Critical path delay and maximal frequency evaluation

A note for this and all tests going forward, the results were obtained for inputs with a rise and fall time of 1 femtosecond which is below the 50ps limit stated in the outline.

Based off the setup of the ALU, it is determined that the critical path is the SUBTRACTION operation which goes from:
clock input of B reg → 4-bit AND → 4-bit XOR → ADDER → 4-bit-OR → input of output reg Y

We begin by measuring the delay of the logic between the 2 clocks and we note that it was measured for different voltages and with and without QRC extraction below.
Different input patterns were used to yield the maximal delay between the rising edge of the input and the rising edge of the output and the one pattern that yielded the longest delay was when we had all 1's at the output to yield 1111 i.e. A=0 and B=1 to yield A-B =-1.

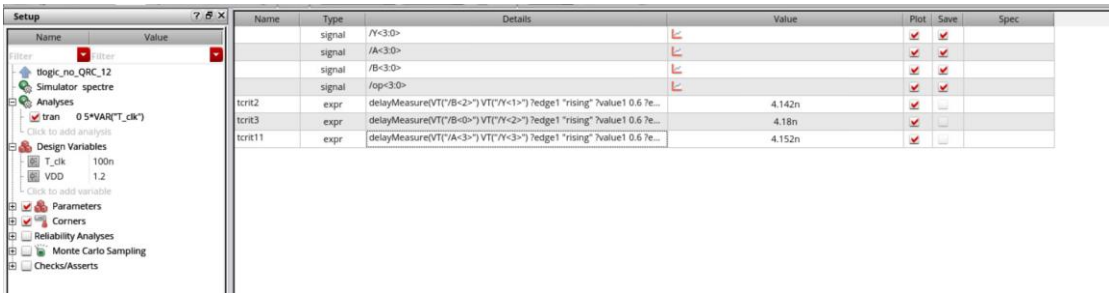

Figure 1: Tcritical without QRC for VDD = 1.2V

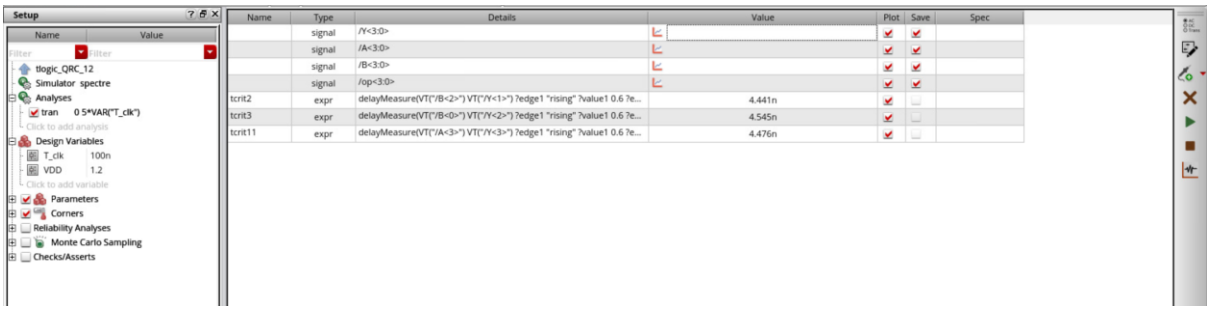

Figure 2: Tcritical with QRC for VDD = 1.2V

Name	Type	Details	Value	Plot	Save	Spec
	signal	/Y<3:0>		✓	✓	
	signal	/A<3:0>		✓	✓	
	signal	/B<3:0>		✓	✓	
	signal	/op<3:0>		✓	✓	
tcrit2	expr	delayMeasure(VT("/B<2>") VT("/...	19.6n	✓		
tcrit3	expr	delayMeasure(VT("/B<0>") VT("/...	19.86n	✓		
tcrit11	expr	delayMeasure(VT("/A<3>") VT("/...	19.79n	✓		

Figure 3: Tcritical without QRC for VDD=0.7V

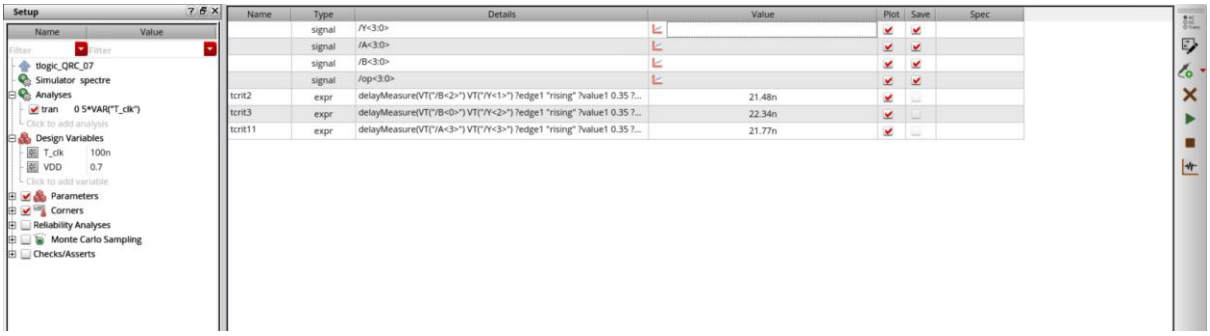

Figure 4: Tcritical with QRC for VDD=0.7V

The **delayMeasure** function proved useful as it enabled us to measure the delay that we have described above.

The above results are tabulated below.

	$V_{DD} = 1.2V$	$V_{DD} = 0.7V$
$t_{crit}\ (without\ QRC)$	4.18n	19.86n
$t_{crit}\ (with\ QRC)$	4.545n	22.34n

To calculate the maximal frequency or minimal period we use the rule: $T_{min} \geq t_{cq} + t_{logic} + t_{su}$

Where in our case t_{su} is the setup time, t_{cq} is the delay from the rising clock edge to when the output of the input register goes high and t_{logic} is the logic delay of the circuit. We've already found t_{logic} and all that's left to find is t_{su} and t_{cq}.

Some conclusions can be made from the above evaluations:

1) A decrease in VDD is detrimental to the delay of the circuit as covered in the course. The lower VDD is, the longer it takes for the circuit to respond hence lowering the overall maximal frequency that the circuit can achieve.

2) Including the QRC extraction increases the delay. This makes sense as often, the simulations that don't take into account the parasitic resistances and capacitances and are far too optimistic and once the parasitics are extracted we get a more pessimistic but also realistic view of the operation of the circuit.

t_{cq} evaluation

We begin by finding t_{cq} as this will later assist us in finding the setup time.
To get t_{cq} we simply set a testbench that triggers the outputs of the shift registers to go high and measure the delay from the rising clock edge before the output goes high to the rising edge at the output. Delay measure is once again useful in this regard and the results are shown below and subsequently tabulated.

Figure 5: tcq measurement at VDD = 1.2V with and without QRC

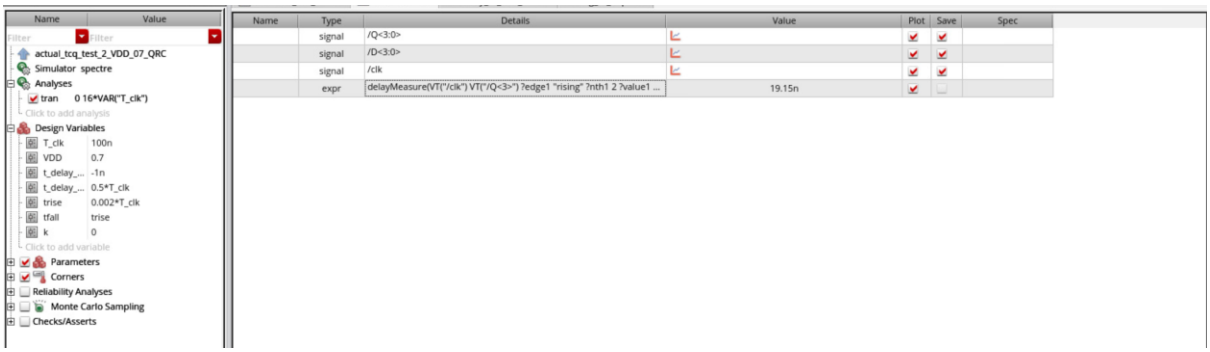

Figure 6: tcq measurement at VDD = 0.7V with and without QRC

	$V_{DD} = 1.2V$	$V_{DD} = 0.7V$
t_{cq} $(with\ and\ without\ QRC)$	4.066n	19.15n

Immediately from the above we can see that the delay is basically identical with and without parasitics and this similarly makes sense. When making the layout for our register, there was a minimal use for wiring and as a result, the performance of the register is pretty much identical to that of the ideal simulation.

As mentioned in the previous evaluation, the decrease in VDD increases the delay. In the course, we mention that increasing V_{DD} (up to a certain point) is one of the ways in which we can reduce the propagation delay of our gates. Thus, a decrease in VDD is only cause of increased delay.

Setup time evaluation

Setup time is the minimum time for which the input must be stable before the clock edge rises in order for proper sampling to occur.
To find this, we use the method shown in the recitation where you kind of brute force using a sweep on the delay of the inputs and then after finding a reasonable range of values, use pass-fail testing based on the tcq delay of the cycle.

The pass-fail testing works such that if the delay between the rising clock edge at the time we expect the input to rise is much greater than t_{cq}, then the test fails else if the delay is within reasonable range, then the test passes. This makes sense as if the setup time constraint isn't met then the input signal will be sampled in the next clock cycle. Below are the results presented visually but also tabulated.

Figure 7: tsetup graph visual for VDD=1.2V

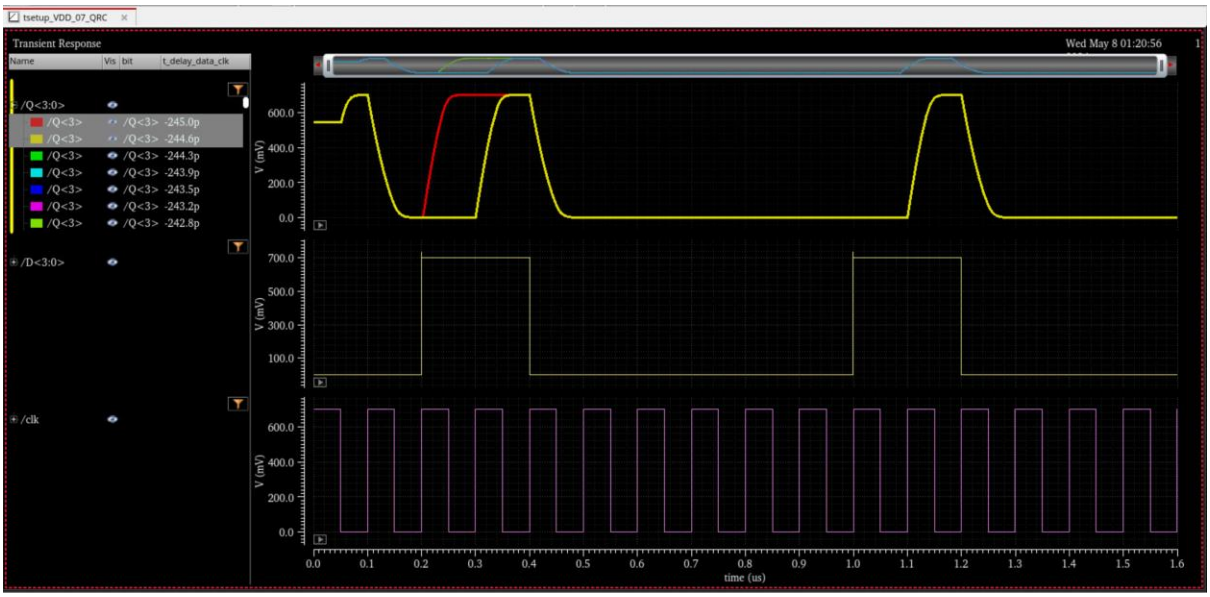

Figure 8: tsetup graph visual for VDD=0.7V

It must be noted that in the above graphs, we were unable to take a print as shown in the recitations as the print wouldn't properly display the transition we were trying to show above.

Next we use pass-fail testing to narrow down the actual t_{su} value.

Parameters: t_delay_data_clk=-244p							
1	tsetup_VDD_07	/Q<3:0>					
1	tsetup_VDD_07	/D<3:0>					
1	tsetup_VDD_07	/clk					
1	tsetup_VDD_07	delayMeasure(VT("/clk") VT("/Q<3>") ?edge1 "rising" ?nth1 2 ?value1 0.35 ?edge2 "rising" ?value2 0.35)	119.2n	tol 20n 10%			fail
Parameters: t_delay_data_clk=-244.6p							
2	tsetup_VDD_07	/Q<3:0>					
2	tsetup_VDD_07	/D<3:0>					
2	tsetup_VDD_07	/clk					
2	tsetup_VDD_07	delayMeasure(VT("/clk") VT("/Q<3>") ?edge1 "rising" ?nth1 2 ?value1 0.35 ?edge2 "rising" ?value2 0.35)	119.2n	tol 20n 10%			fail
Parameters: t_delay_data_clk=-245.1p							
3	tsetup_VDD_07	/Q<3:0>					
3	tsetup_VDD_07	/D<3:0>					
3	tsetup_VDD_07	/clk					
3	tsetup_VDD_07	delayMeasure(VT("/clk") VT("/Q<3>") ?edge1 "rising" ?nth1 2 ?value1 0.35 ?edge2 "rising" ?value2 0.35)	21.28n	tol 20n 10%			pass
Parameters: t_delay_data_clk=-245.7p							
4	tsetup_VDD_07	/Q<3:0>					
4	tsetup_VDD_07	/D<3:0>					
4	tsetup_VDD_07	/clk					
4	tsetup_VDD_07	delayMeasure(VT("/clk") VT("/Q<3>") ?edge1 "rising" ?nth1 2 ?value1 0.35 ?edge2 "rising" ?value2 0.35)	20.68n	tol 20n 10%			pass
Parameters: t_delay_data_clk=-246.2p							
5	tsetup_VDD_07	/Q<3:0>					
5	tsetup_VDD_07	/D<3:0>					
5	tsetup_VDD_07	/clk					
5	tsetup_VDD_07	delayMeasure(VT("/clk") VT("/Q<3>") ?edge1 "rising" ?nth1 2 ?value1 0.35 ?edge2 "rising" ?value2 0.35)	20.42n	tol 20n 10%			pass

Figure 9: t_{su} graph 1 for VDD=0.7V

Parameters: t_delay_data_clk=-246.8p							
6	tsetup_VDD_07	/Q<3:0>					
6	tsetup_VDD_07	/D<3:0>					
6	tsetup_VDD_07	/clk					
6	tsetup_VDD_07	delayMeasure(VT("/clk") VT("/Q<3>") ?edge1 "rising" ?nth1 2 ?value1 0.35 ?edge2 "rising" ?value2 0.35)	20.15n	tol 20n 10%			pass
Parameters: t_delay_data_clk=-247.3p							
7	tsetup_VDD_07	/Q<3:0>					
7	tsetup_VDD_07	/D<3:0>					
7	tsetup_VDD_07	/clk					
7	tsetup_VDD_07	delayMeasure(VT("/clk") VT("/Q<3>") ?edge1 "rising" ?nth1 2 ?value1 0.35 ?edge2 "rising" ?value2 0.35)	20.06n	tol 20n 10%			pass
Parameters: t_delay_data_clk=-247.9p							
8	tsetup_VDD_07	/Q<3:0>					
8	tsetup_VDD_07	/D<3:0>					
8	tsetup_VDD_07	/clk					
8	tsetup_VDD_07	delayMeasure(VT("/clk") VT("/Q<3>") ?edge1 "rising" ?nth1 2 ?value1 0.35 ?edge2 "rising" ?value2 0.35)	19.95n	tol 20n 10%			pass
Parameters: t_delay_data_clk=-248.4p							
9	tsetup_VDD_07	/Q<3:0>					
9	tsetup_VDD_07	/D<3:0>					
9	tsetup_VDD_07	/clk					
9	tsetup_VDD_07	delayMeasure(VT("/clk") VT("/Q<3>") ?edge1 "rising" ?nth1 2 ?value1 0.35 ?edge2 "rising" ?value2 0.35)	19.81n	tol 20n 10%			pass
Parameters: t_delay_data_clk=-249p							
10	tsetup_VDD_07	/Q<3:0>					
10	tsetup_VDD_07	/D<3:0>					
10	tsetup_VDD_07	/clk					
10	tsetup_VDD_07	delayMeasure(VT("/clk") VT("/Q<3>") ?edge1 "rising" ?nth1 2 ?value1 0.35 ?edge2 "rising" ?value2 0.35)	19.76n	tol 20n 10%			pass

Figure 10: t_{su} graph 2 for VDD=0.7V

Parameters: t_delay_data_clk=-117p							
1	tsetup_VDD_12	/Q<3:0>					
1	tsetup_VDD_12	/D<3:0>					
1	tsetup_VDD_12	/clk					
1	tsetup_VDD_12	delayMeasure(V...	102.3n	< 5n			fail
Parameters: t_delay_data_clk=-117.9p							
2	tsetup_VDD_12	/Q<3:0>					
2	tsetup_VDD_12	/D<3:0>					
2	tsetup_VDD_12	/clk					
2	tsetup_VDD_12	delayMeasure(V...	102.3n	< 5n			fail
Parameters: t_delay_data_clk=-118.8p							
3	tsetup_VDD_12	/Q<3:0>					
3	tsetup_VDD_12	/D<3:0>					
3	tsetup_VDD_12	/clk					
3	tsetup_VDD_12	delayMeasure(V...	102.3n	< 5n			fail
Parameters: t_delay_data_clk=-119.7p							
4	tsetup_VDD_12	/Q<3:0>					
4	tsetup_VDD_12	/D<3:0>					
4	tsetup_VDD_12	/clk					
4	tsetup_VDD_12	delayMeasure(V...	102.3n	< 5n			fail
Parameters: t_delay_data_clk=-120.6p							
5	tsetup_VDD_12	/Q<3:0>					
5	tsetup_VDD_12	/D<3:0>					
5	tsetup_VDD_12	/clk					
5	tsetup_VDD_12	delayMeasure(V...	2.321n	< 5n			pass

Figure 11: t_{su} graph 1 for VDD=1.2V

Parameters: t_delay_data_clk=-121.4p						
6	tsetup_VDD_12	/Q<3:0>				
6	tsetup_VDD_12	/D<3:0>				
6	tsetup_VDD_12	/clk				
6	tsetup_VDD_12	delayMeasure(V...	2.308n	< 5n		pass
Parameters: t_delay_data_clk=-122.3p						
7	tsetup_VDD_12	/Q<3:0>				
7	tsetup_VDD_12	/D<3:0>				
7	tsetup_VDD_12	/clk				
7	tsetup_VDD_12	delayMeasure(V...	2.306n	< 5n		pass
Parameters: t_delay_data_clk=-123.2p						
8	tsetup_VDD_12	/Q<3:0>				
8	tsetup_VDD_12	/D<3:0>				
8	tsetup_VDD_12	/clk				
8	tsetup_VDD_12	delayMeasure(V...	2.304n	< 5n		pass
Parameters: t_delay_data_clk=-124.1p						
9	tsetup_VDD_12	/Q<3:0>				
9	tsetup_VDD_12	/D<3:0>				
9	tsetup_VDD_12	/clk				
9	tsetup_VDD_12	delayMeasure(V...	2.301n	< 5n		pass
Parameters: t_delay_data_clk=-125p						
10	tsetup_VDD_12	/Q<3:0>				
10	tsetup_VDD_12	/D<3:0>				
10	tsetup_VDD_12	/clk				
10	tsetup_VDD_12	delayMeasure(V...	2.302n	< 5n		pass

Figure 12: t_{su} graph 2 for VDD=1.2V

From the above, the setup times is tabulated below. It should be noted that the setup times with and without QRC were pretty much identical so these graphs suffice.

	$V_{DD} = 1.2V$	$V_{DD} = 0.7V$
t_{su} $(with\ and\ without\ QRC)$	0.1206n	0.2451n

So from the above, we find the minimal time periods for each scenario:

Source Voltage	**QRC**	**Without QRC**
VDD = 1.2	(0.1206+4.066+4.545) n=8.7316n	(0.1206+4.066+4.18) n = 8.3665n
VDD=0.7	(0.2451+19.15+22.34) n = 41.735n	(0.2451+19.15+19.86) n = 39.255n

Taking reciprocal of minimum period gives maximum frequency:

Source Voltage	**QRC**	**Without QRC**
VDD = 1.2	114.526 MHz	119.524 MHz
VDD = 0.7	23.96 MHz	25.474 MHz

For the t_{cq} measurement and t_{logic}, it's crucial to have the threshold in delay measure set to 0.6V for VDD = 1.2V and 0.35 for VDD =0.7V else you risk erroneous measurements.

From the final evaluated values above, we can conclude that the supply voltage has a very huge effect on the minimal period of the circuit. This is due to the increase in the propagation delays which was already explained above. Similarly, there is a deviation in the minimal period for the same VDD but with extracted QRC parameters. This was also explained above, but in summary QRC extracted circuits are closer to practicality and in reality, interconnects and vias contribute to increased RC delays.