UNIVERSITY OF CALIFORNIA
College of Engineering
Department of Electrical Engineering and Computer Sciences
Last modified on Nov. 14th, 2007.

Elad Alon ***FALL 2007 PROJECT PHASE III*** *EECS 141*
Final Report due Wed. Nov. 28th @ 240 Cory
Poster due Thurs. Nov. 29th @ BWRC

1. Putting it all together

The principal goal of Phase III of the project is to complete a fully functional 32x32 SRAM design. For convenience, the SRAM block diagram is repeated below.

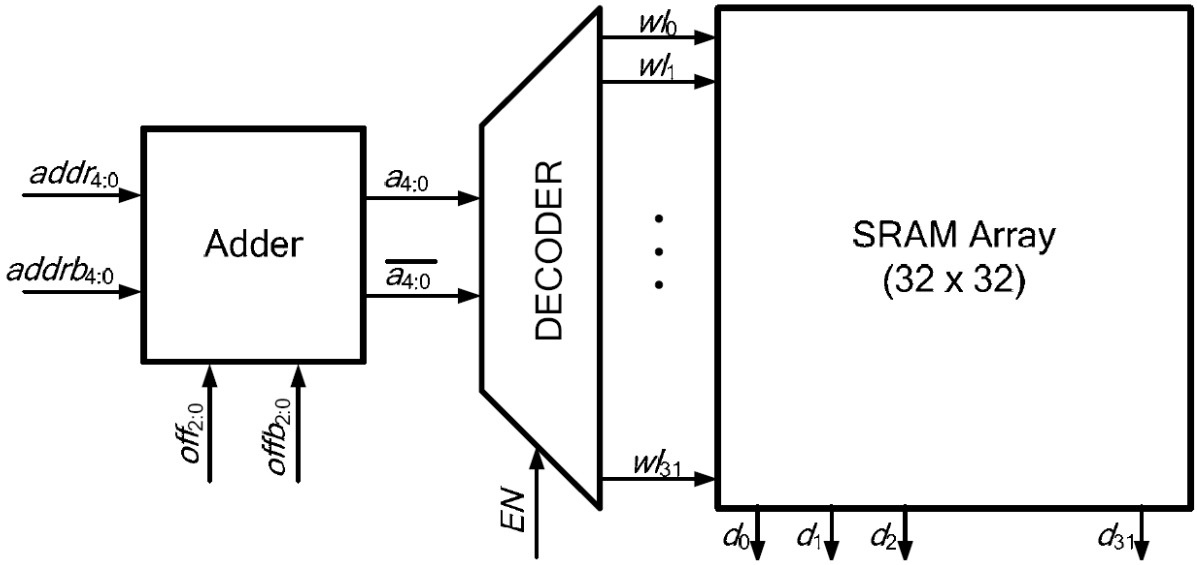

Figure 1. SRAM block diagram.

This phase of the project allows for the most creativity, but will also be the most time consuming of the three phases. We strongly recommend that you start early in order to allow yourself plenty of time to design, optimize, and assemble the complete SRAM.

2. Design Optimization

In the previous two phases, explicit instructions were given on how to implement and assemble the adder, decoder, and SRAM cell. As with these previous phases, the primary constraint on your design is that it must function correctly. However, beyond building a functional SRAM, in this final phase of the project it will be up to you to decide which design metrics you'd like to focus on optimizing. For example, you can try to build the fastest SRAM possible, minimize the SRAM's power consumption, pack the design into the smallest area, or shoot for a balance of two or three of these metrics. No matter which of these goals you pick, your final design should include substantial optimizations/modifications to at least one out of the three principal blocks: the adder, the decoder, and the SRAM array.

Several examples of potential optimizations are described below; note however that these are only a few of the possible optimizations that can be made to your design. The optimizations you decide to apply are limited only by your own creativity, but you do need to adhere to the constraints and guidelines on input capacitance, rise/fall times, margins, etc. provided in the project phase I handout.

1. **Redesign the SRAM cell for improved area, speed, and/or power**
   In phase I of the project, you were given a schematic and layout of a functional 6-T SRAM cell. This cell is far from optimized in terms of area or sizing. Thus, in this phase of the project you are free to design your own cell that improves upon the unoptimized cell given to you in phase I. If you do decide to redesign the cell, you must meet the following requirements: the voltage rise in the cell during a read should be no higher than 350mV, and the worst-case cell voltage during a write must be less than 250mV.

2. **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.

3. **Use a lower V_{DD} and/or add one or more extra power supplies to reduce power**
   Reducing the supply voltage of a digital circuit is one of the most effective means for trading off performance for reduced power consumption, and thus one option you can explore is to lower the V_{DD} for the entire SRAM. If you are shooting for a balance of speed and power, you can also explore using a separate supply voltage for various pieces of the design. For example, you might to choose to use a lower voltage in the final decode drivers in order to reduce their switching power without significantly impacting the overall delay. Note that if you use a lower supply voltage somewhere in your design, you should re-characterize the transistor parameters (C_G, C_D, R_{sq}) at this new voltage.

4. **Redesign the adder/decoder with only the adder's C_{in} constrained to 5fF**
   Since you are doing the design of both the adder and the decoder, your only constraint on C_{in} is that the inputs to the adder must have less than 5fF of capacitance. In other words, you can make the input capacitance of your decoder whatever you would like it to be. This allows you to resize the decoder or even completely change its logical topology. You could even merge gates that are logically part of the decoder into your adder design if you'd like to.

In your final report, you should state what your objectives were for the optimizations you chose to apply. Whether you optimized for speed, power, layout area, or some combination of the three, please be very clear about the methods you used to improve your design. Part of the judging of your project will be based on how well the optimizations you made actually coincide with your stated goal. Whichever optimizations you choose to do, remember that you do need to complete an LVS and DRC clean layout of the entire design – including the 32x32 array, the adder, the decoder, and two additional peripheral circuits described on the next page.

1. **Pre-charge Transistors**
   After every read operation, the bitlines need to be pre-charged back to V_{DD} – 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 V_{DD} is less than the total critical path delay through your adder and decoder

2. **Output buffers**
   In order to drive the capacitance loading the outputs of your memory, an output buffer (inverter) will be used. Figure 2 shows a schematic of the output buffers connected to one column of the SRAM. The buffers will usually be connected to the bottom of the SRAM column, so like the pre-charge transistors, the layout of the buffers should match the width of the SRAM cell. You are free to size the output buffers as you like.

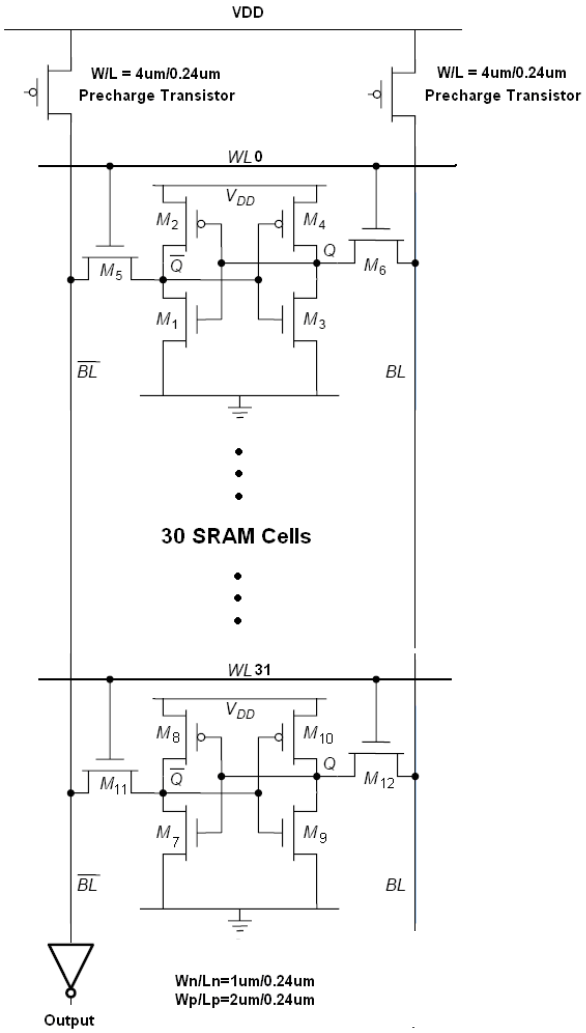

Figure 2. Pre-charge and Output Buffer schematic annotated with appropriate sizes.

To save you some time, we will provide you with schematics and layout of per-column pre-charge transistors and output buffers that you can use in your design – we will make an announcement on the webpage and newsgroup as soon as these cells are ready and available for you to copy. Note that these layouts have been matched to the width of the SRAM cell we gave you, so if you change your SRAM cell design you will need to re-layout the pre-charge transistors and output buffers as well.

3. Analysis and Simulation

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 SRAM array. In order to reduce the scope of the project, we have not asked you to implement the circuits you would need to perform a write, and so you will not be simulating a full write operation. Instead, you will check functionality 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. Further hints and explanations on the set up for checking the SRAM's functionality will be provided in the discussion sessions.

2. **Performance**
   The performance of your design will be characterized by its total latency – i.e., the delay from one the addr or off signals transitioning to the data appearing at the output of the SRAM. You will also be asked to provide the critical path delay of the individual components in your design (i.e., the adder, the decoder, and the SRAM array itself).

3. **Area**
   There are two area measurements that need to be made for the final phase. The first is the area of your SRAM cell, and the second is the area of your complete design. In order to measure the areas, draw (with rulers) the smallest rectangular box that fits around your design. The area is simply the length times the width of the drawn rectangle.

4. **Power**
   You will also need to simulate the power consumption of your circuit. Since the power consumption of the design depends upon the frequency it operates, to compare all of the designs equally you will measure power with a fixed 50 MHz clock frequency in HSPICE. We will provide you with an HSPICE deck that extracts the power consumption of your design (please watch for an announcement on the availability of this deck too) – further details on using this deck will be provided in the discussion sessions.

5. **Noise Margins**
   If you have not changed the SRAM cell, then you already have the noise margins of your cell and no further work is necessary. If you have altered the cell, you will need to re-simulate you read and write noise margins as instructed in phase I.

4. 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.** Do not repeat information we already know. Use the templates provided on the web page (Word and PDF formats). Make sure to fill in the cover page and use the correct units. Turn in the reports for each phase in the homework drop box. In addition, mail an electronic version of your final report and the poster as a Word or PDF file to ee141-project@bwrc.eecs.berkeley.edu. You will also be asked to provide your final netlist.

4.1 Report for Phase III

The format of the final report and instructions on printing your poster will be provided by the end of the week.