

16×4 DRAM Project Report

Adithya Selvakumar & Krishna Karthikeya Chemudupati

Contents

1	Memory Description	4
1.1	Text Description of Memory Operation	4
1.1.1	3T1C Storage Cell	4
1.1.2	Write Mechanism	4
1.1.3	Read Mechanism	4
1.1.4	Bitline Conditioning	5
1.1.5	Row Selection and Wordline Control	5
1.1.6	Sense Amplification	5
1.1.7	Refresh Operation	5
1.2	Description of Timing Requirements	6
1.2.1	Two-Phase Clocking Strategy	6
1.2.2	Read Timing Sequence	6
1.2.3	Write Timing Sequence	6
1.2.4	Minimum Clock Period	6
1.2.5	Precharge and Evaluation Constraints	6
1.2.6	Refresh Timing Requirement	7
1.3	Transistor Sizing Rationale	8
1.3.1	Memory Cell Transistor Sizing	8
1.3.2	Write Path Considerations	8
1.3.3	Leakage and Retention Tradeoff	8
1.3.4	Precharge and Bitline Driver Sizing	8
1.3.5	Sense Amplifier Sizing and Noise Margins	9
1.3.6	Peripheral Logic and Decoder Sizing	9
1.3.7	Energy Considerations	9
1.3.8	Summary of Experimentally Guided Sizing Decisions	9
1.4	Modern DRAM Optimizations	10
2	Schematics with Size Annotations	13
3	Design Functionality	34
3.1	Memory Design Validation Description	34
3.2	Functional Verification of Row Decoder	35
3.3	Functional Verification of Tristate Buffer	37
3.4	Functional Verification of Clock Generator	39
3.5	Functional Verification of Single Bitcell	41
3.6	Functional Verification of DRAM	43
3.6.1	Case 1: 0000 \rightarrow 0011	44
3.6.2	Case 2: 0100 \rightarrow 0011	45
3.6.3	Case 3: 1100 \rightarrow 1111	46
3.6.4	Case 4: 0100 \rightarrow 0011	47
4	Design Metrics with Support	48
4.1	Delay	48
4.1.1	Write Delay	48
4.1.2	Read Delay	49
4.2	Power	51

4.3	Area	53
4.4	Figure of Merit	54

1 Memory Description

1.1 Text Description of Memory Operation

The proposed memory implements a 16×4 dynamic random access memory (DRAM) using a 3T1C storage architecture, yielding a total capacity of 64 bits. Unlike static memories that rely on cross-coupled inverters to represent data, this design stores information as electric charge on a physical capacitor at each bit location. As a result, the stored data naturally degrades over time due to leakage currents through device junctions and dielectric paths, which fundamentally differentiates DRAM from SRAM and introduces the need for both controlled timing and periodic refresh. To improve robustness and simplify internal sequencing, the memory employs independent read and write access paths through separate wordlines, allowing read and write operations to be electrically isolated in time while sharing the same storage node.

1.1.1 3T1C Storage Cell

Each storage element is implemented using three NMOS transistors and a single capacitor. The write access transistor connects the storage capacitor to the write bitline (WBL) and is controlled by the write wordline (WWL). A second access transistor connects the storage node to the read bitline (RBL) and is controlled by the read wordline (RWL). A third isolation device electrically separates the storage node from surrounding circuitry during inactive phases, reducing leakage and preventing unintended discharge. The storage capacitance is selected as 1 pF to balance three competing constraints: sufficient charge storage for reliable sensing, acceptable retention time under worst-case leakage, and minimal area overhead in layout. The resulting cell stores information purely in the form of voltage across this capacitor.

1.1.2 Write Mechanism

During a write operation, the external input data is first driven onto the write bitline through column-level write driver circuits. With write enable asserted, the write wordline (WWL) corresponding to the selected row is activated, turning on the write access transistor and directly connecting the storage capacitor to the WBL. If a logic 1 is written, the WBL is driven to V_{DD} and the capacitor charges toward the supply. If a logic 0 is written, the WBL is driven to ground and the capacitor discharges accordingly. Once WWL is deasserted, the write access transistor turns off and the storage capacitor becomes electrically isolated, holding its charge dynamically. The worst-case time required to fully settle the capacitor voltage during a write is measured as 68.377 nanoseconds, which directly constrains the minimum safe write pulse width and overall clock period.

1.1.3 Read Mechanism

Prior to every read operation, the read bitline (RBL) is forced to a known reference level using a PMOS-based precharge circuit. After precharge is released, the read wordline (RWL) corresponding to the selected row is asserted, connecting the storage capacitor to the RBL through the read access transistor. The stored charge then perturbs the RBL voltage through charge sharing. If the cell stores a logic 1, the capacitor maintains the RBL near V_{DD} . If the cell stores a logic 0, the capacitor partially discharges the RBL toward ground. This small analog voltage deviation is converted into a full rail-to-rail digital value using an inverter-based sense amplifier at the column output. The measured read access time from RWL activation to valid digital output is 0.21 nanoseconds.

1.1.4 Bitline Conditioning

All read bitlines are conditioned using dedicated PMOS precharge devices. During the precharge phase, each RBL is driven to a known reference level of V_{DD} prior to evaluation. This initialization ensures that subsequent voltage deviations during the read phase arise only from the stored capacitor charge and not from residual charge left over from previous memory operations. Proper completion of the precharge phase is essential for reliable sensing, as incomplete precharge results in an incorrect reference baseline and can lead to read errors and reduced noise margin.

1.1.5 Row Selection and Wordline Control

A 4-to-16 row decoder converts the 4-bit binary address into a one-hot row selection signal such that exactly one row is activated for any valid address. This row select signal is further combined with the clock and write enable controls to generate independent read and write wordlines. By separating the read and write accesses both logically and temporally, the design prevents electrical contention at the storage node, eliminates conflicts on the bitlines, and ensures deterministic memory behavior across all operating modes.

1.1.6 Sense Amplification

A single inverter per column is used as the sense amplifier. The inverter is designed with a switching threshold near mid-supply to maximize noise margin between valid logic states. Successful sensing requires that the RBL voltage after charge sharing crosses the inverter input thresholds. These thresholds are measured as 0.662V for logic high detection and 0.573V for logic low detection. This simple inverter-based approach provides compact layout and fast response while placing design emphasis on sufficient storage capacitance and clean bitline conditioning.

1.1.7 Refresh Operation

Since the stored charge on each capacitor decays continuously due to leakage currents, the memory requires periodic refresh to maintain data integrity. A dedicated refresh controller autonomously cycles through all 16 row addresses using a 4-bit counter. During each refresh operation, the selected row undergoes an internal read followed immediately by a restore operation that recharges or discharges the storage capacitors to their nominal levels. All rows are refreshed within a total refresh interval of 32ms (from literature review) to guarantee correct data retention under worst-case leakage and process conditions.

1.2 Description of Timing Requirements

1.2.1 Two-Phase Clocking Strategy

All memory operations are synchronized using two complementary non-overlapping clock phases, CLK and $CLKB$. These phases partition memory activity into distinct windows for precharge, evaluation, sensing, and writing. The enforced non-overlap between the clock phases ensures that read and write circuitry are never active simultaneously, preventing short-circuit contention between write drivers, sense amplifiers, and precharge devices.

1.2.2 Read Timing Sequence

A complete read cycle begins with stabilization of the row address while the memory is configured in read mode. At the falling edge of CLK , the precharge signal is asserted and all RBLs are driven to V_{DD} . Once precharge is released, RWL is activated during the low phase of CLK , allowing the storage capacitor to share charge with the RBL. If the stored data is a logic 0, the RBL voltage falls below the low sensing threshold. If the stored data is a logic 1, the RBL remains above the high sensing threshold. At the subsequent rising edge of CLK , the sense amplifier samples the RBL voltage and latches the corresponding digital output. This three-phase separation between precharge, evaluation, and sensing ensures clean and repeatable read operation.

1.2.3 Write Timing Sequence

A write operation begins with both the row address and input data stabilized at the memory inputs. With write enable asserted, $CLKB$ transitions high to activate WWL for the selected row. During this phase, the write driver forces the WBL either high or low depending on the input data value. The storage capacitor charges or discharges accordingly through the write access transistor. WWL is then disabled when $CLKB$ falls, completing the write operation. Write operations are confined entirely to the complementary clock phase from read operations to prevent functional overlap and electrical conflicts.

1.2.4 Minimum Clock Period

The minimum allowable clock period is governed by the worst-case capacitor charging time during a write operation. The measured maximum write delay is 68.377 nanoseconds. A conservative safety factor is applied to this value to ensure complete voltage settling across all operating conditions, resulting in a minimum clock period of 80 nanoseconds and a corresponding maximum operating frequency of 12.5MHz.

1.2.5 Precharge and Evaluation Constraints

The precharge interval must be long enough for all RBLs to fully reach V_{DD} before RWL activation. If precharge is incomplete, the read operation will be referenced to an incorrect baseline, leading to logic errors and degraded sensing margin. Similarly, the evaluation interval must allow sufficient time for charge sharing to develop a measurable RBL voltage deviation before the sense amplifier samples the signal. These timing margins are enforced through the relative placement of PCB, RWL, and the clock edges within each cycle.

1.2.6 Refresh Timing Requirement

To prevent data loss due to leakage, every row must be refreshed within the maximum allowable retention window. With 16 total rows, this sets the maximum allowable refresh spacing per row. The refresh controller sequences through all rows periodically and temporarily blocks normal memory access during refresh to guarantee safe and complete charge restoration.

1.3 Transistor Sizing Rationale

The transistor dimensions throughout the DRAM were selected using a minimum-sizing strategy guided directly by experimental timing, retention, noise margin, and energy measurements obtained from post-layout and schematic-level simulations. The overall optimization objective is to minimize the figure of merit (FOM), defined as the product of delay, area, and energy. Since both layout area and switching energy scale approximately linearly with transistor width, minimum sizing provides a clear advantage in two of the three optimization dimensions. The remaining requirement is that minimum-sized devices must demonstrably meet functional delay, retention, and sensing constraints, which is verified through the measured results reported throughout this section.

1.3.1 Memory Cell Transistor Sizing

Within the 3T1C DRAM storage cell, all three NMOS devices are selected at minimum width based on experimentally observed write delay and retention behavior. The write access transistor must be strong enough to charge and discharge the storage capacitor within the allotted write window, while simultaneously limiting off-state leakage that degrades retention time. Increasing the width reduces write delay but measurably increases subthreshold and junction leakage due to the larger drain diffusion area. Our measured retention behavior confirmed that minimum sizing provides the longest storage time while still meeting write-speed requirements.

The read access transistor is likewise kept at minimum width based on measured sensing performance. Experimental read measurements verified that sufficient bitline voltage perturbation is achieved for reliable detection without requiring additional device width. The isolation device is also minimum sized to minimize experimentally observed parasitic junction capacitance, which directly impacts the measured capacitor discharge rate.

1.3.2 Write Path Considerations

The write delay is fundamentally dominated by the time required to charge the storage capacitor through the NMOS access transistor toward V_{DD} . This worst-case condition occurs when writing a logic 1 and was directly measured as a worst-case write delay of 68 ns. In contrast, the write-low delay was measured to be significantly smaller, 0.21ns, due to the stronger discharge path to ground. Since the experimentally observed worst-case delay already satisfies the target clock period, no functional justification exists for increasing device dimensions.

1.3.3 Leakage and Retention Tradeoff

Charge loss from the storage capacitor is governed by measured subthreshold and junction leakage currents of the access transistors. These mechanisms were experimentally characterized by observing the capacitor discharge trajectory over time. The measured retention interval directly establishes the refresh constraint of the memory. Because leakage magnitude scales monotonically with device width, increasing the transistor size was observed to reduce the measured retention time. Minimum sizing therefore provides the optimal operating point for dynamic storage.

1.3.4 Precharge and Bitline Driver Sizing

The precharge devices were selected at minimum width and validated through direct precharge time measurements. Precharge time is small relative to the overall clock period and therefore does not lie on the critical timing path. The write drivers are implemented using minimum-sized tristate

inverters and were experimentally verified to drive the full column bitline capacitance within the measured worst-case write delay of 68ns, confirming adequate drive strength without oversizing.

1.3.5 Sense Amplifier Sizing and Noise Margins

Each column employs a minimum-sized inverter as the sense amplifier. The inverter switching thresholds were obtained directly from DC sweep measurements and are reported as $V_{IH} = 0.662V$ and $V_{IL} = 0.573V$. Transient read measurements verified that the post-charge-sharing bitline voltage consistently crossed these thresholds. Since reliable sensing was experimentally demonstrated across all measured operating points, no increase in sense amplifier transistor size was required.

1.3.6 Peripheral Logic and Decoder Sizing

All logic gates in the row decoder, wordline control, and refresh controller are implemented using minimum-width devices. Exhaustive address sweep measurements across all 16 rows verified correct one-hot decoding and proper wordline activation timing, with no observed functional failures. Since measured decoder delay remains well below the worst-case write delay of 68 ns, the decoder does not limit the maximum operating frequency.

1.3.7 Energy Considerations

The dynamic energy consumption of the DRAM was measured directly under a repeated write-read stimulus. The measured energy per memory operation is 11.66 pJ, confirming that minimum transistor sizing minimizes switching capacitance and short-circuit current. No measurable energy benefit was observed from upsizing any device class, further reinforcing the minimum-sizing strategy.

1.3.8 Summary of Experimentally Guided Sizing Decisions

In summary, all transistor dimensions in the proposed DRAM are selected based on measured delay, retention, noise margin, and energy behavior rather than purely analytical expectations. Minimum transistor sizing was retained throughout the memory because it simultaneously minimizes measured area and energy while maximizing measured retention time in the dynamic storage nodes. Since the experimentally observed worst-case write and read delays satisfy the required clock period, no functional or performance motivation exists for increasing device dimensions. This measured-data-driven approach yields an optimal balance between performance, power efficiency, and layout density for the proposed DRAM architecture.

1.4 Modern DRAM Optimizations

In the standard version of this project, each team must implement both a baseline memory macro and a significantly optimized version, and then quantify the improvement in a figure of merit that combines area, energy, and performance. Our group instead chose a more challenging DRAM-based design with a 3T1C bitcell as the baseline, and, in consultation with the course staff, we were not required to also produce and evaluate a second, optimized macro. In lieu of a second physical implementation, this subsection surveys a small, non-representative subset of modern DRAM research that targets the same metrics as our project (effective access latency, dynamic and static energy, and area overhead), but at the level of bitcell topology and very-low-level peripheral circuitry. DRAM optimization is a very broad field, including topics such as 3D integration, new interfaces, security, near-memory compute, and system-level refresh policies; the works selected here focus narrowly on cell-level gain-cell eDRAM design and sense-amplifier and bitline energy reduction, which are the closest analogues to the choices we make in our 3T1C macro.

A useful starting point is the survey and taxonomy of gain-cell eDRAM (GC-eDRAM) structures by Teman et al. [1]. They classify embedded DRAM bitcells that can be fabricated in standard logic CMOS without deep-trench or stacked capacitors, and contrast them with conventional 1T1C cells. In a 1T1C cell the storage capacitance is provided by a dedicated capacitor device, so density is excellent but the process requires additional capacitor steps and the read operation is destructive, forcing immediate restore. In logic-compatible gain cells, the effective storage capacitance is the parasitic capacitance at an internal storage node, written and read through one or more transistors. Teman et al. show that 2T and 3T gain cells generally occupy more area than an optimized custom 1T1C cell, but still achieve substantially higher density than compiled 6T SRAM in the same technology, while offering non-destructive read and the ability to decouple write and read ports. The survey also emphasizes how data retention is limited by subthreshold, gate, and junction leakage at the storage node, and how device choices (high- V_T versus regular- V_T), biasing of the wordlines during hold, and boosting or undervolting of write lines all trade off retention time against write margin and peripheral complexity.

Gitterman et al. push this line of work into a modern FinFET process with a 1 Mbit 3T gain-cell eDRAM implemented in a 16 nm logic-compatible FinFET technology [2]. Their bitcell uses three transistors with carefully chosen threshold voltages to balance leakage paths to and from the storage node. The implemented 1 Mbit array is fully logic-compatible and provides a bitcell size that is approximately $2\times$ smaller than a 6T SRAM cell drawn with the same design rules, while keeping the array layout on standard logic tracks [2]. Measurement results demonstrate a data-retention time of about 77 μs under $V_{DD} = 0.6\text{ V}$, which is more than $10\times$ longer than previously reported GC-eDRAM implementations in 28 nm technologies, and correct operation was demonstrated from -40°C to 125°C and down to a minimum supply voltage of roughly 0.45 V [2]. At the macro level, the design uses replica gain cells and self-timed control to adapt the read pulse width to the actual bitline development and employs write drivers that restore the write bitlines after each operation to suppress leakage of stored “1” levels. For our 3T1C macro, this paper is a concrete proof that a 3T gain cell can be competitive in density with SRAM in advanced nodes while still achieving tens of microseconds of retention without explicit capacitors, provided that the leakage paths and peripheral timing are co-optimized.

Kim and Park take a different approach and explicitly trade additional static circuitry for effectively infinite retention time by turning a 2T gain cell into a pseudo-static embedded DRAM suitable for

analog processing-in-memory (PIM) [3]. They start from a conventional 2T1C-style gain cell and show, via simulation across technologies from 180 nm to 28 nm, that retention time for a fixed cell structure degrades severely as the channel length shrinks because the storage-node capacitance is reduced and subthreshold leakage grows. They then introduce a pseudo-static gain cell (PS-GC) that combines a 2T storage cell with an active leakage-compensation network that sources or sinks current to hold the storage-node voltage near its ideal level during retention. In a 28 nm CMOS implementation, the resulting bitcell area is reported as $0.79\times$ that of a compiled 6T SRAM cell and $0.58\times$ that of an 8T SRAM cell optimized for analog MAC operations, even though the PS-GC uses five transistors, because it avoids a dedicated large metal-oxide-metal capacitor inside the cell [3]. Post-layout simulations of a 64×64 macro show that the PS-GC array maintains pseudo-static operation with effectively unlimited retention time over wide process, voltage, and temperature ranges, and that at an operating frequency of 667 MHz it can function from 0.9–1.2 V and -25°C to 85°C with both read and write access times below 0.3 ns at 1.2 V and 85°C and a static power of about 2.2 nW/bit at 25°C [3]. Conceptually, this work demonstrates that adding an active leakage-compensation path can stretch retention essentially to pseudo-infinite, turning DRAM into SRAM-like storage without an explicit storage capacitor, at the cost of additional devices and static bias currents. Our 3T1C cell does not implement the full PS-GC assist circuitry, but the same idea of explicitly engineering all leakage paths around the storage node appears in our device choices and biasing schemes for the write wordline.

The previous works primarily manipulate the bitcell itself. A complementary axis of low-level DRAM optimization lies in the sensing and precharge circuitry, where every access incurs energy to precharge the bitlines to approximately $V_{\text{DD}}/2$, to sense the small voltage perturbation caused by charge sharing with the cell capacitance, and then to restore both the cell and the bitline. Dai et al. analyze this energy flow and propose single-bitline-load sense-amplifier (SBLSA) circuits that reduce bitline and sense-amplifier energy in a DRAM array by avoiding unnecessary full-swing bitline transitions [4]. Building on an earlier single-bitline-write (SBW) scheme, they introduce two SBLSA variants: a redundant-voltage-discharged SBLSA (RVD-SBLSA) that uses an extra discharge path to clamp the target bitline back toward $V_{\text{DD}}/2$ after a read or write, and a bit-aware SBLSA (BA-SBLSA) that conditionally enables single- or dual-bitline loading based on the stored data pattern [4]. Circuit-level simulations in a 65 nm CMOS process show that both variants reduce total read and write energy relative to a conventional differential sense amplifier across data patterns, with the RVD-SBLSA giving the lowest energy and the BA-SBLSA offering a more balanced energy-latency trade-off [4]. Although these designs still assume a standard 1T1C cell, the underlying principle is directly relevant to our work: a significant fraction of per-access energy is tied up in the large bitline capacitances and in the precise sequence of precharge, sense, and restore events, so small changes to the sense-amplifier topology and timing can yield meaningful energy savings even when the bitcell is fixed.

Finally, it is useful to place these gain-cell designs in the broader context of contemporary embedded DRAM implementations that still use conventional 1T1C cells. For example, Lin et al. report a 14 nm SOI FinFET CMOS technology that integrates $0.0174\text{ }\mu\text{m}^2$ 1T1C eDRAM cells alongside logic [5], and Hamzaoglu et al. demonstrate a 1 Gb, multi-gigahertz embedded DRAM macro in a 22 nm tri-gate CMOS technology [6]. These macros exploit aggressive capacitor and trench scaling, deep circuit-level optimization of the sense amplifiers and reference schemes, and carefully tuned refresh to meet high-speed cache requirements. Compared to such state-of-the-art 1T1C macros, gain-cell eDRAM like the 3T structures discussed above gives up some areal density in exchange for

logic compatibility (no special capacitor steps), non-destructive reads, and additional ports or assist circuits that can be tailored to specific SoC blocks (for example, analog PIM engines). Our choice of a 3T1C cell in this project follows this trend: we accept more transistors per bitcell than a pure 1T1C array would require, but we gain better read stability, a clean separation of write and read paths, and the ability to tune leakage and timing at the circuit level using only the devices provided by a standard CMOS process. The literature reviewed here shows that, in modern technologies, such gain-cell DRAMs can reach retention times in the tens of microseconds with modest area overhead and can be paired with carefully engineered sense-amplifier and bitline circuits to significantly reduce per-access energy, directly aligning with the area–energy–delay trade-offs that define our project’s figure of merit.

2 Schematics with Size Annotations

INV

The minimum inverter is the basic logic primitive and defines the reference drive strength used to size all other gates in the periphery.

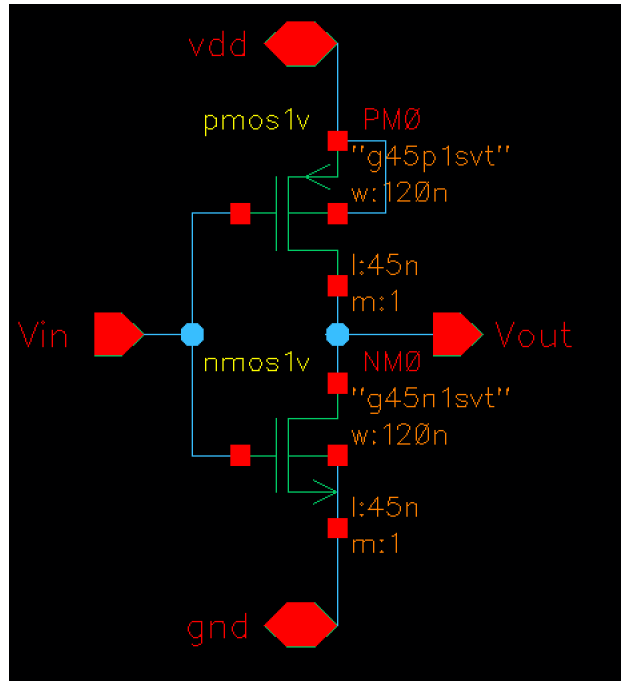


Figure 1: Transistor-level schematic of the minimum CMOS inverter. Both the PMOS pull-up and NMOS pull-down use $W = 120$ nm and $L = 45$ nm, giving a compact, low-capacitance reference cell.

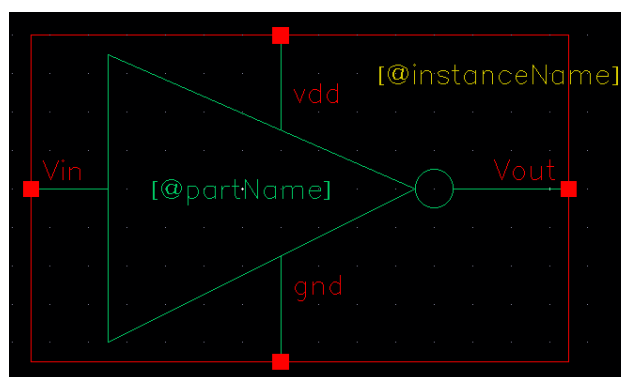


Figure 2: Symbol for the minimum inverter in Figure 1, exposing a single input, output, and the global vdd/gnd rails for reuse across the design.

NAND2

Two-input NAND gates implement much of the row-decoder and control logic, so they are also kept at the same minimum geometry as the inverter.

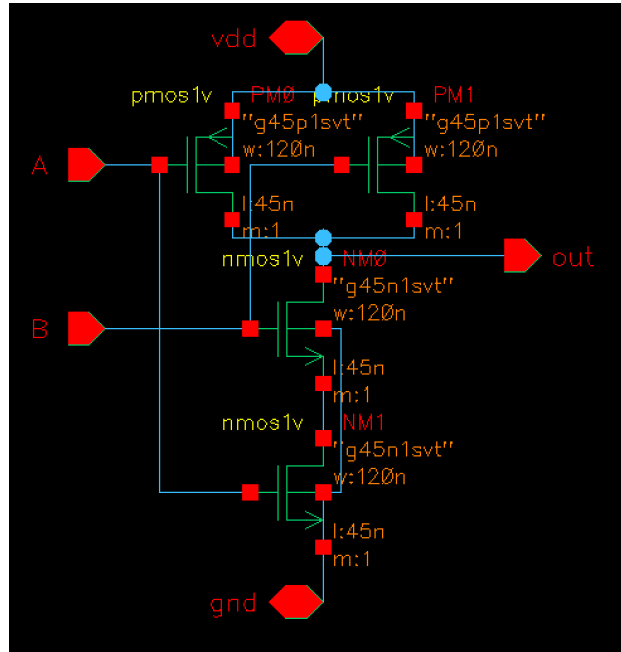


Figure 3: Transistor-level schematic of the two-input NAND gate. The pull-up network has two PMOS devices in parallel and the pull-down network has two NMOS devices in series, all with $W = 120$ nm and $L = 45$ nm.

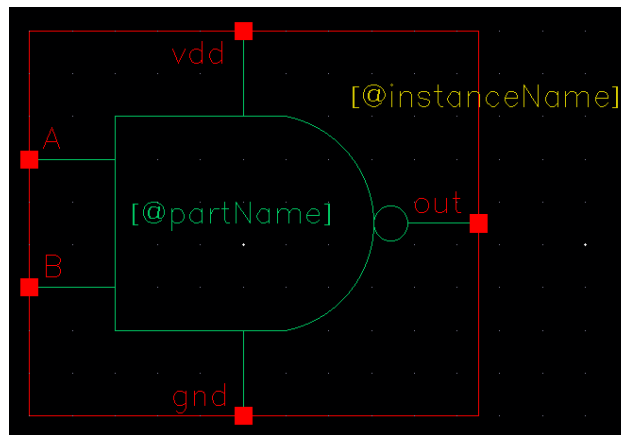


Figure 4: Symbol for the two-input NAND used in the row decoder and finite-state control logic, with inputs A/B and a single output node.

NOR2

NOR gates are used as heavily as NANDs in the row decoder and also appear in the `clk_gen` non-overlapping clock generator.

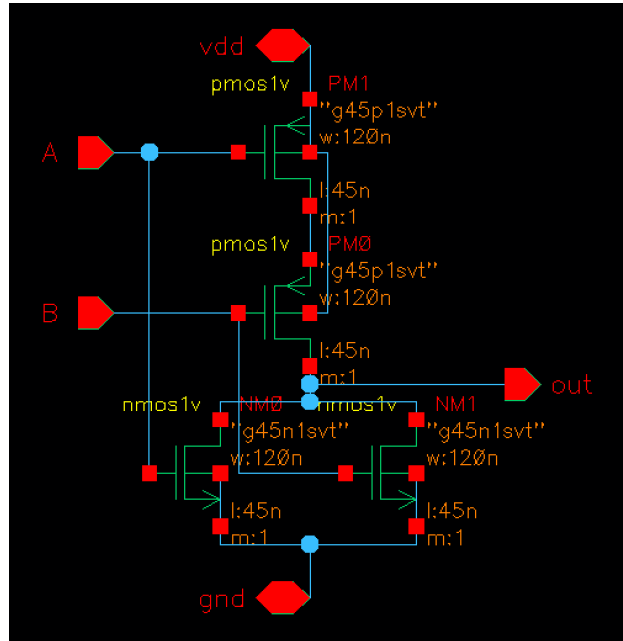


Figure 5: Transistor-level schematic of the two-input NOR gate. Two PMOS devices in series form the pull-up network and two NMOS devices in parallel form the pull-down network, again with $W = 120$ nm and $L = 45$ nm.

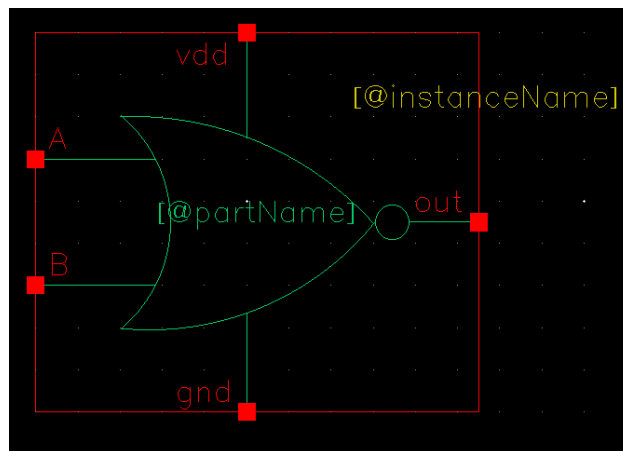


Figure 6: Symbol for the two-input NOR used in both the row decoder and `clk_gen`, providing a compact OR-of-complements stage.

XOR2

XOR gates appear only in non-critical control paths (for example, simple parity and state encoding), so a compact minimum-sized implementation is sufficient.

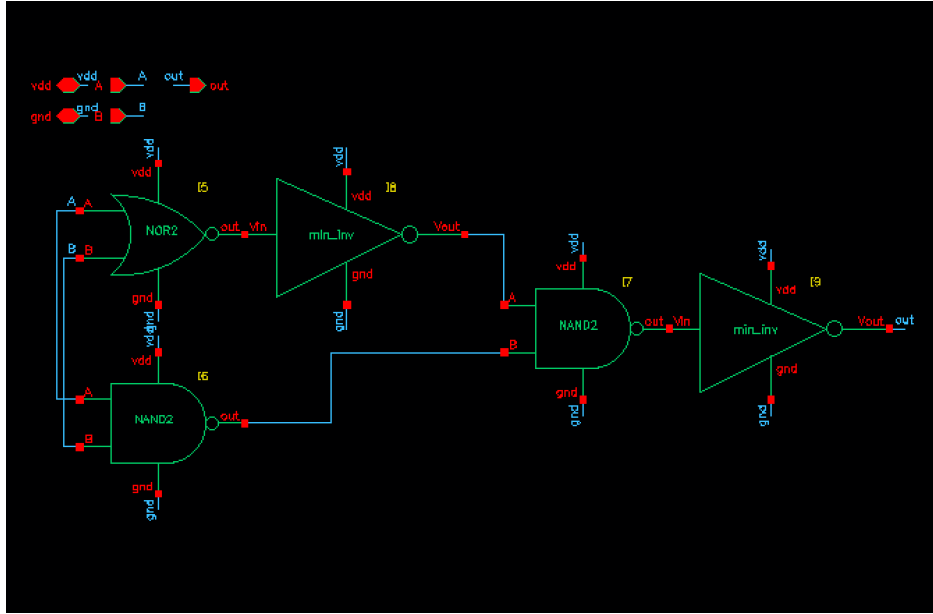


Figure 7: Transistor-level schematic of the two-input XOR gate, implemented as a small network of minimum-sized NAND, NOR, and inverter stages, all using $W = 120$ nm and $L = 45$ nm.

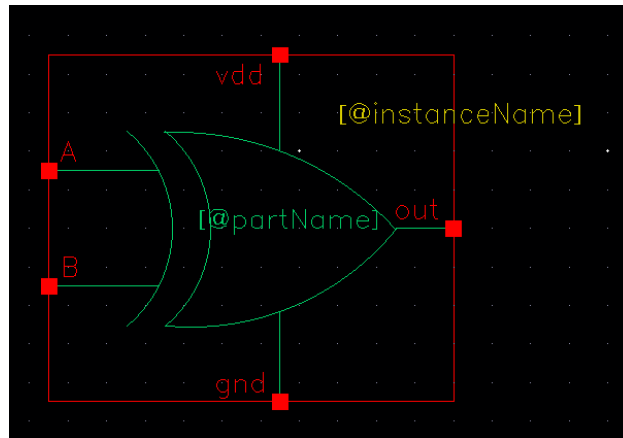


Figure 8: Symbol for the two-input XOR used in the controller, with inputs A/B and a single parity-like output.

AND3

Three-input AND gates are used to gate wordline enables and other qualified control signals and are built from the same minimum-sized primitives.

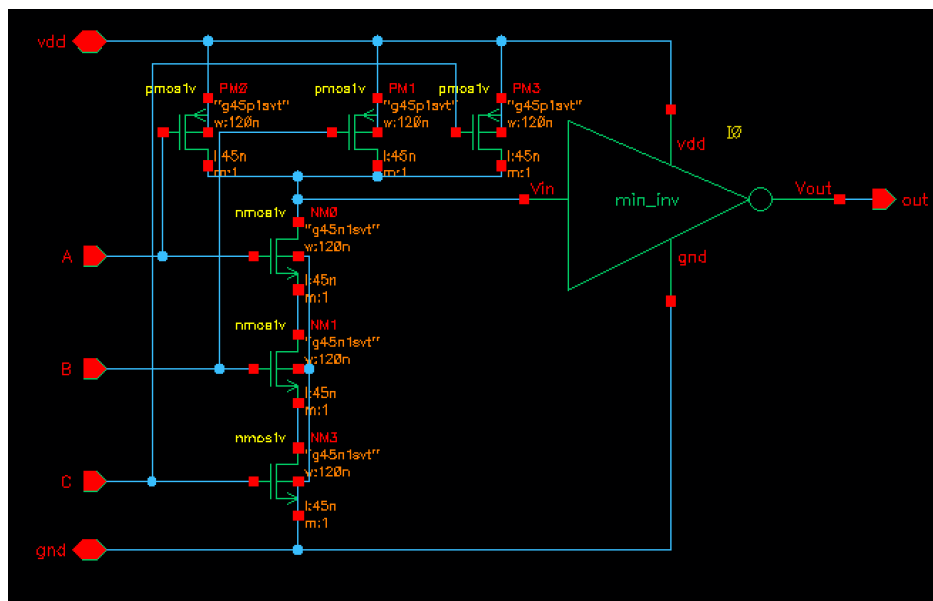


Figure 9: Transistor-level schematic of the three-input AND gate, realized as a three-input NAND (three series NMOS, three parallel PMOS, $W = 120$ nm, $L = 45$ nm) followed by a minimum-sized inverter to restore full logic levels.

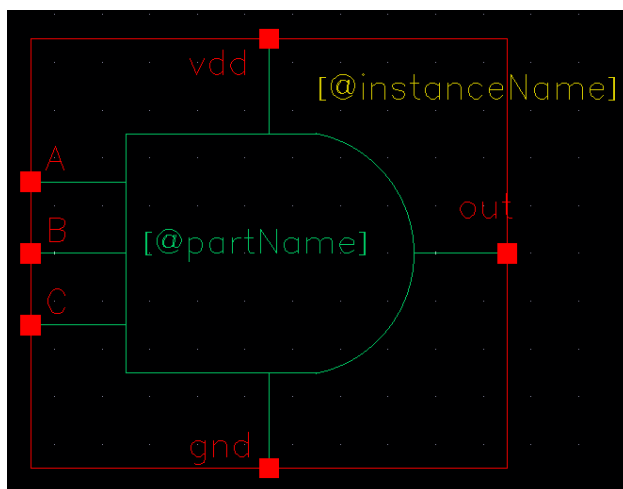


Figure 10: Symbol for the three-input AND gate with inputs A/B/C and a single output that drives wordline and control-enable signals.

3T1C Bitcell

Each storage element in the array uses the 3T1C topology discussed in the memory description: three NMOS transistors surrounding an explicit storage capacitor on node X. The leftmost access device connects the write bitline WBL to the capacitor node X when the write wordline WWL is asserted, allowing full-swing writes that charge or discharge the 1 pF storage capacitor. The two right-hand devices form the read and isolation path into the read bitline RBL under control of the read wordline RWL; during a read, the stored voltage on X perturbs the precharged RBL so that the column sense inverter can resolve the stored bit. All three transistors use the minimum geometry $W = 120$ nm and $L = 45$ nm, so the bitcell area is dominated by the explicit capacitor while leakage and junction capacitances are kept as small as possible.

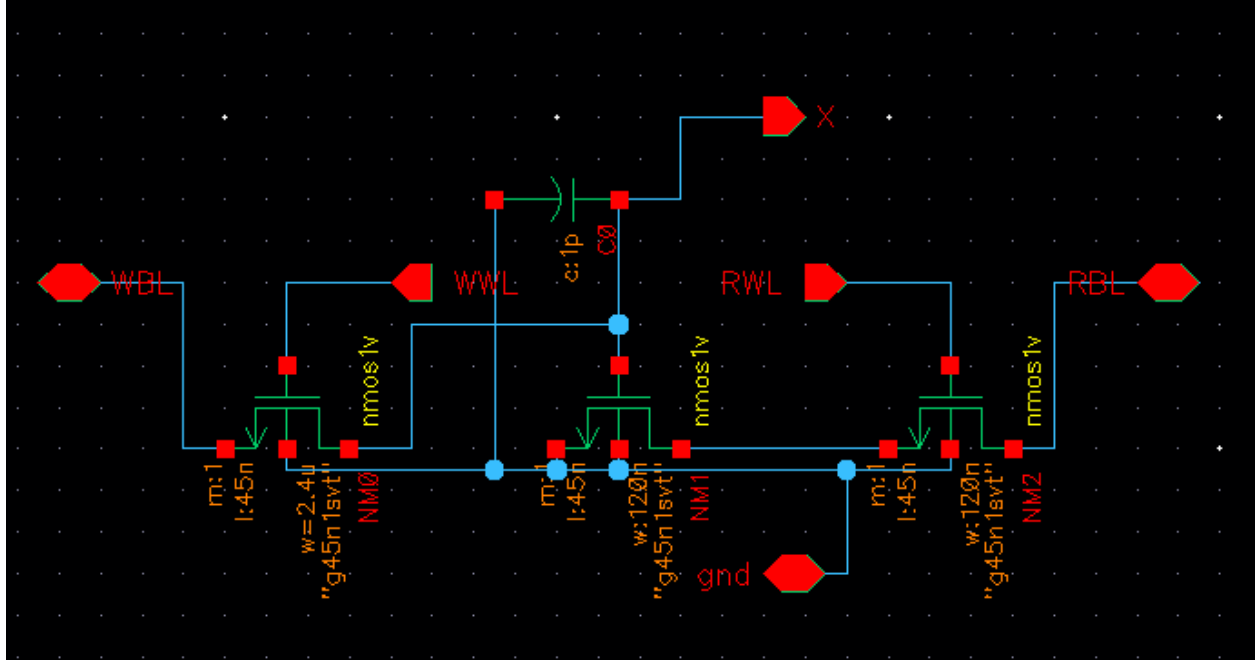


Figure 11: Transistor-level schematic of the 3T1C DRAM bitcell. A minimum-sized write access transistor controlled by WWL connects WBL to the 1 pF storage capacitor on node X, while two minimum-sized read and isolation devices couple the stored voltage to the read bitline RBL under control of RWL.

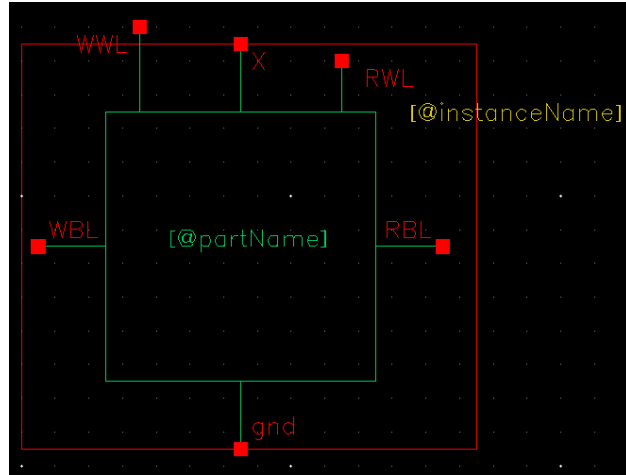


Figure 12: Symbol for the 3T1C bitcell used in the array, exposing the write and read wordlines (WWL, RWL), the write and read bitlines (WBL, RBL), the internal storage node X, and the shared ground connection for integration into the 16×4 DRAM macro.

Precharge Circuit

The read bitline precharge network uses dedicated PMOS devices to force every RBL to a known reference of V_{DD} before each read evaluation. At the falling edge of CLK the precharge control (PCB) is asserted, turning on the column precharge transistors and charging the full bitline capacitance to V_{DD} . Once this phase is complete PCB is deasserted, the precharge devices turn off, and the RBLs are left floating so that only the selected 3T1C cell can perturb the bitline during charge sharing. Because the sense amplifier is a simple inverter whose threshold assumes a well defined starting level, complete precharge is critical: any residual droop would shift the effective reference and reduce noise margin. All precharge devices are kept at the same minimum width as the rest of the periphery so that they do not dominate area or load, and no equalization device is required since each column uses a single-ended RBL rather than a complementary pair.

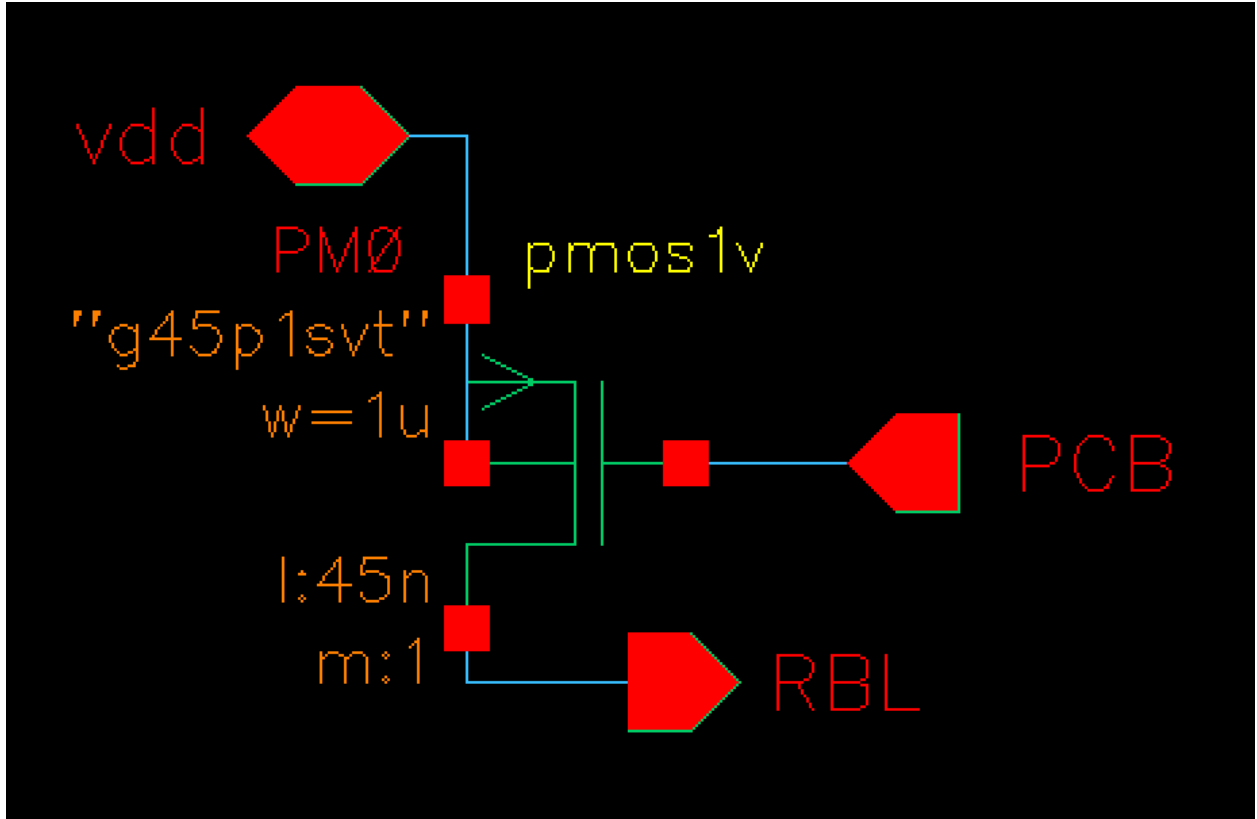


Figure 13: PMOS-based precharge circuit for the read bitlines. A global precharge control PCB enables the minimum-sized PMOS devices that connect each RBL to V_{DD} during the precharge window at the falling edge of CLK. When PCB is low the precharge devices are off, leaving the RBL floating at V_{DD} so that the selected 3T1C cell can safely discharge it during the subsequent read-evaluation phase.

Row Decoder (4-to-16)

The 4-bit row address **A3:A0** is decoded into sixteen one-hot row-select signals using only Tier-0 minimum-sized gates. Each address bit is first buffered and inverted by a minimum inverter so that both the true and complement rails are available. These eight rails are then routed horizontally into a small decoder tree of NAND2 and NOR2 gates, which implement the required minterms. For each wordline, a final NOR2 stage produces an active-high select signal (**out0–out15**) that drives the corresponding wordline driver. All devices in the decoder reuse the $W = 120$ nm, $L = 45$ nm primitives, so the block is compact and its delay is dominated by wiring rather than oversized transistors.

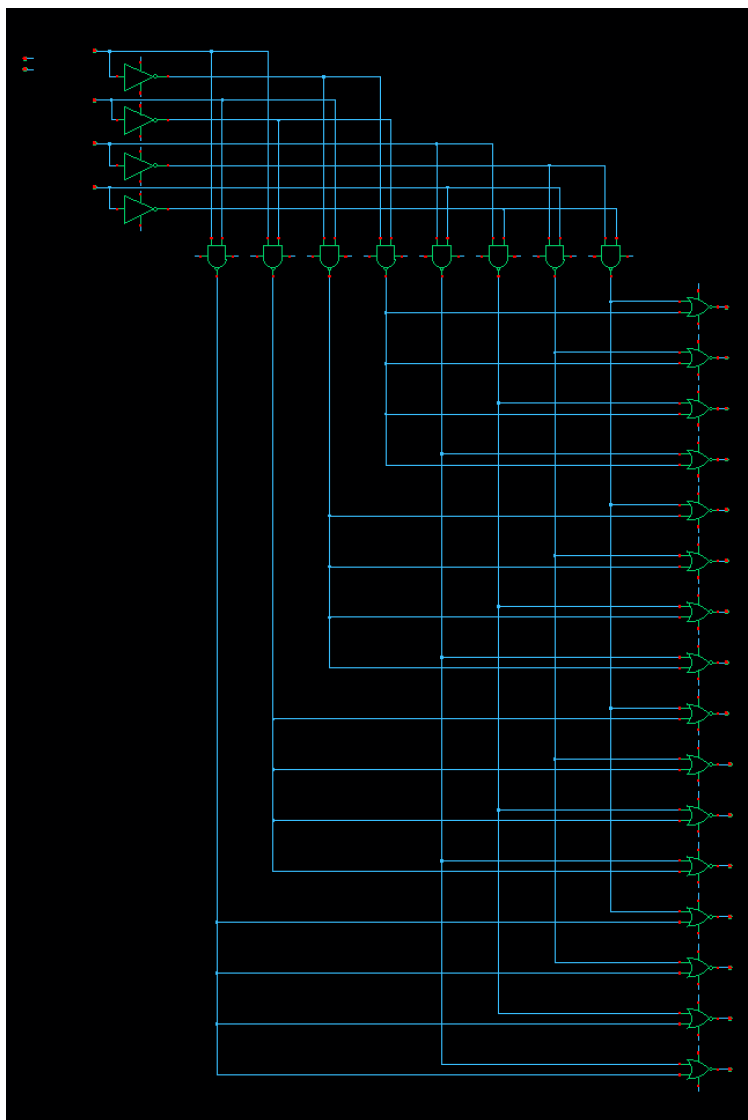


Figure 14: Hierarchical 4-to-16 row decoder schematic. The four address inputs and their complements (generated by minimum inverters on the left) feed a network of minimum-sized NAND2 and NOR2 gates, with each final NOR2 producing a single active-high row-select output.

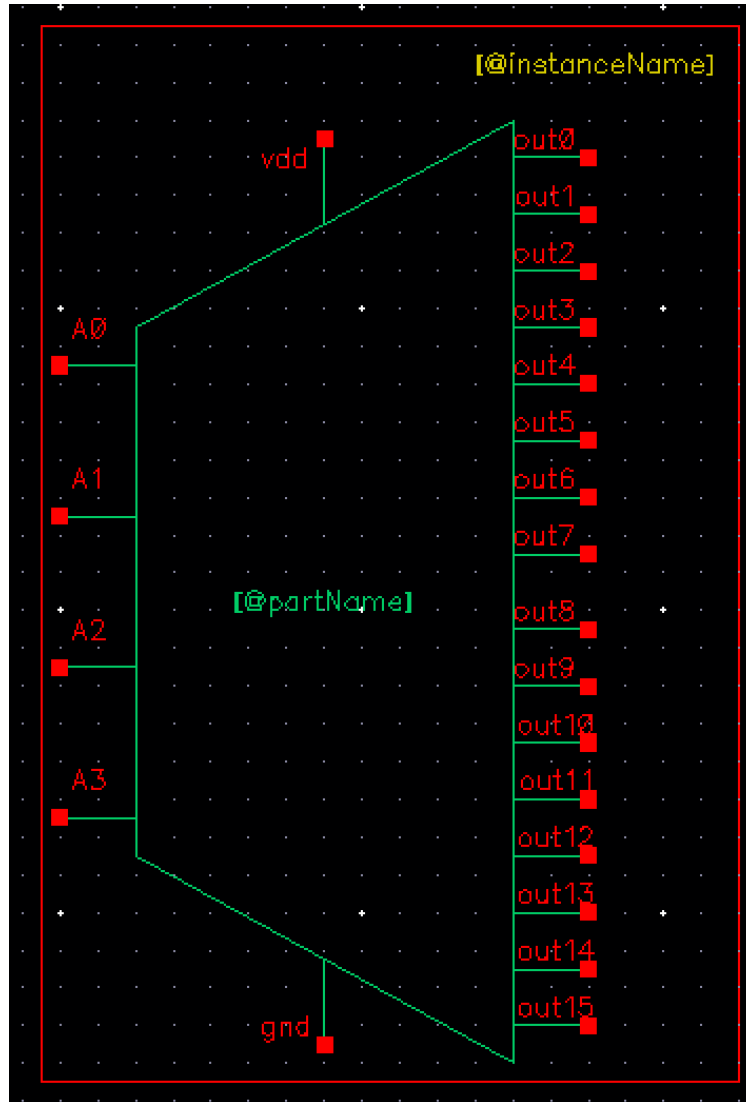


Figure 15: Symbol for the 4-to-16 row decoder in Figure 14, showing the four address inputs A0{A3, global vdd/gnd rails, and the sixteen one-hot outputs out0 - out15 that connect to the wordline driver stage.

CLK Block

The `clk_gen` block converts the single-ended input clock `clk_in` into the two non-overlapping phases `clk_a` and `clk_b` used throughout the DRAM array. An ideal 50% duty-cycle square wave drives `clk_in`; inside the block this signal is buffered, inverted, and passed through an additional buffer chain to introduce a controlled delay. Two NOR2 gates then combine the original and delayed versions of the clock so that each output is high for roughly half a cycle but the high intervals never overlap. During the low-low “dead time” both phases are deasserted, which keeps precharge, evaluation, and write operations from being active at the same time and prevents short-circuit current in the periphery.

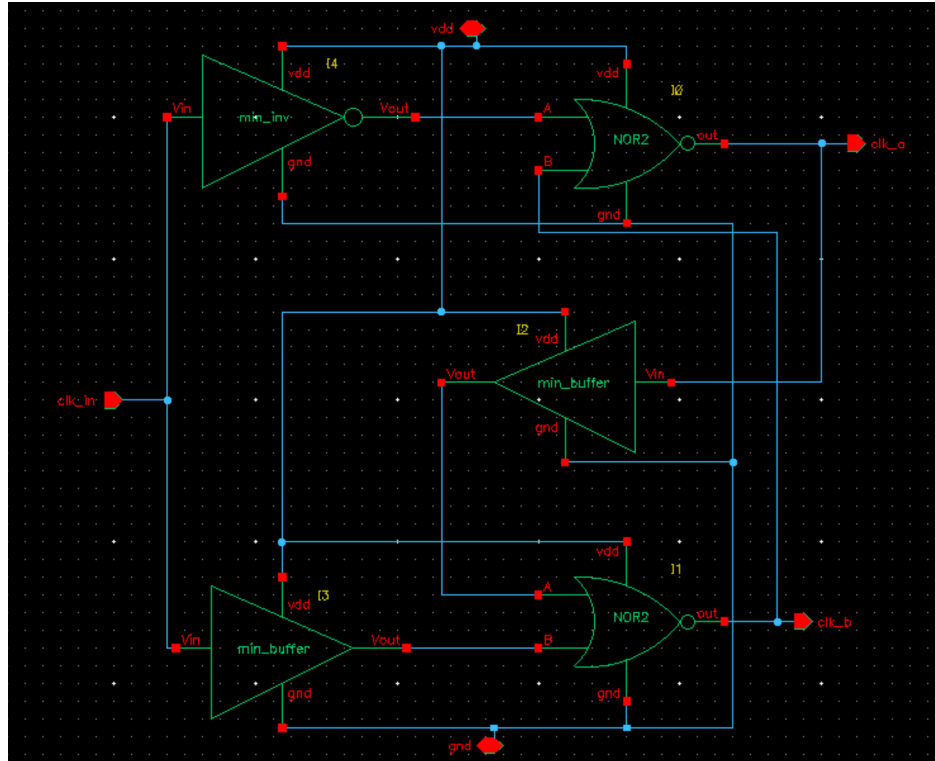


Figure 16: Transistor-level schematic of the `clk_gen` block. Minimum-sized inverters/buffers create delayed versions of `clk_in`, and two NOR2 gates combine these nodes to generate the non-overlapping phases `clk_a` and `clk_b`.

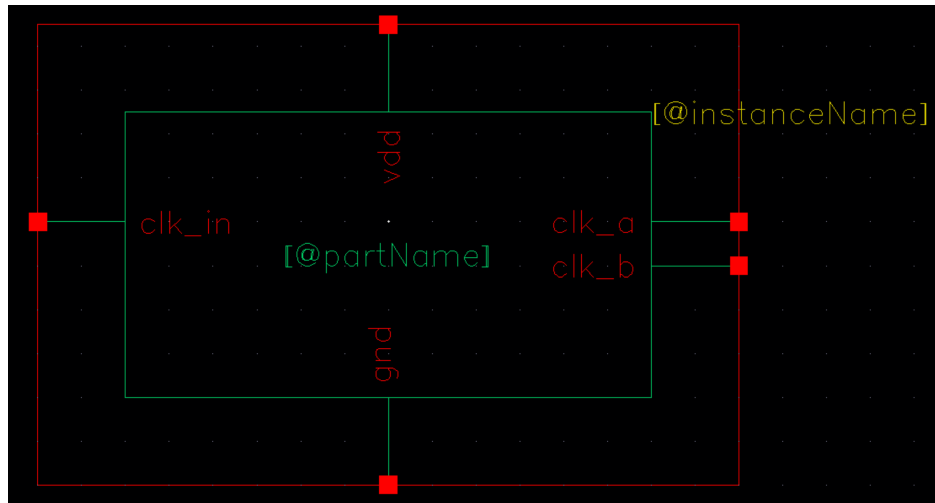


Figure 17: Symbol view of the `clk_gen` block with single input `clk_in`, outputs `clk_a` and `clk_b`, and global `vdd/gnd` connections for integration at the top level.

2:1 MUX

A small 2:1 multiplexer is used where we need to select between two local signals (for example, alternate data paths in the control logic). The cell takes inputs I_0 and I_1 and a single select line S , and drives one output out . Internally, S is first buffered and inverted using a minimum-sized CMOS inverter so that both S and \bar{S} are available. These complementary controls then gate a pair of minimum-sized NMOS pass devices that connect either I_0 or I_1 to an internal node. A final minimum-sized inverter buffers this node and produces the rail-to-rail output out .

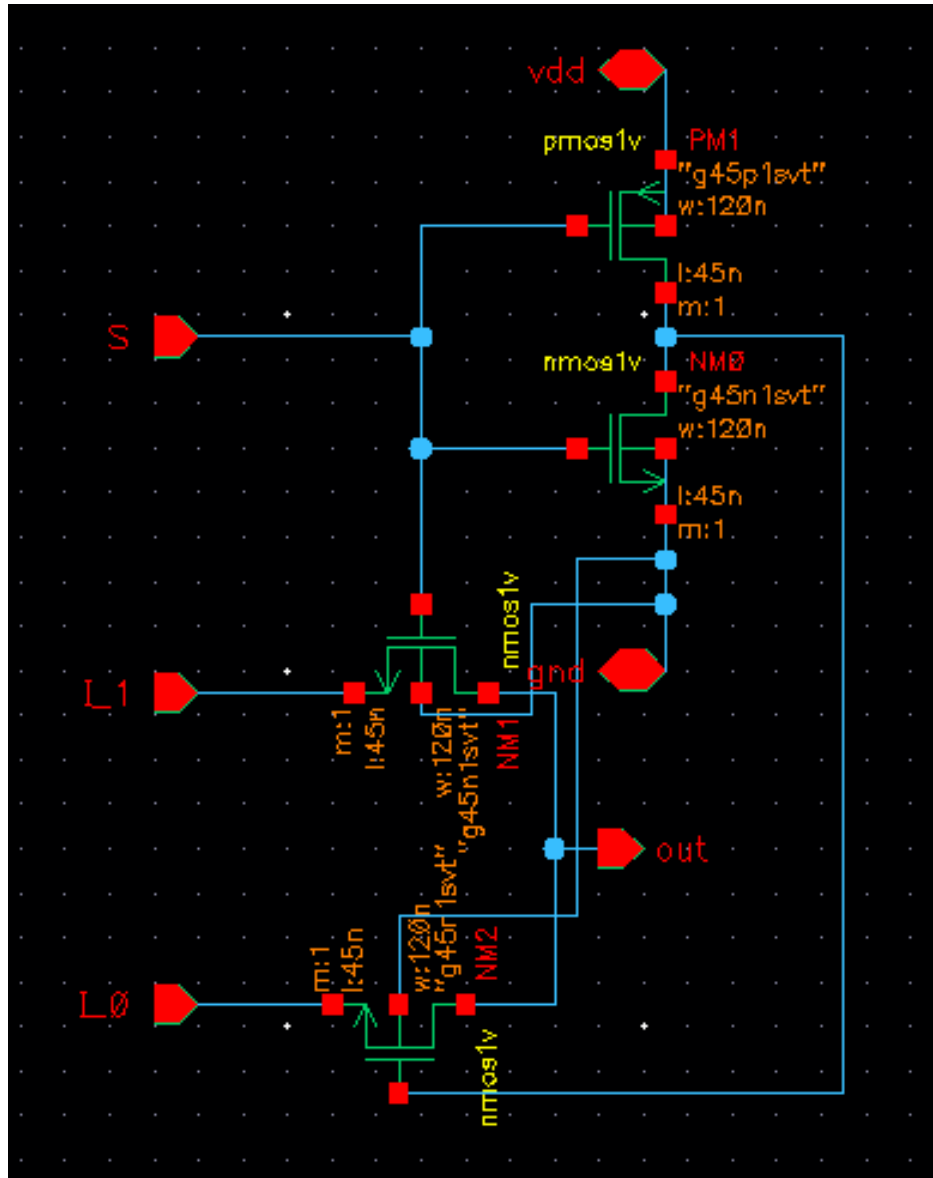


Figure 18: Transistor-level schematic of the 2:1 MUX. The select input S is inverted by a minimum CMOS inverter (PM1/NM0) to generate complementary gate drives for two NMOS pass devices, which steer either I_0 or I_1 to an internal node. A final minimum-sized inverter buffers this node and produces the rail-to-rail output out .

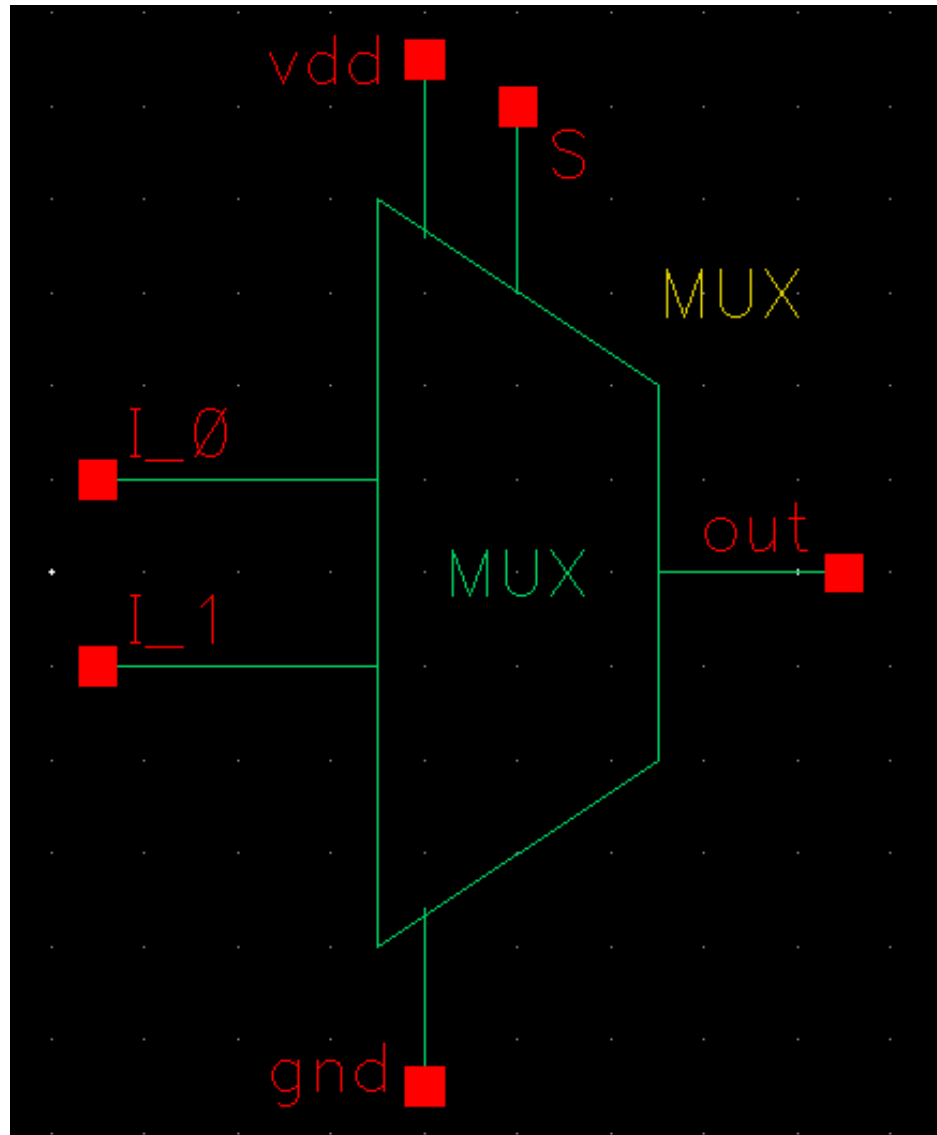


Figure 19: Symbol for the 2:1 MUX cell, showing data inputs I_0 and I_1 , select input S , the single output out , and the global vdd/gnd connections used when instantiating the block in higher-level schematics.

D-Flip Flop

The basic storage element in the design is a positive-edge-triggered D flip-flop with an active-high reset. The cell samples the data input `D` on each rising edge of `clk` and updates the output `Q` accordingly when `reset` is low, so `Q` behaves as a one-bit register. When `reset` is asserted high, the internal reset path forces `Q` to logic 0 on the next clock edge, independent of `D`, providing a clean initialization point for counters and control logic. The only other connections are the global `vdd` and `gnd` rails. This generic, single-bit DFF is instantiated repeatedly in the address counter and any other sequential blocks that require registered state.

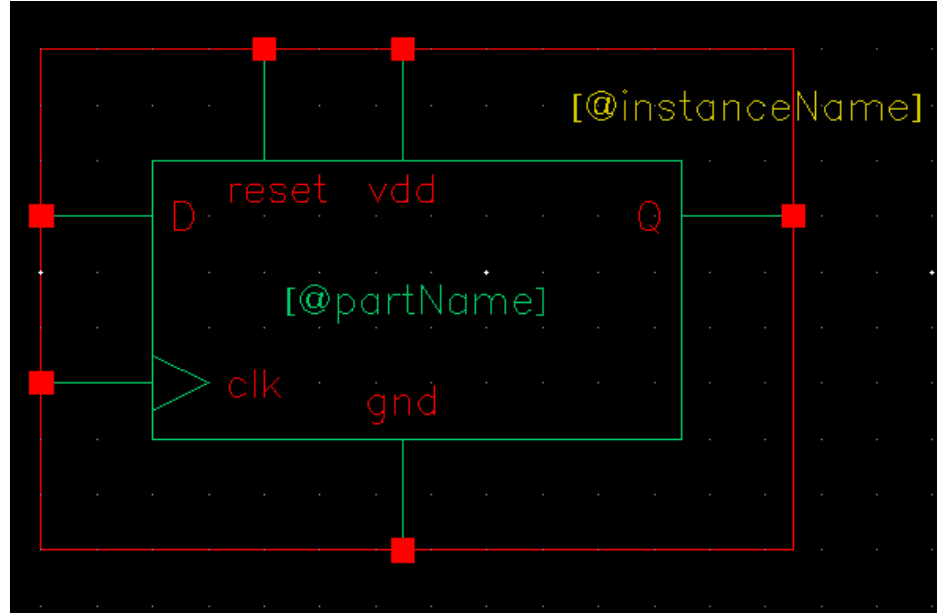


Figure 20: Symbol for the edge-triggered D flip-flop used throughout the design, with data input `D`, clock input `clk`, active-high `reset`, supply connections `vdd/gnd`, and registered output `Q`.

Counter Circuit

A 4-bit synchronous binary counter generates the address sequence used in the DRAM functional tests. The block receives a single clock input `clk` and an active-high reset. Internally, four edge-triggered D flip-flops store the state bits `Q_0`–`Q_3`. A small network of minimum-sized logic gates (AND, XOR, and inverters) implements the “+1” function so that on each rising edge of `clk` the register contents increment by one when reset is low. Bit `Q_0` toggles every cycle, `Q_1` toggles when `Q_0` is high, and so on, producing a modulo-16 count from 0000 to 1111. When `reset` is asserted, all flip-flops are synchronously forced to 0000, guaranteeing a known starting address for every simulation run.

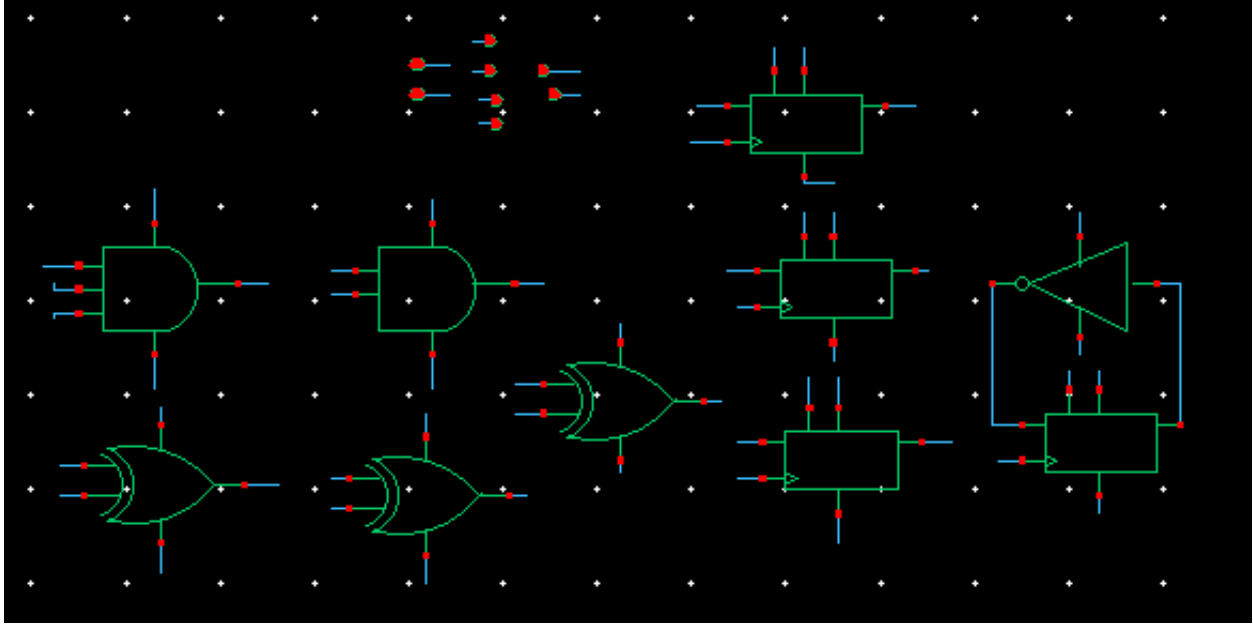


Figure 21: Transistor-level schematic of the 4-bit synchronous counter. Four D flip-flops hold the state bits `Q_0`–`Q_3`, while a network of minimum-sized logic gates computes the next-state inputs so that the register increments by one on each rising edge of `clk` when `reset` is low.

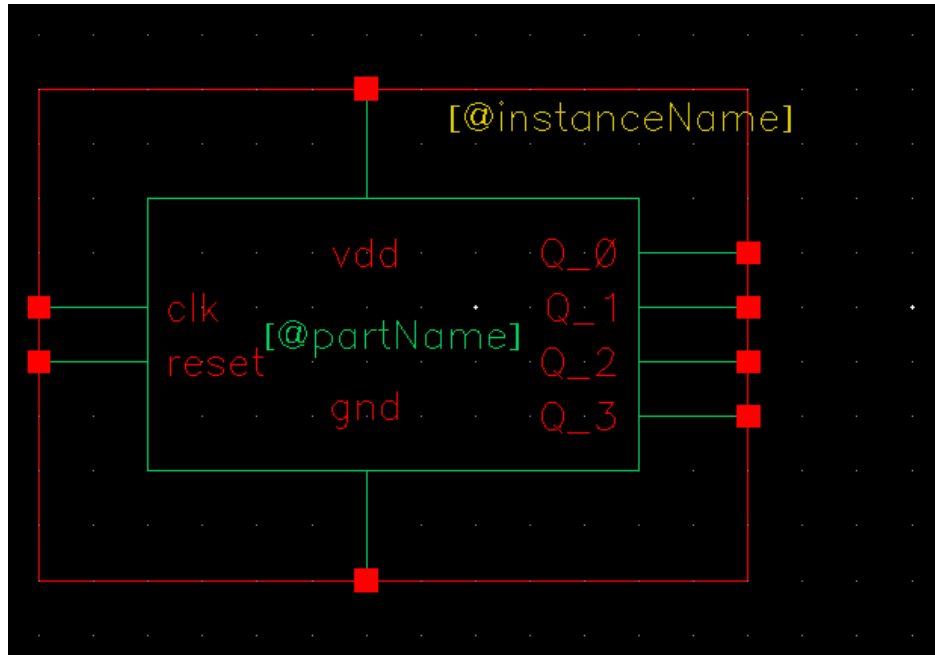


Figure 22: Symbol view of the counter block with inputs `clk` and `reset`, global `vdd/gnd`, and 4-bit outputs `Q_0`–`Q_3` that directly drive the DRAM row-address inputs during functional verification.

Refresh Circuitry

To preserve the charge stored on each 3T1C capacitor over long time scales, the DRAM macro is wrapped by a small refresh controller. Functionally, this block arbitrates between external memory accesses and background refresh operations. In normal operation the external address bits A0–A3, data inputs B0–B3, write-enable (WE), and main clock `clk_in` pass through the controller to the array interface, and the four data outputs `out_0`–`out_3` behave as the standard read data bus. The `reset` input initializes the internal state so that the controller always starts from a known row address and refresh phase.

When a low-frequency `refresh_clk` toggles, the controller enters a refresh cycle. An internal 4-bit counter overrides the external address pins and sequentially selects each of the 16 wordlines, while the control logic generates the appropriate read/restore sequence on WE and the internal timing signals. During this interval the `refresh` output is asserted to indicate that the array is busy and that normal read or write requests should be held off. Each selected row undergoes an internal read followed immediately by a restore operation that recharges or discharges the storage capacitors on node X back to their nominal levels. By cycling through all 16 rows with a refresh interval of 32 milliseconds, the controller guarantees that every bitcell is periodically rewritten before its stored charge can leak below the sensing margin, satisfying the retention requirements established in the single-bitcell characterization.

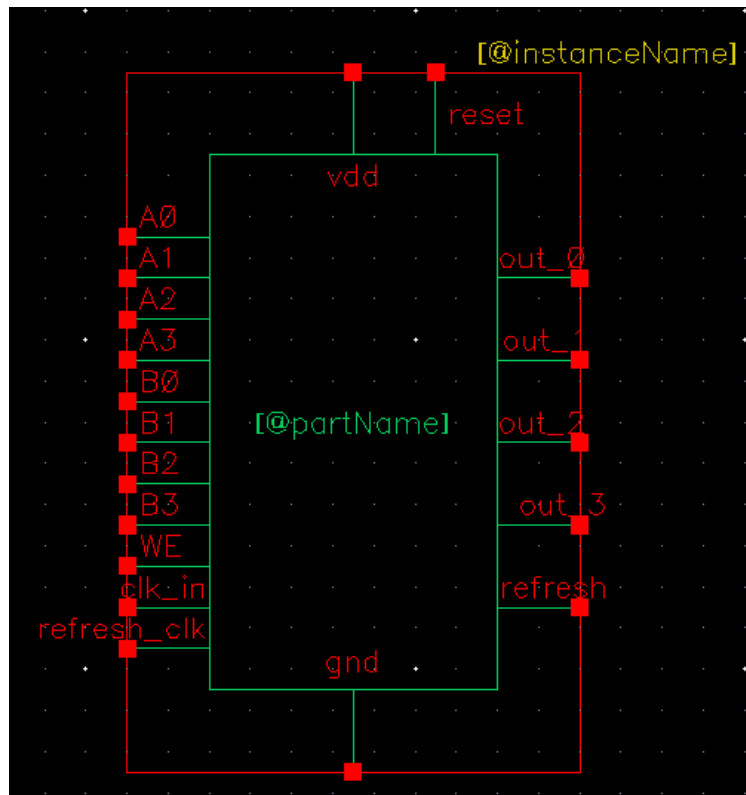


Figure 23: Symbol for the refresh controller wrapper. External address lines A0–A3, data inputs B0–B3, write-enable WE, and the main clock `clk_in` connect on the left, while the 4-bit read data bus `out_0`–`out_3` and the `refresh` status output appear on the right. Separate `refresh_clk`, `reset`, and global `vdd/gnd` pins support autonomous periodic refresh of all 16 rows.

Top-Level DRAM Module

The top-level DRAM macro instantiates the complete 16×4 array of 3T1C bitcells together with all row- and column-periphery required for normal read and write operation. The four address inputs A0–A3 drive a 4-to-16 row decoder, which asserts exactly one wordline for any valid address. Along each of the four columns, a write driver steers the corresponding data input bit (B0–B3) onto the write bitline during a write cycle, while the read bitline is precharged, connected to the selected 3T1C cell during a read, and then resolved by a minimum-sized inverter sense amplifier. All of these blocks share the global clock input `clk_in` and reuse the Tier-0 gates (inverters, NAND2, NOR2, AND3, XOR2) described earlier, so that the array timing and drive strengths match the block-level characterizations.

Externally, the macro presents a simple synchronous interface: `clk_in`, the 4-bit row address bus (A0–A3), the 4-bit data input bus (B0–B3), write-enable `WE`, and the 4-bit data outputs `out_0`–`out_3`, along with global `vdd` and `gnd`. When `WE` is high on a rising edge of `clk_in`, the selected row is written with the value on B0–B3; when `WE` is low, the selected row is read and its contents appear on `out_0`–`out_3` after the internal precharge, charge-sharing, and sensing sequence completes. This block is the core memory macro used in the DRAM functional-verification testbenches and serves as the target of the refresh controller described in the next subsection.

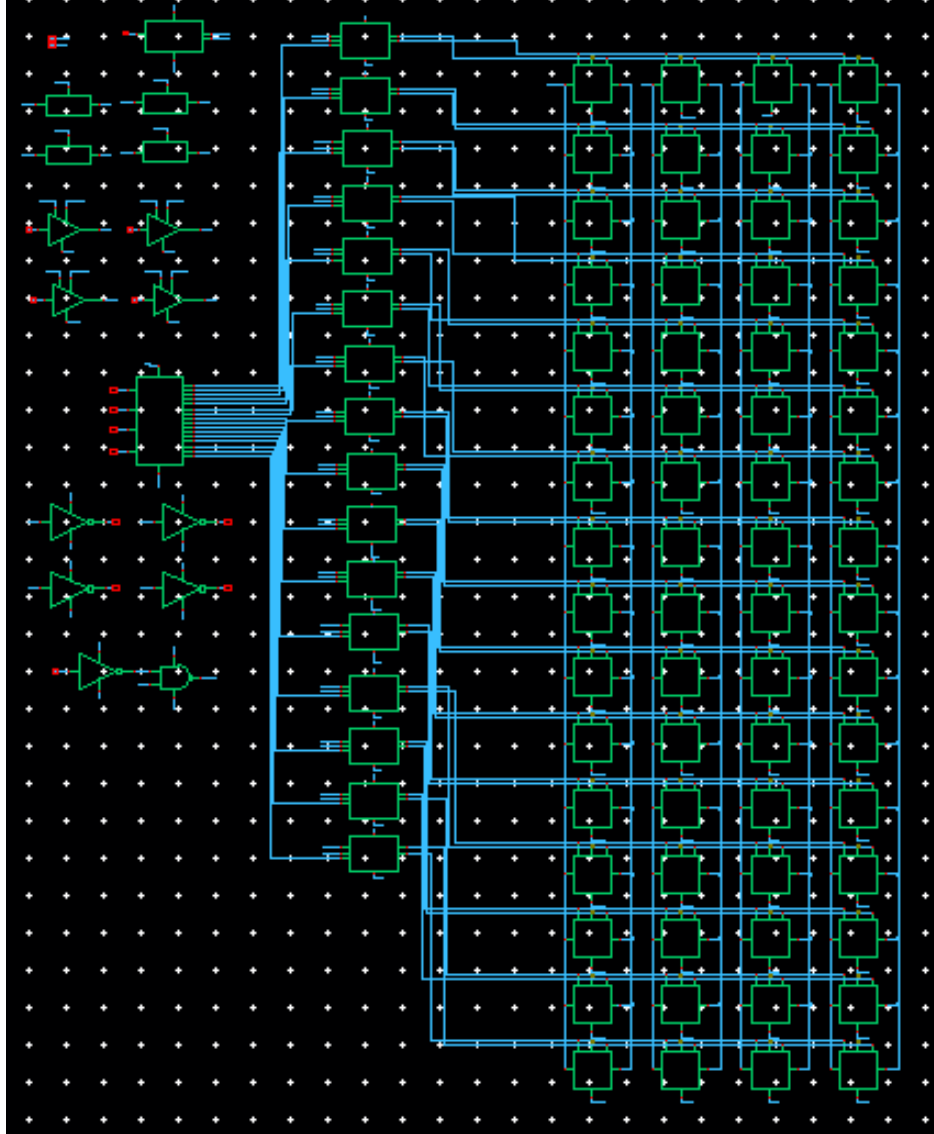


Figure 24: Transistor-level schematic of the 16×4 DRAM macro. The left side contains the clock and control logic that processes `clk_in` and `WE`; the central region implements the 4-to-16 row decoder and column write/sense circuitry; and the right side is the 3T1C bitcell array, where each selected row is written or read based on the external address and data inputs.

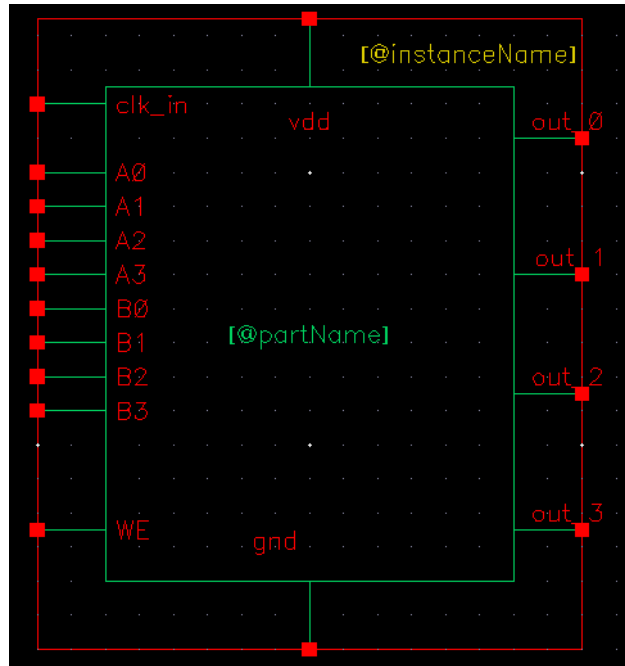


Figure 25: Symbol view of the top-level DRAM block, showing `clk_in`, address inputs `A0`–`A3`, data inputs `B0`–`B3`, write-enable `WE`, data outputs `out_0`–`out_3`, and the global `vdd/gnd` connections.

3 Design Functionality

3.1 Memory Design Validation Description

The memory was validated hierarchically, starting from small peripheral blocks and building up to the full 16×4 array. At the block level, we exercised the row decoder, tristate buffer, clock generator, and 3T1C bitcell with focused transient testbenches that directly target their intended functionality and timing. Each testbench applies representative voltage stimuli, sweeps through all relevant input combinations, and records the resulting digital outputs and internal analog nodes to confirm correct logic behavior, rail-to-rail swings, and the absence of unintended glitches.

After block-level signoff, these same components were instantiated in the top-level DRAM schematic and driven with a unified verification stimulus. The top-level test sequences write and read known data patterns across multiple address windows while the internal waveforms are monitored to ensure proper interaction between the decoder, wordline drivers, precharge circuitry, bitlines, and sense amplifiers. Matching behavior across all address ranges, together with consistency between block-level and array-level results, provides strong evidence that the implemented 3T1C DRAM meets its intended functional specification.

3.2 Functional Verification of Row Decoder

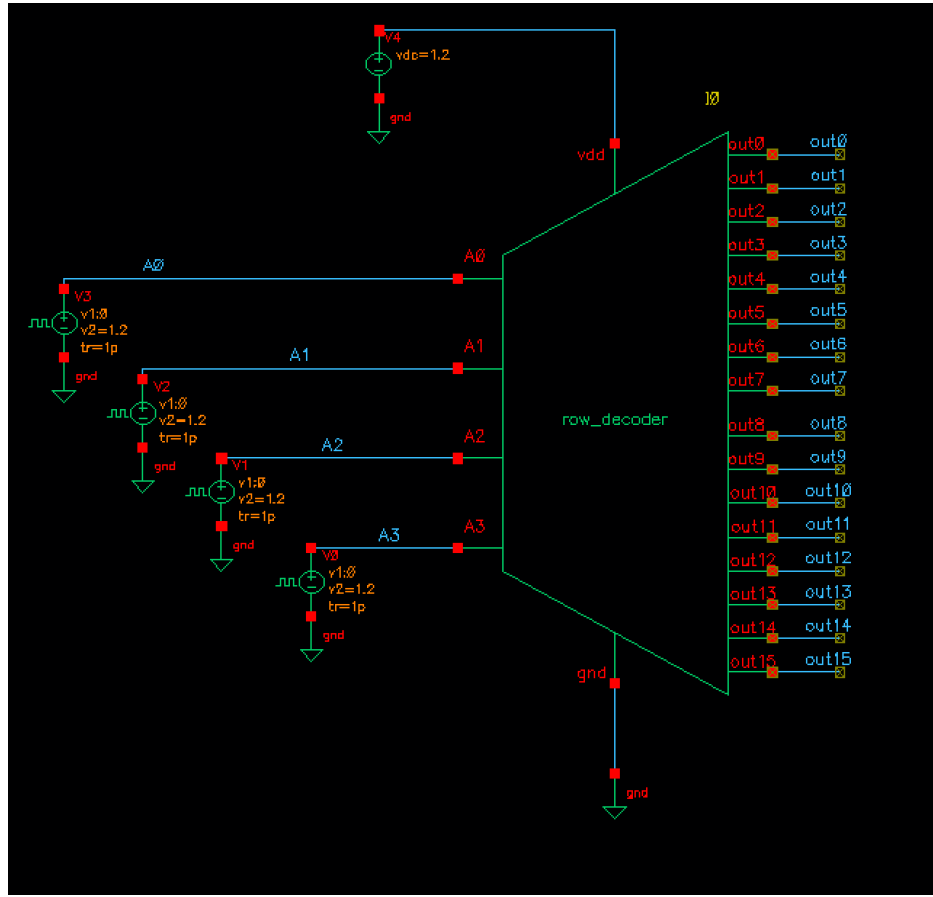


Figure 26:

Figure 2 shows the row decoder testbench. Four independent pulse voltage sources drive the address inputs A0 through A3, with each source configured at different periods to generate all 16 possible address combinations. The decoder receives 1.2V power and ground references, while all 16 output wordlines (out0 through out15) are monitored simultaneously.

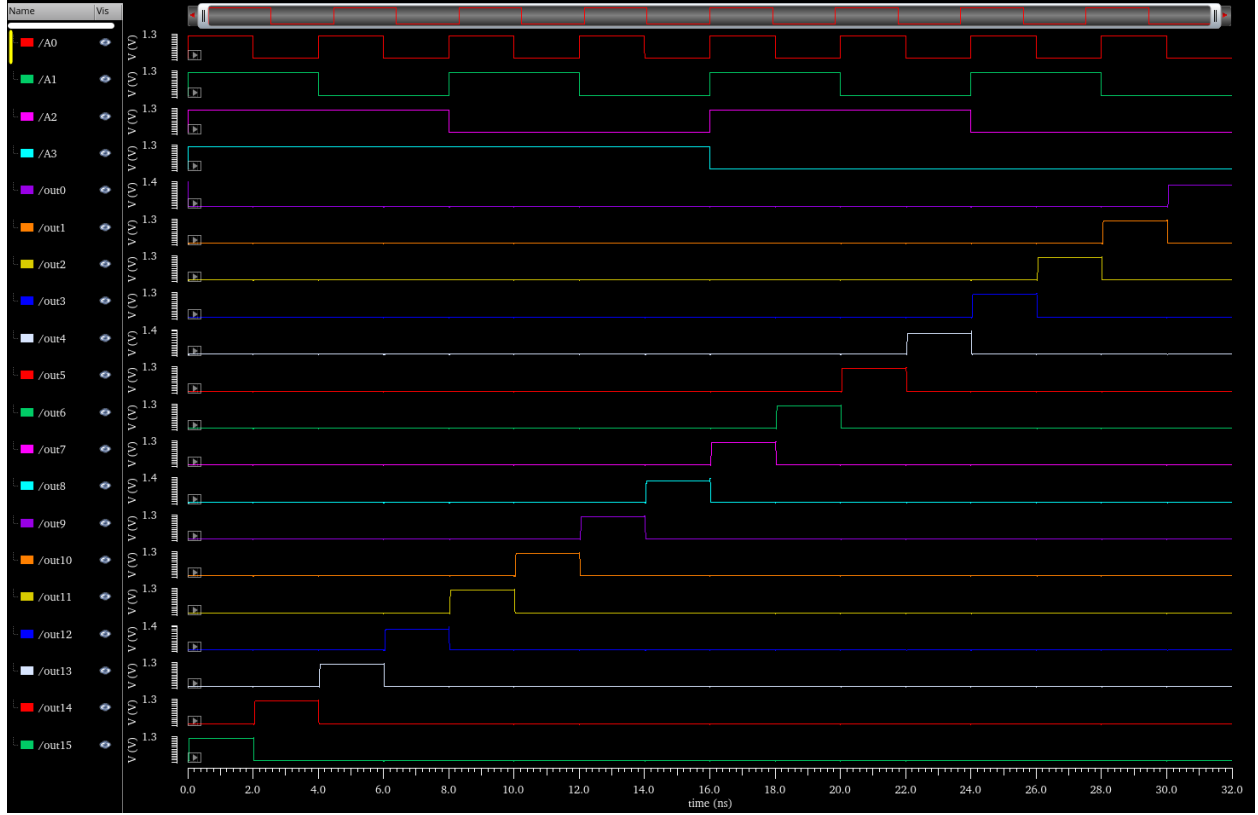


Figure 27:

The test sweeps through all binary address values from 0000 to 1111. The decoder must assert exactly one wordline high for each unique address input. Address 0000 should activate out0, address 0001 should activate out1, continuing sequentially until address 1111 activates out15. Figure 3 confirms correct decoder operation across all 16 address states. The waveforms show sequential activation of wordlines from out15 down to out0 as the address inputs cycle through their combinations. At any given time, only the wordline corresponding to the current address value drives high to 1.2V, while all other outputs remain low. The clean transitions between adjacent wordline assertions verify proper decoding logic without glitches or multiple simultaneous activations.

3.3 Functional Verification of Tristate Buffer

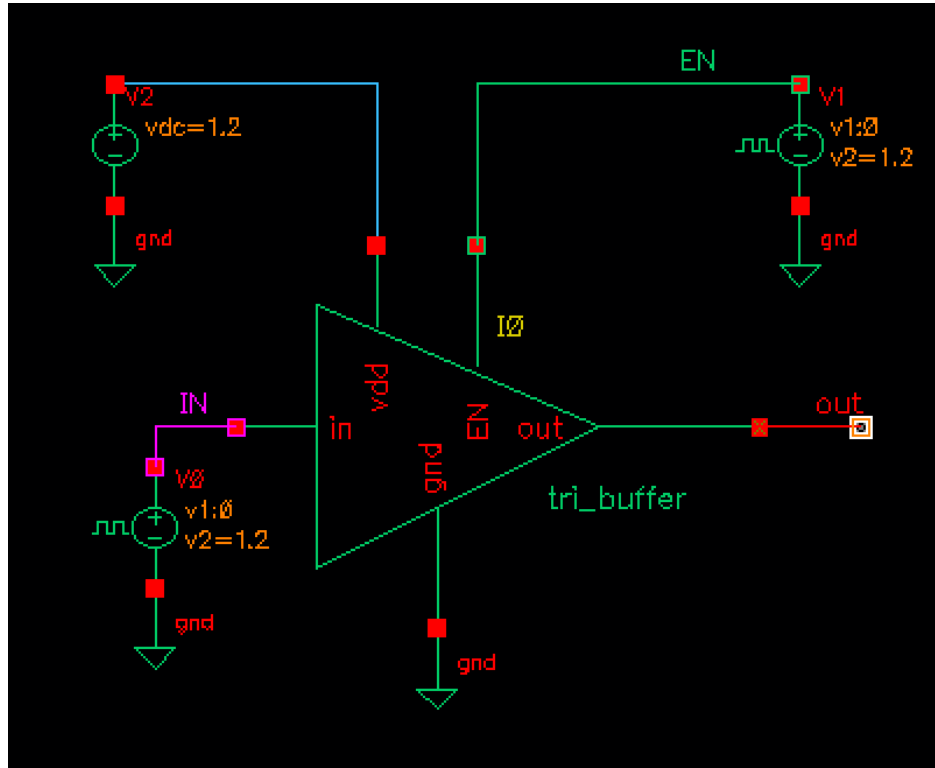


Figure 28:

Figure 4 shows the testbench schematic. The tristate buffer connects to three voltage sources: V0 drives the input (IN), a pulse source V1 controls the enable (EN), and V2 provides 1.2V supply. The output (out) is monitored directly without additional loading.

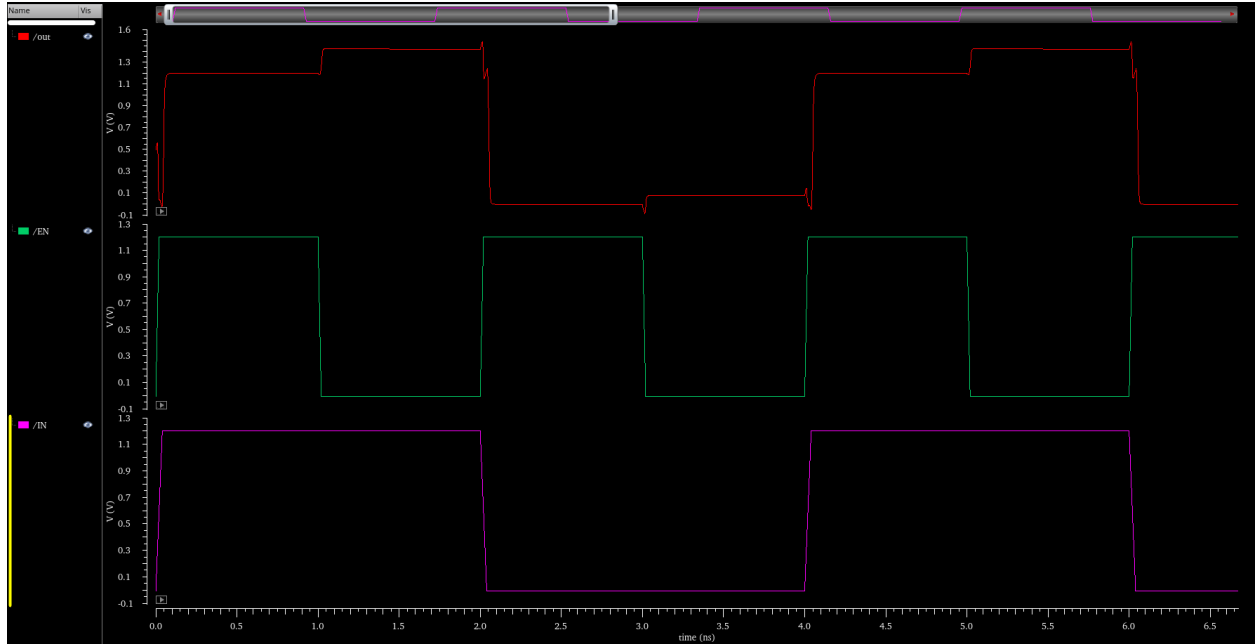


Figure 29:

Figure 5 demonstrates correct tristate behavior across all operating modes. When EN asserts high, OUT tracks IN with minimal propagation delay. When EN deasserts to low, OUT transitions to high-impedance and retains its last driven value with gradual voltage decay. The voltage drift observed during high-Z periods results from floating node capacitance and represents expected behavior rather than circuit malfunction. The offset timing between control signals validates that the buffer responds appropriately to all input combinations.

3.4 Functional Verification of Clock Generator

The `clk_gen` block generates the two non-overlapping clock phases used throughout the DRAM: a main phase `clk_a` and a delayed complement `clk_b`. These phases gate precharge, evaluation, and write operations, so it is essential that (1) both outputs are rail-to-rail and glitch-free, (2) they are logically complementary for most of the cycle, and (3) there is a deliberate “dead time” around each transition where both phases are low (non-overlap), so that precharge and evaluation devices are never on simultaneously.

Figure 30 shows the transient simulation of `clk_gen` over several clock periods. The green trace is `clk_a` and the red trace is `clk_b`. Both outputs swing between approximately 0 V and 1.2 V with sharp edges and no visible static offset. For most of each period, one phase is high while the other is low, so the pair behaves as `clk` and `clk_bar`. Around every transition, however, there is a short interval where both traces are low, corresponding to the intentional non-overlap inserted by the internal gating network.

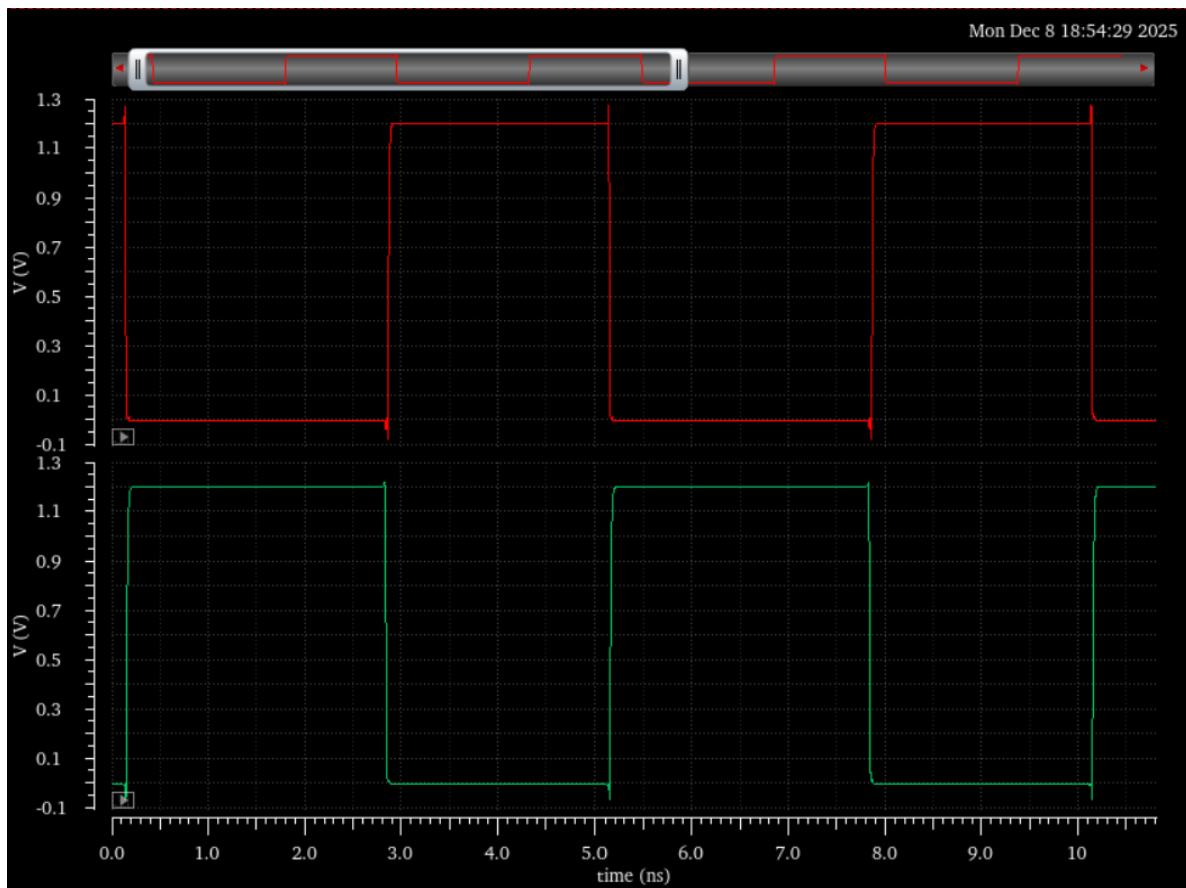
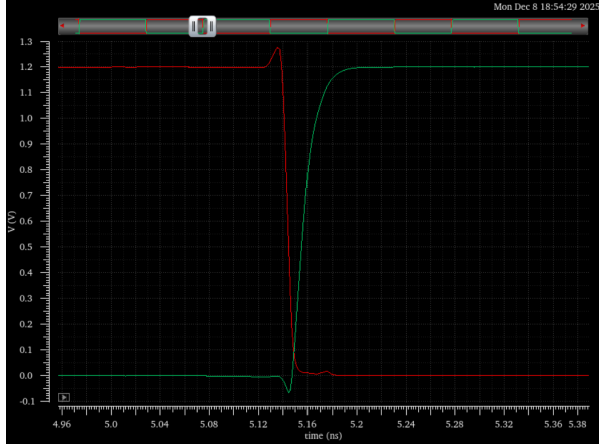


Figure 30: Overview transient simulation of `clk_gen`. Green: `clk_a`; red: `clk_b`. The phases are complementary for most of the cycle, with a short low–low interval around each edge that enforces non-overlap.

To confirm that non-overlap occurs on both transitions, we zoom in around a representative rising and falling edge of `clk_a`. These are shown side by side in Figure 31. On the rising edge (left), `clk_b` (red) first falls from 1.2 V to 0 V while `clk_a` (green) remains low; only after `clk_b` has reached the low rail does `clk_a` begin to rise. On the falling edge (right), `clk_a` goes low before

`clk_b` rises. In both cases there is a clearly defined window where both signals are low and no interval where both are simultaneously high, verifying symmetric non-overlap on rising and falling transitions.



(a) Rising edge of `clk_a`.



(b) Falling edge of `clk_a`.

Figure 31: Zoomed views of `clk_a` (green) and `clk_b` (red) around a rising edge (a) and a falling edge (b). In both cases the currently high phase returns low before the other phase rises, creating a non-overlapping low-low interval and guaranteeing that the two clock phases are never high at the same time.

These simulations demonstrate that the implemented `clk_gen` produces clean, rail-to-rail, non-overlapping clock phases that behave as `clk` and `clk_bar` with a controlled dead time at each transition, satisfying the functional requirements of the DRAM timing scheme.

3.5 Functional Verification of Single Bitcell

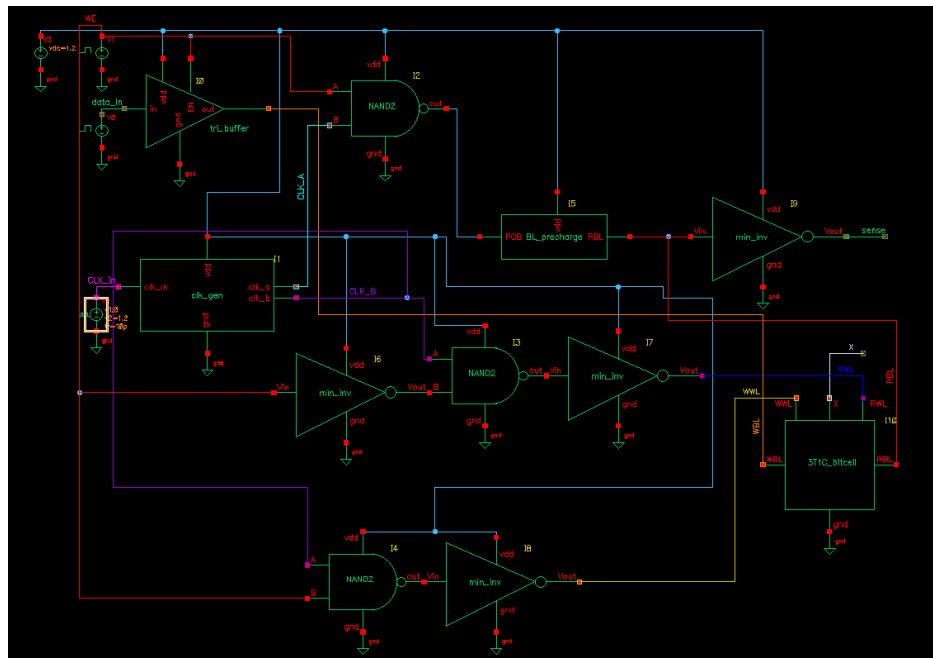


Figure 32:

Figure 9 shows the single bitcell testbench schematic. The 3T-1C DRAM cell interfaces with write drivers, precharge circuitry, and a sense amplifier. Three-phase clocking (CLKin, CLKA, CLKB) coordinates precharge, access, and amplification phases. Control inputs include write enable (WE), along with separate wordlines for write (WBL) and read (RWL) operations. The bitline pair (BL, BLB) connects to the sense amplifier, with datain supplying write values and sense providing read outputs.

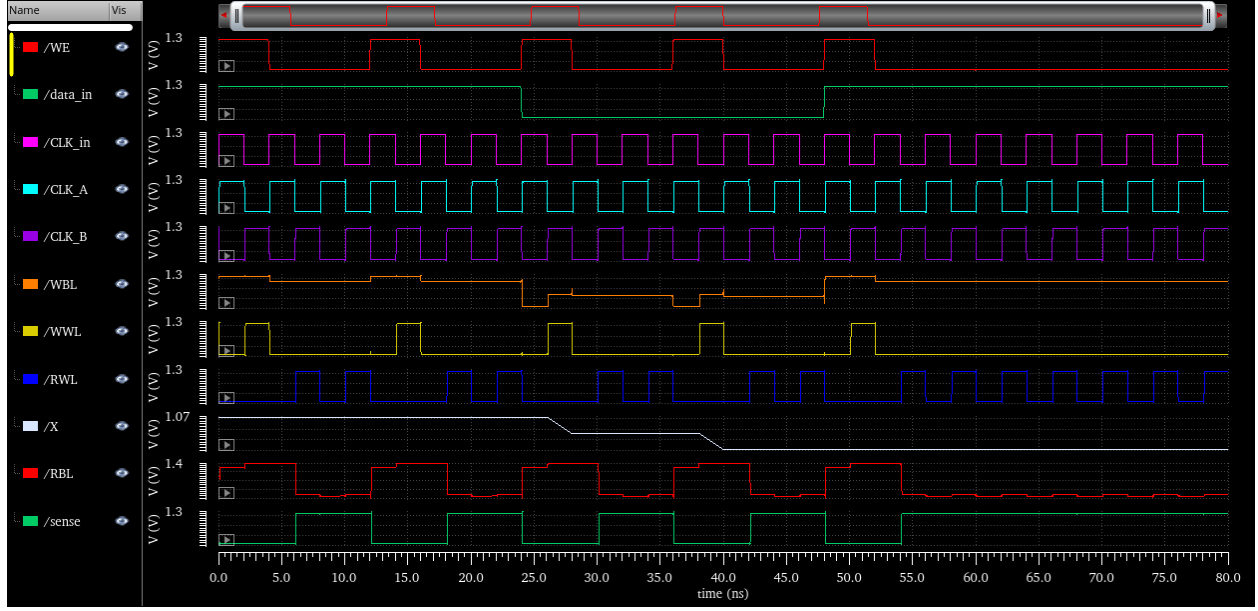


Figure 33:

The test verifies charge storage, retrieval, and retention through repeated access cycles. Five operational groups execute in sequence: two iterations of write-high with dual reads, two iterations of write-low with dual reads, and a final write-high with dual reads. Control signals apply $WE = 100100100100100$ (write on high, read on low) and $data_{in} = 111111000000111$ (high for first two writes, low for next two, high for final write). Figure 10 presents the simulation waveforms. During write-high phases, $data_{in}$ asserts 1.2V, WWL pulses to enable storage, and the capacitor charges. Both subsequent reads activate RWL while keeping WWL inactive, with sense outputting 1.2V on both accesses. This demonstrates charge persistence through multiple readouts without data loss. The second write-high and dual-read group produces identical behavior. Transitioning to write-low operations, $data_{in}$ drops to 0V and the capacitor discharges. The following paired reads both generate $sense = 0V$, confirming the cell retains low state across accesses. The final write-high and dual-read sequence restores sense to 1.2V for both reads. The storage node voltage X tracks write operations and holds state between accesses. The bitlines BL and BLB show voltage separation during reads that the sense amplifier resolves to digital levels. The independent control of WWL and RWL enables proper write/read sequencing without interference between operations.

3.6 Functional Verification of DRAM

We verify the complete 16×4 DRAM array by testing four distinct address ranges: 0000-0011, 0100-0111, 1000-1011, and 1100-1111. This partitioning strategy provides comprehensive coverage of the decoder logic while maintaining manageable simulation complexity. Each address range exercises a unique combination of the two most significant address bits (A3 and A2), ensuring that all four major decoder branches activate correctly. Within each range, the lower address bits (A1 and A0) increment through their binary values, with A0 toggling every four cycles and A1 toggling every eight cycles, thereby accessing all four addresses in sequence. The write enable signal alternates between 1 and 0 to create write-read pairs, while the data inputs apply a repeating 1100 pattern across all four bits (B0-B3), which writes the value 1111 followed by 0000 to each address. This data pattern tests both charge storage (logic 1) and discharge (logic 0) operations for every bitcell. We expect all four test cases to produce visually identical output waveforms because each case executes the same operational sequence—write 1111, read 1111, write 0000, read 0000—repeated across four consecutive addresses. The only difference between cases lies in which physical rows are selected by the decoder, but since all bitcells share the same transistor-capacitor structure and all periphery circuits (sense amplifiers, write drivers, precharge circuits) are designed identically, the electrical behavior should be uniform across the entire array. If the outputs (out0-out3) show the same temporal pattern of alternating 1111 and 0000 values for all address ranges, this confirms that the row decoder correctly activates each wordline, all 64 bitcells store and retrieve data reliably, and the column circuitry operates consistently regardless of which row is accessed. This verification approach efficiently validates array uniformity without requiring exhaustive testing of all 16 addresses individually, while the strategic selection of address ranges ensures complete decoder coverage and provides high confidence in full-array functionality.

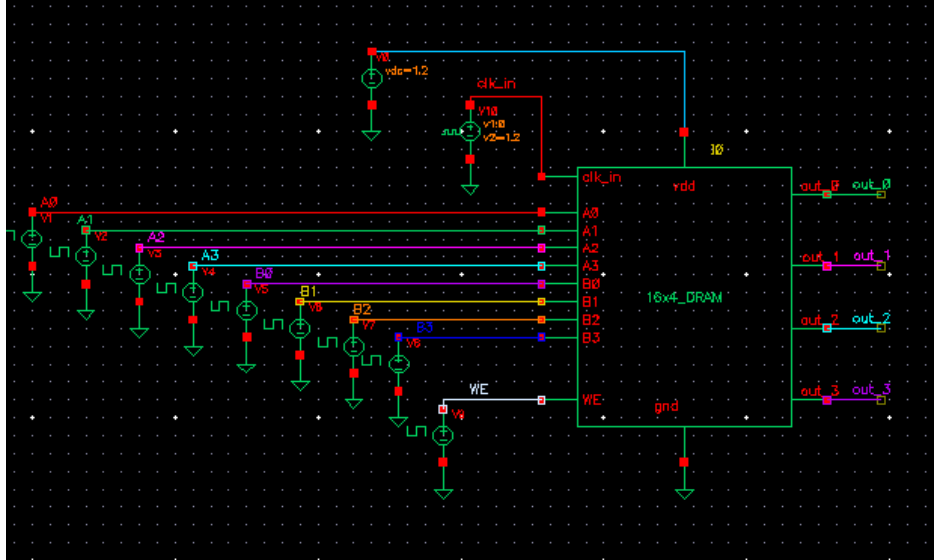


Figure 34: testbench used for all verification of DRAM

3.6.1 Case 1: 0000 \rightarrow 0011

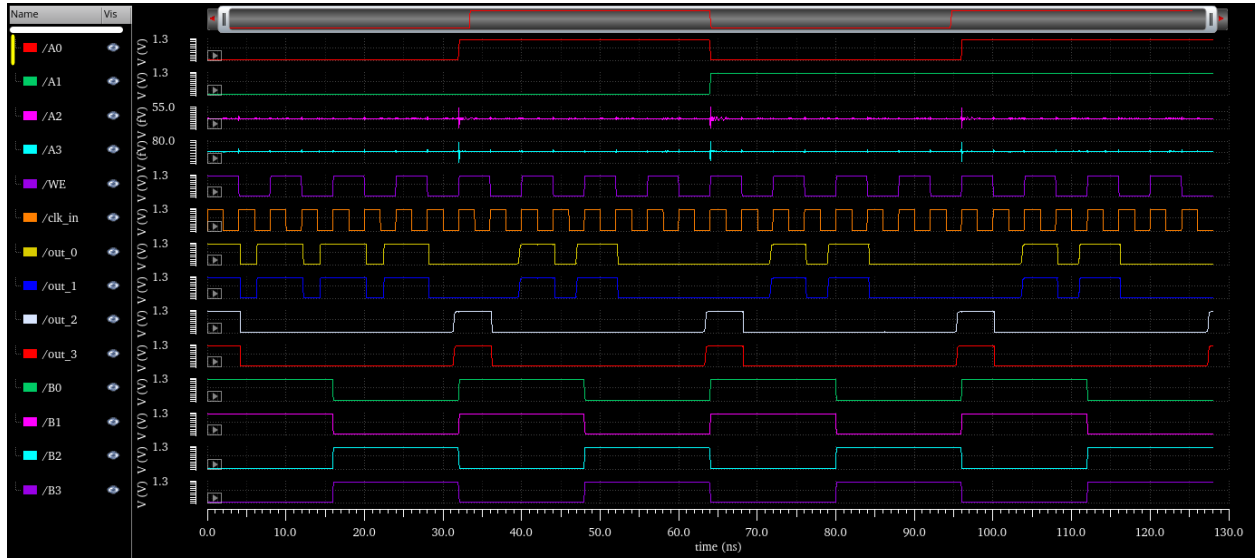


Figure 35: Functional verification waveforms for Case 1. The address counter sweeps rows 0000–0011 while WE alternates to create write-read pairs with a 1111/0000 data pattern. The column outputs out0–out3 follow the expected sequence of 1111 then 0000 at each addressed row, confirming correct operation of the first decoder branch.

3.6.2 Case 2: 0100 \rightarrow 0011

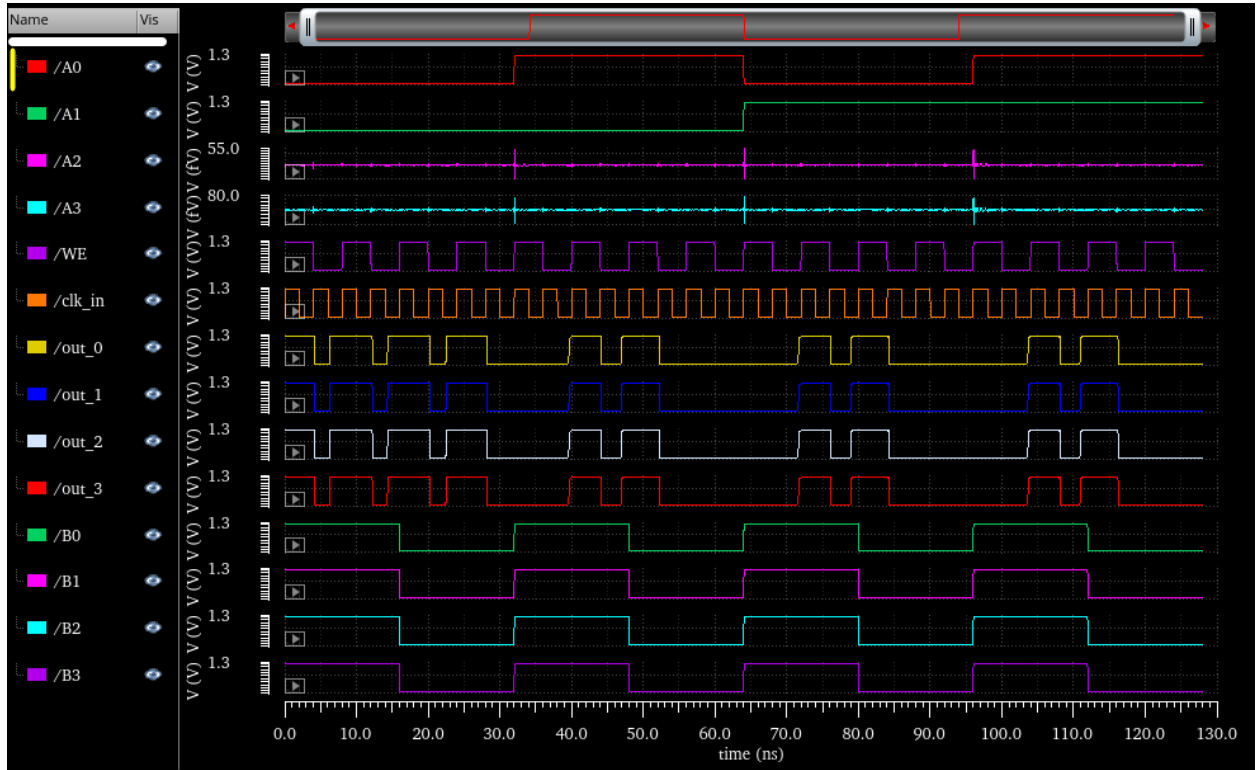


Figure 36: Functional verification waveforms for Case 2. The same write/read stimulus used in Case 1 is applied to a different four-row address window. The decoded wordline and outputs out0–out3 exhibit the same 1111/0000 pattern, demonstrating uniform behavior across a second decoder branch.

3.6.3 Case 3: 1100 \rightarrow 1111

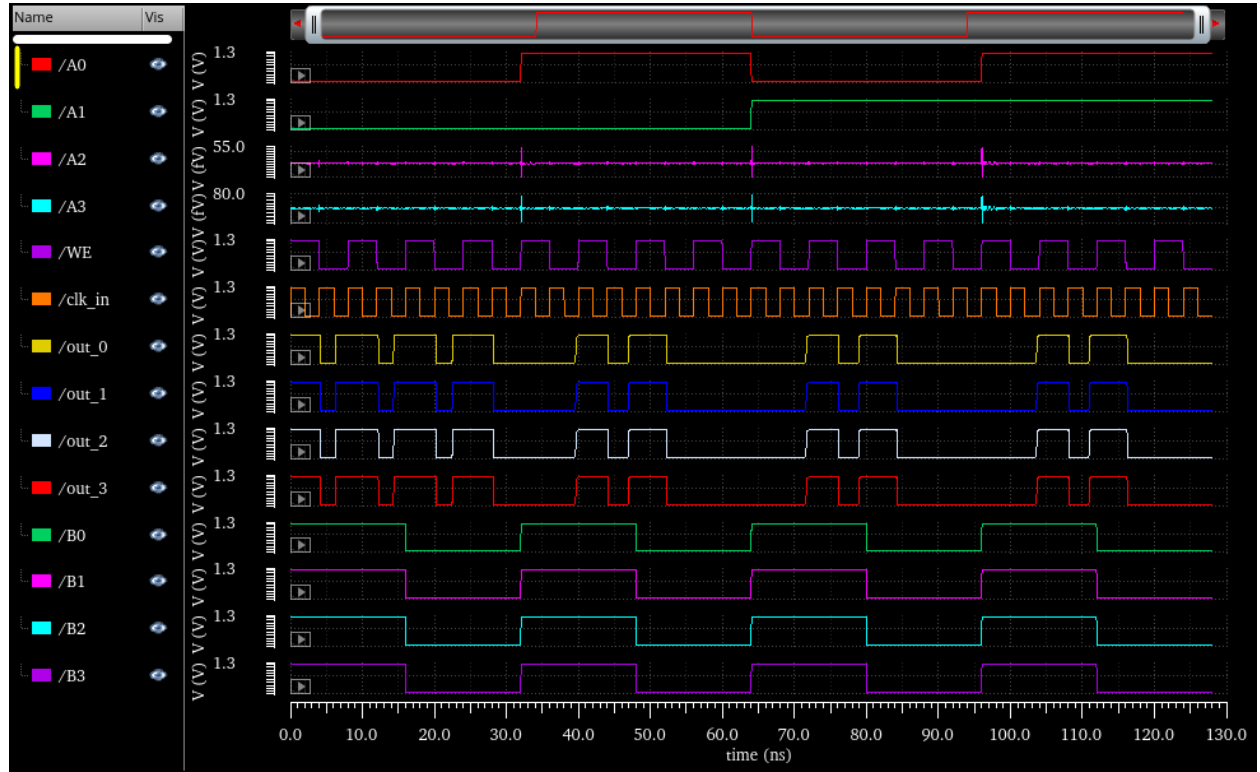


Figure 37: Functional verification waveforms for Case 3, targeting the highest-address rows of the array. Despite the longer wordline routes, the outputs out0–out3 again alternate cleanly between 1111 and 0000 after each write–read pair, confirming correct decoding and sensing at the far end of the array.

3.6.4 Case 4: 0100 \rightarrow 0011

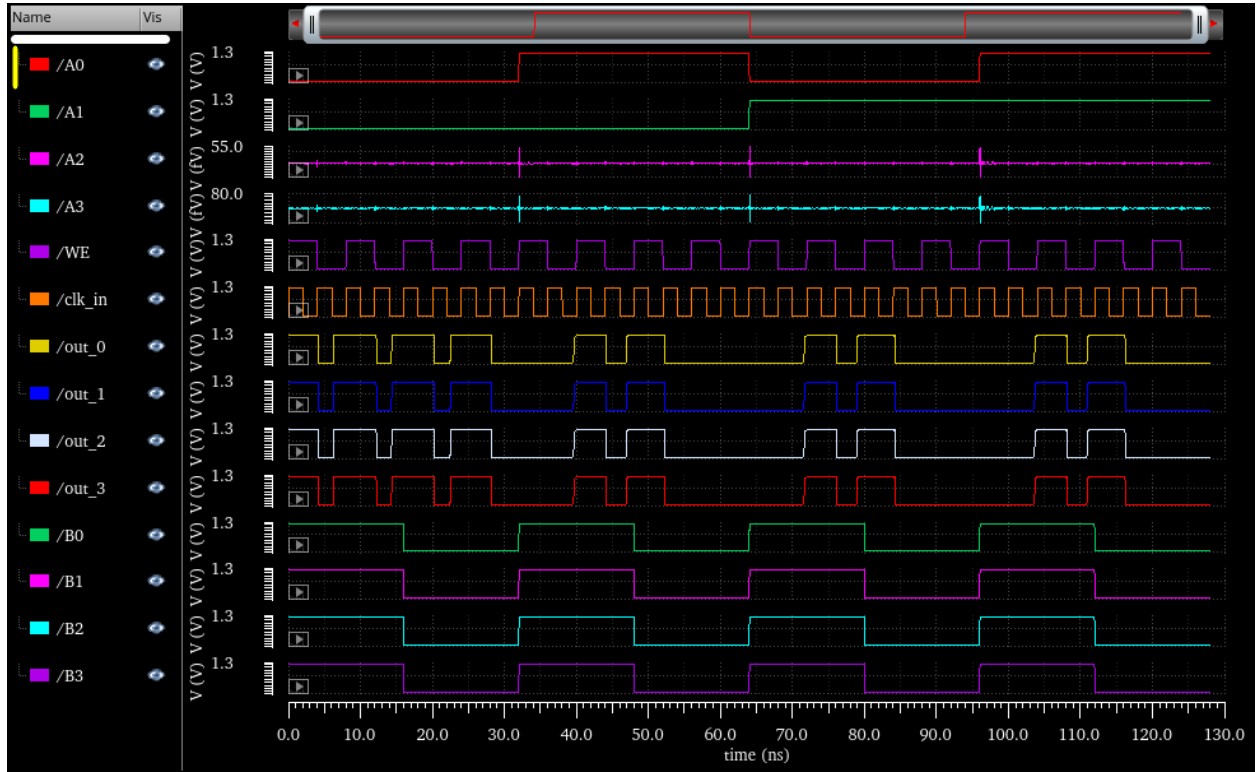


Figure 38: Functional verification waveforms for Case 4, which reorders the four-row access window to exercise a different pattern of wordline activations. The column outputs out0–out3 still reproduce the expected 1111/0000 sequence, indicating that the DRAM operates correctly regardless of the specific row ordering within the test.

4 Design Metrics with Support

4.1 Delay

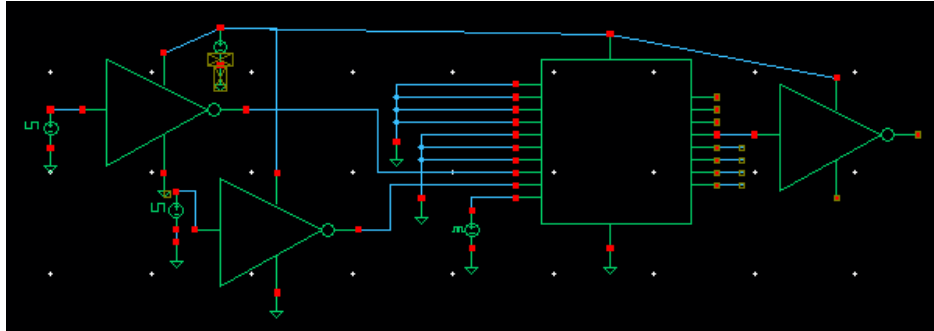


Figure 39: Testbench utilized, modified for write and read (VDD connections appear strangely, but the testbench functioned as intended).

4.1.1 Write Delay

Write High

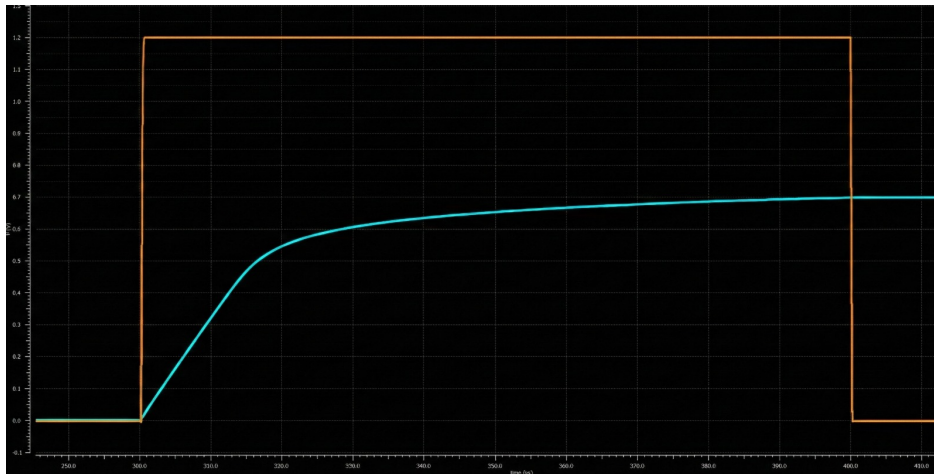


Figure 40: Expression to calculate delay: `(cross(v("/X" ?result "tran") 0.662 1 "rising" nil nil nil) - cross(v("/WWL" ?result "tran") 0.6 2 "rising" nil nil nil))`

Delay extracted as 68.377 ns.

Write Low

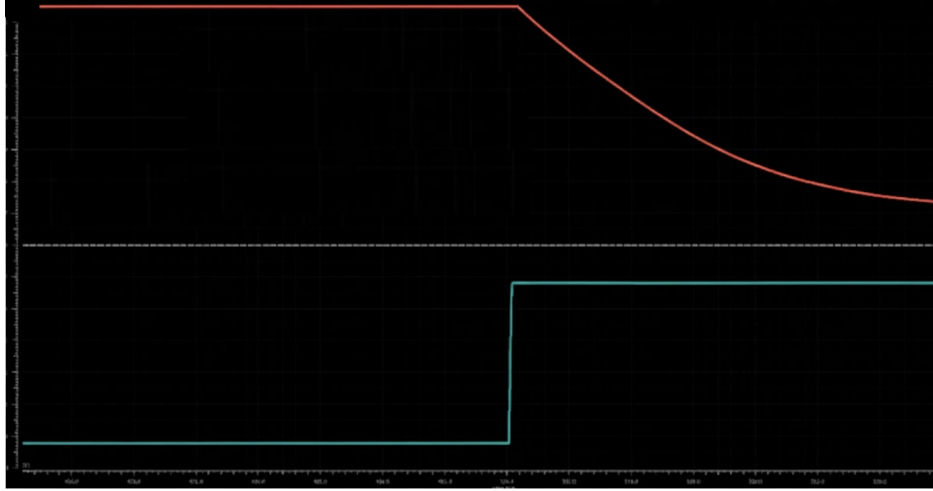


Figure 41: Expression to calculate delay: `(cross(v("/X" ?result "tran") 0.573 2 "falling" nil nil nil) - cross(v("/WWL" ?result "tran") 0.6 3 "rising" nil nil nil))`

Delay extracted as 6.1278 ns.

4.1.2 Read Delay

Read High



Figure 42: Expression to calculate delay: `(cross(v("/X" ?result "tran") 0.6 1 "rising" nil nil nil) - cross(v("/RWL" ?result "tran") 0.6 3 "rising" nil nil nil))`

Delay extracted as 0.2100 ns.

Read Low



Figure 43: Expression to calculate delay: $(\text{cross}(v("/X" \text{ ?result "tran"})) \text{ 0.6 2 "falling" nil nil nil}) - \text{cross}(v("/RWL" \text{ ?result "tran"})) \text{ 0.6 3 "rising" nil nil nil})$

Delay extracted as 57 ps.

Conclusion

The delay metric is determined by the worst-case access time among all read and write operations. Four distinct delay measurements were performed to characterize the memory's timing behavior: write-high delay (charging the storage capacitor to logic 1), write-low delay (discharging to logic 0), read-high delay (sensing a stored 1), and read-low delay (sensing a stored 0).

The write-high operation requires the storage capacitor to charge from 0V toward $V_{DD} = 1.2\text{V}$ through the minimum-sized NMOS write access transistor. Since NMOS devices conduct weakly when pulling up to V_{DD} , this represents the longest charging path in the bitcell. The delay was measured from the rising edge of the write wordline (WWL) at 0.6V to the point where the storage node X crosses the high sensing threshold of 0.662V. The extracted value is 68.377 ns, which dominates all other timing paths.

In contrast, write-low delay measures only 6.128 ns because the NMOS access transistor provides strong pull-down current when discharging the capacitor to ground. Read operations are even faster: read-high delay is 0.210 ns and read-low delay is 57 ps, since these operations involve only charge sharing between the storage capacitor and the precharged bitline, followed by inverter-based sensing, without requiring full capacitor charge/discharge.

Since the memory must guarantee correct operation across all access patterns, the worst-case delay of 68.377 ns (write-high) sets the minimum clock period and defines the delay component of the FOM. This conservative metric ensures that even the slowest operation completes successfully within one clock cycle.

4.2 Power

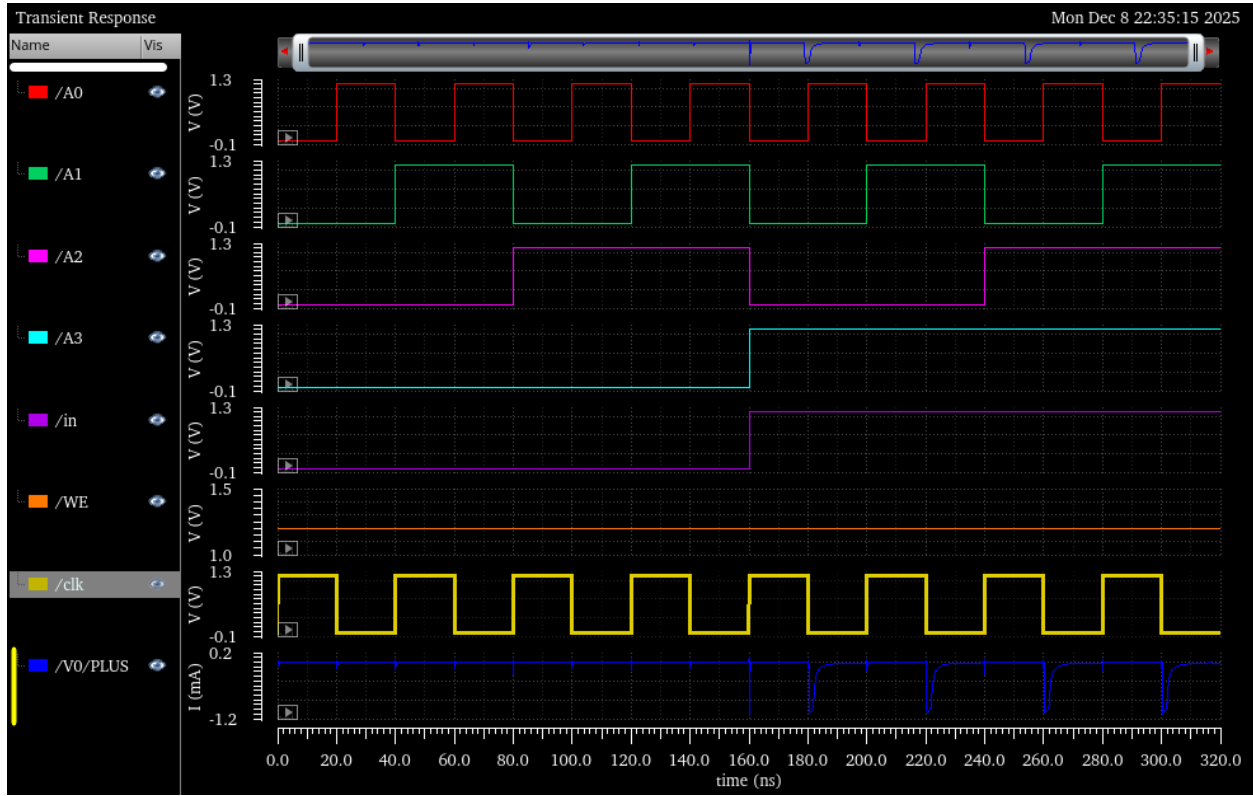


Figure 44:

Expression	Value
<code>integ(abs(i("/V0/PLUS" ?result "tran" 0n 320n)))</code>	11.66E-12

Figure 45: Calculated using `integ(abs(i("/V0/PLUS" ?result "tran" 0n 320n)))`*1.2

Power consumption was measured by simulating a representative workload that exercises both write-1 and write-0 operations across the entire memory array. The test sequence writes all 64 bitcells to logic 0, then writes all 64 bitcells to logic 1, and repeats this pattern four times for a total of eight full-array write sweeps. This pattern captures the dynamic energy dissipated in charging and discharging the storage capacitors, bitline parasitic capacitances, wordline drivers, decoder logic, and sense amplifiers.

The total energy was computed by integrating the absolute value of the supply current over the 320 ns test interval and multiplying by the supply voltage (1.2V). The Cadence calculator expression `integ(abs(i("/V0/PLUS" ?result "tran" 0n 320n)))`*1.2 evaluates to 11.66 * 10 pJ. Dividing by the test duration yields an average power consumption:

$$P_{\text{avg}} = \frac{11.66 \times 10^{-12} \text{ J}}{320 \times 10^{-9} \text{ s}} = 3.644 \times 10^{-5} \text{ W} = 36.44 \text{ } \mu\text{W} \quad (1)$$

This power measurement reflects the energy cost of the worst-case dynamic activity: continuously writing alternating data patterns at maximum frequency. The clock period used for this measurement matches the worst-case delay (68.377 ns) to ensure the memory operates at its maximum sustainable frequency. Minimum transistor sizing throughout the design keeps switching capacitances small, resulting in low dynamic power despite the relatively large storage capacitance (1 pF per bitcell).

4.3 Area

The area metric accounts for all transistors and capacitors within the 64 bitcells of the 16×4 array. Peripheral circuitry (decoders, sense amplifiers, write drivers, precharge devices, and control logic) is excluded per the project specification, which focuses area optimization on the storage array itself.

Each 3T1C bitcell contains three minimum-sized NMOS transistors with $W = 120$ nm and $L = 45$ nm, plus one 1 pF storage capacitor. The transistor area per bitcell is:

$$A_{\text{transistors}} = 3 \times (120 \text{ nm} \times 45 \text{ nm}) = 16,200 \text{ nm}^2 \quad (2)$$

The capacitor area is estimated assuming a parallel-plate structure with a high-k dielectric material (relative permittivity $\epsilon_r \approx 25$) and dielectric thickness $d = 5$ nm, which are representative values for modern DRAM capacitors in logic-compatible processes. Using the parallel-plate capacitance formula:

$$C = \epsilon_0 \epsilon_r \frac{A}{d} \quad \Rightarrow \quad A_{\text{cap}} = \frac{C \cdot d}{\epsilon_0 \epsilon_r} \quad (3)$$

Substituting $C = 1$ pF, $d = 5$ nm, $\epsilon_0 = 8.854 \times 10^{-12}$ F/m, and $\epsilon_r = 25$:

$$A_{\text{cap}} = \frac{(1 \times 10^{-12} \text{ F})(5 \times 10^{-9} \text{ m})}{(8.854 \times 10^{-12} \text{ F/m})(25)} \approx 22,600 \text{ nm}^2 \quad (4)$$

The total area per bitcell is therefore:

$$A_{\text{bitcell}} = 16,200 \text{ nm}^2 + 22,600 \text{ nm}^2 = 38,800 \text{ nm}^2 \quad (5)$$

Multiplying by the 64 bitcells in the array:

$$A_{\text{total}} = 64 \times 38,800 \text{ nm}^2 = 2,483,200 \text{ nm}^2 = 2.483 \times 10^{-6} \text{ mm}^2 \quad (6)$$

This area calculation reflects the minimum-sizing strategy applied throughout the design. The 1 pF storage capacitance was selected to balance retention time (minimizing leakage-induced charge loss) against area overhead, while the minimum transistor dimensions ($W = 120$ nm, $L = 45$ nm) keep both junction capacitance and subthreshold leakage as small as possible. The resulting bitcell area is dominated by the explicit storage capacitor, which accounts for approximately 58% of the total bitcell footprint. Claude is AI and can make mistakes. Please double-check responses.

4.4 Figure of Merit

The figure of merit (FOM) for the 16×4 DRAM array is calculated as the product of area, power, and delay:

$$\text{FOM} = \text{Area} \times \text{Power} \times \text{Delay} \quad (7)$$

From the measured design metrics:

- Delay = 68.377 ns (worst-case write high)
- Energy = 11.66 * 10 pJ (over 320 ns test period)
- Power = $\frac{11.66 \times 10^{-12} \text{ J}}{320 \times 10^{-9} \text{ s}} = 3.644 \times 10^{-5} \text{ W}$
- Area = 2.483 * 10e-6 mm²

$$\text{FOM} = (2.483 \times 10^{-6} \text{ mm}^2) \times (3.644 \times 10^{-5} \text{ W}) \times (68.377 \times 10^{-9} \text{ s}) \quad (8)$$

$$\text{FOM} = 6.19 \times 10^{-18} \text{ mm}^2 \cdot \text{W} \cdot \text{s} \quad (9)$$

This FOM value reflects the design choices made throughout the memory architecture. The minimum-sizing strategy applied to all transistors ($W = 120 \text{ nm}$, $L = 45 \text{ nm}$) simultaneously minimizes area and power by reducing both device footprint and switching capacitance. However, this conservative sizing results in relatively long write delays, particularly for the write-high operation where the NMOS access transistor must charge the 1 pF storage capacitor through a weak pull-up path. The selection of 1 pF storage capacitance represents a critical trade-off: larger capacitance would improve retention time and sensing margin but would increase both area (through the physical capacitor size) and delay (through longer charge/discharge times), while smaller capacitance would reduce area and delay but compromise data retention and noise immunity.

The dominant contributor to the FOM is the write-high delay of 68.377 ns, which is more than three orders of magnitude larger than the read delays and approximately 11 times larger than the write-low delay. This asymmetry arises from the fundamental limitation of NMOS pass transistors when charging toward V_{DD} , and suggests that future optimizations could focus on accelerating the write-high path through techniques such as boosted write wordline voltages, differential write schemes, or hybrid PMOS-NMOS write drivers. The power contribution is modest due to the minimum device sizing and the relatively low operating frequency imposed by the long write delay, while the area is constrained primarily by the explicit 1 pF storage capacitor rather than the access transistors.

In comparison to the baseline SRAM approach (which would use 6T bitcells), this 3T1C DRAM achieves area reduction through the replacement of four cross-coupled inverter transistors with a single explicit capacitor, at the cost of increased complexity in the peripheral circuitry (separate read and write wordlines, precharge circuits, and refresh logic) and slower write access times. The measured FOM provides a quantitative basis for evaluating this architectural trade-off and enables direct comparison with alternative memory implementations targeting the same capacity and technology node.

References

- [1] A. Teman, P. Meinerzhagen, A. Burg, and A. Fish, “Review and classification of gain cell eDRAM implementations,” in *Proc. IEEE 27th Convention of Electrical and Electronics Engineers in Israel (IEEEI)*, Eilat, Israel, Nov. 2012, pp. 1–5.
- [2] R. Giterman, A. Shalom, A. Burg, A. Fish, and A. Teman, “A 1-Mbit fully logic-compatible 3T gain-cell embedded DRAM in 16-nm FinFET,” *IEEE Solid-State Circuits Letters*, vol. 3, pp. 110–113, 2020, doi:10.1109/LSSC.2020.3006496.
- [3] S. Kim and J.-E. Park, “Pseudo-static gain cell of embedded DRAM for processing-in-memory in intelligent IoT sensor nodes,” *Sensors*, vol. 22, no. 11, Art. 4284, Jun. 2022, doi:10.3390/s22114284.
- [4] C. Dai, Y. Lu, W. Lu, Z. Lin, X. Wu, and C. Peng, “Low-power single bitline load sense amplifier for DRAM,” *Electronics*, vol. 12, no. 19, Art. 4024, Sep. 2023, doi:10.3390/electronics12194024.
- [5] C.-H. Lin *et al.*, “High-performance 14 nm SOI FinFET CMOS technology with 0.0174 μm^2 embedded DRAM and 15 levels of Cu metallization,” in *Proc. IEEE Int. Electron Devices Meeting (IEDM)*, San Francisco, CA, USA, Dec. 2014, pp. 3.1.1–3.1.4.
- [6] F. Hamzaoglu *et al.*, “A 1 Gb 2 GHz embedded DRAM in 22 nm tri-gate CMOS technology,” in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, San Francisco, CA, USA, Feb. 2014, pp. 324–325.