המכללה האקדמית להנדסה בראודה בכרמיאל

המחלקה להנדסת חשמל ואלקטרוניקה

Introduction to VLSI 2025A

Introduction to Schematic Entry

S-RAM Final Project

פרנסיס עבוד

בשארה חביב

מריה נחלה

נור זכור

מרצה: אלכסנדר סבטליצה

תאריך: 18/01/2025

1. Introduction

In this documentation, we delve into the complete design flow of an **8×8-bit Static Random-Access Memory (SRAM)** in Cadence Virtuoso. The goal is to illustrate each component's purpose, how it is implemented at the transistor level, and how these components integrate into the final memory system.

1. **Project Overview**
    - Provides a high-level synopsis of why we need an 8×8-bit SRAM, what applications it might serve, and the key design goals (e.g., low power, reliability, and efficient layout).
2. **Design Flow**
    - Outlines the progression from basic building blocks (inverters, logic gates) to more complex structures (decoders, 6T cells), leading to the final SRAM array.
3. **Objectives**
    - Highlight the learning outcomes, such as mastery of CMOS transistor-level design, layout practices, and verification through simulation.

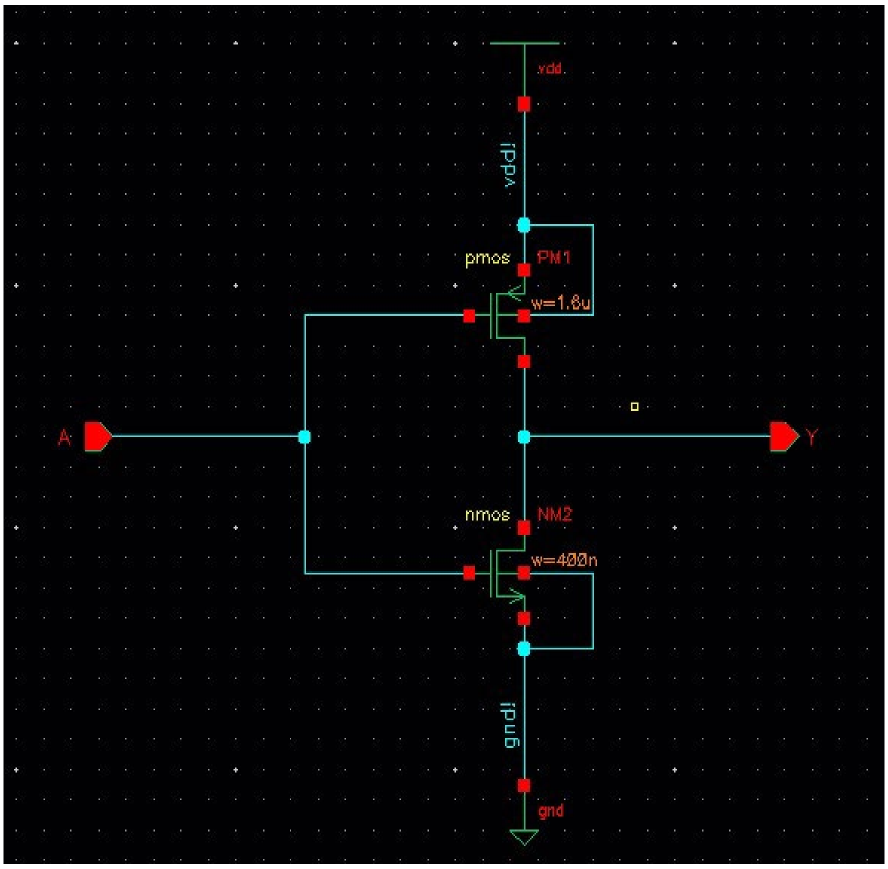
---

2. Inverter (INV)

The **Inverter** is the fundamental logic element in CMOS design. It flips the input logic level—if the input is high, the output goes low, and vice versa. In many circuits, inverters are also used as buffers, drivers, or within more complex gates.

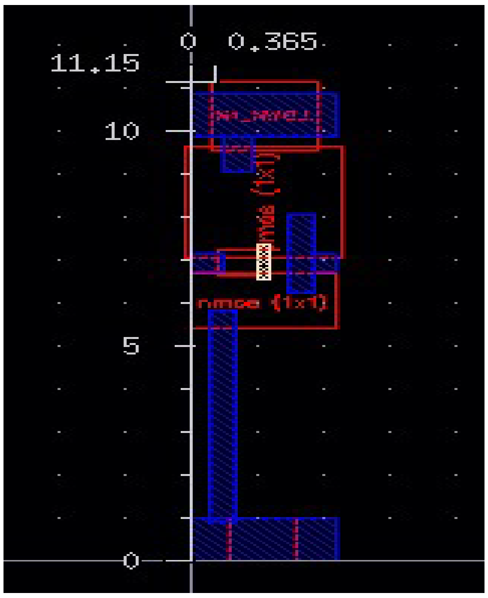
2.1 Schematic

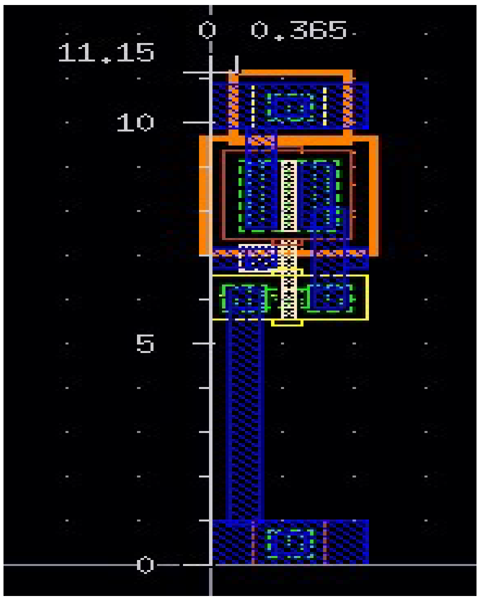
- **Explanation:** Shows the PMOS and NMOS transistors connected in a complementary arrangement.
- **Key Points:**
    - High input turns the NMOS on and PMOS off, driving output low.
    - Low input turns the NMOS off and PMOS on, driving output high.



2.2 Layout

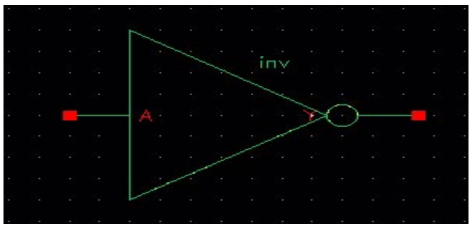
- **Explanation:** Illustrates the physical arrangement of the PMOS and NMOS transistors.
- **Key Points:**
    - Ensures minimal area while respecting design rules (DRC).
    - Properly routes VDD, GND, and input/output signals.





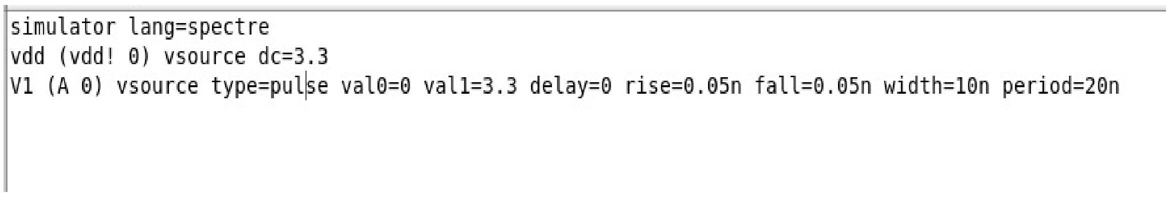
2.3 Symbol

- **Explanation:** A simplified black-box representation of the inverter used in higher-level schematic designs.
- **Key Points:**
    - Clearly labeled input (A) and output (Y).
    - Used wherever an inverter is needed in larger circuits.



2.4 Stimulus File

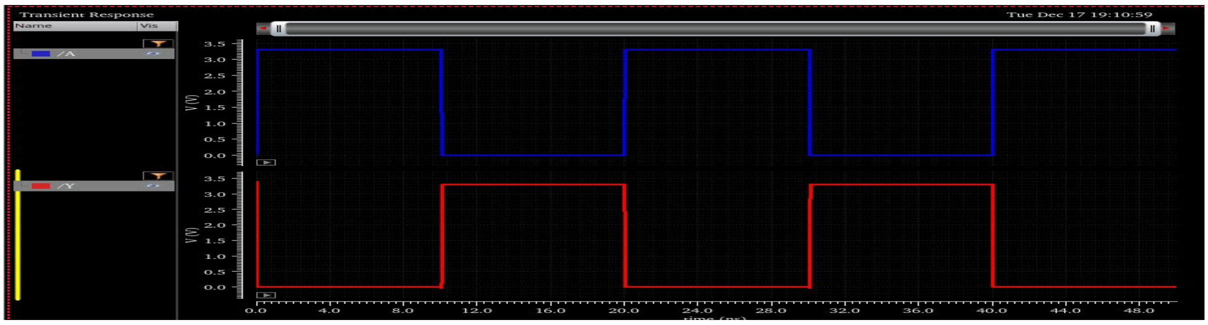
- **Explanation:** Demonstrates how the input signal is driven (e.g., a square wave) in the simulator.
- **Key Points:**
    - Defines voltage levels, simulation time, and source parameters.
    - Allows consistent testing of the inverter's functionality across different conditions.



```
simulator lang=spectre
vdd (vdd! 0) vsource dc=3.3
V1 (A 0) vsource type=pulse val0=0 val1=3.3 delay=0 rise=0.05n fall=0.05n width=10n period=20n
```

2.5 Simulation Graph/Output

- **Explanation:** Displays the waveforms showing the inverter's response.
- **Key Points:**
    - Confirms the output is the logical NOT of the input.
    - Helps measure critical parameters like propagation delay and power consumption if needed.



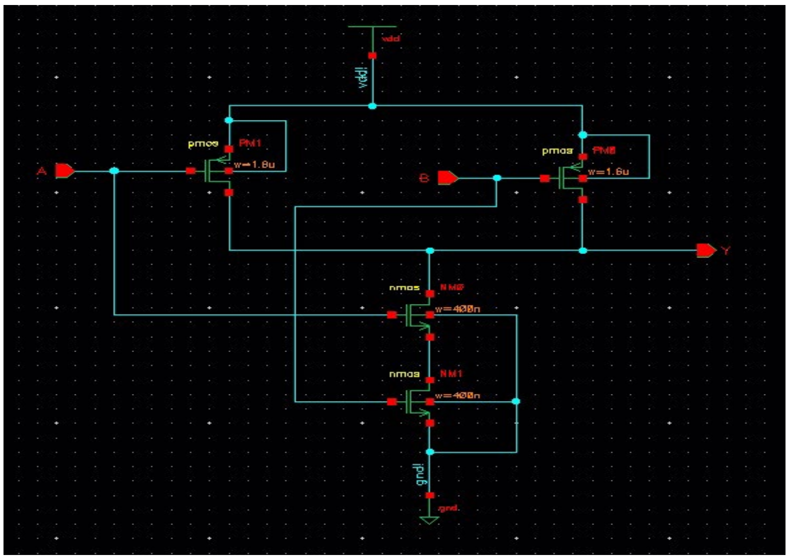
---

3. NAND2 Gate

A **2-input NAND Gate** outputs a logic LOW only when both inputs are HIGH. NAND is a universal gate—meaning you can build any digital system using only NAND gates.

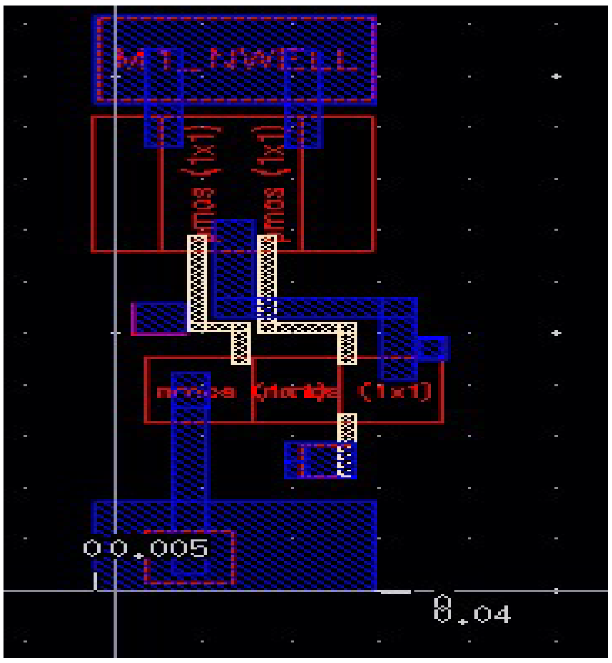
3.1 Schematic

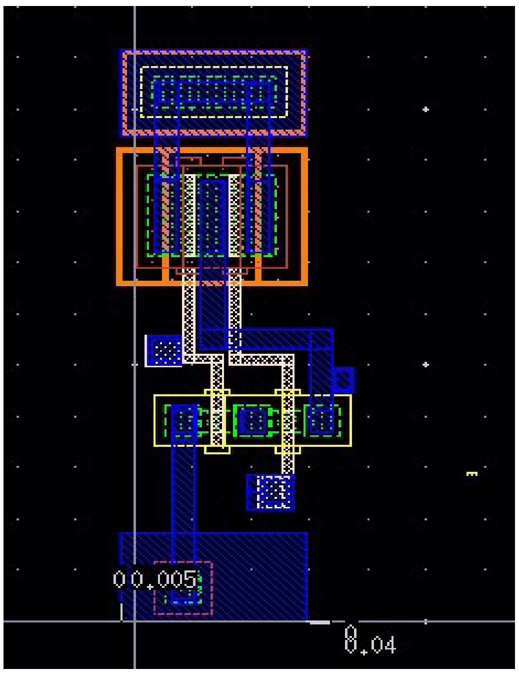
- **Explanation:** Shows two PMOS in parallel and two NMOS in series.
- **Key Points:**
    - When all inputs are HIGH, the NMOS stack conducts strongly, pulling the output LOW.
    - Any LOW input will break the NMOS path or enable the PMOS path, keeping output HIGH.



3.2 Layout

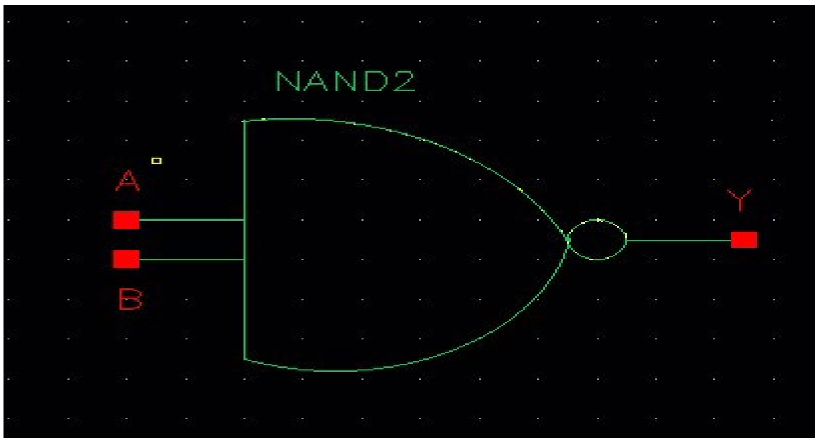
- **Explanation:** Depicts the physical placement of four transistors (2 PMOS, 2 NMOS).
- **Key Points:**
  - Emphasizes symmetrical layout for matching where possible.
  - Minimizes connection lengths between transistors.





3.3 Symbol

- **Explanation:** Provides a straightforward representation of the NAND gate for schematic integration.
- **Key Points:**
  - Two inputs (A, B) and one output (Y).
  - Often used in combinational logic blocks, decoders, and more.



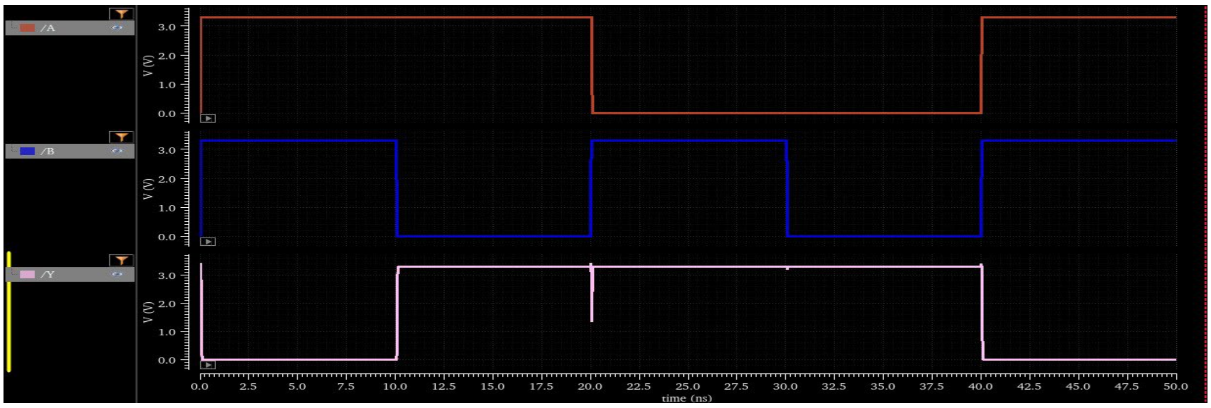
3.4 Stimulus File

- **Explanation:** Describes the test setup applying all input combinations (00, 01, 10, 11).
- **Key Points:**
  - Ensures correctness of each output state under all logical conditions.
  - May include timing analysis if required.

```
simulator lang=spectre
vdd (vdd! 0) vsource dc=3.3
V1 (A 0) vsource type=pulse val0=0 val1=3.3 delay=0 rise=0.05n fall=0.05n width=20n period=40n
V2 (B 0) vsource type=pulse val0=0 val1=3.3 delay=0 rise=0.05n fall=0.05n width=10n period=20n
```

3.5 Simulation Graph/Output

- **Explanation:** Showcases simulation waveforms validating the NAND truth table.
- **Key Points:**
  - Verify that output is LOW only when both inputs are HIGH.
  - Check transitions for correct timing.



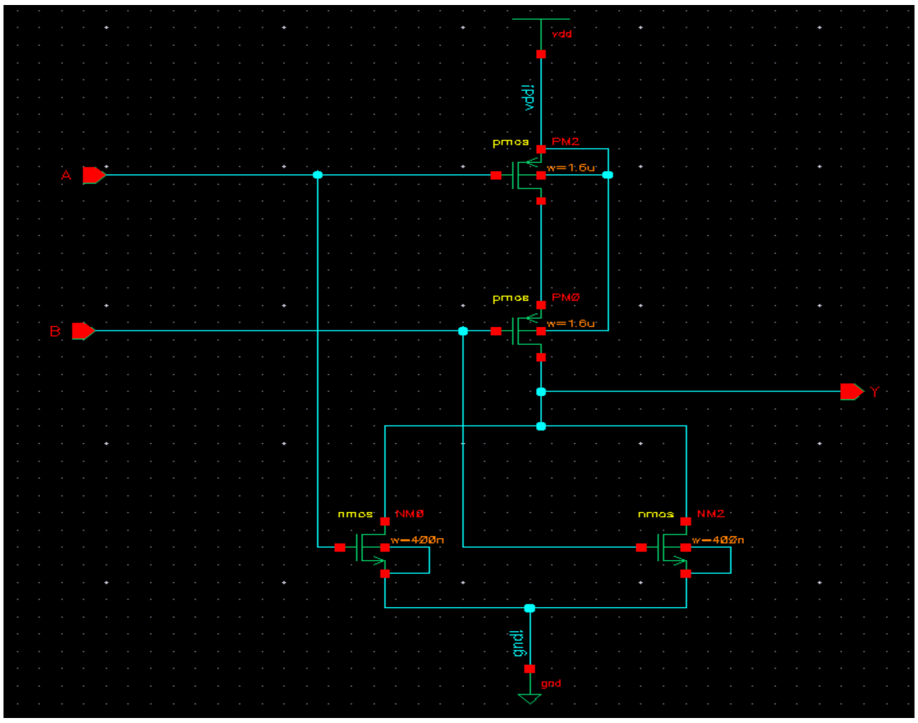
---

4. NOR2 Gate

A **2-input NOR Gate** outputs a logic HIGH only when both inputs are LOW. Like NAND, NOR is also a universal gate.

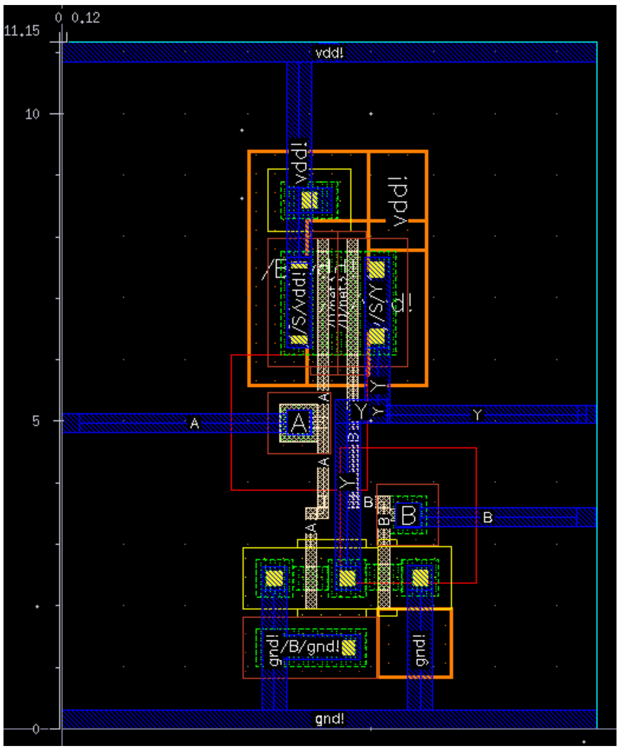
4.1 Schematic

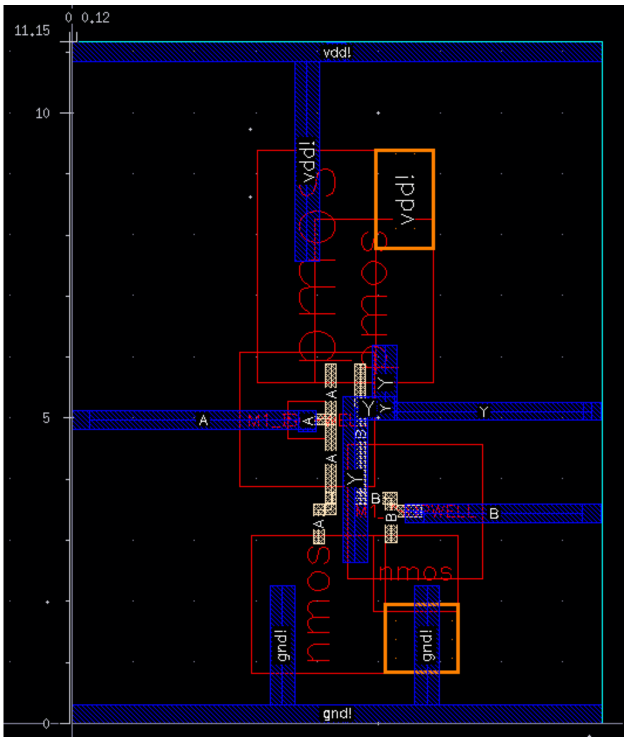
- **Explanation:** Features two PMOS in series and two NMOS in parallel.
- **Key Points:**
  - NOR logic is complementary to NAND logic.
  - Useful in decoder design and control logic blocks.



4.2 Layout

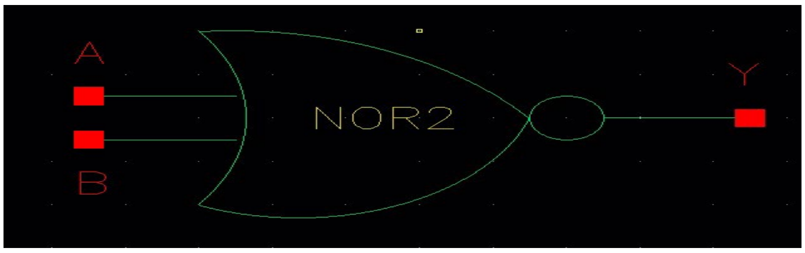
- **Explanation:** Shows the transistor floorplan in silicon.
- **Key Points:**
  - Similar design considerations to NAND (matching devices, minimizing parasitic).
  - Ensures signals are routed with minimal crosstalk.





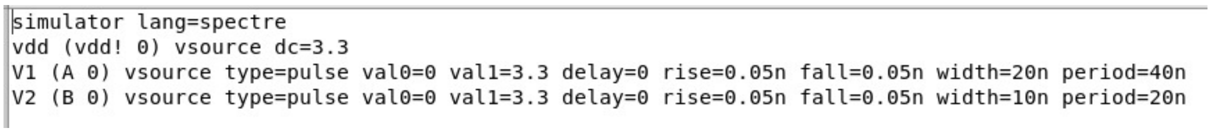
4.3 Symbol

- **Explanation:** Provides the gate-level depiction of NOR.
- **Key Points:**
    - Two inputs (A, B), one output (Y).
    - Standard symbol usage ensures consistency across designs.



4.4 Stimulus File

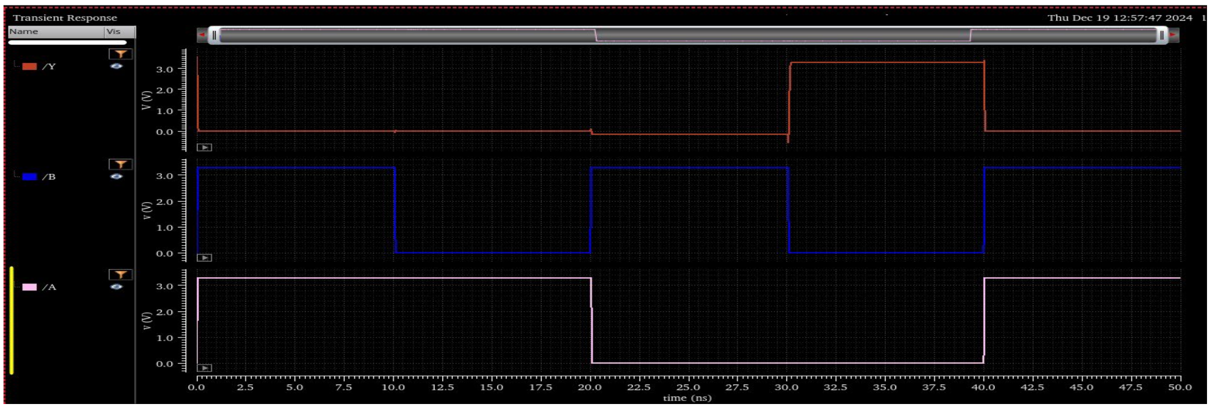
- **Explanation:** Lists the test input patterns (00, 01, 10, 11).
- **Key Points:**
    - Thoroughly tests all input combinations to validate correct logic operation.



```
simulator lang=spectre
vdd (vdd! 0) vsource dc=3.3
V1 (A 0) vsource type=pulse val0=0 val1=3.3 delay=0 rise=0.05n fall=0.05n width=20n period=40n
V2 (B 0) vsource type=pulse val0=0 val1=3.3 delay=0 rise=0.05n fall=0.05n width=10n period=20n
```

4.5 Simulation Graph/Output

- **Explanation:** Displays timing diagrams or waveforms confirming NOR function.
- **Key Points:**
    - Output HIGH only when all inputs are LOW.
    - Observes transitions and any delay measurements.

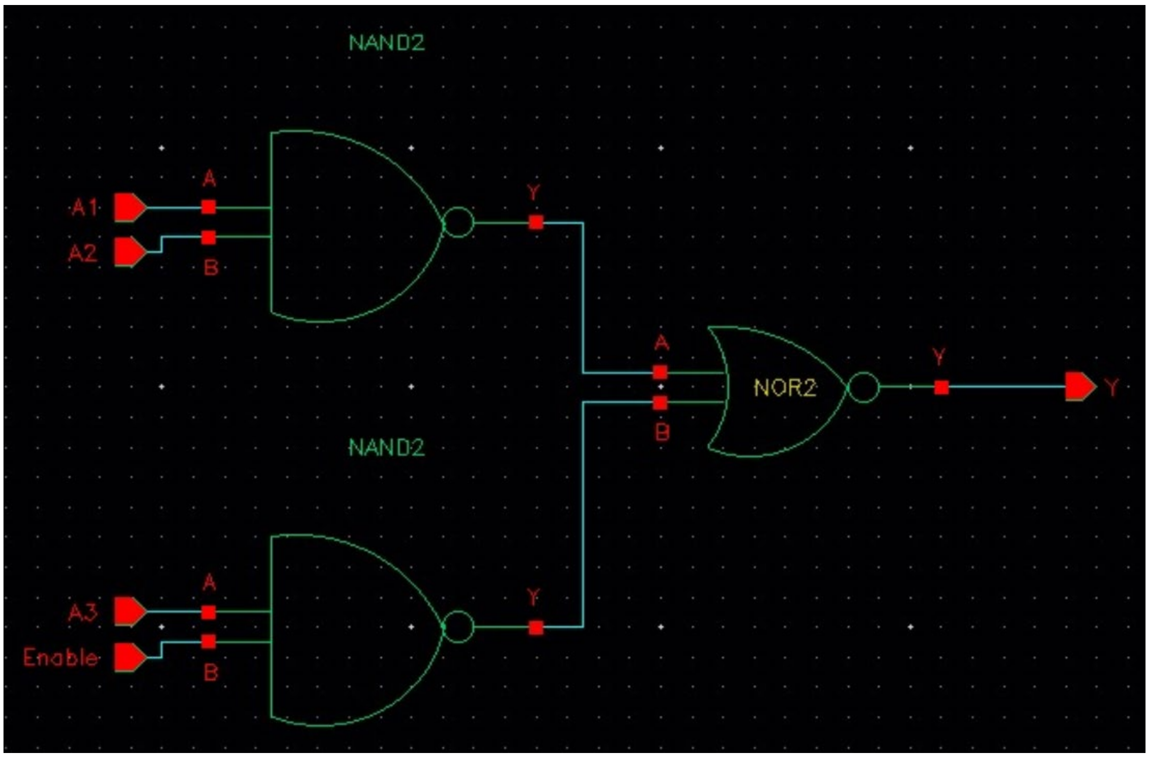


5. AND4 Gate

A **4-input AND Gate** outputs HIGH only when all inputs are HIGH. While not always a standard cell, it can be constructed from multiple NAND/INV gates or as a direct transistor design.

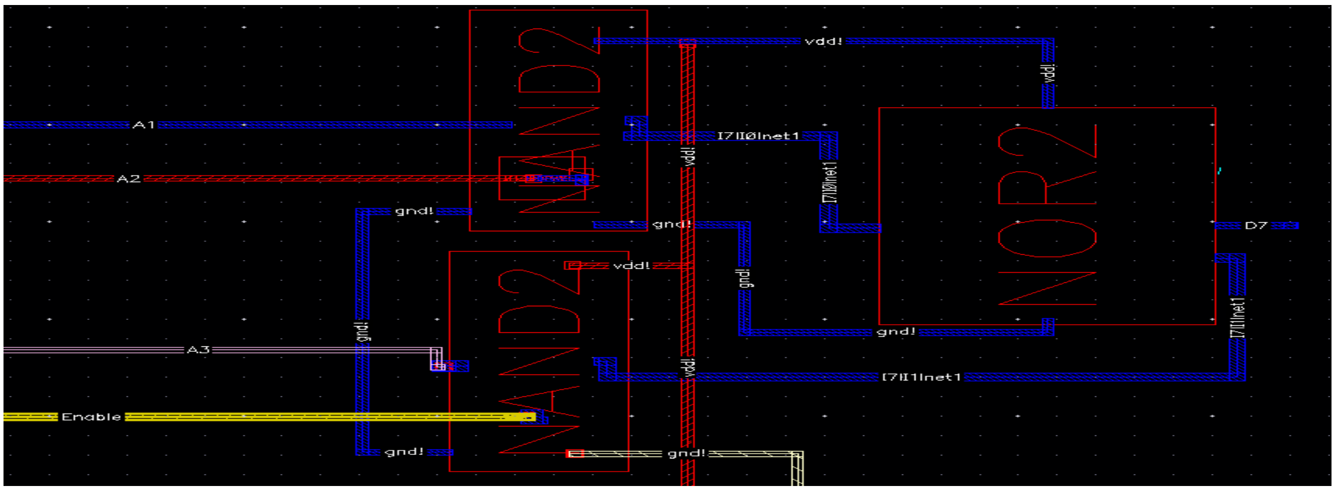
5.1 Schematic

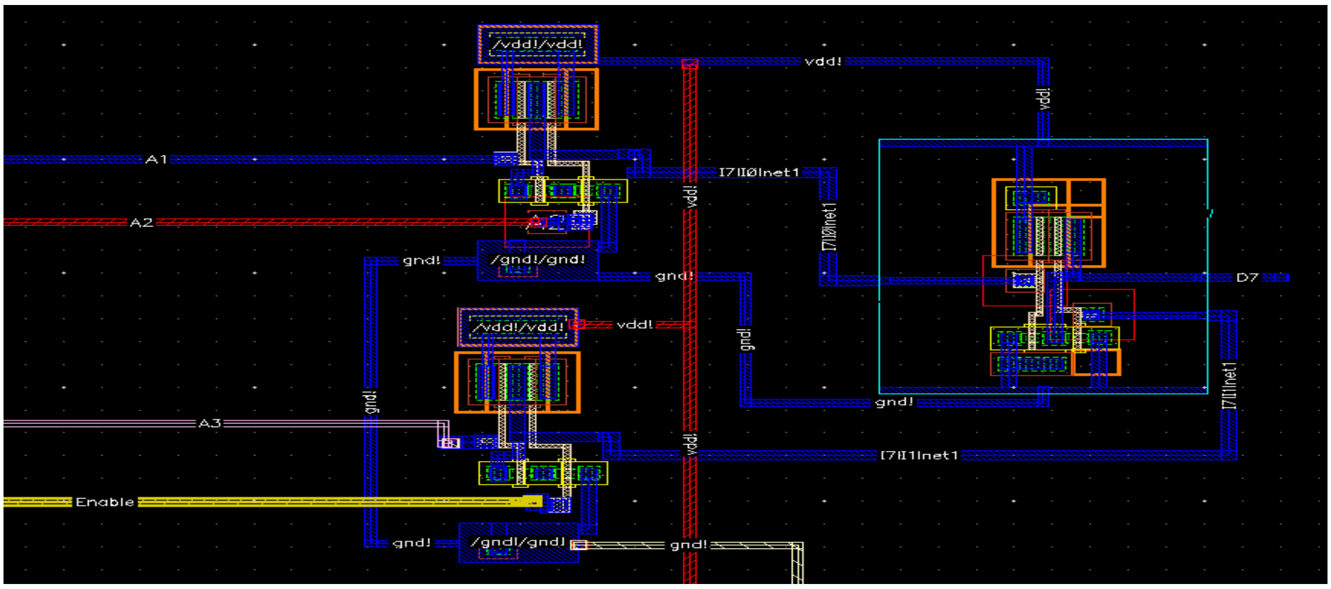
- **Explanation:** Demonstrates the logic-level representation or transistor-level arrangement (depending on approach).
- **Key Points:**
  - Potentially implemented using NAND gates and an inverter (De Morgan's Theorem).
  - Can also be drawn as a custom transistor network.



5.2 Layout

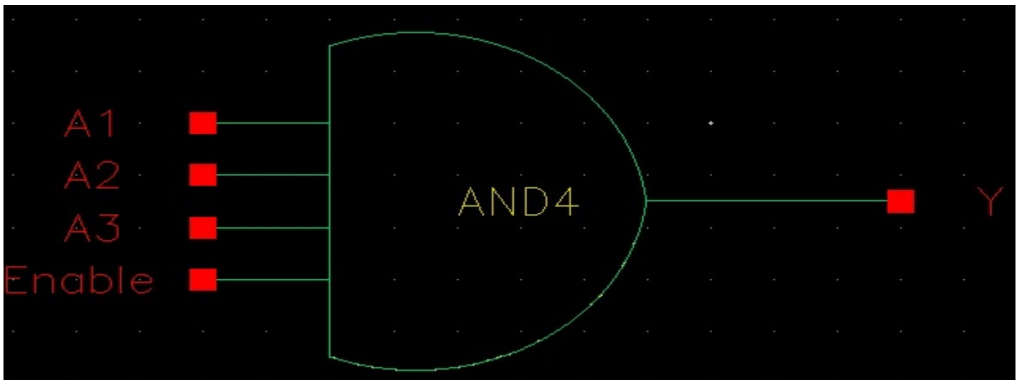
- **Explanation:** Shows the floorplan of a 4-input gate.
- **Key Points:**
  - More inputs means potentially larger area.
  - Pay attention to route signals to avoid overlapping or excessive parasitics.





5.3 Symbol

- **Explanation:** A simplified gate icon with four input pins and one output pin.
- **Key Points:**
  - Labeled (A, B, C, D) inputs and a single output (Y).
  - Helps visually keep track of multiple signals in larger circuits.



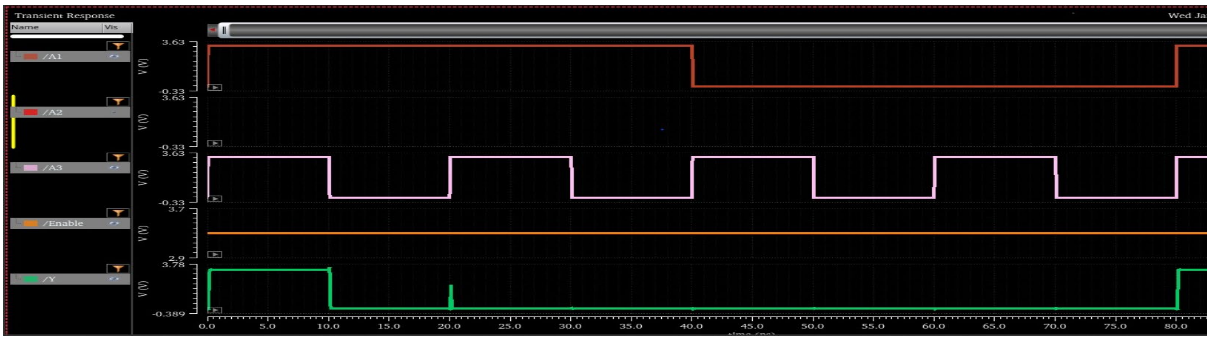
5.4 Stimulus File

- **Explanation:** Covers all 16 input combinations for thorough testing.
- **Key Points:**
    - Ensures no corner cases are missed.
    - May also vary supply or temperature if advanced simulation is desired.

```
simulator lang=spectre
vdd (vdd! 0) vsource dc=3.3
V1 (A1 0) vsource type=pulse val0=0 val1=3.3 delay=0 rise=0.05n fall=0.05n width=40n period=80n
V2 (A2 0) vsource type=pulse val0=0 val1=3.3 delay=0 rise=0.05n fall=0.05n width=20n period=40n
V3 (A3 0) vsource type=pulse val0=0 val1=3.3 delay=0 rise=0.05n fall=0.05n width=10n period=20n
Enable1 (Enable 0) vsource dc=3.3
```

5.5 Simulation Graph/Output

- **Explanation:** Presents the timing waveforms confirming AND operation.
- **Key Points:**
    - Verify the output is HIGH only when A, B, C, D are all HIGH.
    - Observe any delay or glitches if present.

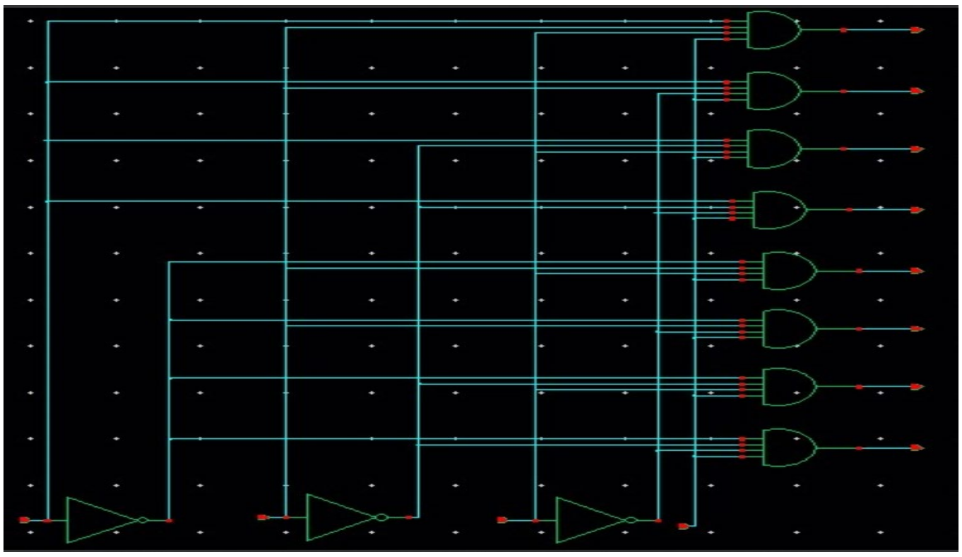


6. 3-to-8 Decoder

A **3-to-8 Decoder** takes 3 address inputs and activates exactly one of its 8 outputs. Decoders are crucial in selecting the correct row or column in memory arrays.

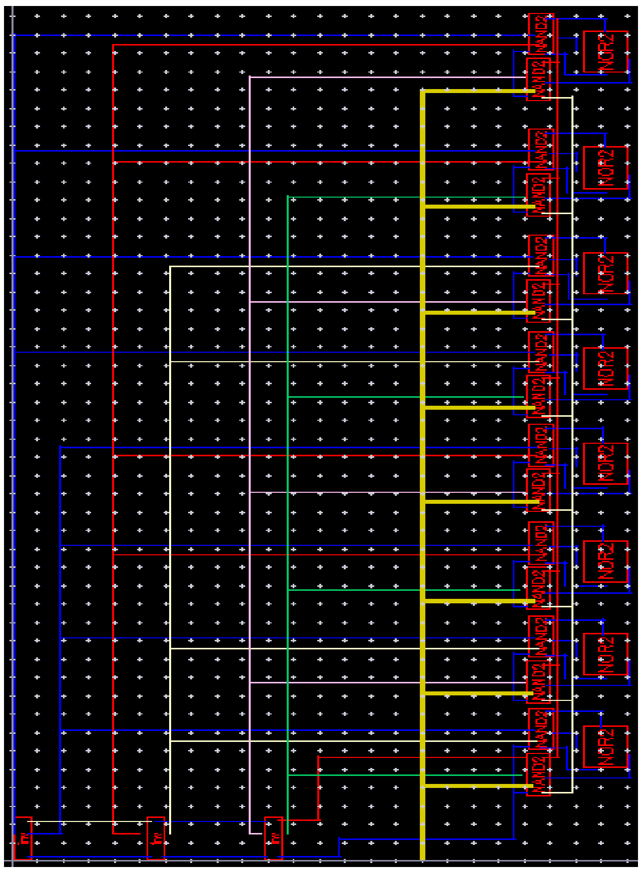
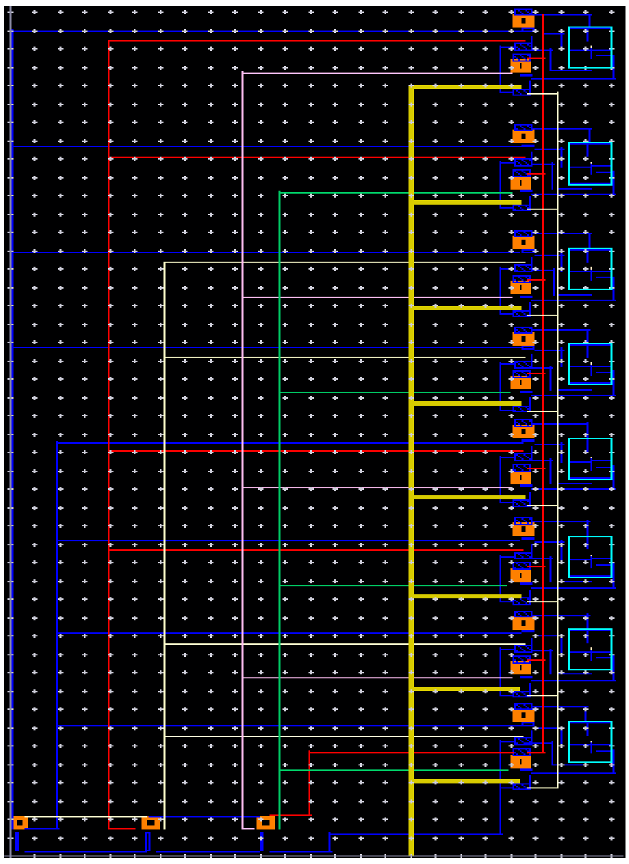
6.1 Schematic

- **Explanation:** Shows the combination of inverters, AND gates (or NAND/NOR variants) that produce 8 unique output lines.
- **Key Points:**
    - Each output corresponds to one unique 3-bit input pattern.
    - Typically used to enable one row of the SRAM at a time.



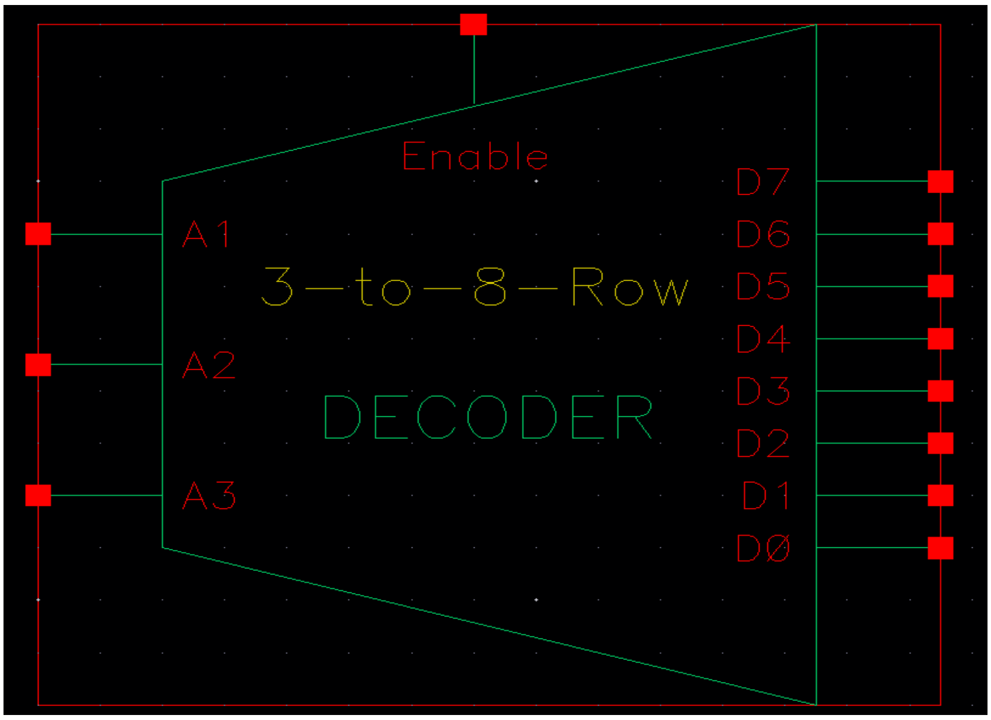
6.2 Layout

- **Explanation:** Depicts how the decoder's gates are physically arranged.
- **Key Points:**
    - The layout can be repetitive, so grouping similar logic blocks can help maintain consistency.
    - Minimize wire length to reduce delay, especially in high-frequency designs.

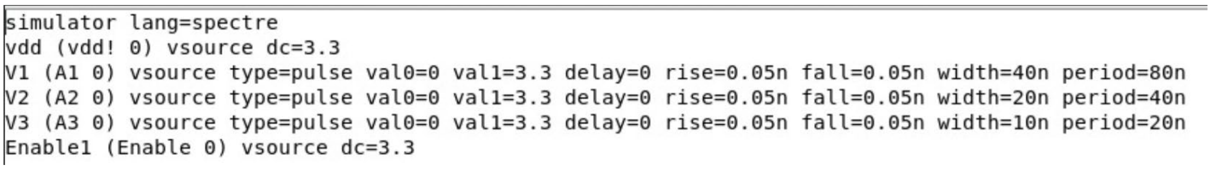
6.3 Symbol

- **Explanation:** Shows a box labeled “3-to-8 Decoder” with 3 inputs and 8 outputs.
- **Key Points:**
    - Properly label address lines (A1, A2, A3) and output lines (D0–D7).
    - This top-level symbol is used directly in the final SRAM schematic.



6.4 Stimulus File

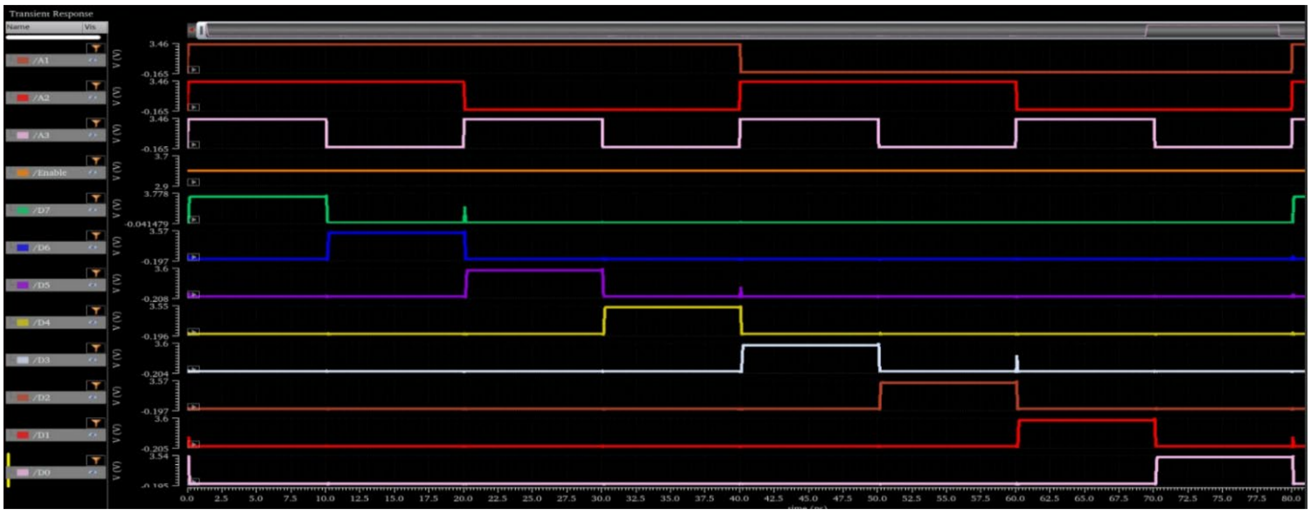
- **Explanation:** Cycles through all 3-bit input combinations (000 to 111).
- **Key Points:**
    - Ensures the correct single-line activation for each address pattern.



```
simulator lang=spectre
vdd (vdd! 0) vsource dc=3.3
V1 (A1 0) vsource type=pulse val0=0 val1=3.3 delay=0 rise=0.05n fall=0.05n width=40n period=80n
V2 (A2 0) vsource type=pulse val0=0 val1=3.3 delay=0 rise=0.05n fall=0.05n width=20n period=40n
V3 (A3 0) vsource type=pulse val0=0 val1=3.3 delay=0 rise=0.05n fall=0.05n width=10n period=20n
Enable1 (Enable 0) vsource dc=3.3
```

6.5 Simulation Graph/Output

- **Explanation:** Demonstrates each of the 8 outputs goes HIGH in turn.
- **Key Points:**
    - Validates the correct one-hot output behavior.
    - Confirms no two outputs are active simultaneously.

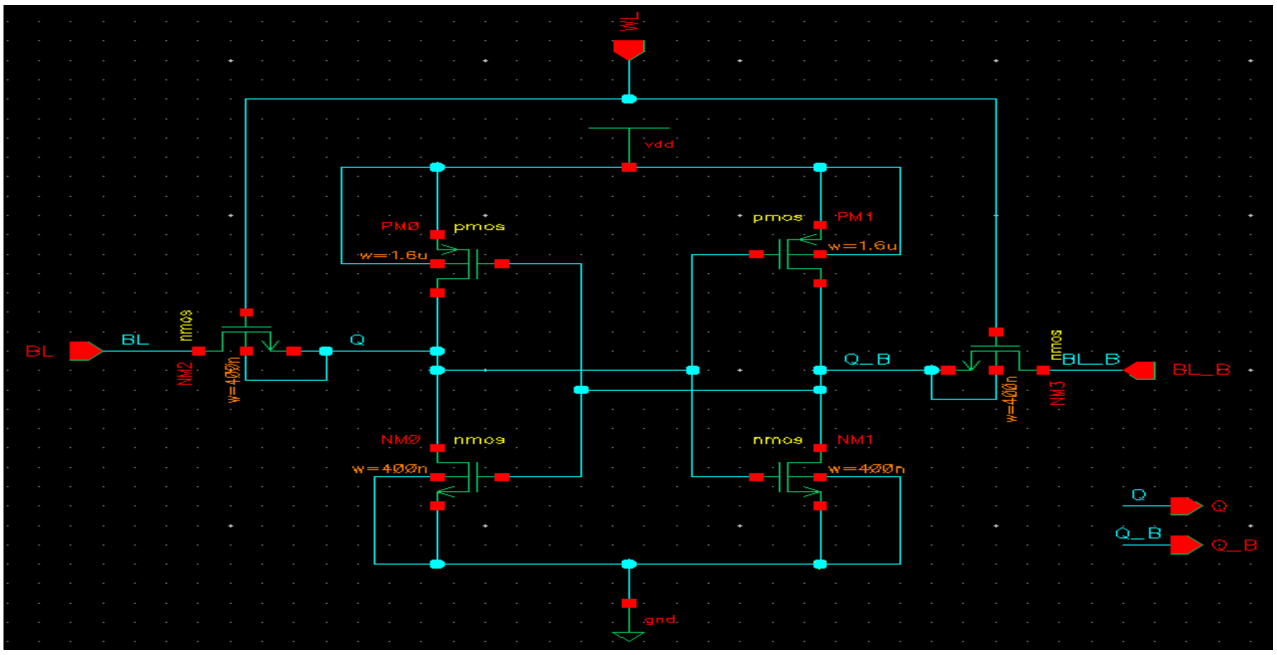


7. 6T SRAM Cell

A standard **6T SRAM Cell** consists of two cross-coupled inverters for data storage and two access transistors for read/write operations.

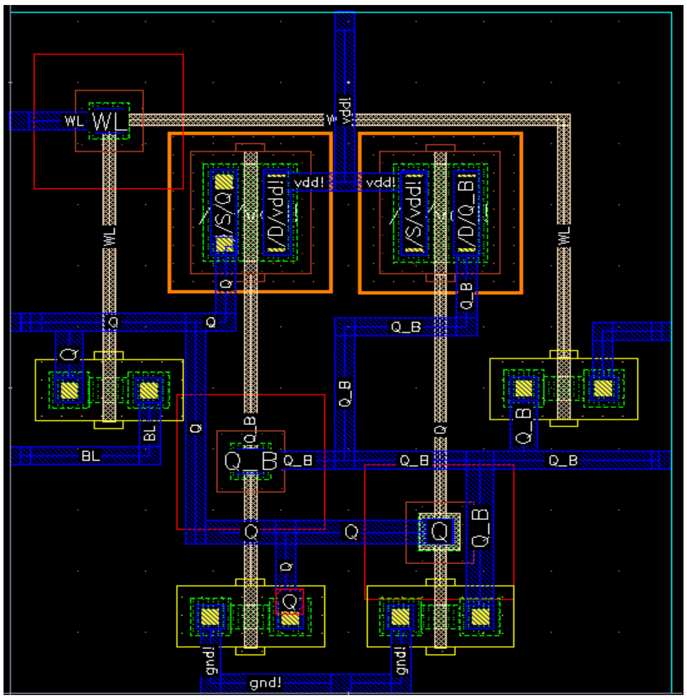
7.1 Schematic

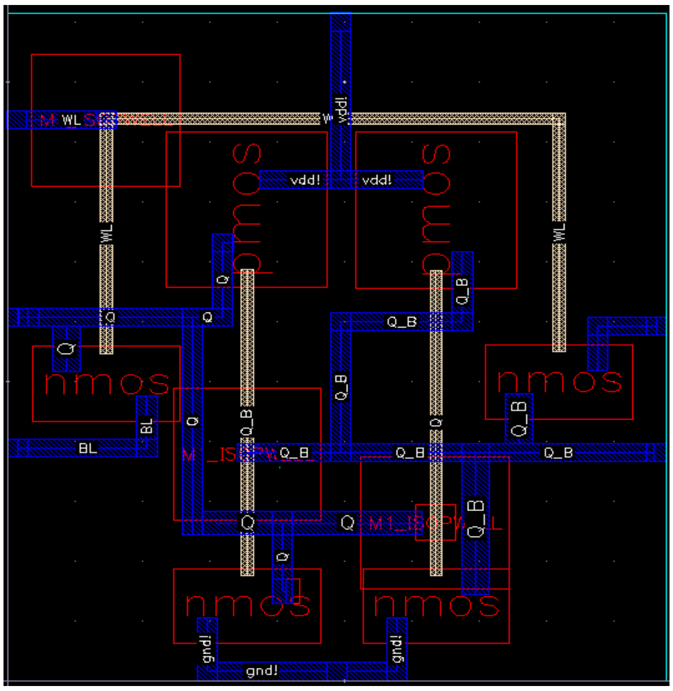
- **Explanation:** Shows the six transistors—four in the inverter pairs, two for bitline access.
- **Key Points:**
  - When Word Line (WL) is enabled, data can be written or read from the cell via bit lines (BL, BLB).
  - When WL is disabled, the inverters maintain the stored value.



7.2 Layout

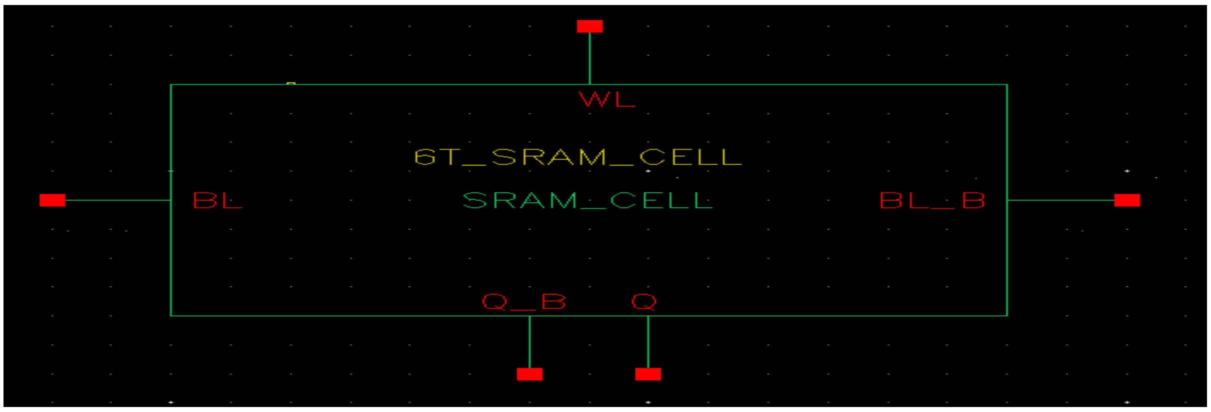
- **Explanation:** Illustrates the optimized physical arrangement of these six transistors.
- **Key Points:**
    - Typically arranged to minimize area.
    - Matching inverter transistor pairs is crucial to reduce static noise margin issues.





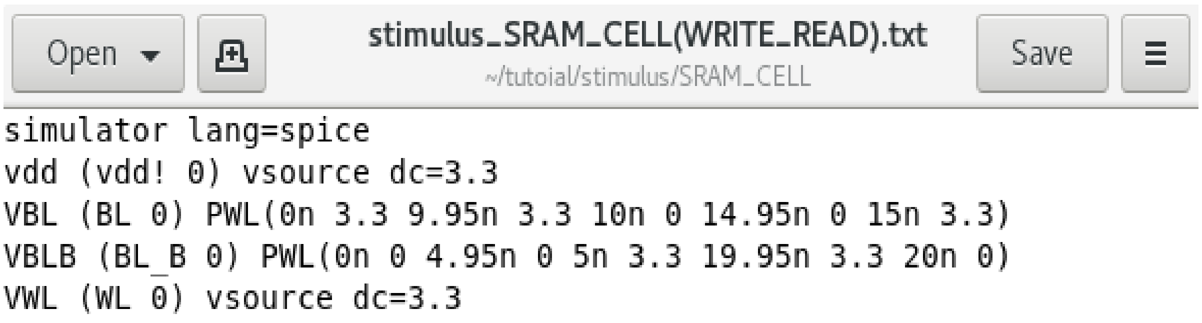
7.3 Symbol

- **Explanation:** Simplified representation with pins for WL, BL, BLB, VDD, and GND.
- **Key Points:**
    - Used as a repeating unit in each row/column of the memory array.



7.4 Stimulus File

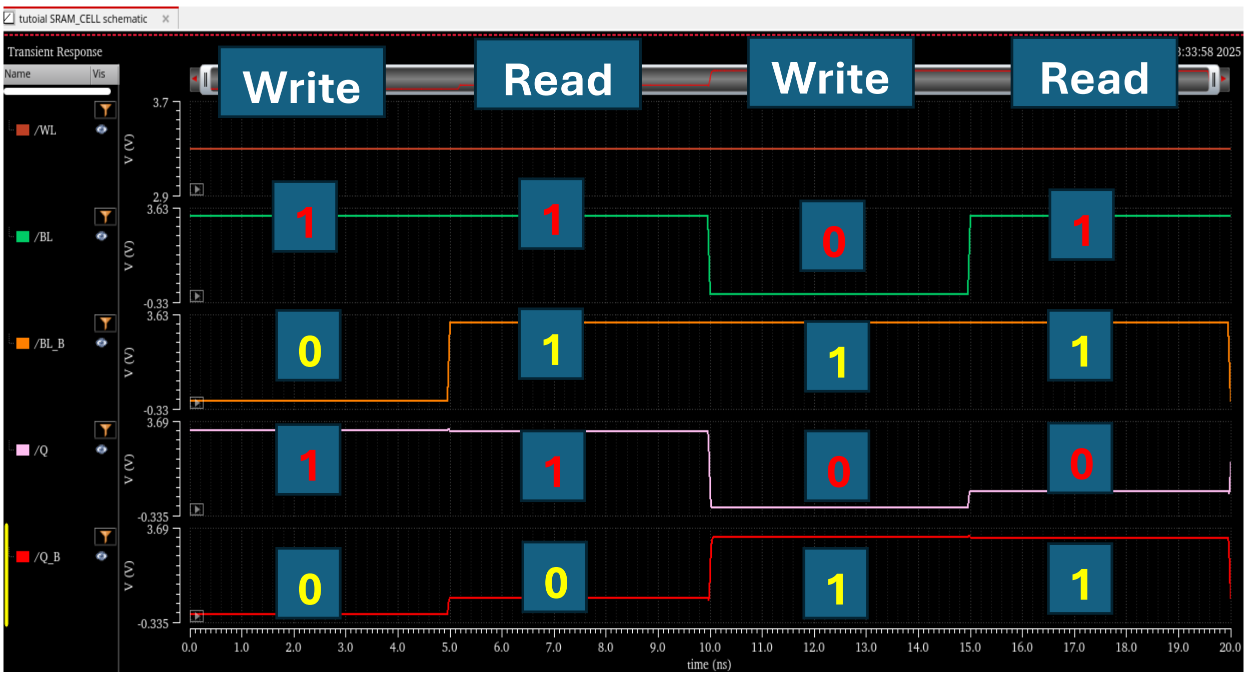
- **Explanation:** Outlines the read/write test conditions:
    - Write “1,” then read it.
    - Write “0,” then read it.
- **Key Points:**
    - Checks voltage levels on the bit lines and ensures the cell retains data when WL is inactive.


Open
Save
stimulus_SRAM_CELL(WRITE_READ).txt
~/tutoial/stimulus/SRAM_CELL

```
simulator lang=spice
vdd (vdd! 0) vsource dc=3.3
VBL (BL 0) PWL(0n 3.3 9.95n 3.3 10n 0 14.95n 0 15n 3.3)
VBLB (BL_B 0) PWL(0n 0 4.95n 0 5n 3.3 19.95n 3.3 20n 0)
VWL (WL 0) vsource dc=3.3
```

7.5 Simulation Graph/Output

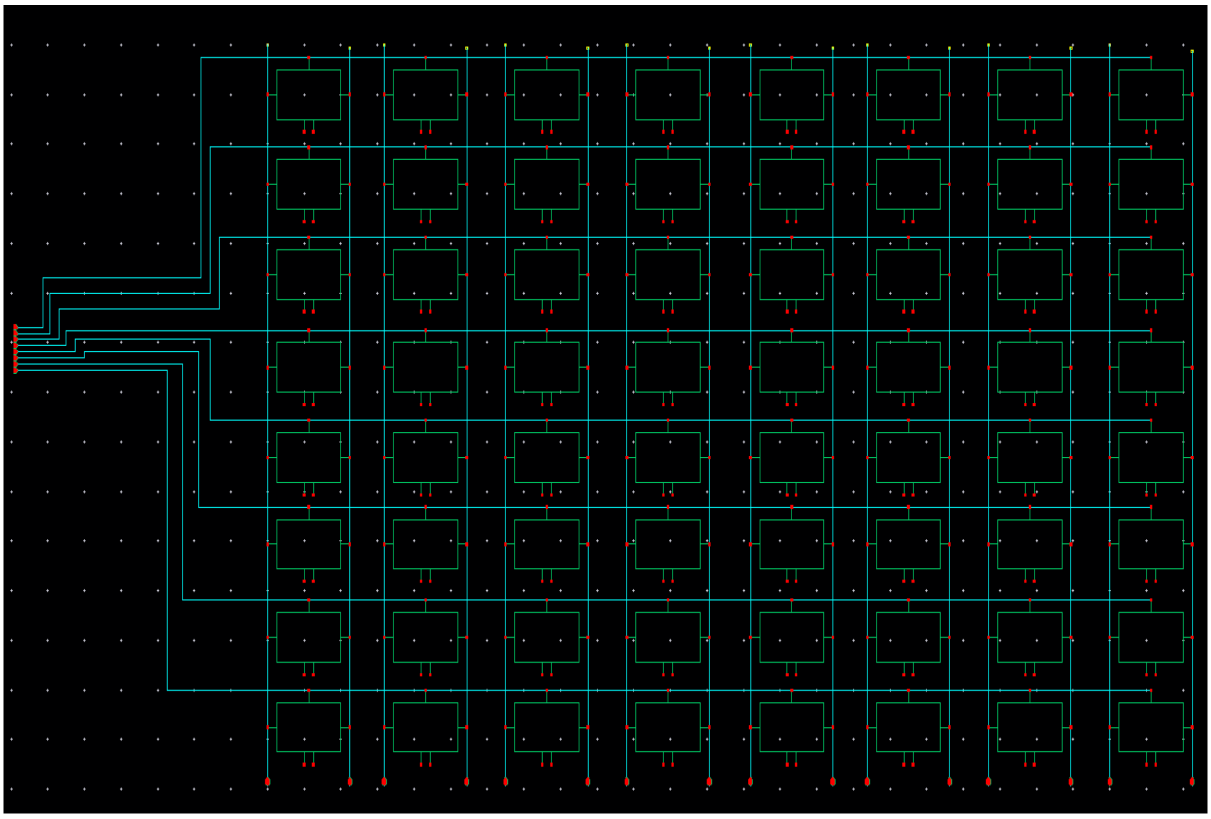
- **Explanation:** Displays the cell's response during write and read cycles.
- **Key Points:**
    - Verifies that data is stored correctly (feedback loop holds data).
    - Confirms the cell is robust against noise or slight voltage variations.



8. 8×8 SRAM Array

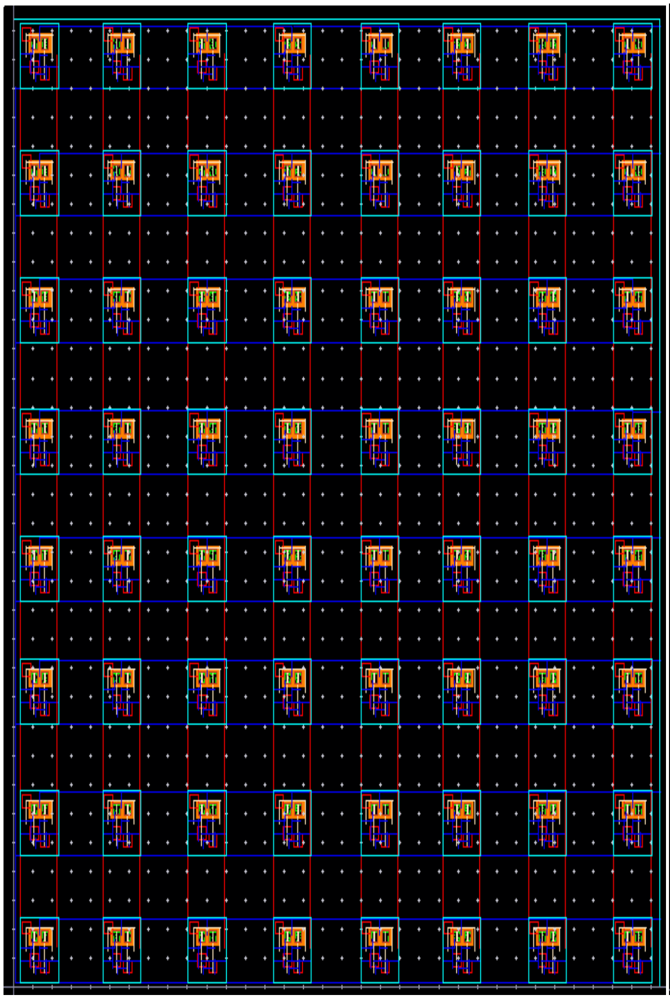
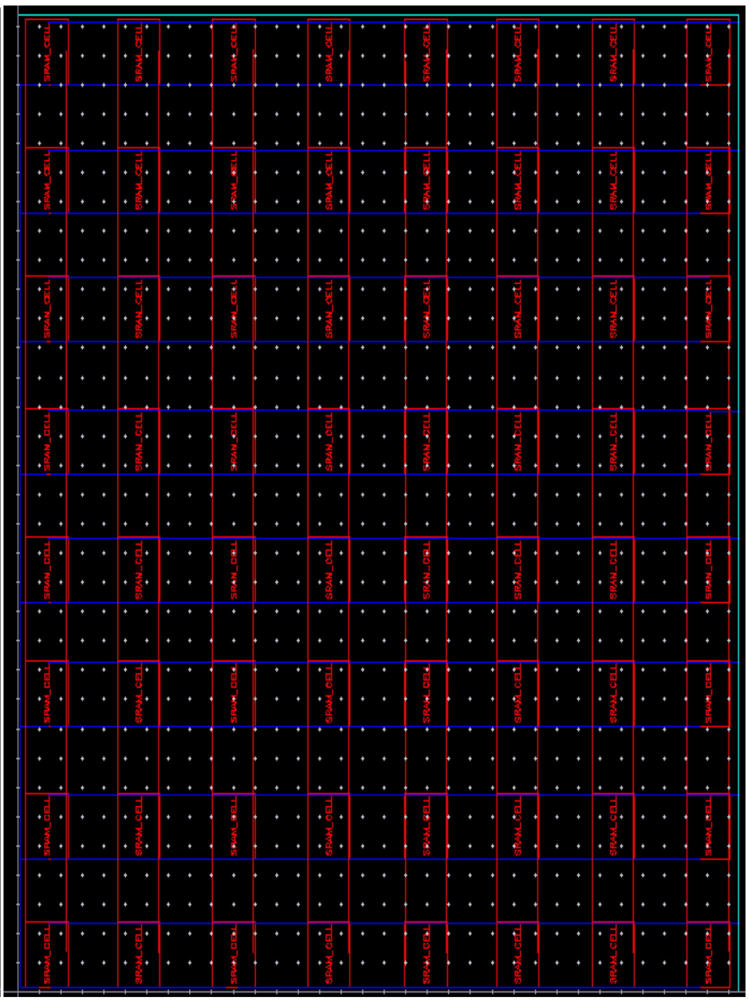
8.1 Schematic

- **Explanation:**
  Shows how sixty-four (8×8) 6T SRAM cells are arranged in rows and columns, with shared word lines (WL) and bit lines (BL, BLB). Each cell maintains the same 6T structure (two cross-coupled inverters + two access transistors).
- **Key Points:**
  - **Row Activation via WL:**
    - **Each of the 8 rows has a dedicated WL that, when enabled, allows read or write access to all cells in that row.**
  - **Common Bit Lines (BL/BLB):**
    - **Each column shares a pair of bit lines (BL, BLB). A column can be read from or written to by enabling its WL and using the bit lines for data transfer.**
  - **Scalability:**
    - **This 8×8 arrangement can be expanded to larger arrays by adding more rows/columns of 6T cells.**



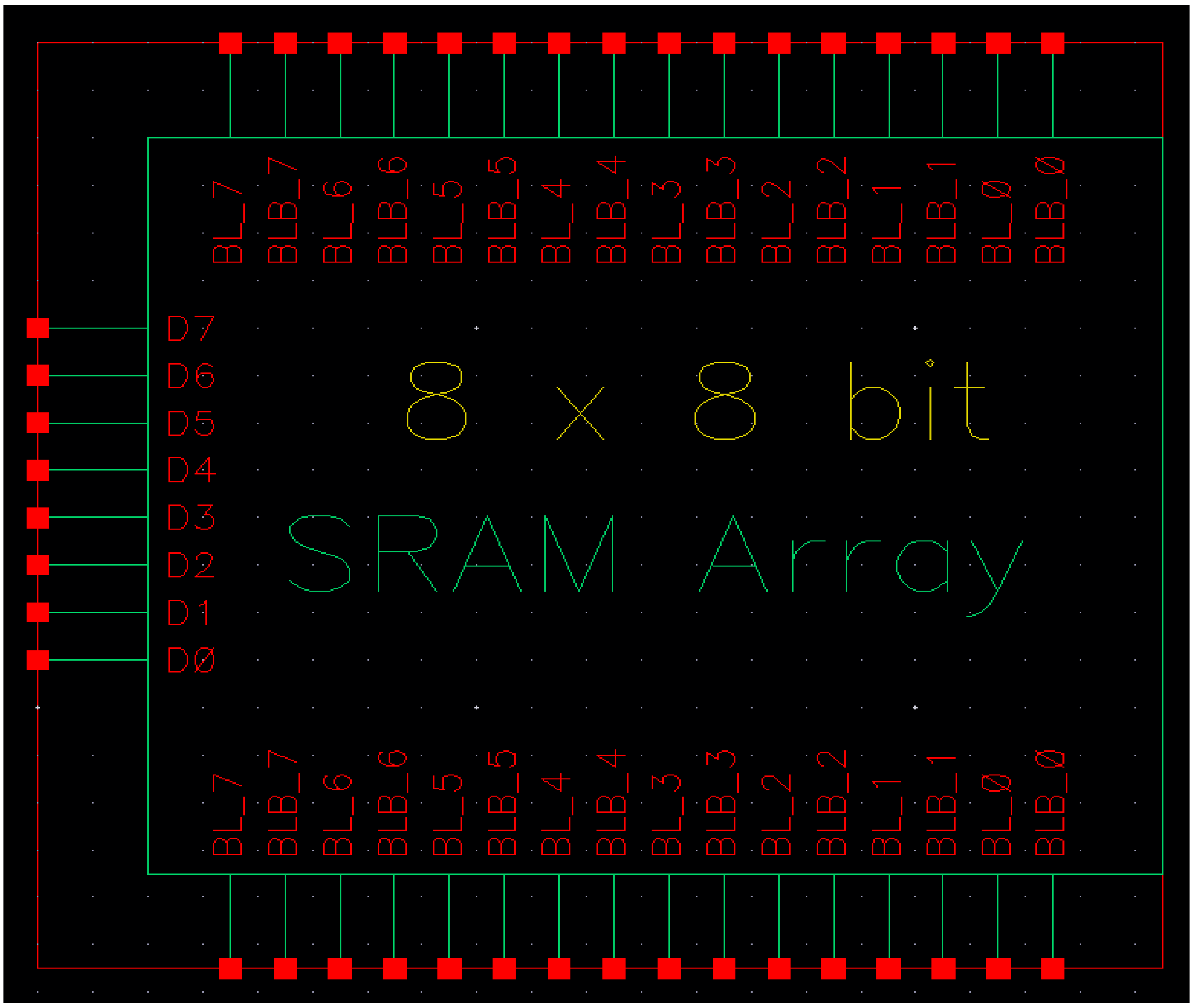
8.2 Layout

- **Explanation:**
  Illustrates the physical placement of the 6T cells in a grid, where rows and columns are laid out for minimal area and optimal routing of power, ground, word lines, and bit lines.
- **Key Points:**
  - **Regular Cell Tiling:**
    - **Repeats the 6T unit cell across 8 rows and 8 columns, ensuring uniformity and easy routing.**
  - **Shared Interconnects:**
    - **Word lines run horizontally across each row, and bit lines run vertically through each column. This reduces wiring complexity.**
  - **Peripheral Placement:**
    - **Any necessary row decoders or sense amplifiers (if used) are typically placed at array edges, maintaining a standard cell pitch in the main array area.**

8.3 Symbol

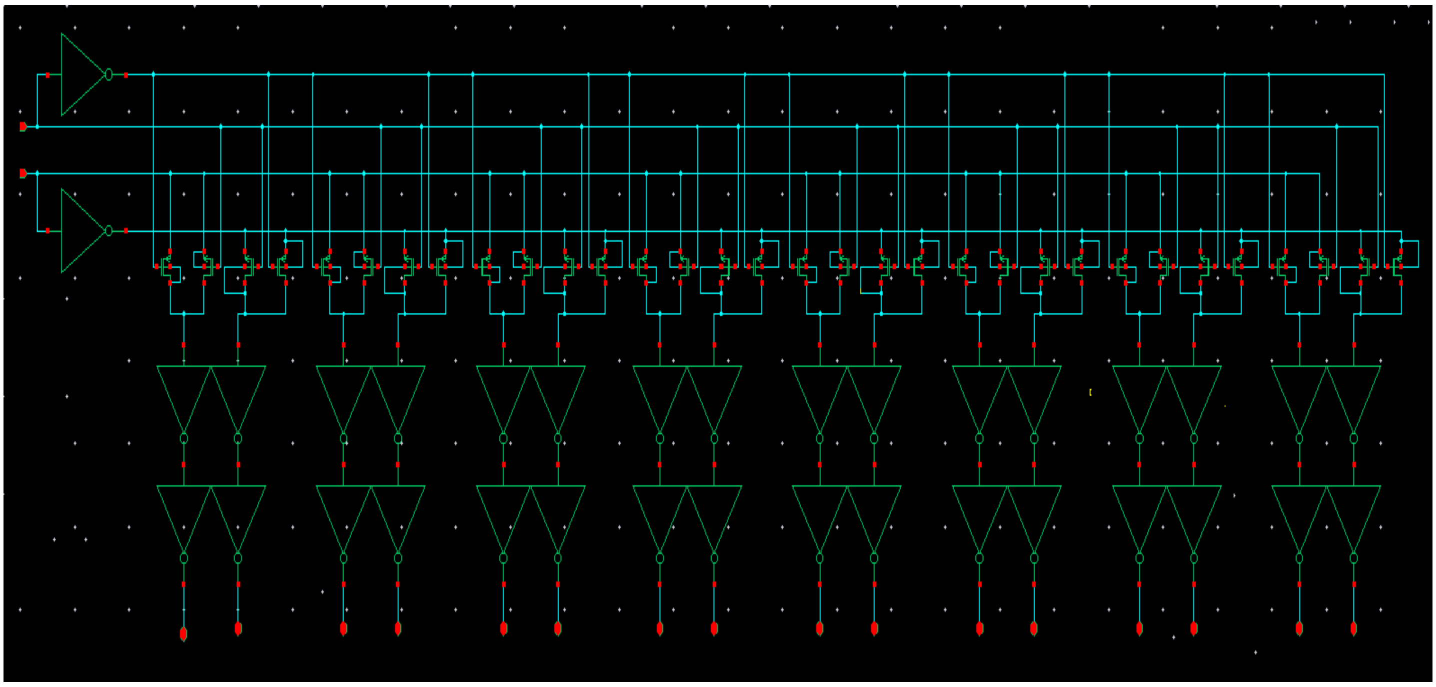
- **Explanation:**
  A high-level representation showing the 8×8 block of SRAM cells with pins for the word lines, bit lines, VDD, and GND.
- **Key Points:**
  - **Address/Control Pins (if used):**
    - **Could be represented in a simplified form (e.g., WL1-WL8 for row select), while BL/BLB pairs represent columns.**
  - **Power Pins (VDD, GND):**
    - **Common supply rails used by all 64 cells.**
  - **Consistent with Single-Cell Symbol:**
    - **Maintains the same WL, BL, BLB concept as the 6T cell, but repeated across the matrix.**



9. SRAM Read/Write

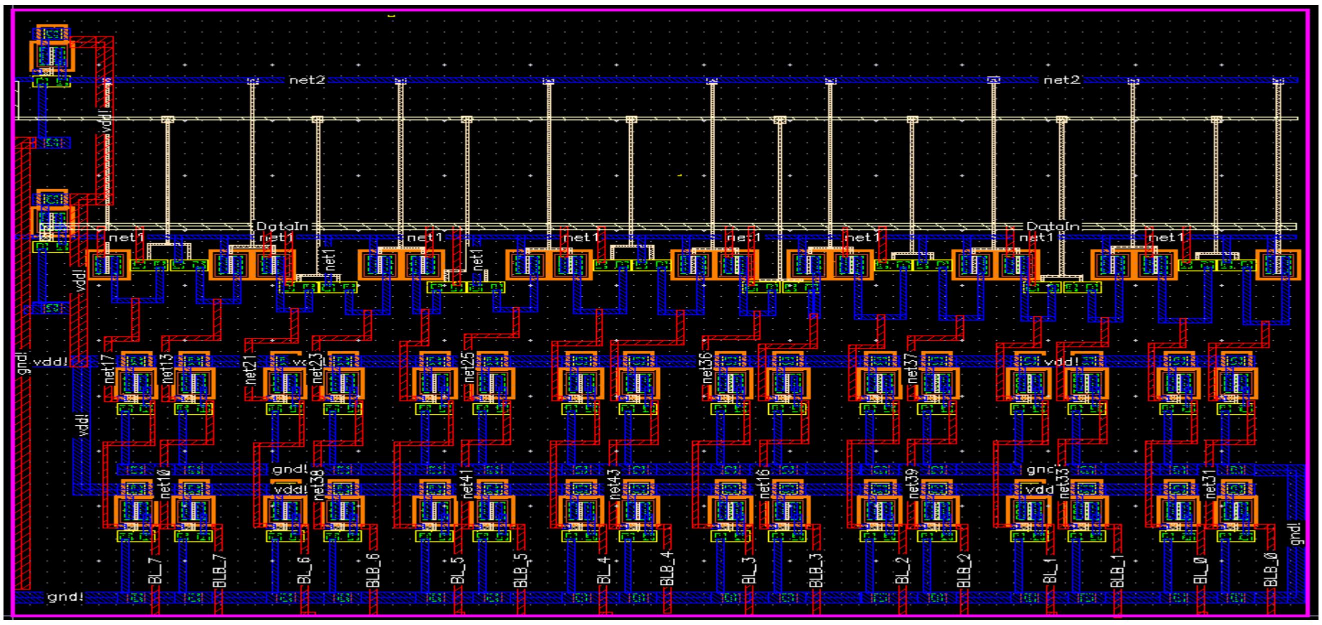
9.1 Schematic

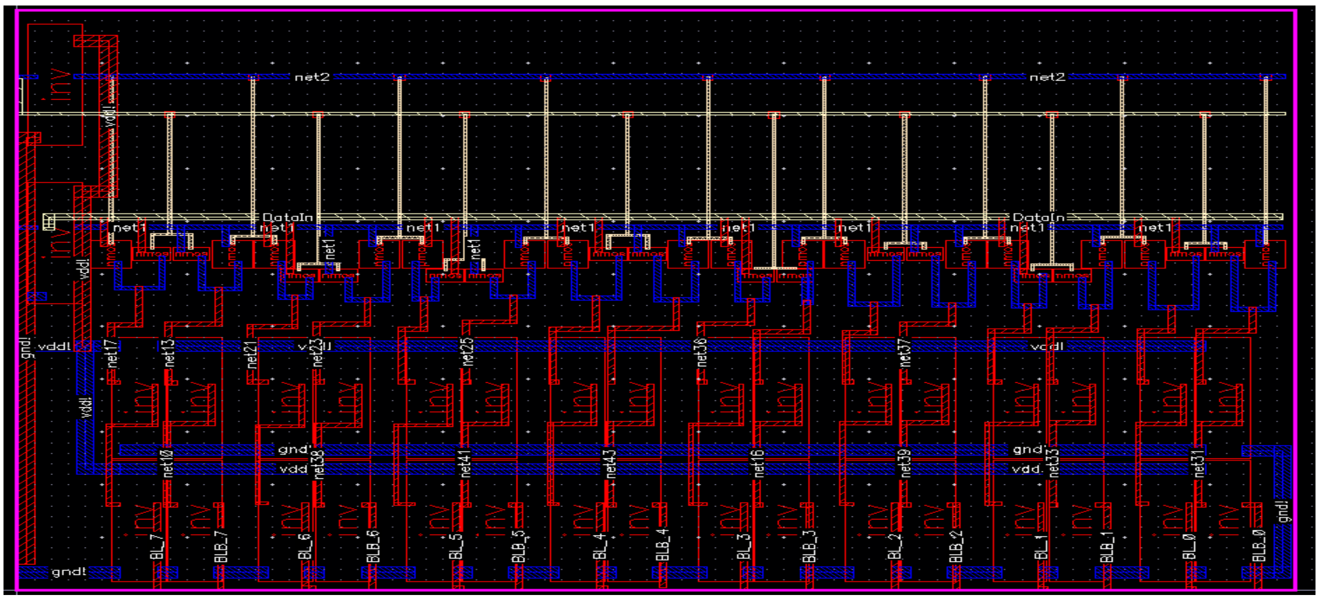
- **Explanation:**
  This schematic focuses on the 6T SRAM cell (or cells) connected with one control signal WR (Write/Read) and a supply node DataIn that indicates the data to be written.
- **Key Points:**
  - **WR Signal:**
    - **WR = 1 → Write mode.**
    - **WR = 0 → Read mode.**
  - **DataIn as Data Input:**
    - **VDD = 1 → Write a '1'.**
    - **VDD = 0 → Write a '0'.**
  - **Outputs (Data Lines):**
    - **When WR = 0, the stored data is driven out as the read value.**
  - **Isolation:**
    - **The cell retains its data when not selected or when WR is low (and not actively reading).**



9.2 Layout

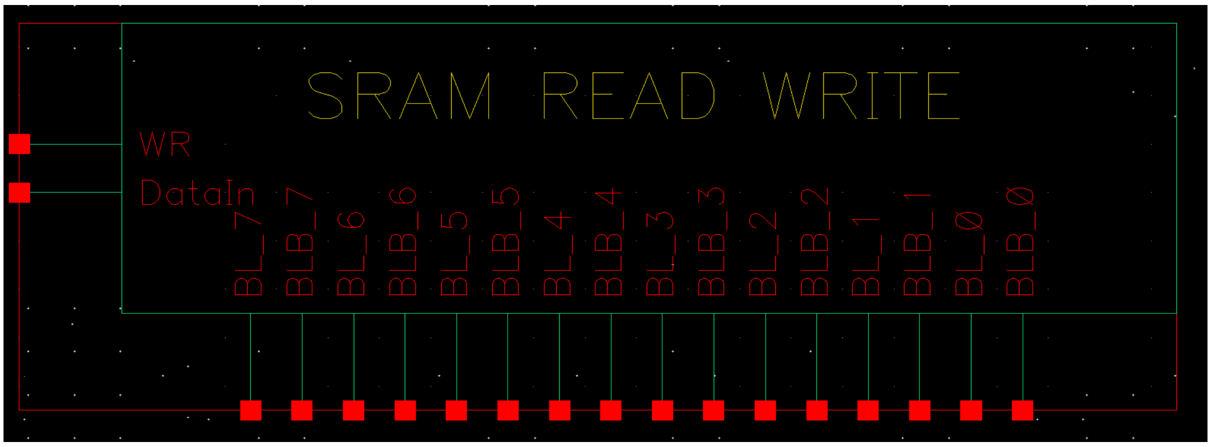
- **Explanation:**
  Depicts the physical arrangement of the 6T cell, showing the two cross-coupled inverters and two access transistors connected to bit lines (BL/BLB).
- **Key Points:**
  1. **Compact Footprint: The 6T cell is laid out to minimize area while maintaining good matching.**
  2. **Power Routing: DataIn and GND must be routed carefully to each cell for stable operation, especially during write cycles.**
  3. **Isolation Structures: Proper well isolation and transistor sizing are crucial to prevent noise coupling.**





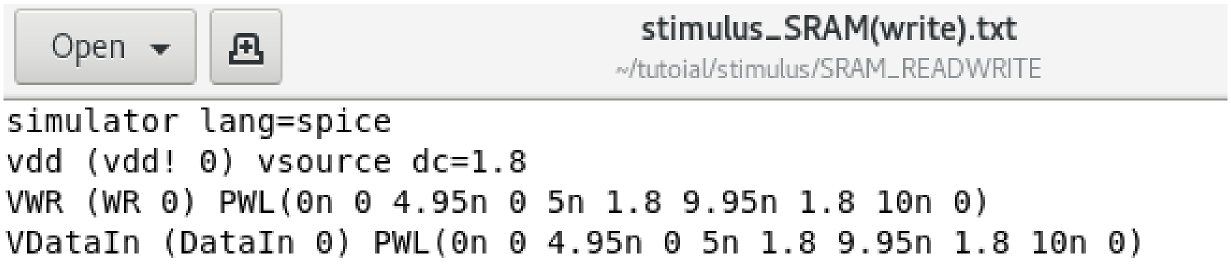
9.3 Symbol

- **Explanation:**
  A simplified representation of the read/write configuration, showing:
  - **WR (switches between read and write).**
  - **DataIn (the data to be written—1 or 0).**
  - **Outputs (represent the stored data during read).**
- **Key Points:**
  - **Minimal Pin Set: Only the essential pins are exposed (WR, DataIn, Outputs).**
  - **Reusable Block: This symbol can be replicated across an array.**



9.4 Stimulus File

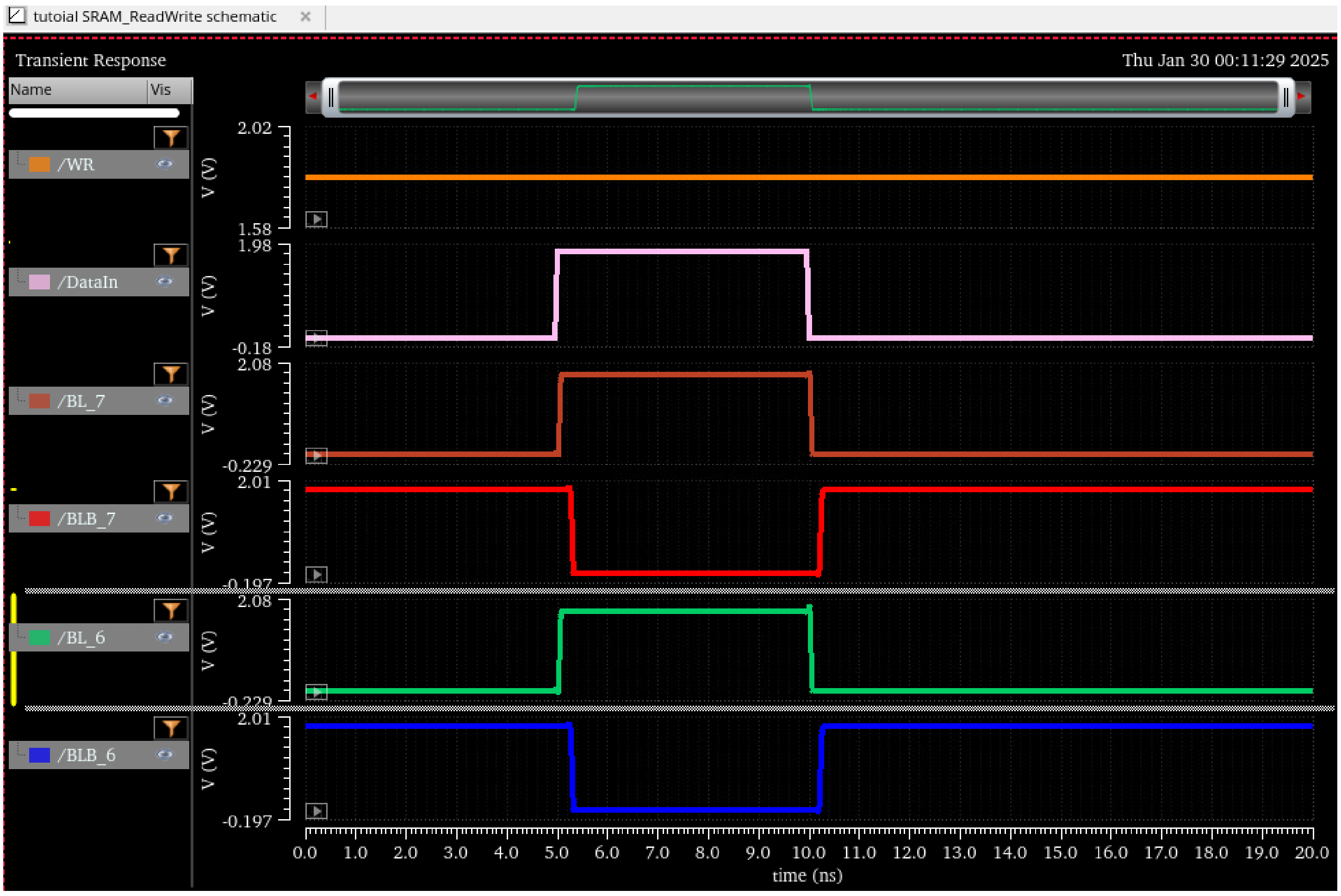
- **Explanation:**
  The testbench sequence illustrating how to write and read data using the WR and DataIn signals.
- **Key Points:**
  1. **Write '1':**
     - **Set WR = 1 (write mode), DataIn = 1 (the value to store).**
     - **Allow enough time for the cell to latch this '1'.**
  2. **Write '0':**
     - **Set WR = 1, DataIn = 0.**
     - **Wait for the cell to store the '0'.**



```
simulator lang=spice
vdd (vdd! 0) vsource dc=1.8
VWR (WR 0) PWL(0n 0 4.95n 0 5n 1.8 9.95n 1.8 10n 0)
VDataIn (DataIn 0) PWL(0n 0 4.95n 0 5n 1.8 9.95n 1.8 10n 0)
```

9.5 Simulation Graph/Output

- **Explanation:**
  Displays the timing waveforms of WR, DataIn, and the output signals during successive write and read operations.
- **Key Points:**
  1. **Write Operation:**
     - **Observe how the internal node of the SRAM cell follows DataIn when WR = 1.**
  2. **Read Operation:**
     - **When WR = 0, the cell's stored value is driven onto the output nodes.**
  3. **Data Retention:**
     - **Confirm the cell maintains its value between operations (no unintended flips).**



9.6 Observations

- **Control Simplicity:**
  A single signal (WR) makes it straightforward to toggle between read and write.
- **Data Integrity:**
  Proper transistor sizing ensures the cell is not accidentally overwritten or corrupted during writes.
- **Power Usage:**
  Toggling DataIn during write cycles can have power implications; efficient design may be needed for large-scale arrays.

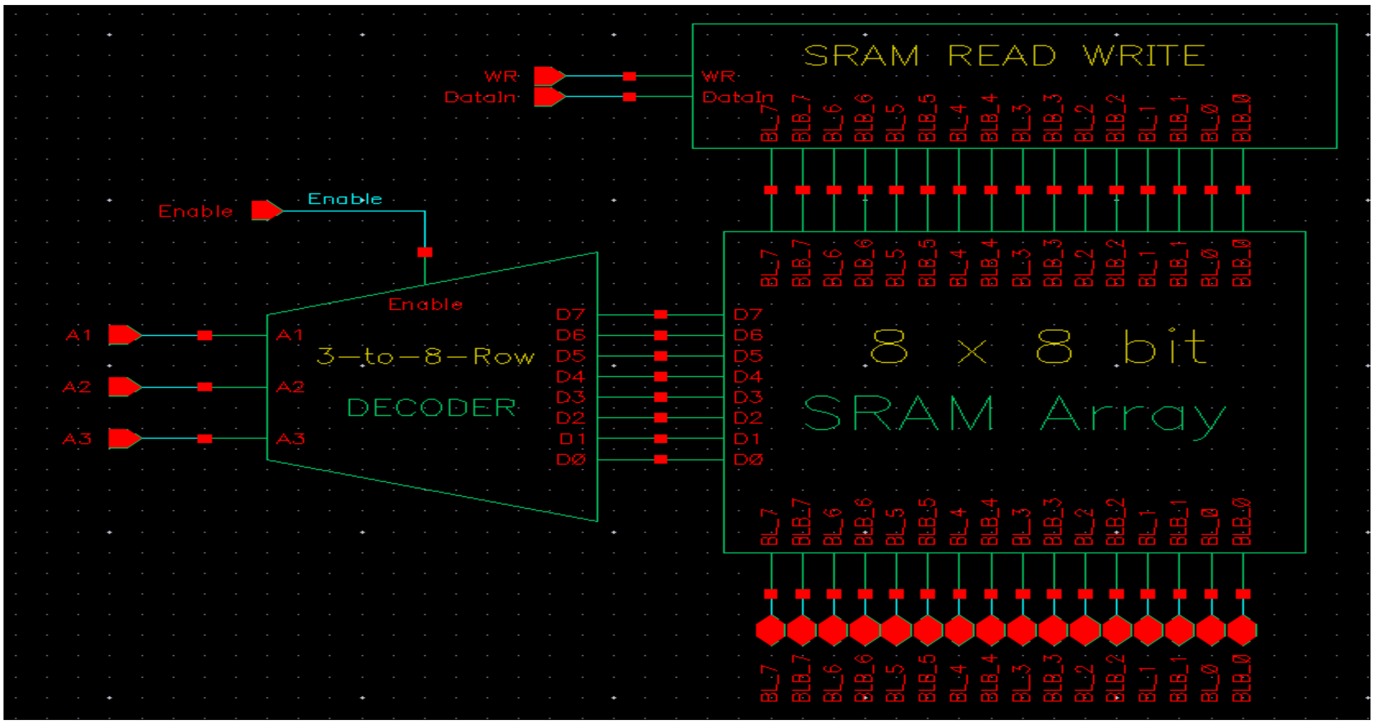
---

10. Complete 8×8 SRAM System

Bringing everything together, the **8×8 SRAM** organizes 64 (6T) memory cells, controlled by row decoders, and appropriate input/output logic.

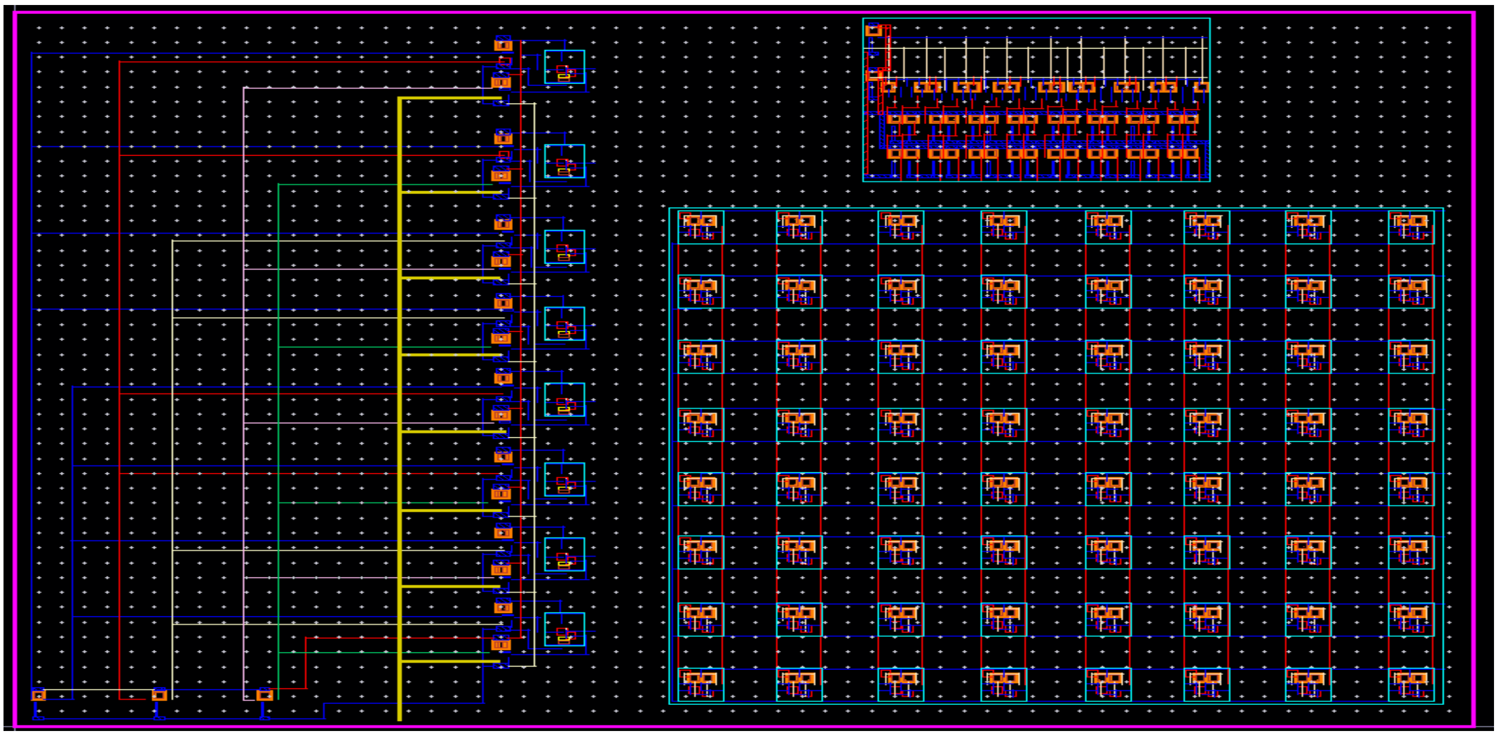
10.1 Schematic

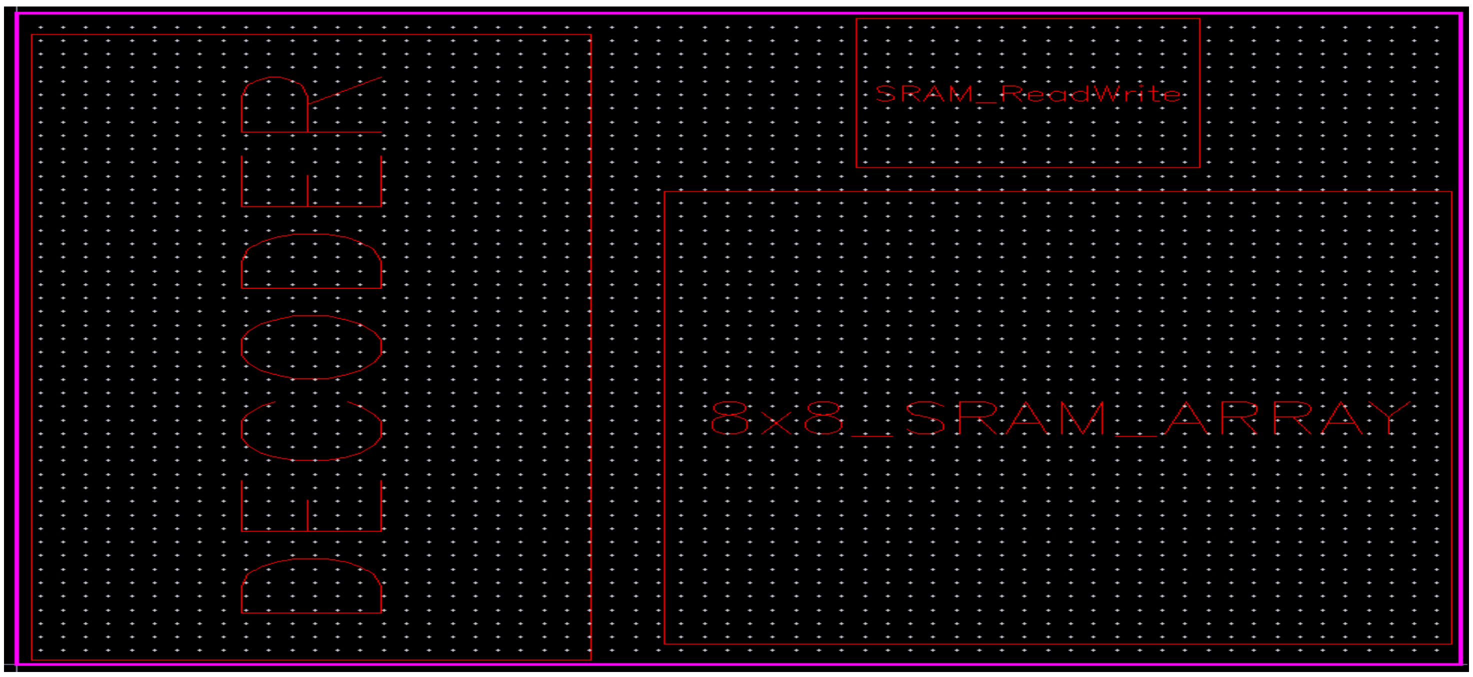
- **Explanation:** Shows the top-level connection of:
  - The 3-to-8 row decoder to activate one of the 8-word lines.
  - Eight columns of memory cells each connecting to bit lines.
  - Any additional control circuitry for read/write enable.
- **Key Points:**
  - Ensures each row is uniquely accessed.
  - Data lines aggregate from all columns.

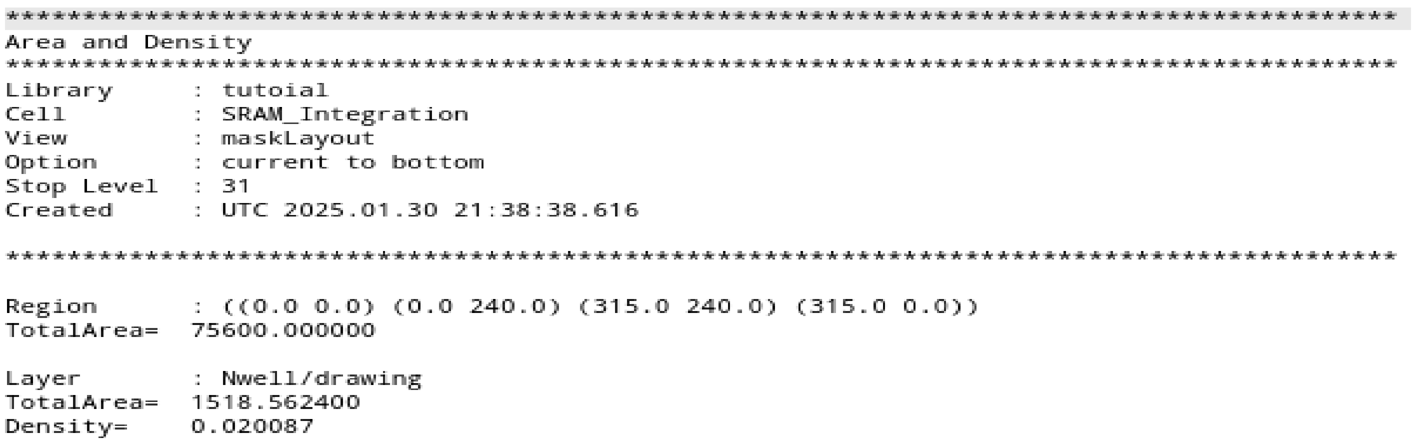


10.2 Layout

- **Explanation:** Depicts the physical arrangement of the 64 cells, row decoders, and supporting logic.
- **Key Points:**
  - Typically laid out in a matrix structure with consistent repeating cell rows/columns.
  - Global power and ground rails must be carefully planned for stable operation.





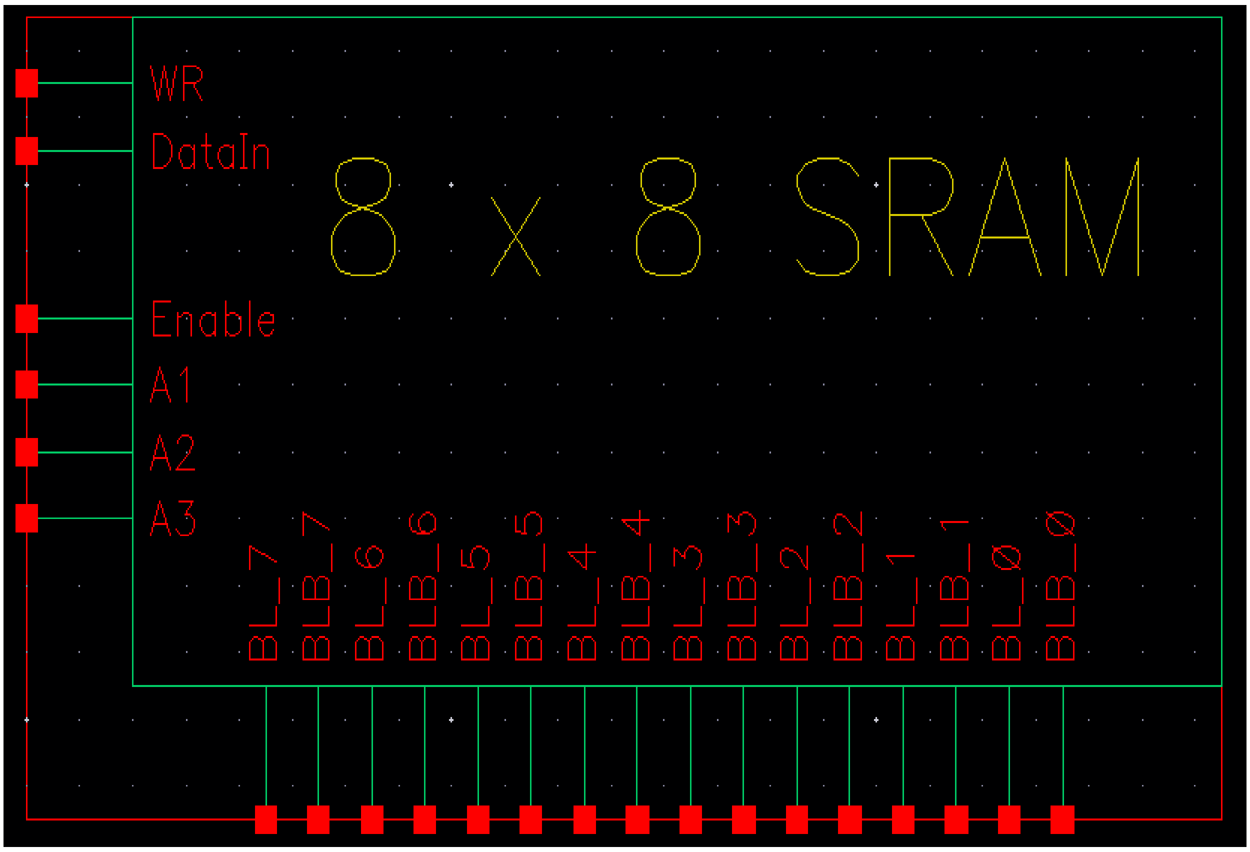


```
*****************************************************************************************
Area and Density
*****************************************************************************************
Library       : tutoial
Cell          : SRAM_Integration
View          : maskLayout
Option        : current to bottom
Stop Level    : 31
Created       : UTC 2025.01.30 21:38:38.616

*****************************************************************************************

Region        : ((0.0 0.0) (0.0 240.0) (315.0 240.0) (315.0 0.0))
TotalArea=    75600.000000

Layer         : Nwell/drawing
TotalArea=    1518.562400
Density=      0.020087
```

Complete 8×8 SRAM System Area equals to 240(height)*315(length)= 75600(area)

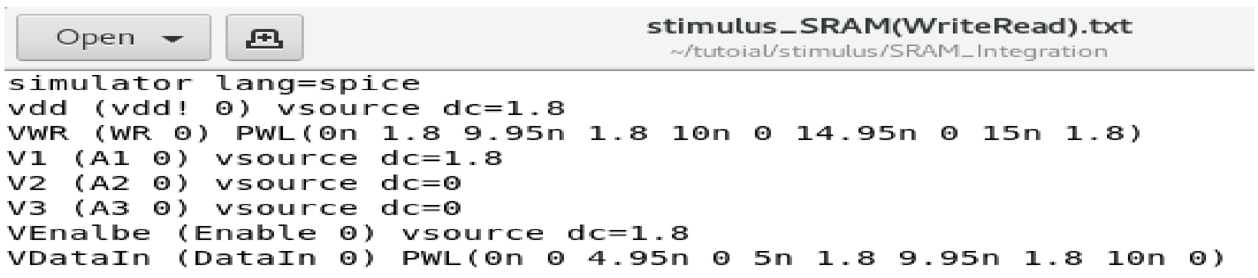
10.3 Symbol

- **Explanation:** A high-level block diagram representing the entire SRAM.
- **Key Points:**
    - Address lines (A0, A1, A2) for row selection, plus any column selection if applicable.
    - Data bus lines (BL0–BL7,BLB0-BLB7) for 8-bit input/output.
    - Control signals (WR for write read enable, DataIn for transferring data to SRAM.) as needed.



10.4 Stimulus File

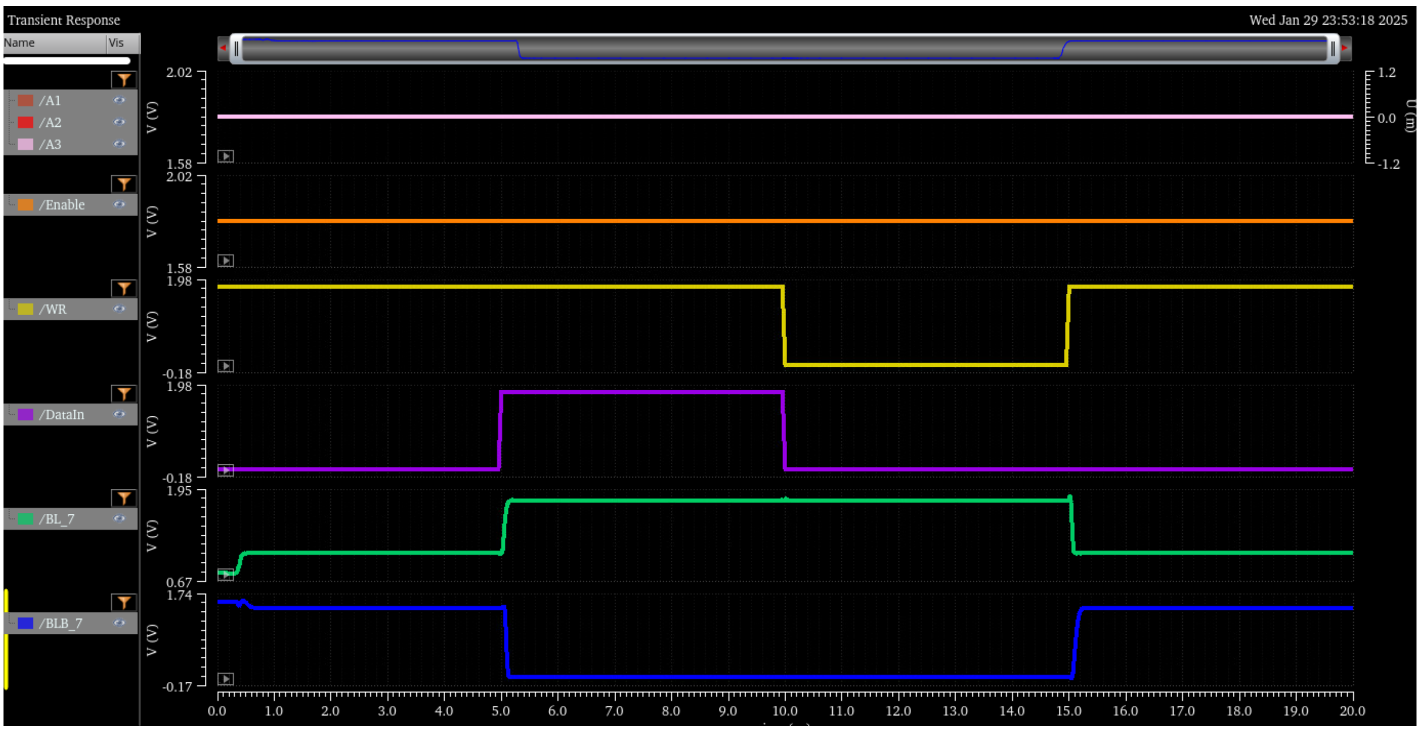
- **Explanation:** A comprehensive testbench that performs multiple reads and writes across various addresses.
- **Key Points:**
  - Demonstrates the correctness of storing data in one address and retrieving it from another.
  - Checks for potential conflicts or data corruption during simultaneous operations.


Open
stimulus_SRAM(WriteRead).txt
~/tutoial/stimulus/SRAM_Integration

```
simulator lang=spice
vdd (vdd! 0) vsource dc=1.8
VWR (WR 0) PWL(0n 1.8 9.95n 1.8 10n 0 14.95n 0 15n 1.8)
V1 (A1 0) vsource dc=1.8
V2 (A2 0) vsource dc=0
V3 (A3 0) vsource dc=0
VEnalbe (Enable 0) vsource dc=1.8
VDataIn (DataIn 0) PWL(0n 0 4.95n 0 5n 1.8 9.95n 1.8 10n 0)
```

10.5 Graph Output

- **Explanation:** Waveforms show the timing for address setup, write operations, and read outputs.
- **Key Points:**
  - Confirms that when a certain address is selected and WE is active, data is written into the correct memory location.
  - During a read operation, data is driven onto the output lines, validating correct access.



11. Conclusion

Summarize the success of the project, emphasizing:

- **Key Lessons:**
  - Importance of thorough verification at each design level (schematic, layout, simulation).
  - Best practices in transistor sizing, layout organization, and power distribution.
- **Project Achievements:**
  - Functional 8×8 SRAM with verified read/write cycles.
  - Mastery of Cadence Virtuoso's schematic entry, layout tools, DRC, LVS, and simulation environment.
- **Future Work:**
  - Potential integration of sense amplifiers or column decoders for larger memory blocks.
  - Exploration of low-power design techniques or advanced CMOS processes for better performance.