An Accurate Power and Timing Modeling Technique Applied To A Low-Power ROM Compiler

A. Turier$^{\diamond}$, L. Ben Ammar$^{\diamond}$ and A. Amara*
$^{\diamond}$ATMEL-ES2, Zone industrielle, 13106 Rousset Cedex, France
*ISEP/LIP6, 21 rue d'Assas, 75270 Paris Cedex 06, France
email : aturier@es2.fr

Abstract

Large memories require too much computing resources to be fully extracted and simulated. In this paper we present an accurate power and timing modeling technique that significantly reduces characterization resources. The model is based on current controlled generators, and can be used in several regular structures. Low-power memories are more easily characterized using this technique, since they are block partitioned. Moreover, this modeling technique is fully parameterable. It is very useful for developing and optimizing memory compilers efficiently. It has been successfully applied to characterize and improve performances of a low-power ROM compiler.

1 Introduction

Since entering the era of 'system on a chip', large embedded memories with high throughputs and efficient power management architectures have become more and more necessary [1]. Such modules can be generated by dedicated compilers that enable the system designer to choose the appropriate memory features, for example, the geometrical aspect ratio, according to the timing and power requirements : the ratio between the number of rows and columns can highly affect the resulting performances. Thus, several electrical simulations must be performed to characterize each possible configuration. Since high performance memory cells are very complex, the number of different cells that can be generated by the compiler is huge. So, a full characterization of the compiler is not easy and requires a lot of computing resources. Moreover, in some cases, Spice simulations can not be performed, due to the huge number of devices in the extracted netlists. One solution is to use faster electrical simulators that accept bigger netlists, such as TimeMill and PowerMill simulators [2]. Unfortunately, these simulators are not as accurate as the Spice one and in some cases converge with difficulty.

The objective of our approach is to reduce the number of devices in the extracted netlists and, for accuracy, to perform simulations using Spice. Reduction of netlists can be achieved by taking advantage from the regularity of the circuit. It consists of representing a little but significant part of the circuit and simply modeling the remaining parts. Large and low-power memories with several identical memory blocks can be efficiently modeled using this technique.

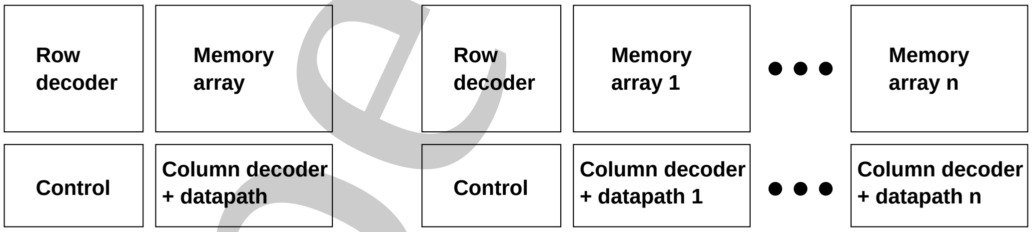
In this paper we describe the suggested method, when applied to a low-power ROM compiler. In order to locate regularities in the memory structure, we introduce in section 2 the architecture of the low-power ROMs that may be generated by the compiler. In section 3 we describe the modeling concept and give models of the different memory parts. Simulation results of some modeled ROM parts corresponding to different configurations are presented in section 4 and compared to those obtained with the corresponding full extracted ROM parts.

2 Architecture of the low-power ROMs

The structure of embedded memory blocks is well known [3, page 556]. Conventional ROMs have a similar structure with a control logic, a row decoder, a column decoder + datapath and a memory array (Figure 1(a)). Large memories using such a structure are high density but low performance cells. In order to improve delay and power features, we implement a multi-block architecture with a selective precharge circuit and a pipe-lined decoding scheme.

2.1 Multi-block partitioning

As shown in figure 1(b), compiled memories are divided into several blocks. All bits of a given word are in the same block, so during a read cycle, only one block needs to be activated, while the others can be idle. To determine which block must be activated, we decode the appropriate bits of the address word by using a block decoder (Z-decoder). The number of blocks (Memory array and Column decoder + datapath) increases with the total capacity of the memory.



(a) conventional structure (b) multi-block structure

Figure 1: memory block diagram

For the word-line activation (figure 2), we use the concept of divided wordline [4] : The local word line of each block is activated by the global word line and the block selection signal. Both the selection delay and the power consumption are reduced, since the capacitance of the global word line decreases.

In our ROMs, this concept has been extended to all the control lines of the datapath such as multiplexer lines, precharge lines, sensing lines and output buffer lines : global buses go through each of the different blocks and a signal from the Z-decoder enables the right block to be activated. Thus, only one block datapath has dynamic power consumption during a read access.

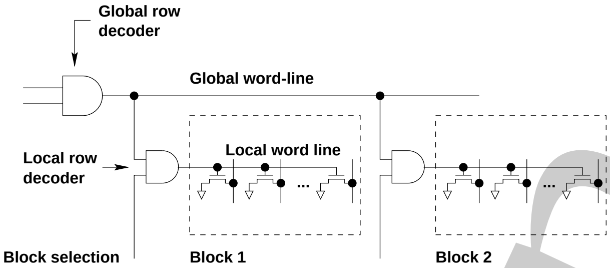


Figure 2: Divided Word-Line concept

2.2 Selective precharge

During the precharge phase to vdd, large capacitances attached to the bit-lines are being switched. If we precharge all the bit-lines of the memory, unread bit-lines can be discharged, if programmed to zero, because the row NMOS devices are activated by the same word-line (figure 3 : black circles represent NMOS devices). The selective precharge consists of precharging only the necessary bit-lines that will be read. In the best case, the number of precharged bit-lines can be reduced to the number of bits per word. This technique achieves an important improvement of the power consumption. The precharge selection can be simplified by using the same decoder for the bit-lines multiplexer and the precharge parts. In order to optimize speed and area, we place the precharge part just above the multiplexer and not on the top of the memory, as we can see in many figures [5].

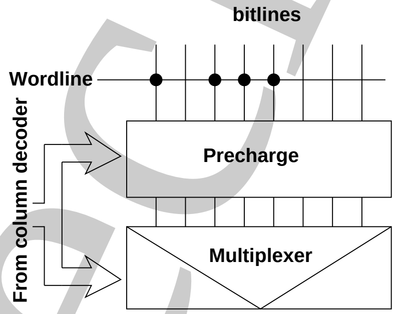


Figure 3: Selective precharge

2.3 Pipe-lined decoding scheme

The drawback of the selective precharge is the increase of the access time, since the precharge phase must be performed just before the read phase. The selective precharge implies knowing which bit-lines are about to be read before starting the precharge cycle. To limit the effect of this drawback on the access time, we use a 2 stage pipe-line decoding scheme (figure 4 : Xdec and Ydec are respectively row decoder and column decoder) : the first stage performs the address decoding; the second stage precharges and then reads the selected bit-lines. When Ck is low, all the decoders are enabled and when Ck goes high the addresses can change without affecting the selected lines.

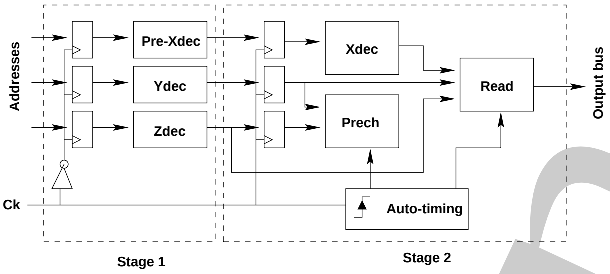


Figure 4: Pipe-lined decoders

3 Modeling techniques

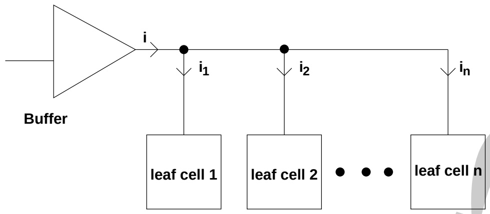
The ROM compiler is able to generate hundreds of different configurations, depending on the parameters entered by the system designer [6]. In order to provide data-sheets and timing / power views for each configuration, we need to run several accurate electrical simulations. With large memories, generated netlists are huge and simulations are very time consuming. Many parts of the design are repetitive, for example memory cells and output cells. Moreover, low-power memories are often splitted into identical blocks. So we can use only one part for simulations and model the remaining regular parts. Thus, netlist size and simulation time are significantly reduced, and Spice simulator can be used to perform accurate timing and power characterization.

3.1 Concept introduction

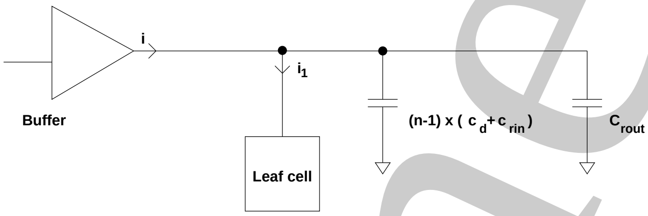
As shown in figure 5(a), in regular architectures, at a given hierarchical level, tree structures with identical leaf-cells are often used. The total capacitance driven by the buffer is equal to the sum of the n leaf-cell device capacitances ($C_d = n \times c_d$), the routing capacitance inside the n leaf-cells ($C_{rin} = n \times c_{rin}$) and the possible routing capacitance outside the n leaf-cells ($C_{rout} = n \times c_{rout}$). For power and timing characterization it is useless to represent all the leaf-cells. Only one leaf-cell is necessary, but we must take into account the capacitances of the non-represented leaf-cells. One solution consists of modeling the non-represented capacitances by a constant value multiplied by the number of non-represented leaf-cells, plus the total external routing capacitance : $(n-1) \times (c_{rin} + c_d) + C_{rout}$ (Figure 5(b)). Since the device capacitance is dynamic and depends on several electrical and technology parameters [7, page 180], for more accuracy we replace the capacitances in the non-represented leaf-cells by a current generator controlled by an ammeter (see figure 5(c)). As the leaf-cells are identical, the total current in the ammeter is equal to $(n-1) \times i$, where i corresponds to the current in the represented leaf-cell.

3.2 Memory block modeling

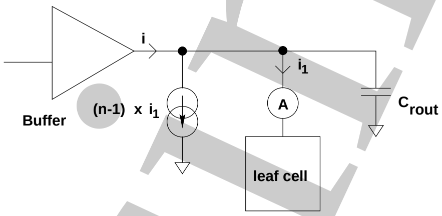
According to section 2, the ROM has a block partitioning architecture and the number of blocks connected to a driver changes from one configuration to another. Global buses go through each of the blocks (Memory array and datapath) as shown in figure 6, so that simulation scheme can be reduced to one block. If, for a given configuration, the number of blocks is equal to 1, the model is still valid since the value of the additional current produced by the generator becomes zero.



(a) n leafcells driven by a buffer



(b) corresponding model



(c) current controlled model

Figure 5: Current controlled generator concept

3.3 Word-line modeling

A local word-line scheme is depicted in figure 7(a), where capacitances driven by the buffer are the routing capacitance $C_R(n)$ and the gate capacitances represented in the *n* devices. The corresponding model is given in figure 7(b) using the current controlled generator concept as described in section 3.1.

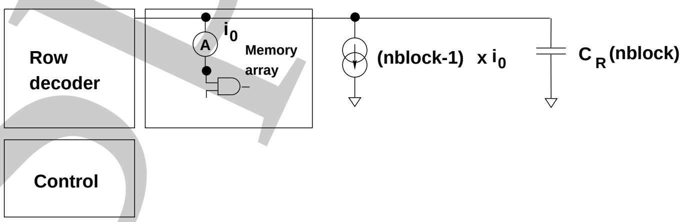
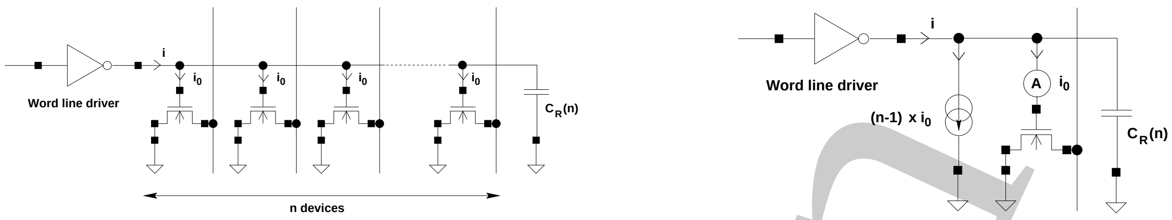


Figure 6: Block model



(a) A local word-line (b) Corresponding model

Figure 7: word-line model

3.4 Bit-line modeling

As depicted in figure 8, we can extend the word-line model by adding a bit-line . If we suppose that the number of devices attached to a bit-line is equal to m, then the value of the current in the second current generator is $(m-2) \times i_1$. The device, that is modeling the bit line, has its gate connected to ground because it represents a non activated word-line. $C_{Rwl}(n)$ and $C_{Rbl}(m)$ are the routing capacitances respectively attached to the word-line and the bit-line. As we want to characterize a ROM compiler, it's very easy to modify the n (number of columns) and m (number of rows) parameters according to the aspect ratio of the generated ROM.

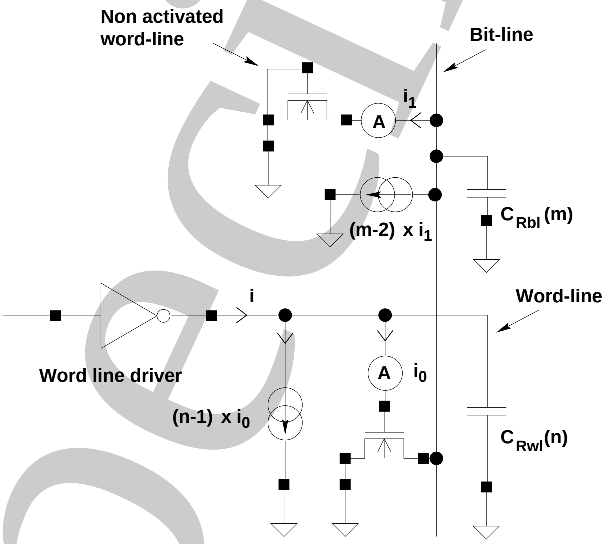


Figure 8: bit-line and word-line model

3.5 Column decoder and datapath modeling

Since we use selective precharge, the number of bit-lines to simulate for the power analysis is reduced. Moreover, by using our approach, only three columns of the datapath need to be represented in the model (Figure 9).

The first one (bit-line d) is used to measure the delay in the worst case : when the maximum of devices are connected to the column. According to figure 8, the m factor is maximum when each word-line is connected with the bit-line by a transistor, i.e. all memory points of the column are programmed to “zero”.

The two other columns (bit-lines 0 and 1) are used to measure the power consumption when we are reading respectively a "zero" or a "one". The number of devices connected to these bit-lines is half of the maximum since on average, we have an equal number of "zeros" and "ones" attached to a bit-line. Two current values are measured : I_{D0} for the column where a "zero" is read and $I_{D1}$the column where a "one" is read.

The rest of columns are taken in account using the block modeling with current generators and routing capacitances representing the load driven by multiplexer, precharge and output control buffers.

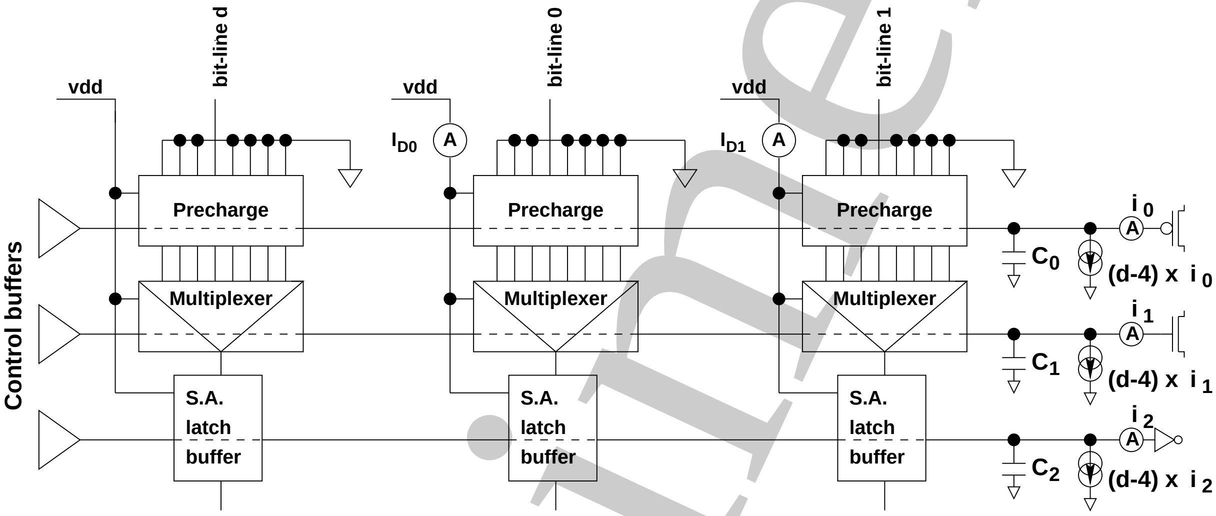


Figure 9: datapath model.

3.6 Control representation

As the control block is a logic block with no significant regularities, there is no specific model. For simulations, we use an extracted view of the layout with all parasitic capacitances. With an ammeter, we measure the consumed current (I_C) which includes all the currents consumed through the global drivers for the precharge, multiplexer and output cells.

3.7 Row decoder modeling

During Spice simulations we use only three memory cells located on the same row. Thus, only one row decoder is necessary to be represented. Capacitances in the other row decoders are modeled using the current generator technique and a routing capacitance. I_R is the measured current in both row decoder and word-line buffer.

4 Simulation results

It is possible to take the number of "zero" terms in account by multiplying n and m by the ratio of "zero" terms. To provide timings for a given configuration, we take a ratio of 1, since we want to estimate timings in the worst case. Whereas, to characterize the power consumption, we take a ratio equal to 0.5, because statistically the number of "zero" terms is very close to the number of "one" terms. Then, the total current in the memory can be estimated as follows :

$I = I_C + I_R + (0.5 \times number\ of\ bits\ per\ word \times (I_{D0} + I_{D1}))$

To predict the timing characteristics, such as the access time, Spice simulations are performed using models for repetitive structures and full extracted views for parts with logic. To estimate the average power consumption, P_{avg}, we measure the current in the supply for each part and then we integrate the value of the total current :

$$P_{avg.} = \frac{V_{dd}}{T} \int_0^T I(t)dt$$

In order to validate the modeling technique applied to our ROM compiler, we measure delays and power consumption on several compiled ROM parts such as the global word-line circuit. The global word-line buffer is driving respectively for each configuration, 1 (config 1), 2 (config2) or 3 (config 3) blocks of 256 columns. Then, for each configuration, we compare the values obtained with the model to those obtained on the full extracted corresponding part. For config 3, the size of the netlist with the model is 80% smaller than the one with the full extracted part. In the same way, simulation time of config 3 with the model is 5 times faster than the simulation time with full extracted part. As shown in table 1, the results are very close to each other, for both delay and power, in spite of the netlist reduction.

	Config 1	Config 2	Config 3
Delay from model	$6.597\ ns$	$6.859\ ns$	$7.108\ ns$
Delay from full extracted	$6.538\ ns$	$6.724\ ns$	$6.935\ ns$
Power from model	$4.215\ \mu W$	$6.006\ \mu W$	$7.912\ \mu W$
Power from full extracted	$4.518\ \mu W$	$6.018\ \mu W$	$7.613\ \mu W$

Table 1: Comparison between results with the model and those with a full extracted layout.

The currents measured in the memory during a read cycle are shown in figure 10. Table 2 gives the equivalent percentage for each part. Similar figures for different configurations help the memory designer to optimize the total power consumption by minimizing energy in each part.

Control	Row decoder	Precharge	Multiplexer and output	Local commands
$41.8\,\%$	$9.2\,\%$	$8.0\,\%$	$31.9\,\%$	$9.0\,\%$

Table 2: The contribution to the power consumption for each block

5 Conclusion

A modeling technique for regular and complex architectures has been presented in this paper. Since the method exploits regularities to significantly reduce extracted netlists, it is very suitable to accurately characterize low-power memories, where parallel blocks are often used. Our approach can be used for both timing and power characterization using the same model. It has been successfully applied to a low-power ROM compiler. Simulation results demonstrate that we can obtain accurate measures even with reduced netlists, by using current control generators to represent repetitive parts. This modeling technique can be efficiently used to develop, analyse, and improve performances of regular, but complex, structures. ROM compiler is a good example, but not the only one.

References

[1] K. Itoh, K. Sasaki, and Y. Nakagome, "Trends in low-power RAM circuit technologies," *Proc. of the IEEE*, vol. 83, pp. 524-543, Apr. 1995.

[2] O. Coudert, R. Hadda, and K. Keutzer, "What is the state of the art in commercial EDA tools for low power," in *Proc. ISLPED 199*6, Monterey, CA., 1996, pp. 181-187.

[3] J. M. Rabaey, "Digital integrated circuits, a design perspective," Upper Saddle River, NJ : Prentice Hall, 1996.

[4] M. Yoshimito, K. Anami, H. Shinohara, T. Yoshihara, H. Takagi, *et al.*, "A divided word-line structure in the static RAM and its application to a 64k full CMOS RAM," *IEEE J. Solid-State Circuits*, vol. SC-18, pp. 479-485, Oct. 1983.

[5] T. Hirose, H. Kuriyama, S. Murakami, K. Yuzuriha, T. Mukai, et al., "A 20-ns 4-Mb CMOS SRAM with hierarchical word decoding architecture," *IEEE J. Solid-State Circuits*, vol. 25, pp. 1068-1074, Oct. 1990.

[6] ATMEL CBIC AT55K/ECAT05 Library Databook, Sept. 1997.

[7] N. H. E. Weste, and K. Eshragian, "Principles of CMOS VLSI design, a systems perspective," 2nd edition, Reading, MA : Addison Wesley, 1994.

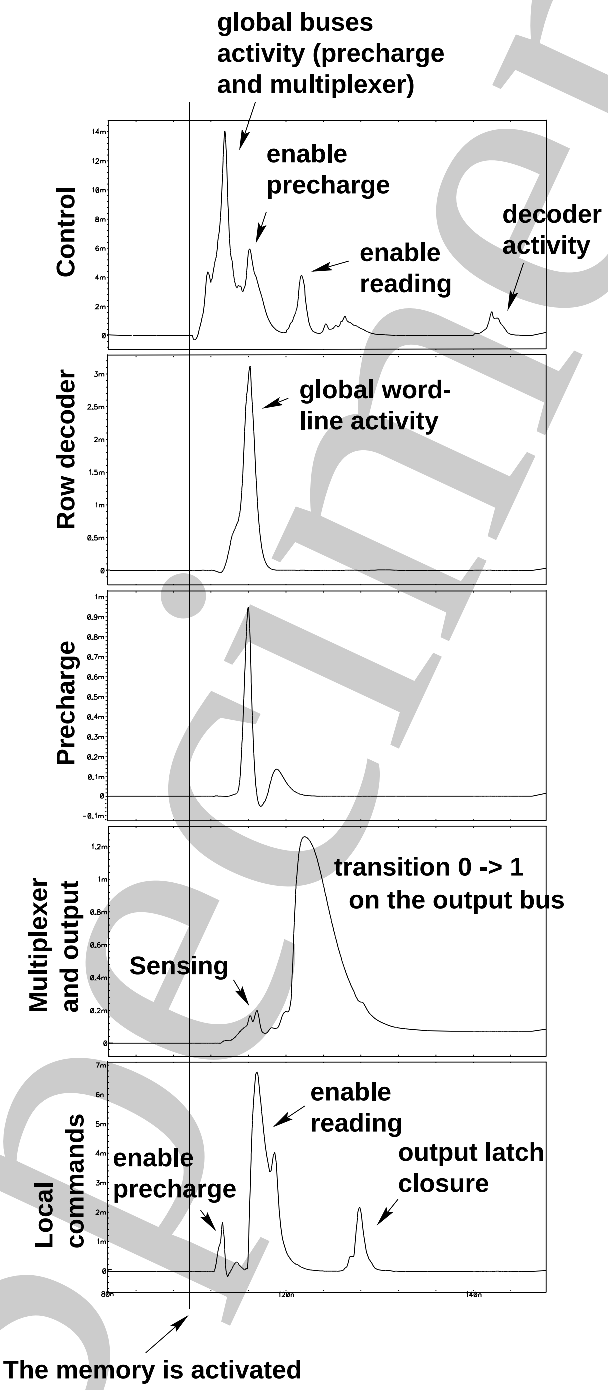


Figure 10: Dynamic current distribution during a read cycle