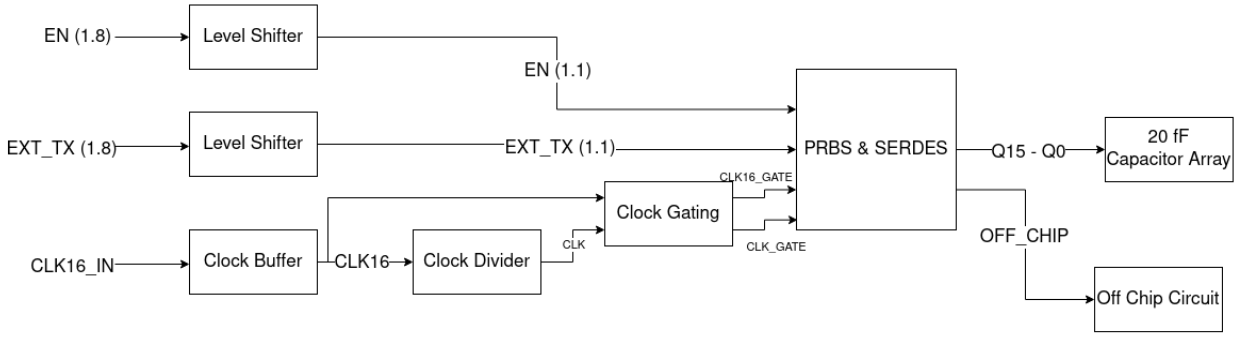
ECE 482 Final Project Report

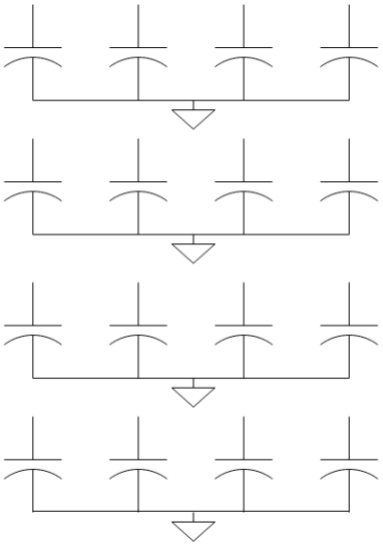
Larry Du, Candy Gao, Dev Patel, Himalaya Rautela

1. Overall Design Approach



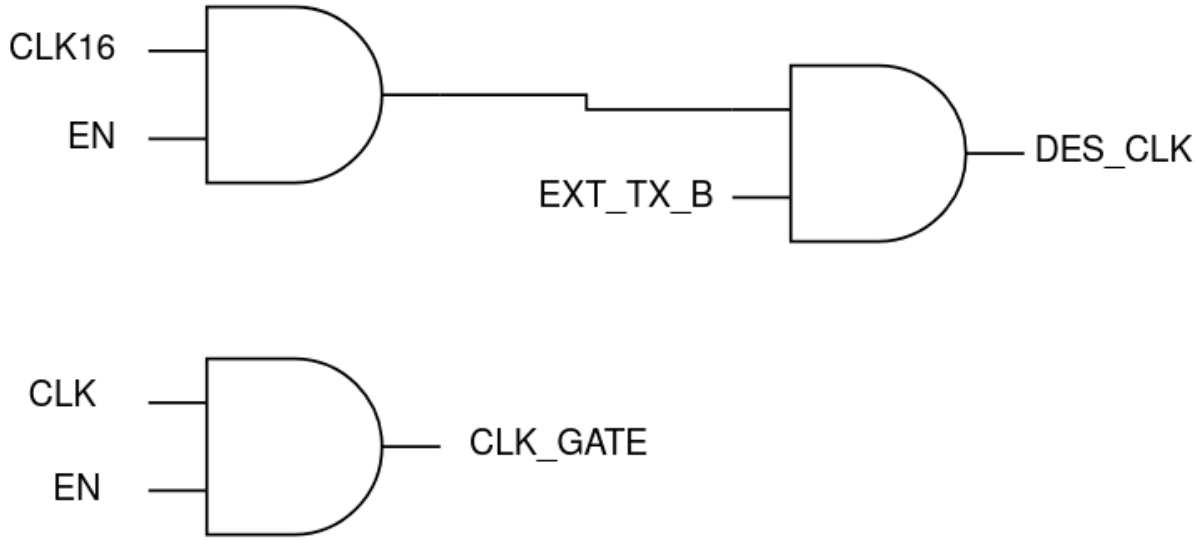
Top Level Block Diagram.

The top level architecture shown above compartmentalizes large components, for ease of unit testing.



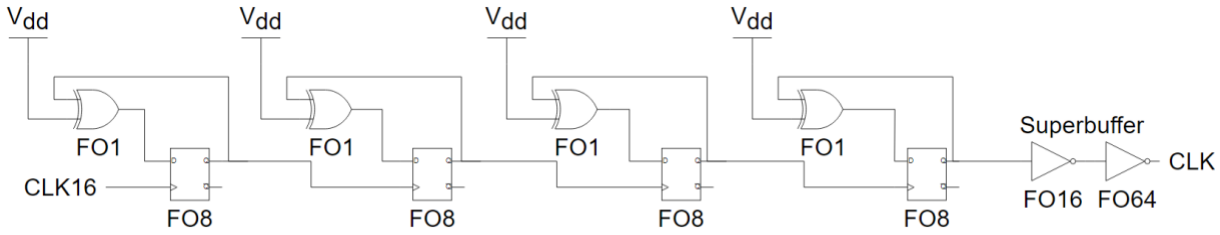
Capacitor Array.

Each capacitor shown above is 20 fF, and connects to one of the deserializer outputs Q0 to Q15 as required by the specifications.



Clock Gating.

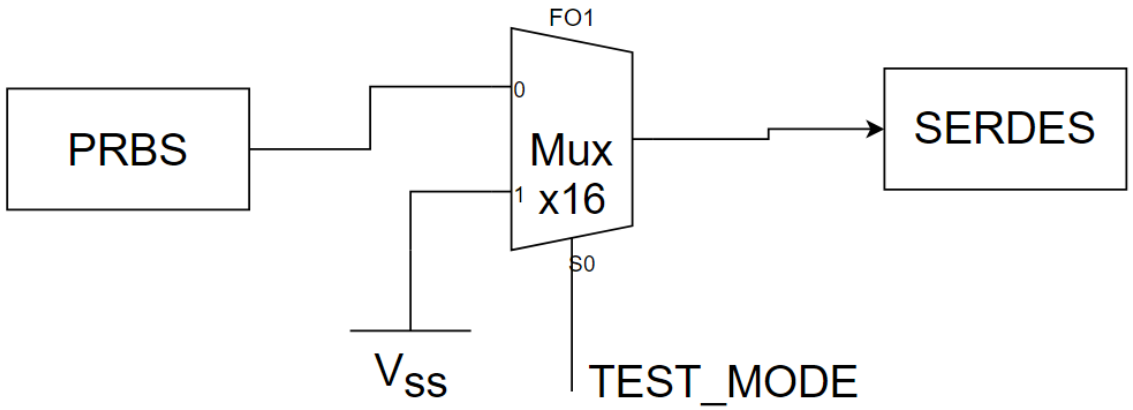
The clock gating circuitry gates CLK and CLK16 based on EN, and then further gates the deserializer clock with the inverted EXT_TX signal so that the deserializer will be turned off if we want to send the data off chip. This saves power.



Clock Divider.

The clock divider takes a 1600 MHz CLK16 input and outputs a 100 MHz CLK output. It utilizes size 8 registers to reduce propagation delay and increase drive strength for the critical CLK signal. It also features a two inverter superbuffer to further increase the drive strength of the CLK.

The clock divider works in four stages. At every stage, the register will invert its output every clock cycle, due to the xor circuitry. We cascade this four times to produce a divide by 16 output.

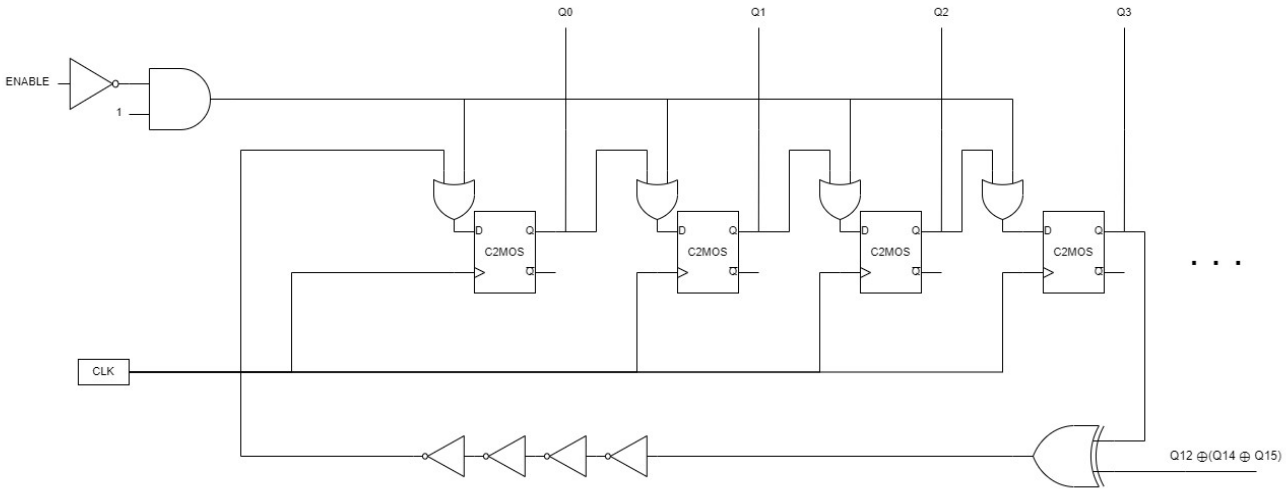


PRBS & SERDES.

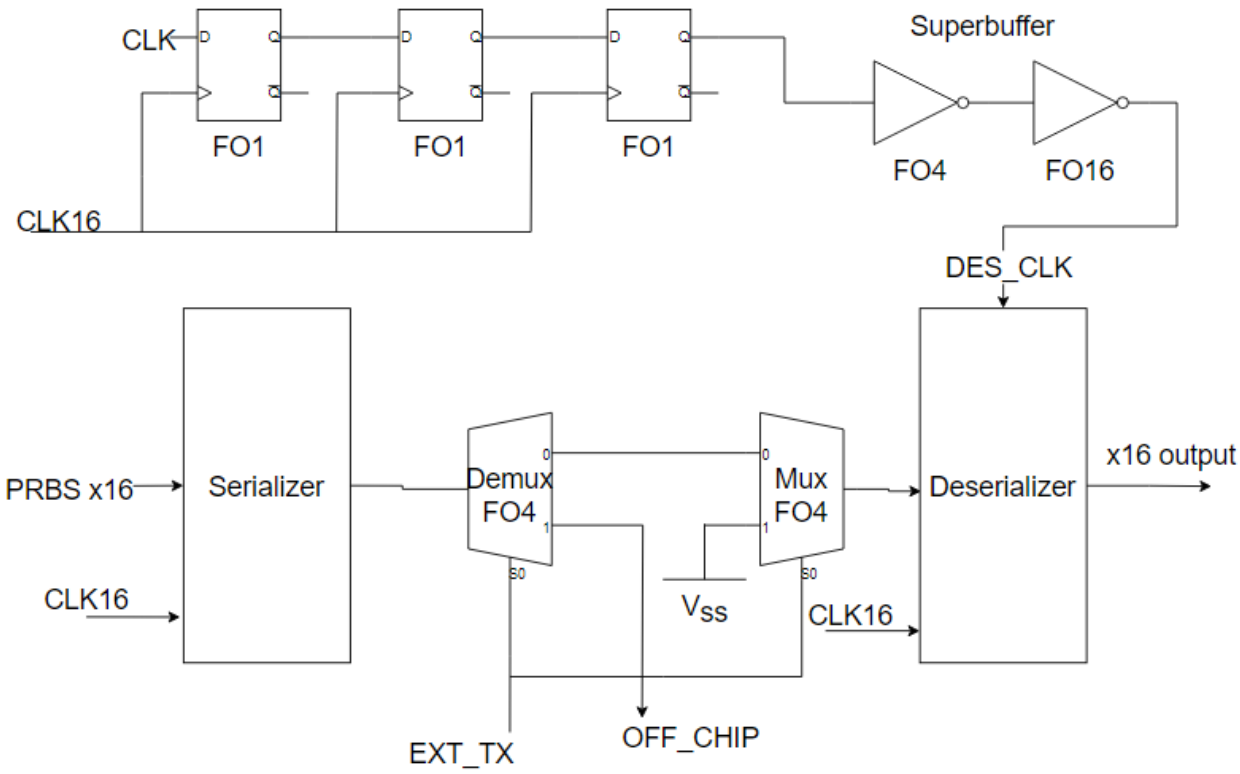
Moving along in the hierarchy, the PRBS & SERDES block consists very simply of the PRBS, SERDES, and 16 muxes. The reason for this extra hierarchy level is to minimize layout routing errors, allowing us to ensure this block passes LVS without having to implement everything in the top level. All muxes are minimum size to reduce area, and since these muxes drive registers clocked at 100 MHz, they do not have to be fast.

The TEST_MODE signal is used to differentiate between test data from the PRBS and “real” data. In our case, since we are not utilizing the SERDES in a context outside of PRBS input, all “real” data is connected to VSS. This can easily be changed if we were actually designing a commercial chip.

PRBS GENERATOR



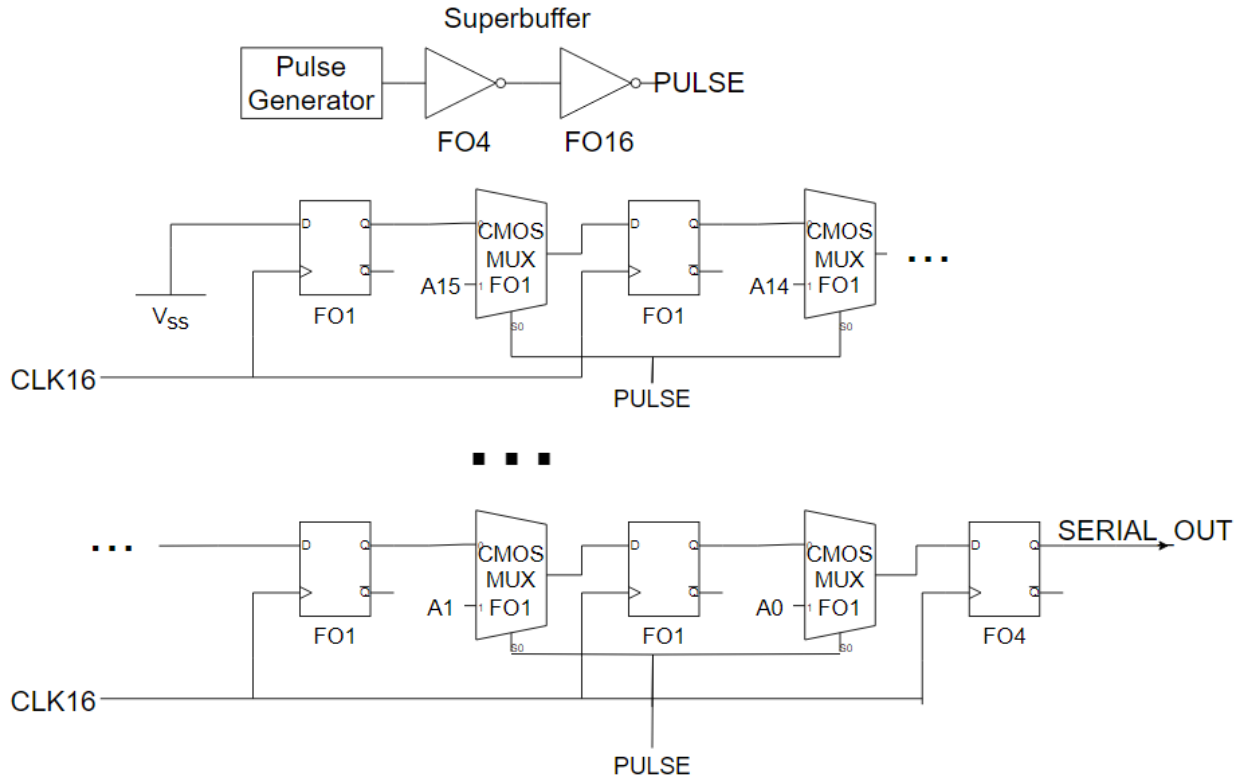
The PRBS generator was design following the specifications listed in the digikey article. To meet the requirement that when PRBS is activated (EN is high), we added a OR gate such that when EN is low 1 is inputted into all the registers. In this way when PRBS is activated, all the registers will output a 1 on the first positive clock edge. A small buffer is used to delay the feedback input into register 1 so that the correct the input can be latched on. C2MOS registers are chosen due to its insensitivity to clock skew.



SERDES.

The SERDES block consists of the serializer, deserializer, off chip connectivity, and some circuitry to generate a delayed 100 MHz clock for the deserializer. The serializer receives the 16 parallel inputs from the PRBS, and outputs them at 1600 MHz. In the center, we have a demux and mux for toggling of loopback vs off chip mode. Both are size 4 to reduce delay at the faster clock speed.

The circuitry at the top is used to delay CLK by exactly 3 CLK16 cycles, followed by standard superbuffer circuitry. The reason for this is because the serializer will parallel load on a 1600 MHz pulse generated every 100 MHz, and this pulse will be generated 2 CLK16 cycles after the CLK rising edge (refer to the pulse generator section for details). Therefore, the deserializer is receiving serial data at a delay of 3 CLK16 cycles relative to CLK, and so it should only lock in parallel data after this delay.

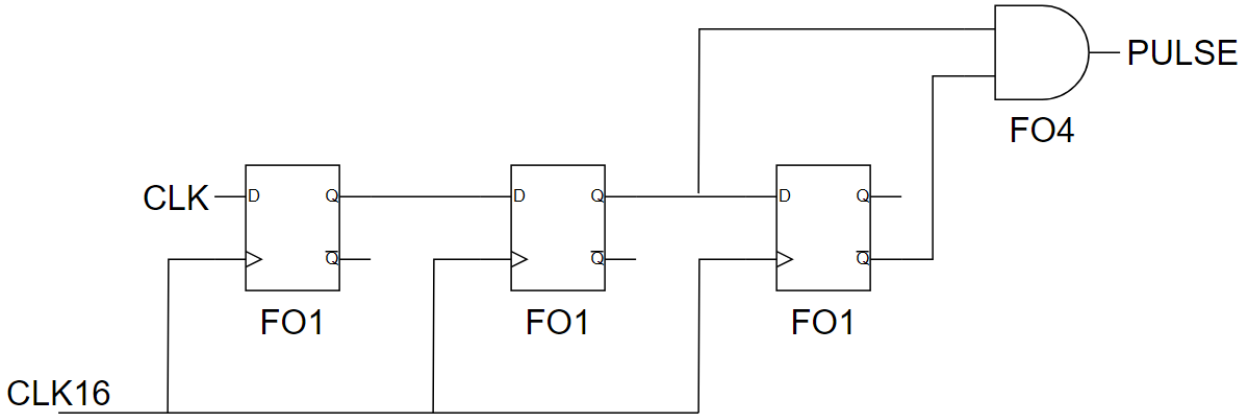


Serializer.

The serializer consists of a pulse generator that generates a 1600 MHz pulse every 100 MHz, as well as 16 register mux modules plus a final register at the output.

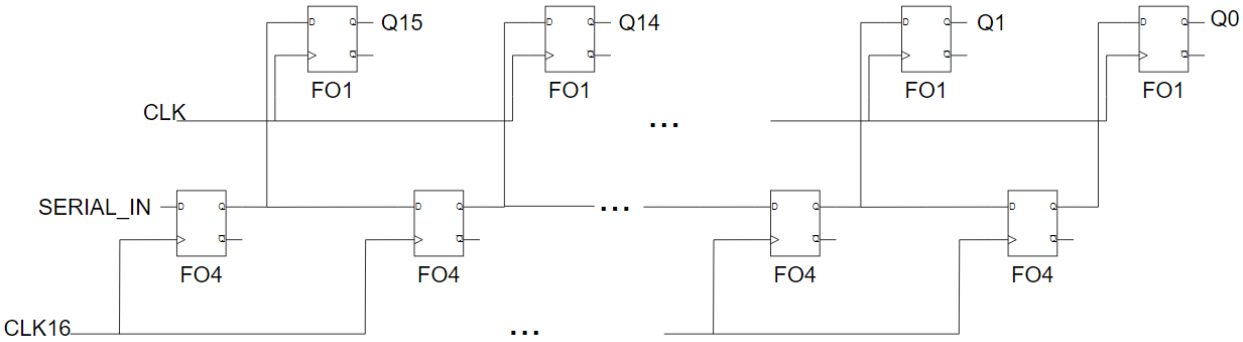
Every 100 MHz, the pulse will go high, and the serializer will induce a parallel load: every register receives one of the 16 A inputs. Then, at every subsequent 1600 MHz rising edge, the serializer will act as a shift register, shifting all 16 inputs to SERIAL_OUT. All registers are minimum size because they don't drive much, except for the final register, which is size 4 to increase its drive strength going into SERDES space.

All muxes in the serializer are CMOS. This is to ensure that the area between registers is never high Z and is always driven either to VDD or VSS. Due to the fast nature of the serializer circuitry, it was common for us to see that the PULSE signal would not drive low fast enough for the mux to latch onto the previous register's output before the CLK16 signal went low. This resulted in a high Z node which kept the parallel load input, rather than being driven to the register output. The CMOS muxes prevent such a problem.



Pulse Generator.

The pulse generator follows standard edge detector design. When the CLK undergoes a low to high transition, the Q and Q_bar inputs to the AND gate will be the same for exactly one cycle of CLK16, before the rightmost register latches onto Q and the inputs to the AND gate are inverted. The result is a 1600 MHz pulse every 100 MHz rising edge.



Deserializer.

The deserializer is 16 FO1 registers and 16 FO4 registers. It acts as a giant shift register through the FO4s with parallel output through the FO1s. Every CLK edge, the FO1s will lock in whatever is being shifted through the FO4s, making proper timing critical in the deserializer. We use FO4s to reduce propagation delay of the shift registers in this time sensitive environment.

Off-Chip Components: Driver and Level-shifter

The off-chip driver for the final layout (designed to meet specs for both C-only and RC extraction) is a superbuffer consisting of 4 inverter gates of sizes 3.125, 6.25, 18.75 and 62.5 respectively. All of the gates are set up for $\beta = 2$.

It is desired that the off-chip driver should meet two criteria:

1. Be able to drive the output of the buffer with capacitance $C_{TOT} = C_{int}+C_{pkg}+C_L$ from 0 V to 0.9 V_{DD} within 625 ps – C_L = 2pF and C_{pkg} = 3fF – without having a transmission line attached.
2. The driver should be connected to the circuit as shown in Figure _, with the driver replacing the resistor R_{dr}. The circuit should be simulated and the output should meet the specifications of the Figure of Merit (FOM), i.e., FOM > 80%. The FOM is given as:

$$FOM\ (\%)\ = \frac{\int_{t_{peak}-UI}^{t_{peak}+UI} |V_{RX}|\ dt}{\int_{0}^{\infty} |V_{RX}|\ dt}\ X\ 100$$

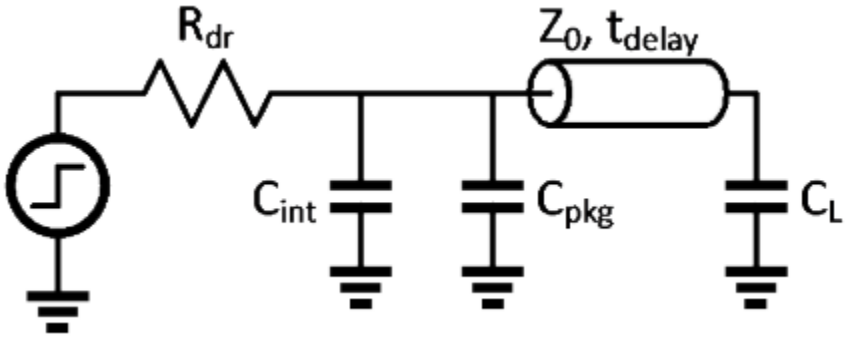


Figure _:

Initially, to meet the criterion of output driven from 0 to 0.9 V_{DD} without a transmission line, four inverters of sizes 4, 16, 64 and 256 are set up to drive a total capacitance of 2.3 pF plus the internal capacitance of the driver. The optimum fanout per gate is around 3.5, which is rounded to 4. However, the setup does not result in an adequate figure of merit (FOM) greater than 80%, as shown in Figure _. With further experimentation, it was determined that, to meet the FOM criterion, the last gate in the superbuffer chain that outputs to the transmission line should be no larger than s = 64 with β = 2. However, the gate is noticeably slower in meeting the first criterion. Therefore any preceding inverters should be designed to meet criterion 1. To ensure output and input are high at the same time, the number of gates in the superbuffer should be even. It is possible to meet both criteria with a 2-gate superbuffer and a low input capacitance when C-only extraction is chosen. However if RC is selected, the 2-gate superbuffer is unable to meet either criteria. Hence it is decided to proceed with a 4-gate superbuffer.

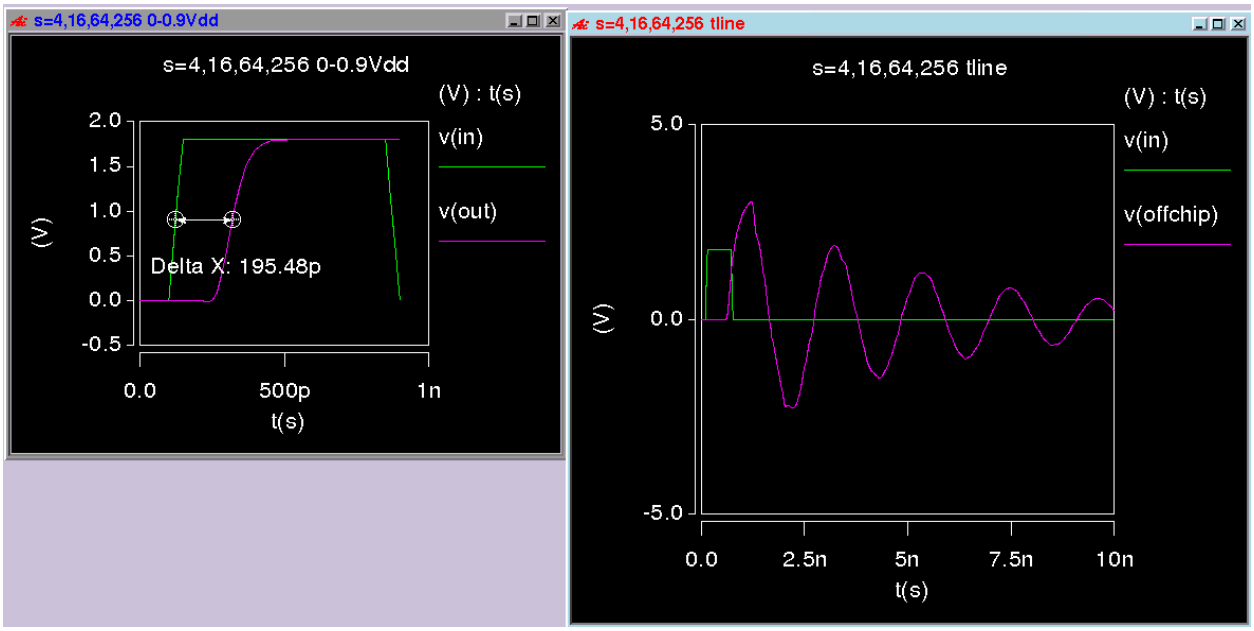


Figure _: The results from a 4-gate off-chip driver with inverters of sizes s=4, 16, 64, 256, with C-only extraction. Criterion 1 is satisfied, but as seen on the right, the ringing seen in the off-chip response is significant. This inter-symbol interference will compromise the ability of the receiver to distinguish the transmitted bit.

The gate sizes chosen – s = 3.125, 6.25, 18.75 and 62.5 – are considered functionally equivalent to sizes 4, 8, 16 and 64. The reason fractional sizes are chosen is because their dimensions are multiples of 1 μm – for example the gate size of 3.125 corresponds to a NMOS width of 1 μm (for thick oxide, minimum width is 320 nm).

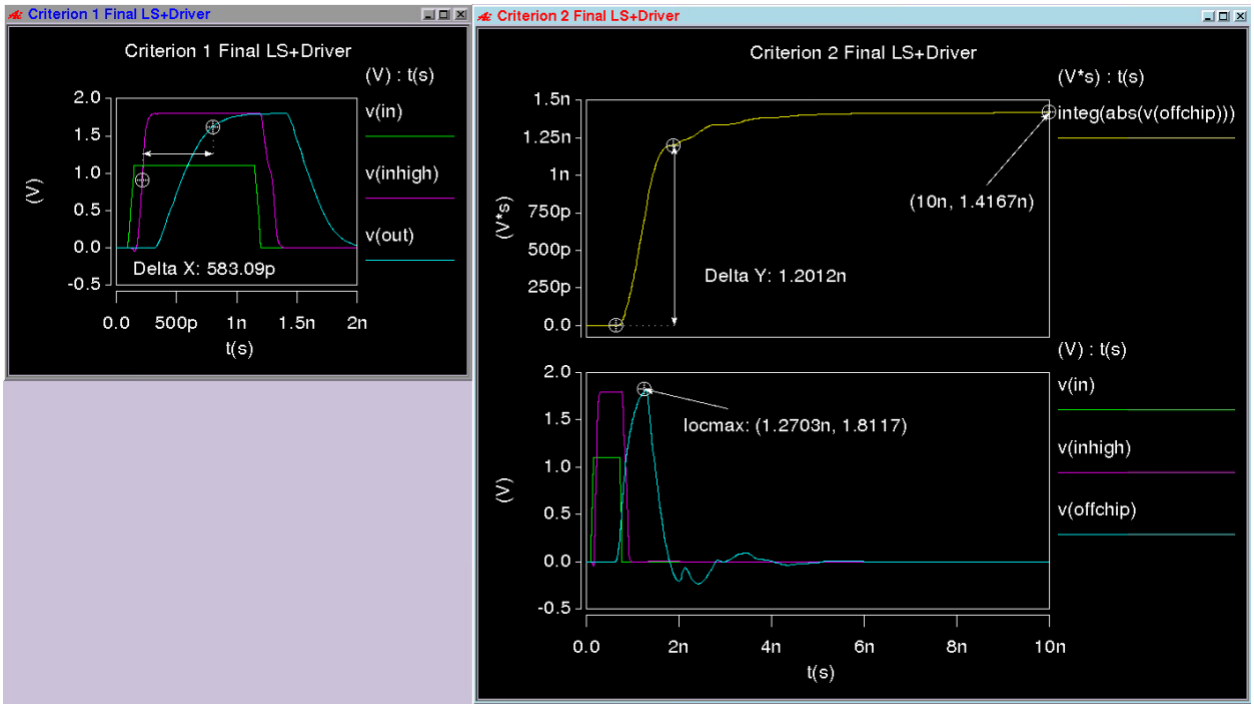
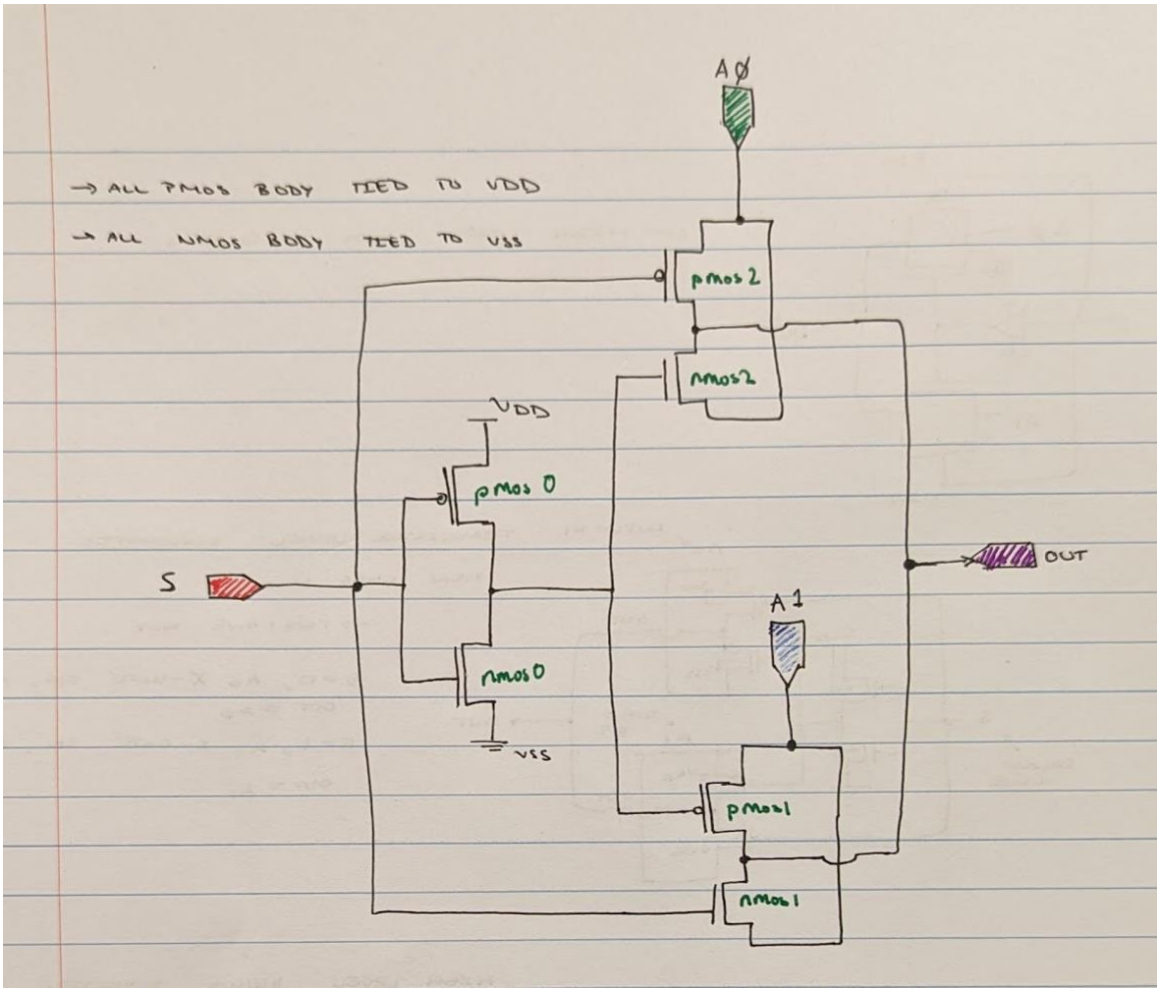


Figure _: The results from a level-shifter followed by a 4-gate off-chip driver with inverters of sizes s = 3.125, 6.25, 18.75 and 62.5, with RC extraction. v(in) and v(inhigh) represent the inputs to the level-shifter and the driver respectively. v(out) and v(offchip) represent the output of the gate and the transmission line respectively. Both criteria are met comfortably (driver delay of 583.09 ps and FOM = 84.79 %). This is the final design.

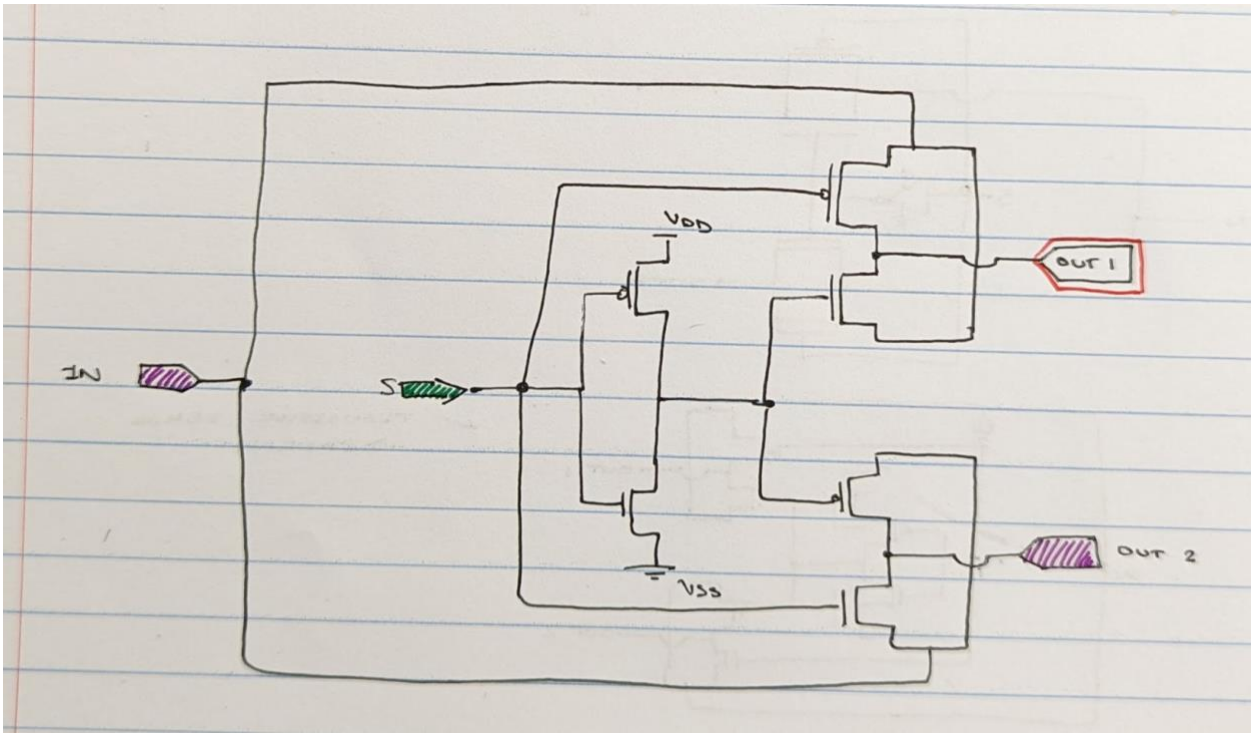
2. Transistor Level Schematics

Pass Mux Transistor Level Schematic:



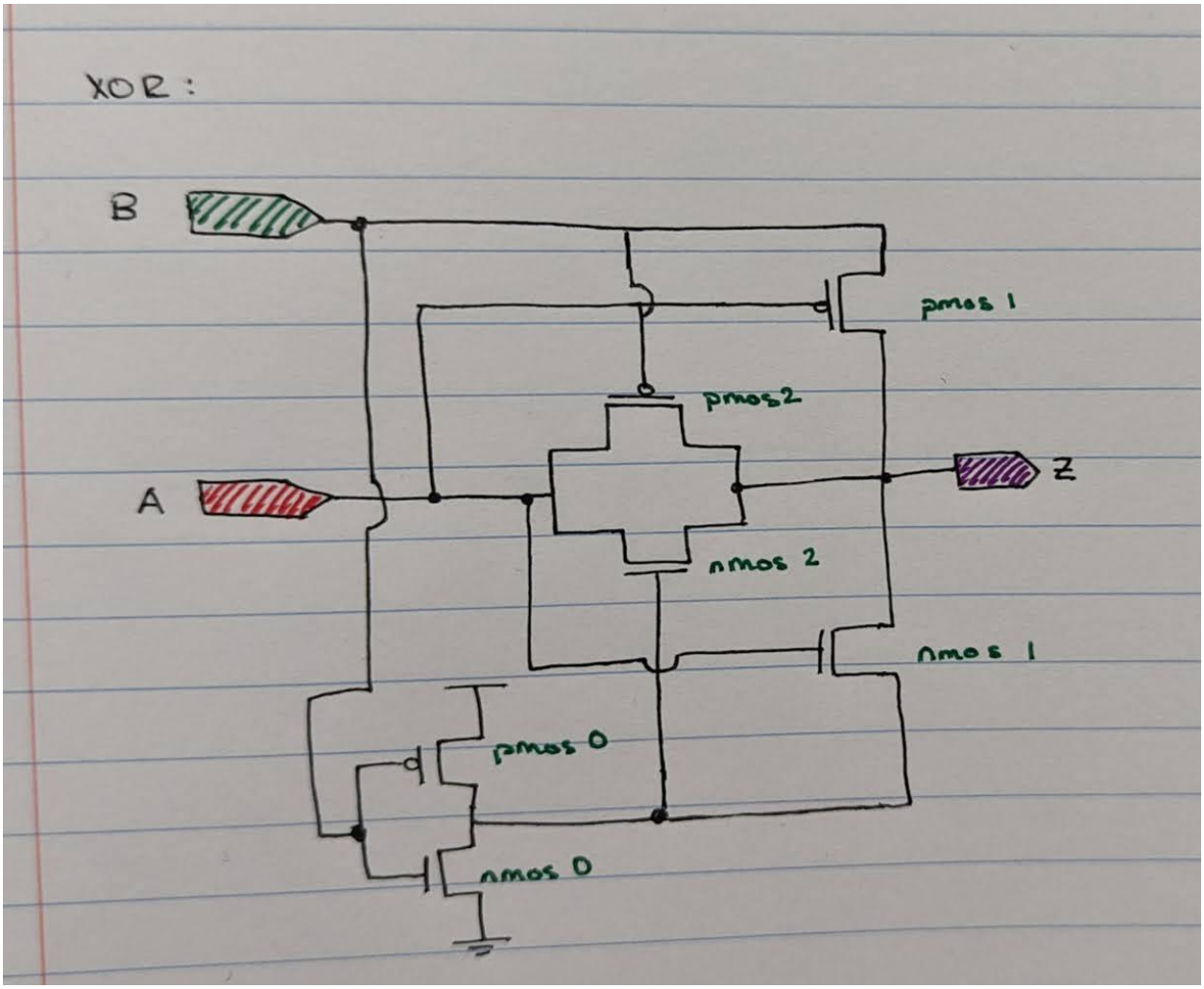
Mux Sizing	Width	Length
NMOS0	120n	45n
PMOS0	240n	45n
NMOS1	120n	45n
PMOS1	240n	45n
NMOS2	120n	45n
PMOS2	240n	45n

Demux Transistor Level Schematic:



Transmission Gate Sizing	Width	Length
NMOS0	120n	45n
PMOS0	240n	45n
NMOS1	120n	45n
PMOS1	240n	45n
NMOS2	120n	45n
PMOS2	240n	45n

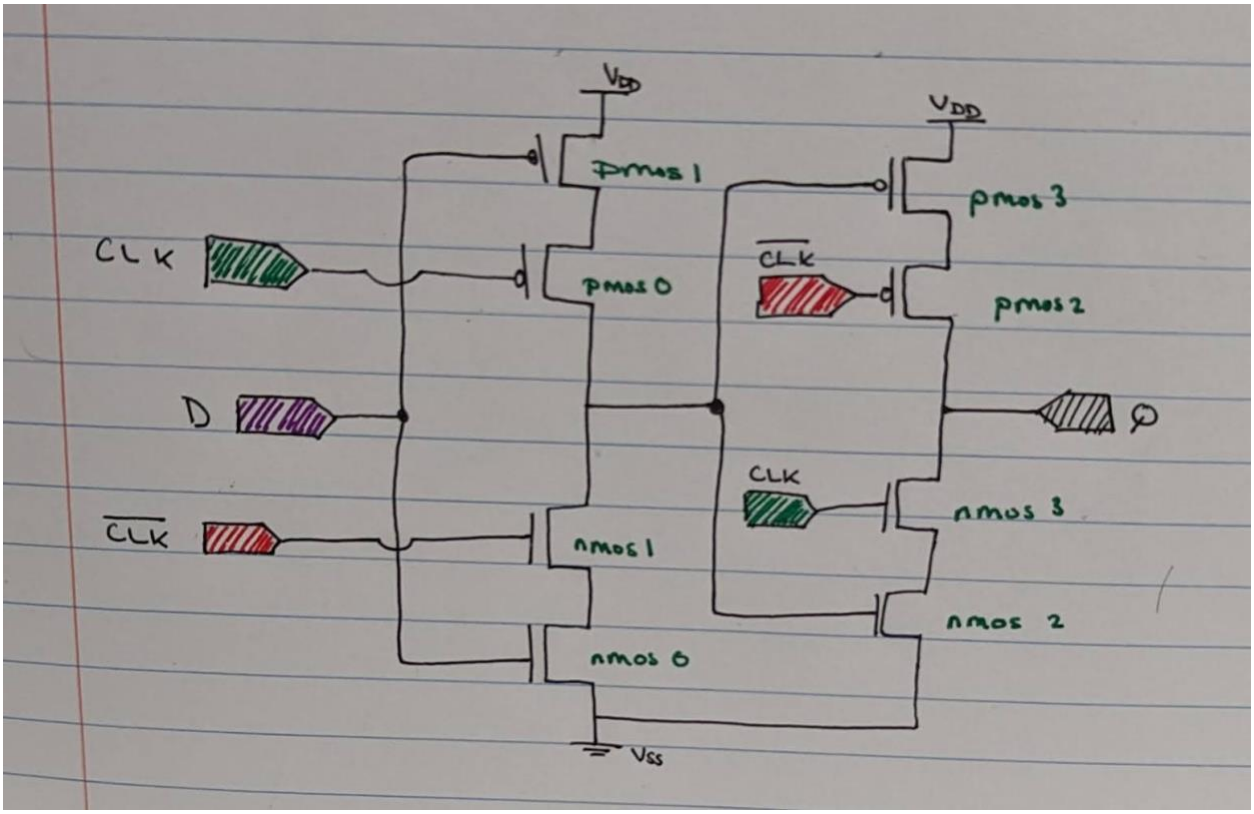
XOR Transistor Level Schematic:



Transistor Sizing Table:

XOR Gate Sizing	Width	Length
NMOS0	120n	45n
PMOS0	240n	45n
NMOS1	120n	45n
PMOS1	240n	45n
NMOS2	120n	45n
PMOS2	240n	45n

Register Transistor Level Schematic:



Transistor Sizing Table:

Register Gate Sizing	Width	Length
NMOS0	240n	45n
PMOS0	480n	45n
NMOS1	240n	45n
PMOS1	480n	45n
NMOS2	240n	45n
PMOS2	480n	45n
NMOS3	240n	45n
PMOS3	480n	45n

Transistor Sizing Table:

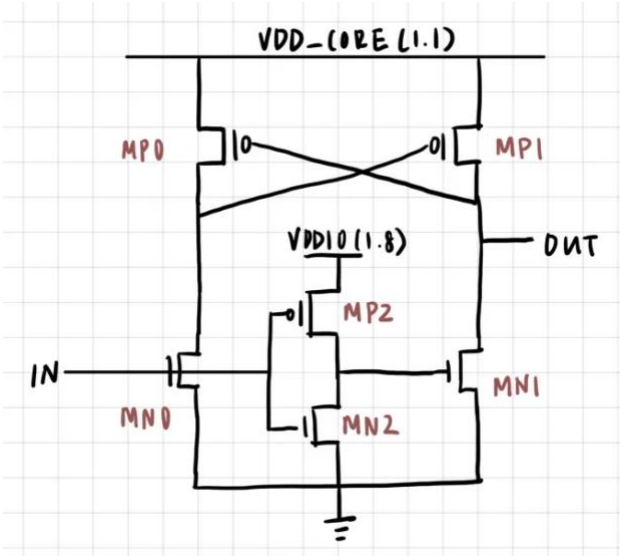
Register Gate Sizing (F04)	Width	Length
NMOS0	480n	45n
PMOS0	960n	45n
NMOS1	480n	45n
PMOS1	960n	45n
NMOS2	480n	45n
PMOS2	960n	45n
NMOS3	480n	45n
PMOS3	960n	45n

Transistor Sizing Table:

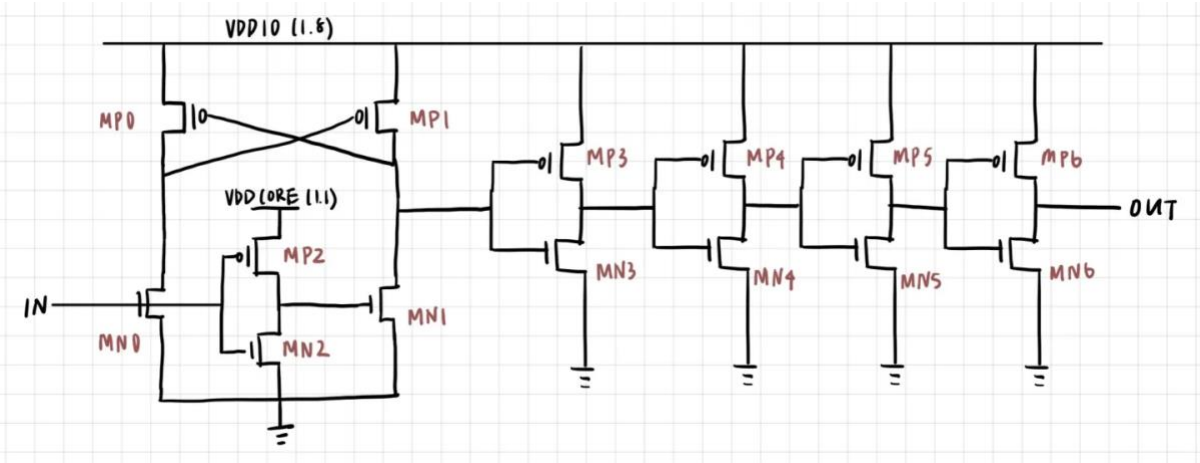
Register Gate Sizing (F08)	Width	Length
NMOS0	960n	45n
PMOS0	1920n	45n
NMOS1	960n	45n
PMOS1	1920n	45n
NMOS2	960n	45n
PMOS2	1920n	45n
NMOS3	960n	45n
PMOS3	1920n	45n

Level Shifter (From 1.8 to 1.1):

Transistor	**W/L**
MP0 MP1 (pmos1v)	120n/45n
MN0 MN1 MN2 (nmos2v)	320n/150n
MP2 (pmos2v)	640n/150n



Level Shifter (From 1.1 to 1.8) + Off-chip driver:



Transistor	W/L
MP0 MP1 (pmos2v)	1µ/150n
MN0 MN1 (nmos2v)	1.5µ/150n
MP2 (pmos1v)	240n/45n
MN2 (nmos1v)	120n/45n
MP3 (pmos2v)	2µ/150n
MN3 (nmos2v)	1µ/150n

Transistor	W/L
MP4 (pmos2v)	4µ/150n
MN4 (nmos2v)	2µ/150n
MP5 (pmos2v)	12µ/150n
MN5 (nmos2v)	6µ/150n
MP6 (pmos2v)	40µ/150n
MN6 (nmos2v)	20µ/150n

3. TT Corner Circuit Functionality

PRBS Manual Analysis

	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13	Q14	Q15
0	**1**	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	**0**	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	**0**	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	**0**	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1
4	**0**	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1

Simulated Results

TT Process Corner

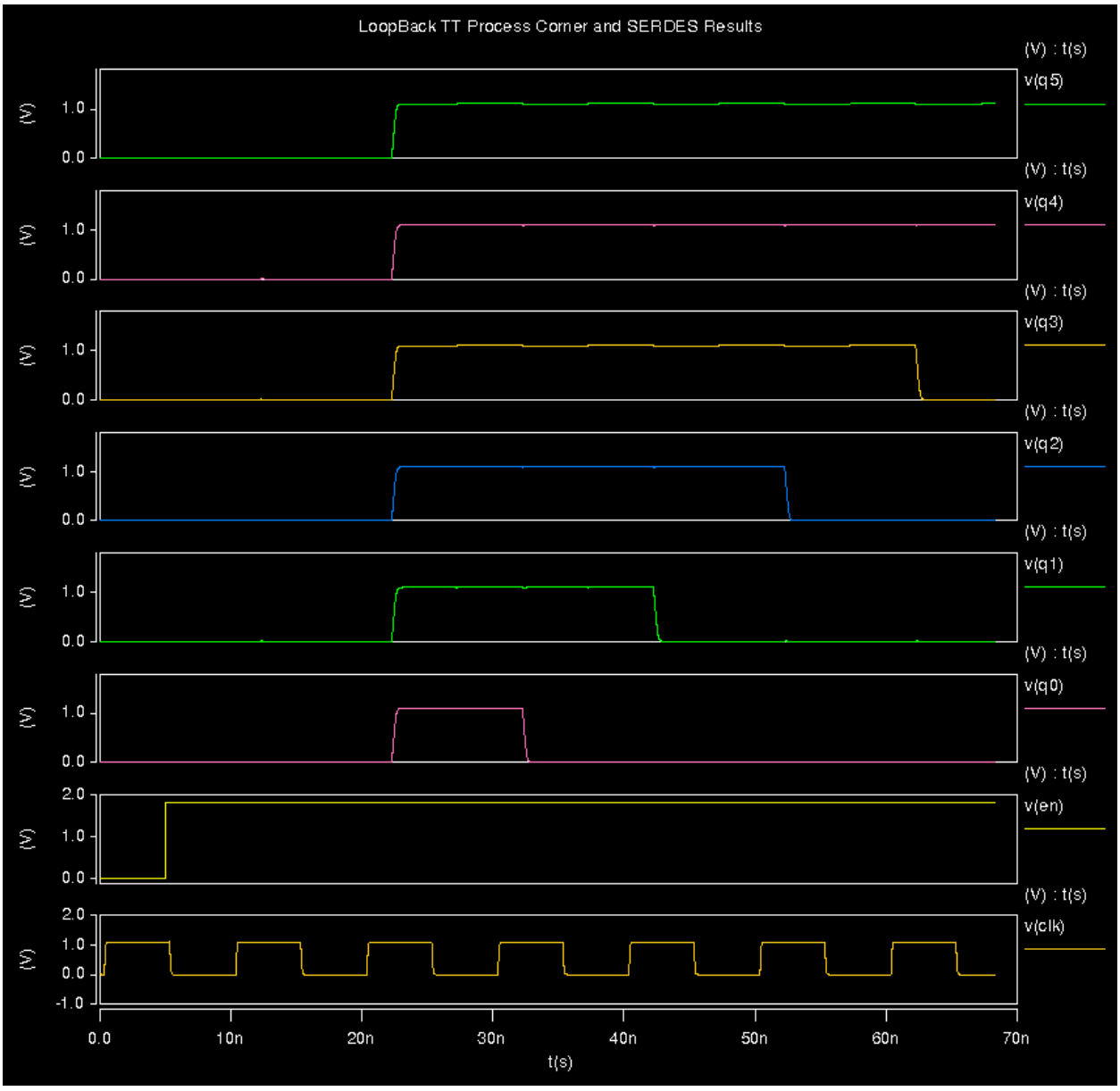


Figure 1: SERDES results for the TT Process Corner.
For all process corners, we elected not to show q6-q15, as they were the exact same as q4 and q5.

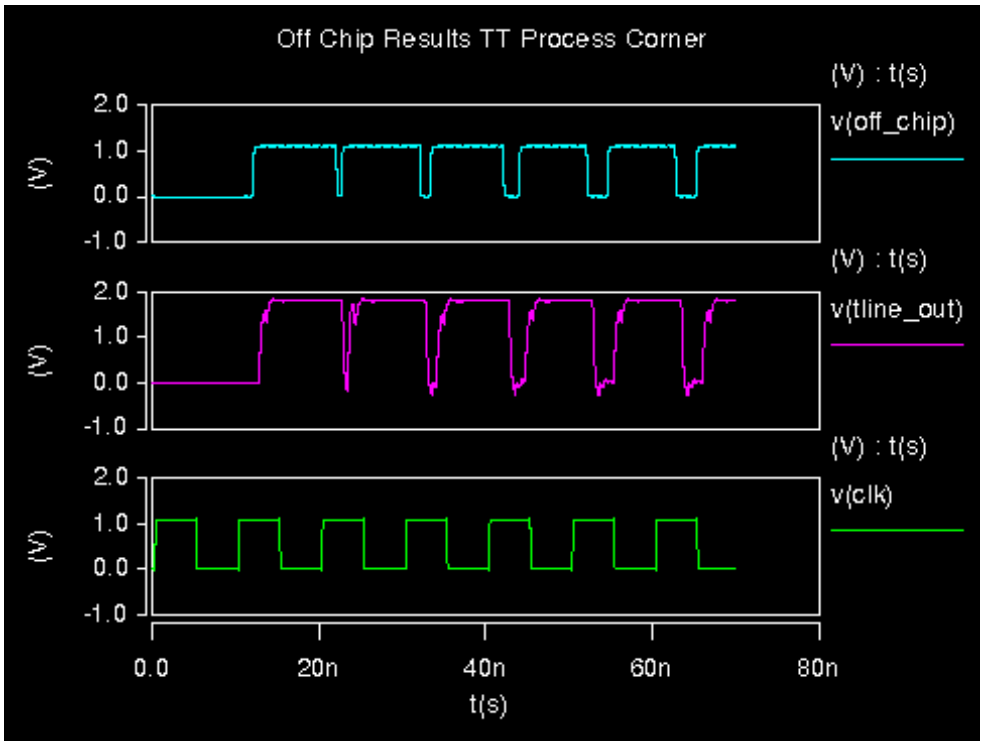
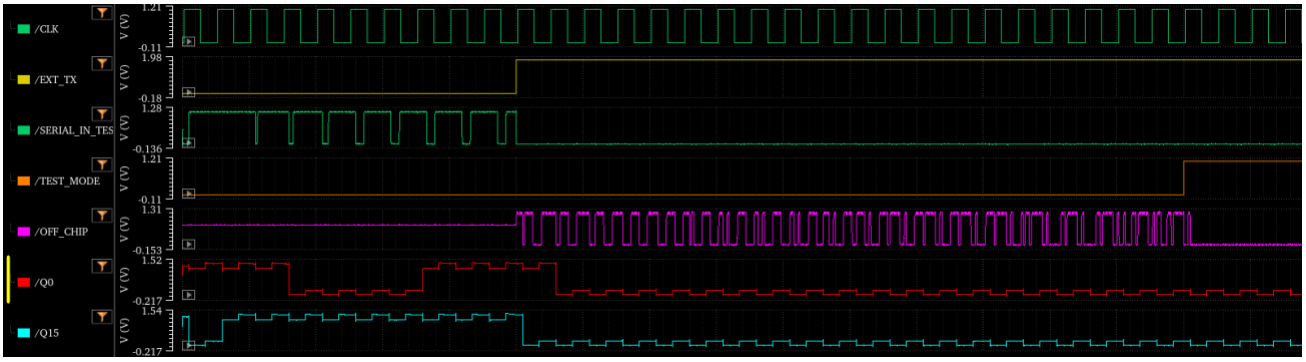


Figure 2: Off-chip results for TT process corner. Here, contrary to the "Off-Chip Components" section mentioned above, v(off_chip) here refers to voltage at the **input** of the level-shifter that precedes the off-chip driver. v(tline_out) refers to the voltage at the load end of the transmission line.

Functionality of the Multiplexer Circuit:



To demonstrate functionality of the multiplexer circuit in the PRBS and SERDES, refer above. Initially, EXT_TX = 0 and TEST_MODE = 0, so we should expect the deserializer to receive PRBS data, as can be seen by SERIAL_IN_TEST and output is reflected in Q0. Remember that TEST_MODE = 0 means PRBS input, and TEST_MODE = 1 means “real” data, which for our case is tied to VSS. At 100ns, EXT_TX goes high, and input to the deserializer immediately ceases as OFF_CHIP begins to output serial data. At 300ns, TEST_MODE goes high, and OFF_CHIP goes low, indicating that it is receiving the appropriate VSS data.

Power Measurements for TT:

EXT_TX	EN	Total Power
0	**0**	1.61 mW (VDD) + 4.05 mW (VDD_IO = 4.66 mW
0	**1**	1.058mW (VDD) + 0.285mW (Vdd_IO) = 1.343 mW
1	**1**	0.77mW (VDD) + 4.3mW (VDD_IO) = 5.07 mW

To reduce power, we primarily utilize clock gates. We also made sure to use small logic whenever possible to reduce power draw. In addition, we have little dynamic logic in order to further reduce power consumption.

Clearly the output for EXT_TX = 0 and EN = 0 are not correct. This is likely an issue with our simulation netlist. In practicality it should be less than the other two outputs.

4. SS, SF, FS, and FF Corner Circuit Functionality

FF Process Corner

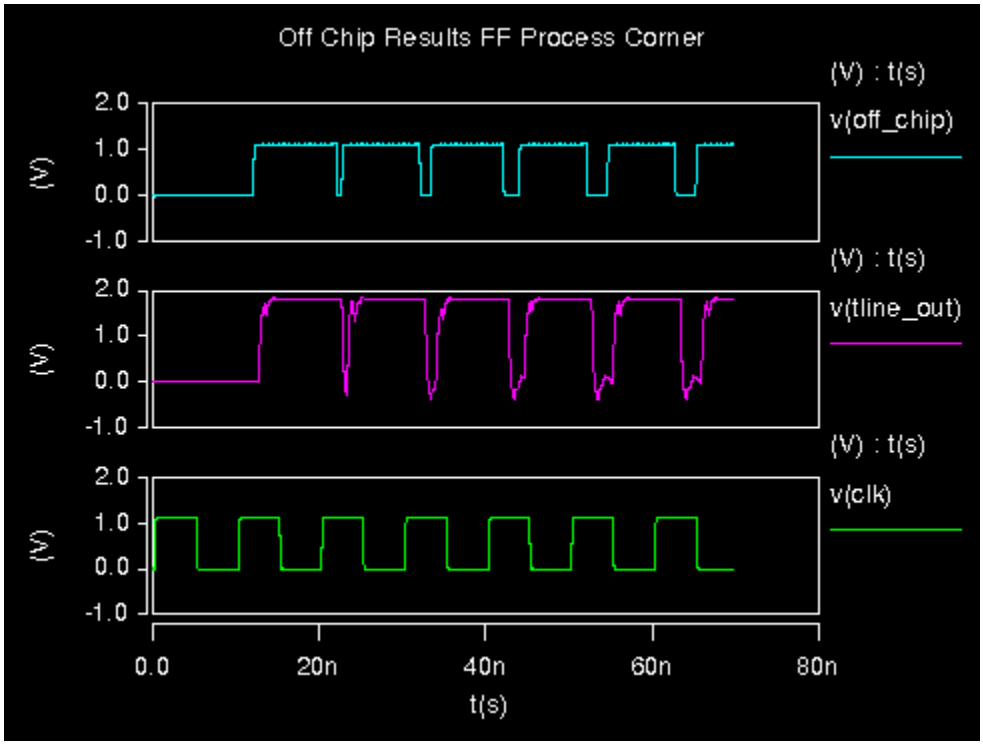


Figure 3: Off-chip results for the FF Process Corner. Here, v(off_chip) refers to voltage at the **input** of the level-shifter that precedes the off-chip driver. v(tline_out) refers to the voltage at the load end of the transmission line.

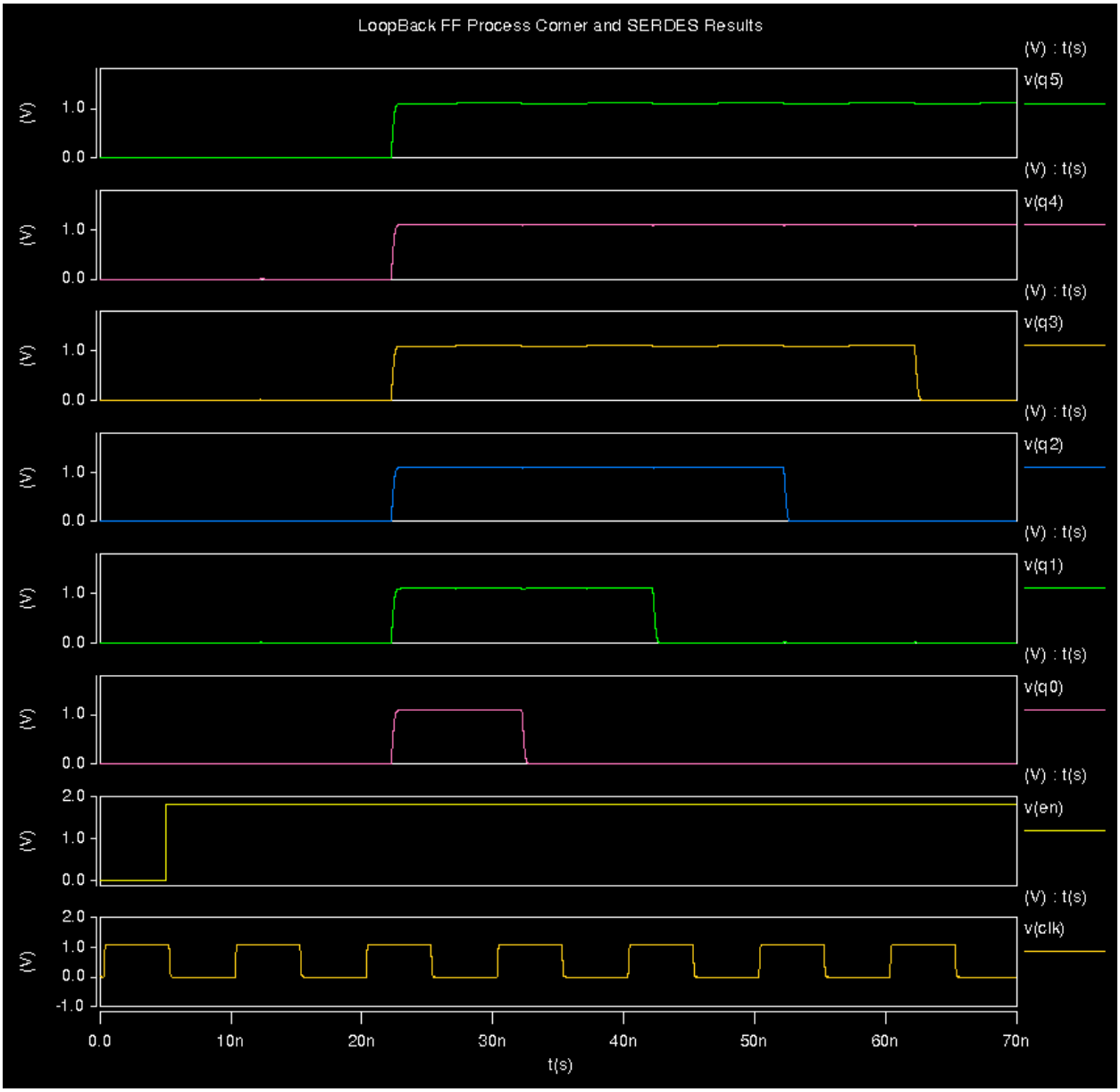


Figure 4: SERDES results for the FF Process Corner.

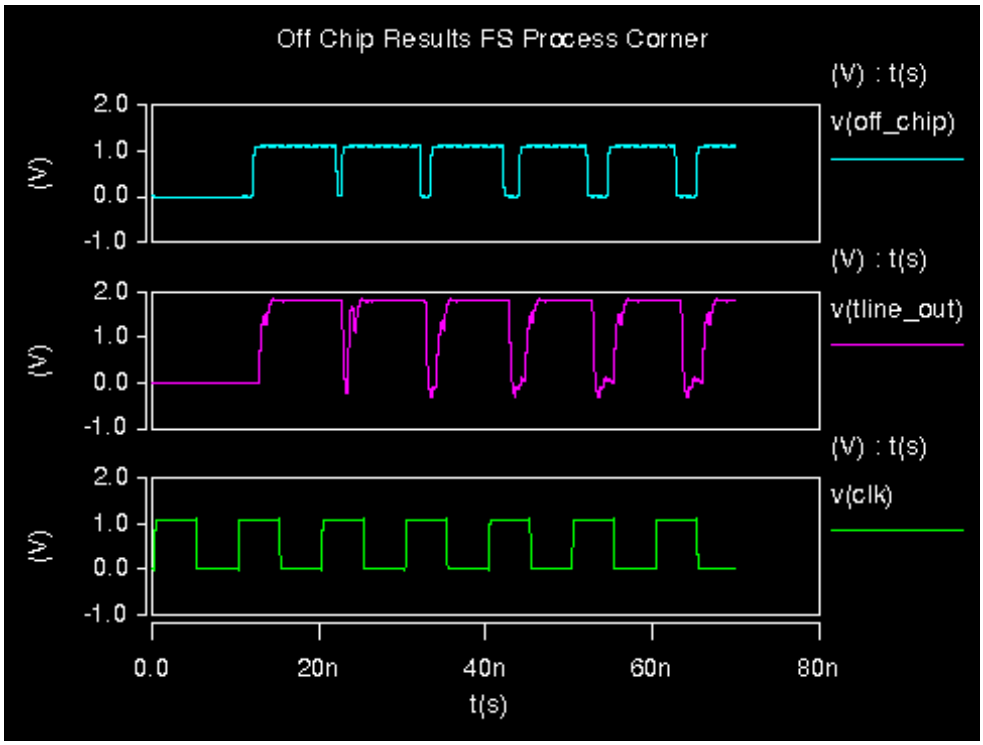


Figure 5: Off-chip results for the FS Process Corner.

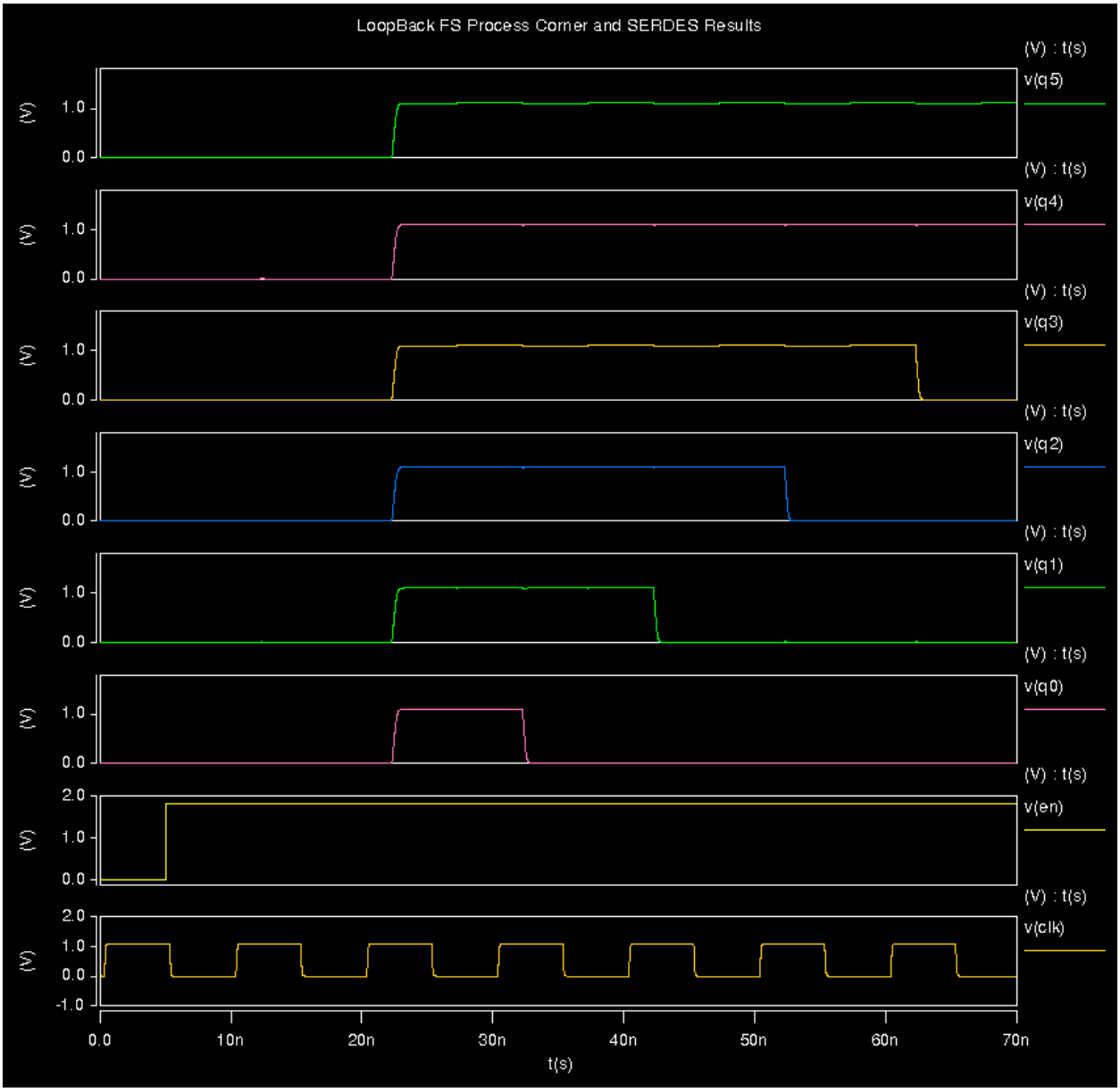


Figure 6: SERDES results for the FS Process Corner.

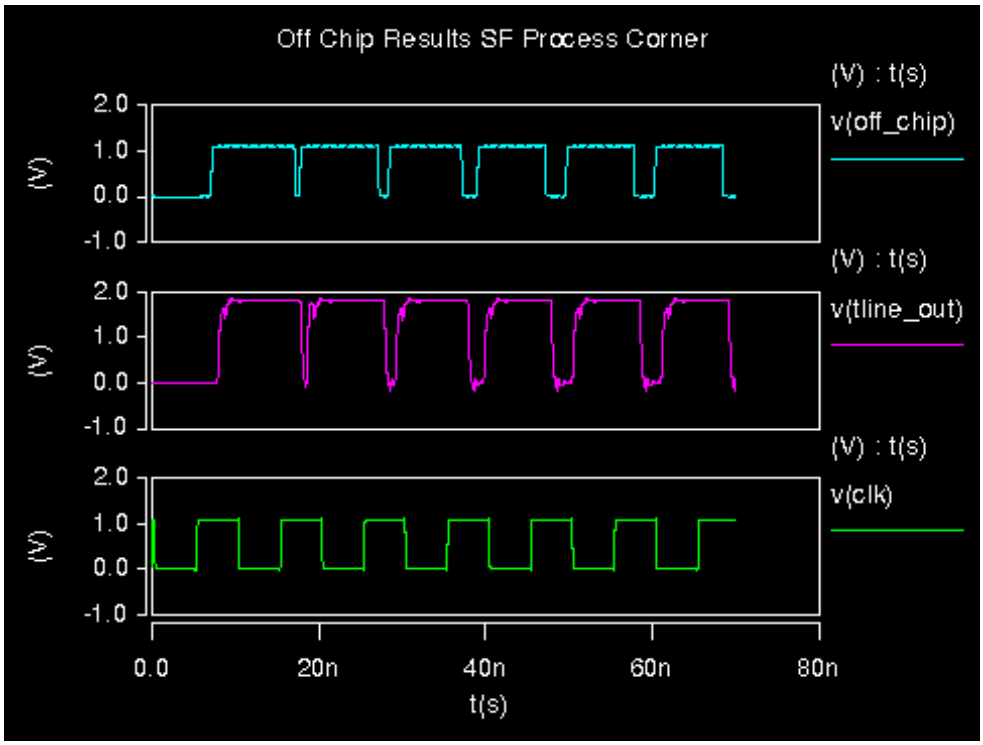


Figure 7: Off-chip results for the SF Process Corner.

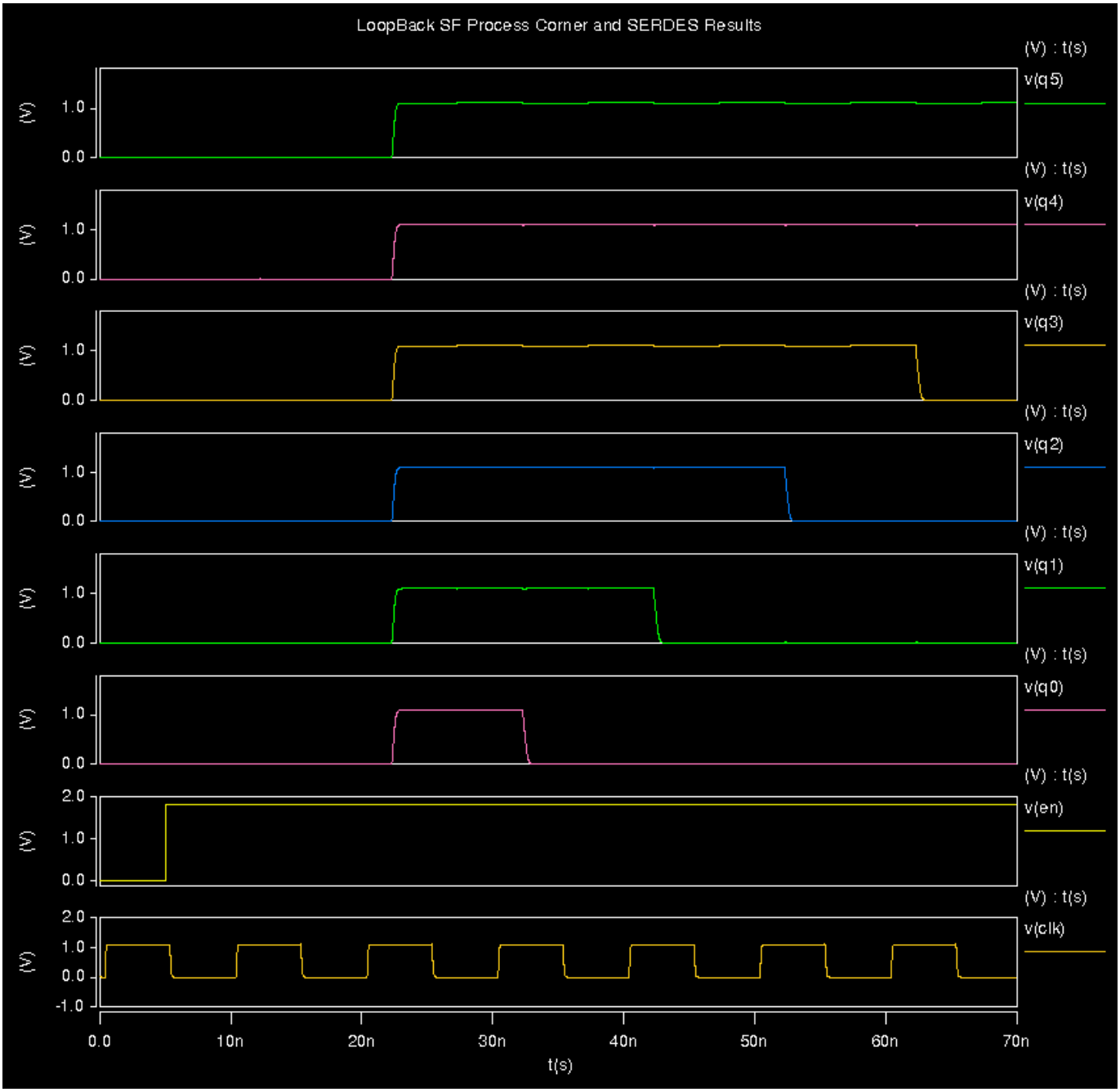


Figure 8: SERDES results for the SF Process Corner.

SS Results:

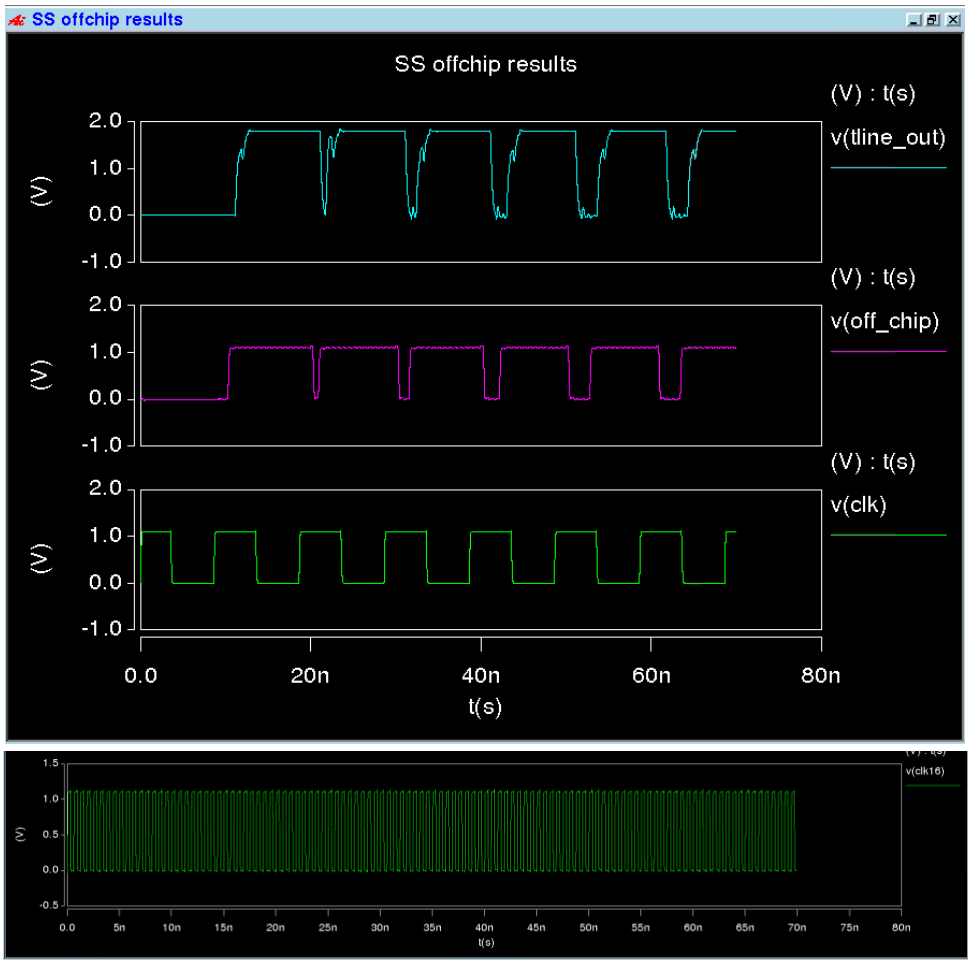
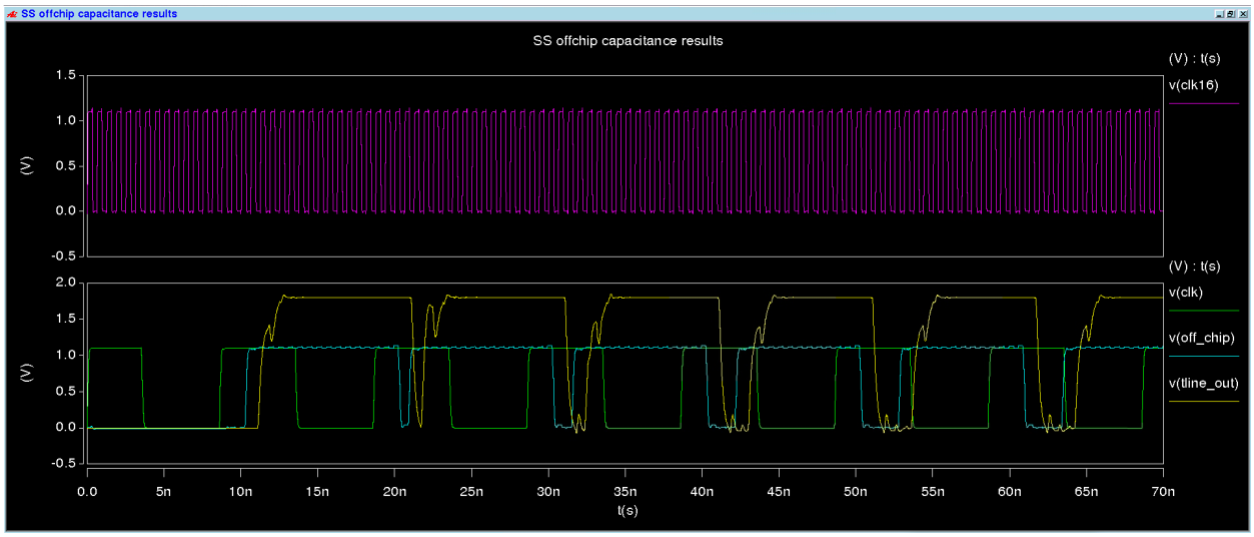
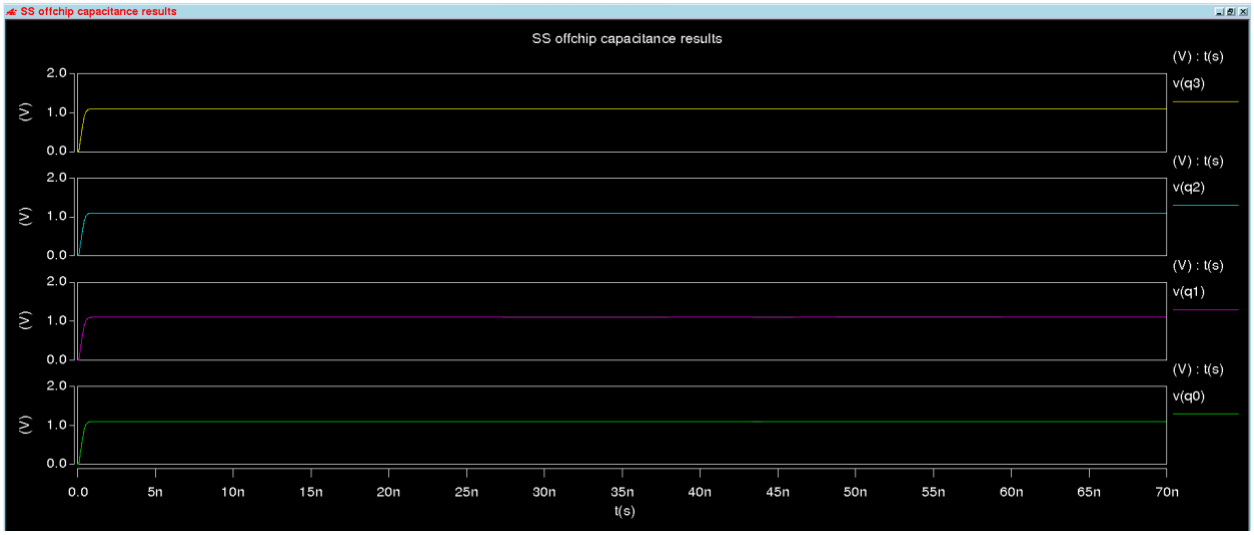
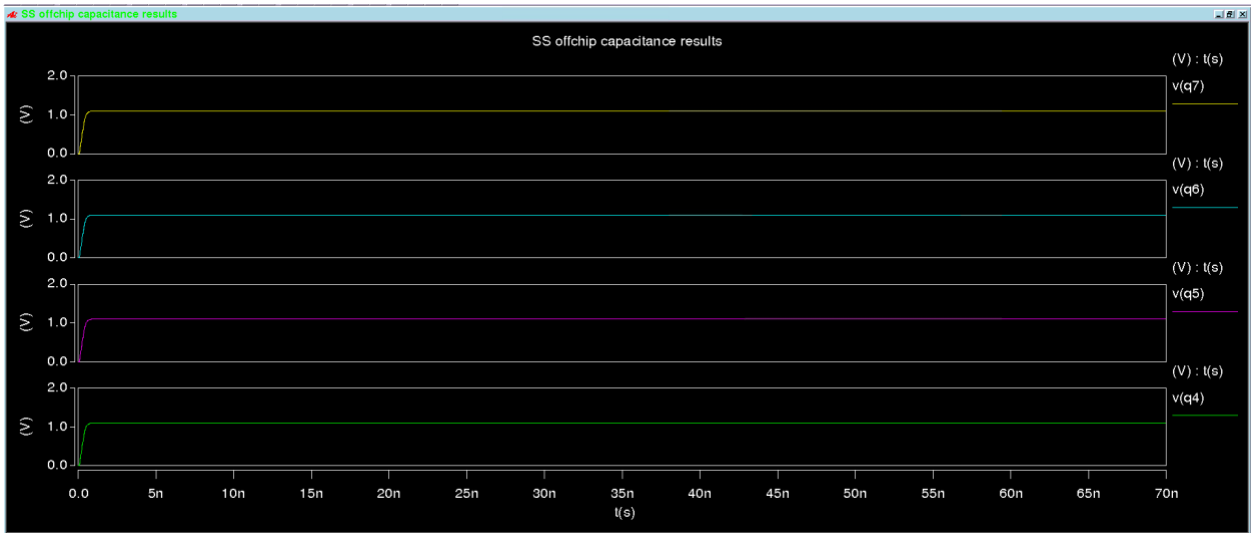
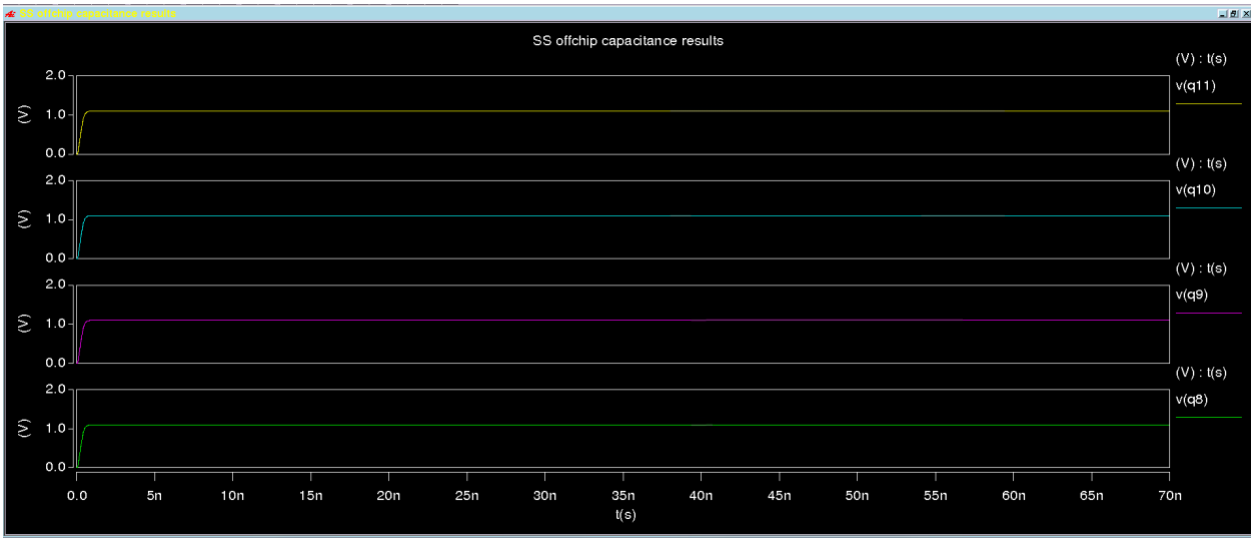


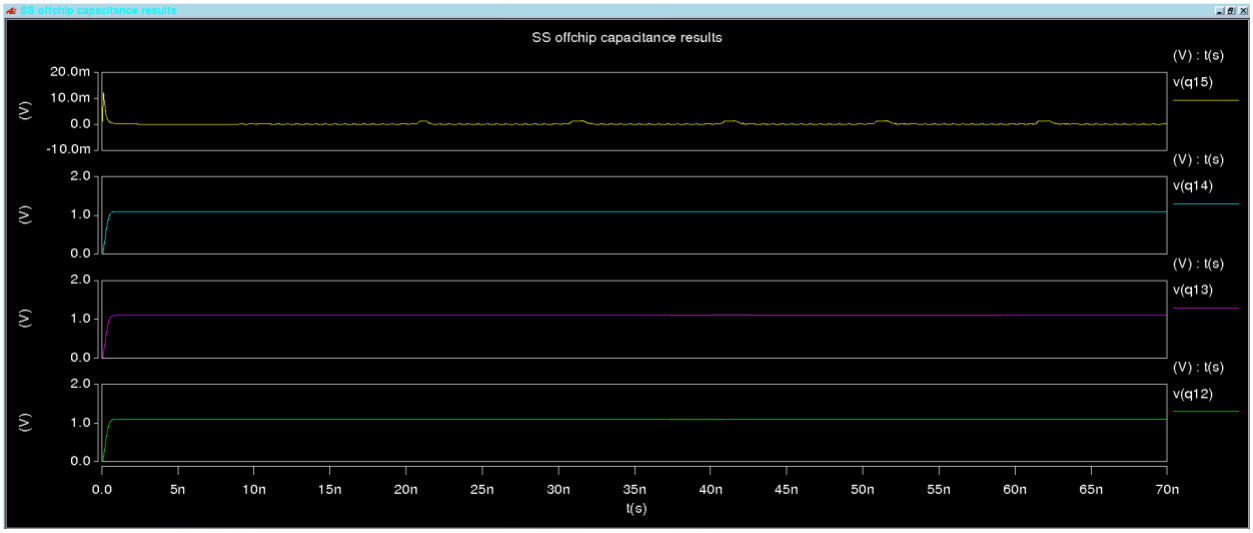
Figure 9: Off-chip results for the SS Process Corner. Shown on the bottom is the CLK16 signal.


SS offchip capacitance results
SS offchip capacitance results
(V) : t(s)
v(clk16)
(V) : t(s)
v(clk)
v(off_chip)
v(tline_out)
(V)
1.5
1.0
0.5
0.0
-0.5
2.0
0.0
5n
10n
15n
20n
25n
30n
35n
40n
45n
50n
55n
60n
65n
70n
t(s)


SS offchip capacitance results
SS offchip capacitance results
(V) : t(s)
v(q3)
v(q2)
v(q1)
v(q0)
(V)
2.0
1.0
0.0
0.0
5n
10n
15n
20n
25n
30n
35n
40n
45n
50n
55n
60n
65n
70n
t(s)


SS offchip capacitance results
SS offchip capacitance results
(V) : t(s)
v(q7)
v(q6)
v(q5)
v(q4)
(V)
2.0
1.0
0.0
0.0
5n
10n
15n
20n
25n
30n
35n
40n
45n
50n
55n
60n
65n
70n
t(s)


SS offchip capacitance results
SS offchip capacitance results
(V) : t(s)
v(q11)
(V) : t(s)
v(q10)
(V) : t(s)
v(q9)
(V) : t(s)
v(q8)
(V)
2.0
1.0
0.0
0.0
5n
10n
15n
20n
25n
30n
35n
40n
45n
50n
55n
60n
65n
70n
t(s)


SS offchip capacitance results
SS offchip capacitance results
(V) : t(s)
v(q15)
(V) : t(s)
v(q14)
(V) : t(s)
v(q13)
(V) : t(s)
v(q12)
(V)
20.0m
10.0m
0.0
-10.0m
2.0
1.0
0.0
0.0
5n
10n
15n
20n
25n
30n
35n
40n
45n
50n
55n
60n
65n
70n
t(s)

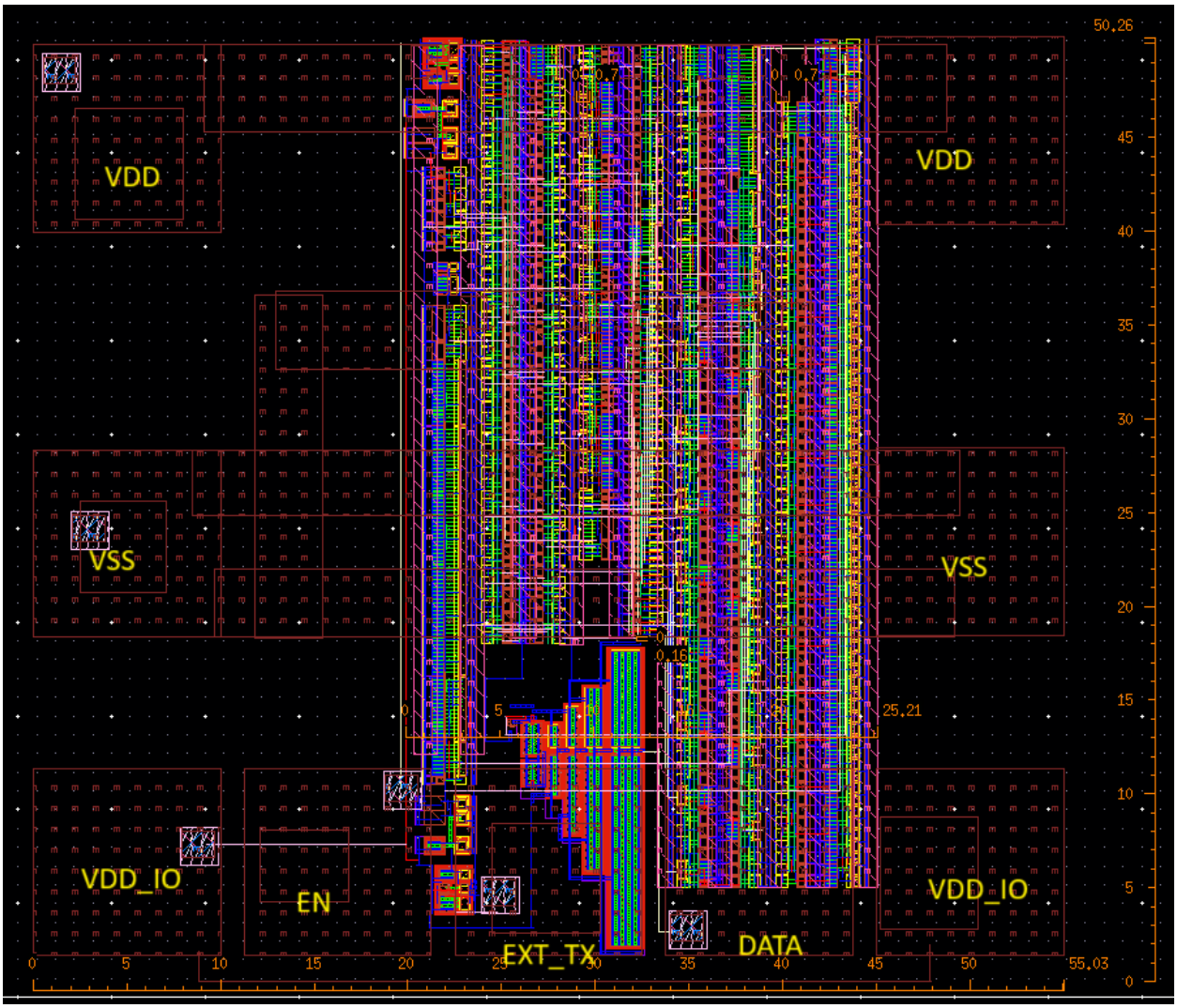
5. Discussion of Results

As seen from the simulated plots above, the results were as expected for all the process corners. For the loopback path, the outputs matched the correct prbs outputs from the manual analysis. Since we are only running for 4 fclk cycles, we can see that there is a 0 signal passing through q0 through q3 while q4 through q15 remains high for the entire period. For the off chip, we see that it reflects the PRBS manual analysis in a serial format.

To better minimize power consumption, we implemented gated clocks such that when EN is low the clock is turned off which turns off the prbs and serdes. This is accomplished by using AND gate with the CLK and EN signal. Similar approach was used to gate the deserializer. Since the deserializer is used only when EXT_TX is 0 (loopback mode), we gated the CLK into the deserializer with the inverted EXT_TX signal. This helped us further save power since only the active components are turned on.

Our layout is not super optimized, and could have been made more square rather than its current tall and thin status. But it still passes LVS and produces the expected results as hoped.

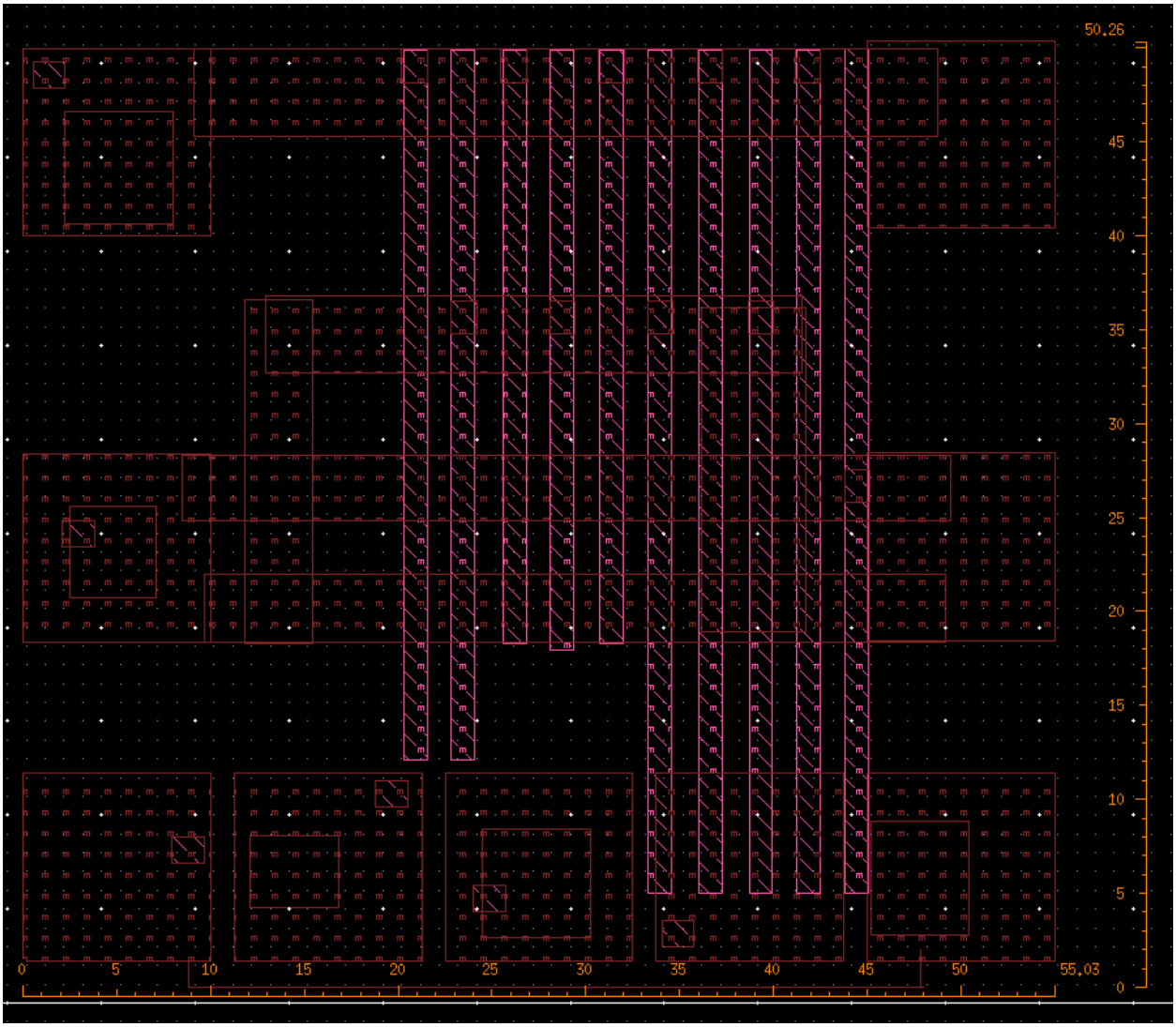
6. Layout



Top Level View of the Layout.

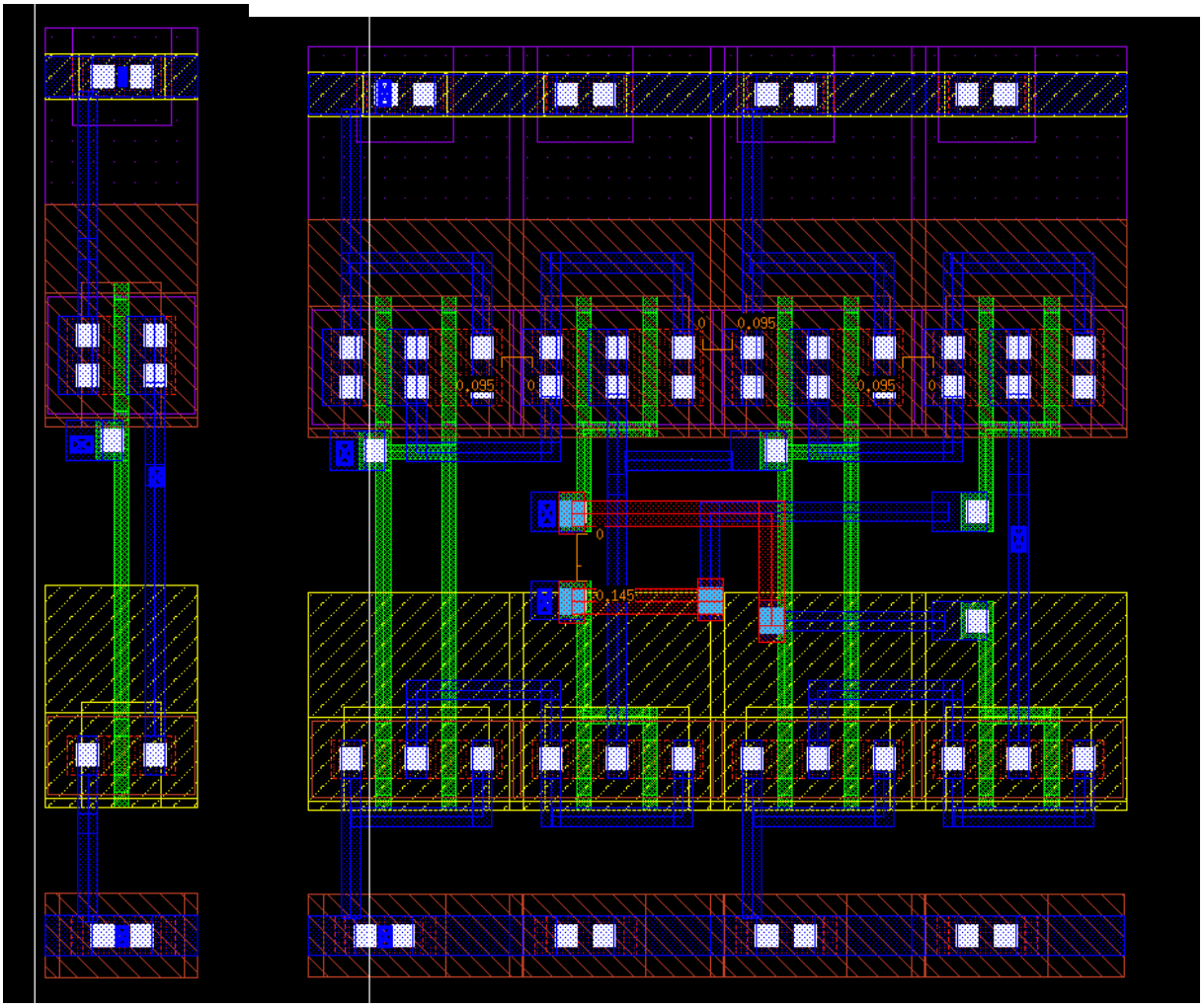
The entire layout is exactly 55.03 um by 50.26. Note that most of this is a result of the metal9 bond pads, the real width of the layout without the metal9 bond pads is 25.21 um by 50.26 um.

We begin with a discussion of the VDD and VSS layout strategy.



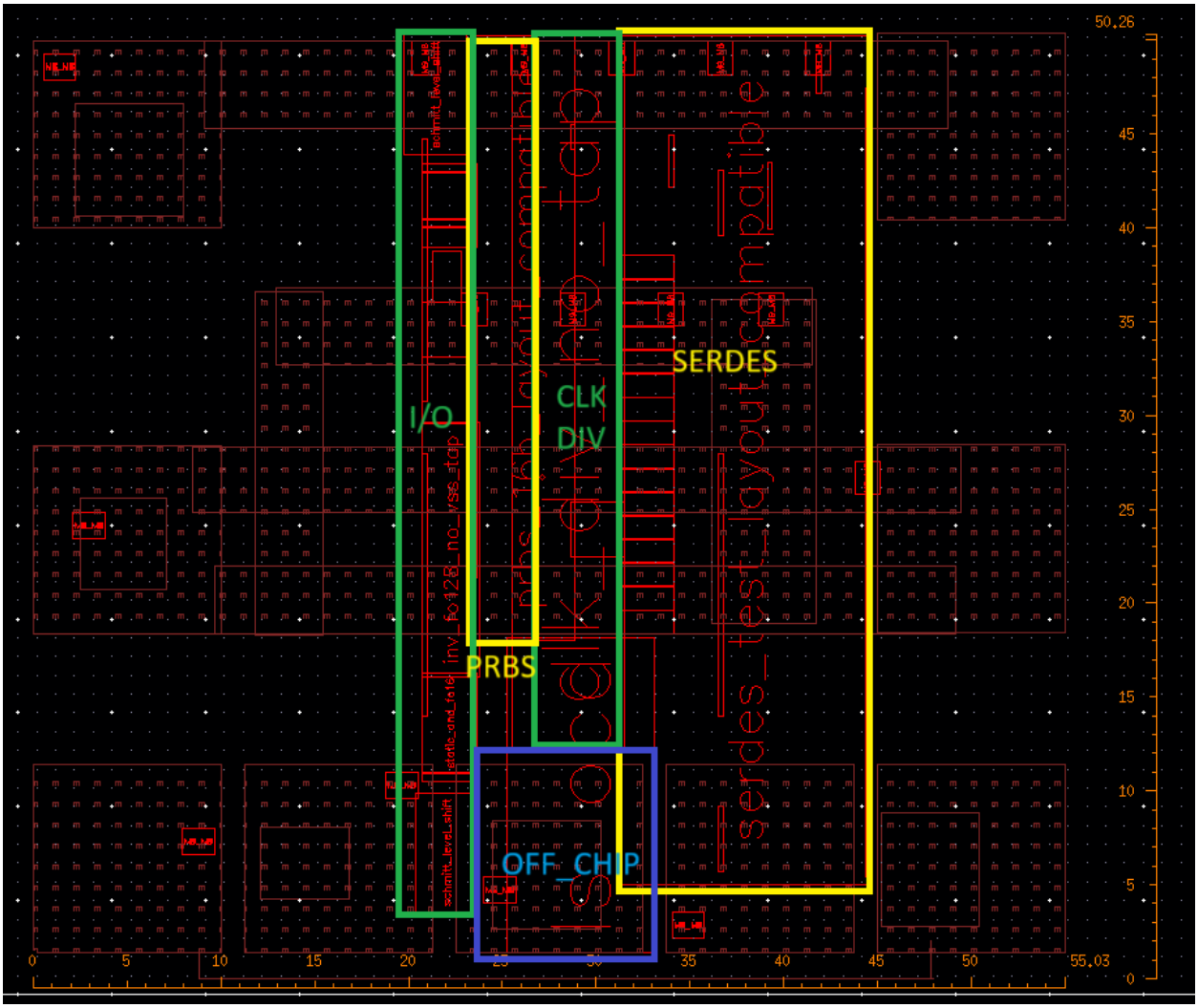
Layout only showing Metal 9 and Metal 8.

As seen, we have metal 9 running horizontally with very thick metal lines in order to prevent any issues with power draw. Metal 8 runs vertically, with alternating VDD or VSS. The strategy here was to utilize standardized stacked components, shown below. Wherever possible, we via from metal 8 down to metal 1 on these vertical lines.



Left: The classic inverter layed out.
Right: c2mos register layed out.

For each component, such as the inverter shown above, we ensure that the height of everything is identical, making it very easy for us to lay out more complicated circuits by simply placing the sub layouts next to each other, as can be seen in the register above. Vertical stacking is made simple as well, by overlapping the p-wells or n-wells in larger layouts. This greatly simplifies the layout process, and assists greatly when debugging even though it is not optimized for area.



Layout Subcomponents.

The layout has five major sections: I/O, PRBS, CLK DIV, SERDES, and OFF_CHIP.

1. I/O includes 1.8 to 1.1 level shifters for the EN and EXT_TX inputs, 1600 MHz clock buffer, and clock gating logic.
2. PRBS consists of the PRBS generator.
3. CLK DIV consists of the clock divider that generates the 100 MHz clock.
4. SERDES consists of the serializer, deserializer, and any mux logic that goes in between the two.
5. OFF_CHIP includes the 1.1 to 1.8 level shifter and the off chip driver.

7. Workload

Larry focused his efforts on the Serdes, delving into the intricate details of its design and functionality. Candy took charge of the PRBS, ensuring its precision and reliability. Himalaya tackled the off-Chip aspects, contributing his expertise to areas beyond the immediate circuit. Meanwhile, Dev played a crucial role in developing the Clock divider and addressing miscellaneous system components. Each team member dedicated their skills to creating the schematic and layout for their respective systems.

As a team, we met consistently every week to work on the schematic and layout. In addition to weekly meetings, whenever members had time, we worked on aspects of the project asynchronously, to allow us to debug and help each other during our meeting times. The report, simulations, and integration of subcomponents into the final layout was shared equally amongst the team, with all of us working together at the same time. Simulations were split equally, with constant collaboration between the team sharing files and other project resources to save time and lean on each other's strengths and weaknesses.