

# Ring Oscillators

## Background

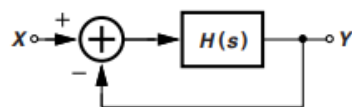
An oscillator is the beating heart of many applications. You can think of it as the deterministic trigger. Based on a set frequency of oscillation, it acts as the executioner of a pre-planned sequence. Think about it as waves that push a boat forward – that's how a sequence of bits propagates in a digital circuit, with the clock acting as the vessel. We can extend the applications of the clock to other areas of interest, such as a clocked comparator. Why would a comparator require a clock you might ask? One reason would be to conserve power by triggering the comparator (turning it on) when required to do a comparison and then turning it off again when not needed – that's assuming our comparisons are equally spaced in time. Generating another clock based on the main one for an asynchronous flow is a solution when comparisons are required but based on a previous condition that needs to be met.

What metrics are of interest in the case of an oscillator? Frequency should be the first thing that comes to mind. Another would be the output clock's shape and voltage swing (typically oscillates from ground till  $V+$ ). Another metric that deems very crucial is the variation of the clock i.e. does the rising/falling edge fall exactly at the same point over many cycles? Theoretically speaking, an ideal clock with a period of  $T = 1\text{ms}$  will have the rising/falling edges equally spaced apart by  $k \cdot T$  where  $k$  is an integer. What happens when the spacing becomes  $1.001\text{ms}$ ? Would that spar an issue in our system?

Naturally we arrive to the discussion of noise, an inherent aspect of many circuits and especially for oscillators that, funnily enough, require noise to start (more on that later) but largely affects the clock's performance when in steady state. Someone might ask why would a small variation be that critical? If you're using the clock to trigger a comparator on and off it wouldn't matter as much as using it to upconvert or downconvert an RF signal, this added variation is an added term in the modulation/demodulation function that could corrupt a symbol and create an error. Jitter is the word used to quantify this variation; we'll be simulating that later.

We've covered a good amount of information on oscillators, but the question now is how to build them? With the aforementioned metrics in mind, we choose the appropriate topology that meets out specs. We'll choose the Inverter based **Ring Oscillator** as our topology [1].

Mathematically speaking, an oscillator should satisfy the Barkhausen Criteria to oscillate, this means that, looking at the oscillator as a system



$$\frac{Y(s)}{X(s)} = \frac{H(s)}{1 + H(s)}, \quad |H(s)| = 1, \quad \angle H(s) = 180^\circ$$

A three-stage inverter-based oscillator is the minimum number of stages that can oscillate, keep in mind that it should be an odd number of stages.

How is the frequency determined? The size of the inverters is the main determining factor; another is the node capacitance seen by the output of the inverter. Since our specs requires us to achieve the targeted frequency over PVT, we should construct a method to change one of the mentioned factors. We've decided to change the output node capacitance, since it paves the way for the idea of calibration i.e. digital control of an analog circuit.

## Schematic Design

We'll build the oscillator together in Virtuoso based on the GPD45 technology. A modular approach is key when building circuits for the sake of individual testing and most importantly parasitic extraction at the sub-block level. When designs get more complex, a divide and conquer strategy shines in pin-pointing the main contributors of parasitics.

We'll start with the inverter first. The design approach is a loop, in this case we build a testbench that checks our criteria, in this case the frequency of oscillation over PVT (corners setup is crucial, identifying the worst corners first is key) and we build our circuit using three stages of inverters with feedback.

The choice of devices depends on the supply voltage and their performance over PVT. I chose hvt (high voltage devices) in this case based on my simulations. The second consideration is the ratios between the PMOS and NMOS, the point here is to have them equally conduct so that the resulting duty cycle (which is of interest) be around 50%. Of course having it fixed over PVT is not very straightforward, but that's a topic for another time. I chose a ratio of 2, which is set by the PMOS having a multiplier of 4 while the NMOS having a multiplier of 2. W/L should be the same in both, I try to choose a ratio with layout in mind – a good shape is key. Other considerations are current consumption – increasing W increases the current consumption which increases the frequency of oscillation. The inverse is true for L. Keep in mind that our choice of sizing (L = 1.5um and W = 3um) plays a role in setting the output node capacitance. The minimum L in the case of hvt devices is 45nm and the minimum W is 120nm. Our numbers are large if compared with the minimum, and that's for the sake of increasing the delay of each stage which sets the frequency to the MHz range.

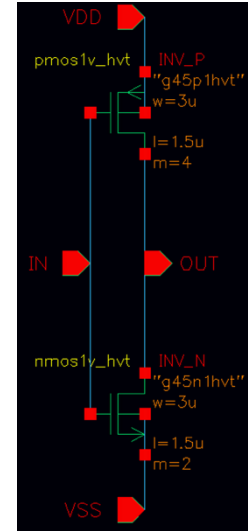


Fig. 1 Inverter Schematic

We now bring our focus on the idea of calibration i.e. tuning the circuit. We know that the frequency changes over PVT, but we want the same frequency (within our margins) to be achievable under any conditions. For this we work on setting the output node capacitance in a way that can be changed or calibrated. A binary weighted capacitor bank comes into frame.

The capacitor bank defines the capacitance seen between the output of the inverter and ground. The use of individual unit caps of 48 fF that can be enabled/disabled using a switch is our calibration circuit which consists of  $2^5 - 1$  unit caps bringing the total capacitance to 1488 fF in addition to a fixed cap of size 80 fF. Setting the 5 calibration bits to zero means we have 0 unit caps enabled and only the fixed cap of 80 fF. 5'b CAL = 1 in decimal means that only one unit cap is enabled bringing the total capacitance to 80 fF + 44fF, CAL = 2 is 80 fF + 88fF and so on. With this configuration, the target frequency can be reached over PVT.

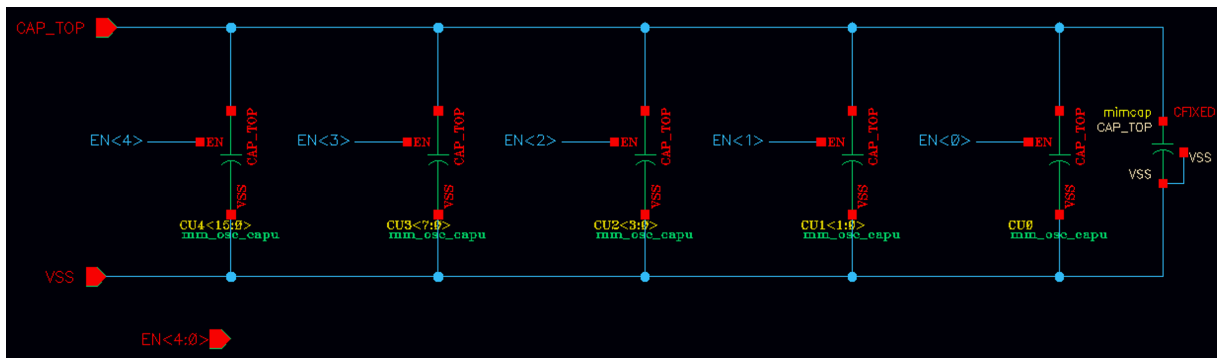


Fig. 2 Binary Weighted Capacitor Bank

The unit capacitor is a mimcap of size 48 fF with a nmos switch acting as a pull up for its top plate. When  $EN = 1$ , X is shorted to CAP\_TOP, if  $EN = 0$ , CAP\_TOP sees high impedance. You should know that other considerations come into play when designing a capacitor bank mainly  $R_{ON}/R_{OFF}$  and  $C_{ON}/C_{OFF}$  of a unit cap. Ideally, we design it so that the output cap is a linear function of the input code, but realistically it's not perfect. A more detailed explanation can be found in [2].

The unit cap is basically an RC circuit with the switch acting as the resistor, so the choice of device type and sizing is of importance. Note that the bulk of the NMOS switch is connected to VSS, a choice very critical to layout as since not doing so will cost us greatly in area, keeping in mind that we have  $3 * 31$  unit caps.

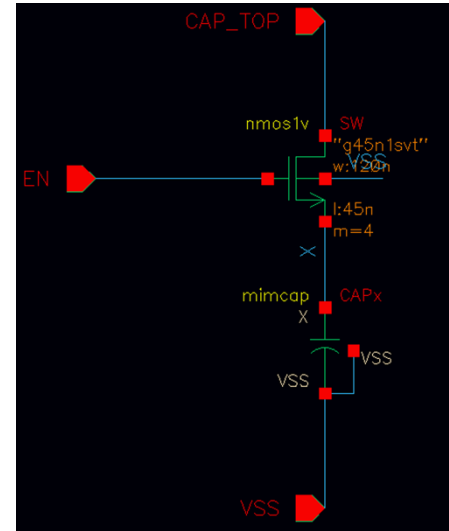


Fig. 3 Unit Cap

Below is the schematic for the final design, note that the output inverters are just buffers of minimum length and width that sharpens the edges.

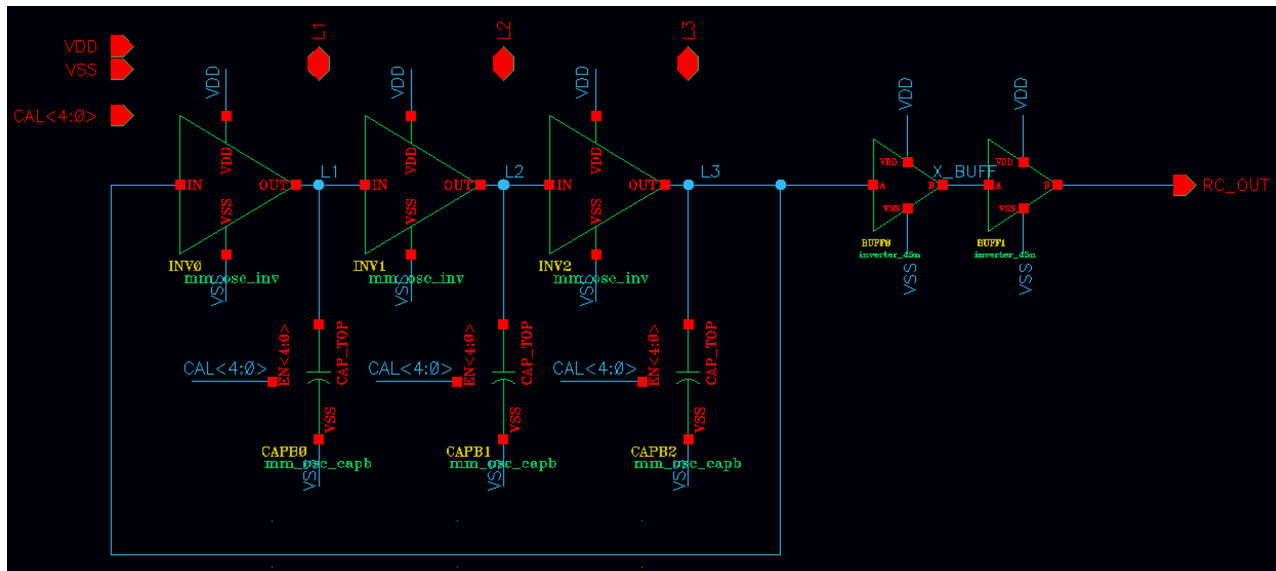


Fig. 4 RC Oscillator Schematic



## Layout Design

Before starting the layout of this design, a few points must be thought of:

1. Dimensions: Preferably a uniform shape, such as square or rectangular. Having a tall or wide block would complicate the process of integrating it with other blocks at the chip level. Exceptions can be made when you have full control over the floorplan.
2. Current consumption: determines how to approach the connections of the supply, how wide they must be to accommodate the influx of current going through without any voltage drop.
3. Bulks: To conserve area, we connect the bulks of the PMOS to VDD and that of the NMOS to VSS. In some cases, we connect the bulk to the source and that's a crucial consideration for the layout. When several bulks are defined, they must be separated by a certain spacing based on the design rules of the PDK.
4. Sensitive nets: certain nets are sensitive to any extra parasitics, either resistance or capacitance can drastically affect the results. For that reason, sensitive nets are identified in the beginning and certain measure are taken to avoid/decrease the effect of parasitics on the final results.
5. Input/Output pins: must be placed where the actual final connection would be and not at a random location on the net. You want to consider all routing parasitics when extracting parasitics and running post layout simulations.

In the following sections, I will go over the layouts of the sub-blocks and the final top layout of this oscillator. I'll list the considerations I have thought of for my approach. Bear in mind that you have multitude of ways you can do a layout, what's important is that the schematic is partitioned into small pieces that can be approached and tested on its own and that's why the schematic design should be modular, better for testing, debugging, layout and post layout fixes.

Two main considerations for the layout of any-sub block:

- Floorplan i.e. placement of the active devices, passive devices and bulks
- Routing i.e. connecting everything together based on the schematic



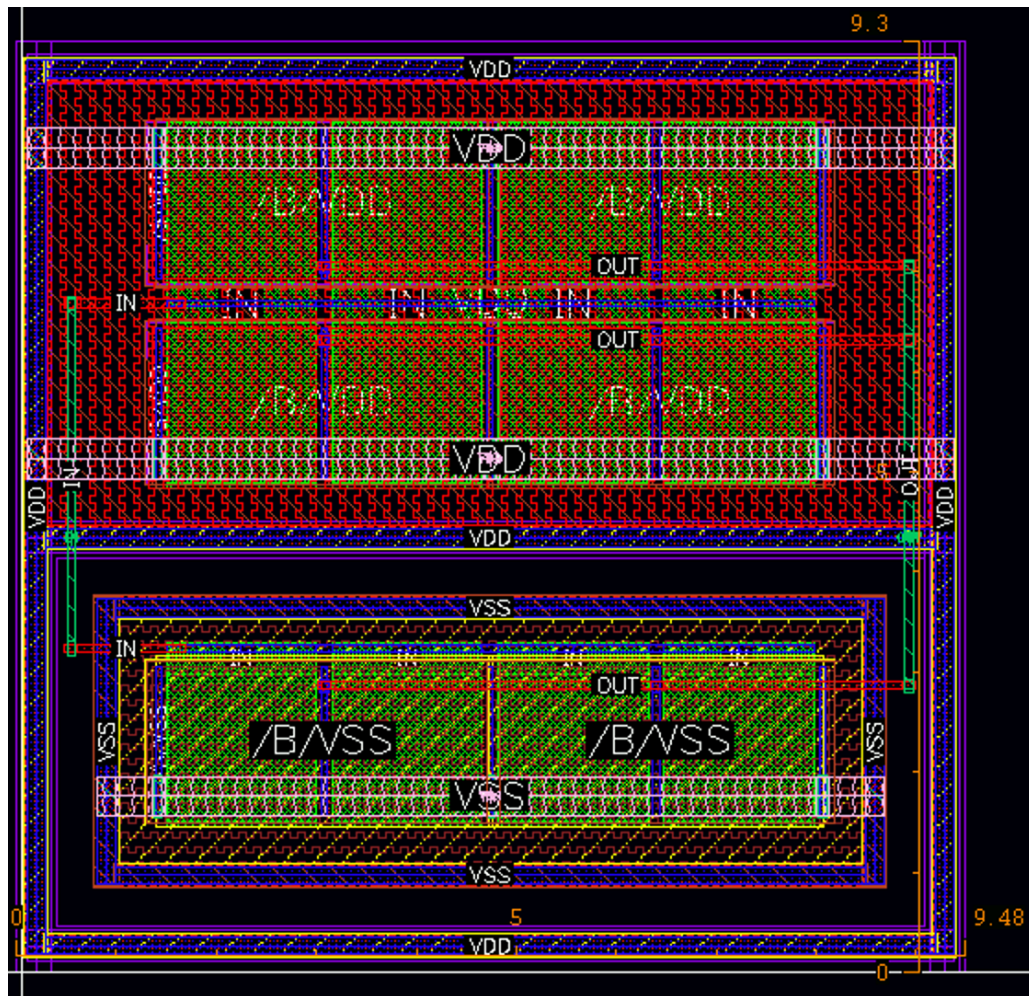
## Layout Design: Main Inverters

### Floorplan:

- For a rectangular shape, the 2 PMOSs (multiplier of 2) are placed above the NMOS.
- Substrate taps are placed surrounding the PMOS and NMOS (based on PDK's design rules).

### Routing:

- Gates of the devices are drawn: for the two PMOSs, their Poly layers are connected together and vias to Metall1 are placed. For the single NMOS, the gate connection is drawn on top of the device.
- Input net is connected vertically to the two extended horizontal gate connections.
- Output net is also connected similar to the input net with horizontal connections connecting to the drains of the devices using vias.
- VDD/VSS connections are drawn horizontally on a higher-level metal with a plan to connect all inverters in the same horizontal manner. Bunks are also connected to their respective nets through vias.



*Fig. 5 Main Inverters Layout*

### Layout Design: Unit Capacitor

#### Floorplan:

- We're limited in how the 4-unit capacitors in series could be placed. If we place them all on a single row, the final shape would be very wide. If we place them all on a single column, the final shape would be very tall, and the connections would add a lot of parasitics. We choose a solution which is in the middle: two capacitors per row.
- The NMOS switch is placed on the top with the proper bulk tap.

#### Routing:

- A solid ground connection is a key consideration. Extending the idea, how can we route it so that it can be shared with the neighboring cell? This is exactly what I did here by routing it to the left of the block.
- CAP\_TOP net is routed horizontally on the switch itself with the minimum width in order to reduce the parasitic capacitance.

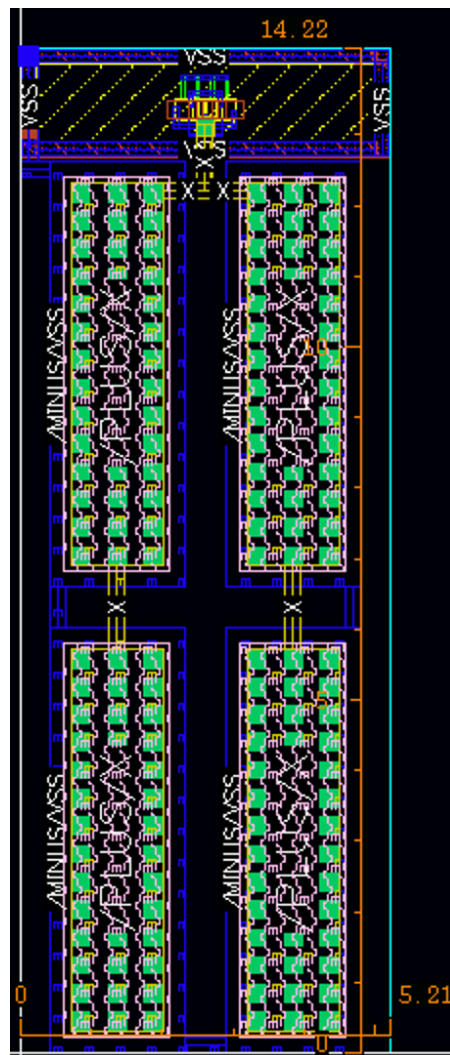


Fig. 6 Unit Capacitor Layout

## Layout Design: Capacitor Bank

### Floorplan:

- Having the final top layout in mind and knowing that three capacitor banks will be placed with 3 inverters, I decided to take that into account at this level and decided to have the main inverters in the middle of the capacitor bank. This means that ideally, the capacitor bank would have the same width as the main inverters.
- I decided to place half of the unit caps corresponding to the MSB at the upper half and the rest of the unit caps at the lower half.
- Fixed caps are arranged between the two halves equally.

### Routing:

- CAP\_TOP net is extended horizontally on all unit caps rows. To connect all rows together, a vertical route is extended at the middle all the way down.
- The ground connection is also extended vertically on a higher metal in anticipation to be connected to the same bus that exists in the main inverters



Fig. 7 Capacitor Bank Layout



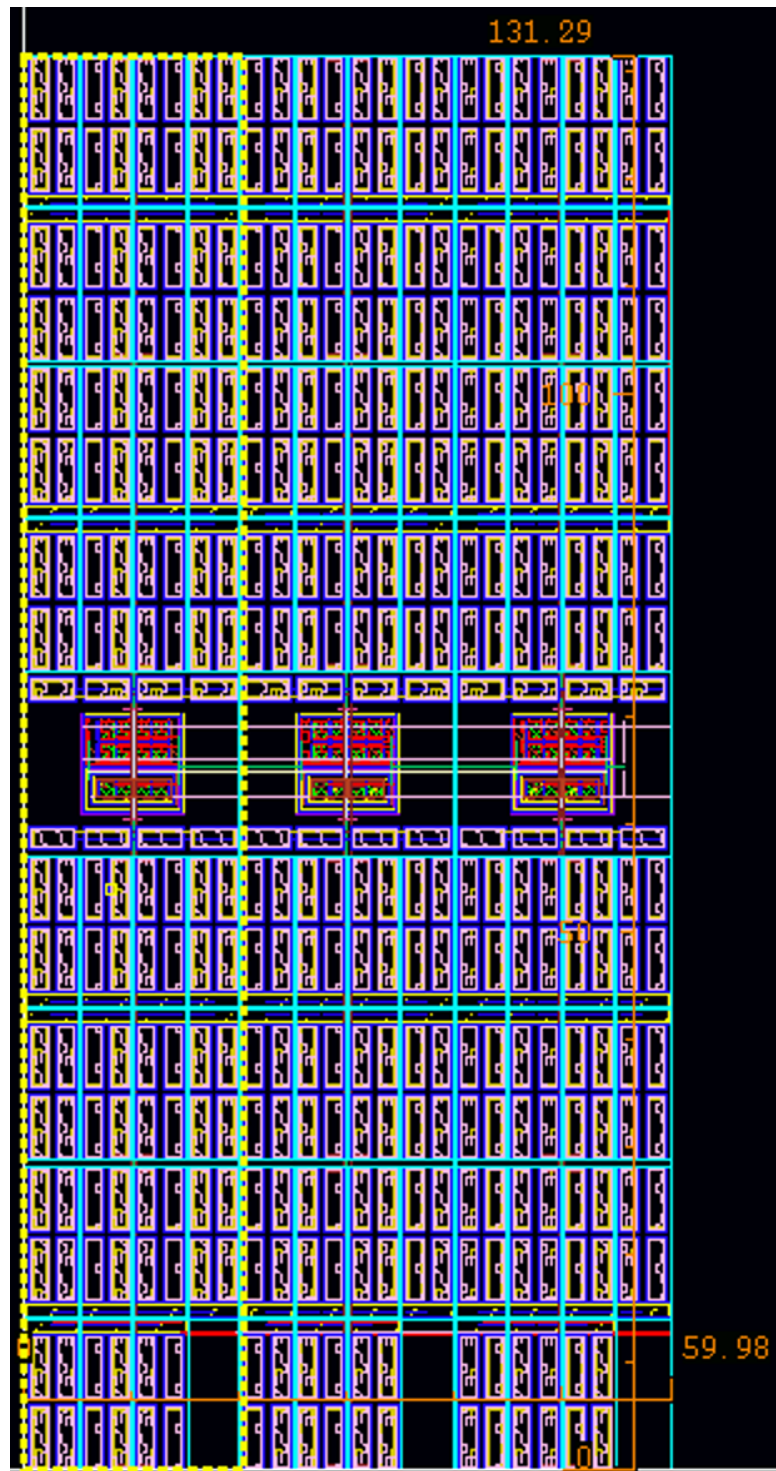
## Layout Design: Oscillator Top

### Floorplan:

- As planned out, the three capacitor banks are placed in a row with the three main inverters in between.
- The two inverters at the output are placed on the right-hand side of the third main inverters cell.

### Routing:

- VDD/VSS nets are extended horizontally. VSS connection of the capacitor bank is connected to the existing bus through vias.
- Input/Output connected the three inverters is now a simple task of extending the nets horizontally. For the last feedback loop, nets are routed from the last inverter cell to the first inverter cell, keeping in mind that parasitic capacitance should be kept at minimum for all these nets.



*Fig. 8 Oscillator Top Layout*

## Oscillator Design Specs

| SPECIFICATION       | VALUE         | NOTES      |
|---------------------|---------------|------------|
| TEMPERATURE         | -40°C to 85°C |            |
| PROCESS             | ff   tt   ss  |            |
| SUPPLY VOLTAGE      | 1V            | ±5%        |
| OUTPUT FREQUENCY    | 60 MHz        | ±5 MHz     |
| DUTY CYCLE          | 50%           | ±10%       |
| CURRENT CONSUMPTION | -             | Max 200uA  |
| CLOCK JITTER        | <10p          | At nominal |

Table 1 Oscillator Design Specs

## Testbench Setup

To be tested, the oscillator is connected in the following manner:

- VDD supply source is a vpwl. Since we're going to run a transient analysis, the oscillator should be kick started by the supply's sharp rise.
- A 10fF ideal capacitor is placed at the oscillator's output to model the output routing to the next block.
- A decimal to binary VerilogA cell is instantiated to be used to control the on/off unit caps. The whole point is to be able to achieve the output frequency spec for different conditions.

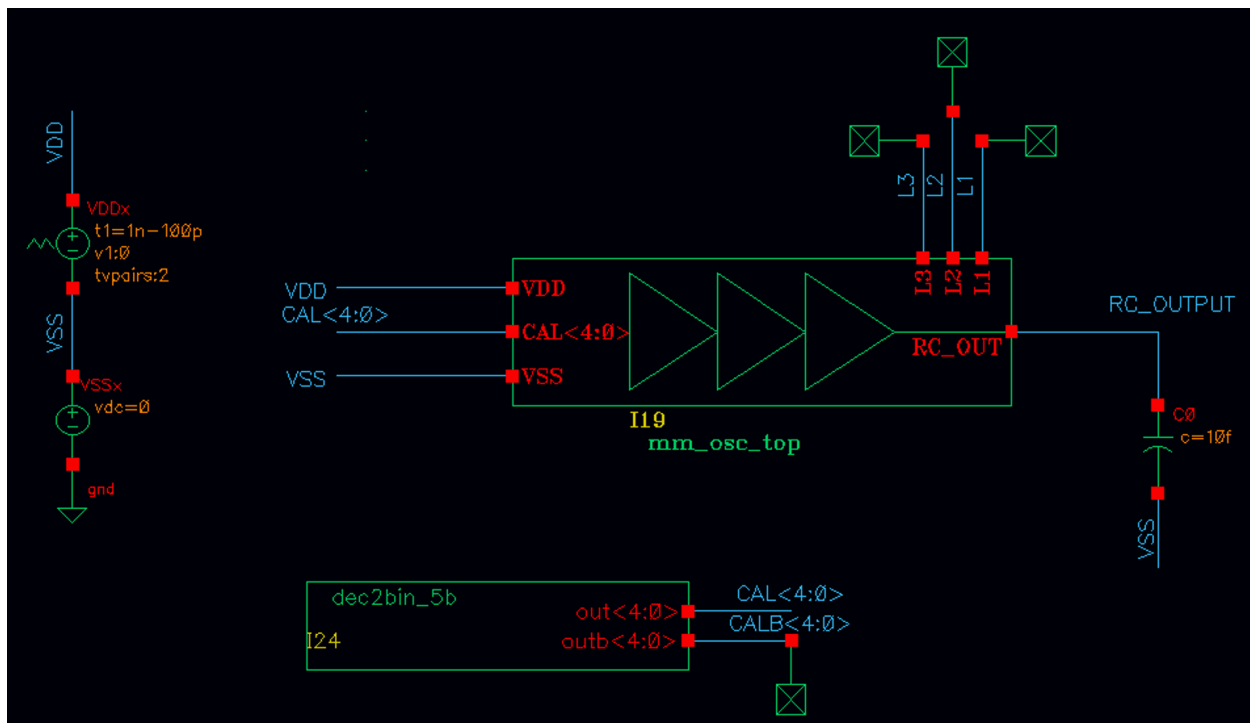
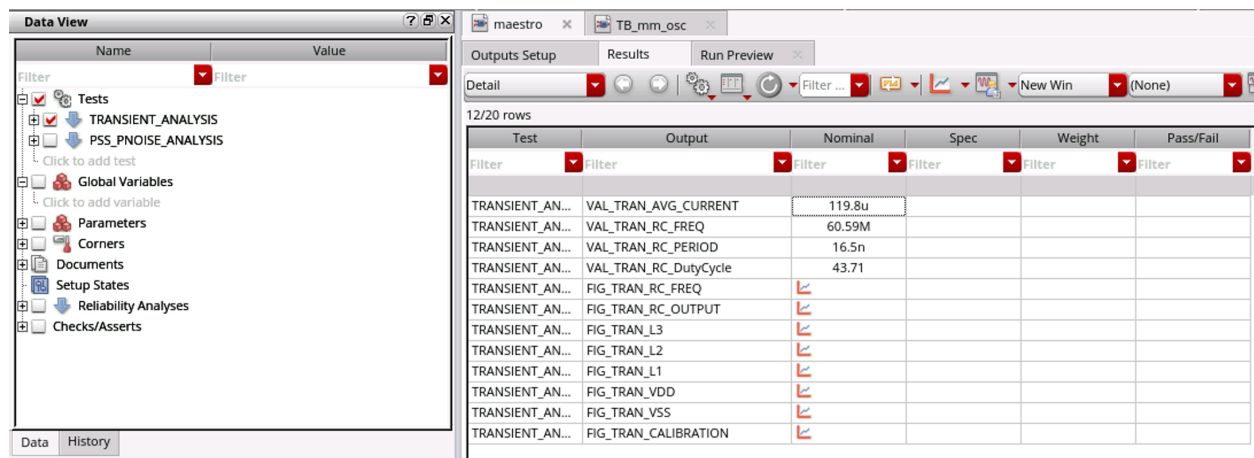


Fig. 9 Testbench Schematic

Fig. 10 shows the ADE Assembler setup for the testbench. Two tests are created:

- TRANSIENT\_ANALYSIS: output frequency, duty cycle and current consumption are computed, and relevant figures are plotted.
- PSS\_PNOISE\_ANALYSIS: a periodic steady state analysis in addition to a phase noise analysis are ran to compute the phase noise of the oscillator and calculate the clock's jitter and cycle to cycle jitter.

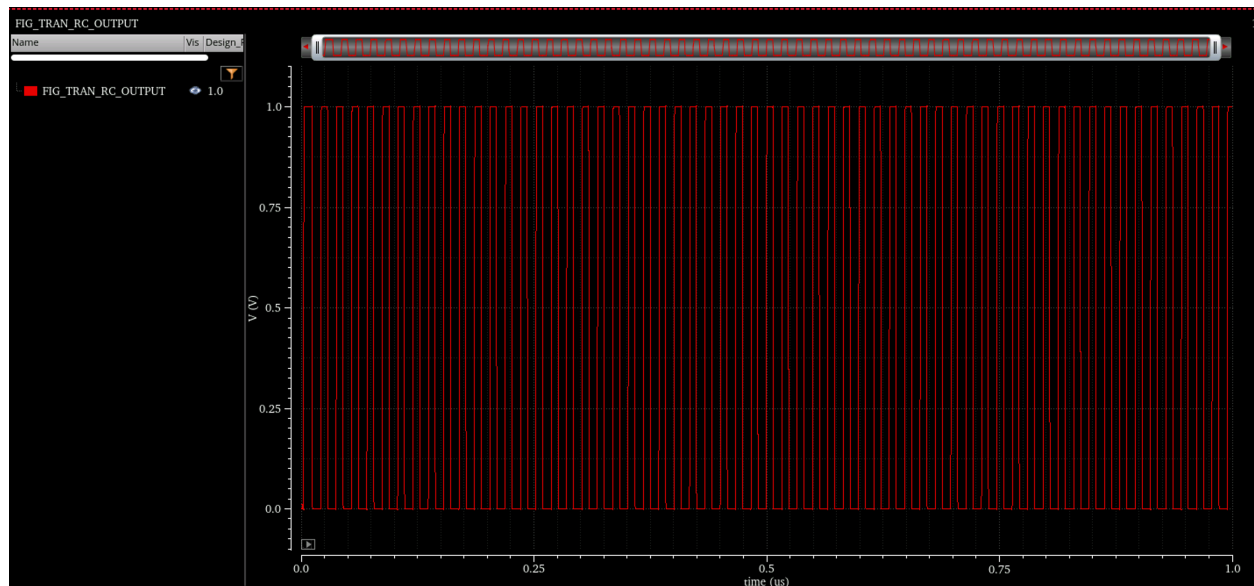


The screenshot shows the ADE Assembler interface. On the left, the 'Data View' pane lists 'Tests' including TRANSIENT\_ANALYSIS and PSS\_PNOISE\_ANALYSIS. On the right, the 'Outputs Setup' pane displays a table of 12/20 rows with columns: Test, Output, Nominal, Spec, Weight, and Pass/Fail.

| Test               | Output                | Nominal | Spec | Weight | Pass/Fail |
|--------------------|-----------------------|---------|------|--------|-----------|
| TRANSIENT_ANALYSIS | VAL_TRAN_AVG_CURRENT  | 119.8u  |      |        |           |
| TRANSIENT_ANALYSIS | VAL_TRAN_RC_FREQ      | 60.59M  |      |        |           |
| TRANSIENT_ANALYSIS | VAL_TRAN_RC_PERIOD    | 16.5n   |      |        |           |
| TRANSIENT_ANALYSIS | VAL_TRAN_RC_DutyCycle | 43.71   |      |        |           |
| TRANSIENT_ANALYSIS | FIG_TRAN_RC_FREQ      |         |      |        |           |
| TRANSIENT_ANALYSIS | FIG_TRAN_RC_OUTPUT    |         |      |        |           |
| TRANSIENT_ANALYSIS | FIG_TRAN_L3           |         |      |        |           |
| TRANSIENT_ANALYSIS | FIG_TRAN_L2           |         |      |        |           |
| TRANSIENT_ANALYSIS | FIG_TRAN_L1           |         |      |        |           |
| TRANSIENT_ANALYSIS | FIG_TRAN_VDD          |         |      |        |           |
| TRANSIENT_ANALYSIS | FIG_TRAN_VSS          |         |      |        |           |
| TRANSIENT_ANALYSIS | FIG_TRAN_CALIBRATION  |         |      |        |           |

*Fig. 10 ADE Assembler Setup*

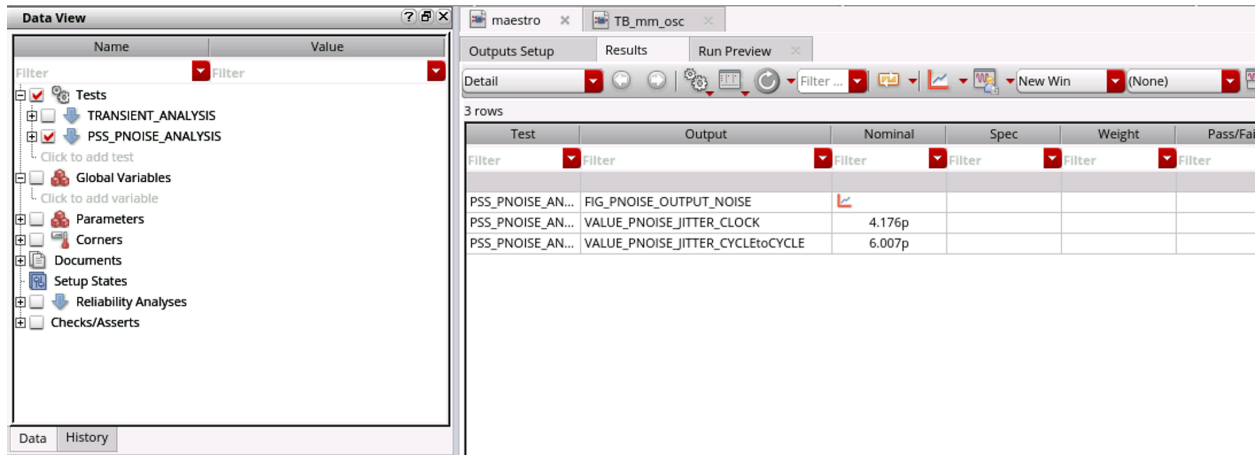
Plotting FIG\_TRAN\_RC\_OUTPUT shows the output of the oscillator: a clean square signal with a maximum value of  $V_{supply}$ . These results are at nominal with a calibration value of 15. This means that 15-unit caps were turned on out of the available 31-unit caps in order to achieve the output frequency of 60.59 MHz.



*Fig. 11 Plot of the Output Clock Net*

*M. El Mokdad | mge41*

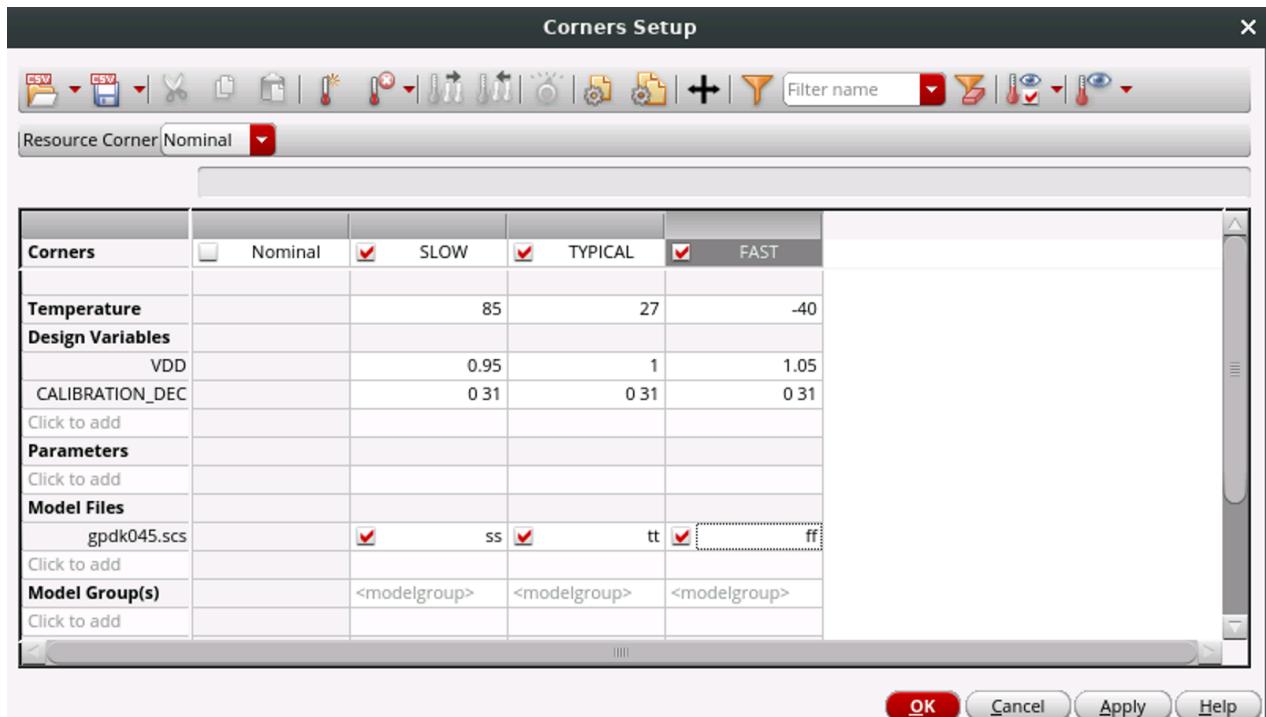
Results of the second test is shown below in Fig. 12. These results correspond to the nominal conditions. In order to run this test under different conditions, the calibration value must be found that yields a 60 MHz output frequency. In the analysis setup, the frequency of 60MHz is provided, operating at another frequency would result in a simulation error (based on the current analysis settings).



| Test                | Output                           | Nominal | Spec | Weight | Pass/Fail |
|---------------------|----------------------------------|---------|------|--------|-----------|
| PSS_PNOISE_ANALYSIS | FIG_PNOISE_OUTPUT_NOISE          |         |      |        |           |
| PSS_PNOISE_ANALYSIS | VALUE_PNOISE_JITTER_CLOCK        | 4.176p  |      |        |           |
| PSS_PNOISE_ANALYSIS | VALUE_PNOISE_JITTER_CYCLEtoCYCLE | 6.007p  |      |        |           |

*Fig. 12 PSS and PNOISE Analysis Results*

Corners are setup as seen in Fig. 13. These are the worst corners (ff process & lowest temperature & highest supply || ss process & highest temperature & lowest supply) in addition to the typical process, nominal voltage and room temperature. Both extreme ends of the calibration range are set so that we can simulate the covered range and make sure a 60 MHz output frequency can be achieved.



| Corners          | Nominal                             | SLOW         | TYPICAL      | FAST         |
|------------------|-------------------------------------|--------------|--------------|--------------|
| Temperature      |                                     | 85           | 27           | -40          |
| Design Variables |                                     |              |              |              |
| VDD              |                                     | 0.95         | 1            | 1.05         |
| CALIBRATION_DEC  |                                     | 0.31         | 0.31         | 0.31         |
| Parameters       |                                     |              |              |              |
| Model Files      |                                     |              |              |              |
| gpd045.scs       | <input checked="" type="checkbox"/> | ss           | tt           | ff           |
| Model Group(s)   |                                     | <modelgroup> | <modelgroup> | <modelgroup> |

*Fig. 13 Corners Setup*



Fig. 14 shows the results of the corners simulations for the design.

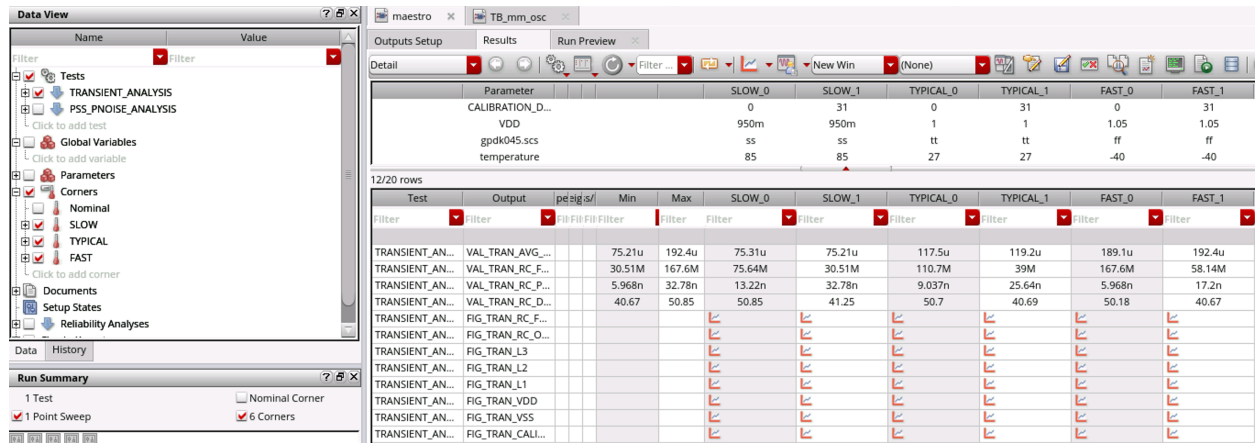


Fig. 14 Schematic Design Corners Results

Fig. 15 shows the results of the corners simulations for the extracted view that is generated using QRC.

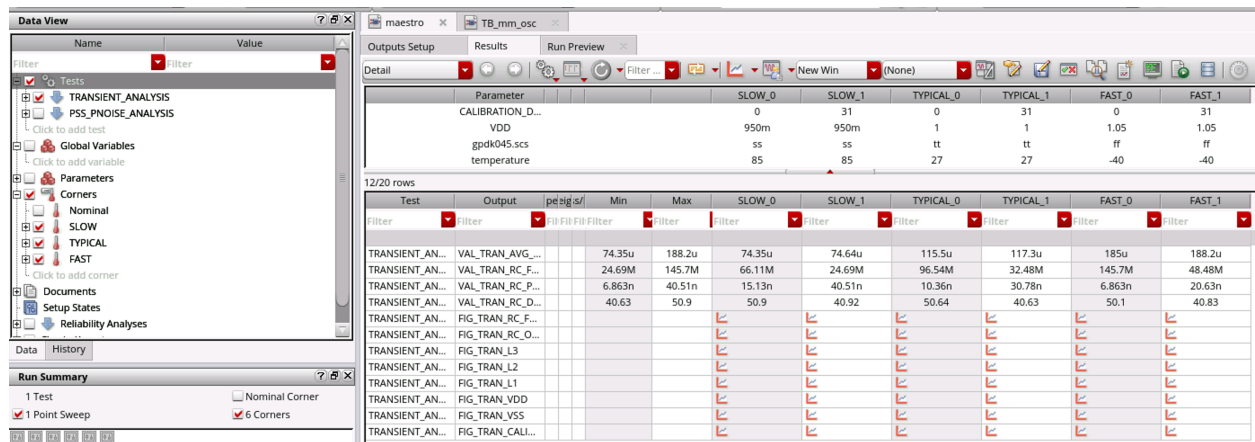


Fig. 15 Extracted View Corners Results

A config view was created and the extraction view named “typical\_092525” was selected to be able to simulate the QRC parasitic extraction. Select “schematic” to simulate the schematic without parasitics. Fig. 16 shows the config view setup.



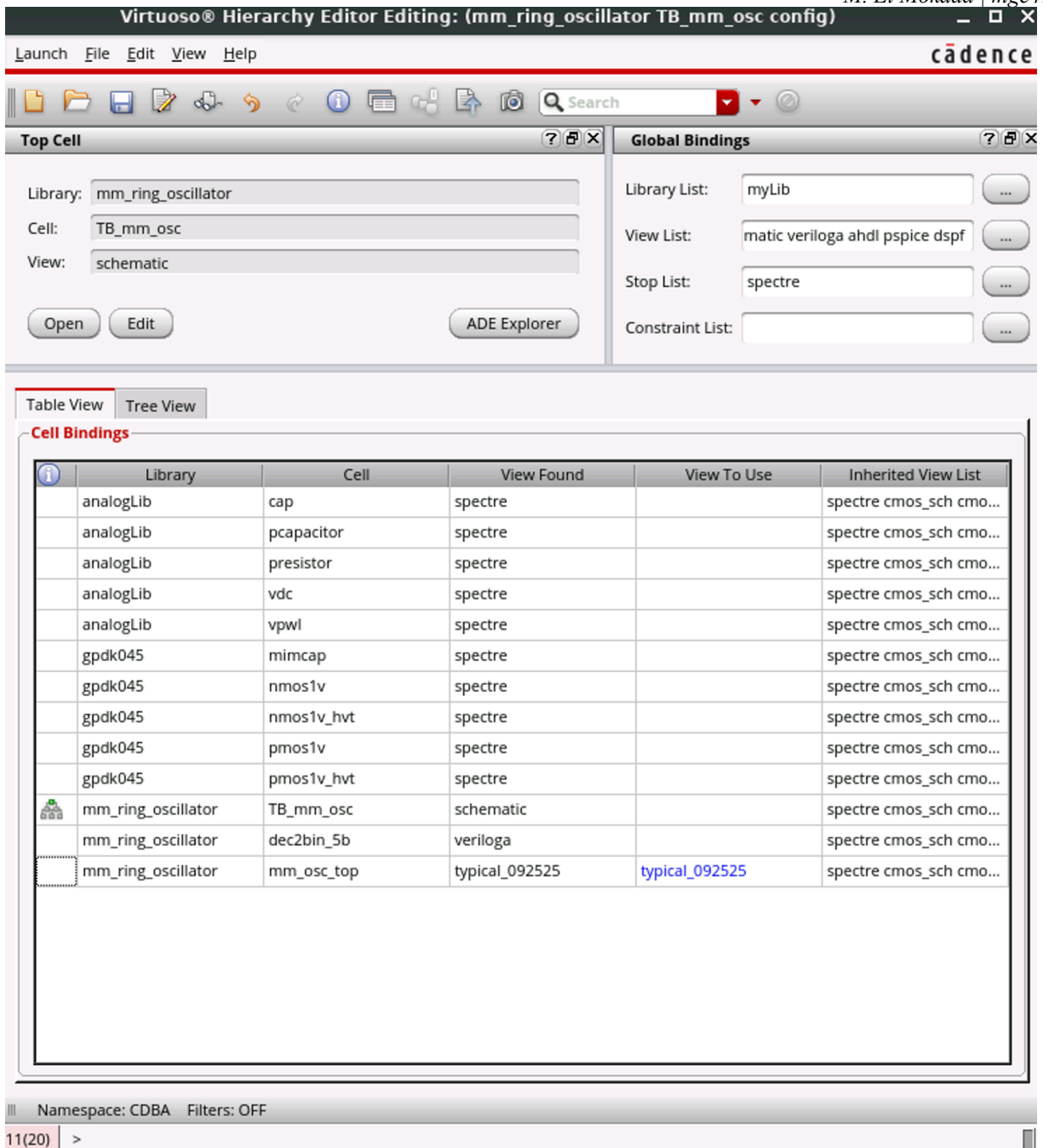


Fig. 16 Config View Setup

## References

- [1] J. G. Atallah and M. Ismail, *Integrated Electronic Circuits*. Springer Nature, 2024.
- [2] Behzad Razavi, *Design of CMOS phase-locked loops : from circuit level to architecture level*. Cambridge, United Kingdom ; New York, Ny: Cambridge University Press, 2020.