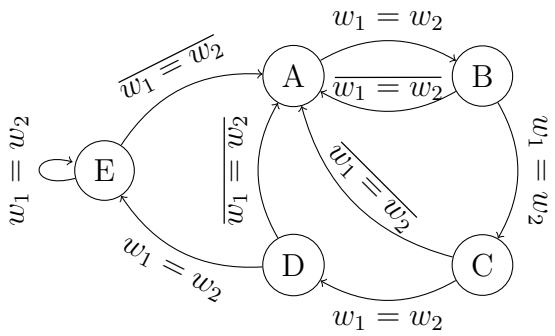
ECEN 2350 Spring 2017 Homework 8
Samuel Cuthbertson

---

1. *Book Problems: 6.9, 6.11*

   6.9: *A sequential circuit has two inputs, w_1 and w_2, and an output z. Its function is to compare the input sequences on the two inputs. If $w_1 = w_2$ during any four consecutive clock cycles, the circuit produces $z = 1$; otherwise, $z = 0$. Derive a suitable circuit.*

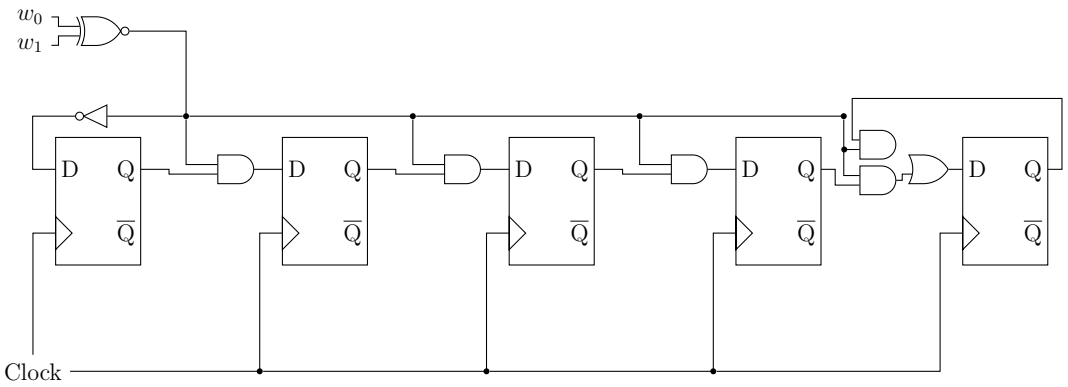
   As we are looking to detect *any* four consecutive clock cycles, we will only need five states, as shown below in the graph and state diagram. Encoding states as $A = 00001$, $B = 00010$, $C = 00100$, $D = 01000$, and $E = 10000$, as well as using $w_1 \oplus w_2$ will simplify the below table.



S	$w_1 \oplus w_2$	N	z
00001	0	00010	0
00010	0	00100	0
00100	0	01000	0
01000	0	10000	1
10000	0	10000	1
x	1	00001	0

   Using the above state table, we can see that each digit i of N is equal to 1 if and only if $S_{i-1} = 1$ and $w_0 \oplus w_1 = 0$. The exceptions here would be N_0, which equals $w_0 \oplus w_1$ and N_4 which would equal $S_3\overline{(w_0 \oplus w_1)} + S_4\overline{(w_0 \oplus w_1)}$. All other digits would follow the form $N_i = S_{i-1}\overline{(w_0 \oplus w_1)}$.

   Finally, it's fairly obvious in the above scheme that $z = N_4$, as $N_4 = 1$ only when the next state is E, exactly the same as z. The circuit derived from the above is as follows:



6.11: *A given FSM has an input, w, and an output, z. During four consecutive clock pulses, a sequence of four values of the w signal is applied. Derive a state table for the FSM that produces $z = 1$ when it detects that either the sequence $w : 0010$ or $w : 1110$ has been applied; otherwise, $z = 0$. After the fourth clock pulse, the machince has to be again in the reset state, ready for the next sequence. Minimize the number of states needed.*

Using a very similar approach to above, we can use the state table shown below.

State	Next State		z
	$w = 0$	$w = 1$	
A	B	E	0
B	C	A	0
C	A	D	0
D	H	A	0
E	A	F	0
F	A	G	0
G	H	A	0
H	B	E	1

We can then use partitioning as follows:

$$P_1 = (ABCDEFGH)$$
$$P_2 = (ABCDEFG)(H)$$
$$P_3 = (A)(B)(C)(D)(E)(F)(G)(H)$$

And thus our minimum number of states is eight.

2. *Book Problem: B.2*

   a: *Show that the circuit in Figure PB.2 is fuctionally equivalent to the circuit in Figure PB.1.*

   Figure PB.1 is simple to examine, as it is just a cannonical sum-of-products implmenetation. It's formula is $f = \overline{x_1}\,\overline{x_2}x_3 + \overline{x_1}x_2\overline{x_3} + x_1\overline{x_2}\,\overline{x_3} + x_1x_2x_3$. Figure PB.2 is best to examine as a look up table, where x_2 is the selector in the first layer and x_1 is the selector in the second layer. Looking at it as such, we can say that when x_1x_2 is true, our circuit will return x_3, and when $x_1\overline{x_2}$ is true our circuit will return $\overline{x_3}$, etc. This all comes out to $f = \overline{x_1}\,\overline{x_2}x_3 + \overline{x_1}x_2\overline{x_3} + x_1\overline{x_2}\,\overline{x_3} + x_1x_2x_3$, which is the exact same as our other circuit.

   b: *How many transistors are needed to build the CMOS circuit in PB.2? Draw the CMOS circuit.*

   Using the transmission gates desribed on page 787 of the textbook, we can implement the multiplexer circuit as shown:

3. *Book Problem: B.7*

a: *Give the truth table for the CMOS circuit in the Figure PB.5.*

x_0	x_1	x_2	x_3	f
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

b: *Derive the simplest sum-of-products expression for the truth table in (a). How many transistors are needed to build the sum-of-products circuit using the CMOS AND, OR and NOT gates?*

x_0x_1 \ x_2x_3	00	01	11	10
00	1	0	0	0
01	1	0	0	0
11	0	0	0	0
10	1	0	0	0

And therefor the simplest SOP expression is $f = \overline{x_0}\overline{x_2}\overline{x_3} + x_0\overline{x_1}\overline{x_2}\overline{x_3}$. As one NOT gate takes 2 transitors, a two input OR gate takes 6 transistors, a four input AND gate takes 10 transistors, and a 3 input AND gate takes 8 transistors, we can sum up the above into:

$$f = 2 * 4 + 6 + 10 + 8 = \boxed{32}$$

4. *Given the following schematic from the Texas Instruments CMOS BCD to 7-segment decoder CD4511B, estimate the number of transistors this circuit needs. You cannot ignore the FlipFlops at the inputs to the circuit (the box that is labeled as a PTGN). These are toggle flip flops with preset/clear. Model these like the example from class.* Assuming the following:

   - Each NAND or NOR gate uses 2n transistors, where n is the number of inputs.
   - A NOT gate is 2 transistors.
   - Each flip flop is composed of an NOT, two ANDs, and an OR gate around a D-Flip Flop with Preset/Clear. Cost of $2 + (4 + 2) + (4 + 2) + (4 + 2) + DFF$
   - The cost of a D-Flip-Flop with Preset/Clear is six NAND gates with three inputs, or 36.

   With all of the the above assumptions, we can see that the circuit contains:

   - 22 NOT gates
   - 8 T-Flip-Flops
   - 6 4-Input NORs
   - 8 3-Input NORs
   - 12 2-Input NORs and NANDs
   - 4 Transitors in the "Driver Logic"

   All this sums to:

$$22 * 2 + 8 * 56 + 6 * 4 + 8 * 3 + 12 * 2 + 4 = \boxed{568}$$

   But this could all vary drastically depending on many factors, such as which implementation one choses for D-Flip-Flops. I picked the one on Page 262 of the textbook.