

I designed an 8-bit shift register capable of performing both serial-in/parallel-out (SIPO) and parallel-in/parallel-out (PIPO) operations, controlled by an asynchronous select signal. The design uses TSPC (True Single-Phase Clock) D flip-flops along with 2x1 multiplexers. When the select signal is 1, the register operates in PIPO mode, and when the select signal is 0, it operates in SIPO mode. The system functions at a clock frequency of 4 GHz (250 ps clock period), with all transistors sized at the minimum 120/45 nm. I have created 8 bit shift register to perform serial-in/parallel-out[SIPO] and parallel-in/parallel-out[PIPO] operations based on asynchronous select signal using TSPC (True single phase) D flipflop and 2x1Mux. Select 1 corresponds to PIPO and select 0 to SIPO. Design functions at 4Ghz[250ps] All my transistors are minimum size 120/45 nm.

Description of 8-bit Shift Register using TSPC D Flip-flops:

Architecture:

- Implemented using True Single Phase Clock (TSPC) D flip-flops
- Each stage comprises a 2:1 multiplexer feeding into a D flip-flop
- Total of 8 stages for 8-bit operation
- Clock period: 250ps (4GHz frequency)

Data Path Configuration:

1. Serial Operation:
 - Serial input propagates through the register stages
 - Each bit is shifted one position per clock cycle
 - Complete 8-bit serial transfer takes 8 clock cycles (2ns total)
 - Output (Q) of each flip-flop feeds into the multiplexer of the subsequent stage
2. Parallel Operation:
 - Parallel inputs are available at each stage through the multiplexers
 - All 8 bits are loaded simultaneously on a single clock edge
 - One clock cycle (250ps) latency for parallel data to appear at outputs

Interface:

- Control Signal: Multiplexer select line determines serial/parallel operation
- Inputs:
 - Serial input (1-bit)

- Parallel input (8-bit bus)
- Outputs:
 - 8-bit parallel output bus
 - Each bit available as individual output (Q) from respective flip-flop

Working:

A 2x1 multiplexer (Mux) before every register(D FF) to select between either a serial input or parallel input to the D flip-flop, depending on the select signal of the Mux. The outputs (Q) of the flip-flops serve as the parallel outputs and are also fed back to the subsequent Muxes to facilitate the serial shifting of inputs across the flip-flops. Parallel inputs are provided at each stage as well parallel-in operation.

Creating a 8-bit bus for the parallel outputs. With every clock pulse (250 ps per clock cycle), each bit in the serial input gets shifted (delayed) by one clock pulse to the next flip-flop. It takes 8 clock pulses (2 ns in total) for the entire 8-bit serial input to propagate through the shift register.

In the case of parallel input mode, the entire 8-bit parallel input gets shifted to the register chain within a single clock pulse.

WHY TSPC Register ?

I have designed TSPC register for following reasons compared to CMOS/PNAR CMOS/TX GATES/C2MOS

1. we use only 1 clock for TSPC, which reduces clocking complexity and power consumption.
Most registers gates(CMOS/PNAR CMOS/TX GATES/C2MOS) have two non-overlapping phases (clk and clk-bar) with potential timing issues due to clock skew between the phases increasing design complexity and power consumption.
2. TSPC designs are faster with fewer transistors in the critical path resulting in lower propagation delay and faster switching times.
3. It is Area efficient reducing the overall transistor count and layout complexity.
4. With the lower power and faster speed . This makes an Ideal Power-Delay Product(PDP).
5. Reduced switching Noise and is important in high-speed, low-power applications.

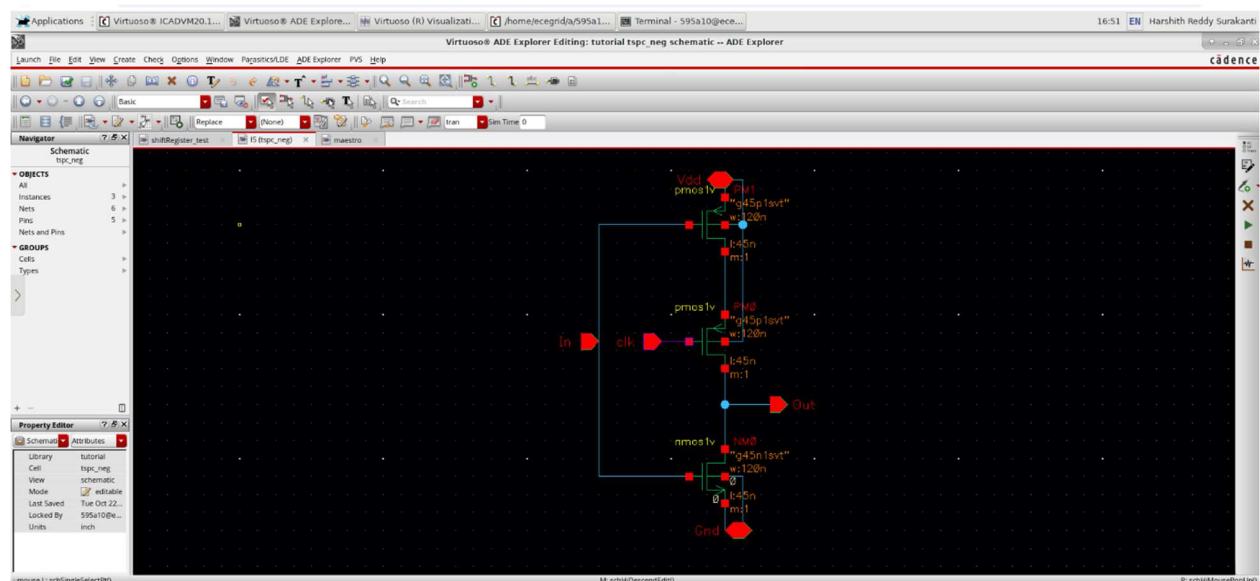
TSPC D FLIPFLOP

D Flipflop Using True single phase(TSPC) Register is designed in 4 stages.

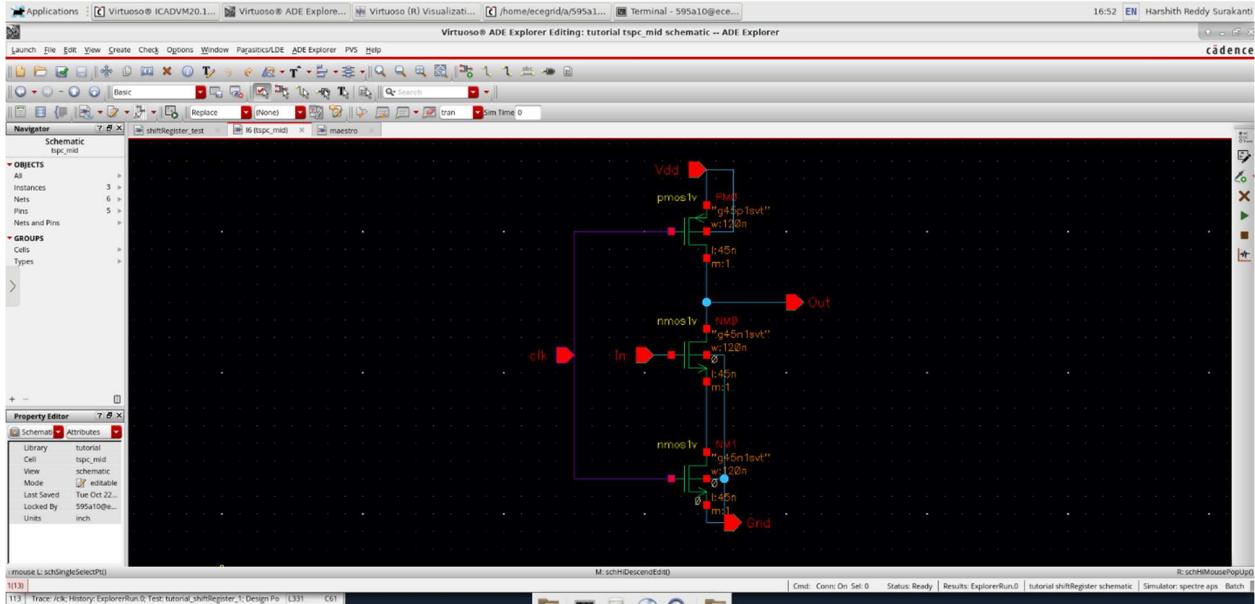
1. Negative Latch stage
2. Mid stage
3. Positive Latch
4. Inverter

Attached the schematics below

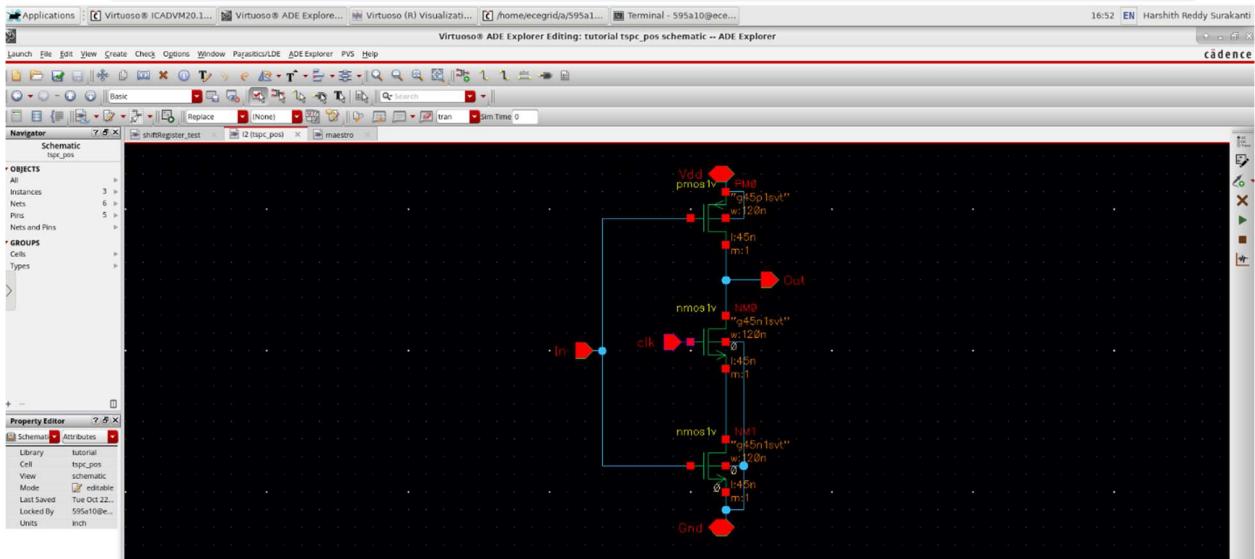
1st stage – TSPC NEGATIVE LATCH



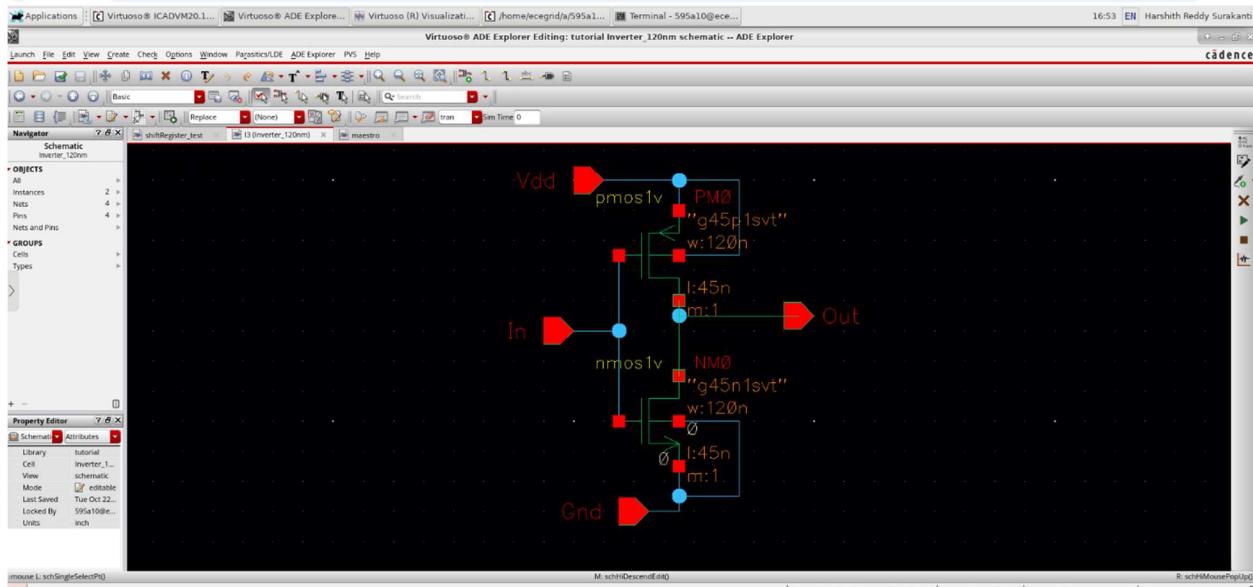
2nd Stage – Mid-stage



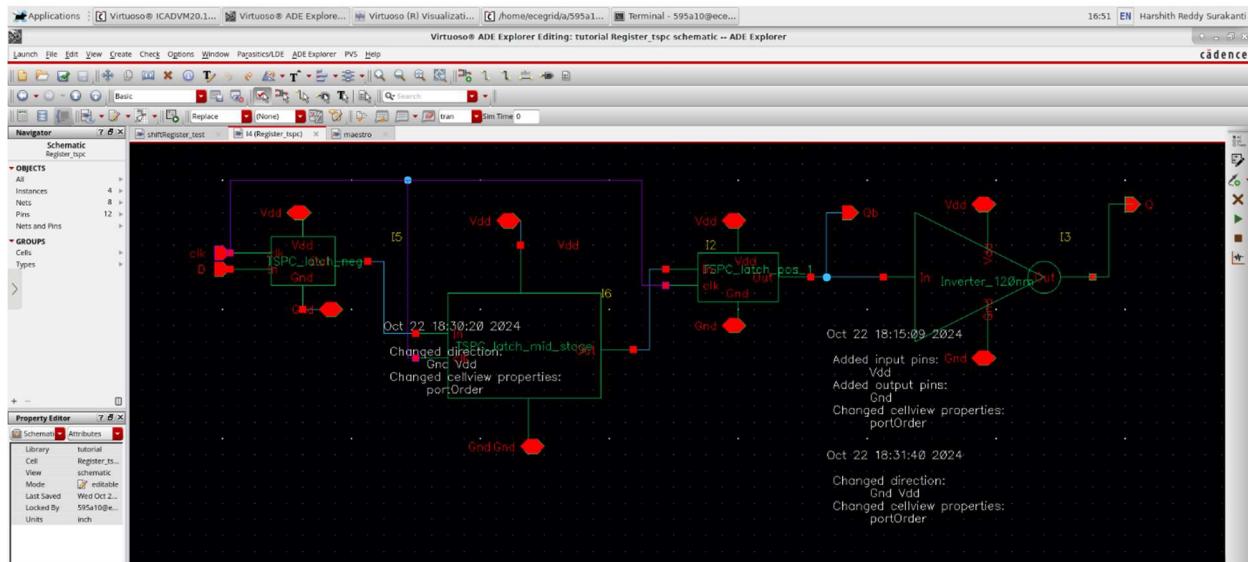
TSPC – POSITIVE LATCH Stage



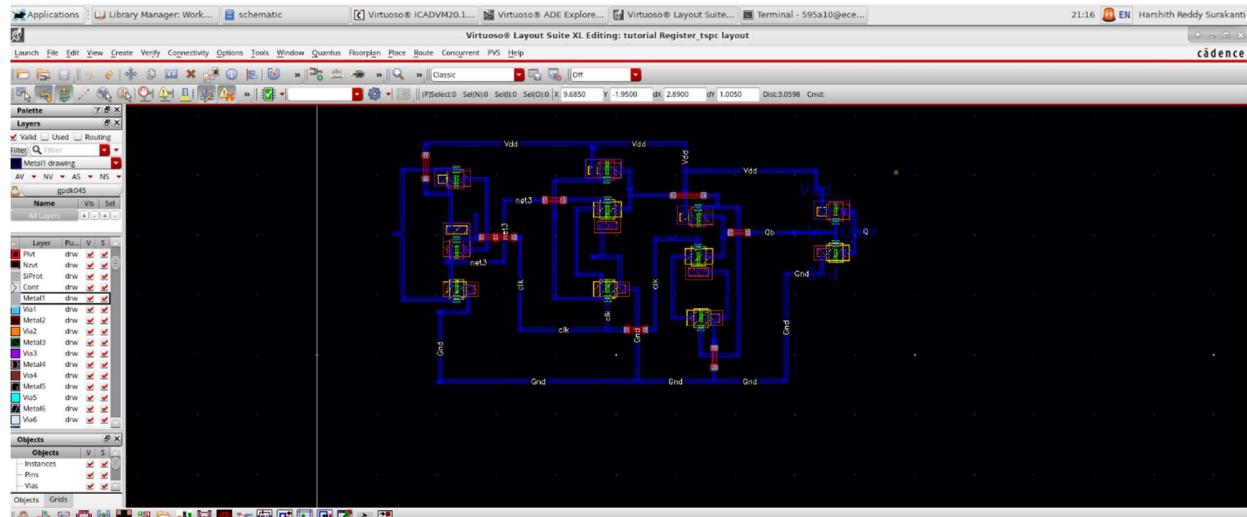
Inverter 120/45 size:



TSPC D FLIPFLOP (please Ignore the symbol comments)



Layout of D flipflop:



DRC check for D-flipflop:

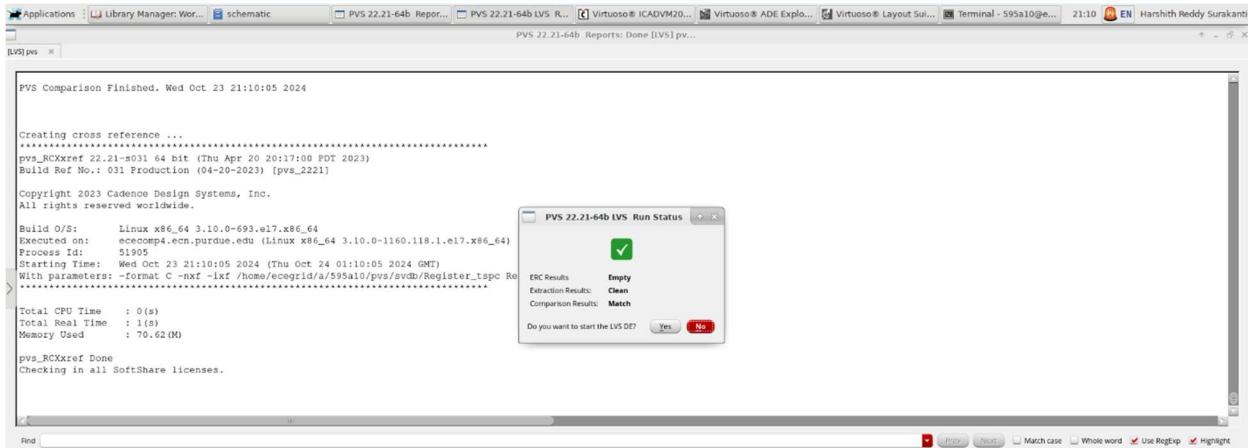
```
Applications Library Manager... schematic PVS 22.21-64b DR... PVS 22.21-64b DR... PVS 22.21-64b Re... C PVS 22.21-64b Re... C Virtuoso® ICADV... Virtuoso® ADE Ex... Virtuoso® Layout ... Terminal - 595a10... 20:37 EN Harshith Reddy Surakanti [DRC] pvs : x

ONE LAYER DRC: Cumulative Time CPU = 0(s) REAL = 0(s)
TWO LAYER DRC: Cumulative Time CPU = 0(s) REAL = 0(s)
NET AREA: Cumulative Time CPU = 0(s) REAL = 0(s)
DENSITY: Cumulative Time CPU = 0(s) REAL = 0(s)
MISCELLANEOUS: Cumulative Time CPU = 0(s) REAL = 0(s)
CONNECT: Cumulative Time CPU = 0(s) REAL = 0(s)
DEVICE: Cumulative Time CPU = 0(s) REAL = 0(s)
ERC: Cumulative Time CPU = 0(s) REAL = 0(s)
PATTERN_MATCH: Cumulative Time CPU = 0(s) REAL = 0(s)
DFM_FILL: Cumulative Time CPU = 0(s) REAL = 0(s)

Total CPU Time : 1(s)
Peak CPU Time : 1(s)
Peak Memory Used : 19(M)
Total Original Geometry : 285(353)
Total DRC RuleChecks : 562
Total DRC Results : 0 (0)
Summary can be found in file Register_tspc.sum
ASCII report database is /home/ecgrid/a/595a10/pvs/Register_tspc.drc_errors.ascii
Checking in all SoftShare licenses.

Design Rule Check Finished Normally. Wed Oct 23 20:36:25 2024
```

LVS Check for D flipflop:



PVS Comparison finished. Wed Oct 23 21:10:05 2024

```
Creating cross reference ...
*****
pvs_RCXref 22.21-s031 64 bit (Thu Apr 20 20:17:00 PDT 2023)
Build Ref No.: 031 Production (04-20-2023) [pvs_2221]
Copyright 2023 Cadence Design Systems, Inc.
All rights reserved worldwide.

Build O/S: Linux x86_64 3.10.0-693.el7.x86_64
Executed on: eecomp4.ecn.purdue.edu (Linux x86_64 3.10.0-1160.118.1.el7.x86_64)
Process Id: 51905
Start Time: Wed Oct 23 21:10:05 2024 (Thu Oct 24 01:10:05 2024 GMT)
With parameters: format C-naf -lxf /home/ecegrid/a/595a10/pvs/svdb/Register_tspc Re
*****
Total CPU Time : 0 (s)
Total Real Time : 1 (s)
Memory Used : 70.62 (M)

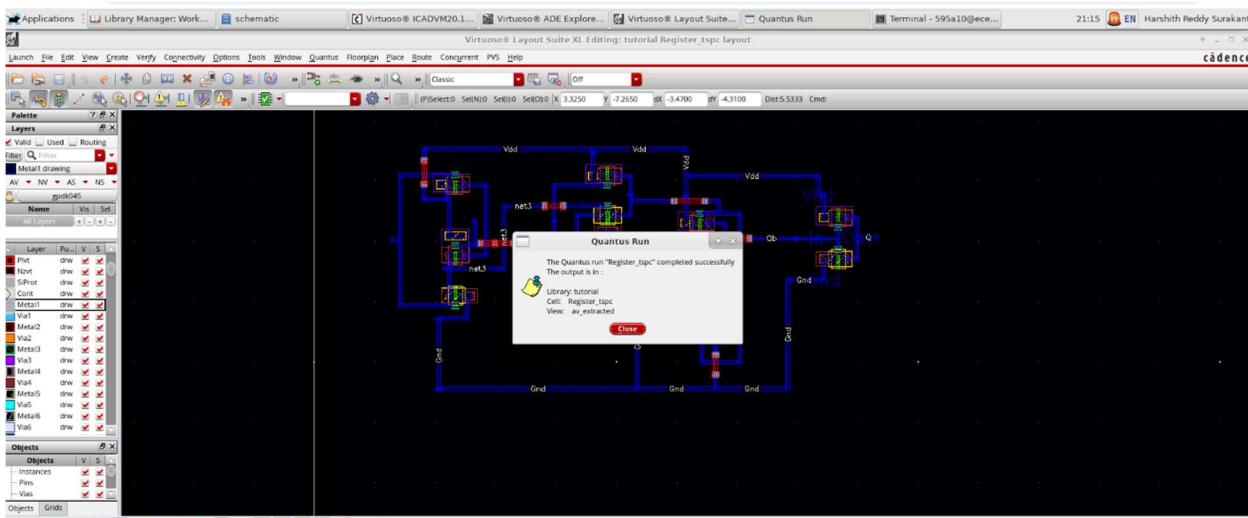
pvs_RCXref Done
Checking in all SoftShare licenses.
```

PVS 22.21-64b LVS Run Status

ESC Results	Empty
Extraction Results:	Clean
Comparison Results:	Match

You want to start the LVS DE? Yes No

AV extraction successful:

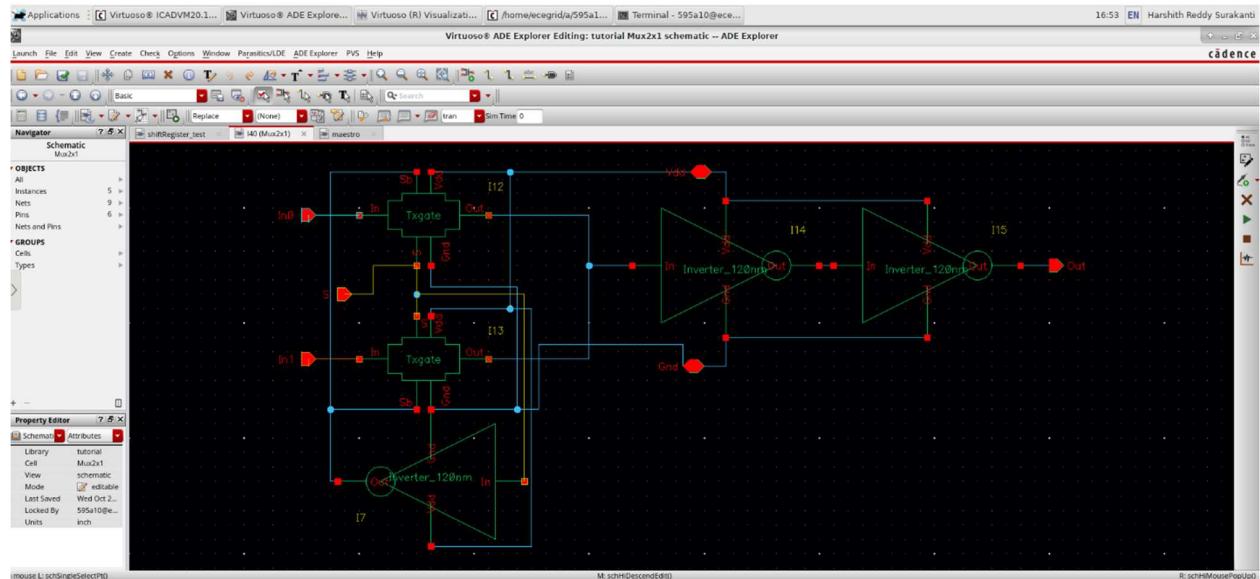


2x1 MUX

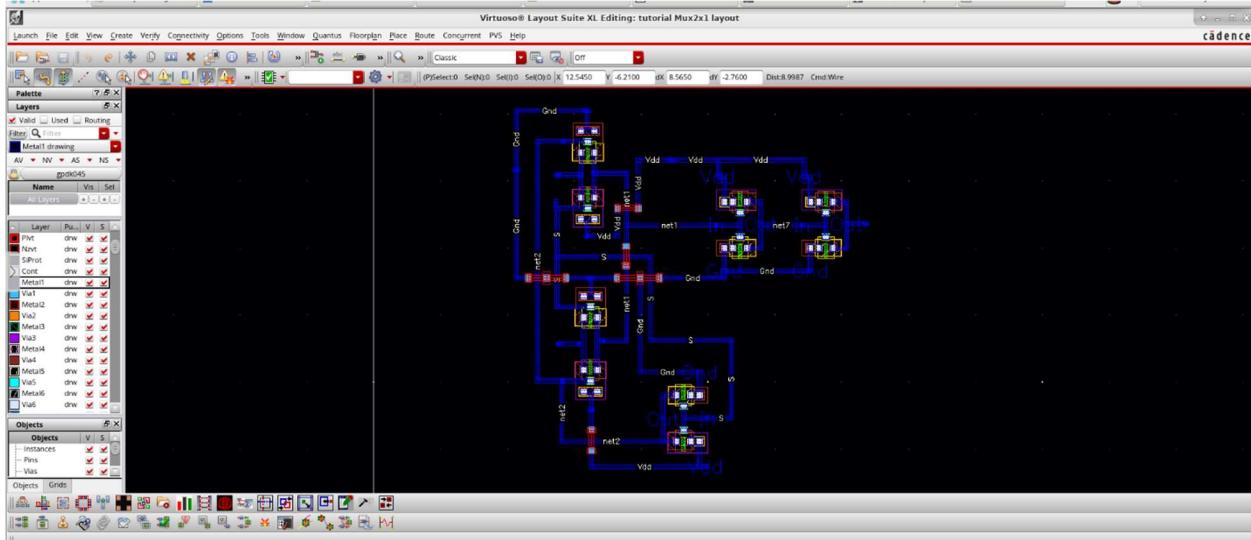
I have created 2x1Mux using Transmission gates, It helped me reduce the transistor sizing and faster switching speeds.

I added two inverters as buffers at the output to achieve correct signal levels. Before adding them, the output voltages were 857 mV for logic high and 357 mV for logic low. The buffers helped ensure proper signaling.

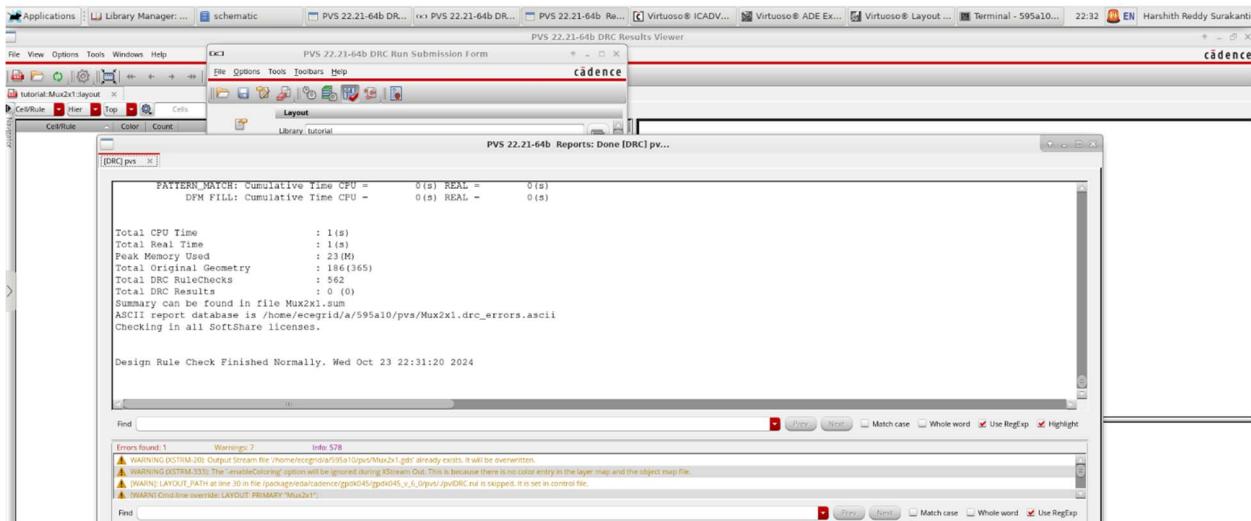
Attached are the schematics



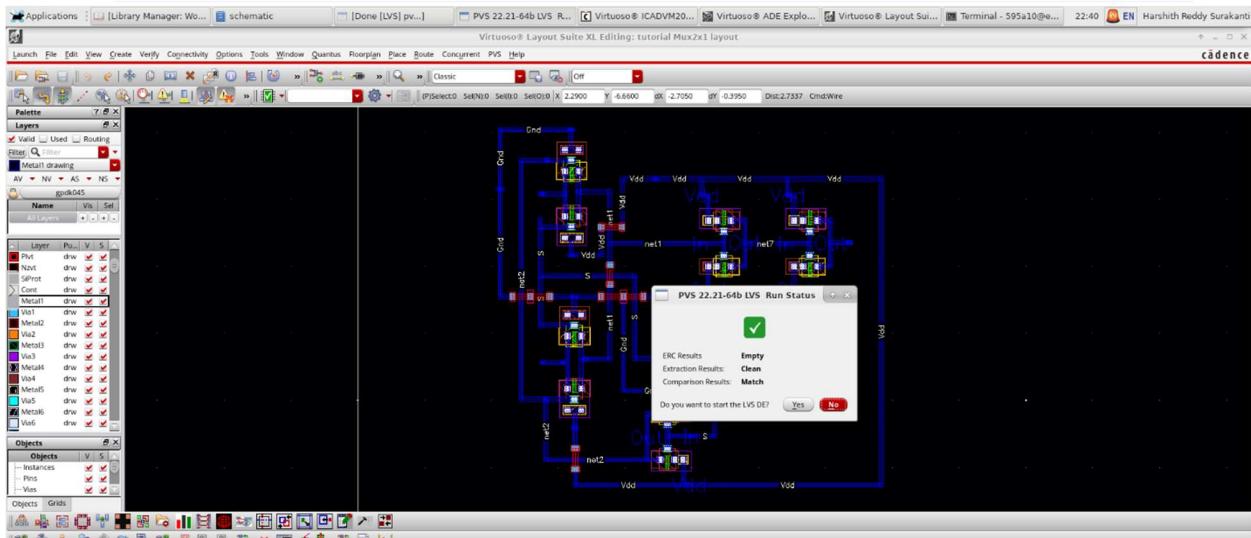
Layout:



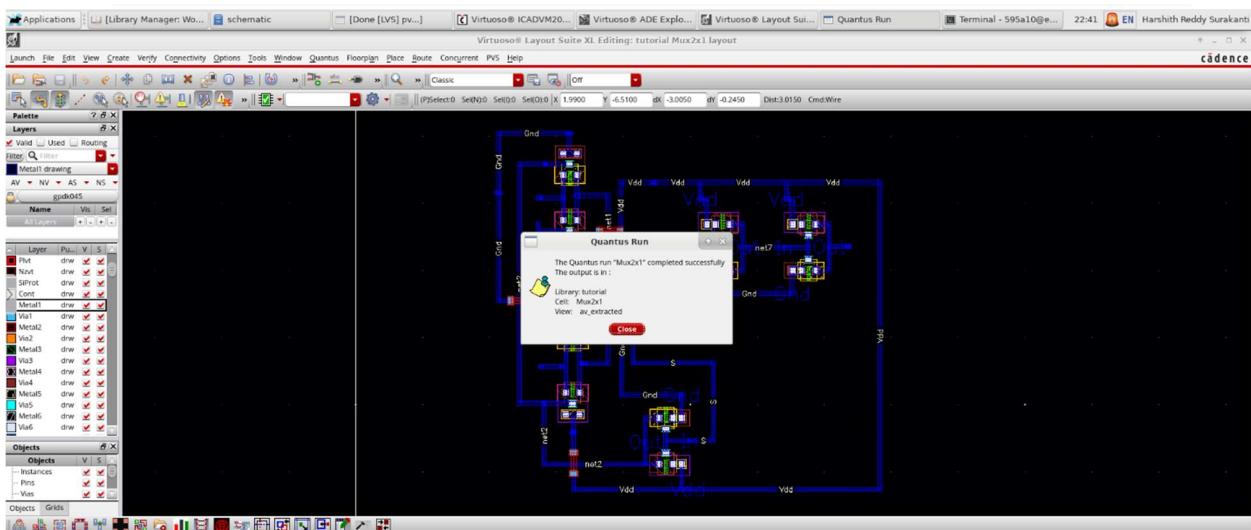
DRC Check:



LVS Check:

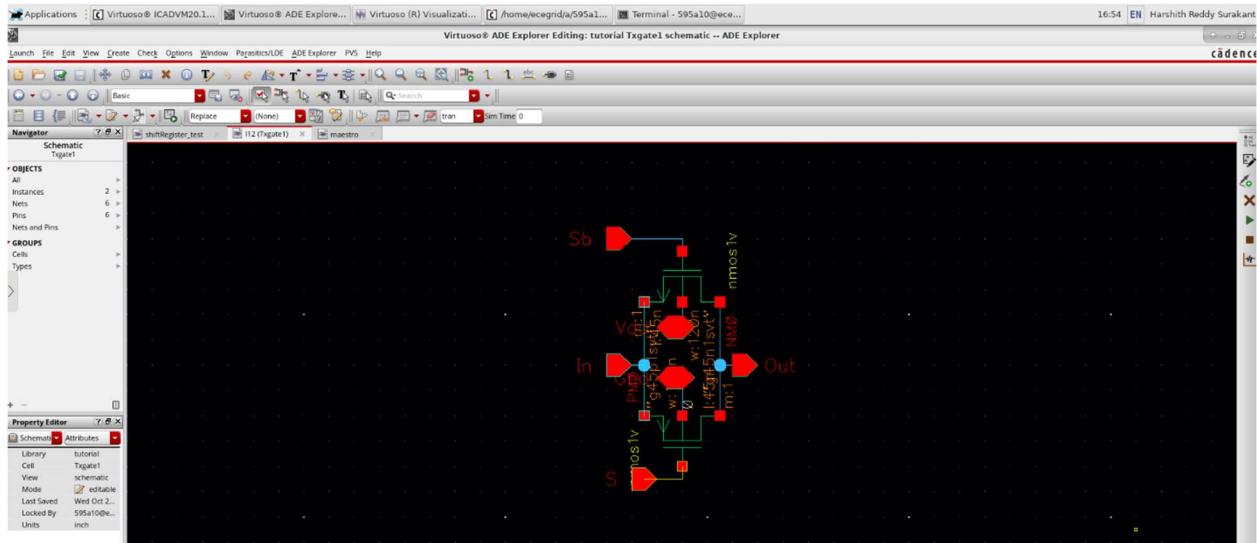


AV extraction successful :

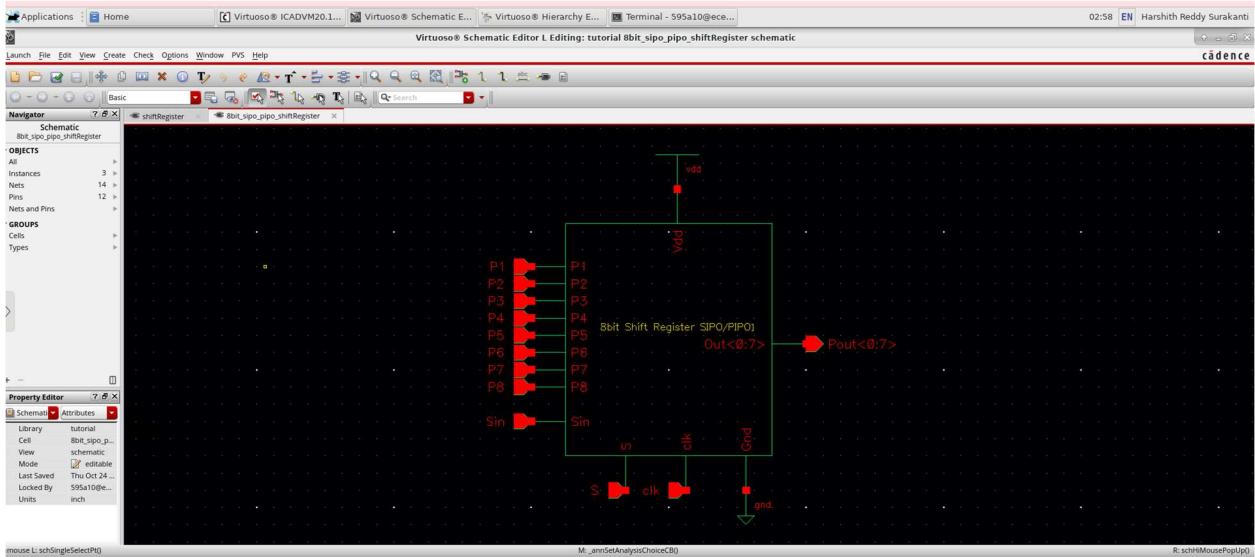


TX Gate

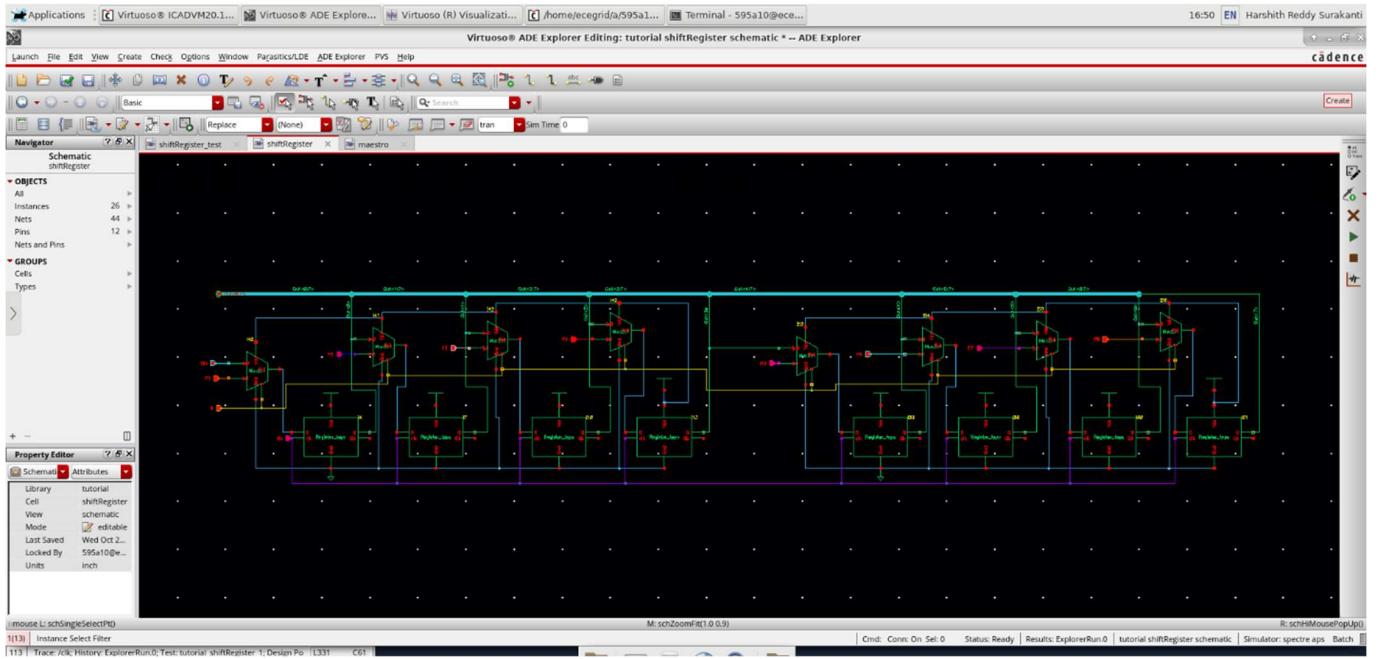
2x1 Mux gates are designed using Transmission gates.



8 Bit Shift Register Symbol with Pins and Connections



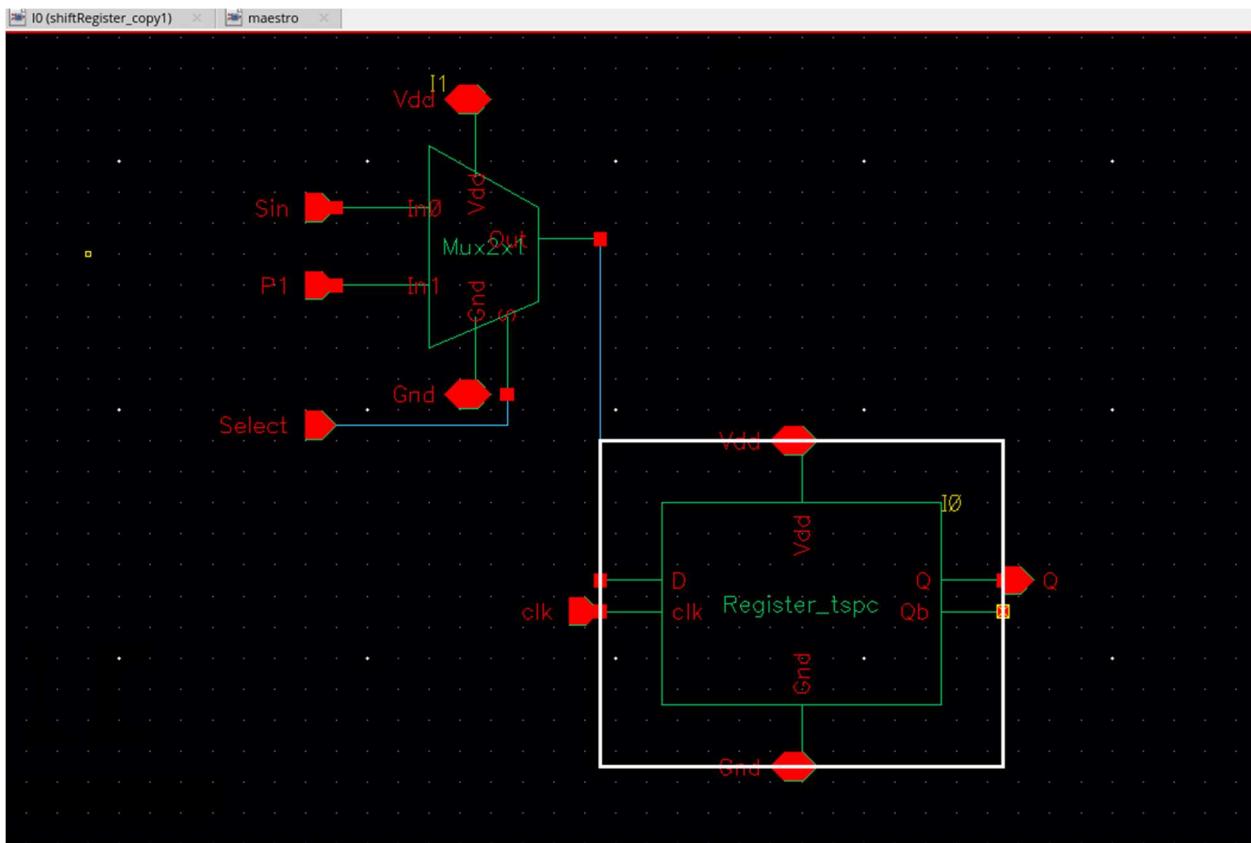
8 Bit Shift Register Schematic



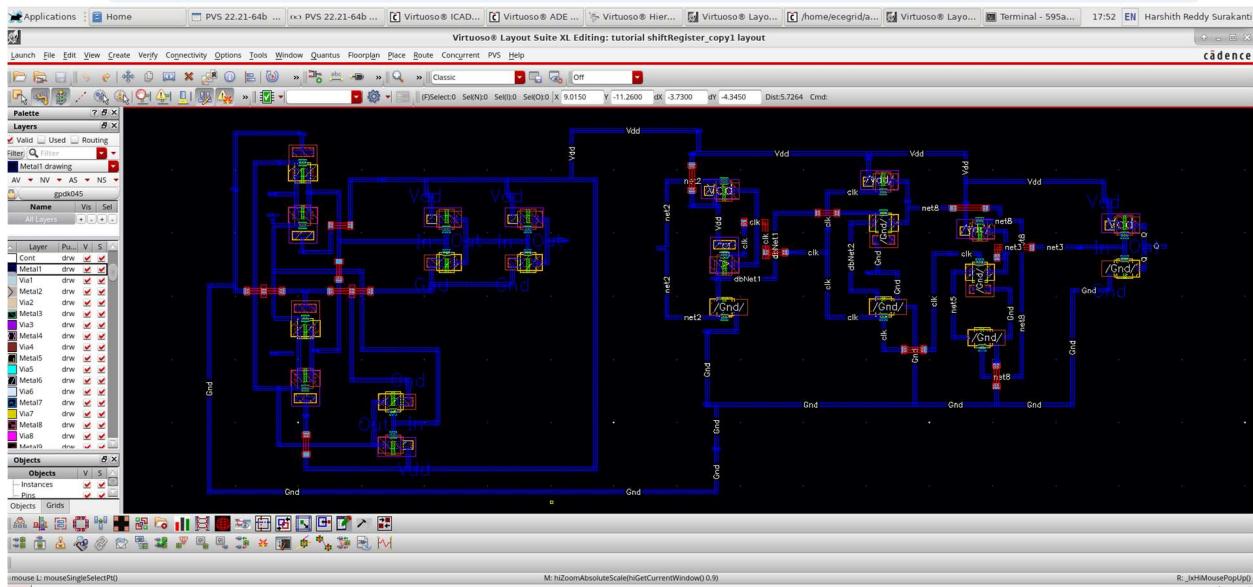
To perform the layout hassle free, I have made a symbol combining 1 register and 1 mux.
Schematic, layout, drc and LVS are attached below.

And subsequently I have also made a 8bit register using this symbol and verified functionality.

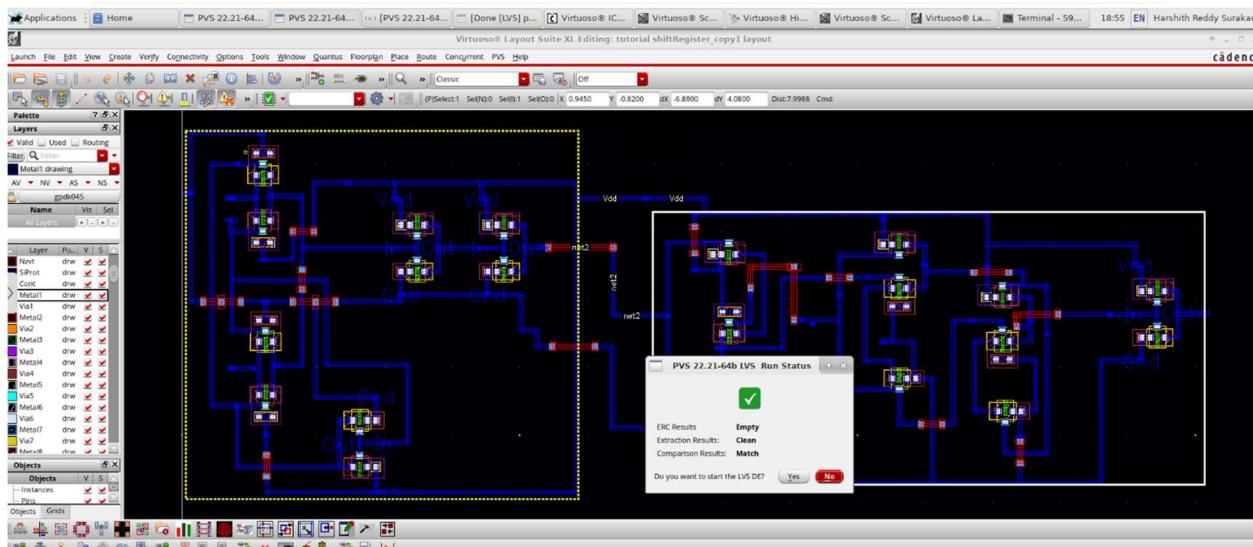
Schematic of 1 bit and 1 mux



Layout of 1 mux and 1 register



DRC/LVS Check of 1 mux and 1 register



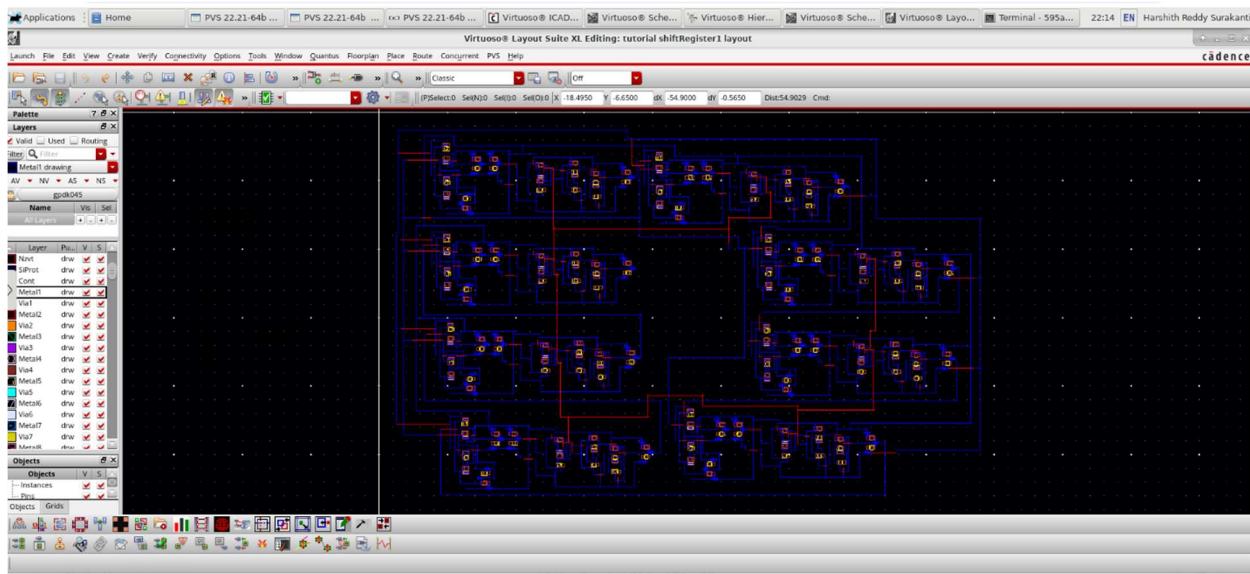


8 Bit Shift Register Schematic

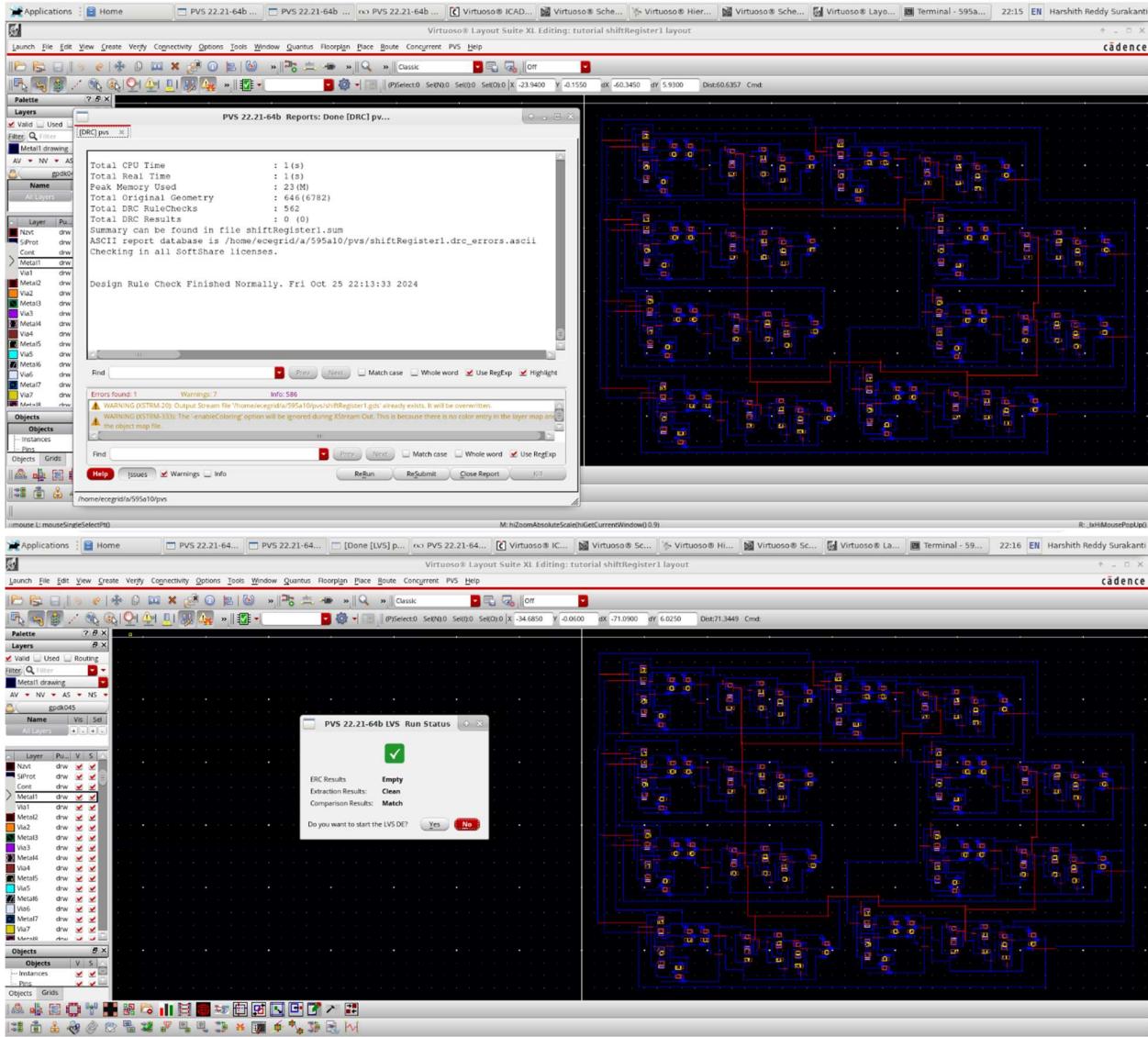
8 Bit shift Register after combining a mux and a register into symbol



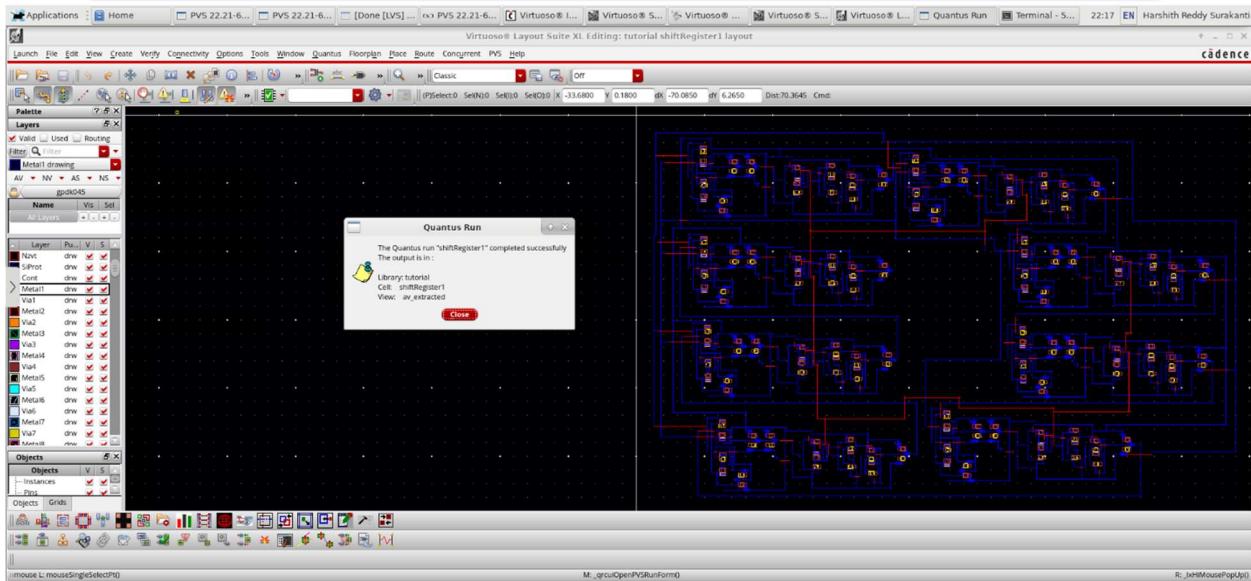
8 Bit Shift Register Layout



8 Bit Shift Register DRC Check and LVS Check



8 Bit Shift Register AV extraction successful



Pre and Post Layout Comparisons

[Select, Clock waveforms belongs to schematic simulation]

Parallel in and Parallel out [PIPO]

3 test cases were provided as parallel input during first 3 clock pulses at the beginning [Ignore more test cases that are added]

All the parallel in are delayed by 1 clock pulse [250ps] can be observed from the waveforms below.

Test Case 1

Parallel in: 11111111

Parallel out: we can observe the 11111111 after 250ps in the Waveform

Test Case 2

Parallel in: 10111110

Parallel out: we observe 10111110 after 500ps [because we gave parallel in at 250ps]

Test Case 3

Parallel in: 00000000

Parallel out: we observe 00000000 after 750ps [because we gave parallel in at 500ps]



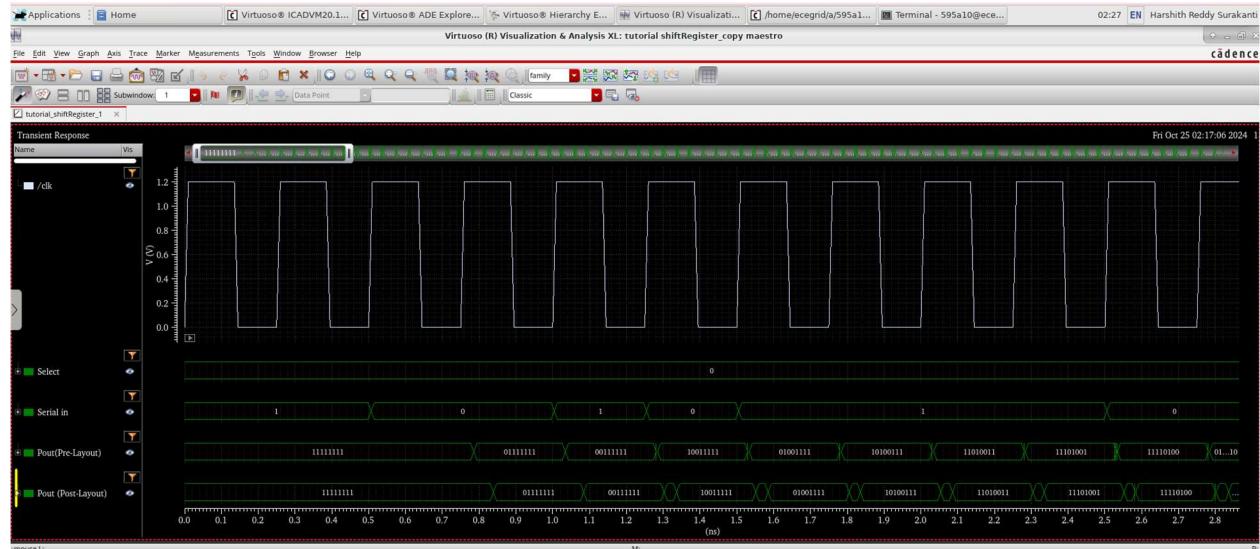
Serial IN Parallel out [SIPO]

For all SIPOs , Because of 4 Ghz[250ps period], if we check after 2ns[250ps*8] . we get Complete parallel out that was given as Serial in :

Test case 4

Serial in: 11010011

Parallelout:11010011 [We can observe at/after 2 ns in both pre-layout and post layout outputs.
Post layout gives a more delay because of parasitic capacitances]

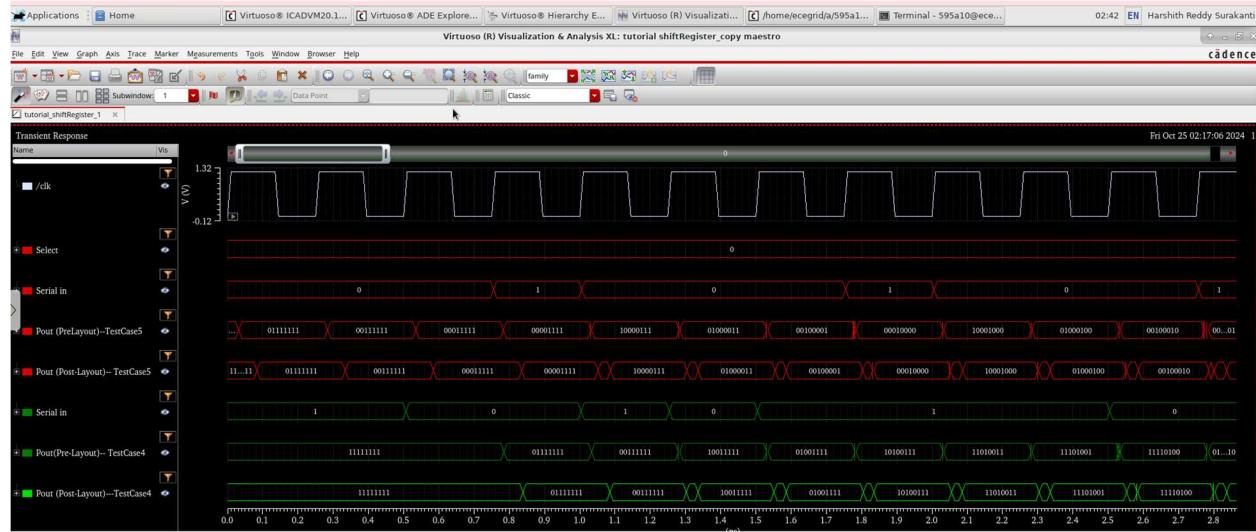


Test case 5

Serial in: 10001000

Parallelout:10001000

[Note: Testcases 4 and 5 are appended in the below waveform to verify functionality]



Test case 6

Serial in: 01010101

Parallelout:01010101



Comments from Outputs :

We can see from the outputs that all test cases were functional through the 8-bit shift register[SIPO/PIPO] designed. Post layout results have more delay than Pre-layout design because post layout considers parasitic capacitance/resistance, real wire lengths, Real metal interconnects and more