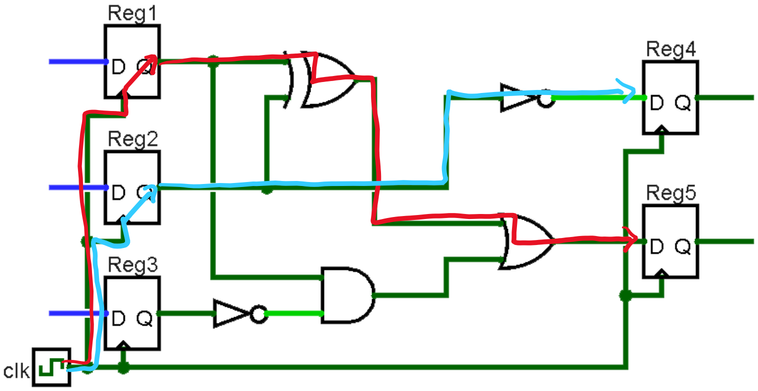
Design of Digital Circuits and Systems, Quiz 5

Timing: STA, Pipelining, CDC

Solution Outlines

Static Timing Analysis:



Longest data path and shortest data path are shown on the circuit diagram above. Note that the clock network delays are ordered $t_{clk1} > t_{clk2} > t_{clk3}$ and that $t_{XOR} > t_{NOT} + t_{AND}$. The longest path has Reg5 and the shortest path has Reg4 as the destination registers for the clock paths.

setup slack = min clock path – max data path

$= (T + t_{clk5,min} - t_{su,max}) - (t_{clk1,max} + t_{CO,max} + 3 \cdot t_{wire,max} + t_{XOR,max} + t_{OR,max})$

$= (160 + 3 - 15) - (7 + 16 + 3 \cdot 0 + 70 + 50)$ = **5 ns**

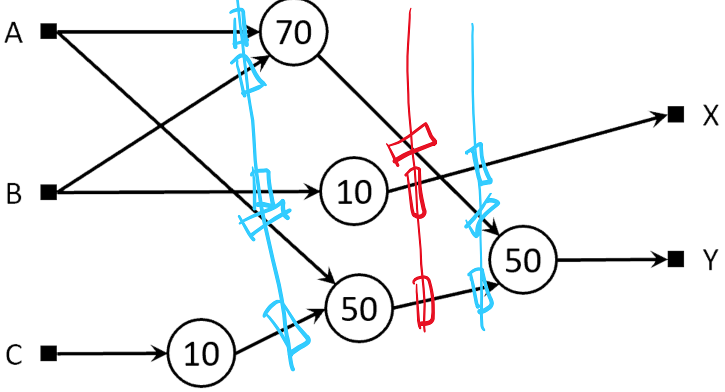
hold slack = min data path – max clock path

$= (t_{clk2,min} + t_{CO,min} + 2 \cdot t_{wire} + t_{NOT,min}) - (0 + t_{clk4,max} + t_{h,max})$

$= (2 + 12 + 2 \cdot 0 + 9) - (0 + 8 + 11)$ = **4 ns**

Pipelining:

The DFG and example cutsets (for **2-stage** and **3-stage** pipelines; multiple exist) might look like:



The key is to separate the OR delay from the XOR and AND delays; no stage should exceed the 70 delay of the XOR. The middle NOT delay can be placed in any stage with another node.

The 2-stage critical path gives us a T_{min} of $t_{CO} + t_{XOR}$ = $16 + 70$ = **86 ns**. Since it takes 2 clock cycles for the inputs to reach the output, latency = $2 \cdot T_{min} + t_{CO}$ = **188 ns**.

The 3-stage critical path is the same as the 2-stage one = **86 ns**. Since it takes 3 clock cycles for the inputs to reach the output, latency = $3 \cdot T_{min} + t_{CO}$ = **274 ns**.

Here, a 3-stage pipeline is unnecessary and increases our latency for no gain in throughput.

Clock Domain Crossing:

We have a *synchronous* clock domain crossing from a faster sending clock domain to a slower receiving clock domain.

For **data loss**, this would happen if a quick change in the sending domain (*e.g.*, 0 → 1 → 0) gets missed in the receiving domain (*e.g.*, read 0 the whole time) because it is sampling 3 times slower than the sending domain can generate changes. Mentions of a change at the input of Reg4 or Reg5 causing metastability that results in a bad/incorrect read were also accepted.

For **data incoherence**, this means that a *multi-bit signal* passed across a CDC is read in an invalid state (*i.e.*, the signals no longer accurately reflect the logic of the circuit/inputs). We mainly accepted two answers: (1) the two signals being sent across the clock domain crossing experience metastability differently (or at different times) and (2) the clk2 trigger falls in-between the times that the Reg4 and Reg5 inputs update due to the difference in path delays.