Memory management for virtual memory system

Cam Tu Hoang, Tyson Skogerboe, Suwei Yang

University of Utah

Abstract: This project presents the design of an 8-bit virtual memory system consisting of a Memory Management Unit (MMU), Dram and Secondary Storage. The system supports address translation from MMU and Dram, data retrieval from Secondary Storage and fault handling. The report includes simulation result of Register Transfer Level (RTL), logic synthesis result, performance metrics and layout detail of the VLSI of 8-bit virtual memory system. This work demonstrates the implementation of virtual memory in hardware following Cadence RTL-to-GDSII Flow.

I. Introduction

Virtual Memory is a crucial system in computing as it helps automatically transfer data between main memory and secondary storage to give the appearance of unified large memory space [1]. This technique greatly eased programmer tasks, particularly when dealing with data surpassing memory size [1]. This motivates our project to create an 8-bit VLSI implementation of a virtual memory to physical memory converter. The project includes three parts, Memory Management Unit (MMU)[3], Dram, and secondary storage. The system is able to obtain a physical address (PA) and data from a given 8-bit virtual address (VA). Ideally, we will get a page table address from MMU directly. However, several scenarios happen during the process: MMU miss and Dram miss. The system will handle those scenarios by issuing a page fault signal or loading page table from Dram to MMU. This report will show results for simulation, synthesis, performance, and layout.

II. Implementation steps

To implement the design, four steps are necessary. They are Modelsim, synthesis, layout, and output GDS. Modelsim is used to check RTL functionality. When verification goes well, RTL codes should be frozen. Synthesis tool (genus) then turns RTL code into gate-level netlist RTL. Some IO pads are added to gate-

level netlist to provide interfaces for users. Innovus then uses gate-level netlist with IO pads to do the layout/ floorplan/ routing… etc. Designers are supposed to see the chip schematic at this point. Finally, the designed chip will go through Virtuoso for design rule check (DRC)/ layout versus schematic (LVS). When everything goes well, a basic chip is built. Before producing GDS, the chip needs sealring, polyfill, and metal fill for moisture, sawing, and density requirements. Once those things above are done and DRC/LVS are clean, Virtuoso produces a GDS file for manufacture.

III. Modelsim simulation

Three parts are verified before synthesis, i.e MMU, Dram, storage. Four things need to be verified in MMU: content-addressable memory(cam) translation, cache translation, cache and cam interaction, and pagefault. Cam translation is verified as shown in figure 1 top. By sending VA (00010100) into cam, it is able to translate the PA, 0xae. Cache translation is verified as shown in figure 1 Modelsim second row. By sending VA(00010100), cache translates the correct PA(0xae) after several cycles. Cache and cam interaction is verified as shown in figure 1 third row. In the beginning, no data was in cam (left side, dut2/storage). After sending data from cache, cam will see the data (right side, dut/storage). Pagefault is verified as figure 1 bottom. By sending invalid VA, both cam and cache report pagefault.

For the Dram Modelsim simulation, initially, Dram was empty and then it is populated with data from the input array through the write command, accurately storing data into Dram's memory space as shown in figure 2 and figure 3.

The system first checks cam to translate the virtual address to PA. If Cam lookup fails, the system proceeds to check Cache for obtaining a PA. If Cache also results in a miss, the system then fetches a PA from Dram. If any of the Cache, Cam, Dram hits, the corresponding PA is forwarded to memory storage for data access. However, if all levels result in a miss, the system sends a page fault signal, indicating that there is no PA corresponding to the input VA, as shown in figure 2-3. The memory storage module was modelled like a flash memory storage with some variations. It consists of parallel input and output data lines

[7:0], process ID pins [3:0], process valid pins, and clock and reset pins. This module uses a simple state machine to maintain functionality with states including idle, read, write, and erase. The state machine triggers upon the positive edge clock signal. Other important flags set within the module are cleared and monitored under separate conditions that trigger as they arise, rather than by clock signal. Another function of the machine is that it requires input from an outside user to initialize any read, write, or erase functions. Similarly, it requires user input to end those functions to be able to run other operations of read, write, and erase as necessary. Individual testing of this secondary storage module yielded appropriate results and correct functionality. Figure 4 demonstrates the results of the secondary storage and its desired functionality. After verifying the secondary storage validity, it is implemented in the top-level module including the Dram and MMU modules. The entire system is tested in Modelsim by simulating the flow of data through memory hierarchy, mimicking the operating system's attempt to read data.

IV. Logic Synthesis, P&R, Sealring and poly&metal fill

After verifying the function of Verilog module, it is synthesized using Genus to obtain the Gate-level netlist. With this gate-level netlist, Innovus is used to implement the system into a physical layout. The gate-level netlist input and output signals are then connected with pin pads to enable interfacing with external components. Following the Cadence RTL-to-GDSII flow, the physical layout is generated through place and route (P&R) operation including floor planning, standard cell placement, clock tree synthesis and routing using the Innovus tool. Once the layout is obtained, it is verified through both DRC and LVS to ensure the circuit meets manufacturing requirements and logical consistency with the original design. Figure 6 shows the clean DRC and LVS results of the circuit after synthesis. The power of the final design is found to have leakage of 3800 nW, and a total power of 3900 uW as seen in Figure 5. For timing analysis, Figure 5 also shows that the final slack was found to be 86,684 ps. Positive slack indicates that critical paths are within timing constraints, or the design meets timing requirements to operate accurately with a clock frequency of 10 MHz. Lastly, Figure 5 shows the area of the final layout is reported to be 2.299103 mm^2 including the

I/O pads, which is within 2mmx2mm and will fit inside the sealring. Final layout of the chip prior to adding sealring, polyfill and metal-fill is shown in Figure 7. Then, after incorporating sealring, polyfill and metal-fill, the design passed full-chip DRC and LVS shown in Figure 8 indicating the design met density requirements. Finally, Figure 9 shows a standalone view of the final layout, and the respective GDS file was exported for fabrication.

V. Conclusion

The project successfully implements a 8-bit virtual memory in VLSI using Cadence RTL-to-GDSII flow. The design including MMU, Dram and Secondary Storage demonstrates a correct VA to PA translation, data retrieval and page fault handling. The physical layout meets area, timing and density requirements and passes full-chip DRC and LVS checks.

VI. Reference

[1] B. Jacob and T. Mudge, "Virtual memory: issues of implementation," in Computer, vol. 31, no. 6, pp. 33-43, June 1998, doi: 10.1109/2.683005.

[2] GeeksforGeeks, “What is flash memory?,” GeeksforGeeks, https://www.geeksforgeeks.org/what-is-flash-memory/ (accessed Nov. 27, 2024).

[3] Memory management unit. (2024, Oct 27). In *Wikipedia*. https://en.wikipedia.org/wiki/Memory_management_unit

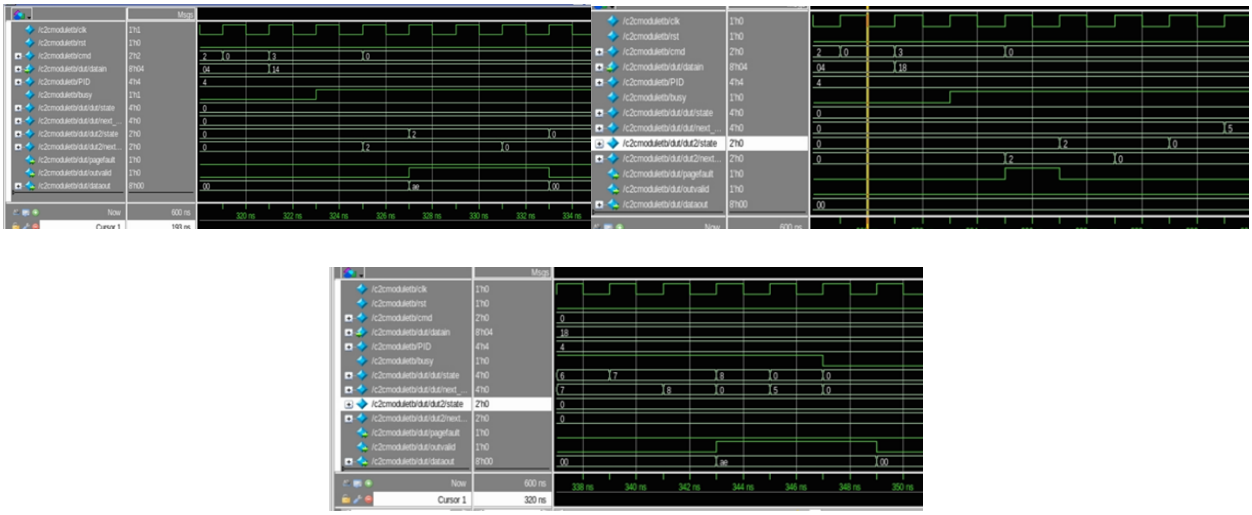
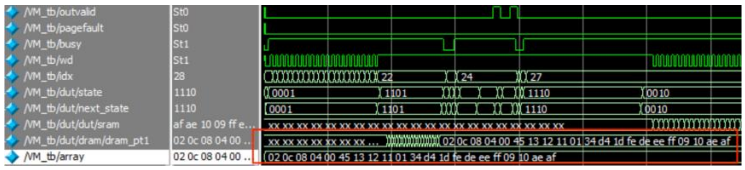
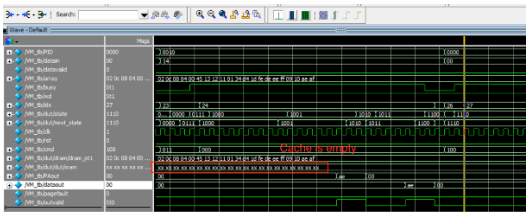
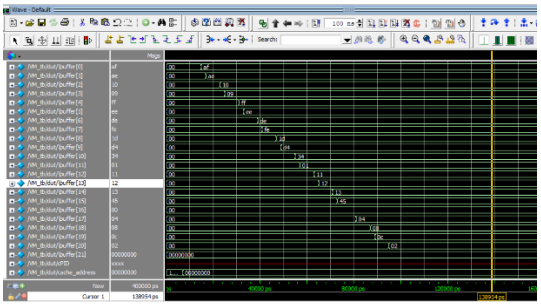
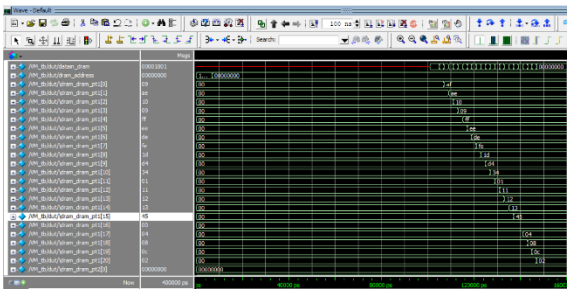


Figure 1. MMU modelsim









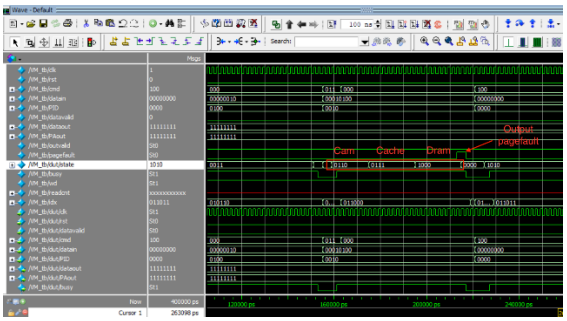


Figure 2. DRAM modelsim1

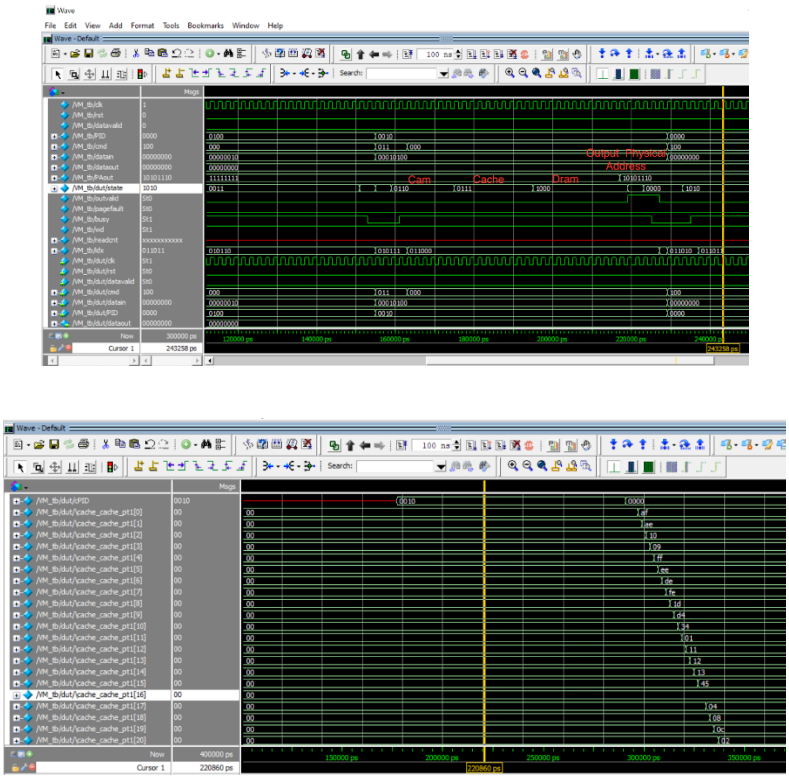


Figure 3. DRAM modelsim2

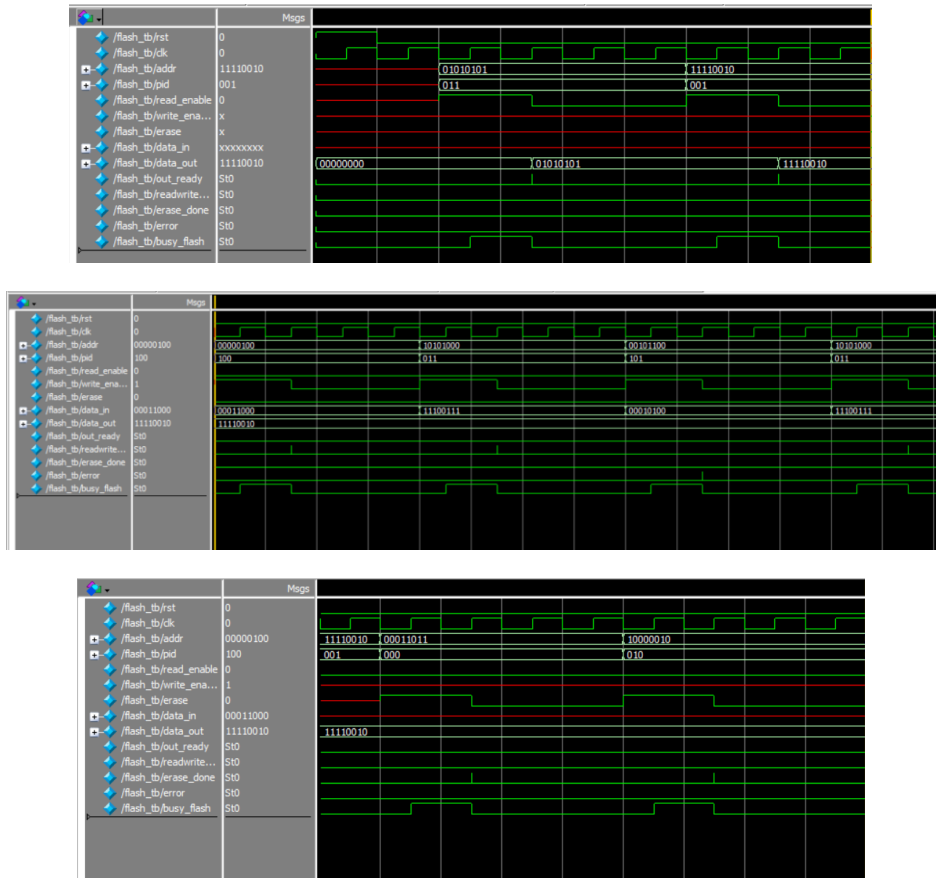
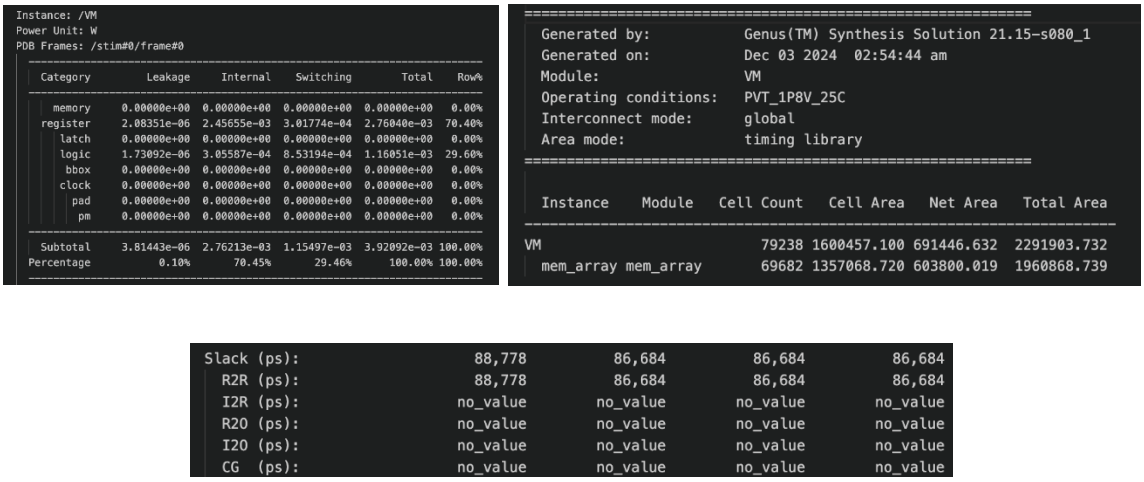


Figure 4. Secondary Storage Modelsim



```
Instance: /VM
Power Unit: W
PDB Frames: /stim#0/frame#0
-------------------------------------------------------------------------
  Category         Leakage     Internal    Switching        Total    Row%
-------------------------------------------------------------------------
    memory     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
  register     2.08351e-06  2.45655e-03  3.01774e-04  2.76040e-03  70.40%
     latch     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
     logic     1.73092e-06  3.05587e-04  8.53194e-04  1.16051e-03  29.60%
      bbox     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
     clock     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
       pad     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
        pm     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
-------------------------------------------------------------------------
  Subtotal     3.81443e-06  2.76213e-03  1.15497e-03  3.92092e-03 100.00%
Percentage           0.10%       70.45%       29.46%      100.00% 100.00%
-------------------------------------------------------------------------
```

```
============================================================
  Generated by:           Genus(TM) Synthesis Solution 21.15-s080_1
  Generated on:           Dec 03 2024  02:54:44 am
  Module:                 VM
  Operating conditions:   PVT_1P8V_25C
  Interconnect mode:      global
  Area mode:              timing library
============================================================

  Instance    Module   Cell Count   Cell Area   Net Area   Total Area
-----------------------------------------------------------------------
VM                         79238 1600457.100 691446.632  2291903.732
  mem_array mem_array      69682 1357068.720 603800.019  1960868.739
```

Slack (ps):	88,778	86,684	86,684	86,684
R2R (ps):	88,778	86,684	86,684	86,684
I2R (ps):	no_value	no_value	no_value	no_value
R2O (ps):	no_value	no_value	no_value	no_value
I2O (ps):	no_value	no_value	no_value	no_value
CG (ps):	no_value	no_value	no_value	no_value

Figure 5. Timing, Area and Power Report

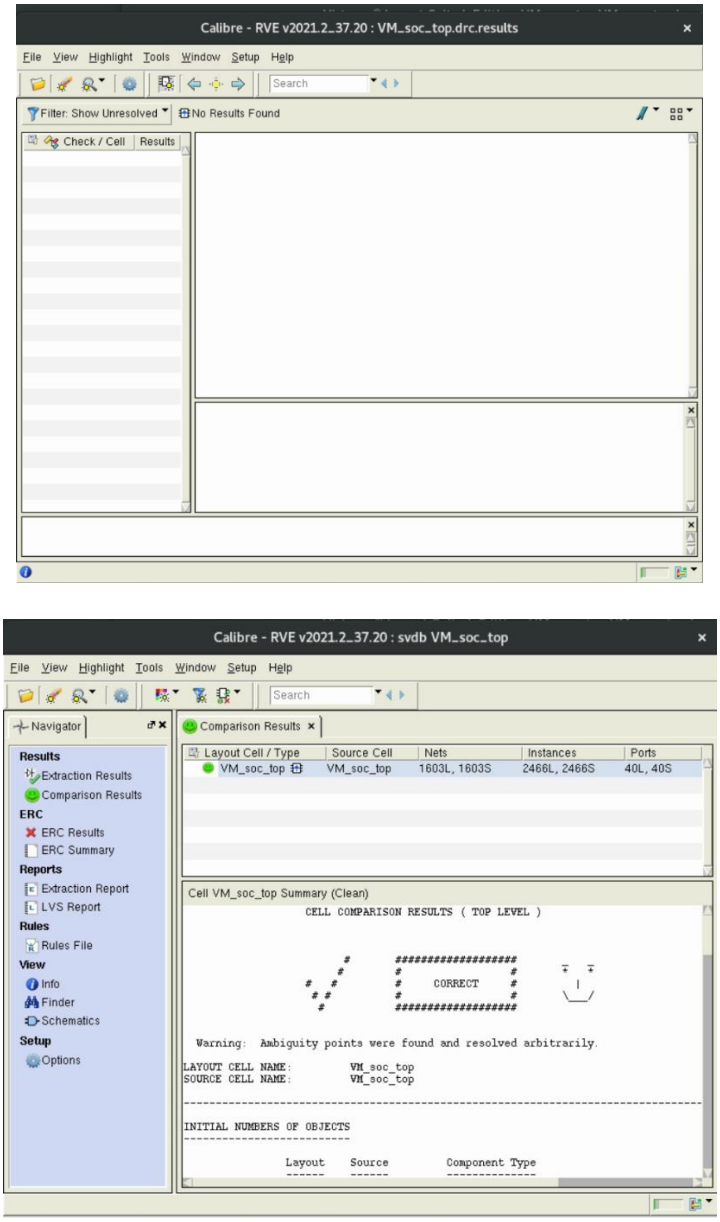


Figure 6. DRC & LVS pass after synthesis

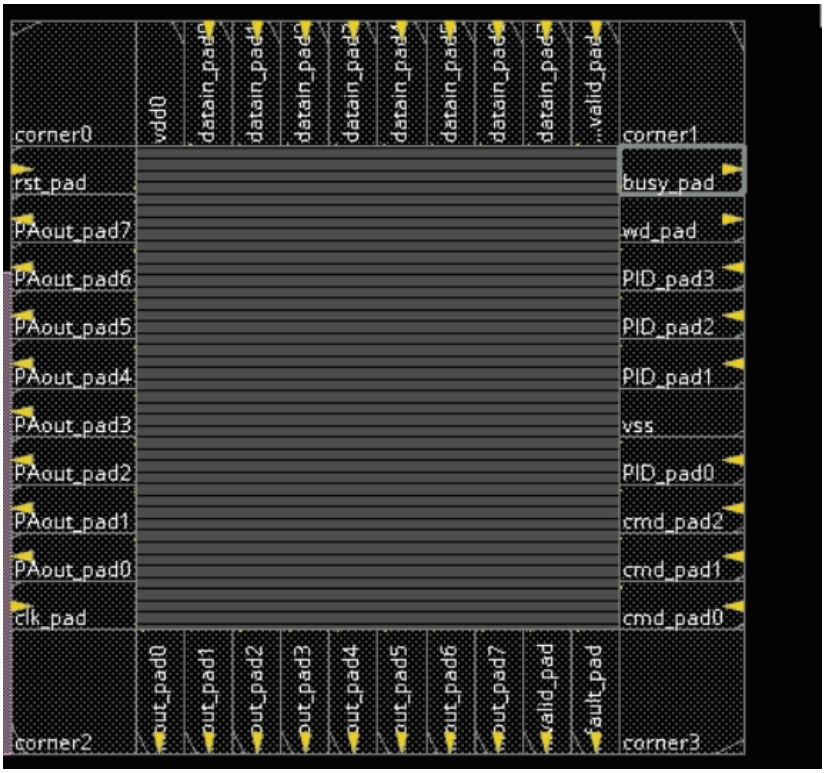


Figure 7. Design prior to running Innovus scripts.

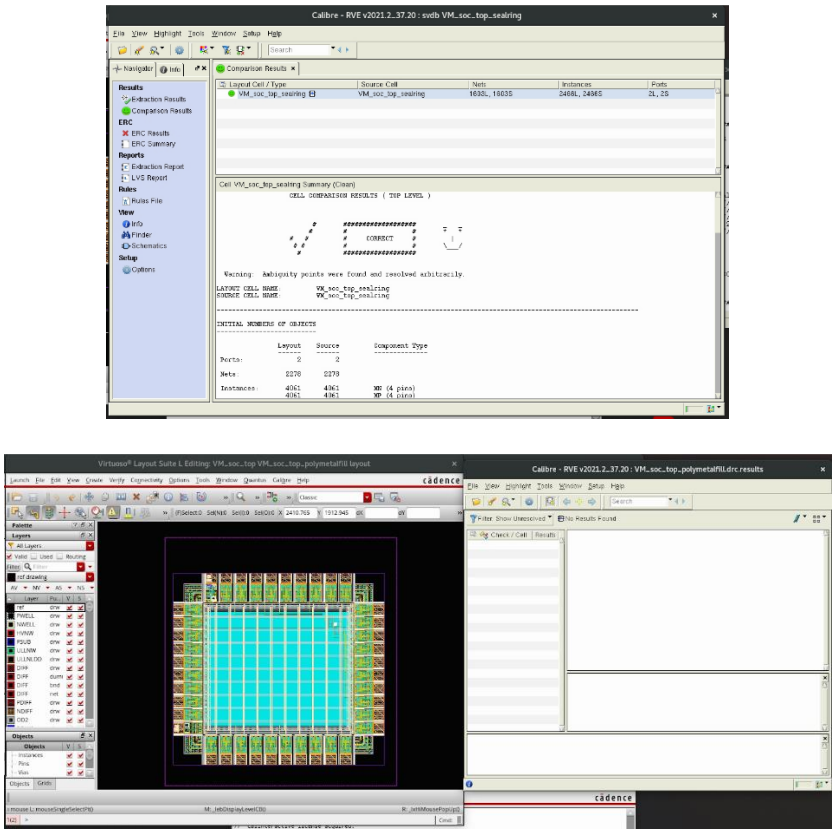


Figure 8. DRC & LVS pass after poly&metal fill and sealring

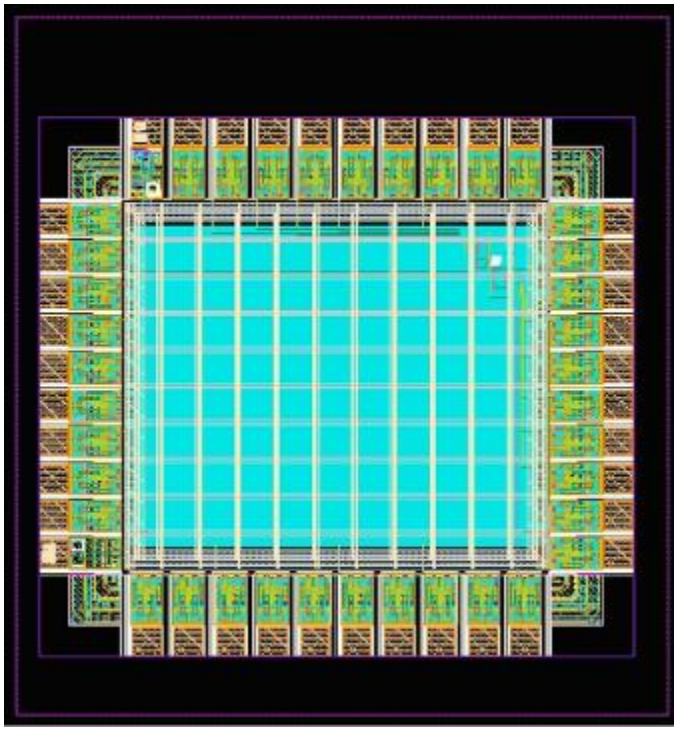


Figure 9. Final Chip Layout