General instructions

Based on the material you learned during the term, you need to complete a final project. This is a group project with groups of 2 or 3 students, depending on the number of students in the class. You will have multiple projects options to choose from, and **after the instructor's approval**, you can start working on the project with your team. The grade for each team member will derive 50% from the team members and 50% from the instructor. This means that along with the submission of your project, each student needs to provide a grade for their team members, followed with a short paragraph to justify the grade. The instructor reserves the right to reject the grade provided by the students if there are concerns that the grade does not reflect the students' work. The members for each team will be selected randomly through the CANVAS' during class on Feb 18th. After the announcement of the groups, each student is responsible to contact their group members and schedule their time to work on the project. The projects or any other issues should be announced/decided until Feb 24th. Changes can be done later, but I suggest not to, based on the limited time that may be left.

The length of the report is unlimited. That means that you have to write enough to explain your subject. The report should at least have an introduction, background, main body, conclusion and the solutions to the exercises or code if you have any. Try to avoid repetitions in your report. Explaining something again and again will cause losing points. Be clear, precise and at the right place in your reports. Explaining a concept after is been already introduced to explain another concept will be considered as a mistake. Organize your ideas and how these are structured one after another before you start writing, so your report flows naturally.

Attention to the detail will be taken into account. That means than the report should be coherent. Be sure to devote some time as a group to decide about the formatting details to accomplish that. See the "Formatting guidelines" and the "Wrong example" sections for more information. In addition, this is an academic document, that means academic language will be expected.

Format guidelines

Project should have a front cover page and a back cover page. Front cover page should include the subject of the project, the name of the members in your group, the class' name, the university and the term. You can also add graphics on your front/back cover pages if you like. Your report should also contain a contents page where you will list the parts of the project. You can use the project template document to start with.

Guidelines for the project's body formatting.

The project should be grammatically correct with no spelling mistakes.

Font:

- Use Times New Roman 12 point for body content.
- Major headings may be 12 to 16 points.
- Use bold and capitalization to distinguish between heading levels.

Line Spacing:

- Use single spaced block paragraphs with 5 spaces indentations.
- Leave one blank line between paragraphs.
- There should be no default extra white space between paragraphs.

Guidelines for integrating visuals into your project

Visuals must be at the correct size so their meaning is clear. Since there is no page limit, don't oversize visuals just to fill pages. Distorted visuals are also not accepted. If you have graphs in your report, it's better if you add a page break and change the orientation to landscape so the graph is clear.

Labels and captions:

e.g.: Figure 1. Multiple views of Hale Tower.

Rules for labels and captions:

- Table is used to label tables.
- Figure is used to label illustrations, drawings, graphs, charts, etc.
- Captions may be phrases, sentences, or paragraphs.
- Include a period after a caption.
- Tables are labeled above the table. Figures are labeled below the figure.

In-text references:

e.g. These three goals are illustrated in Figure 4.

Rules for referencing visuals in text:

- Capitalize Figure and Table in the text, but do not use bold font.
- Use the same label within text and with visual.

e.g. **WRONG:** These measurements are illustrated in the graph below.
RIGHT: These measurements are illustrated in Figure 6.3.

- Whenever possible, in-text the reference should appear before the visual and as close to the visual as possible.
- The reference may be integrated into the sentence, as in the example above.
- OR, the label may be included in parentheses after the reference

  e.g. The top bolt is larger than the lower bolts (Figure 2).

- The reference may be integrated into the sentence, as in the example above.

Citations and references:

Your report and your presentation should contain a "References" page. All the sources you will use in the project, as well as borrowed pictures should be referenced in that page. See the "IEEE citation reference" for more information how to properly cite sourses.

Wrong project example

This is an example of a wrong project report. It doesn't contain front and back page, contents, and it also contains various formatting mistakes. Be sure you read all the instructions and that you understand them before start.

Introduction

As the technology advances, devices in CMOS Technology have been scaled down aggressively with each technology generation to achieve a higher integration density and performance [1]. However, the leakage current problem has increased drastically due to technology scaling and since modern microprocessors employ large sizes of caches, SRAM leakage exacerbates the total chip power consumption.

In addition, in Nano-scaled CMOS technology, supply voltage and nodal capacitance is reduced. Thus, low energy particles can flip the values in SRAM cells, making cache memory cells more sensitive to atmospheric neutrons and alpha particles.

Furthermore, the mismatch in the strength between transistors of conventional SRAM cell due to process variations can results in failure during read operation [2, 3]. Therefore, conventional six transistor SRAM cell (CV 6T SRAM cell) has a poor stability at very small feature sizes. Leakage current, read static noise-margin (SNM) and soft error rate (SER) of SRAM cell are three important parameters in designing SRAM cells for cache memory in Nano-scaled CMOS technology.

In the following project, we will present the latest high level technologies on cache designs like cache partitioning and SMART caches, as well as circuit level solutions like power gating and asymmetric SRAM design that address the recent issues on the SRAM cells.

Background

*If you decide that you will leave 1 line between title and text, stay with that.

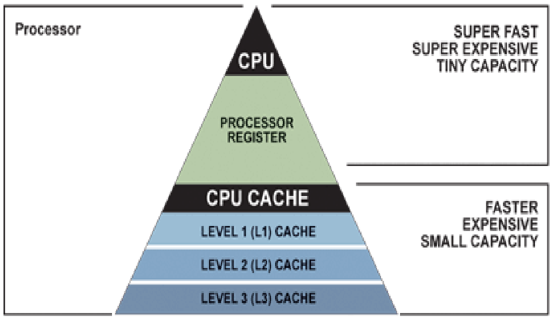
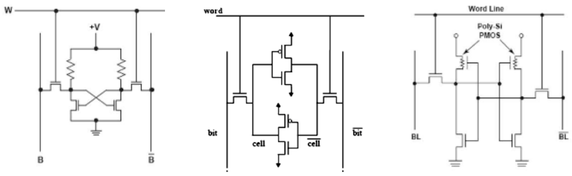


Figure 1. Memory hierarchy in a modern processor.

At early 1980's, microprocessor's clock speeds started to increase, but memory access times improved at a slower pace. As this gap was growing, it became obvious that a new type of fast memory was needed to bridge the gap. Cache memory was first introduced on 1985 in personal computers with the Intel's 386 architecture. This helped to bridge the gap temporary but few years later the problem became current again. Thus, manufactures started introducing different levels of cache at 1989 and form the memory hierarchy as we know it today (Figure 1) [4].

The basic memory cell

On the modern multicore processors multiple cache memory is needed in order to improve the efficiency of the system. In order to understand the functionality of the multiport cache, first we have to understand the memory cell. The basic memory cell designs for memory is the four transistor (4T) cell, the six transistor cell (6T) and the thin film transistor cell (TFT) (Figure 2).



- If you decide that the figures will be center alligned, stay with that.

Figure 2. The 4T (Left), 6T (Center) and TFT (Right) memory cells.

The 4T cell is constructed from four transistors and two poly-silicon resistors. This memory cell has a smaller size from the 6T cell because the resistors can be placed on top of the transistors. This design has some limitations in compare to the 6T design and those are:

1. High standby current due to the resistors.
2. It is sensitive to noises due to the high resistance
3. It is slower than the 6T cell.

The 6T memory cell is constructed out of 6 transistors from which four are in static inverter circuit forming a latch and the other two are connected to the word line and the bit lines and operate as gates. The 6T cell has the bigger size of all the cell types but it offers better speed, noise insulation and standby power consumption.

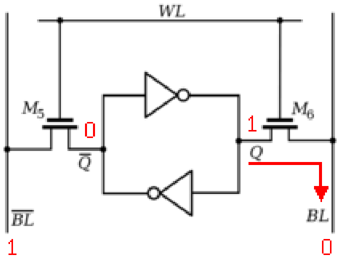
Finally, the TFT cell is constructed out of four transistors and two poly-silicon transistors. This allows to the cell to have lower standby consumption than the 4T cell but its performance is lower than the 6T cell [5].

Other implementations of memory cells are constructed from seven, eight, ten or more transistors trying to add more ports to the memory cell or reducing issues like the leakage current we will analyze later on this paper.

Although the devices around the basic memory cell depend on the implementation there are some basic components that are standard for its basic operations, hold, read and write. These components are the pre-charge circuit with the equalizer, the sense amplifier, the write driver and the decoder [6]. The pre-charge circuit, equalizer, write driver and sense amplifier are all connected to the bit lines and they are used for the read and write operations. The pre- charge circuit is used to charge the bit lines, the equalizer to minimize the voltage difference between them, the write driver to transfer the data to the bit lines and the sense amplifier is a differential amplifier used for reading the value in the memory cell. The decoder is a circuit that decodes the address that comes from the requester and use it to access the specific memory location in the memory array.

During the write operation, the write driver charges the bit lines with the data we want to store. According to which side of the latch holds the low voltage, the write driver charges that line high and the side that holds the high voltage is charged low. Then the word line is charged and the two transistors that are connected to the word line operating as gates, open so the voltage flows from the bit line that is low charged and causes the cell to change state (Figure 3).

Figure 3. Write operation in the memory cell. * Labels for figures go below.



On the other hand, during the read operation, the pre-charge circuit charges both bit lines high and the equalizer ensures that the voltage difference between them is minimum. In different case the sense amplifier will read wrong data that are coming from the charging of the two bit lines and not from the memory cell. Then the word line is charged high, the gates open and then bit line that is connected to the low voltage side is getting discharged creating a voltage difference to the bit lines thus the sense amplifier produces the value to its output (Figure 4).

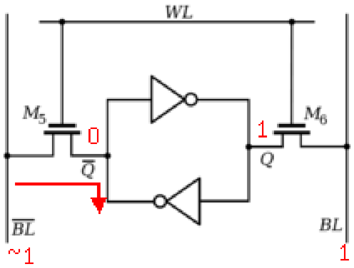


Figure 4. Read operation in the memory cell.

During the hold operation the gate transistors remain close and the memory cell keeps its value unchanged.

Multiport SRAM designs * If you choose to align left the text, don't arbitrary change it to justify and back.

Now that we have analyzed the basic SRAM memory cell and its operations we can proceed to the multiport implementations and their characteristics. The need that drove to the production of multiport caches is the fact that parallel execution due to multicore processors, system on chips and DSPs can have a huge benefit from parallel reading and writing on memories. Manufacturers design multiport caches in various ways. The two major categories that multiport caches are separated are the synchronous and asynchronous caches.

Synchronous caches are receiving a clock signal from an external clock device, which usually takes into consideration the processor's clock and they perform the write and read operations at the edges, rising or falling, of that clock. Asynchronous caches operate without any clock device and they perform the operations whenever they have instructions feeding their pins.

The differences between the two implementations are that the asynchronous caches lack performance because the "wait states" they introduce due to the fact they don't consider the processor's clock, however, they are easier to implement. On the opposite side, synchronous caches have more complicated interface due to the clocking considerations but they have higher bandwidth [7] [8].

Apart from the timing categorization of the multiport caches the most common technologies that are used to add ports on the memory cell are:

- True multiport caches.
- Banked multiport caches.
- Multi-pumping multiport caches.
- Stream buffered multiport caches.
- Cached multiport caches.

* If you decided to use numbers
for sub-lists then stay with it. If you also decided
that those lists will have indentation don't change it

The true multiport cache design offers a separate read and write ports to each agent that is connected to it. These implementations are used when the access latency is critical. This design has an area penalty because of the transistors that are used for each port on each cell. On figure 4 we can see how these different ports can be implemented and understand how the size increases for each port addition. A derivative of the true multiport design is the replicated state multiport cell which uses two memory cells which are identical with one write port and two read ports. This design is used when more read ports are important [9].

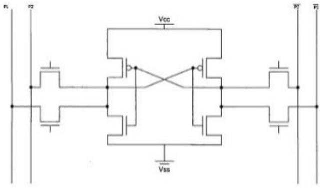


Figure 5. A one read – one write multiport cell design.

The banked multiport design divides the cache memory into several memory banks which are controlled by an arbitrator circuit. This memory design is used when we have multiple agents to be connected connect to the memory on different banks but only one each time to each bank. This design has a drawback from the arbitrator circuit delay (Figure 6) [10].

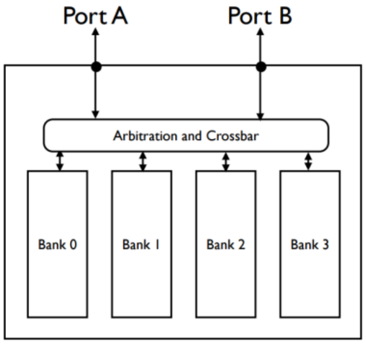


Figure 6. The banked multiport cache design.

The multi-pumping design uses a basic one read – one write cell with additional temporary registers to increase the ports of the cell. Then by an internal clock processes the requests for each port. This design is limited due to the fact that the internal clock degrades the processor's clock (Figure 7) [9].

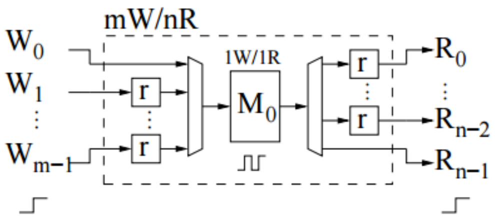


Figure 7. The multi-pumping cache design.

The stream buffered design uses a large cache memory with a wide port, an arbitrator circuit and streams each of its ports. Each stream is assigned to each agent that is connected to the memory and the access is organized by the arbitrator circuit. Similarly, with the banked design this design suffers from the arbitrator circuit delay (Figure 8) [10].

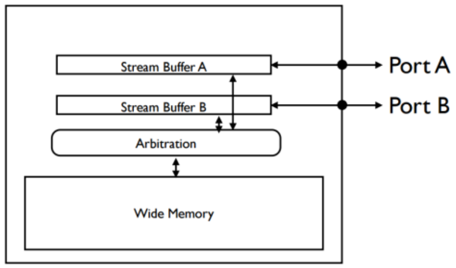


Figure 8. The stream buffered cache design.

The cached multiport design is similar to the stream buffered design. It uses a large memory and the arbitrator circuit but at the higher level each of its ports has cache buffers. This design suffers from size overhead, arbitrator delay and cache coherence complexity (Figure 9) 10].

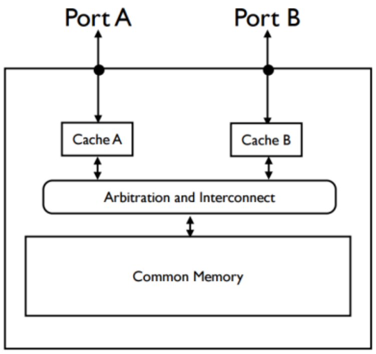


Figure 9. The stream buffered cache design.

* Having a figure alone on one page makes readability difficult, try to avoid it.

List of Acronyms

ILP:	Instruction Level Parallelism
CMOS:	Complementary Metal-Oxide Semiconductor
SRAM:	Static Random Access Memory
CV 6T:	Conventional Six Transistor
SNM:	Static-Noise-Margin
SER:	Soft Error Rate
CPU:	Central Processing Unit
L1 (2, 3):	Level 1, Level 2, Level 3
4T:	Four Transistor
6T:	Six Transistor
TFT:	Thin Film Transistor
BL:	Bit Line
$\overline{\mathbf{BL}}$:	Bit Line Bar
WL:	Word Line
DSP:	Digital Signal Processor

References

1. A. Agarwal, C. H. Kim, S. Mukhopadhyay, and K. Roy, “*Leakage in nano-scale technologies: mechanisms, impact and design considerations*”. Proceeding of the 41st Design Automation Conference, 2004, pp. 6-11.

2. S. Mukhopadhyay, H. Mahmoodi, and K. Roy, “*Modeling of Failure Probability and Statistical Design of SRAM Array for Yield Enhancement in Nanoscaled CMOS*”. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 24 (12) (2005) 1859-1880.

3. K. Takeda, Y. Hagihara, Y. Aimoto, M. Nomura, Y. Nakazawa, T. Ishii, H. Kobatake, “*A read-static-noise-margin-free SRAM cell for low-VDD and high-speed applications*”. IEEE Journal of Solid-State Circuits, 41(1) (2006) 113-121.

4. Wikipedia, The Free Encyclopedia. Wikipedia, https://www.wikipedia.org/. "*CPU cache*". Accessed on 15 Feb. 2016. Web. 22 Feb. 2016.

5. http://www.eeherald.com/section/design-guide/esmod15.html. Accessed on Feb. 10 of 2016.

6. Sunil K. Lakkakula, “*Vlsi design and comparison of bank memory with multiport memory cell versus conventional multiport and multibank SRAM memory*”. Oklahoma State University, 2009.

7. http://www.cypress.com/knowledge-base-article/comparision-between-asynchronous-and-synchronous-dual-port-rams. Accessed on Feb. 12 of 2016.

8. https://www.cs.umd.edu/~meesh/cmsc411/website/projects/ramguide/cache/cache.html#whatwait. Accessed on Feb. 12 of 2016.

9. https://tspace.library.utoronto.ca/bitstream/1807/18801/1/LaForest_Charles_E_200911_MASc_thesis.pdf. Accessed on Feb. 14 of 2016.

10. https://inst.eecs.berkeley.edu/~cs294-88/sp13/lectures/patterns2.pdf. Accessed on Feb. 14 of 2016.