

# Survey of Low Power Techniques for ROMs

Edwin de Angel

Crystal Semiconductor Corporation  
P.O Box 17847  
Austin, TX 78744

Earl E. Swartzlander, Jr.

Department of Electrical and  
Computer Engineering  
University of Texas at Austin  
Austin, TX 78712

## Abstract

*This paper presents a survey of low power techniques for Read Only Memories (ROMs). Significant savings in power dissipation are achieved through the use of techniques at the circuit and architecture level. The ROM circuits have been designed in 0.35  $\mu\text{m}$  CMOS technology and simulated using PowerMill.*

## Introduction

With the development of submicron technologies and the increase of complexity on VLSI chips, the market for portable applications, digital signal processors and ASIC implementations has focused significant effort on the design of low power systems [1]. ROMs (Read Only Memories) are an important part of many digital systems (e.g., digital filters, digital signal processors, microprocessors etc). The high area density of ROMs makes these types of circuits very attractive to store fixed information (e.g., coefficients of a digital filter). As new submicron technologies are developed, the fast speeds of these processes allow the implementation of architectures which could not be implemented in the past. Also the increase in the number of metal layers becomes a main instrument to reduce switched capacitance without penalty in the density of the ROM. Significant savings in power are achieved through the implementation of several techniques. The focus of this paper is on techniques at the architecture and transistor levels and their global impact on power dissipation.

The first section of the paper explains traditional ROM designs and the sources of power dissipation. The second part of this paper discusses low power techniques at the architecture level. The next section presents techniques that are applicable at the circuit level. The last section shows results and conclusions.

## 1 Sources of Power Dissipation

Figure 1 shows the traditional architecture of a ROM. The decoder selects among the row lines that run through the ROM core, turning on only one row line at a given time. The column multiplexer and driver select which column is being read and drive the data bus. The control logic generates the internal signals of the ROM (i.e., precharge, read etc.). The ROM core is used to store information through the placement of transistors. There are two main types of ROMs: NAND array, where pull down transistors are in series and NOR array where the pull down transistors are in parallel. This

paper focuses on ROMs using a NOR array since these structures are faster than NAND arrays and are the most frequently used [2].

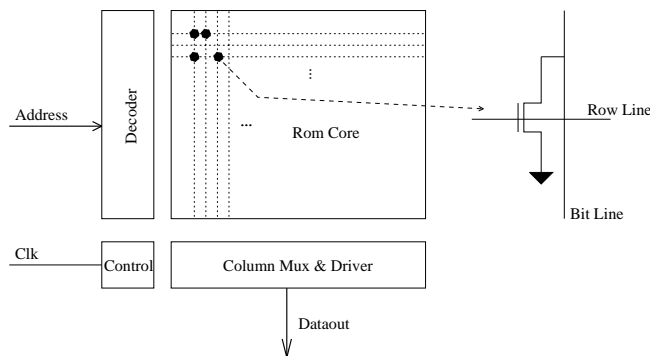


Figure 1: ROM Block Diagram

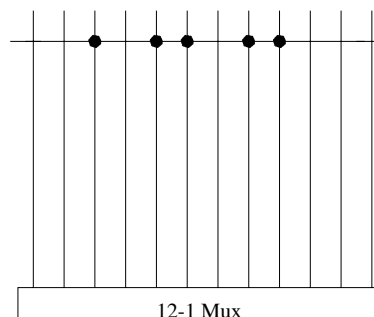


Figure 2: ROM Bitlines

In order to save power, most ROMs precharge during one phase of the clock and evaluate in the other. Table 1 shows the power dissipation in a 2K x 18 ROM designed in 0.6  $\mu\text{m}$  technology at 3.3V and clocked at 10 MHz. As the table shows, the precharge of the bit lines in the ROM core dissipates most of the power. There are two main reasons for this. First, bit lines have large capacitance (drain capacitance of the transistors tied to this line, parallel plate and fringe component to substrate plus the overlap of the row lines and other metal layers). Second, more than 18 bit lines are switched per access; this is due to the word line selecting more bit lines than

Table 1: Power Dissipation ROM 2k x 18

Block **	Power (mW)	Percentage (%)
Decoder	0.06	2.1
ROM Core	2.24	89
Control	0.18	7.2
Drivers	0.05	1.7

is necessary (see figure 2). The example presents a multiplexer ratio 12 to 1. As a result at least 4 more bit lines will switch instead of one.

The power dissipated in the control logic is because it contains all the drivers to generate the signals feeding the decoder. Also the control logic generate the precharge signal which is used to precharge the ROM core, enable the output drivers, and enable the decode logic. The power dissipated in the decoder is not much since only one row line switches per access.

## 2 Low Power Techniques: Architecture

Since most of the power dissipated is due to switching of the bit lines, a significant number of the following techniques focus on the ROM core.

### 2.1 Hierarchical Word Line

This concept has been proposed for static random access memories (SRAMS) [3]. The basic idea is to divide the memory in different blocks and run the block word line in one layer (i.e., metal1 or poly) and a global word line in other layer. As a result only the bit cells of the desired block are accessed. The same concept can be applied to ROMs. The ROM can be divided in several blocks and a given block is enabled through the address bits. Although a significant amount of the power dissipated can be reduced through this technique, it does not solve the problem completely, the main reason is that due to layout considerations a ratio of at least 4 to 1 is required in the multiplexer. A significant reduction in power is obtained but still more than one bit line per bit could be switching.

### 2.2 Selective Precharge

Large capacitance is being switched per cycle because every bit line is being precharged high during the first part of the cycle and many bit lines are discharged even when these locations are not accessed. Through selective precharge only bit lines which will be accessed are precharged [5]. The hardware overhead of this technique is low since most of this control logic is the same control logic required to control the multiplexers at the bottom of the ROM.

### 2.3 Minimization of Non-zero Terms

This technique focuses in the reduction of the capacitance in the bit lines and the row lines. This can be achieved by minimizing the number of non-zero terms in the ROM table which reduces the number of NMOS devices in the ROM core. This technique is very efficient since zero terms do not switch bit lines and reduce capacitance in both bit lines and row lines.

#### 2.3.1 Inverted ROM

If the number of ones is very high, the whole ROM core can be inverted and the final data inverted in the drivers. The efficiency of this type of encoding depends on the original number of non-zero

terms. If the number of non-zero terms is close to half the number of bits in the ROM core then the reduction of non-zero terms will be small or none.

#### 2.3.2 Inverted Row

The reduction of non-zero terms can be performed or done on a row by row basis. A given row is inverted if more than half of the bits are non-zero terms. Figure 3 shows two original rows and the result after the technique has been applied. It is important to observe that an extra bit per row is required to perform the encoding. Also note that if the the whole ROM would have been inverted the reduction of non-zero terms in one row would have been offset by the increase in the other one.

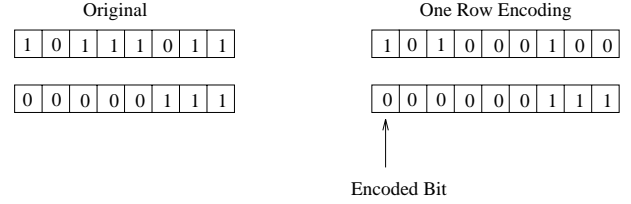


Figure 3: Inverted Row

#### 2.3.3 Sign Magnitude Representation

Often a ROM is used to store the coefficients of a digital filter. As a result, a significant amount of the non-zero terms are due to the sign extension of the negative coefficients. Sign Magnitude representation can be used to reduce a significant number of the ones. The main drawback of this type of encoding is that a conversion to two's complement is required at the end of a cycle, which slows down the ROM. Still for applications like mixed-signal systems where speed is not an issue, this type of encoding can be very useful.

#### 2.3.4 Sign Magnitude and Inverted Block

The number of non-zero terms can be reduced further more if the sign magnitude representation is implemented along with the inverted row encoding. After the sign magnitude is done, the inverted row encoding could be applied in a subset of the row (e.g., the 5 least significant bits).

### 2.4 Difference Encoding

Difference encoding can be used to reduce the whole size of the ROM core. For digital filters and other applications the ROM is accessed sequentially. If the values between adjacent data do not change significantly between one address and the next, the ROM core can store the difference between the data instead of the whole value [4]. The main disadvantage is that an adder is required to calculate the original value.

A variation of the same concept is to hard wire different constants (i.e., offsets) and store only the difference with respect to the constant.

### 2.5 Smaller ROMs

Figure 4 shows the coefficients of a 102 tap FIR filter. If these coefficients are stored in ROM, the largest coefficients will determine the size of the ROM required. More than 70% of the coefficients are below 18 bits. Still the largest coefficient goes up to 24 bits. As

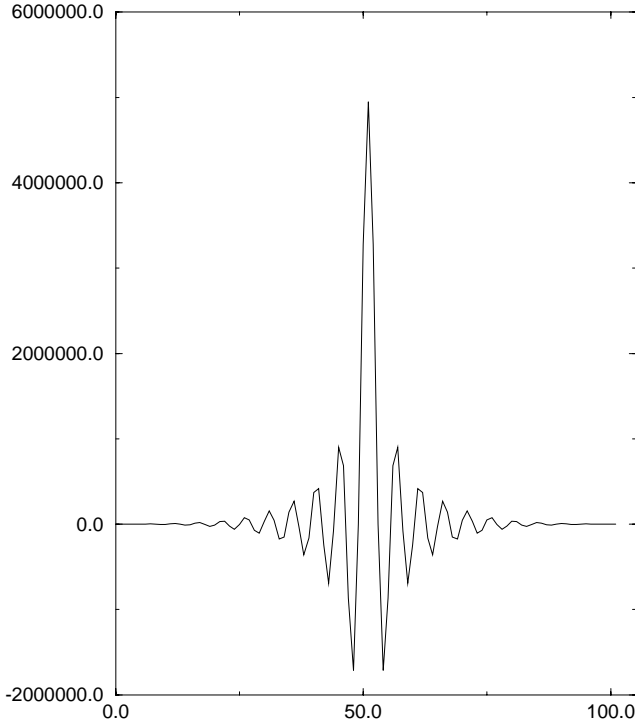


Figure 4: 102 Tap FIR Filter

### 3 Low Power Techniques: Circuit Level

Low power techniques at the circuit level can be powerful tools to reduce the power in VLSI systems [7].

### 3.1 NMOS Precharge

An important technique to reduce the power dissipated in the bit lines is limiting the voltage swing. This can be done through NMOS precharge of the ROM core; NMOS transistors are used to precharge bit lines high. As a result, bit lines are precharged to  $V_{dd} - V_t$ , where  $V_t$  is the threshold voltage. Since the bit lines switch only between  $V_{dd} - V_t$  and ground significant savings can be achieved. A drawback of this technique is degradation of noise margins and the body bias effect (which increases the threshold voltage) requiring careful design of the output drivers.

### 3.2 Voltage Keeper

Once the number of non-zero terms has been minimized, switching of bit lines is reduced. Still even if the same location of the ROM is accessed repeatedly, bit lines need to be precharged every time.

In order to avoid switching in the data bus or the adder required to convert from sign magnitude to two's complement a voltage keeper is used to minimize switching.

Figure 5 shows a possible implementation of the keeper with the invert logic. The voltage keeper is used to store past history and avoid transitions in the data bus and adder (if sign magnitude is implemented). Fire signal is enabled after the ROM core has evaluated. Pass and Invert signals are used if sign magnitude or Row Invert are implemented.

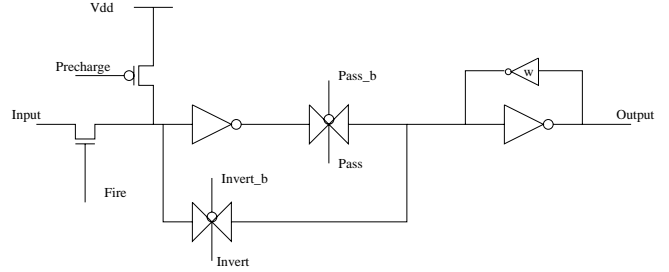


Figure 5: Output Stage

### 3.3 Buffer Sizing

A large set of buffers is required in the control logic to drive the address lines through the decoder, generate the control signals for the column multiplexers, drive the row lines and drive the precharge signals. For a long time, the optimum buffer tapering factor  $e = 2.72$  has been used [8]. Figure 6 presents the model used. In the figure  $g$  represents the conductance while  $\lambda$  represents the taper defined as:

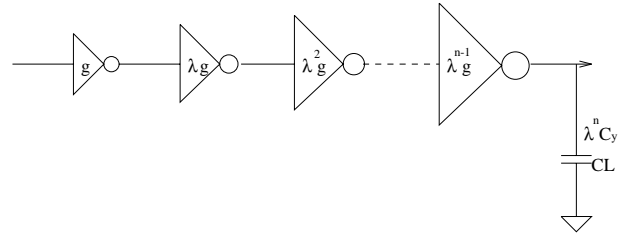


Figure 6: Driving Large Capacitive Loads

$$(W/L)_{k+1} = \lambda(W/L)_k \quad (1)$$

where  $W$  and  $L$  are the width and length of transistors in a given stage. In this case  $\lambda$  indicates the size of stage  $k + 1$  relative to stage  $k$ . The number of stages required for a given capacitive load is:

$$n = \frac{\ln C_L / C_i}{\ln \lambda} \quad (2)$$

This model ignores the effect of parasitic capacitances at the output of each stage. Haviland [9] includes the parasitic capacitance in the calculations using a split capacitor model (see figure 7).  $C_x$  and  $C_y$  are the inherent output capacitance and the incidental load capacitance respectively. Using this model and developing an equation to minimize delay the optimum taper factor is:

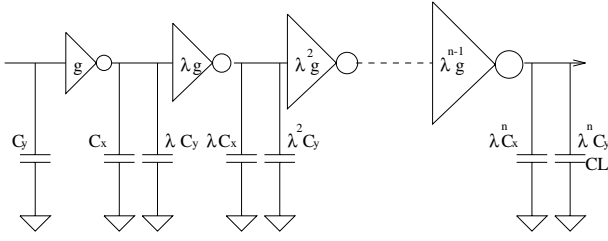


Figure 7: Improved Model

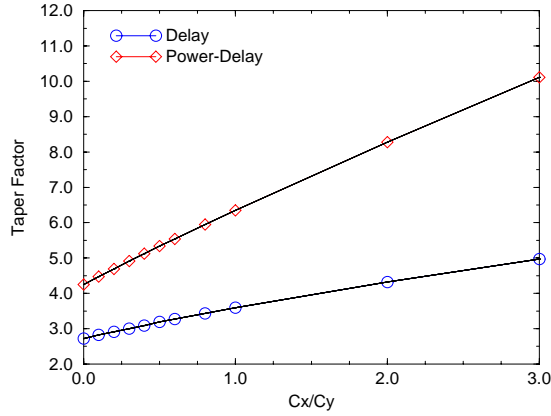


Figure 8: Power-Delay Product versus Delay

$$\lambda[\ln(\lambda) - 1] = \frac{C_x}{C_y} \quad (3)$$

This equation shows that the optimum taper  $\lambda$  depends on the ratio of  $C_x/C_y$ . Still, this equation has been developed to minimize delay. For power dissipation, there are often large capacitive loads which are not in the critical path. Choi [10] derived the tapering factor to minimize power-delay product using the same model. The optimum  $\lambda$  can be expressed as:

$$(\lambda - 2) \ln(\lambda) - (\lambda - 1) = 0 \quad (4)$$

If the parasitic capacitances are neglected  $\lambda = 4.25$ . Haviland [9] shows that both tapering factors can be related by:

$$\lambda_{Power-Delay} = (\lambda_{Delay})^\alpha \quad (5)$$

where  $\alpha \cong 1.44$ . Figure 8 shows a graph comparing the  $\lambda$  for different ratios of  $C_x/C_y$ . A different derivation to minimize power under delay constraint have been done by Figueras [13].

### 3.4 Reduction of Short Circuit Currents

Careful design of the control logic is required in order to avoid turning on row lines when the precharged circuitry is on. Also output drivers need to be enabled after the ROM core has evaluated. Delay lines can be used to generate signals with precise timing [12]. A robust design of the delay lines is needed to avoid performance degradation through process variations.

A significant reduction of the short circuit dissipation can also be achieved through scaling of the power supply. Accurate expressions to estimate short circuit currents has been done by Cauhape [13].

Table 2: ROM Encoding

Encoding **	Power (mW)
Two's Complement	0.80
Sign Magnitude	0.78
Row Invert	0.69

Table 3: Selective Precharge

Selective Precharge **	Power (mW)
Before	0.69
After	0.58

### 3.5 Voltage Scaling

Voltage scaling is one of the most powerful tools to reduce the power dissipation. A quadratic improvement can be easily achieved through voltage scaling. Although this technique is very effective in reducing power the speed of the circuits is degraded as the voltage goes down. A first order derivation [1] shows that the delay of CMOS gates can be expressed as:

$$T_{delay} = \frac{C_L V_{dd}}{I} = \frac{2C_L V_{dd}}{\mu C_{ox} (W/L) (V_{dd} - V_t)^2} \quad (6)$$

The speed of ROMs is degraded significantly because the transistor driving the bit lines is close to minimum size.

## Results

Table 2, 3 and 4 shows the cumulative effects of applying multiple low power methods. First a conventional 256 x 24 ROM using two's complement was designed. Next sign magnitude was applied to the data plugged into the ROM. The next design implements the row invert encoding in addition to sign magnitude. Table 2 compares the results of the several encodings in a 256 x 24 ROM. The data stored in the ROM was generated through a pseudo random function in C language. The ROMs were designed with a mux ratio of 4 to 1, simulated with PowerMill [14] at 3.3V, 10MHz in 0.35μm technology.

From the table it can be observed that since the data in the ROM is random, power savings using row invert encoding are greater than using sign magnitude encoding. For digital filters (see figure 4) and other applications where small negative numbers are required sign magnitude gives better results.

Table 3 shows a comparison of the ROM with row invert encoding before and after selective precharge has been implemented. Through selective precharge only 1 out of 4 columns are precharging resulting in significant savings in power.

Table 4 shows the power dissipation of the ROM when the voltage is scaled to 2.5V. Although significant savings are reached quadratic savings are not achieved due to increase in short circuit currents.

Table 4: Voltage Scaling

Voltage **	Power (mW)
3.3V	0.58
2.5V	0.39

Table 5: Voltage Scaling

Technique **	Conditions ***	Power Savings (%)
Sign Magnitude	Random Data	2.5
Row Invert	After Sign Magnitude	11
Selective Precharge	After Sign Magnitude and Row Invert	14
Voltage Scaling	After Other Techniques	24
Total	After all techniques	51

Table 5 shows the power savings of the different techniques. The power savings shown for selective precharge and voltage scaling are after the other techniques have been applied.

## Conclusion

ROM Low power techniques at the architectural and the circuit level have been presented. The use of several of these technique significantly reduces the power dissipated in the ROM. The efficiency of the different techniques depends on the data stored to be stored in the ROM core, speed requirements and area overhead. High power savings can only be achieved through the use of multiple techniques.

## REFERENCES

- [1] A. P. Chandrakasan, S. Sheng and R. W. Brodersen, "Low-Power CMOS Digital Design," *IEEE Journal of Solid-State Circuits*, vol. 27, pp. 473-483, 1992.
- [2] D. A. Hodges and H. G. Jackson, *Analysis and Design of Digital Integrated Circuits*, Second edition, McGraw-Hill Publishing Company. pp. 346-353, 1988.
- [3] M. Yoshimito, K. Anami, H. Shinohara, T. Yoshihara, H. Takagi, S. Nagao, S. Kayano, and T. Nakano, "A Divided Word-Line Structure in the Static RAM and its Application to a 64K Full CMOS RAM," *IEEE Journal of Solid-State Circuits*, vol. SC-18, pp. 479-485, 1983.
- [4] N. Sankarayya and K. Roy, "Algorithms for Low Power FIR Filter Realization Using Differential Coefficients," *IEEE 10th International Conference on VLSI Design*, Hyderabad, India, pp. 174-178, 1997.
- [5] N. Weste, and K. Eshraghian, *Principles of CMOS VLSI Design: A Systems Perspective*, Second edition, Addison-Wesley, pp. 585-588, 1993.
- [6] C. Piguet, "Low-Power Microprocessors and Memories," *NATO Seminar on Low Power Design in Deep Submicro Electronics*, Lucca, Tuscany, Italy, August 20-30, 1996.
- [7] E. de Angel and E. E. Swartzlander Jr., "Survey of Techniques for Low Power VLSI Design," *International Conference on Innovative Systems in Silicon*, pp. 159-169, 1996.
- [8] R. C. Jaeger, "Comments on 'An optimized output state for MOS integrated circuits,'" *IEEE Journal of Solid-State Circuits*, vol. 10, pp. 185-186, 1975.
- [9] G. L. Haviland and A. A. Tuszynski, "CMOS Tapered Buffer," *IEEE Journal of Solid-State Circuits*, vol. 25, pp. 1005-1008, 1990.
- [10] J. Choi and K. Lee, "Design of CMOS Tapered Buffer for Minimum Power-Delay Product," *IEEE Journal of Solid-State Circuits*, vol. 29, pp. 1142-1145, 1994.
- [11] H. J. Veendrick, "Short-Circuit Dissipation of Static CMOS Circuitry and Its Impact on the Design of Buffer Circuits," *IEEE Journal of Solid-State Circuits*, vol. SC-19, pp. 468-473, 1984.
- [12] M. Santoro, *Design and Clocking of VLSI Multipliers*, Ph.D. Dissertation, Stanford University, 1990.
- [13] J. Figueras, "Power Modeling," *NATO Seminar on Low Power Design in Deep Submicro Electronics*, Lucca, Tuscany, Italy, August 20-30, 1996.
- [14] C.X. Huang, B. Zhang, A-C. Deng, and B. Swirski, "The Design and Implementation of PowerMill," *Proceedings 1995 International Symposium on Low Power Design*, pp. 105-109, 1994.