EE413 Term Project: SRAM + 8-bit Adder

Emre Ant
Middle East Technical University
Electrical and Electronics Engineering
Student ID: 2518561
emre.ant@metu.edu.tr

Emre Kumlu
Middle East Technical University
Electrical and Electronics Engineering
Student ID: 2516474
emre.kumlu@metu.edu.tr

Abstract*—This project focuses on implementing a digital system that combines 16x8 SRAM (Static-Random Access Memory) and 8-bit CLA (Carry Lookahead Adder). SRAM operates as a storage unit with both reading and writing operations. SRAM block contains submodules such as "Precharge", "6T Memory Cell", "Sense Amplifier", "Write driver", "2-phase clock", and "Decoder". Moreover, CLA is a type of an adder circuit that can execute faster addition operation compared to traditional ripple carry adder. The overall circuit is capable of 8-bit addition of input data with the word that is written at the SRAM. In this paper, overall design along with all submodules will be explained, and schematic, layout design with simulation results will be shared.

Keywords*—SRAM (Static Random Access Memory), CLA (Carry Lookahead Adder), Layout, Sense Amplifier, 6T Memory Cell, Power Consumption

I. INTRODUCTION

In modern digital systems, memory and arithmetic operations are essential components. This project aims to design and integrate CLA and SRAM which are highly used in digital integrated circuits. SRAM functions as a data storage circuit and CLA is responsible for high-speed addition operation. The objective is to generate the design with minimum area, faster operation and power efficiency. Before starting to work on this project, we did literature research on SRAM and CLA. We found out that "sizing" and "timing" are two significant points that needs to be paid attention. We chose best option for "timing" and "sizing" to be used in our circuit. Moreover, in the design, a 10 MHz clock is used, so that delay of the design should be smaller than 5 ns to satisfy the operating frequency. To reduce the delay, the total chip area should be small enough in order not to suffer from the large parasitic capacitances. This report explains the design procedure of SRAM+CLA with theoretical explanations and shows simulation results of SRAM and 8-bit CLA.

II. STATIC RANDOM ACCESS MEMORY (SRAM)

Static random access memory (SRAM) is a memory type that can store bit data in a memory cell array as long as power is supplied to the circuit. This type of memory is called volatile memory. The designed SRAM is 16x8 which means that there are 16 words where each of the words is 8 bits. The SRAM operates with 1.8 V supply voltage. It takes an active high write enable (WR) input, and it is 1 for a write, 0 for a read operation. It also has a 4-bit address input for the selection of the memory word for both write and read operations. It takes 8-bit parallel input data for writing into a memory word, and it gives a parallel data output that is read. The SRAM also has an asynchronous reset input.

In every column, there are two lines called bit and its complement bitbar line. There are also 16 memory cells, 1 precharge circuit, 1 write drive circuit, and 1 sense amplifier circuit in each column, and they are all connected to the same bit lines. Moreover, there are 8 memory cells for 8-bit words in each row. The cells in a word are activated by word line signals. Therefore, there are 16 word line signals, and only one of them is high at a time which ensures only one memory word is activated at a time. For the SRAM operation timing, two-phase non-overlapping clocks are used. The SRAM is fed by a 100 MHz clock signal, and phase 1 clock is high when the external clock is low, whereas phase 2 clock is high when the external clock is high. Phase 1 is defined as the precharge cycle, and phase 2 is defined as the write/read cycle. Therefore, read and write operations are completed within one clock cycle (in phase 2). The schematic for the SRAM circuit is shown in Figure 1, and the layout together with DRC/LVS reports, and the extracted view are given in Figure A.1, Figure A.2, Figure A.3, and Figure A.4 in Appendix A. The area of the smallest rectangle of the SRAM layout is calculated as 33726 μm^2 which is given in Table 1 at the end of the paper.

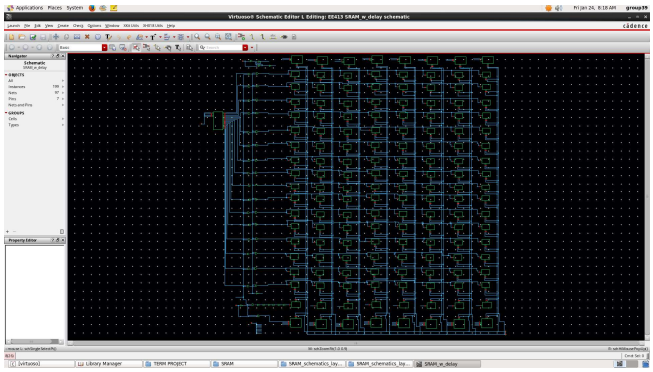

Fig. 1. SRAM schematic

A. SRAM 6-T Memory Cell

The fundamental unit of SRAM is called the memory cell. It is responsible for holding the bit data of SRAM. There are several configurations for the design. In this project, 6-T memory cell architecture containing six transistors is used. In the hearth of the cell, there are only back-to-back inverters. These inverters constitutes a bistable latch which has two stable states so that it can hold a single bit together with its complement as long as the power supply is on. In both sides of the latch, an NMOS transistor is connected that is used for accessing the bit data hold in the cell. Therefore, they are called access transistors. Word line enable signal (WL) is connected to the gates of the access transistors. When it is enabled, bit line (BL), and its complement bit bar line (BLbar) are shorted to the latch via the access transistors. The schematic of the 6-T memory cell is shown in Figure 2. Also, its layout is given in Figure A.5 in Appendix A.

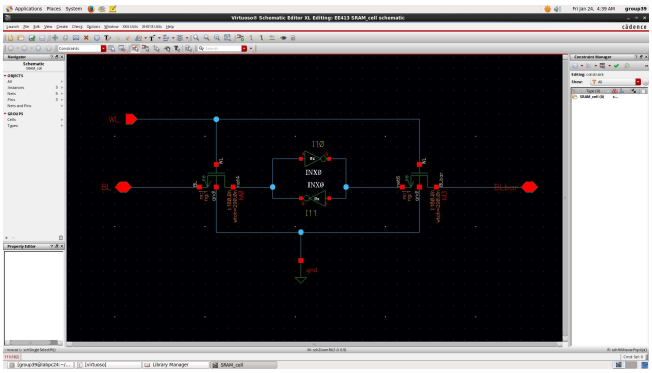

Fig. 2. SRAM 6-T memory cell schematic

- *Read Operation*

In order to read a bit stored in a memory cell, the bit lines are precharged to Vdd in the precharge cycle (phase 1). Then, in phase 2, the word line (WL) is activated to turn on the access transistors. Assumming that 0 is stored in the memory cell meaning that Q, and its complement Qbar are 0 and 1, respectively. When the word line becomes on, the bitbar line voltage does not change since it has the same voltage as the Qbar. However, the bit line voltage starts to be pulled down through the access transistor, and the pull-down transistor in the memory latch. When the voltage drop of the bit line exceeds some small value, the sense amplifier is activated which senses the voltage difference between bit and bitbar lines. After sensing, the sense amplifier helps to pull-down the voltage further and much faster. Finally, the bit line is pulled down to zero which is the desired bit value (Q), bitbar line remains zero, the read operation is completed, and the data can be read from the bit line. In order to achieve a stable read operation, the bit data in the memory cell should not change after reading it. Therefore, the bit value Q should not exceed the switching voltage of the latch inverters. For this, the NMOS pull-down transistors should be stronger than the access transistors to have stronger pull down to zero. The ratio of the W/L ratios of these transistors ({W/L} of access transistors / {W/L} of pull-down transistors) is called as cell ratio. Although it should be greater than 1 theoretically, after some simulations, the cell ratio is selected as 1 since it satisfied proper read operation. The sizes are tried to be minimized since there are too many memory cells in the SRAM, and they are selected as 290nm / 180nm for both transistors. Furthermore, for a successful read, the operation must be completed in 5 ns since the memory is at the precharge cycle for half period of the clock (5 ns), and the read/write operations are executed in the other half (phase 2). [1], [2]

- *Write Operation*

When WL is active (high) according to the address input of the circuit, NMOS pass transistors are turned on and transmits voltage that waits at the “bit line” and “bit line bar”. After write operation, NMOS pass transistors are turned off with logic low of the word line, and voltages at the bit line and bit line bar are stored at the at the back-to-back inverters (Q and -Q nodes respectively). In order to write logic “0” to the cell which was previously logic “1” (at the precharged cycle of the Φ1), write driver block of the circuit pull down the bit line to zero which was previously precharged to VDD. The corresponding word line of the address becomes logic 1 and access transistors (NMOS pass transistor logic) is turned on, bit line pulls Q to 0, and both bit line bar and -Q stay at 1.

In CMOS inverters, NMOS gate width and PMOS gate width are chosen as 290 nm and 810 nm to obtain symmetric rising edge and falling edge (to equalize the beta value of the transistors). Gate length is chosen as the minimum length of the technology which is 180 nm to reduce the area of the 6-T transistor cell. In SRAM, a large number of 6T memory cells are used, therefore it is crucial to minimize memory cell to obtain minimum total area of the total circuit layout. To write the data successfully, NMOS pass transistors (access transistors) need to be stronger or equal to PMOS transistors. Their ratio is referred as the “Pull-Up ratio (“W/L” of PMOS / “W/L” of access transistors) in the literature and it may vary with different circuits. According to our simulation results, Pull- Ratio is determined as 810/290 meaning that “beta” value of the access transistors and PMOS pull-up transistors are equal. Furthermore, for a successful write, the operation must be again completed in 5 ns similar to the write operation. [1], [2]

B. Two-phase Clock Generator

Two-Phase clock is used for ensuring that clock cycle of the different blocks is synchronized, otherwise; if we used one-phase clock, at a particular time, a block is taking logic high value of the clock, another block may take logic low value of the clock because of the delays though the clock distribution. To avoid this situation and ensure that clock is non-overlapping, two phase clock circuit is used. In order to generate a two-phase clock, SR latch with NOR gates have been used as shown in Figure 3. In two phase clock circuit, the outputs which are Φ1 and Φ2 are used as non-overlapping clocks. In overall SRAM, two phase clock is used in order to ensure that sense amplifier does not turn on during the precharge cycle. To do this, precharge block is supplied with Φ1 and read/write blocks are supplied with Φ2. Size of the inverters are determined to have enough separation between the two phases of the clock.

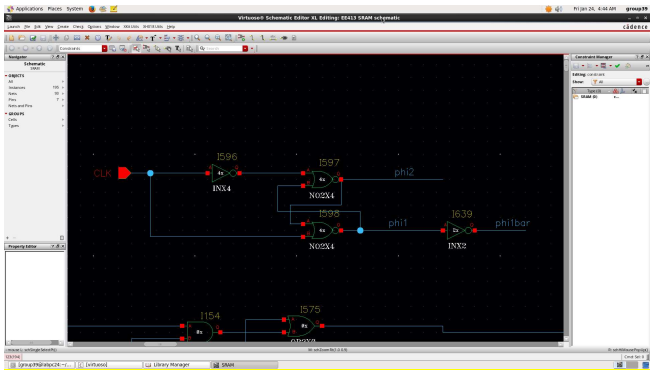

Fig. 3. Two-phase clock generator circuit schematic

C. *Precharge*

During phase 1 which is also called the precharge cycle, the precharge circuitry shown in Figure 4 is enabled. For every column (bit line) of the SRAM, there is a single precharge unit. This unit pulls up the voltage of both bit lines to Vdd during phase 1. As a core of the unit, two pull-up PMOS transistors, and one equalizer PMOS transistor are used. When phase 1 is active, precharge enable signal (PCbar – active low) becomes zero, and both three transistors become on pulling up the voltages of the bit lines. Equalizer transistor guarantees equal voltages in both lines. Moreover, extra two PMOS transistors are added to achieve asynchronous reset functionality. When reset is active, the bit line is disconnected from the pull-up circuitry, whereas bitbar line is always connected the pull-up via the extra PMOS transistor. For sizing of the transistors, large W/L ratios and strong PMOS transistors are chosen to achieve faster precharge since the bit lines are highly capacitive. For all transistors, the W/L ratios are selected as 2µm / 180nm. Moreover, the layout of the precharge circuit is given in Figure A.6 in Appendix A.

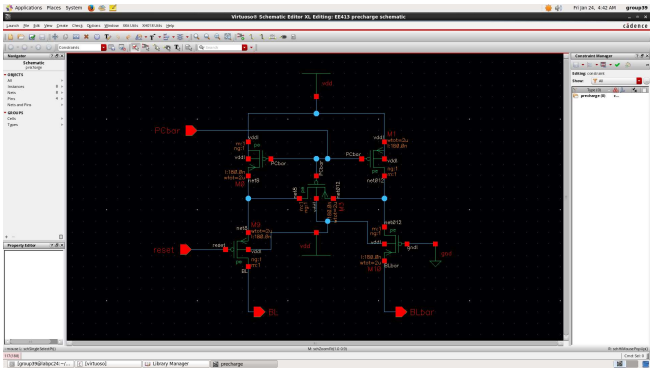

Fig. 4. Precharge circuit schematic

D. *Sense Amplifier*

Sense amplifier senses the voltage difference between the bit lines, and further pulls down one of them to complete read operation faster. As similar to the memory cell, a bistable latch with back-to-back inverters is used. These inverters are directly connected to the bit lines. However, it is not active at all times. An enable NMOS transistor is added which is only activated when WR (write/read) signal is 0 meaning that read operation is requested, and phase 2 (read/write cycle) signal is active. Therefore, the sense amplifier enable signal is generated by the logic WR'Φ_2 = WR' CLK. When it is enabled, the latch becomes active, and pulls the bit lines to the stable point. In the read operation, the timing of the enable signal is very critical. For proper read operation, the precharge cycle should be completely over, and the sense amplifier enable signal should become active after a certain period of time in order not to sense wrong voltage differences between the lines. To overcome this problem, six inverters are added in the sense enable path to create a proper delay. Moreover, the sizing of the transistor are selected in a way that the pull-up and pull-down are symmetric. These sizes are also selected large for faster read operation, due to large bit line capacitances. The PMOS transistors are selected as 1.5µm/180nm whereas the NMOS transistors are 1.2µm/180nm. The circuit shematic is shown in Figure 5, and the layout is given in Figure A.7 in Appendix A.

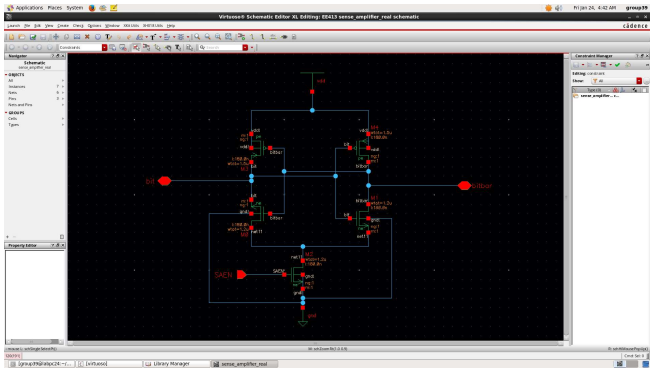

Fig. 5. Sense amplifier circuit schematic

E. *Write Driver*

Write driver circuit handles the task of writing the input data on the bit line and bit line bar. When WR (writing operation) is logic 1 and Φ2 is 1, then WE (Write enable) becomes 1 with the 2 input AND circuit (Φ2 and WR are the inputs of AND), which is placed before the write driver block. When reset is not activated and WE is 1, M0 and M1 transistors in Figure 6 is turned on. If the input data is 1 during this period, M3 transistor is turned on and 0 V (ground) is transmitted to the bitbar. If the data is 0, during high write enable, M2 transistor is turned on, and it pulls down the bit line to zero. To sum up, when data is 1, bit line bar is pulled down to zero, and when data is 0, bit line is pulled down to zero. Sizes of the transistors are chosen strong enough to be able to drive the bit lines. Because long metal layers at the layout results in high parasitic capacitance. Layout of the circuit is given in Figure A.8 in Appendix A.

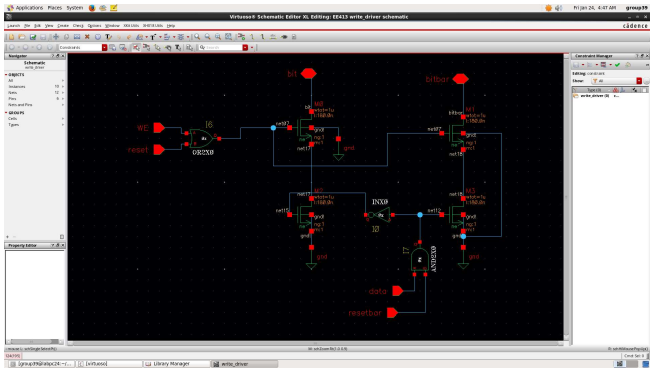

Fig. 6. Write driver circuit schematic

F. *Row Decoder*

Row decoder is used for selecting an 8-bit word in the memory to execute read or write operations on the selected word. It takes 4 bit address inputs, and gives 16 word line outputs for 16 memory words. Only one of the 16 word line outputs is made high depending on the address inputs. For example, when the address inputs are "0010", only the fourth word line becomes high meaning that word 4 is selected. For the design, 4 inverters which generates the inverted address inputs, and 16 four-input AND gates are used to generate minterms. The schematic and the layout of the row decoder are given in Figure A.9, and Figure A.10 in Appendix A, respectively.

G. SRAM Simulations

In order to justify that the SRAM design works correctly, a general simulation is made given in Figure A.11 in Appendix A. In that simulation, different random input data is written to 6 different memory words (from word 0 to word 5), and then these words are read to compare with the written data. All of the simulations are done with analog extracted views to account for the effects of the physical layout.

- *Power Consumption*

In every read and write operation, some power is consumed by the SRAM. In order to calculate the power consumption during write operation, three simulations are done. In the first simulation, the entire SRAM is written with full of 0s by activating the words (from word 0 to word 15) one by one in each clock cycle. Then the average current over 16 clock cycle is measured, and the power consumption is calculated by multiplying the average current with Vdd (1.8 V). In the second simulation, same steps are done except that the entire memory is written with full of 1s instead of 0s. Lastly, both entire memory write operations are done back to back which means that first, 0 is written to the entire array, then 1 is written in the same way. These two operations are repeated 4 times in the same simulation (in 128 clock cycle). Then the average current and power over 128 clock cycle are measured. The results are tabulated in Table 1, and the simulation waveforms are given in Figure A.12, Figure A.13, and Figure A.14 in Appendix A.

- *Read/Write Access Time*

To measure the read access time, we wrote zeros to a particular address (WR=1), then we read the bitline (WR=0). We measured the required time to read the data from bit line after rising edge of clock (Φ2). This delay corresponds to the worst case read access time since reading 1 does not take any time due to previous precharge of bit lines. It is measured as 1.79 ns as shown in Figure A.15 in Appendix A. To measure write access time from low to high (write 1), we wrote all 1s to a particular memory word which was previously contains all 0s, and we measured the delay between rising edge of the clock and change of Q output from 0s to 1s. The write access time from low to high (write 1) is obtained as 2.162 ns as shown in Figure A.16 in Appendix A, which is the worst case. To measure the write access time from high to low (write 0), we wrote all 0s to a particular word which previously contains all 1s. The write access time for this case is obtained as 2.038 ns as shown in Figure A.17 in Appendix A. The worst-case access times are also shown in Table 1.

III. 8-BIT CARRY LOOKAHEAD ADDER (CLA)

Carry lookahead adder is an adder type which is much faster than the traditional binary ripple carry adder due to the difference in carry generations. In standard adders (ripple carry adders), the adder is a combination of full adder blocks connected in series. Each full adder adds three bits (two one bit numbers and a carry in), and gives the sum and the carry as the output. The output carry of one full adder is connected to the input carry of the next full adder to add multiple bit numbers. However, the subsequent adder has to wait for the output carry of the previous adder to complete the addition. Therefore, in order to obtain the 8-bit sum result, the carry propagation delays for 8 stage should be waited. For the solution of the large total carry propagation delay, a carry lookahead adder is used. In this type of adder, all the internal carries are generated through 3 stages of gates by using only the first input carry. This reduces the overall delay significantly since the last sum stage waits only the carry delay of 3 stages. In the design, first, the carry generator circuit is designed. Then, using the carry generator, a 4-bit carry lookahead adder (CLA) is designed which takes an input carry and 2 four-bit numbers as inputs, and gives the 4-bit sum and the output carry as outputs. Finally, 8-bit carry lookahead adder is built by combining two 4-bit CLA in series.

A. Carry Generator

In order to reduce delay time of each carry output, carry lookahead generator is used. This circuit generates carry outputs with at most 3 stages of gates instead of generating each carry output from previous carry output as explained previously. Therefore, each carry out is calculated individually. To generate the schematic of the carry generator circuit, Cout equations from Equation B.1 to Equation B.5 in Appendix B are analyzed. After analyzing the equations, Couts are generated with sum of products by using AND, OR gates. The circuit schematic is shown in Figure 7. From the figure, it is seen that 5 input AND, OR gates have been used but at the internal layout of the 5 input gates, it is seen that it does not contain more than 4 series transistors at pull-down or pull-up path because 5 input gates are obtained with series connection of two different gate, so it does not violate the maximum allowed 4 stacked (series) transistors. Minimum size transistors have been used such that high to low and low to high transitions are symmetric. The layout for carry generator circuit is given in Figure B.1 in Appendix B.

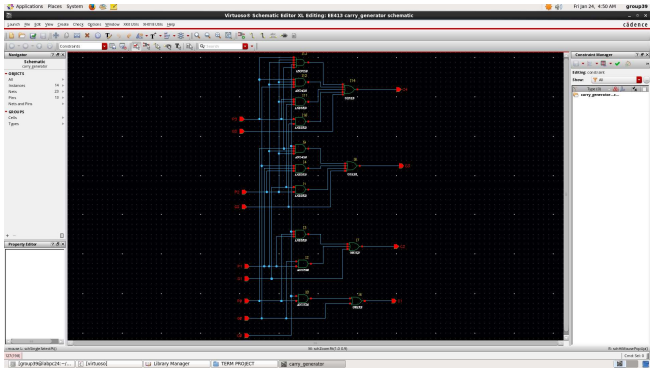

Fig. 7. Carry generator circuit schematic

B. 4-bit Carry Lookahead Adder (CLA)

4- bit CLA is generated by using carry lookahead generator block. Summation operation is simply EXOR of the inputs A, B, Cin (A XOR B XOR Cin) traditionally. The difference from the traditional adder is Cin is generated individually as explained previously, which leads to faster summation. The schematic of the 4-bit CLA is shown in Figure 8. Again, minimum size transistors are used. Physical layout of the circuit is shown in Figure B.2 in Appendix B.

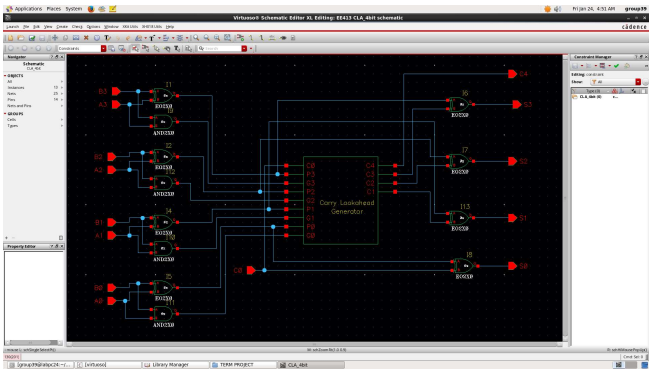

Fig. 8. 4-bit CLA circuit schematic

C. 8-bit Carry Lookahead Adder (CLA)

The 8-bit carry lookahead adder is built with two 4-bit CLA blocks connected in series. The carry output (C4) of one of them is connected to the carry input (C0) of the other one. Therefore, the upper more significant 4 bits only wait for the generation of the output carry of the lower less significant 4 bits. This design is slower than a single block 8-bit CLA since upper 4 carry bits cannot be generated until C4 is generated. However, it is much more modular and easy to integrate as a trade-off. The circuit schematic for the 8-bit CLA is shown in Figure 9. The layout together with the DRC/LVS reports, and the analog extracted view are given in Figure B.3, Figure B.4, Figure B.5, and Figure B.6 in Appendix B. The area of the smallest rectangle of the 8-bit CLA layout is calculated as 3982 μm^2 which is also given in Table 1.

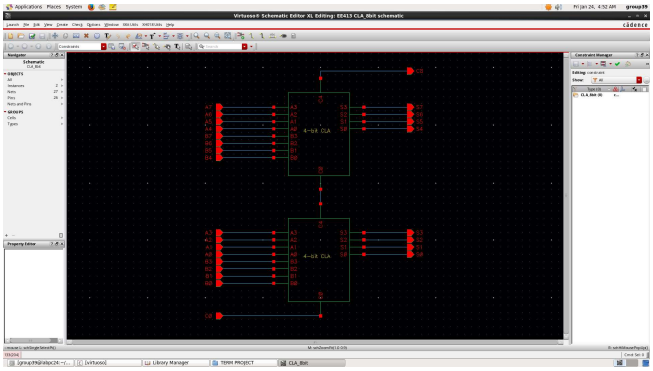

Fig. 9. 8-bit CLA circuit schematic

D. 8-bit CLA Simulations

For the test of the 8-bit CLA, pre-layout simulations were done to test functionality. Then, the parasitics were extracted for the post-layout simulations to obtain the delays and power consumptions of the 8-bit CLA.

- *Pre-Layout Simulations*

To realize the functionality test of the 8-bit CLA, desired numbers to be added are generated at the test bench, and in every 5 ns, inputs are renewed. The simulation result for the functionality test of 8-bit CLA is given in Figure B.7 in Appendix B. The results of the simulation are also tabulated as shown in Table B.1 in Appendix B. It can be concluded that the 8-bit CLA functions correctly.

- *Post-Layout Simulations*

a) Delays and Switching Power Consumptions

When the inputs are changed, the outputs are updated after some delay, and these delays differ for different paths in the circuit. In order to find the worst-case delay paths, the inputs are arranged so that the outputs are switched from low to high, or high to low. The input combinations to measure all the HL and LH delays at the outputs are given in Table B.2 in Appendix B. The low to high delays are measured for all sum output bits from 50 percent of the input to 50 percent of the outputs. The worst case high to low, and low to high delays are given in Table 1, and the simulation waveforms showing them are given in Figure B.8, and Figure B.9 in Appendix B.

It can be seen from Table 1, and the waveforms, the worst case low to high transition occurs at the carry output (C8), and the worst case high to low transition occurs at the sixth significant sum bit (S6). In overall, S6 is found to be the worst case delay path. These results are acceptable and can be expected. The delays at the upper 4 most significant bits are larger than the lower half bits since the upper half waits for the carry out of the lower half block. When C4 is generated by the lower block, S4 can be expected to transition first since it is only waiting for the carry input (C4). Moreover, if we very roughly assume that each carry is generated at equal times since both of them are generated through 3 gate stages, C8 can be expected to be generated next. Then, the carries are EXORed with the inputs to obtain S7, S6, and S5. Therefore, the worst-case delay path can be expected to be one of them as a rough assumption. However, there are many things to consider the delays such as the parasitics extracted from the layout, and the condition of the previous signals.

Average power dissipation is calculated at the transient duration for HL and LH cases. Average supplied current and average power dissipation during the worst case low to high transition are 223 µA and 402 µW respectively, whereas average supplied current and average power dissipation during the worst case high to low transition are 121 µA and 218 µW, respectively. Simulation results are also tabulated in Table 1, and the simulation waveforms showing the current waveforms are given in Figure B.10, and Figure B.11 in Appendix B.

b) Leakage Current (Non-Switching)

Although the inputs are not switching, there is a leakage current drawn from the source which leads to a static power consumption. Two cases were simulated where all the inputs are 1 (high), and all the inputs are 0 (low). The test waveforms are given in Figure B.12, and Figure B.13 in Appendix B. In the test, all the inputs are set to 0 for the first 50 ns, and they are switched to all 1s for a duration of 50 ns. Then, to remove the dynamic power consumption during switchings, only the middle 30 ns periods are considered to calculate the average leakage current. The leakage currents and the corresponding leakage powers are given in Table 1. It can be seen that the leakage current and power are much larger when both inputs are full of 1s due to the gate leakage, and the subthreshold conduction leakage.

IV. Overall Circuit with SRAM + Adder

After designing the 16x8 bit SRAM, and the 8-bit carry lookahead adder (CLA), both blocks are combined in the overall circuit as shown in Figure 10. The overall circuit has 5 inputs which are 4-bit address input, active high write enable input (WR), 8-bit data input, clock input with 100 MHz frequency, asynchronous reset input, and input carry. By activating the write signal (WR = 1), 8-bit data input can be written to any memory word in the SRAM specified by the 4-bit address input. Moreover, the bit lines of the SRAM are connected to one of the parallel 8-bit input of the CLA. The other parallel 8-bit input of the CLA is fed by the same 8-bit data input of the overall circuit. When the write signal is disactivated (WR = 0), SRAM executes a read operation, and the memory word specified by the address input is read from the bit lines as the output of the SRAM. Therefore, in reading operation, the memory word that is read from the bit lines is added to the 8-bit data input together with the input carry. Then, the summation result which is the 8-bit sum output is available at the outputs of the CLA. It also gives a carry output generated from the sum. As a result, the 8-bit sum output and the carry out are valid only in phase 2 (when external clock is high), and read operation is requested by setting WR to 0. The layout of the overall circuit together with the DRC/LVS reports, and the analog extracted view are given in Figure C.1, Figure C.2, Figure C.3, and Figure C.4 in Appendix C. The total layout area is calculated as 49424 μm^2.

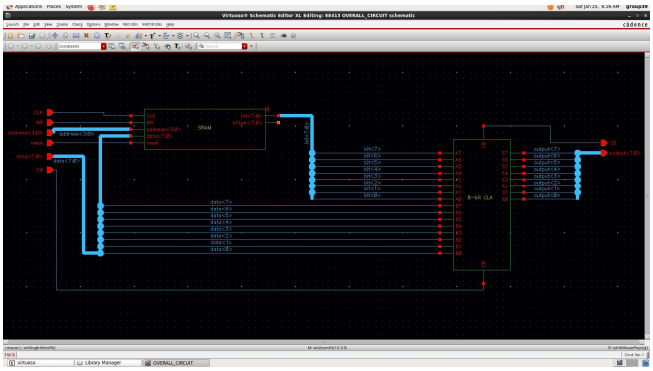

Fig. 10. Overall circuit (SRAM + CLA) schematic

A. Overall Design Simulation Results

Analog extracted simulation result of the overall circuit (SRAM+CLA) is shown in Figure C.5 in Appendix C. When WR=1 (write operation), data at the input (11011111) is written to the memory word which is specified by the "0010" address in the SRAM at the rising edge of the clock (Φ2). Then when WR=0 (read operation) previously stored data (11011111) and new data (00011001) are summed at the rising edge of the clock (Φ2). The result at the output is 11111000 and Cout is 0 which is indeed the correct result. Clock to Q delay is measured as 3.93 ns for this configuration. For different input configurations clock to Q delay may differ from 2.5 ns to 4.5 ns. Delays for some other cases are shown in Figure C.6, Figure C.7, and Figure C.8 in Appendix C. The delays in the analog extracted simulations are quite larger than the schematic simulations. At the schematic, average delay is near 2 ns, whereas at the analog extracted simulations, average delay is near 3.5 ns. This is expected since lots of parasitics are added in post-layout simulations, which introduces extra delays compared to the schematic simulations.

Using the same input test vectors in the adder simulation, one of the inputs are written to the memory, then it is read in the next clock cycle, and it is added to the other input. This operation is done for all four test vectors in Table 1, and the average power dissipation are obtained as shown in Figure C.9 in Appendix C that are also tabulated in Table 1. Again, these values are larger than that of the schematic simulations. To have a final comparison, all of the results for the SRAM, CLA, and the overall circuit are given in Table 1.

TABLE I. Results for The Sub-Blocks and The Overall Design

SRAM (Static Random Access Memory)	
Power Dissipation / Writing 0s	494.8 µW
Power Dissipation / Writing 1s	466.6 µW
Power Dissipation / 4 times 2 Full Array	512 µW
Worst Case Write Access	2.162 ns
Worst Case Read Access	1.79 ns
Worst Case Delay	2.162 ns
Layout Area	33726 μm^2
CLA (Carry Lookahead Adder)	
Worst Case HL Delay / Power	2.59 ns / 218 µW
Worst Case LH Delay / Power	2.45 ns / 402 µW
Leakage Current / Power (High Inputs)	56 nA / 100 nW
Leakage Current / Power (Low Inputs)	1.03 nA / 1.86 nW
Layout Area	3982 μm^2
Overall Circuit (SRAM + CLA)	
Layout Area	49424 μm^2
Power Dissipation	616 µW

V. Conclusion

This project offers us significant opportunities for implementing digital circuits in a semi-custom environment. We gain a deep understanding of IC design by designing and combining Static-Random Access Memory (SRAM) and Carry-Lookahead Adder (CLA). We analyze and think deeply about how subblock of the circuits can be combined such that they do not affect each other badly. "Sizing" and "Timing" are two points to be designed selectively. They directly affect the operation of the circuit. Large circuits are almost impossible to design and test wholly. It is necessary to partition the circuits into submodules which can be tested and designed individually. Semi-custom environment enables us to design physical layout much less time-consuming, but may affect adversely parameters such as area, power consumption and delay. It has limited flexibility to change the design. We needed to balance between time-efficiency and performance. Overall, this project developed our design abilities that need to be combined with theoretical background.

References

[1] N. Weste and D. Harris, *CMOS VLSI Design : A Circuits and Systems Perspective*. 2004.

[2] Singh, J., Mohanty, S. P., & Pradhan, D. (2012). *Robust SRAM designs and analysis*. Springer.

APPENDIX A
STATIC RANDOM ACCESS MEMORY (SRAM)

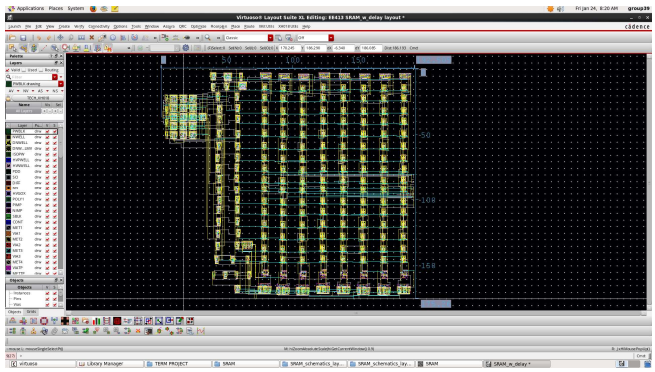

Fig. A.1. SRAM layout

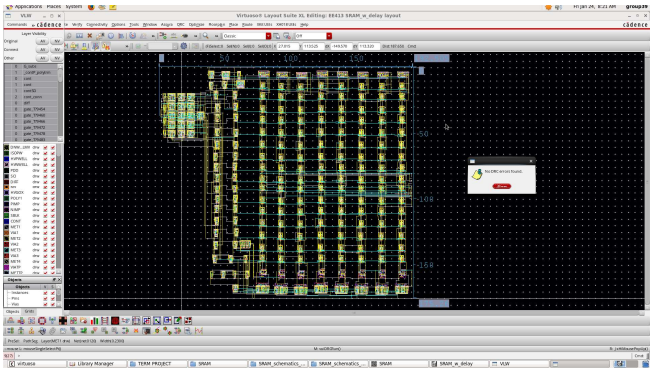

Fig. A.2. SRAM layout – DRC result

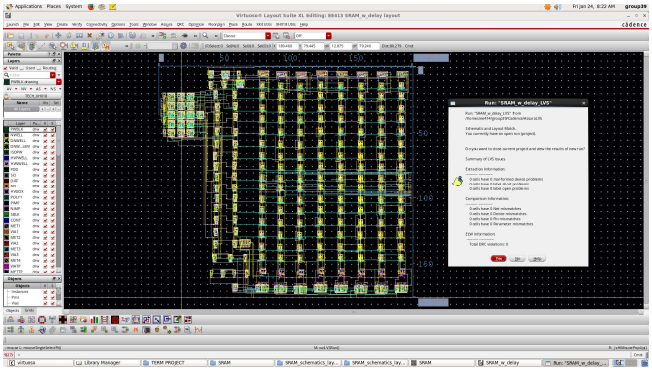

Fig. A.3. SRAM layout – LVS result

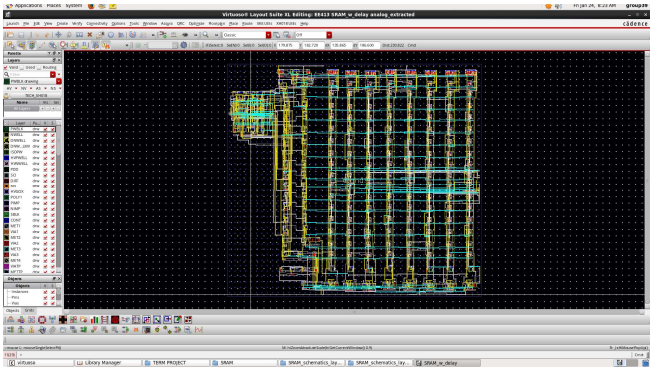

Fig. A.4. SRAM layout – Analog extracted view

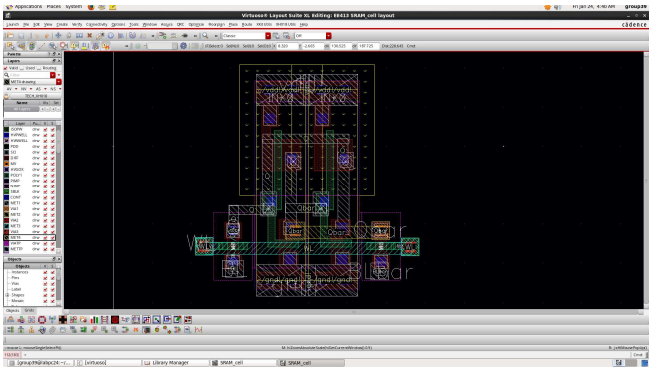

Fig. A.5. Memory cell layout

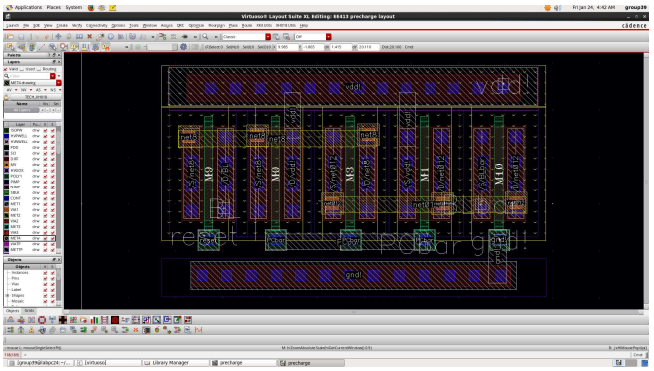

Fig. A.6. Precharge circuit layout

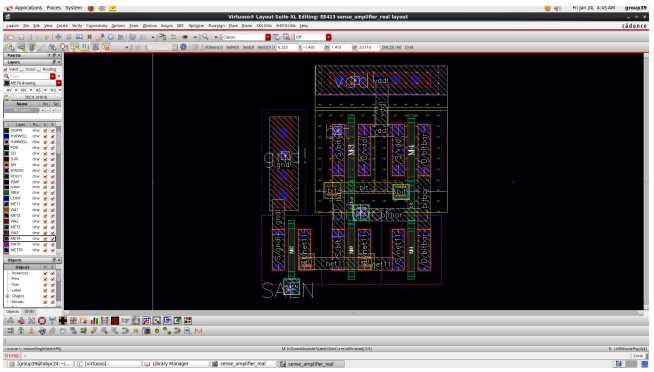

Fig. A.7. Sense amplifier layout

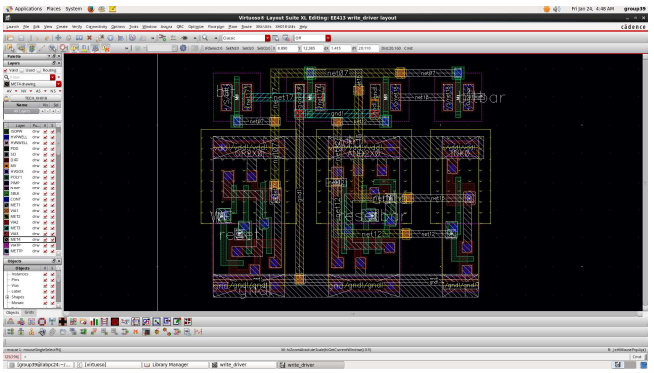

Fig. A.8. Write driver layout

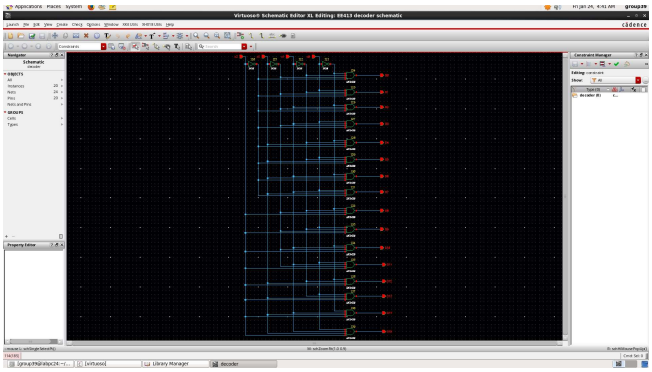

Fig. A.9. Row decoder schematic

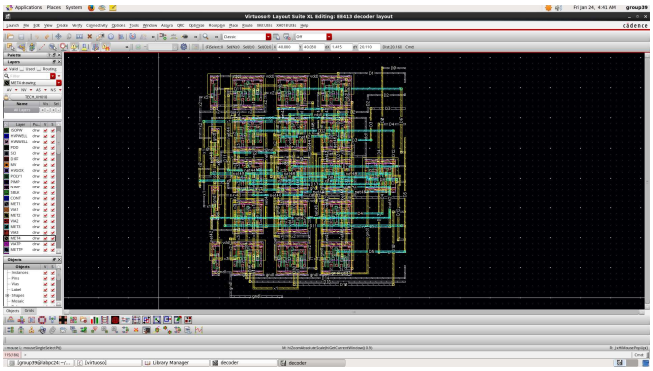

Fig. A.10. Row decoder layout

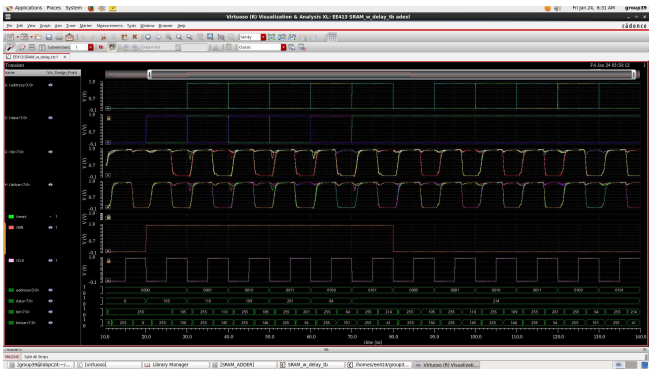

Fig. A.11. Simulation result for SRAM general operation

Fig. A.12. Simulation result when the entire SRAM is written with 0s (average current is shown)

Fig. A.13. Simulation result when the entire SRAM is written with 1s (average current is shown)

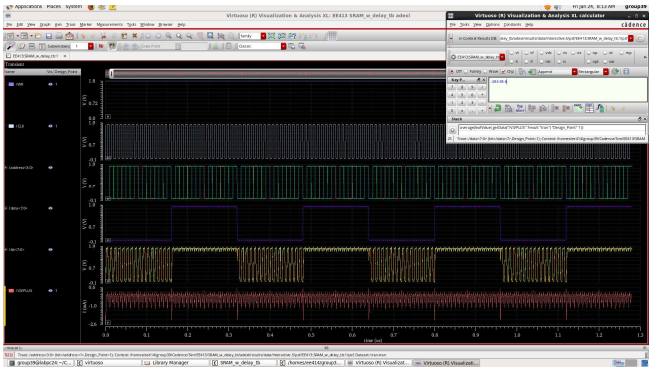

Fig. A.14. Simulation result when two full array write operations are done 4 times (average current is shown)

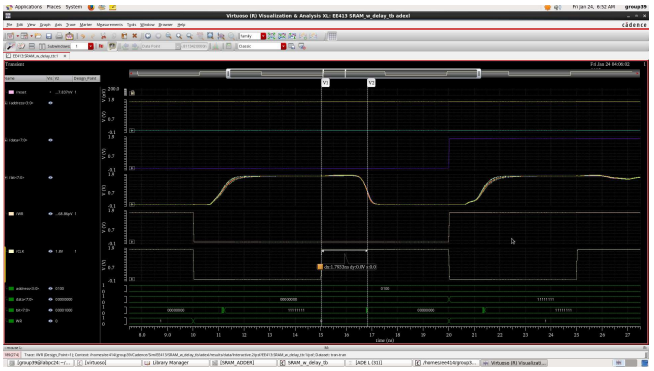

Fig. A.15. Simulation result for the worst-case read access time

Fig. A.16. Simulation result for the worst-case write (1) access operation

Fig. A.17. Simulation result for the worst-case write (0) access operation

APPENDIX B
8-BIT CARRY LOOKAHEAD ADDER (CLA)

$$C0 = Input\ Carry\ \rightarrow No\ computation \qquad (Equation\ B.1)$$

$$C1 = G0 + P0C0 \qquad (Equation\ B.2)$$

$$C2 = G1 + P1(G0 + P0C0) \qquad (Equation\ B3)$$

$$C3 = G2 + P2\big(G1 + P1(G0 + P0C0)\big) \qquad (Equation\ B.4)$$

$$C4 = G3 + P3\Big(G2 + P2\big(G1 + P1(G0 + P0C0)\big)\Big) \qquad (Equation\ B.5)$$

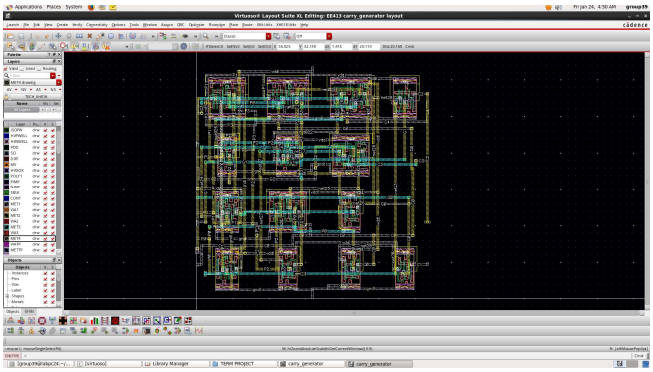

Fig. B.1. Carry generator layout

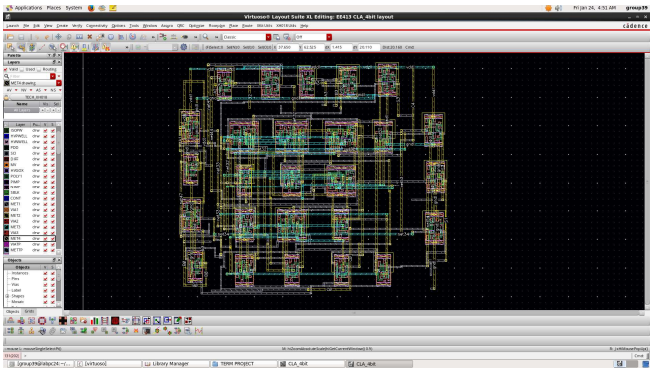

Fig. B.2. 4-bit CLA layout

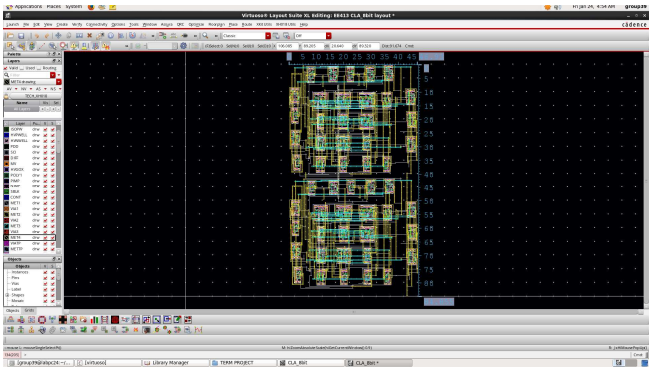

Fig. B.3. 8-bit CLA layout

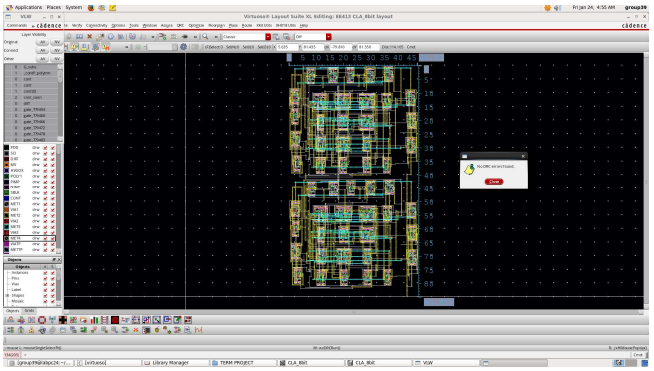

Fig. B.4. 8-bit CLA layout – DRC result

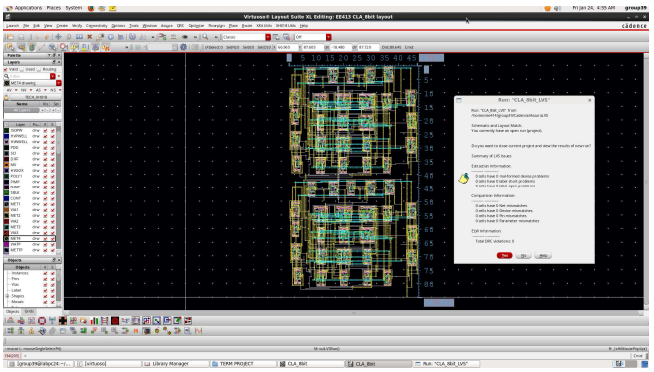

Fig. B.5. 8-bit CLA layout – LVS result

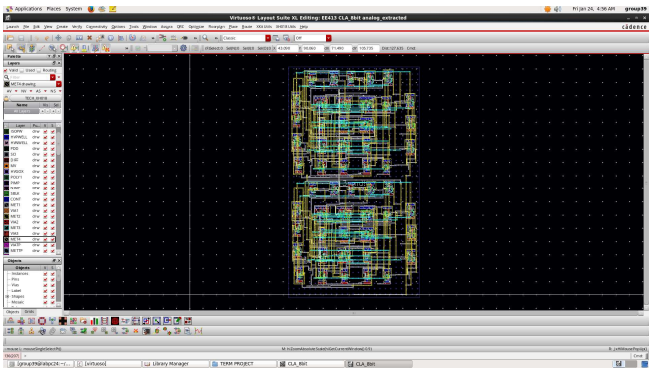

Fig. B.6. 8-bit CLA layout – Analog extracted view

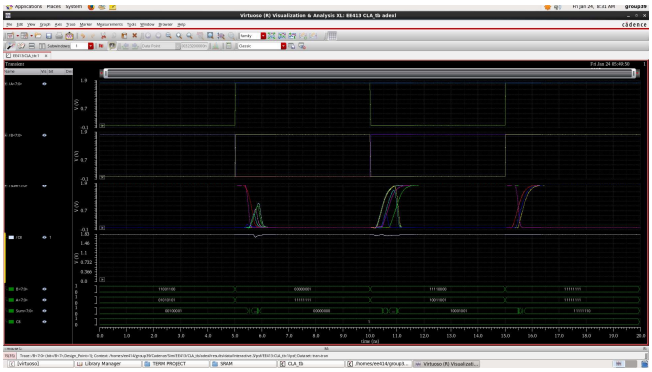

Fig. B.7. Functionality test of 8-bit CLA with 4 different input vectors

TABLE B.1. SUMMATION OF FOUR DIFFERENT INPUT VECTORS

A<7:0>	B<7:0>	SUM<7:0>	C8
01010101	11001100	00100001	1
11111111	00000001	00000000	1
10011001	11110000	10001001	1
11111111	11111111	11111110	1

TABLE B.2. INPUT COMBINATIONS FOR MEASURING DELAYS

A<7:0>	B<7:0>	SUM<7:0>	C8
00000000	00000000	00000000	0
00000000	11111111	11111111	0
00000001	11111111	00000000	1
00000000	00000000	00000000	0

Fig. B.8. The worst-case high-to-low delay (S6)

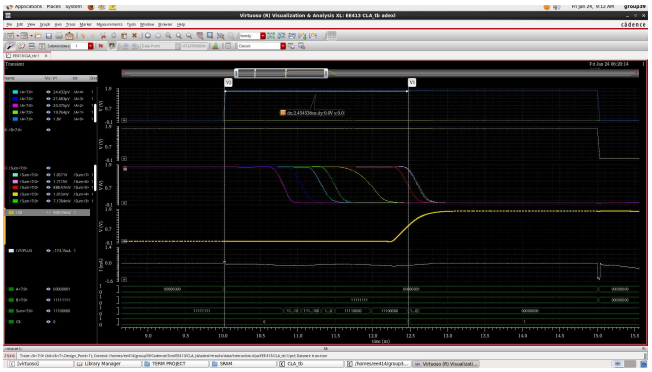

Fig. B.9. The worst-case low-to-high delay (C8)

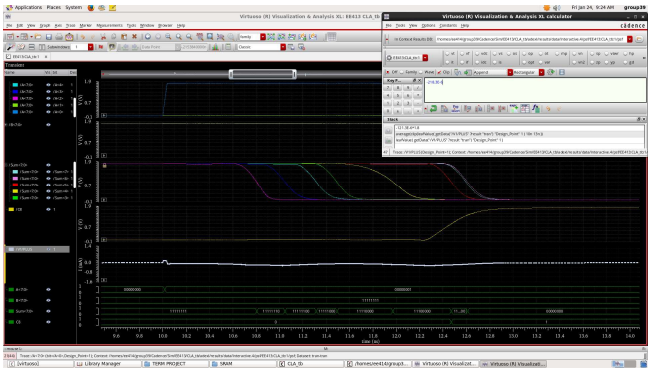

Fig. B.10. Power consumption over the worst-case high-to-low transition

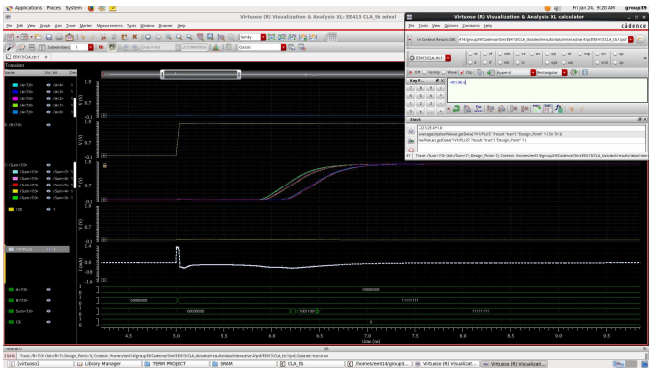

Fig. B.11. Power consumption over the worst-case low-to-high transition

Fig. B.12. Average leakage current when all inputs are 1 (non-switching)

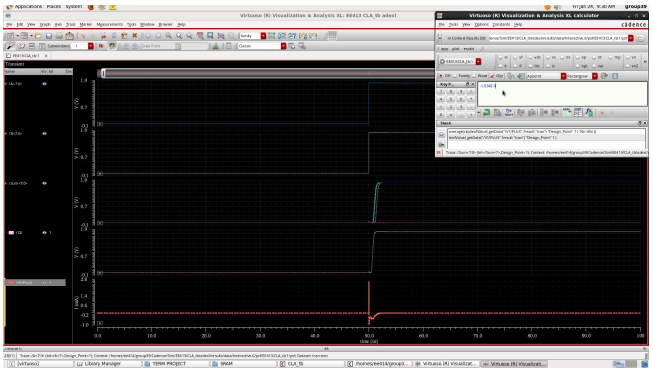

Fig. B.13. Average leakage current when all inputs are 0 (non-switching)

APPENDIX C
OVERALL CIRCUIT WITH SRAM + ADDER

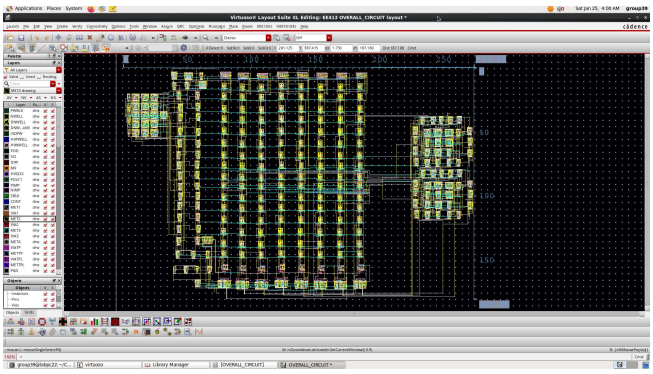

Fig. C.1. Overall circuit (SRAM + CLA) layout

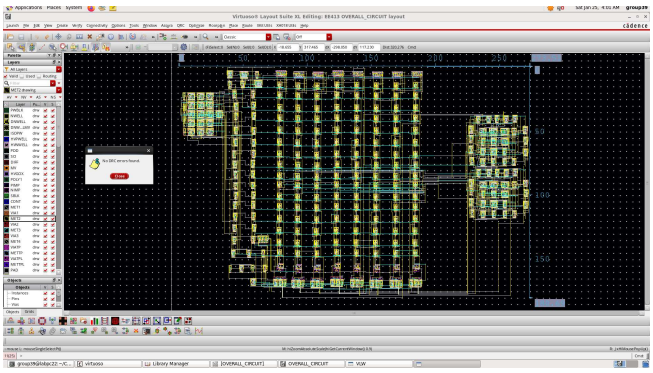

Fig. C.2. Overall circuit (SRAM + CLA) layout – DRC result

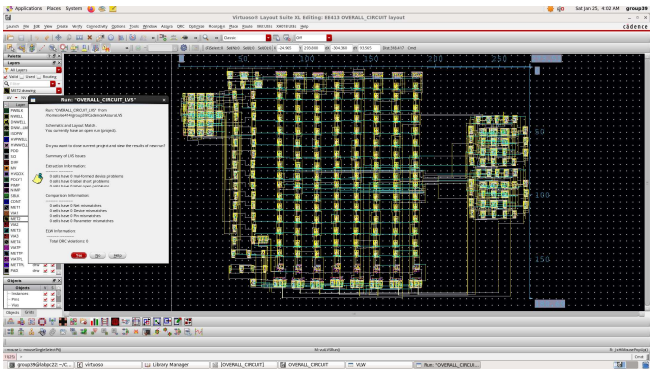

Fig. C.3. Overall circuit (SRAM + CLA) layout – LVS result

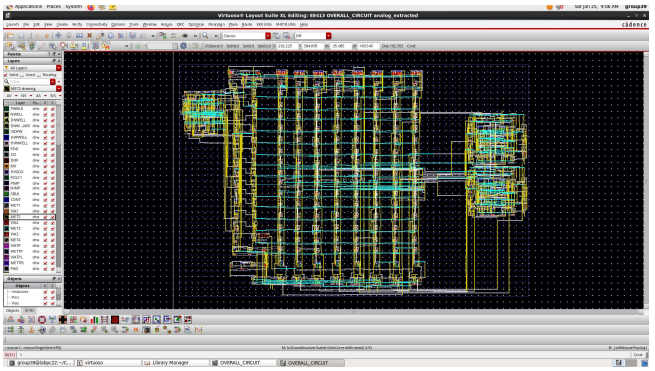

Fig. C.4. Overall circuit (SRAM + CLA) layout – Analog extracted view

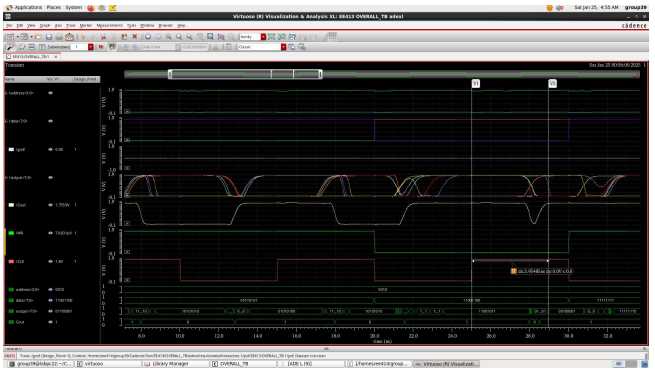

Fig. C.5. Simulation result of the overall circuit with delay = 3.95 ns

Fig. C.6. Simulation result of the overall circuit with delay = 4.53 ns

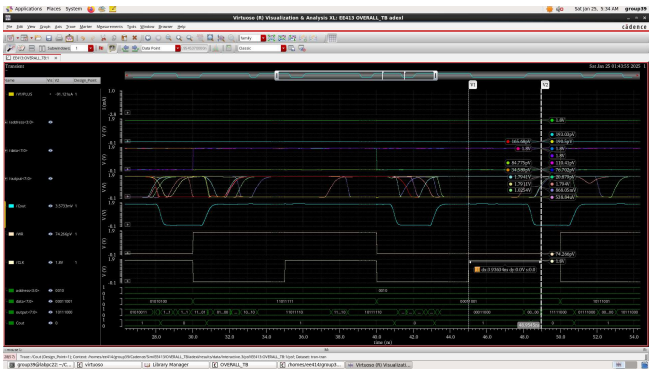

Fig. C.7. Simulation result of the overall circuit with delay = 3.93 ns

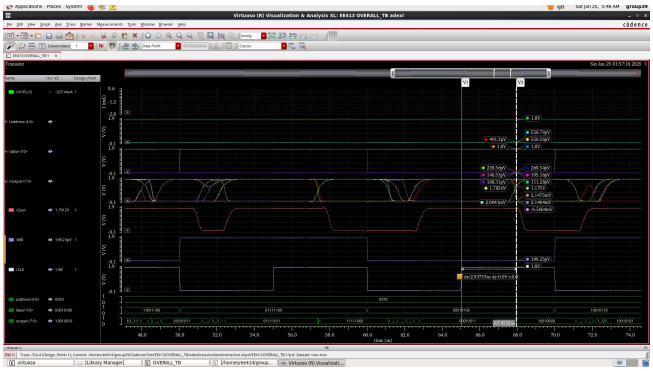

Fig. C.8. Simulation result of the overall circuit with delay = 2.94 ns

Fig. C.9. Average power dissipation of the overall circuit