ELEN 21 HW #6

Tamir Enkhjargal
November 2018

Question #1:

a. If we had a small bank of memory with 64 addressable locations, how many bits would we need for the address?

If there are 64 unique addresses, then we only need a 6-bit number to be able to address all of the locations.

b. What is the value 128 expressed as 2 to the power of N?

$2^7 = 128$

c. What power of 2 evaluates to 8?

$2^3 = 8$

d. Given that ($2^N \cdot 2^M = 2^{N+M}$, what is $128 \cdot 8$ when expressed as a power of 2?

$2^7 \cdot 2^3 = 2^{10}$

e. What is that as a normal decimal number?

1024

Question #2:

We have a comparator that takes two inputs, A and B. Let's call these input ports, so we have port A and port B. And the comparator has two outputs. one that asserts if the value passed into A and B are equal, and another that asserts if the value passed into port B is greater than the value passed into port A. Now consider two arbitrary values, X and Y.

a. If we wanted to know if Y<X, which value would we send into which input port?

If B passes through, it is the larger value, then we need X input to the B port.

b. Looking at the two outputs, how would we know if Y<X?

If the second output asserts, then we know that X is greater than Y.

c. If we wanted to determine if Y≤X, how would we determine this using the two available outputs?

Basically, if any of the outputs are asserted (i.e. '1'), then we know that X is greater than or equal to Y. If the outputs are both 0, then we know that Y must be greater than X.

Question #3:

We went through the example of a counter that counts up. Now let's look at a counter that counts down.

a. Assuming a 3-bit counter, write down the sequence of values that the counter will go through from one clock cycle to the next.

A down counter is just the opposite of an up counter. It counts from 111 to 000. Q_0 still resets Q_1 when it hits a positive edge (e.g. positive-edged counter), and Q_2 resets when Q_1Q_0 is equal to 11. Therefore the sequence of values are 111, 101, 011, 001, 111.

b. For each bit position, identify the condition that must be present in order to change the value in that bit position.

Q_0 needs to hit a positive edge to reset/decrement Q_1. Q_2 resets/decrements when Q_1Q_0 is 11.

c. Assuming the use of D flip-flops, write the logic equations for the D inputs of each bit position: $D_2D_1D_0$

When the clock hits a positive edge, if $D = 0$, then $Q = 0$ (resets). If $D = 1$, then $Q = 1$ (sets). When the clock hits a negative edge, then there is no change in Q.

Question #4:

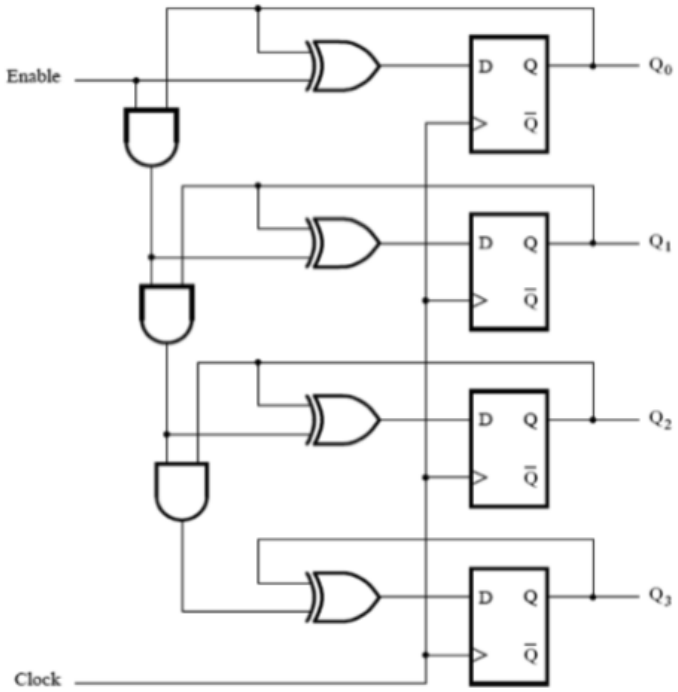

Figure 1: Question #4 and #5 Circuit

a. Identify the critical path: from which Q output does it start and at which D input does it end?

Q_0 to the D input of Q_3 is the critical path.

b. Clock-to-Q time is 0.5ns and setup time is 0.2ns. AND gates have a delay of 1.0ns, and XOR gates have a delay of 1.2ns. What is the smallest clock period that can be used to run this counter correctly?

$$\begin{aligned} Setuptime + 4 * Clock - to - Q + 3 * AND + 4 * XOR = \\ 0.2 + 4 * 0.5 + 3 * 1.0 + 4 * 1.2 = \\ 10.92ns = \end{aligned}$$

Question #5:

Say the rising edge of the clock takes 0.1ns to propagate from one flop to the next, in the order shown in the diagram. In other words, flop3 will see the rising edge first, then flop2 will see it 0.1ns later, flop1 will see it 0.1ns after flop2, etc. If we say that the clock rises at flop3 at time $t = 0$:

a. When does the clock rise at the source flop for the critical path above?

The clock will always hit a positive edge at a constant rate. The first flop will propagate when the clock its $t_1 + 0.1ns$

b. When does the clock rise at the destination flop?

Based off of just this rising edge skew, it will take an extra $0.4ns$ to propagate the final flop, so the clock will be at $t_7 + 0.4ns$

c. If we define the skew as the difference in time between the rising edge at the destination and the rising edge at the source, what is the skew that should be applied for this critical path?

If we wanted to have the flop change values exactly when the clock hits a positive edge, then we need a negative skew of $-0.4ns$.

d. And therefore, what is the minimum cycle time that can be used, based on this analysis of the skew?

At the final flop, we found that it will take at least $10.92ns$ delay with the difference logic gates and pass through values. If we add on $4 * 0.1ns$ then it will take $t_1 + 10.92 + 0.4 = t_1 + 11.32ns$