1. Design of an H.264 Motion Estimation Engine – Background

Video compression is commonly implemented in hardware for modern portable applications. There are many video compression/decompression standards (e.g. MPEG2, MPEG4), and the one most commonly used today is H.264. Video compression typically implements motion estimation techniques to reduce the amount of incremental data that needs to be compressed/decompressed between the frames. Manipulation of frame data involves numerous repetitive operations on the data that is locally stored in a memory array. This term we will implement one architecture for manipulating data in the array, which is a part of motion estimation engine in H.264.

1.1. High level structure

A very abstract block diagram of the target block to be designed is shown in Figure 1. Each segment of the SRAM is labeled as ‘divided SRAM’ and consists of a number of memory cells (MCs), X- and Y-decoders. It is followed by a barrel shifter that ‘reshuffles’ the data before processing (by processing elements, PE’s). You will be designing two major parts of the architecture in Figure 1: (1) the SRAM array (shown in detail in Figure 2), and (2) the barrel shifter.

Here is a more detailed description of your tasks in the project:

1. **SRAM array:** You will need to design one SRAM segment in the course of this project.
2. **Memory cells:** You will be given the memory cell schematic and layout, that you will need to characterize.
3. **Address decoders:** There are two address decoders that need to be designed, one for X-access and one for the Y-access.
4. **Peripheral circuitry and array assembly:** You will design the necessary circuitry for precharging and multiplexing the bitlines to get an operational array.
5. **Barrel shifter:** You will add the barrel shifter to the design. An important component of the design will be to match the layout of barrel shifter to the memory array.
Figure 1: Block diagram of the segmented SRAM array for video encoding.
Figure 2: More detailed block diagram of the divided SRAM architecture.
2. Implementation and Constraints

The goal of this project is to design a functional, compact and fast SRAM array for use in high-performance or mobile video accelerator with a particular set of optimization criteria. The project will be completed in FOUR phases.

PHASE 1: Cell Characterization. Due Friday, October 21, at 10am.

In the first phase of the project, you are provided with a pre-designed SRAM cell. Characterize the cell stability by using Cadence to obtain an extracted netlist and HSPICE to perform simulations to get the read and write margins.

To obtain the SRAM cell:
- In Cadence, create a library “sram” linked to the gpdk090 90nm technology (see lab 2).
- Now in an x-terminal, go to the directory ~/ee141/sram/ (this assumes that the library you just created is in ~/ee141/sram) and type the following command:
```
cp –R ~/ee141/fall11/project/sram_cell/ .
```
- Go back to Cadence. You should see the SRAM cell design under the cell view sram_cell in your sram library.

Recalling that the wordline and bitlines are held at $V_{DD}$ during a read, Figure 3 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 $V_1$ (the inverter’s input) from 0 to $V_{DD}$ and measuring $V_2$ (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 HSPICE) 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 $V_2$ when $V_1$ is at $V_{DD}$) and provide that value in your report.

Next, you must find the write static noise margin. During a write, VDD is applied to the wordline, and the value to be written into the memory cell is driven onto the bitlines. The procedure to extract the write noise margin (WNM) of the cell was shown in the class. Again, the feedback
from the cross coupled inverters is broken, and the VTC of the “inverters” are measured – make sure to double check your setup. Note that in some cases WNM may be very large and hard to simulate. You do not have to calculate the WNM, but you should generate the butterfly plot (again using HSPICE) and graphically indicate the WNM. Is there a better way to measure the writeability of the cell? You should also measure the worst-case cell voltage during a write (which is found from measuring V₁ when V₂ is at 0V).

Finally, assemble a column of 32 cells, extract the netlist from the layout and simulate the read current.

**Prelab:** Schematic for read and write noise margin simulation (on paper).

**PHASE 2: Column Design. Due Friday, October 28, at 10am.**

In this phase, we will assemble one column of 32 SRAM cells, and add the peripheral circuitry for precharging, multiplexing along with the sense amplifier. We will test the functionality of the design.

**Prelab:** Sense amplifier design.

**PHASE 3: Design of 5-32 Memory Decoder. Due Friday, November 11, at 10am.**

The input loading of each of the true and complementary address lines is constrained to be less than 10fF. The output loading of the decoder is determined from the wordline loading of the SRAM array and the wireload. The length of the wordline can be determined from the horizontal dimension of the cell.

The decoding is performed in two phases: predecoding and the final row decoding similarly to Figure 2, with N=4. The predecoder drives the final decoders together with the wire that whose length equals the height of the memory array.

**Prelab:** hand design of the decoder, using the method of logical effort. The complete decoder in schematic and layout should be done in Cadence. Note that decoder pitch in layout has to match the core array pitch.

**PHASE 4: Design of Barrel Shifter/Array Completion. Due Friday, November 18, at 10am.**

This phase will finalize the project by matching the logic to the SRAM array.

**PROJECT POSTERS,** time and date TBD.

*In each phase of the design you will turn in a short report. A longer report, together with a poster presentation is due at the end of the semester.*
Physical and electrical specifications:

2.1. TECHNOLOGY: The design is to be implemented in a 90 nm CMOS process with 5 metal layers. You should only use up to 4 metal layers for the entire design. The SPICE library is located at /home/ff/ee141/MODELS/gpdk090_mos.sp and is the model we have been using all semester (TT_s1v). At a supply voltage of 1.2V, for hand calculations you can assume that $L=100\text{nm}$, $C_G = 1.91\text{fF}/\mu\text{m}$, $C_D = 0.77\text{fF}/\mu\text{m}$, $R_{\text{sqn}} = 9.43\text{k}\Omega/\square$, and $R_{\text{sqp}} = 22.32\text{k}\Omega/\square$.

2.2. POWER SUPPLY: You are free to choose any supply voltage and logic swing up to 1.2V. Make sure that you use the appropriate model when you perform any hand analysis.

2.3. PERFORMANCE METRIC: The propagation delay of your memory is defined as the time interval between the 50% point of the inputs and the 50% point of the worst-case output signal. Make sure you pick the worst-case condition and state EXPLICITLY in your report what that condition is!

2.4. POWER: Since everyone’s design will be capable of running at different frequencies, we will measure power consumption with a 100MHz clock. Further instructions on how to run simulations to measure power will be provided in Phase 3.

2.5. AREA: The area is defined as the smallest rectangular box that can be drawn around the design. Since the circuit must interface with the rest of the chip, all inputs and outputs must be accessible from the boundary of the layout.

2.6. NAMING CONVENTIONS: Please use the signal naming consistent with Figure 2.

2.7. REGISTERS: You don’t need to use any registers in this design.

2.8. CLOCKS: You can use as many clocks as you would like for this design. You will need at least one clock to enable/disable the decoder outputs and precharge the bitlines. Remember that the load capacitance of any clock you use should be included in the power analysis.

2.9. $V_{\text{OH}}$, $V_{\text{OL}}$, NOISE MARGINS: You are free to choose your logic swing in the decoder. The noise margins should be at least 10% of the supply voltage.

2.10. RISE AND FALL TIMES: All input signals have rise and fall times of 100 ps. The rise and fall times of the output signals (10% to 90%) should not exceed 400ps (unless otherwise noted).

2.11. LOAD CAPACITANCE: Each SRAM must drive a 15fF load.

2.12. INPUT CAPACITANCE: For the third phase of the project, the maximum capacitance you can place on each polarity of the address inputs (true and complementary) is 10fF.

2.13. EXTRACTION: When performing extraction on your layout, you must use Assura to do parasitic extraction following the directions provided on the website.
3. Simulation

You should always begin your designs by hand analyzing the circuits. Having done this, you should also use HSPICE to simulate the design and prove that it functions correctly. Keep in mind that you will need to determine the input pattern that causes the worst-case propagation delay or energy consumption.

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 all of your graphs are easily readable, have clearly labeled axes, a meaningful title, and have legends where needed. All figures should have captions that clearly describe what is shown in the figure. 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 1

The total report should not contain more than four pages. You are not allowed to add any other sheets. The organization of the report should be based on the following outline:

- **Cover page:** Names, summary of simulated SRAM cell parameters, wordline capacitance, read current.
- **Page 1:** Simulation of the static noise margins.
- **Page 2:** SRAM column layout and schematic.

Remember, a good report is like a good circuit: it should perform its function (convey information) in the smallest possible area with the least delay and energy (to the reader) possible.

The quality of the report is an important (major) part of the grade!

The total project grade is divided into four parts: four phases of the project and the final report/poster. Each carries 20% of the total grade.

For each of the four phases, the grade will be divided as follows:

- 5% Prelab
- 30% Approach and correctness
- 30% Results
- 30% Report
- 5% Creativity