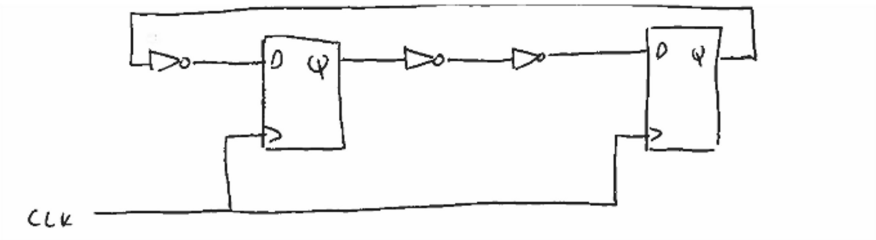
Assignment 2

March 3, 2023 2:56 PM

Alex Bartella
400308868

1. Consider the following circuit:



(a) Suppose that the propagation delay of an inverter is 1.0 ns. The flip-flops have $t_{clk-q} = 1.0$ ns and $t_{su} = 1.3$ ns. What is the maximum clock frequency?

$t_{pd} = 1ns$

$T_{clk} \geq 1.0ns + 2 \times 1.0ns + 1.3ns$
$\geq 3.3nS$

$f_{clk} \leq 1/T_{clk}$
$\leq 1/(4.3 \times 10^{-9} s)$
$\leq 232.56\ MHz$

(b) Suppose that for the flip-flops $t_h = 1.1$ ns. Explain why the circuit does not operate correctly.

The circuit does not operate correctly because the hold time is less than the total propagation delay (t_clk-q + t_pd from inverters)

1.1 < 1.0 + 2*1.0 = 3.0

This means that the input data to the flip flop is allowed to change before the output becomes valid (i.e. before the propagation delay time passes), which could cause a race condition for the output value, and the incorrect value could be outputted.

(c) How can clock skew between the two flip-flops remedy the situation in (b)? Give details.

t_clk-q + t_pd > t_h + **δ**
3.0 > 1.1 + **δ**
1.9 > **δ**

Issues occur when **δ** < 1.9, so if we can use a clock skew of 2, then the situation is remedied.

2. (a) Give a representation of the decimal number 4.30 in Q6.2. Is there any error in your representation? If so, what is its magnitude?

Q6.2

$b_5 2^5 + b_4 2^4 + b_3 2^3 + b_2 2^2 + b_1 2^1 + b_0 2^0 + b_{-1} 2^{-1} + b_{-2} 2^{-2}$

$00010001_2 = 4.25$ } we can't represent
$00010010_2 = 4.50$ } exactly 4.30 with this precision

4.25 is the closest we can get
magnitude of error is 0.05

(b) Write a Verilog module that takes two Q6.2 numbers and multiplies them together, with the output in Q6.2 format. The output of your module should have the result of the multiplication as well as a single bit flag that indicates if the result of the multiplication was an overflow.

```
module multiply(input [7:0] a, b, output reg [7:0] c, output reg overflow);
  reg [15:0] c_temp;
  always @(*) begin
    c_temp = (a*b);
    if (c_temp[15:10]>0 && c_temp[9:3]==7'b0)
      overflow = 1'b1;
    else
      overflow = 1'b0;
    c[7:0] = c_temp[9:2];
  end
endmodule
```

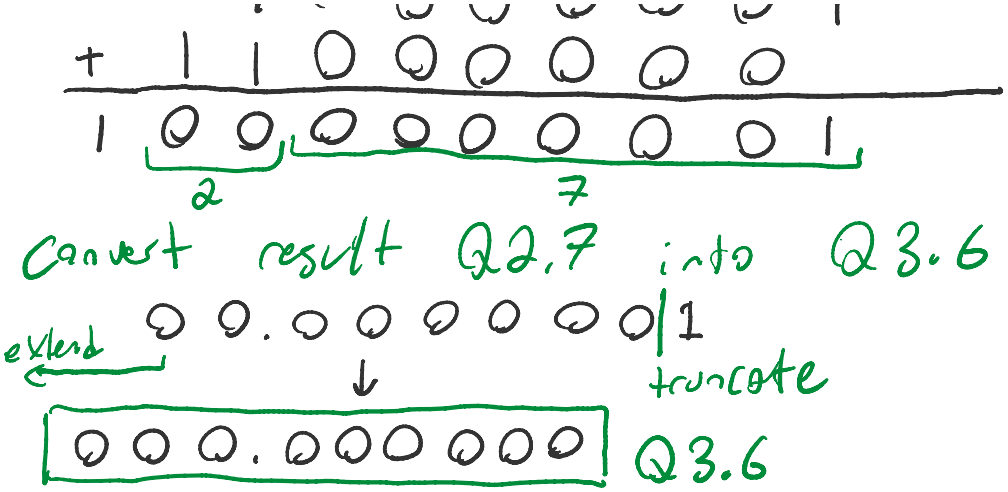
3. Suppose that we have two numbers, A in $Q1.7$ format and B in $Q2.6$ format. Suppose that the result of $A + B$ is assigned to C, which is in $Q3.6$ format.

(a) Give an example of A and B (each in binary representation) such that both A and B are both not zero, $A \neq -B$, but C is zero.

$$\begin{array}{r} b_0\ b_{-1}\ b_{-2}\ b_{-3}\ b_{-4}\ b_{-5}\ b_{-6}\ b_{-7} \\ +\ b_1\ b_0\ b_{-1}\ b_{-2}\ b_{-3}\ b_{-4}\ b_{-5}\ b_{-6} \\ \hline b_1\ b_0\ b_{-1}\ b_{-2}\ b_{-3}\ b_{-4}\ b_{-5}\ b_{-6} \end{array}$$

Q1.7
Q2.6
Q2.7

1	0	-1	-2	-3	-4	-5	-6	-7
	1	0	0	0	0	0	0	1
+	1	1	0	0	0	0	0	0



(b) Give a Verilog module that performs the assignment of $A + B$ to C.

```
module add(input signed [7:0] A, B, output reg signed [8:0] C);
    reg signed [8:0] a, b, c_temp;
    reg signed [9:0] c;
    always @(*) begin
        a[8] = A[7]; //preserve sign bits during left padding
        b[0] = 1'b0; //extend least significant bit (right padding)
        a[7:0] = A[7:0];
        b[8:1] = B[7:0];

        c = a + b; //calculate (result in Q2.7)

        c_temp[8:0] = c[8:0]; //remove most significant bit
        C[8:0] = c_temp[8:0] >>> 1; //shift right
    end

endmodule
```