ECE 482

Fall 2024

Final Project

Digital Image Comparator Integrated Circuit

Andrew Kim, Kelly Chang, Mahir Koseli, Yibai Wang

12/11/24

i. Description of Overall Design

The overall design approach for this project was structured around modularity, performance optimization, and a clear hierarchy to ensure efficient functionality and replicability. At the top level, the design was divided into six main subparts: Parallel-In Parallel-Out (PIPO) registers, clock divider, XOR with adder units, Parallel-In Serial-Out (PISO) register, level shifter, and off-chip driver. Each module was carefully designed and optimized for specific performance metrics, including power, area, and delay, with the overarching goal of achieving high-frequency circuit operation.

The design process began with creating schematics for each module, ensuring individual functionality, and then integrating these components at the top level to verify overall performance. Following this, we completed the layouts for each subpart, optimizing critical paths and minimizing parasitic to enhance the circuit's operating frequency. Because we modularized each part of the image comparator, we were able to meet high performance expectations both for each part of the complete design and each subcircuit. In order to ensure proper functionality, we added pseudo-intrinsic capacitance at each gate (2fF), and simulated an approximate load to be driven (10fF).

In the following sections, we provide a detailed description of each subpart, including schematic diagrams down to the logic gate level. We also justify architectural decisions and highlight optimizations made to improve performance metrics. Key features, such as specific logic gate choices and clever layout strategies, are explicitly outlined to ensure replicability by another ECE 482 student. Furthermore, the overall high-level diagram is provided in the project documentation and shown below.

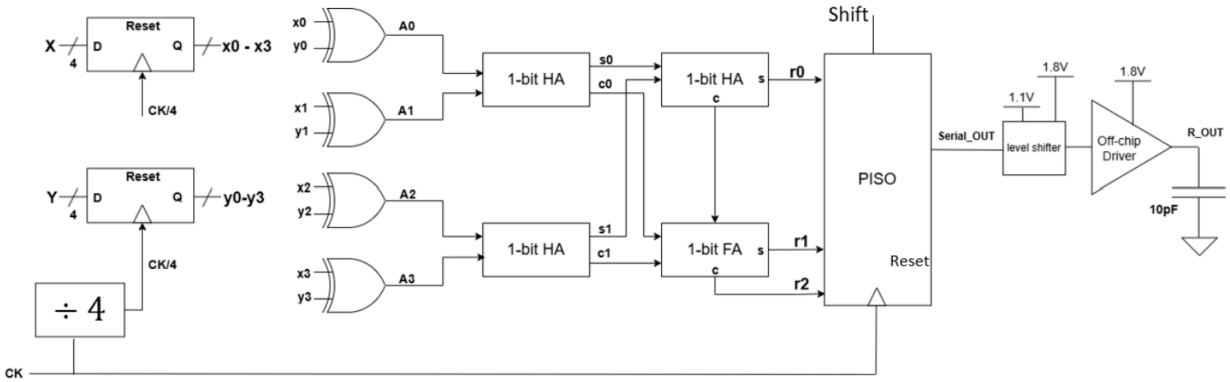

Figure 1. High-level block diagram of the overall design (taken from project docs)

PIPO

For PIPO we have 4 registers in parallel, where we latch the provided x and y inputs at the rising edge of the clk / 4 signal. The latched output will then be routed into the XOR gates to be used by the adders to compute R0-R2. Specifically, we used a positive-edge triggered register to catch the data inputs on the positive edge of the clock. This was to achieve full swing at the output node of the PIPO. Furthermore, to implement the reset logic for the register, we AND-ed the output of the register with reset. Furthermore, to account for clock skew, we designed a clock deskew circuit to ascertaing that the clock signals that would trigger the PIPO would arrive at the

same time. To account for the impact of parasitic capacitances, we increased the PIPO register sizings to 4*Wmin.

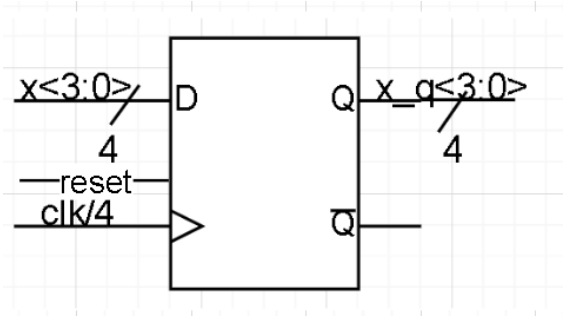

Figure 2. 4-Bit Register (PIPO)

Clock Divider

For the clock divider, we use 2 registers (without the reset) chained together, and each feeds the inverted output back into the input, which effectively halves the clock frequency with each iteration. Additional super-buffer with inverter size S=1 and S=2 is connected to the output to further minimize the delay. Following is the overall high-level diagram for the clock divider.

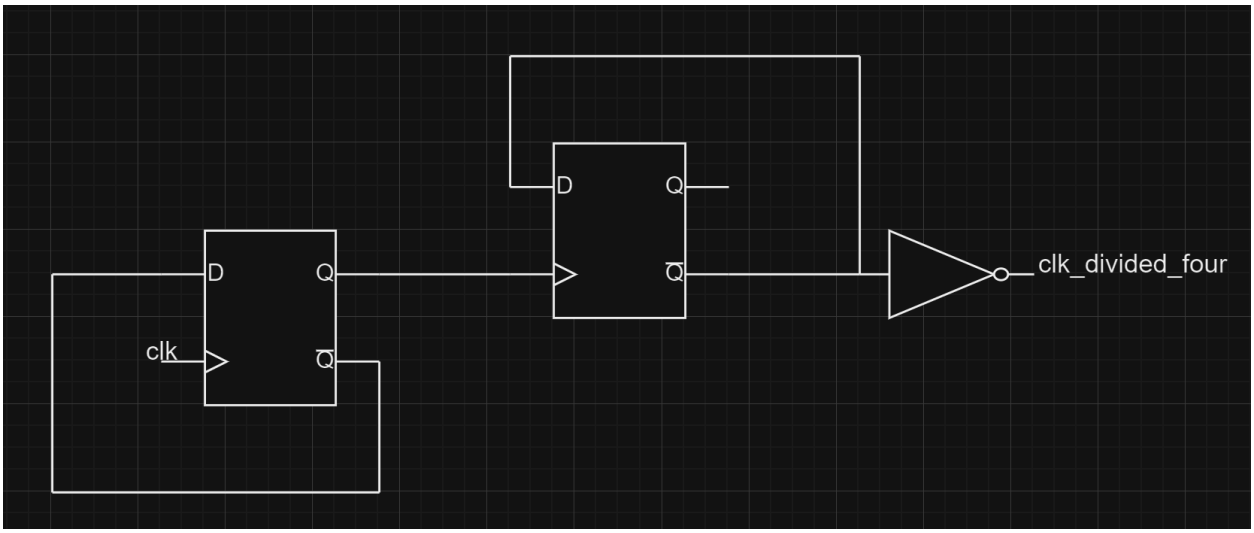

Figure 3. Clock divider circuit

XOR with Adders

For the XOR gates, we used a design from a research article found online. The design consists of three NMOSs and PMOSs, including the inverter connected at the output. This design is similar to the pass transistor XOR design but uses more transistors and adds an inverter at the end to restore the output signal to full swing, meaning there is static power dissipation. We used this design because the time delay was very short, output was full swing, and saved the number of transistors used compared to CMOS XOR. The same XOR design was used for the half adder

design, combining it with a CMOS AND gate. The AND gate was a CMOS NAND combined with an inverter. For the full adder, a mirror full adder design was used, for it greatly reduced the time delay.

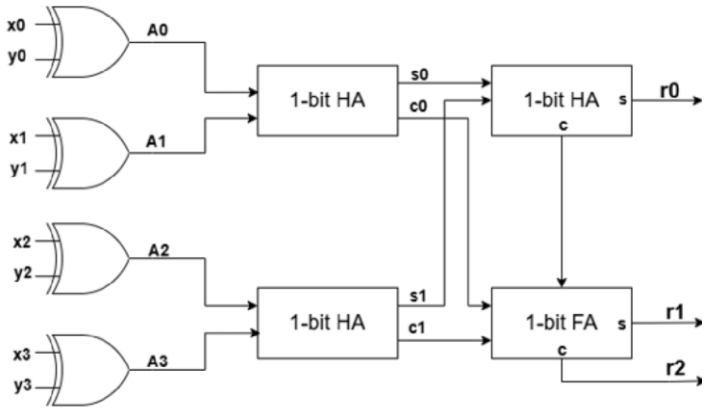

Figure 4. Block design for XOR, half adders, and full adder

PISO

For the design of our PISO (parallel-in-serial-out) we utilize 2:1 MUXes and positive-edge triggered D-flip-flops. PISO is practically 3 registers connected in series with MUXes in between to select the state of our registers, either shift or load state. In the shift state the MUXes select the output of the previous register, and logical 0 in the case of the first register. In the load state the MUXes select R0, R1, and R2, outputs produced by the adders, to load into their respective registers. The shift-signal is produced in the clock divider subcircuit by NANDing the clk / 2_bar and clk /4_bar signals. The overall top level diagram for the PISO is provided in the project documents, and also shown below.

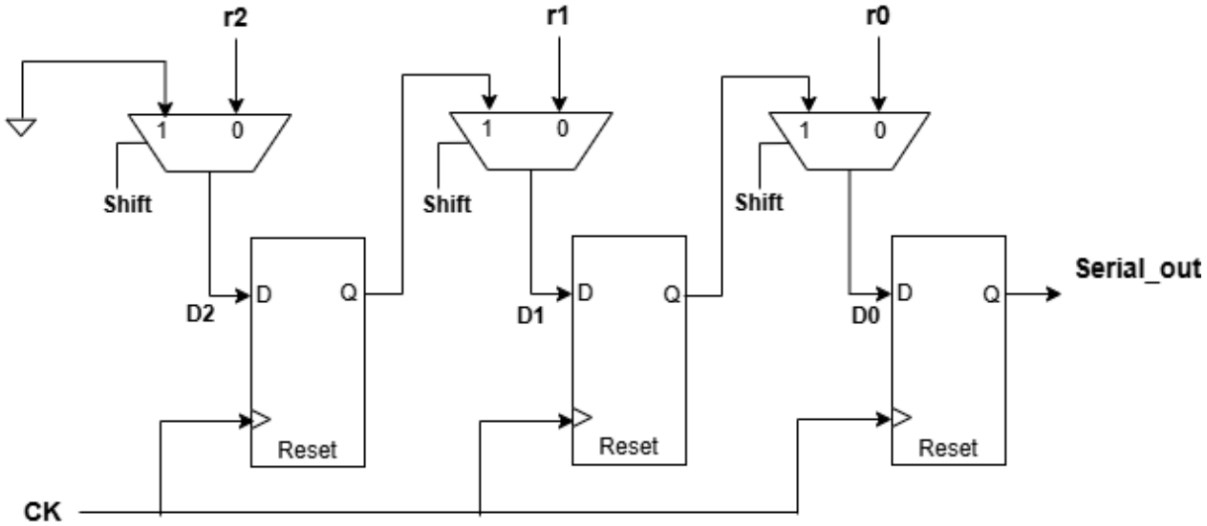

Figure 5. PISO overall diagram (taken from project documentation)

Level Shifters

When designing the level-shifter, we needed to utilize a mix of IO transistors, and core logic transistors. If one of transistors' 3 terminals (gate, source, and drain) is expected to have a voltage value of 1.8v we use IO transistors, and if we do not expect 1.8v in any of the terminals we use core logic transistors. For the design of the level-shifter we have taken a lot of inspiration from HW10. The following figure from HW10 summarizes our approach to the level-shifter design.

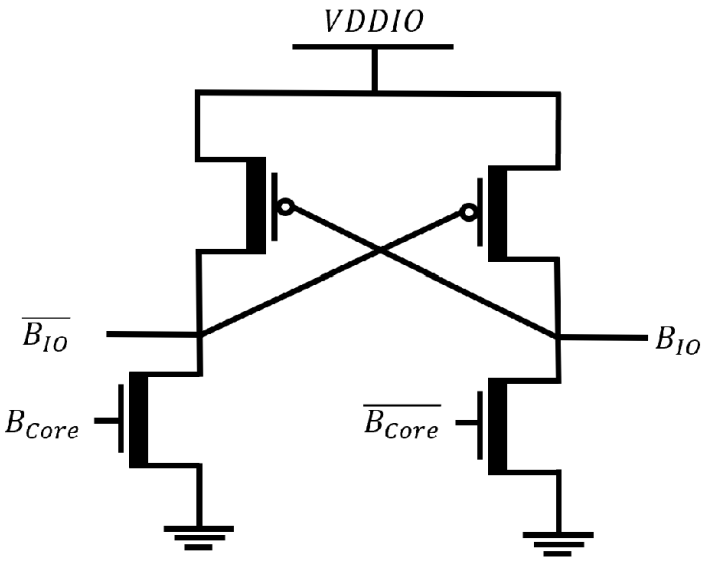

Figure 6. Rough schematic of the level-shifter (taken from HW10 solutions)

One significant difference in our approach to level-shifter design from HW10 was that the sizings for HW10 would not work for very high frequencies, so we had to increase the NMOS sizing. In order to figure out the most optimal NMOS sizing we have swept various values for the width of the NMOS and conclude that setting the width to be 3 times larger than W_{min} yields correct functionality with the least amount of delay.

Off-Chip Driver

When designing the off-chip driver, we utilized a super-buffer in order to optimize the delay of driving the 10pF load. A super-buffer is an even-numbered inverter chain, where each inverter is f times larger than the one before. To decide the optimal sizing for each inverter we first started off by finding f using the formula $f_{opt} = \exp(1 + f_{opt}/\gamma)$, which gives us $f_{opt} = 3.865$. Using this f_{opt} value we then find N_{opt} using the formula $N_{opt} = \ln(C_{out}/C_{g1}) / \ln(f_{opt})$. We should also mention that for our sizing of the first inverter we picked S=3, since the largest size of a MOS is limited to 1mm, this was done to ensure we wouldn't go over this limit when laying out the last inverter. Moving on, we get a value of N_{opt} to be a little over 6, which we round down since we can only have an integer value of inverters. Afterwards we calculate the value of f which is the actual

scaling factor used to size up each inverter. It is calculated as $f = (C_{out}/C_{g1})^{1/N}$ where N was found to be 6 earlier. We get 3.714 as the value of f which is the scaling factor for each subsequent inverter starting off with the first inverter that is sized at S=3. Using all of this knowledge we get the following top-level diagram, also note that our design utilizes $\beta = 1$.

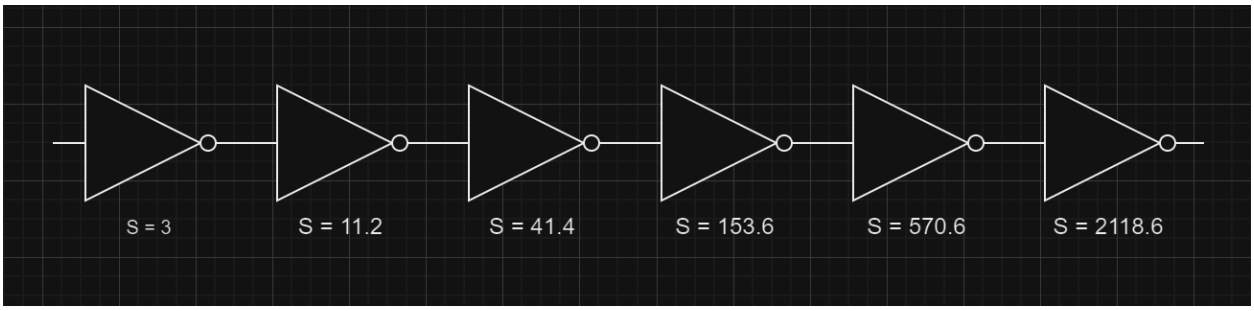

Figure 7. Super-buffer top level diagram with sizes

ii. Transistor-Level Rendition

Registers

We chose C2MOS(Clocked CMOS) register for our register design. It is advantageous due to its insensitivity to the clock overlaps, avoiding conflicts caused by simultaneous activation of both pull-up (PMOS) and pull-down (NMOS) networks and ensuring stable operation at high speeds. Additionally, it provides full swing at the output, ensuring reliable logic levels and reducing the possibility of signal degradation. Comparing to transmission-gate registers, the disadvantages are to ensure the stability of input data and more power consumption if the clock rise/fall times are slow; comparing to dynamic latches, it has slower operation speed and requires more transistors to operate, which eventually leads to larger layout area.

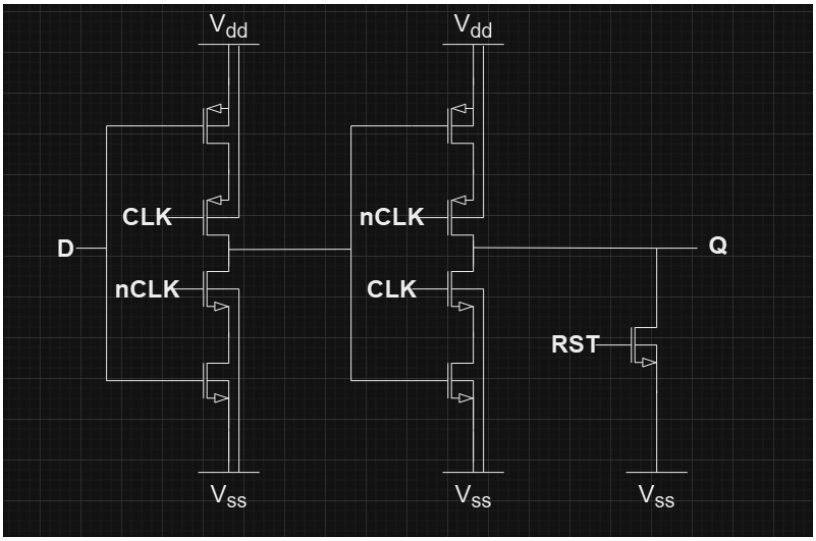

Figure 8. Register transistor level schematic

W_p	960nm
W_n	480nm

Table 1. Transistor sizes for register

XOR

The article *Investigation on Power, Delay and Area optimization of XOR Gate* by Thamizharasan V. and Ramya M. provides several XOR designs as well as their performance. We decided to use the "Proposed EX-OR" XOR gate design, which has the least power consumption, average number of transistors, and least area. This XOR design connects the sources of the NMOS drain to the inputs and connects the drain of the NMOS to the inverted output. The design uses the attribute of current being able to flow through either direction of a MOSFET, from drain to source or from source to drain. When the inputs A and B are the same (either VDD or VSS), the inverted output becomes VDD due to the pull-up from PMOS or NMOS, and the inverter changes output to VSS. When the two inputs are different, the source of one NMOS is VDD and source of another NMOS is VSS, so none of the transistors can pull-up, so the inverted output becomes VSS and inverter changes output to VDD. As for the sizing, we used minimal sizing to save space and reduce overall area.

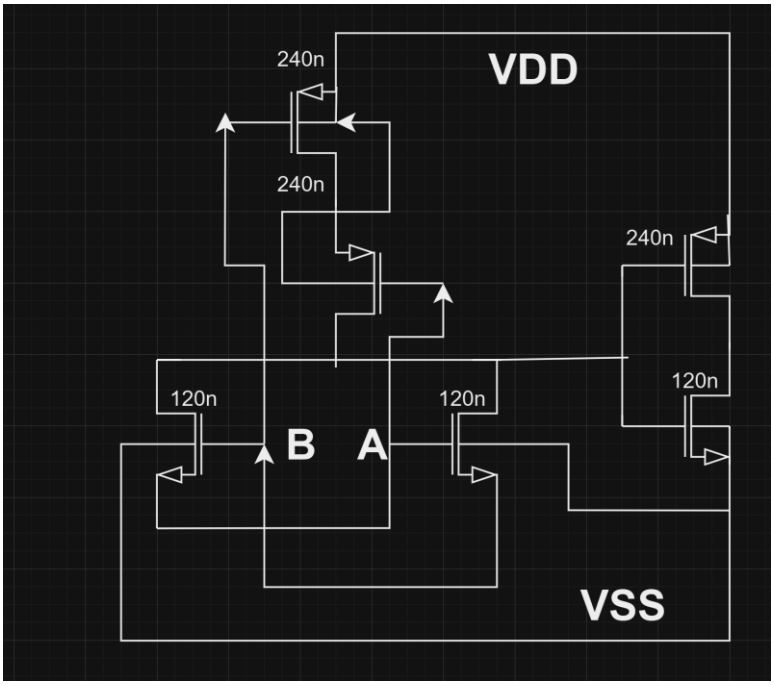

Figure 9. XOR transistor level schematic

W_p	240nm
W_n	120nm
L	45nm

Table 2. Transistor sizes for XOR

Full-Adder

We used a mirror adder full adder design from lecture. As for sizing, we found a sizing guide from lecture slides of University of Michigan but divided the sizing for all left-side transistors by 2 and divided all right-side transistors by 3. We decided with this approach because the time delay difference was not critical when using smaller sized transistors, since the mirror adder design produced fast enough results. The rightmost side transistors also made no difference in time delay when dividing the size by 3. We preferred to use smaller sized transistors to save area and make layout easier. Minimum sizing with beta=2 was used for the inverters to invert the output, and this was possible because most of the transistors in the mirror adder used similar sizing, except for the leftmost side. In the end, the pull-up and pull-down strength in the mirror adder was small enough to allow the inverters to invert output signals.

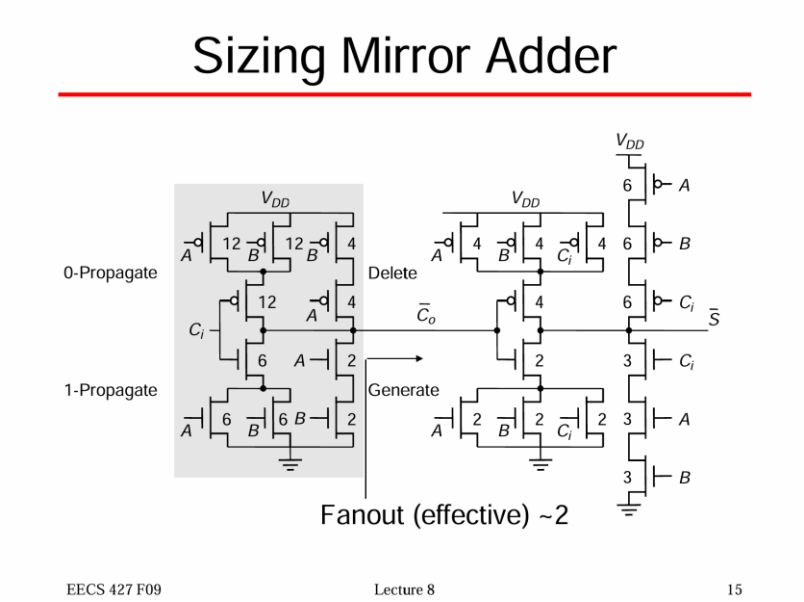

Figure 10. Sizing guide for mirror adder from University of Michigan EECS 427 lecture 8

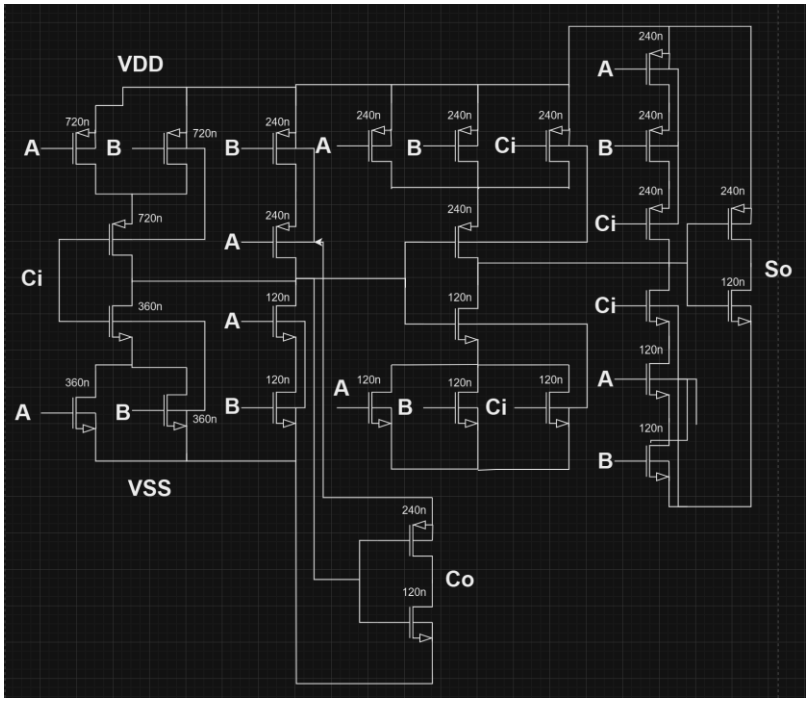

Figure 11. Mirror adder transistor level schematic

W_p (left side)	720nm
W_n (left side)	360nm
W_p (right side)	240nm
W_n (right side)	120nm
L	45nm

Table 3. Transistor sizes for mirror adder

MUX

For 2:1 MUX design we use 2 transmission gates, where depending on the select signal, if the select signal is logic level one we transmit signal in_one, and if it is logic level zero we transmit signal in_two. Also note that body contacts have been ommited for the sake of having a clean schematic, for PMOS the body should be connected to global ground, and for NMOS the body should be connected to VDD=1.1V.

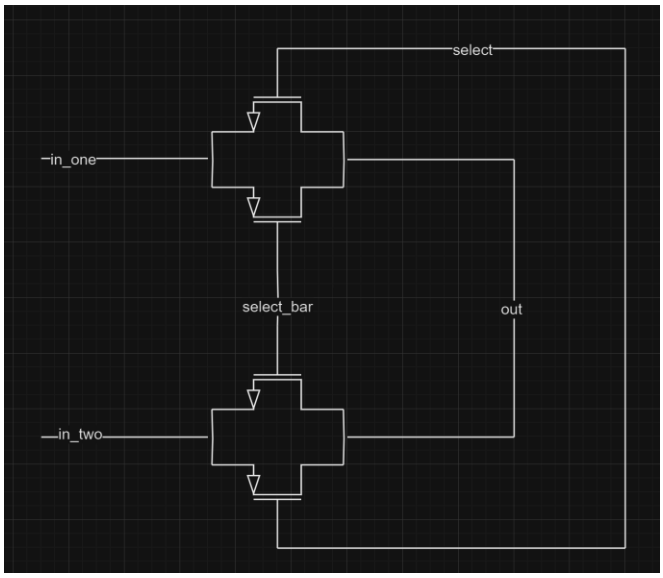

Figure 12. Mux transistor level rendition

W_N	480n
W_P	960n

Table 4. Transistor sizes for MUX

Level-Shifter

For the level-shifter design, we use 2 types of transistors, one of them operates at the IO logic level, and the other operates at the core logic level. We note that the NMOS transistors have their bodies connected to the ground, transistors labeled MP1 have their bodies connected to VDD=1.1V, and transistors labeled MP2 have their bodies connected to VDDIO=1.8V.

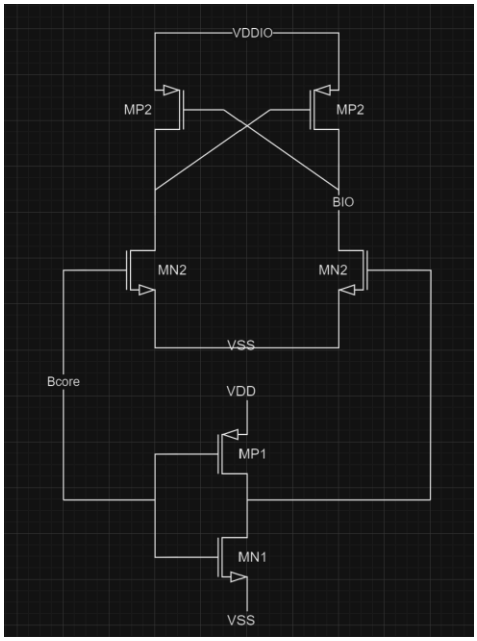

Figure 13. Level shifter transistor level rendition

W_{P1}	240n
W_{N1}	120n
W_{P2}	960n
W_{N2}	320n
L_1	45n
L_2	150n

Table 5. Transistor sizing for level-shifter

iii. Performance Table

1	Maximum f_{CK} for correct functionality at all process corners	7.752 GHz	2.4GHz
2	Average power drawn from VDD at MIN (f_{CK} listed in row 1, 2.4 GHz)	-530.483uW	-257.611uW
3	Core supply figure-of-merit: Power / frequency (same power and frequency as in row 2)	-6.843*10e-14	-1.073e-13
4	Average power drawn from VDDIO at MIN (f_{CK} listed in row 1, 2.4 GHz)	-6.4646mW	-7.4784mW
5	IO supply figure-of-merit: Power / frequency (same power and frequency as in row 4)	-8.339e-13	-3.116e-12
6	Area of Top-level layout (µmµm)	63.23 µm × 32.255 µm	63.23 µm × 32.255 µm
7	Latency (measured in number of CK cycles)	2	2

iv. Demonstration of Circuit Functionality

Used provided Project.cir netlist, called power.cir in our folder.

Bcore: PISO output

Q: r_out, output of superbuffer

Serial_in: output of level shifter

_net4: x3

_net0: y3

_net1: y2

_net5: x2

_net2: y1

_net6: x1

_net3: y0

_net7: x0

Testbench 1:

(a). TT

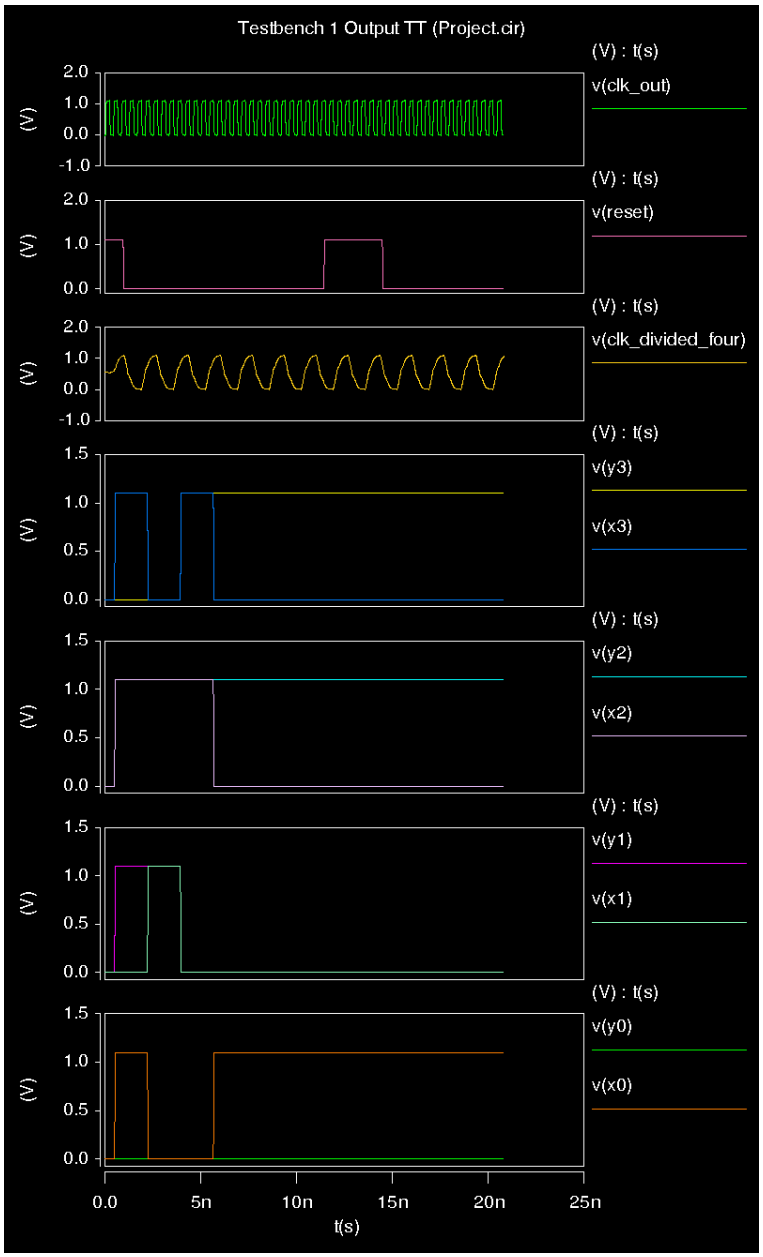

Figure 14. 2.4 GHz testbench, TT

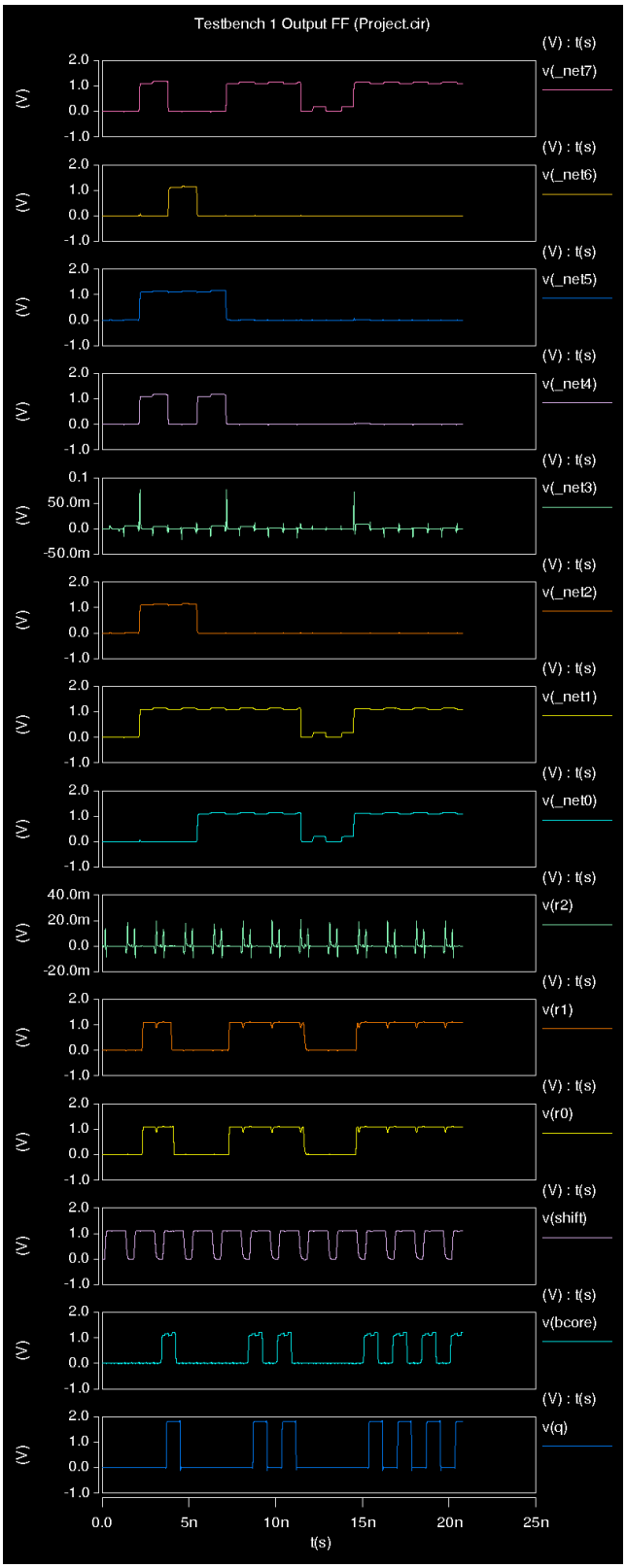

Figure 15. 2.4 GHz QRC testbench, TT

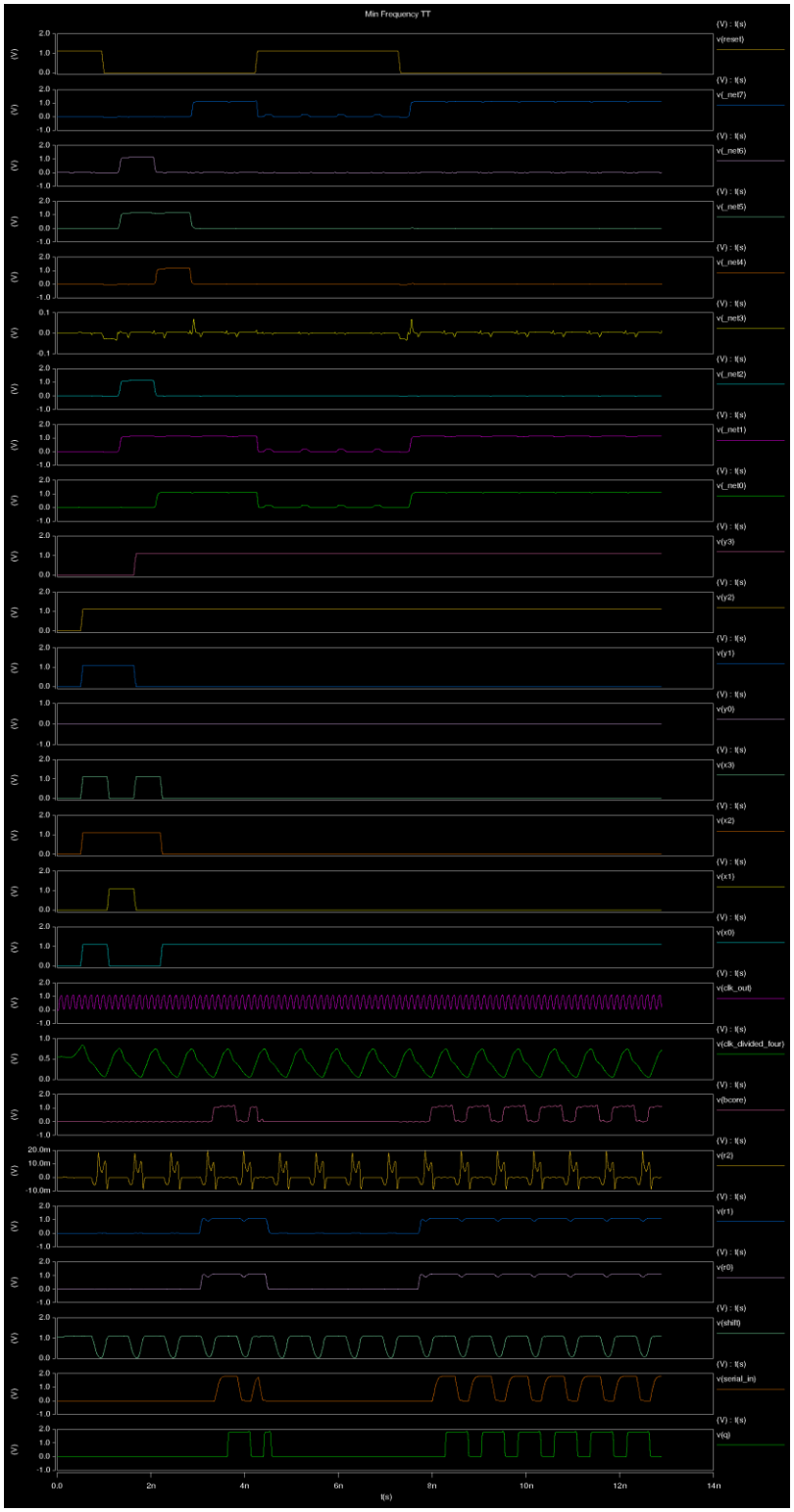

Figure 16. Minimum frequency QRC testbench, TT

(b). SS

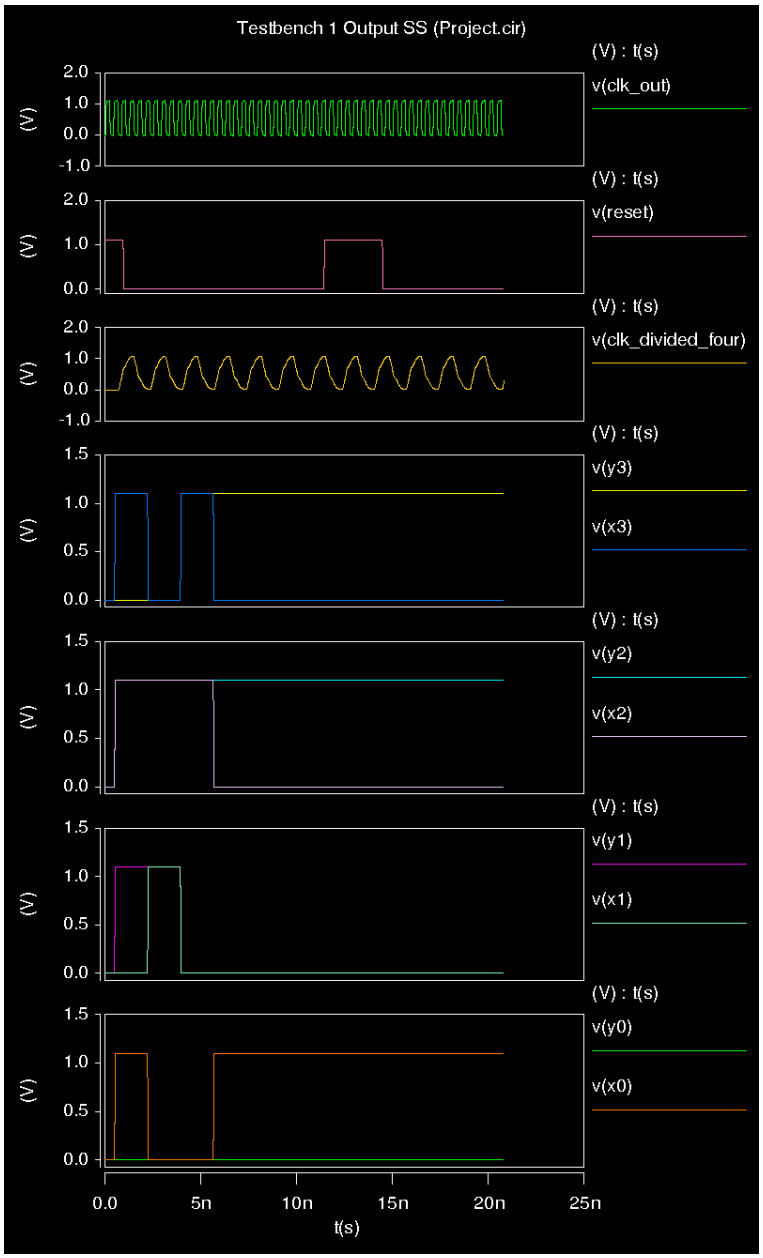

Figure 17. 2.4 GHz testbench, SS

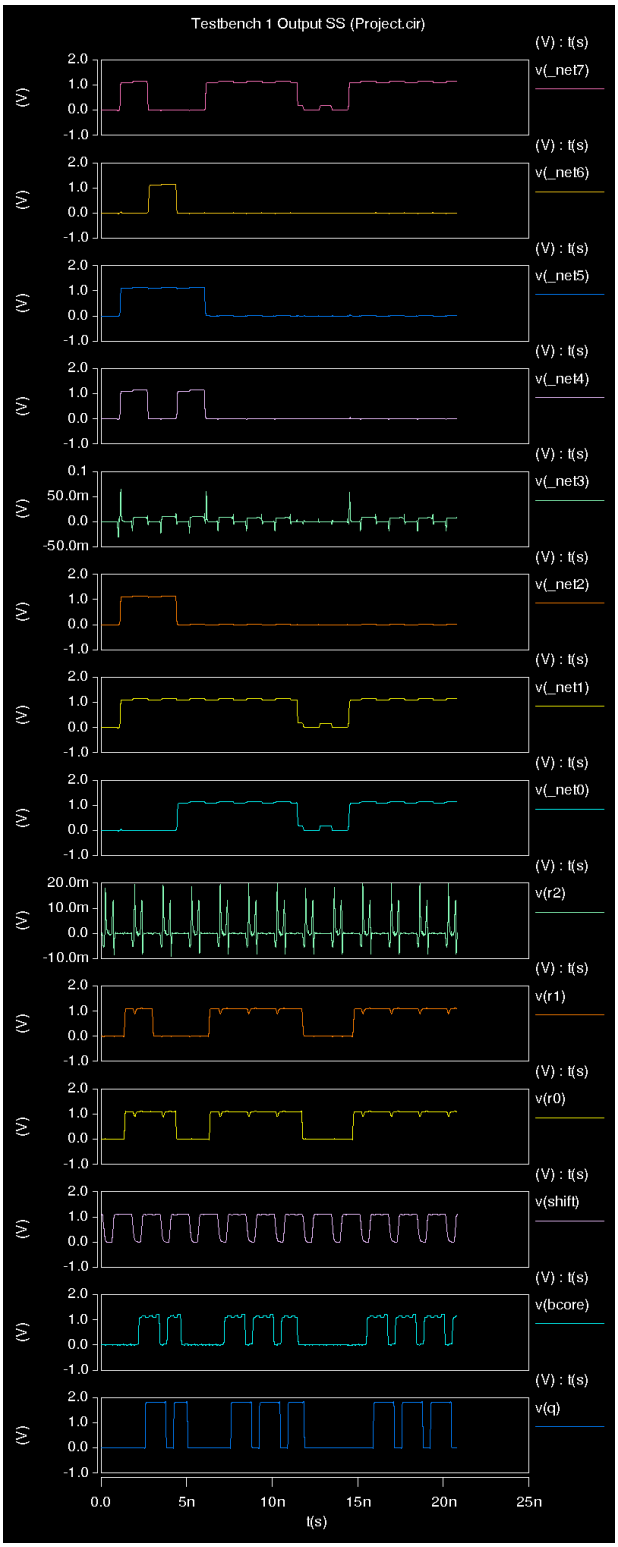

Figure 18. 2.4 GHz QRC testbench, SS

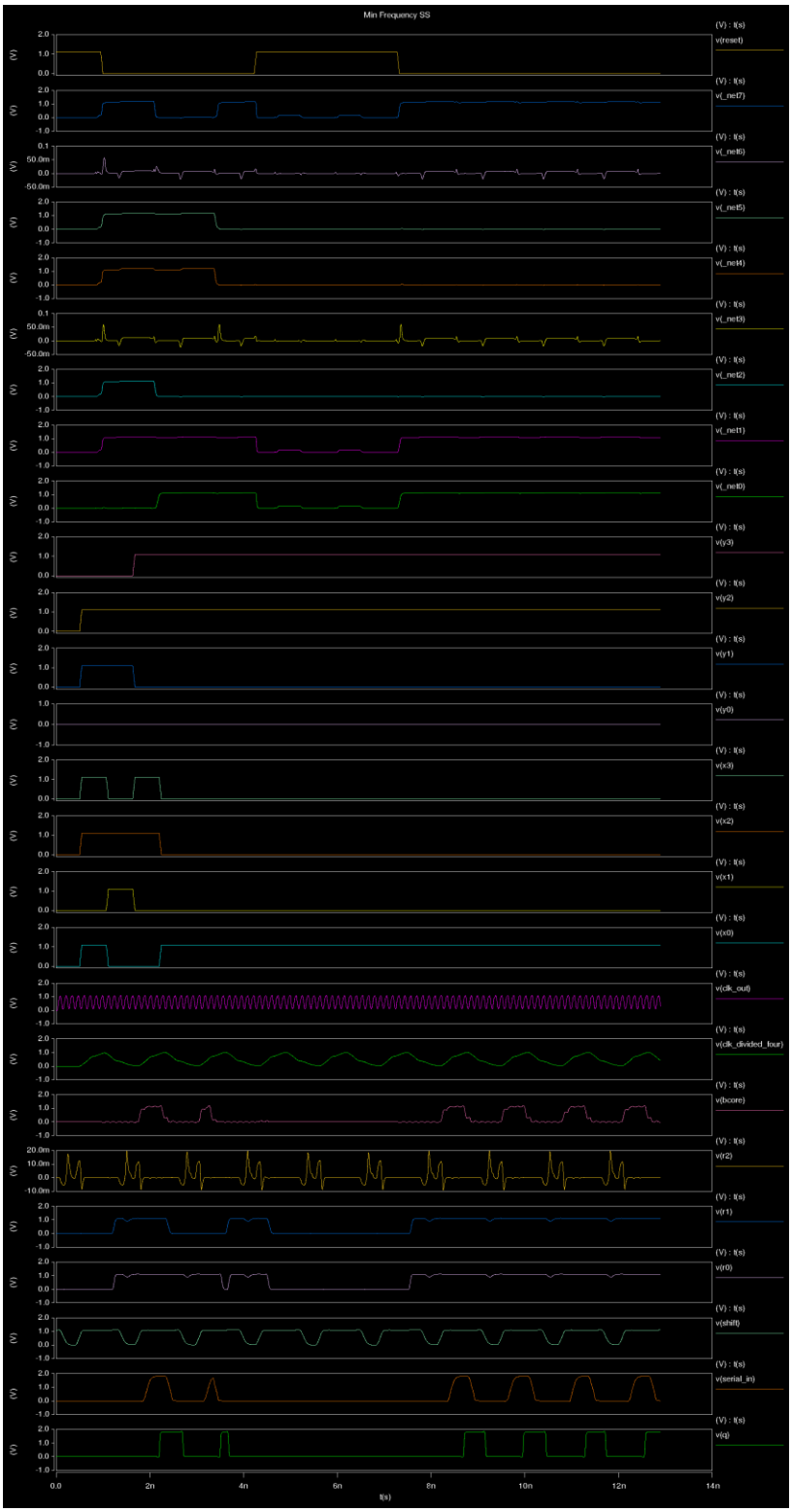

Figure 19. Minimum frequency QRC testbench, SS

(c). SF

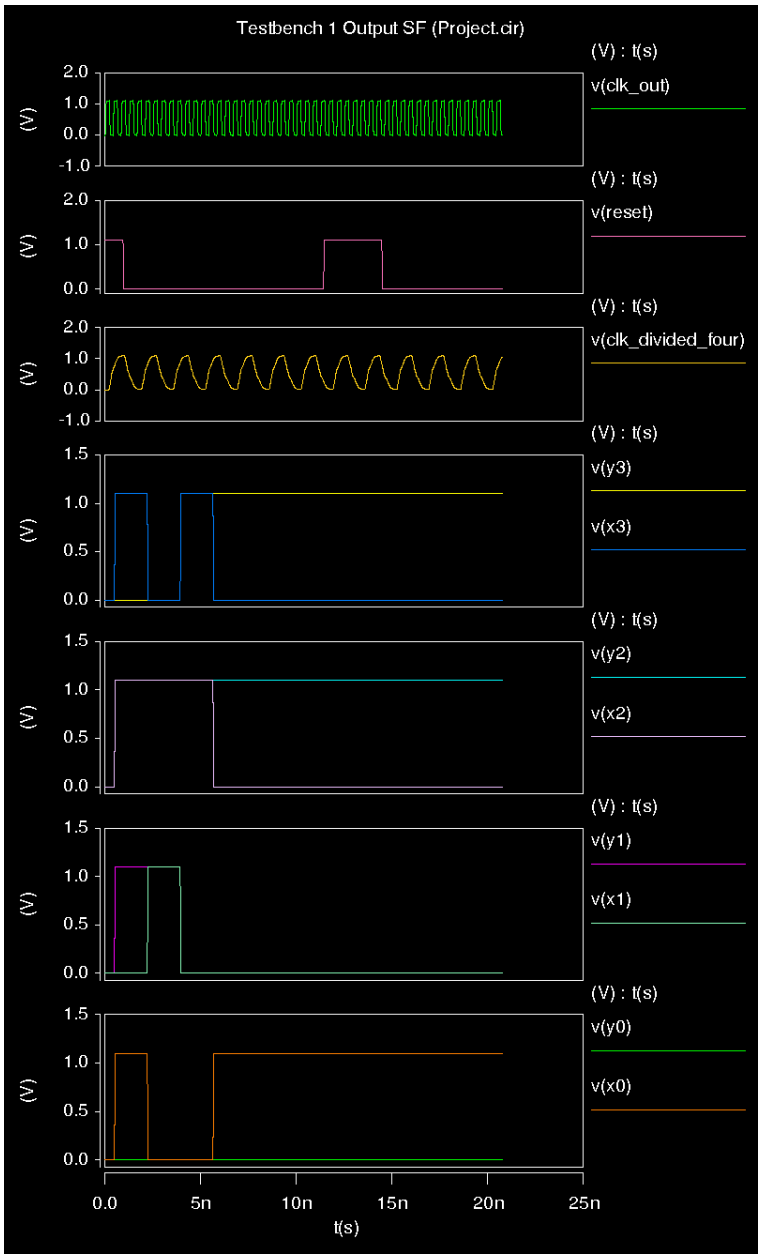

Figure 20. 2.4 GHz testbench, SF

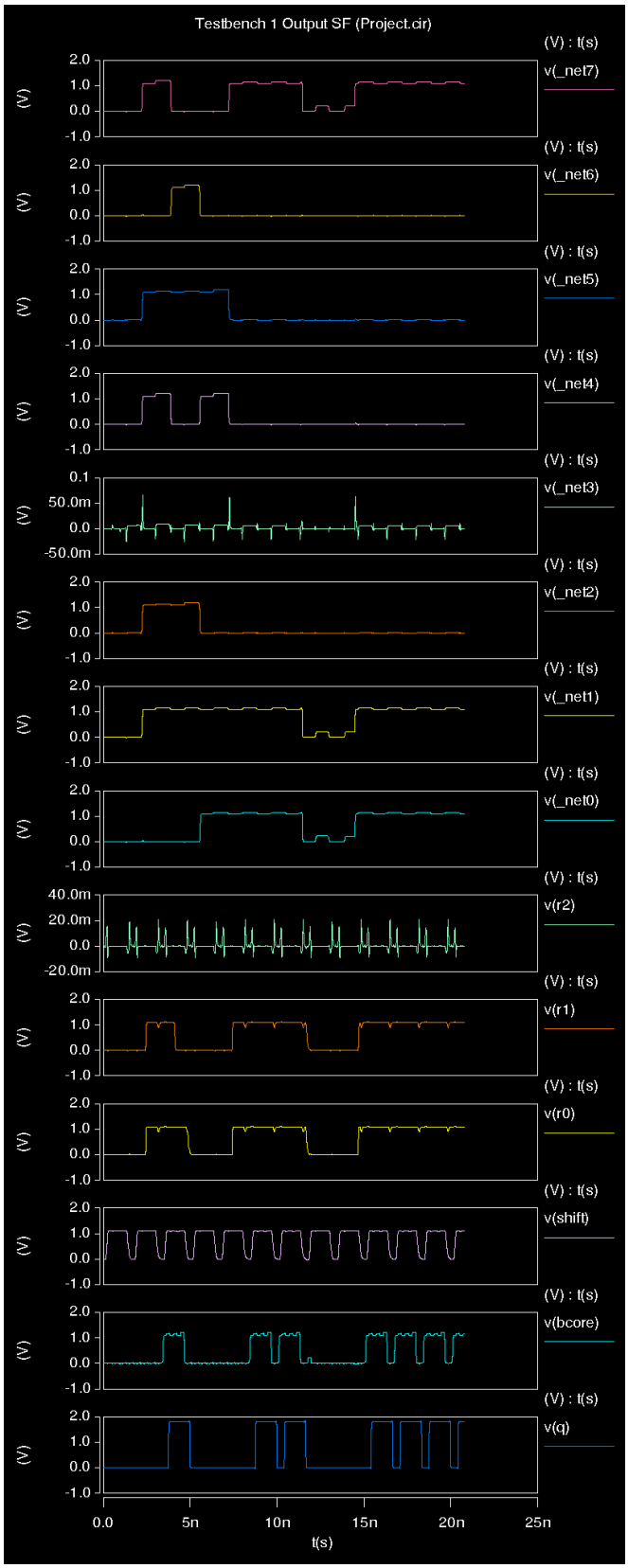

Figure 21. 2.4 GHz QRC testbench, SF

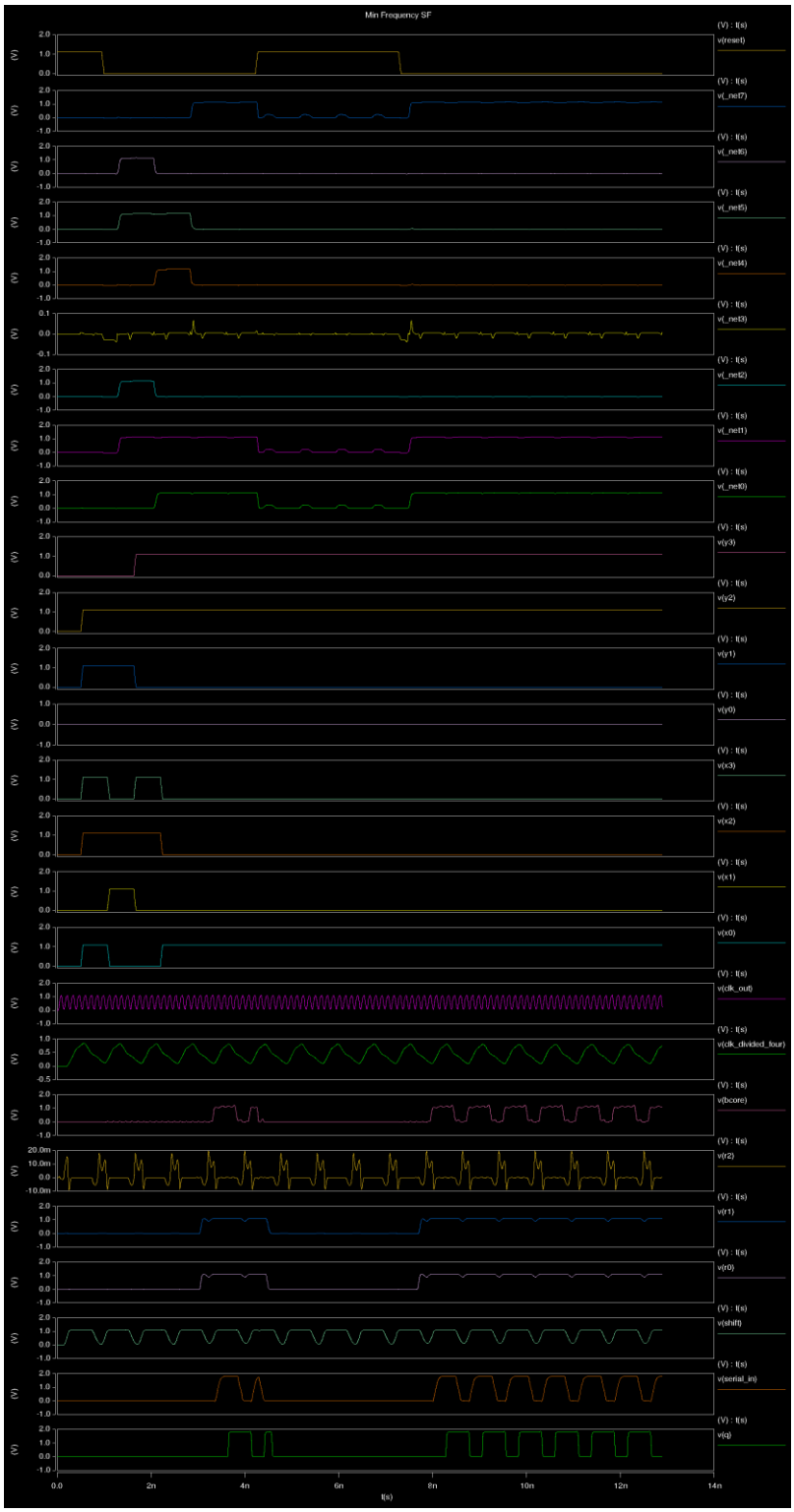

Figure 22. Minimum frequency QRC testbench, SF

(d). FS

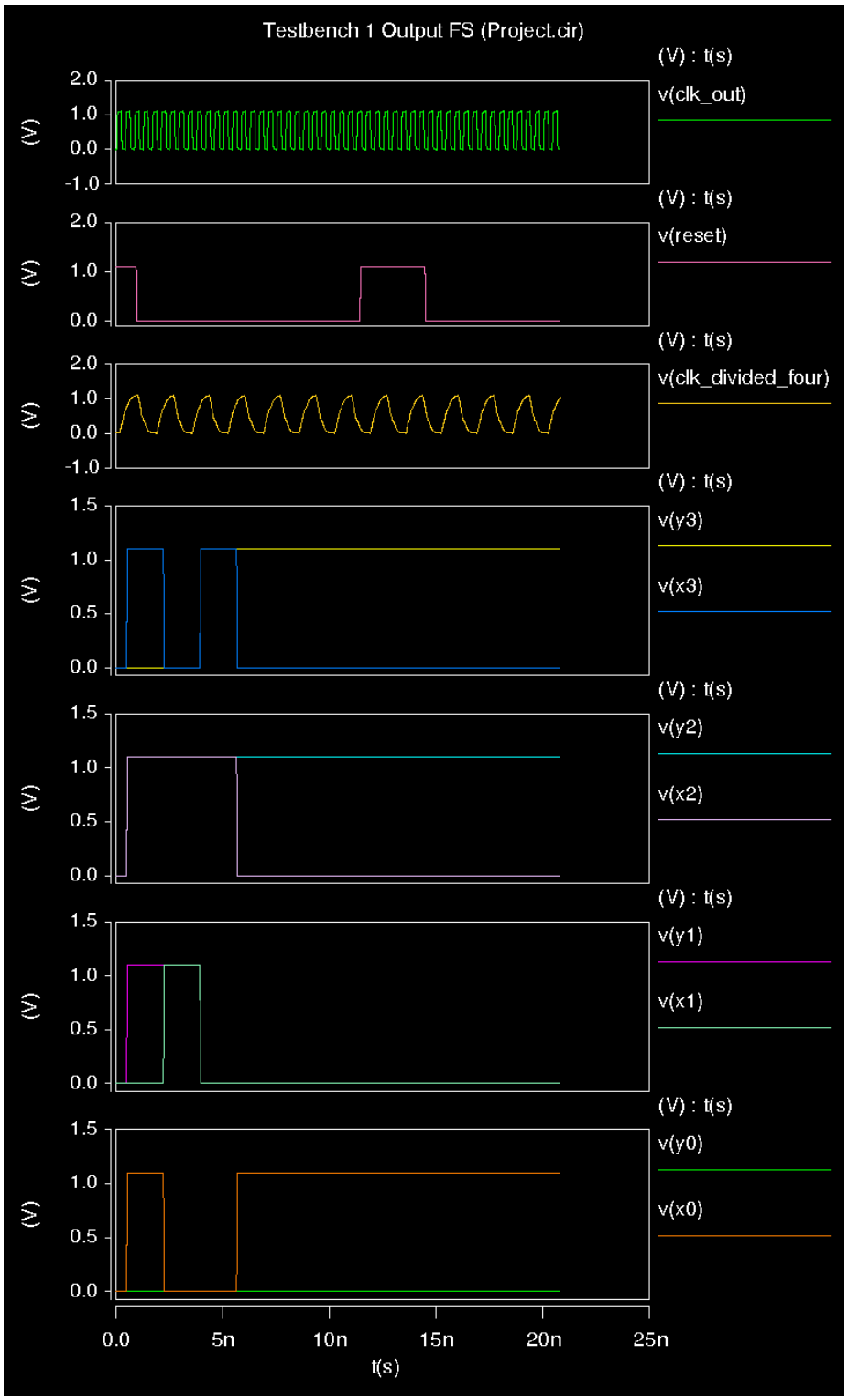

Figure 23. 2.4 GHz testbench, FS

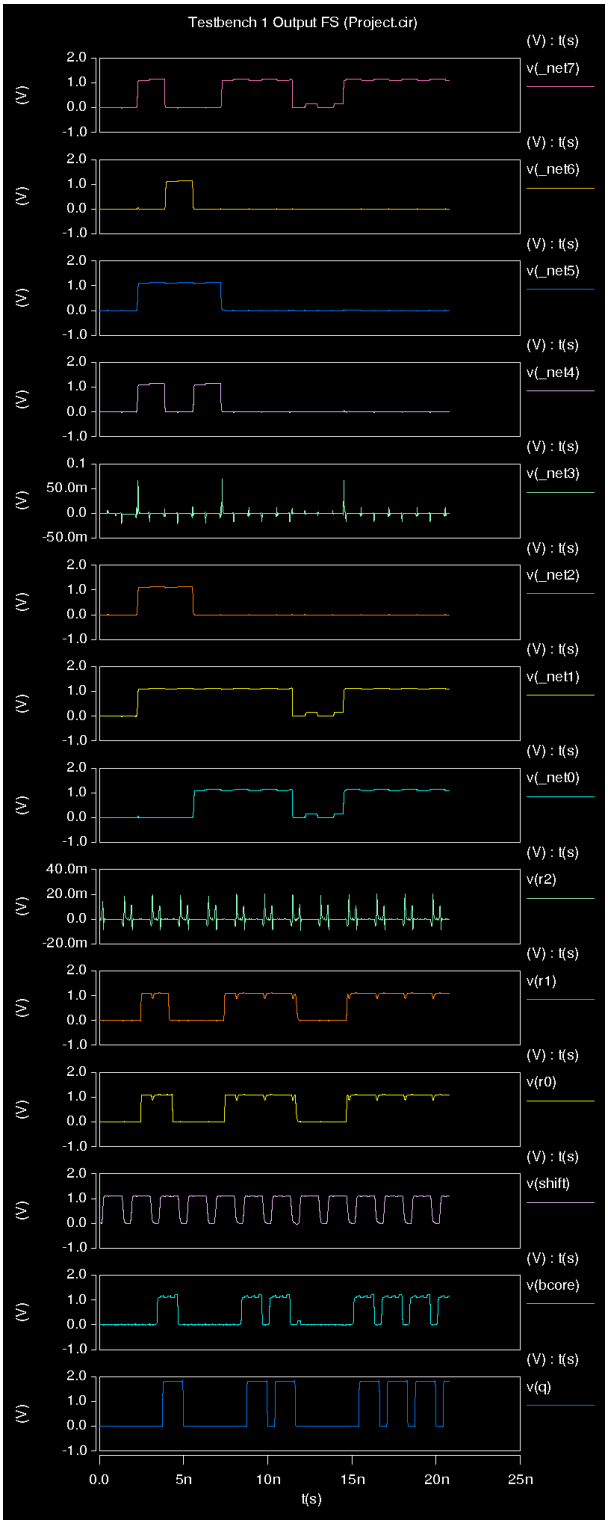

Figure 24. 2.4 GHz QRC testbench, FS

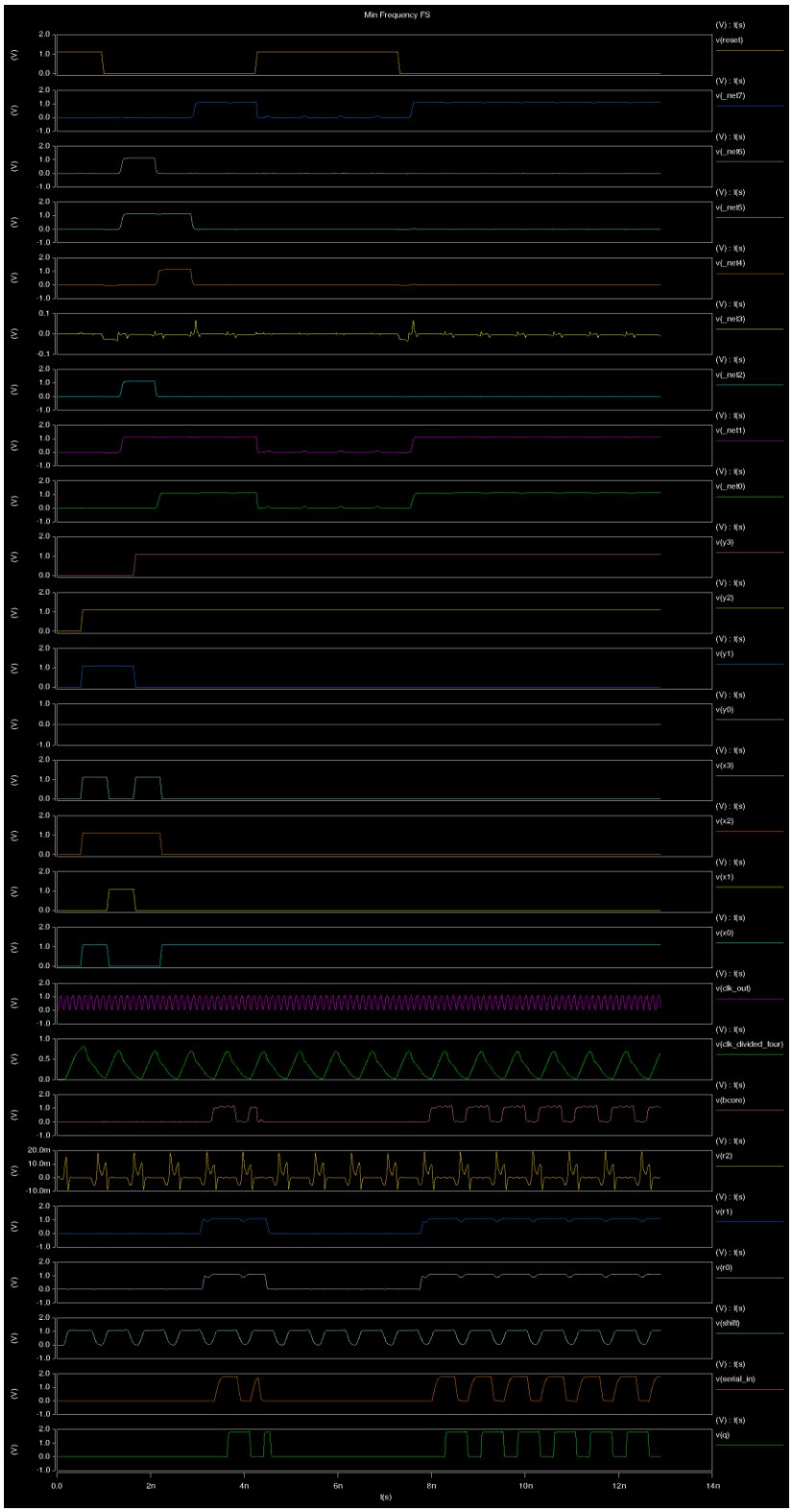

Figure 25. Minimum frequency QRC testbench, FS

(e). FF

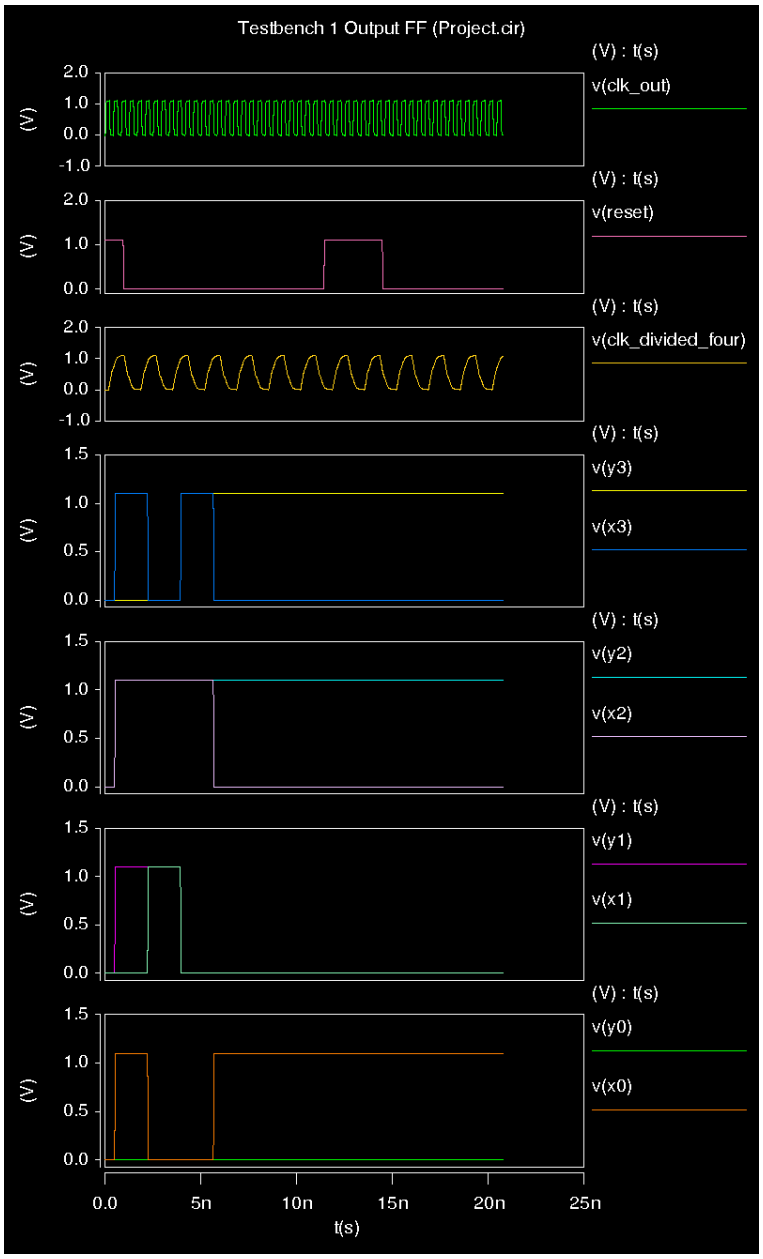

Figure 26. 2.4 GHz testbench, FF

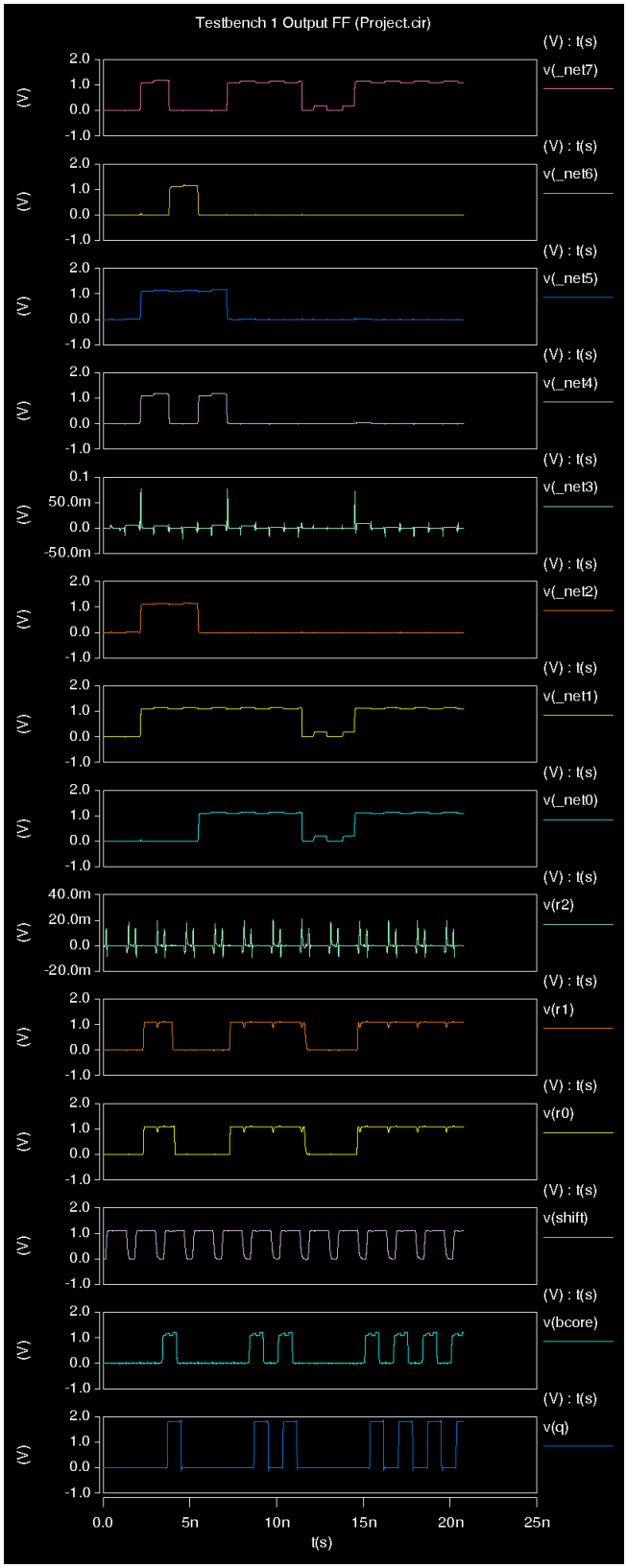

Figure 27. 2.4 GHz QRC testbench, FF

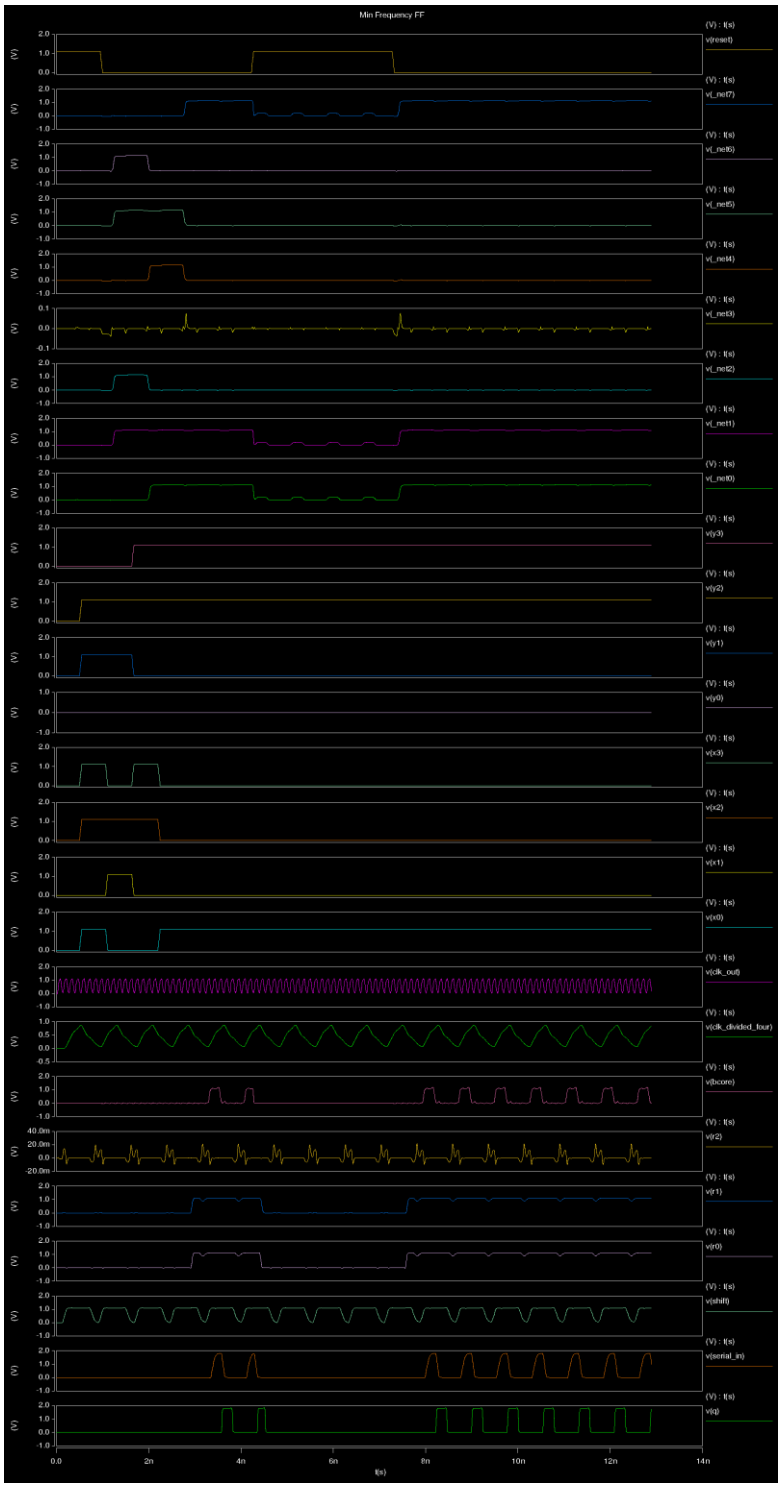

Figure 28. Minimum frequency QRC testbench, FF

v. Discussion of Results

Overall, our image comparator circuit performs as expected across all test corners, apart from the initial stages of the Slow-Slow (SS) corner. During this phase, a timing issue arises due to the inherent delay in the clk / 4 signal. As a result, the registers prematurely latch the x and y input signals when they are not yet stable, leading to an unintended output. This issue occurs because the delayed clk / 4 signal does not properly synchronize with the input signal transitions in the SS corner, where both the drive strength and propagation speeds are significantly reduced.

Upon analyzing this behavior, we identified that the problem stems from the mismatch between the setup and hold times of the registers and the delayed clock signal. While this issue is isolated to the beginning of the SS corner, it highlights a potential improvement area in the clock generation and synchronization circuitry. To address this in future iterations, we could explore techniques such as increasing the robustness of the clock divider or implementing additional clock synchronization measures to account for extreme process variations. Overall, these findings emphasize the importance of careful clock design and timing verification, particularly in circuits operating under challenging corner conditions.

vi. Top View of Entire Layout

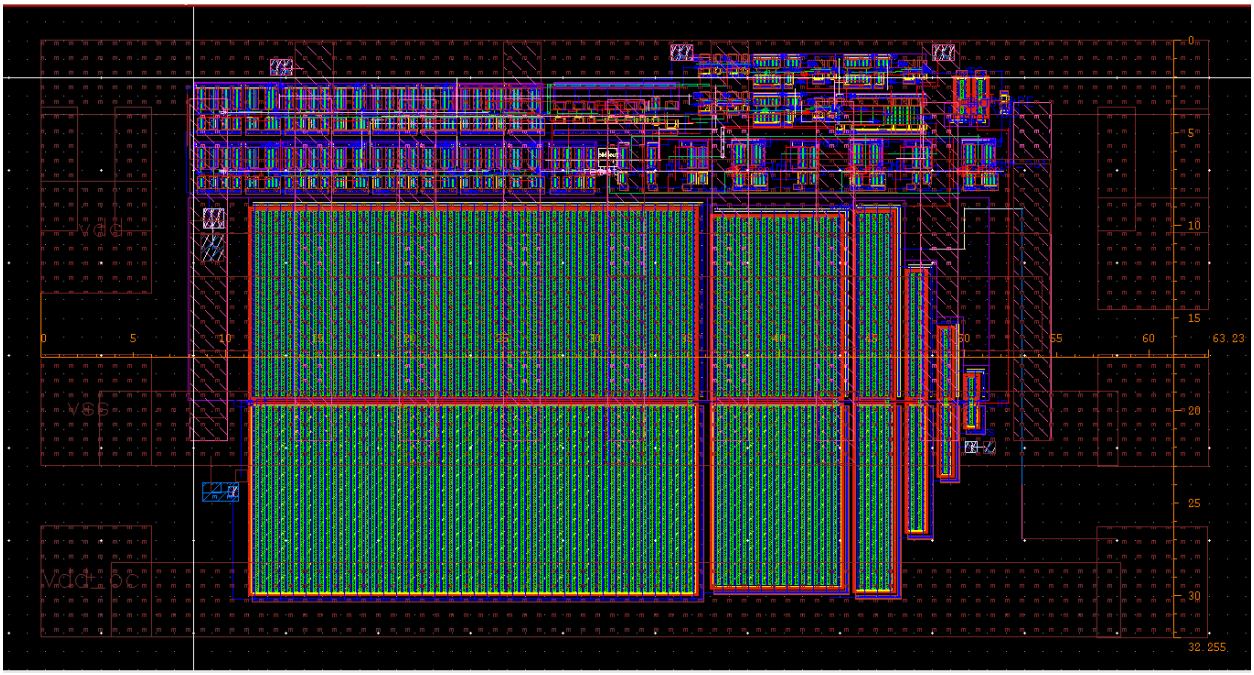

Figure 29. Top view of the entire layout

The overall layout strategy for the image comparator was to connect everything as close to power rails as possible. We first laid out all the components individually before connecting all the parts together. Then we laid out everything with the exception of the superbuffer and the level shifter individually. Before anything was connected together, we unit-tested each of our designs to confirm if it matched the expected functionality. While connecting everything together, we connected everything with the exception of the level shifter and the super-buffer to

ascertain correct functionality for the PISO. Then, after the output was confirmed to match expected functionality, we connected the level shifter and the super-buffer, before finally laying out the power grid. Throughout the process, we had to re-layout the adders, XORs, and clk-divider to decrease the delay. After those were re-laid out, we wired the overall layout.

For the overall layout, we also tried our best to minimize wiring between components to reduce delay. To do this, we made sure that the layout of each component was as minimal as possible. Furthermore, we tried to align our design's heights with each other to align the power rails together with the exception of the full adders, half adders, XORs, and AND gate. However, because the design was so minimal, this did not pose a problem later when creating the power grid.

During the layout of the power grid, we tried to minimize parasitic capacitances as much as possible by adding maximum-sized metal 8 and metal 9 wires. Furthermore, we tried to connect the power grid by using large vias to reduce the resistance of the power grid.

vi. Subcircuit Layouts

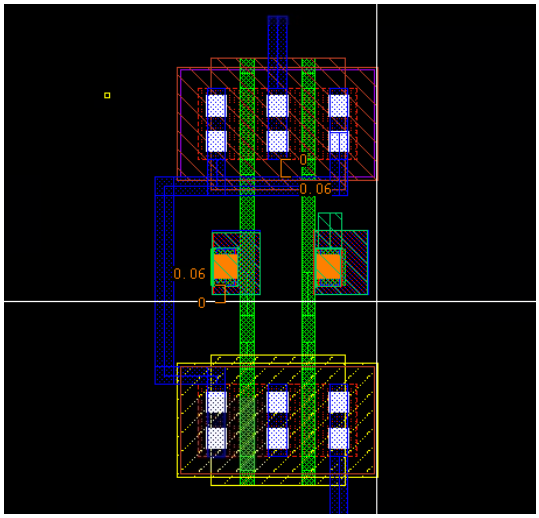

Figure 30. NAND gate

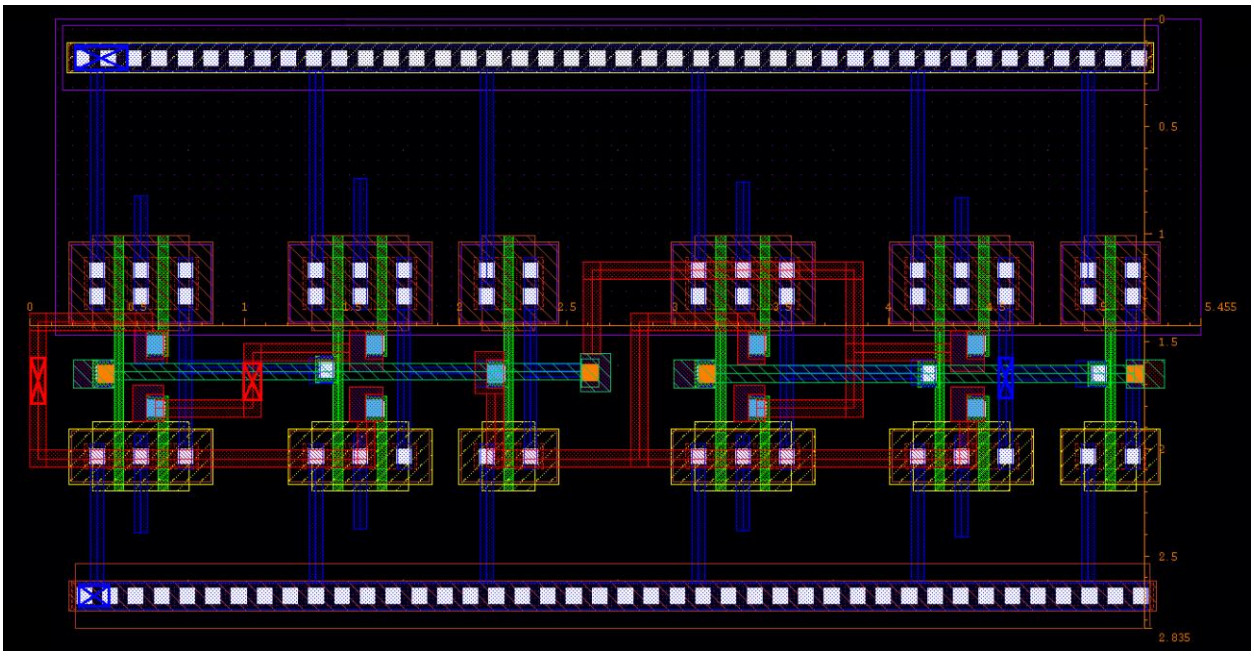

Figure 31. Clock divider

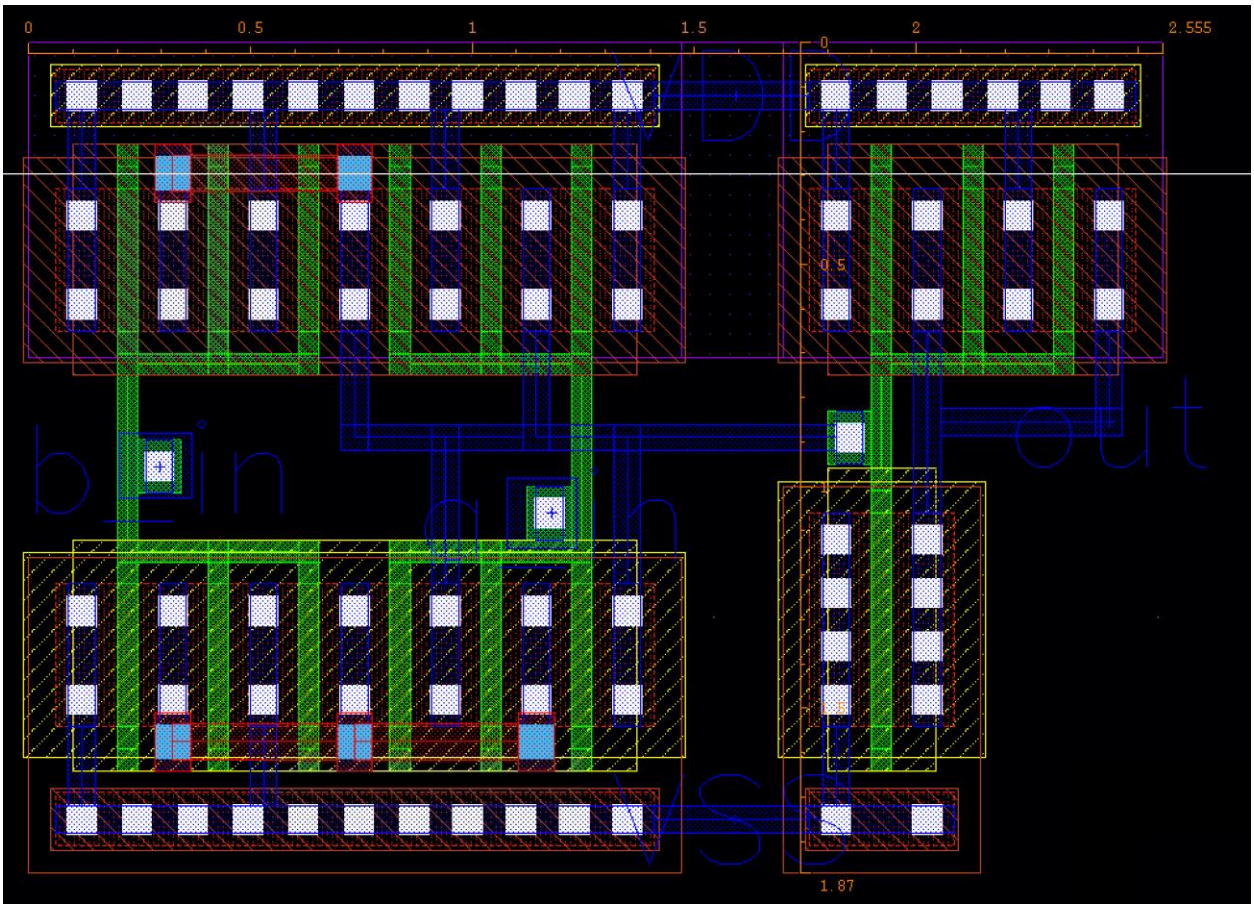

Figure 32. And

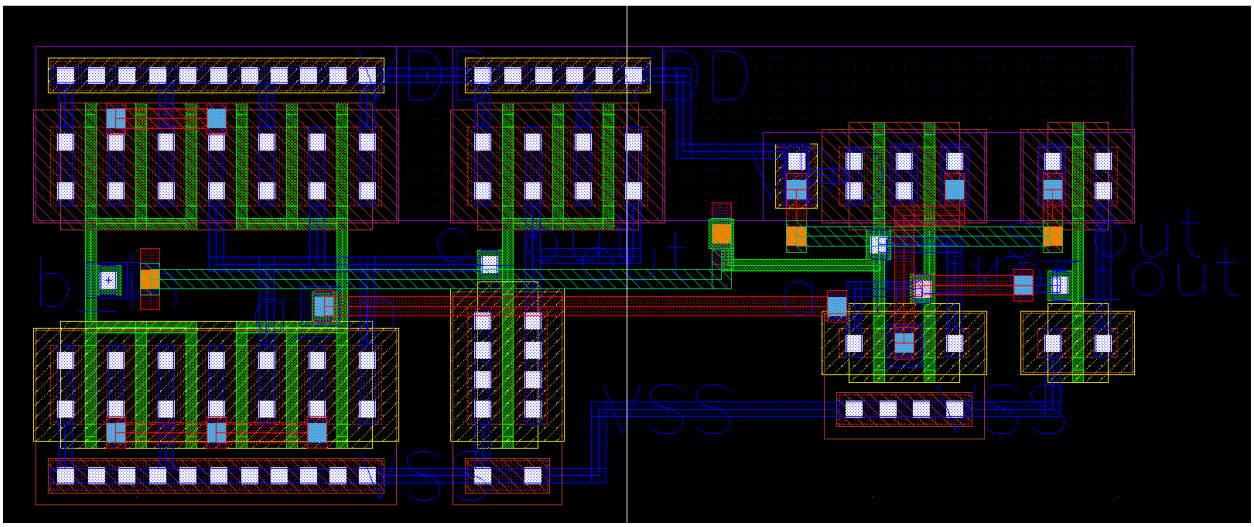

Figure 33. Half-Adder

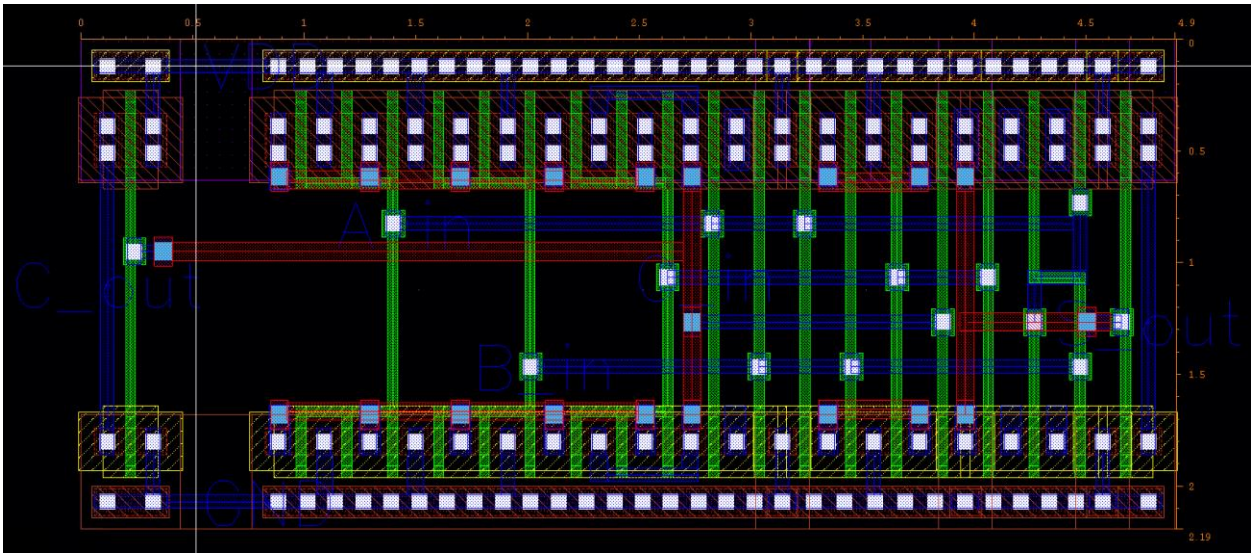

Figure 34. Full Adder Layout

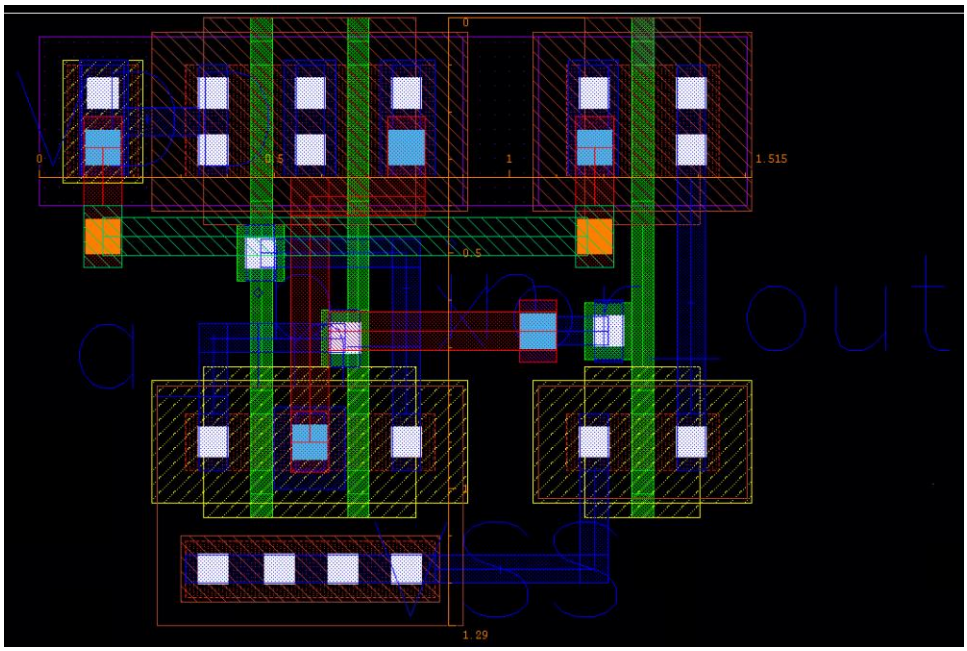

Figure 35. XOR Gate

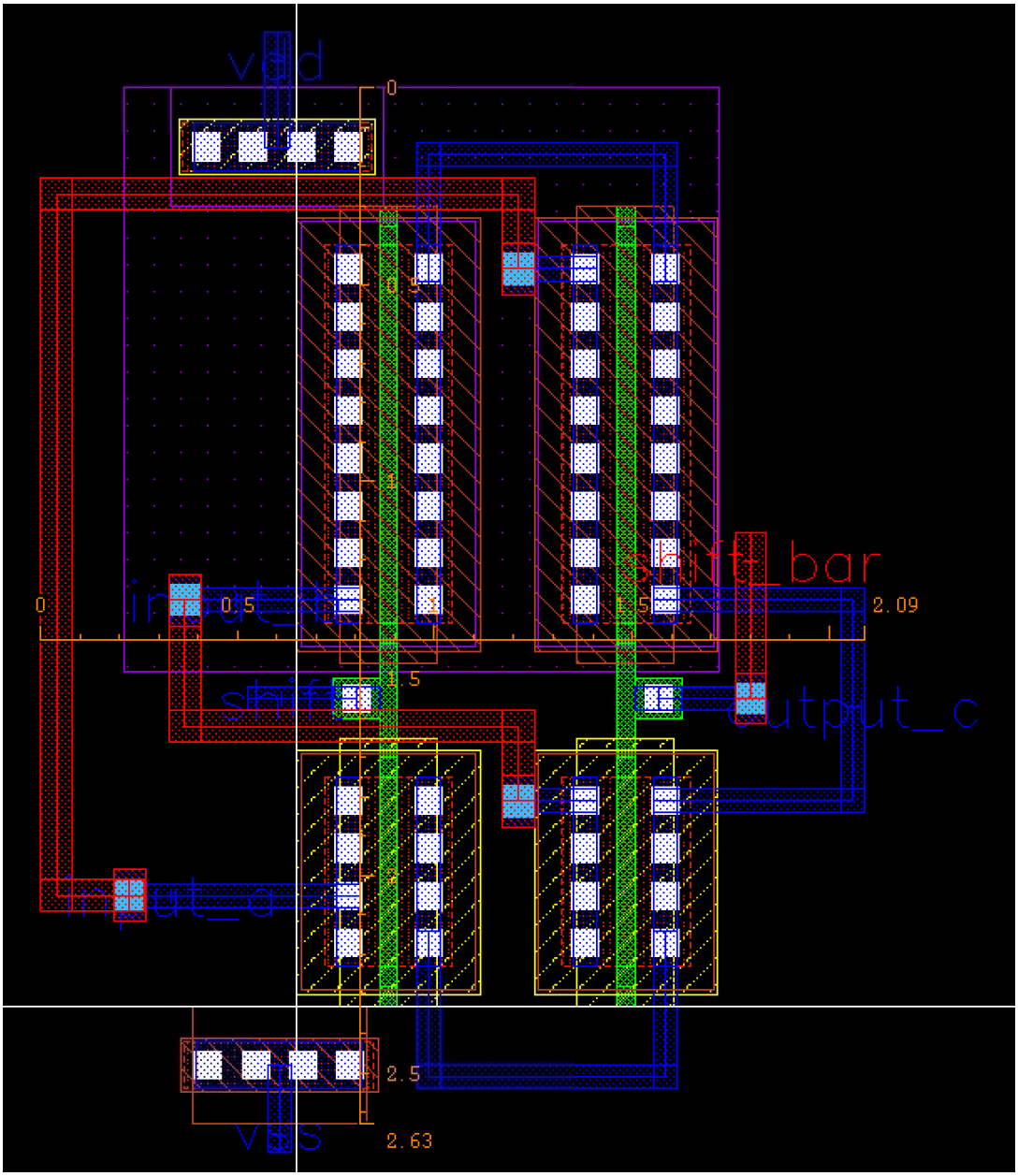

Figure 36. MUX Layout

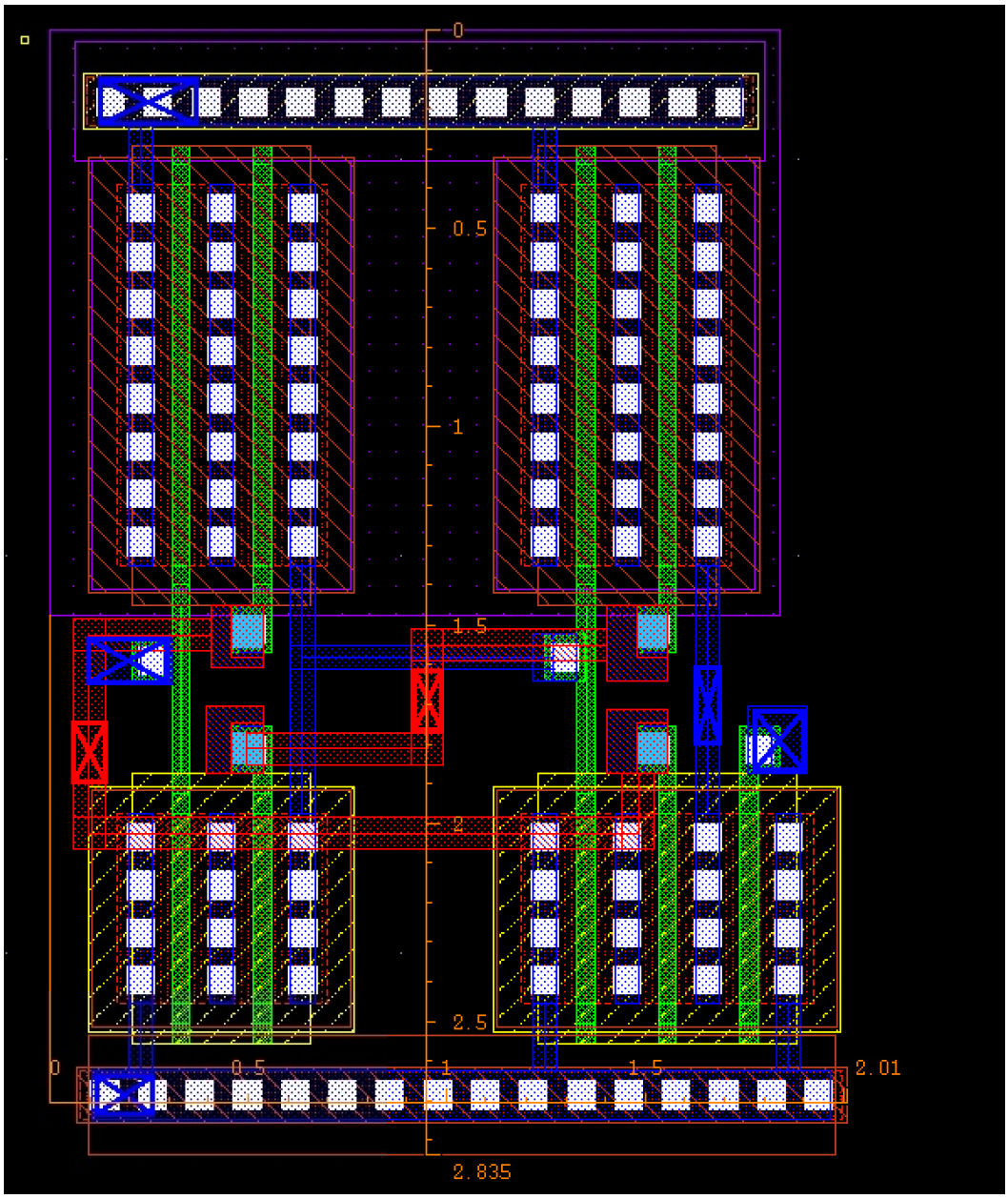

Figure 37. Register Layout

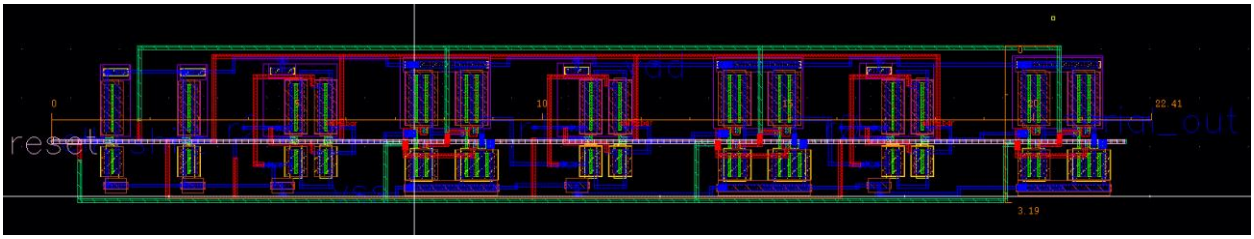

Figure 38. PISO Layout

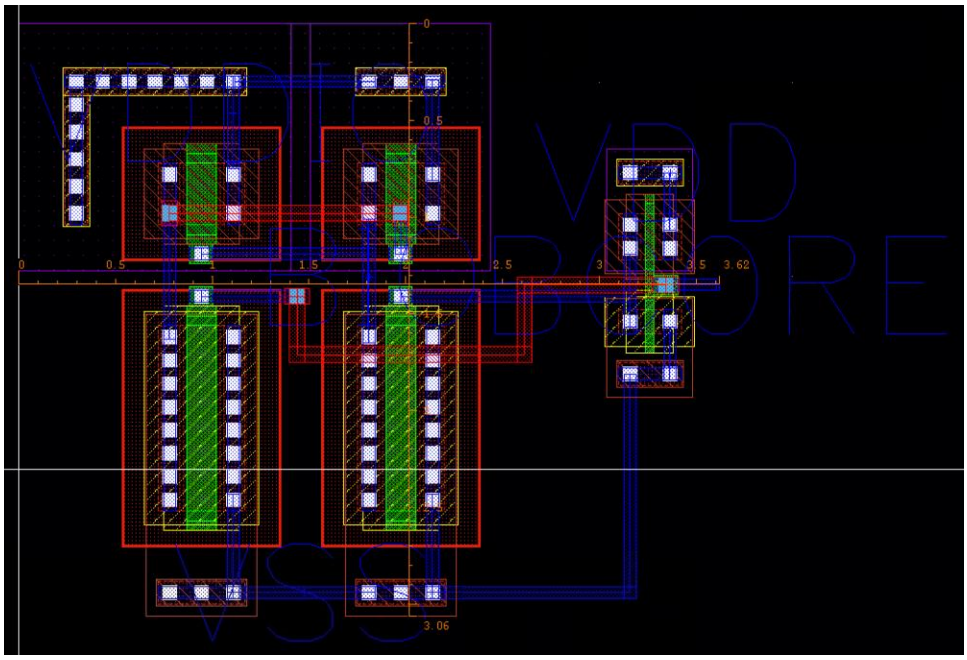

Figure 39. Level shifter

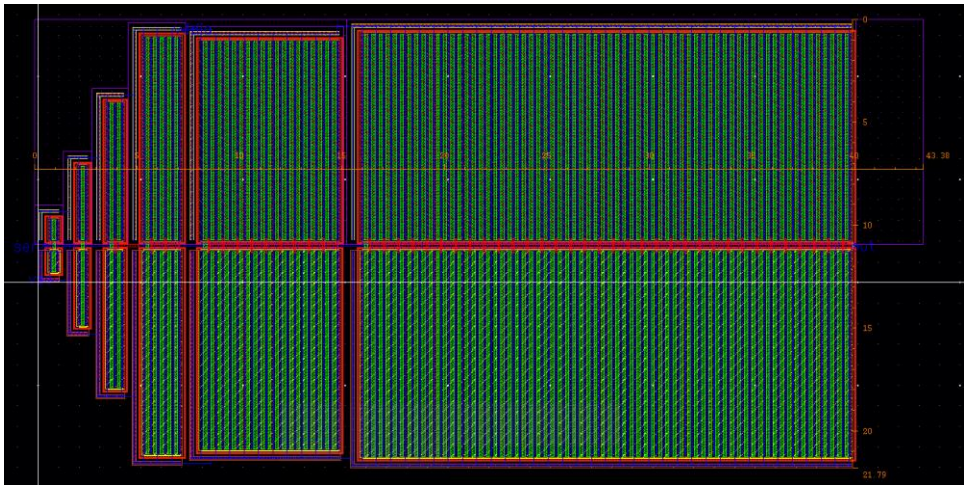

Figure 40. Superbuffer

vii. How Workload was Distributed

The workload was distributed evenly and fairly, playing to each of our layout strengths. Andrew was in charge of the design for XOR, half adder, and full adder since he had the strongest understanding of the functionality of these components. Kelly did 4-bit registers (PIPO), power grid, setup Github for sharing designs, connected and tested all schematics and layouts since she had the most solid understanding of the overall design. Mahir designed the PISO, superbuffer, and redesigned layout for level shifter since he had the most comprehensive understanding of how to design these components. Finally, Yibai designed the register, clock

divider and helped debug the overall circuit since she best understood how to layout the fundamental components of the design.