CS 201 Section 1 Homework 6 Suggested Solutions

1. (4.1) In the figure, A and B are ORed together, but with B not enabled. One different way would be to add them instead of ORing them; to do this, just set F_0 to 1.

2. (4.2) This question is best understood by looking at Figure 4-3. The data path takes $1+1+3+1=6$ nsec; since the clock pulse is 2 nsec, the total clock cycle time (e.g., falling edge to falling edge) must be at least $6+2=8$ nsec. With a 100 MHz clock, the cycle time is 10 nsec, so the machine can run fine. With a 150 MHz clock, the cycle time is 6.67 nsec, so the machine cannot work.

3. (3.7) This problem is just a variant on the majority function example in Figure 3-12(b). As noted in class, the idea is to compute the parity of the *control* inputs A,B, and C (rather than of the data inputs). To do this, we simply wire inputs D_j for which the representation of j has odd parity (i.e., D_1, D_2, D_4, D_7) to V_{CC} and wire the other D inputs to ground.

4. (3.9) It is a circuit with one data input I, two control inputs A and B, and four outputs $O_{\overline{AB}}, O_{\overline{A}B}, O_{A\overline{B}}, O_{AB}$ selected by the combination of inputs indicated in their subscript.

It's too tedious to get the graphic representation of the circuit into this solution set, but its equations are simply:

$$\begin{aligned} O_{\overline{AB}} &= I\overline{AB} \\ O_{\overline{A}B} &= I\overline{A}B \\ O_{A\overline{B}} &= IA\overline{B} \\ O_{AB} &= IAB \end{aligned}$$

Note that this is just like a 2-bit decoder, except that the input is ANDed into each output.

5. (3.12) It computes $C = \overline{(\overline{A\overline{AB}})(\overline{B\overline{AB}})} = (A+B)(\overline{AB}) = A \oplus B$ and $D = \overline{\overline{AB}} = AB$. In other words, it is a half-adder, with C as the sum and D as the carry.

6. (3.15) The timing of the circuit can be found by writing a 0 on each of the input lines and then tracing them through the circuit, adding 10 at each gate. The time at the output of a gate is 10 more than the *maximum* of the times of the inputs. The A input takes 20 nsec to become available to the logic unit, as do the decode lines, so the output of the logic unit to the final output gate takes 40 nsec. The adder also must wait 20 nsec to get A; it then produces its contribution to the final output gate after 50 nsec. Worst case through the whole circuit is therefore 60 nsec.

7. (3.16) Yes, it can already subtract. In 2's complement, $B-A = B+(-A)$ and $-A = \overline{A}+1$. Thus to subtract A from B, we need to add B to $\overline{A}$ and then increment the sum. The circuit can already do this by asserting INVA and INC and performing an ADD.

8. (3.18) We need 16 adds at 10 nsec each plus 15 propagates at 1 nsec each, for a total of 175 nsec.