$. In the absence of stalls, the machine would run at 1 cycle/instruction, so the stalling due to conditionals makes the machine 60% slower; i.e., it runs a $1/1.6 = 5/8$ of full speed.

8. (6.2) $2^{32}/(8 \times 1024) = 2^{19}$ pages.

9. (6.3) (a) These addresses cause faults: 2048 to 4095, 5120 to 6143, 7168 to 8191.

(b)

Virtual Address	Physical Address
0	3072
3728	fault
1023	4095
1024	1024
1025	1025
7800	fault
4096	2048

10. (6.4)

(a) With LRU, memory looks like this after each reference:

Reference	Fault?	Memory (in LRU to MRU order)			
0	Yes	-	-	-	0
7	Yes	-	-	0	7
2	Yes	-	0	7	2
7	No	-	0	2	7
5	Yes	0	2	7	5
8	Yes	2	7	5	8
9	Yes	7	5	8	9
2	Yes	5	8	9	2
4	Yes	8	9	2	4

(b) With FIFO, memory look looks like this after each reference:

Reference	Fault?	Memory (in FI order)			
0	Yes	-	-	-	0
7	Yes	-	-	0	7
2	Yes	-	0	7	2
7	No	-	0	7	2
5	Yes	0	7	2	5
8	Yes	7	2	5	8
9	Yes	2	5	8	9
2	No	2	5	8	9
4	Yes	5	8	9	4