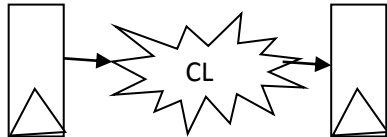


CS150 Discussion 2, Spring 2013

1. Build a FF with synchronous reset using simple DFF and gates. Write Verilog to describe a FF with asynchronous reset.
2. In the circuit below, assume we have clock-to-q delay of 1ns and setup time of 1ns, and the combinational logic has worst case delay of 8ns, and also assume we always run the clock at max frequency
 - a) What is the minimum clock period? If the first register clocks in a set of input at time 0, when would the result be produced? How many sets of input can it process every second?



- b) Now, if you insert a register in the middle of the combinational logic, what is the minimum clock period? And the latency? How many set of input can it process per second?
 - c) If we are not allowed to increase the latency, is there any way we can increase the clock frequency? What problem can it cause?
3. A sequential circuit is used to compare two inputs, in1 and in2. If in1 = in2 for three consecutive clock cycles, the circuit produces out = 1; otherwise out = 0 is produced. For instance

in1: 00010101

in2: 00110100

out: 00000110

- a) How many states does your FSM have? How many flip-flops does it use to implement the FSM?
 - b) Build your state machine using 2 input gates and FFs.
 - c) If clock-to-q delay is 1ns, delay for each gate is 0.8ns, setup time for FFs is 0.7ns, what is the highest clock frequency of your FSM?
4. Draw the circuit described by the Verilog code below. What would the circuit look like if you change your non-blocking assignment to blocking ones.

```
reg [2:0] m;
```

```
always@(posedge clk)
```

```
begin
```

```
    m[0]<= in;
```

```
    m[1]<=m[0];
```

```
    m[2]<=m[1];
```

```
end
```