Development of a Low-Power SRAM Compiler

Meenatchi Jagasivamani[1] and Dong Sam Ha[2]
[1]Intel, 5000 W. Chandler Blvd., Mailstop CH6-210, Chandler, AZ 85226, USA
[2] Virginia Tech VLSI for Telecommunications Lab
Dept. of Electrical and Computer Eng., Virginia Tech, Blacksburg, Virginia 24061,USA
Phone: +1-540-231-4942 Fax: +1-540-231-3362
E-mail: meenatchi.jagasivamani@intel.com, ha@vt.edu

ABSTRACT

With the proliferation of portable consumer electronics, power-consumption becomes a key design criterion, while the speed of memories is still the bottleneck for high-speed applications. In this paper, we discuss the development of an SRAM compiler with the capability to choose between a low-power and a high-speed SRAM. Experimental results show that the low-power version of our 1-kB SRAM can function at a minimum operating voltage of 2.1 V and dissipates 17.4 mW of average power at 20 MHz.

I. INTRODUCTION

SRAMs are a critical component in both hand-held devices and embedded processors and play a key role in power-consumption [1-2]. An important factor in improving the performance of a system is to use an optimum sized SRAM as it avoids waste of area, power, and speed.

Trends show that low power design techniques are becoming more important in the current industry. There are numerous ways to reduce the power dissipation at the cost of area and/or speed, both in the cell and architectural levels [3]. In this paper we discuss the development of an SRAM compiler for automating layouts of memory elements in the ASIC environment. Our SRAM compiler employs array partitioning to reduce power dissipation [4].

Our compiler generates an SRAM layout of a given size for high-speed or low-power dissipation. Leaf cells of the SRAM core (a 6T SRAM cell), sense amplifiers, decoders, and other peripheral circuitry were custom laid-out using Cadence Virtuoso. Cadence SKILL code was used to instantiate leaf-cells and to make necessary routings at appropriate coordinates. Simulations were performed using Avanti Star-HSPICE. TSMC 0.35 μm process technology was used for the design.

The paper is organized as follows. In Section II, a brief overview of the architecture and the architectural-level power reducing technique, array partitioning, used for the low-power version [4] is discussed. Section III discusses implementation of the SRAM compiler using Cadence SKILL code. Section IV presents the simulation results on the performance of the compiler and discusses notable trends. Section V concludes the paper.

II. SRAM ARCHITECTURE

An asynchronous SRAM compiler will be implemented by first performing custom layout of the basic components of the SRAM, such as the SRAM core, bit-line conditioning circuits, the sense amplifier, and the address decoders in a compact manner. Figure 1 gives the SRAM structure with the major components.

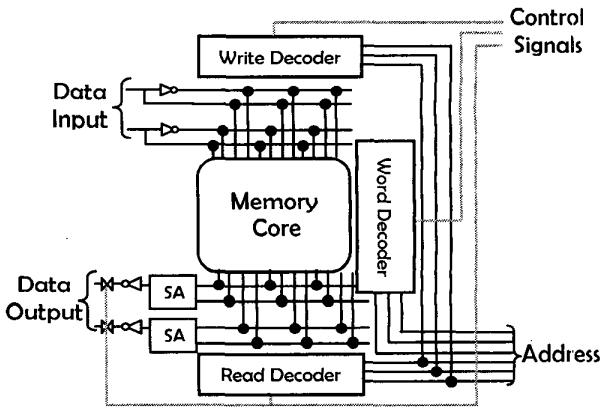


Figure 1 – SRAM Structure

A basic 6-transistor SRAM cell containing a latch is used for the SRAM core. The sense-amplifier is implemented as a cross-coupled amplifier, which is shared among multiple columns, as shown in Figure 2. An NFET is used

0-7803-6685-9/01/$10.00©2001 IEEE

to separate columns from the bus, which is amplified by a single sense amplifier. The NFETs are tied to their column enable signal so that only one column drives the sense amp at a time. This can be contrasted with having a sense amp for each column, which increases both the number of sense amps needed, as well as having a larger drive-load requirement for each sense-amp because of long output bus lines.

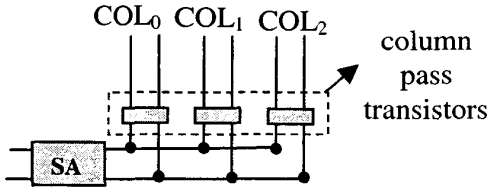


Figure 2 – Sense Amplifier Architecture

Address decoders are implemented in tree-decoder fashion [3]. Pull-up buffers comprised of weak PFETs and inverter chains are used at the output of the decoders. The SRAM array's columns are logically divided into word-size blocks. This prevents overlap of data blocks between rows and simplifies column decoding.

An array-partitioning scheme [4] was adopted in our SRAM compiler to reduce the power dissipation. Figure 3 illustrates the concept. The SRAM array is split into 4 subblocks and a controller is used to select one of the blocks from which the current data is being accessed.

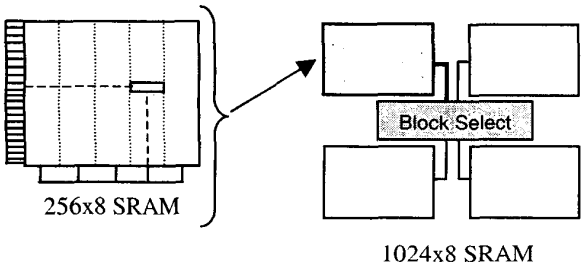


Figure 3 – Array Partitioned SRAM

The array-partitioning technique reduces dynamic power dissipation by reducing the total switching capacitances of the bit and word lines. By splitting the array into four blocks, the capacitance of the word-line and bit-line that switch during the read or write operation reduced to half. Also, the access time just for individual blocks will be smaller when compared to single-partitioned array. For our compiler, a fully functioning SRAM is used for the each of the blocks.

III. SKILL CODE IMPLEMENTATION

Cadence's SKILL functions are used to instantiate the leaf cells to form an SRAM array of the given size. Figure 4, shows the structure of the SRAM Compiler. The top block, *sram,* calls other modules to generate the SRAM circuit containing core array and periphery circuitry.

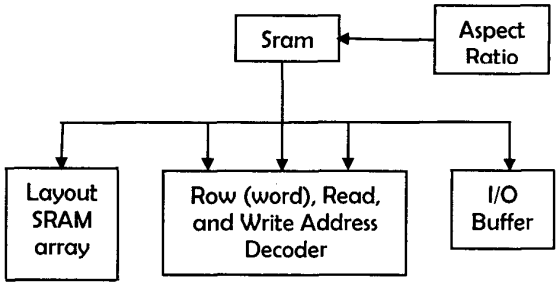


Figure 4 – SRAM Compiler Structure

The steps for the SKILL code compiler are as follows:

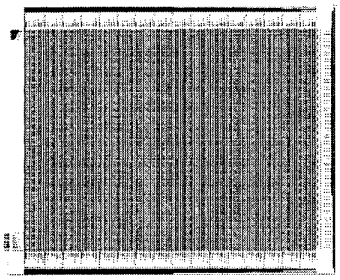
1. Calculation of the number of rows and columns for aspect ratio close to 1
   It is undesirable to have extremely long or wide arrays. We aim for a square shape.
2. Instantiation of leaf-cells into an array
   Based on the numbers of rows and columns the SRAM core is instantiated to form the array and peripheral circuitry is added subsequently.
3. Routing the necessary I/O lines to cells from primary inputs.
4. Generation of decoder layouts.
5. Incorporating array-partitioning and block selection for the low-power version.
   We add control circuits to isolate subblocks that are not being used for the current access. This feature is not performed for the high-speed version.

A separate SKILL module is created for the array-partitioning part. To partition the array, a memory that is ¼ the requested size is first generated using the compiler. Next, the block-select is implemented using the tree-decoders used for

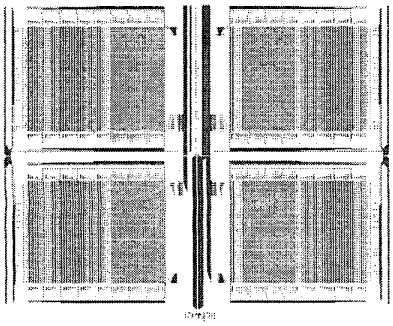
address decoding and necessary routing is performed between blocks.

V. RESULTS

A layout of a 1-kB SRAM was generated and SPICE simulations were used to evaluate performance. Performance was compared on three criteria – area, speed, and power. Layouts for both partitioned-array (for low-power) and single-array (for high speed) were generated for comparison of the performance and are shown in Figure 5.



(a) Non-Partitioned Array



(b) Partitioned Array

Figure 5: Layouts of a 1 KB SRAM

The performance of the non-partitioned 1 KB SRAM is as follows:

- Core area = 700 μm x 580 μm = 0.406 mm^2
- Transistor count = 50,513
- Access time is 15.0 ns operating at 3.3V.

In contrast, the performance of the partitioned SRAM is as follows.

- Area = 860 μm x 730 μm = 0.606 mm^2
- Transistor count = 52,157
- Access time is 21.8 ns operating at 3.3V

The extra circuitry for partitioning the memory incurs larger area. Although a partitioned block will have less access time than a single-array block, there is added delay caused by routing to and from the block-select, the delay for block-select, and time for the sense-amp to drive data onto the primary output. This trend of increased delay for the partitioned SRAM decreases with increasing size (25% less difference from 512-B to 1-kB).

Table 1 – Power Measurements (mW)

(mW)	256x8		512x8		1024x8	
	S	P	S	P	S	P
Dynamic	31.11	25.86	61.15	30.00	79.21	41.54
Ratio	0.83		0.49		0.52	
Static	0.65	3.46	0.73	3.62	0.96	3.57
Ratio	5.32		4.95		3.72	
Average	24.68	21.39	48.08	24.16	66.59	36.83
Ratio	0.87		0.50		0.55	

Table 1 gives the power measurements at the operating voltage of 3.3 V for the two types of SRAMs (S for Single-array and P for Partitioned) and for three different sizes. Dynamic power is measured as the power dissipated during the time when the memory is accessed. Static power dissipation is measured during standby mode, when all nodes should be at a steady value. The power dissipated during the entire simulation is taken to be the average power dissipated. The ratio indicates the difference between Static and Partitioned array memory.

The results show that the array partition for the 1KB SRAM reduces the dynamic power dissipation by 45%, as the switched capacitance on the bit and word lines are halved. The static power dissipation actually increases with partitioned array due to increased overhead, especially ones that contain a resistive load. However, the overall average power dissipation is still reduced as it is dominated by dynamic power dissipation.

An interesting trend to note with these results is that the power saving is not linear to its size. The average power dissipation for a non-partitioned 512x8 SRAM is double that of the non-partitioned 256x8 SRAM. However, the trend does not hold for the partitioned SRAMs. The reason for this nonlinearity can be explained by inspecting the SRAM shape/aspect ratio as shown in Figure 6.

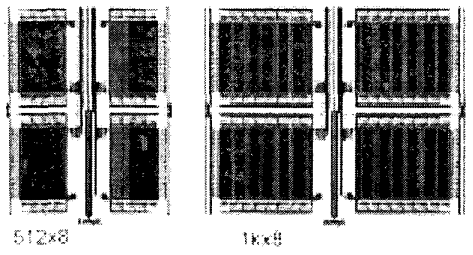


Figure 6 – Aspect Ratio Comparison

The array of the partitioned 512x8 SRAM is thinner (aspect ratio < 1) than that for the 1KB size (aspect ratio > 1). This is caused by the requirement that the shape be close to a square. Since the word-lines use the polysilicon layer, while the bit-lines use a less resistive metal layer, it is desirable to make word-lines proportionally shorter than the bit-lines, that is the aspect ratio of the core should be less than 1. This indicates that the shape of the SRAM core is a critical parameter in the final performance of the SRAM. Further analysis is necessary to determine the optimum word-line to bit-line ratio for an SRAM array.

VI. CONCLUSION

Our SRAM compiler generates two different types of SRAMs, which can be used as embedded ASICs. Partitioned SRAMs save substantial power over non-partitioned SRAMs at the cost of speed and area. Our partitioned 1-kB SRAM reduces the dynamic power dissipation by 48% and the total power dissipation by 45 %. However, the access time for the partitioned SRAM is slower than the non-partitioned SRAM by 31%, and it increases the area by 33%. We can increase power savings by lowering the supply voltage to 2.1 V, where the partitioned SRAM reduces the average power dissipation to 17.39 mW (from 36.83 mW at 3.3 V) at the speed of 20 MHz.

REFERENCES

1. R. Baker, H. Li, and D. Boyce, *CMOS: Circuit Design, Layout, and Simulation*, New York, IEEE Press, 1999.
2. N. Weste and K. Eshraghian, *Principles of CMOS VLSI Design: A Systems Perspective*, New York, Addison-Wesley, 1993.
3. J. Rabaey, *Digital Integrated Circuits*, New York, Prentice Hall, 1996.
4. J. Caravella, "A Low Voltage SRAM For Embedded Applications," *IEEE Journal of Solid-State Circuits*, Vol. 32, No. 3, pp. 428-432, Mar. 1997.
5. H. Tran, "Demonstration of 5T SRAM and 6T Dual-Port RAM Cell Arrays," 1996 Symposium on VLSI Circuits, pp 68-69, June 1996.
6. A. Karandikar and K. Parhi, "Low Power SRAM Design using Hierarchial Divided Bit-Line Approach," *Proceedings of the International Conference on Computer Design*, pp. 82-88, Oct. 1998.
7. J. Wang, P. Yang, and W. Tseng, "Low-Power Embedded SRAM Macros with Current-Mode Read/Write Operations," Proceedings of the International Symposium on Low Power Electronics and Design, Digest of Technical Papers, pp. 282-287, 1998.
8. H. Morimura, S. Shigematsu, and S. Konaka, "A Shared-Bitline SRAM Cell Architecture for 1-V Ultra Low-Power Word-Bit Configurable Macrocells," International Symposium on Low-Power Design & Electronics, pp. 12-17, 1999.
9. H. Morimura and N. Shibata, "A 1-V 1-Mb SRAM for Portable Equipment," International Symposium on Low-Power Design & Electronics, pp. 61-66, 1996.
10. J. Alowersson and P. Andersson, "SRAM Cells for Low-Power Write in Buffer Memories," Procedings of the 1995 Symposium on Low Power Electronics, San Jose, CA, October 9-11, 1995.
11. M. Izumikawa, H. Igura, K. Furuta, H. Ito, H. Wakabayashi, K. Nakajima, T. Mogami, T. Horiuchi, and M. Yamashina, "A 0.25 μm CMOS 0.9 V 100-MHz DSP Core," IEEE Journal of Solid-State Circuits, vol. 32, pp. 52-61, Jan. 1997.
12. H. Nambu, K. Kanetani, K. Yamasaki, K. Higeta, M. Usami, Y. Fujimura, K. Ando, T. Kusunoki, K. Yamaguchi, and N. Homma, "A 1.8-ns Access, 550-MHz, 4.5-Mb CMOS SRAM," IEEE Journal of Solid-State Circuits, vol. 33, pp. 1650-1656, no. 11, November 1998.