Digital VLSI Design

Course Project

October 1, 2025

Instructions

1. Read the document carefully before proceeding to actual work. As the project deals with system-level circuits, make sure you start as early as possible.
2. The project mainly contains 4 sub-blocks. It is highly recommended to divide the work accordingly and collaborate efficiently.
3. Use of symbol blocks in simulation is highly recommended. Once you make a sub-circuit, create the respective symbol so that you can reuse it.
4. For your first cut design, operation and functionality is more important than performance and optimization. Make sure your circuits work initially before going into optimization. Working of the circuit carries higher percentage.
5. **Evaluation Policy:** Project grades will be based on both the functionality of the complete circuit and the correctness of individual sub-blocks. Marks will be awarded strictly according to individual contribution and performance.
   Any form of plagiarism will result in an F grade. If you are unable to explain your own work, design choices, simulation results, or your specific contribution, it will be graded as zero.

Design of a 32 x 32-bit SRAM– Background

Memory arrays are an essential building block of all digital systems. In this semester's project, we will design an SRAM array that contains 32 x 32-bit words. In order to support operation as a FIFO, the memory is accessed using a 5-bit address.

Block level Description

- **Adder:** Accepts two 5-bit address inputs (`addra(4:0)`, `addrb(4:0)`) along with a carry-in (`Cin`). Produces a 5-bit summed address (`a(4:0)`) that is forwarded to the decoder.
- **Decoder:** Translates the 5-bit summed address into one-hot word line signals (`W0-W31`). This ensures that exactly one row of the SRAM array is activated at a time.
- **SRAM Array (32 x 32):** A memory block with 32 words, each 32 bits wide. The active word line connects the selected row to the differential bit line pairs (`b0-b31`, `b'0-b'31`).
  Note: In this project, the memory cells are pre-loaded with data using the initial condition feature in Cadence. Write operation is not considered within the scope of this project.
- **Sense Amplifier:** Detects the small voltage difference on each bit line pair and amplifies it to produce full-swing digital outputs (`out0-out31`).

Overall Operation

Two input addresses are first added in the **Adder**. The resulting address is decoded by the **Decoder** to activate a specific word line in the **SRAM Array**. The corresponding 32-bit data word is placed on the bit lines, and the **Sense Amplifier** converts these signals into stable logic outputs.

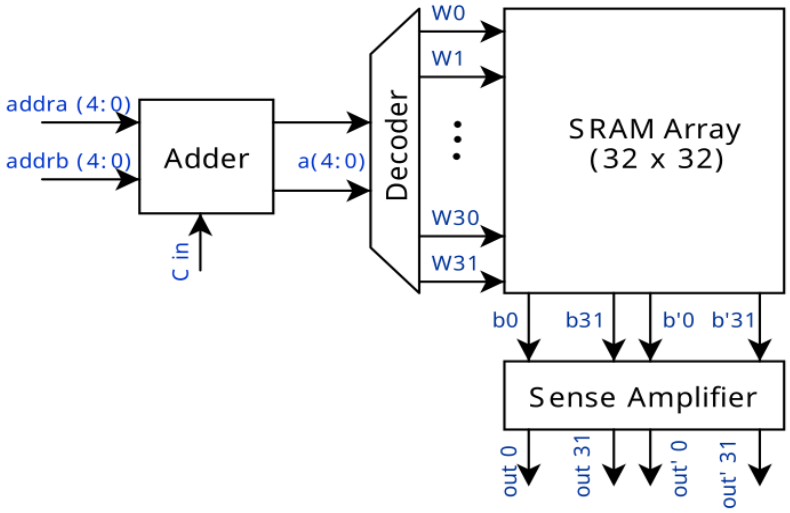

Figure 1: SRAM block diagram

Stage 1: SRAM Cell Characterization

To design and characterize the basic 6T SRAM cell as shown in figure 2. The concept and working of SRAM is given here.

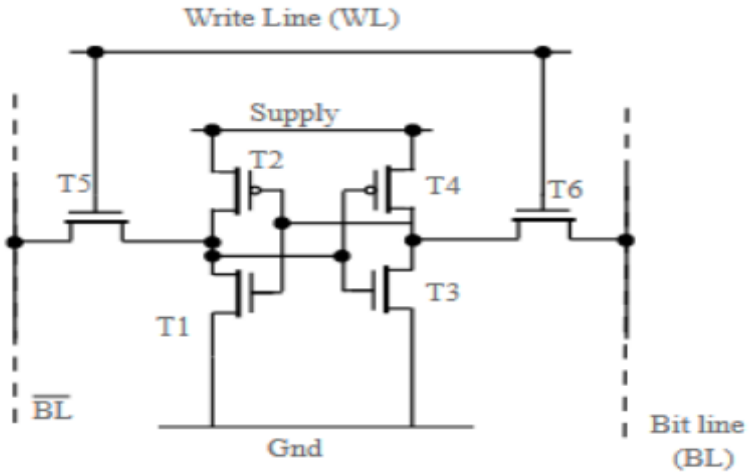

Figure 2: SRAM 6T cell

Recalling that the wordline and bitlines are held at VDD during a read, Figure 2 shows how to extract the read static noise margin (SNM) of the cell. First, the feedback from the cross coupled inverters is broken. Next, the VTC of the "inverter" formed by half of the SRAM cell is found by sweeping V1 (the inverter's input) from 0 to VDD and measuring V2 (the inverter's output). This plot is then used to construct the "butterfly plot" that is representative of the two halves of the cell driving each other. The read SNM is the side length of the maximum possible square that can fit inside of the butterfly plot. You do not have to calculate the size of this maximum square, but you should submit the butterfly

plot (generated using SPICE tool) that graphically indicates the SNM. You should also measure the worst-case voltage rise in the SRAM cell during a read (i.e., the value of V2 when V1 is at VDD) and provide that value in your report. During a write, VDD is applied to the worldine, and the value to be written into the memory cell is driven onto the bitlines.

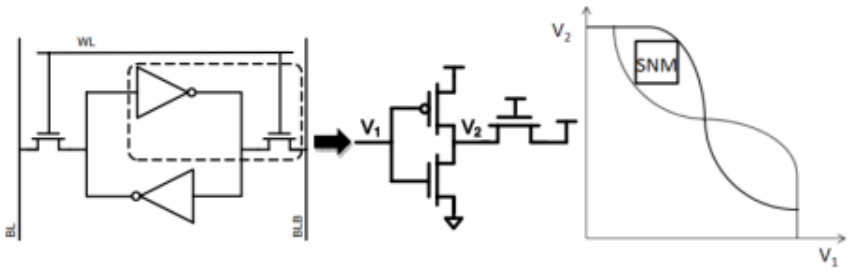

Figure 3: SRAM read static noise margin

Thus, Figure 3 shows how to extract the writenoise margin (WNM) of the cell. Again, the feedback from the cross coupled inverters is broken, and the VTC of the "inverters" are measured. Note however that in this case, the VTCs of the two halves of the SRAM are no longer the same (since one of the bitlines is driven to 0V, and the other to VDD). These VTCs are used to create a butterfly plot, and the WNM is the side length of the largest square that can fit inside of the butterfly plot. You do not have to calculate the WNM, but you should generate the butterfly plot (again using SPICE) and graphically indicate the WNM. You should however measure the worst-case cell voltage during a write (which is found from measuring V1 when V2 is at 0V).

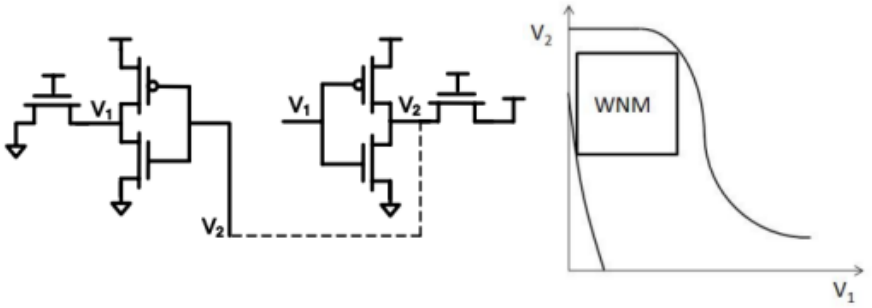

Figure 4: SRAM Write noise margin

Decoder design

The next stage is to design the 5-to-32 decoder. The goal will be to minimize the delay of the decoder from the address inputs transitioning to the wordline rising. The enable signal EN is active high and enables the decoder outputs – this is required in order to be able to precharge the bitlines once a read or write has been completed.
Note that before you enter the decoder into Cadence, you should manually design the decoder (on paper) for the minimum delay using the method of logical effort.

Adder and Sense Amplifier Design

Adder design

A adder has to designed which accepts two 5-bit address inputs (`addra(4:0)`, `addrb(4:0)`) along with a carry-in (`Cin`).

Sense Amplifier Design

A sense amplifier in SRAM (Static Random Access Memory) is a specialized circuit used to read data stored in memory cells. It plays a critical role in ensuring reliable and efficient data retrieval during a read operation.It Detects the small voltage difference on each bit line pair and amplifies it to produce full-swing digital outputs (`out0-out31`).(Hint: A Differential Amplifier can be used as SENSE Amplifier.)

Analysis and Simulation

Your primary goal in any IC design should be to ensure that the circuit you have designed functions as intended. Make sure you create a proper testbench for every block designed, making it easier for you to design and verify. As you are assembling your adder, you should keep in mind which input pattern will result in the worst-case delay, and thus you should optimize your design to minimize the delay in this case.

1. Functionality:In order to ensure that your design functions properly, you will need to simulate the entire operation of your SRAM (including your optimizations) from the adder to the output of the sense amplifier.
   Functionality has to be checked by simulating two back-to-back read operations, and ensuring that each of them has the correct output. The two read operations will need to access two different wordline addresses, with the data stored in the cells initialized so that the output data flips polarities between the two cycles.

2. Performance: The performance of your design will be characterized by its total latency – i.e., the delay from the adder transitioning to the data appearing at the output of the sense amplifier.Critical path delay of the individual components in your design has to be calculated (i.e., the adder, the decoder, and the SRAM array itself).

Design Optimization

Design Optimizations should be done to reduce the latency and the optimisations can be done to any part of the total design.Some of the optimisations are listed below but they are not limited to them:

1. **Use more advanced logic styles (domino, pass-transistor, pseudo-nmos) :** Instead of using static CMOS logic to implement the adder and decoder, you can explore the use of alternate logic styles. For example, you might try to implement the decoder using domino logic, or use a Manchester carry chain for the adder.Remember there are almost always tradeoffs when choosing between logic styles, so it will be up to you to determine which logic style best fits with your original optimization goals.
2. **Optimize supply voltage partitioning:** Reducing the supply voltage is one of the most effective ways to lower dynamic power consumption in digital circuits. However, a uniform reduction of V_{DD} can significantly degrade speed. To balance power and performance, one option is to selectively use multiple supply voltages within the SRAM. For example, non-critical blocks such as the final decode drivers can operate at a lower voltage to cut switching power, while timing-critical paths (bitlines, sense amplifiers) remain at nominal V_{DD} to preserve speed. If multiple supply domains are used, transistor parameters (e.g., C_G, C_D, R_{sq}) should be re-characterized at the chosen voltage levels to ensure accurate modeling.
3. **Pre-charge Transistors:** After every read operation, the bitlines need to be pre-charged back to VDD - this is easily achieved by connecting a PMOS pre-charge transistor to each of the bitlines. Figure 2 shows a schematic of the pre-charge transistors connected to one column of the SRAM. Typically, the precharge transistors would be physically placed at the top of the SRAM array, so the dimensions of the cell containing these devices should match the width of the SRAM cell. You are free to size the pre-charge transistors as you like, but you must make sure that the time it takes for the bitlines to get precharged to within 10% of VDD is less than the total critical path delay through your adder and decoder.
4. **Output buffer:**In order to drive the capacitance loading the outputs of your memory, an output buffer (inverter) will be used. The buffers will usually be connected to the bottom of the SRAM

column, so like the pre-charge transistors, the buffer's layout should match the SRAM cell's width. You are free to size the output buffers as you like.

Report

The quality of your report is as important as the quality of your design. Be sure to provide all relevant information and eliminate unnecessary material. Organization, conciseness, and completeness are of paramount importance.Some of the details which can be included in the report are as follows:

1. **Block-level Design Description:** Brief explanation of each block (adder, decoder, SRAM array, sense amplifier), its function, and how the blocks are designed and interconnected.
2. **SRAM Cell Analysis:** Discuss the purpose of each transistor as well as how the W/L ratio is chosen for each transistor in the 6T SRAM cell and how the cell performs read and write operations.
   Include the static noise margin (SNM) and write noise margin (WNM) plots with clear interpretations.
3. **Critical Path Analysis:** Identify the longest delay path in the design (e.g., from address input to stable data output).
   Provide timing measurements for each stage , critical path and worst case delay(adder, decoder, bitline, sense amplifier).
4. **Latency and Performance:** Report the total latency of the 32X32 SRAM (from input transition to output stabilization).
   Compare performance before and after optimizations.
5. **Power Considerations:** Estimate dynamic power (from switching activity and supply voltage) and discuss how optimizations like voltage partitioning, transistor sizing, or buffers affected it.
6. **Design Optimizations:** Summarize the optimizations you explored (logic style changes, voltage partitioning, output buffers, etc.), and explain:
   - What each optimization achieved (e.g., lower delay, reduced power).
   - What trade-offs were introduced (e.g., area overhead, increased complexity, or reduced noise margin).
7. **Simulation Results:**
   - Back-to-back read operation correctness.
   - Delay measurements for each block.
   - SNM/WNM butterfly plots.
   - Any measured voltage swings on bitlines and sense amplifier outputs.
8. **Comparison with Initial Design:** Highlight improvements over the intial baseline design, in terms of delay, power and stability.
9. **Conclusion:** Summarize your final design, its strengths and the main challenges faced. Briefly comment on what further improvements could be explored beyond the scope of this project.
   Also mention the contribution by each member in the team.