

# AN AREA-SAVING 3-DIMENSIONAL DECODER STRUCTURE FOR ROMS\*

Chua-Chin Wang<sup>†</sup>, Ya-Hsin Hsueh, Ying-Pei Chen<sup>‡</sup>

Department of Electrical Engineering, National Sun Yat-Sen University, Kaohsiung, Taiwan 80424

Tel : 886-7-525-2000 ext. 4144 , Fax : 886-7-5254199

E-Mail: ccwang@ee.nsysu.edu.tw

**ABSTRACT:** ROMs (Read-Only Memory) are widely used in both digital communication systems and daily consumer electronics. The major functions of ROMs are storage of data, program, firmware, etc. In this work, an area-saving decoder structure for ROMs is proposed. The stages of address decoding are drastically shortened owing to the a 3-dimensional decoding method is employed. A real 256×8 ROM possessing the proposed decoder is physically fabricated by 0.5 μm 2P2M CMOS technology.

## 1. INTRODUCTION

ROMs are an important part of many digital systems, e.g., DSP, microprocessors, digital filters, etc. They are particularly important in portable systems due to the storage of programs and data. Hence, the chip size and the power consumption need to be enhanced besides the improvement of speed. Prior ROM designs were mainly focused on the technology evolution [1], [4], [6], core architecture [5], [7], or special-purposed circuit and logic [2], [3]. The improvement of address decoders and data encoding for ROMs has long been ignored. Most of the prior decoders for ROMs utilized multiplexers to decode the row and column addresses. However, the characteristics of P+implant ROMs cause the Hamming distance of adjacent words to be 1. This feature leads the order of the data words to be stored will depend on the decoder structure such that they appear in a non-natural order. Thus, the following problems will be introduced.

- 1). Different ROM users will store the data words in different patterns owing to the different decoder structures.
- 2). Due to the data words stored in various patterns, programs to call the data stored in the ROMs must be adjusted accordingly.
- 3). Such kind of ROMs are very difficult to be tested.

Our method simplifies the procedure to encode the ROMs and decode the address such that application programs need no adjustment and testing becomes much easier. Besides, not only is the size of the entire decoder shrunk, but also the access time and power dissipation are greatly reduced.

\*This research was partially supported by Nation Science Council under grant NSC 89-2215-E-110-017 and NSC 89-2215-E-110-014.

<sup>†</sup> Contact author

<sup>‡</sup> Ying-Pei Chen is currently working as an IC design engineer in VIA Technologies, Taipei, Taiwan.

## 2. AREA-SAVING DECODER FOR ROMS

### 2.1. 1-dimensional decoder structure of ROMs

Referring to Fig. 1, a straightforward decoding circuit, for a P+implant ROM is shown. The “P+implant” NMOS transistor denotes that a NMOS is added a layer of P+ implant. Then, the gate of this NMOS is always opened such that it can not be turned on by a voltage asserted on its gate. Meanwhile, PMOS transistors only appear in the designated module in which PMOSs are used as pull-up loads. The operations of such a P+implant ROM in Fig. 1 are as follows :

- 1). precharging : When clock is low, the outputs, W7 - W0, are all precharged to high.
- 2). evaluation : When clock turns high, address inputs, A0, A1, and A2, will then determine which output line is connected to ground.

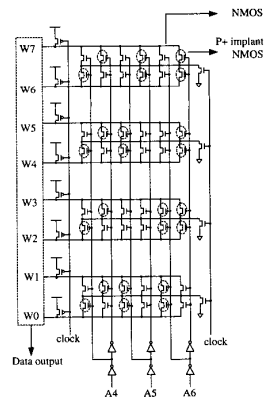


Figure 1: A simple ROM 1-dimensional decoding circuit

### 2.2. 2-dimensional decoder structure

When the address size of ROMs increases linearly, the 1-dimensional decoder structure will inevitable make the core of the ROM grow exponentially. Not only such a decoding scheme leads to a large area, but also increases the access time of data. A 2-dimensional decoder structure was proposed to resolve such a problem. It was also named as “X-Y” decoding. Fig. 2 shows a 128×1 ROM. The operations of such a design are illustrated as follows.

- 1). The address lines are equally divided into two parts. A6, A5, and A4 are fed into a P+implant decoder which in fact is a 1-dimensional ROM decoder to generate a word selection signal, W7, W6, ..., or W0. The generated word selection signal will be fed to the ROM data core and determine which row of the core is activated. For instance, if A6A5A4 = 111, W7 is activated.
- 2). The upper decoder is fed with A3, A2, and A1, to decode which pair of columns in the data core is selected. Notably, N+implant NMOS transistors are utilized in the upper decoder. The N+implant NMOS possesses a N+ layer such that the transistor is always turned on regardless of the asserted gate voltage. Hence, if A3A2A1= 111, then the pair of column 14 and 15 is selected.
- 3). The lower decoder is fed with A3, A2, A1, and A0, to determine which one of the selected column is granted to be the output. Assume A0 is 0. Then, column 14 is chosen to be output.
- 4). The states of the 128 bits in the 128x1 ROM is determined by whether there is a P+implant NMOS in the corresponding position.

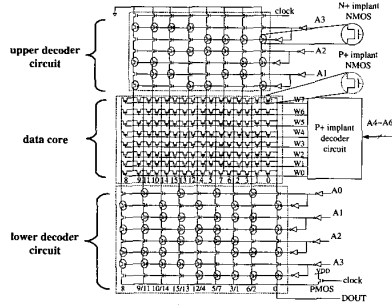


Figure 2: Detailed schematic of the 2-dimensional decoding circuit

### 2.3. 3-dimensional decoder structure

The major disadvantage of the 2-dimensional decoding scheme is that the column-sharing design leads to a non-natural order of data arrangement, i.e., the order of the data bit is 0, 1, 3, 2, 6, 7, ... This drawback brings up two side effects: hard to program the needed data, and difficult to debug. We, thus, propose a 3-dimensional decoder structure to resolve these problems without any format conversion circuit. Fig. 3 shows the detailed schematic of the 3-dimensional decoder. Take the same 128x1 ROM as an example.

- 1). The address lines are divided into three parts: A6, A5, and A4 are used to decode the row number of the data core; A3, A2, and A1 are fed to an upper decoder; and only one A0 is required in the lower decoder circuit.
- 2). Two additional modules are needed to resolve the non-natural encoding problem in prior ROM decoder designs. They are the upper(lower) pass block associated with the upper(lower) decoder. The elements of these two blocks are NMOSs.

- 4). A6, A5, and A4 selects a row (word) in the data core. A3, A2, and A1 then determine which shared column is activated. The binary-tree-like decoder can be used for saving area.
- 5). Two pass transistors gated with A0, then, decide which side of the activated column to be the output, "DOUT".

Notably the order of the data encoded in the ROM core is in a natural order of sequence in such a decoder design. That is, 0, 1, 2, 3, ... Besides, the binary-tree-like decoders without any N+implant saves a large number of transistors such that the area becomes much smaller.

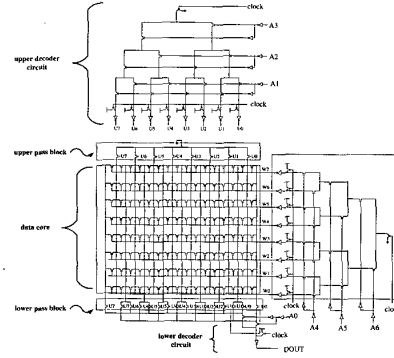


Figure 3: Schematic diagram of the 3-dimensional decoding circuit

### 2.4. performance analysis

**Transistor Count:** Assume there  $n$  address lines to be decoded for a ROM. The following analysis exclude the inverters (buffers) on the address lines and the pre-charging PMOSs at the output lines, since they are all required in any decoder circuit.

**2-dimensional scheme:** Referring to Fig. 2, assume the  $y$  address line for the data core,  $(n - y - 1)$  address line for the upper decoder, and  $(n - y)$  address line for the lower decoder. We conclude the following facts while not counting the N+implant NMOS which are always on.

No. of MOSs in the upper decoder circuit :

$$2^{n-y-1-1+1} \cdot (n - y - 1 + 1)$$

No. of MOSs in the lower decoder circuit :

$$(2^{n-y-1} - 1) \cdot (n - y - 1) + 2(n - y)$$

No. of MOSs in the P+implant core decoder circuit :

$$2^{y-1} + 2^y \cdot (2y) + 2^y$$

Thus, the total transistor count in this scheme is the summation of the above three terms.

$$M_2(n, y) = (2^{n-y-1}) \cdot (n - y) + (2^{n-y-1} - 1) \cdot (n - y - 1) + 2(n - y) + 2^{y-1} + 2^y \cdot (2y) + 2^y \quad (1)$$

**3-dimensional scheme:** Referring to Fig. 3, a total of  $n$  address lines are divided into three parts,  $y$  for the data core decoding,  $z$  for the upper decoder, and 1 for the lower decoder.

No. of MOSs in the upper decoder circuit :  
 $(2^{z+1} - 1) + 2^z + 2 \cdot 2^z$   
 No. of MOSs in the upper pass block :  $2^z$   
 No. of MOSs in the lower decoder circuit : 2  
 No. of MOSs in the lower pass block :  $2 \cdot 2^z$   
 No. of MOSs in the P+implant core decoder circuit :  
 $(2^{y+1} - 1) + 2^y + 2 \cdot 2^y$

Thus, the total transistor count in this scheme is the summation of the above five terms. Besides, we also substitute  $z = n - y - 1$  to derive the following result.

$$\begin{aligned} M_3(n, y, z) |_{z=n-y-1} &= (2^{z+1} - 1) + 2^z + 2 \cdot 2^z + 2^z + 2 \\ &\quad + 2 \cdot 2^z + (2^{y+1} - 1) + 2^y + 2 \cdot 2^y \\ M_3(n, y) &= 4 \cdot 2^{n-y} + 5 \cdot 2^y \end{aligned} \quad (2)$$

By using Eqn.(2), we can make a comparison in the following table.

$n$	$y$	$M_3$	$M_2$
7	4	112	176
8	3	168	210
8	4	144	213
8	5	192	392

Table 1: Comparison of transistor count in different scenarios

It is trivial to find that  $M_1$ , which is the transistor count of the 1-dimensional decoder, is definitely much larger than  $M_2$  and  $M_3$ . Hence,  $M_3$  is the most transistor (area) saving in the above three decoder designs.

**Speed analysis :** Since a design with minimal number of MOSs might not be the fastest circuit, the cost of delay using the 3-dimensional decoder structure should be evaluated before the selection of any  $y$  or  $z$ . The wiring delay is negligible when compared with the transistor delay. We, thus, assume that the "delay count" is proportional to the number of MOSs on the data path from input to output. Hence, the delay measure of the 3-dimensional decoding scheme is formulated as follows.

$$\begin{aligned} D_3(n, y, z) |_{z=n-y-1} &\approx z + 1 + 1 + 1 + 2^y \\ &\approx z + 3 + 2^y = n - y + 2 + 2^y \\ &= D_3(n, y) \end{aligned} \quad (3)$$

The following table is then derived to discover what is the best choice for the decoder for a  $256 \times 8$  ROM.

$y$	$M_3$	$D_3$	$M_3 \cdot D_3$
2	286	12	3432
3	168	15	2520
4	144	22	3168
5	192	37	7104

Table 2: Comparison of transistor-delay product when  $n = 8$

According to Table 2, although the scenarios given  $y = 3$  and  $y = 4$  appear to possess almost the same transistor-delay product, the delay is generally deemed as a parameter with a higher priority. This why we choose  $n = 8$  and  $y = 3$  to implement a  $256 \times 8$  ROM in the following text.

## 2.5. $256 \times 8$ ROM

A  $4 \times 4$  integer multiplier can be realized by a  $256 \times 8$  ROM in which 4 address lines denotes the multiplicand while the rest 4 represents the multiplier, and the byte wide output is the product. The  $256 \times 8$  ROM is composed of eight  $256 \times 1$  ROMs, i.e., ROM0, ROM1, ..., ROM7. Their individual output is d0, d1, ..., d7, respectively. Fig. 4 is the layout of the entire  $256 \times 8$  ROM which is fabricated with UMC (United Microelectronics Company)  $0.5 \mu\text{m}$  2P2M CMOS technology. The size of the chip is  $1.8 \times 1.8 \text{ mm}^2$ .

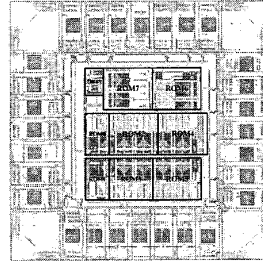


Figure 4: Layout of the chip

## 3. SIMULATION AND TESTING

**Pre-Layout Simulation :** Fig. 5 shows the simulation waveforms of the address inputs and the output product given by thorough HSPICE simulations. The maximal access time (delay) is 3.3 ns without pads. The post-layout simulations shows that the highest clock rate is 50 MHz (period = 20ns) with pads. Table 3 reveals the performance comparison with several prior ROM designs.

Design	access time	technology
Neruke's [4]	120 ns	$0.72 \mu\text{m}$ CMOS
Sunaga's [5]	60 ns	$1.0 \mu\text{m}$ CMOS
ours	20 ns	$0.5 \mu\text{m}$ CMOS

Table 3: Comparison with prior ROMs (with pads)

**Physical Chip Testing :** The proposed ROM decoder was approved by CIC (Chip Implementation Center) of NSC (National Science Council), and then fabricated by UMC. The chip number is U05-89C-22u. Fig. 6 is the die photo of the  $256 \times 8$  ROM and the proposed decoder. Fig. 7 are the snapshots generated by HP 1660CP when the chip is under test. The maximum clock rate is 20 MHz, while the maximal access delay is measured to be 16 ns given 1000 test patterns. These results justify our proposed design.

## 4. CONCLUSION

We present a novel area-saving and high-speed decoder structure for ROMs. The transistor count and the delay measure are clearly analyzed and verified. The physical chip implementation using the proposed 3-dimensional decoding schemes is also present. The simulation results turn out to be very appealing.

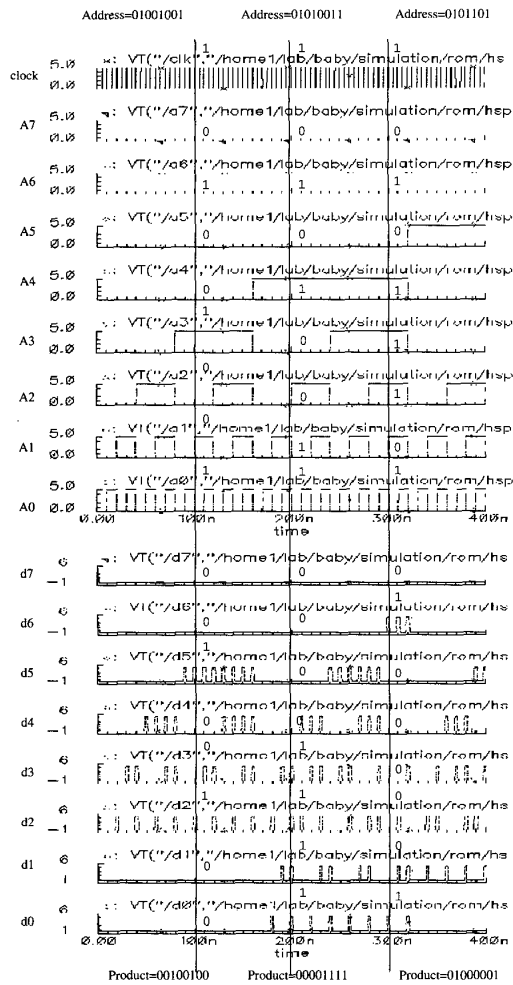


Figure 5: Simulations waveforms

### 5. REFERENCES

- [1] E. Bertagnoli, *et al.*, "ROS : an extremely high mask ROM technology based on vertical transistor cells," *1996 Symp. on VLSI Technology Digest of Technical Papers*, pp. 58, 1996.
- [2] D. A. Hodges, and H. G. Jackson, "Analysis and design of digital integrated circuits," Reading: 2nd ed., McGraw-Hill Publishing Company, 1988.
- [3] R. Kanan, A. Guyot, B. Hochet, and M. Delclercq, "A divided decoder-matrix (DDM) structure and its application to a 8Kb GaAs MESFET ROM," *1997 IEEE Inter. Symp. on Circuits and Systems*, pp. 1888-1891, June 1997.
- [4] Y. Naruke, T. Iwase, M. Takizawa, K. Saito, M. Asano, H. Nishimura, and T. Mochizuki, "A 16 Mb

mask ROM with programmable redundancy," *IEEE Inter. Solid-State Circuits Conference*, pp. 128-129, Feb. 1989.

- [5] T. Sunaga, "A 30-ns cycle time 4-Mb mask ROM," *IEEE J. of Solid-State Circuits*, vol. 29, no. 11, pp. 1353-1358, Nov. 1994.
- [6] H. Takahashi, S. Muramatsu, and M. Itoigawa, "A new contact programming ROM architecture for digital signal processor," *1998 Symp. on VLSI Circuits Digest of Technical Papers*, pp. 158-161, 1998.
- [7] A. Tuminaro, "A 400MHz, 144Kb CMOS ROM macro for an IBM S/390-class microprocessor," *1997 Inter. Conf. on Computer Design*, pp. 253-255, Oct. 1997.

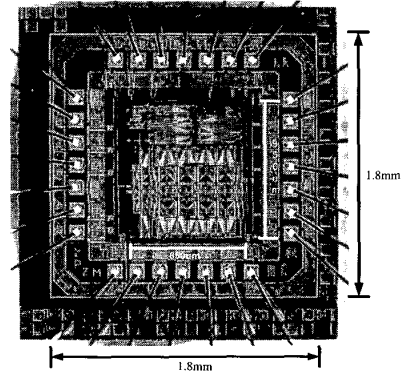


Figure 6: Die photo

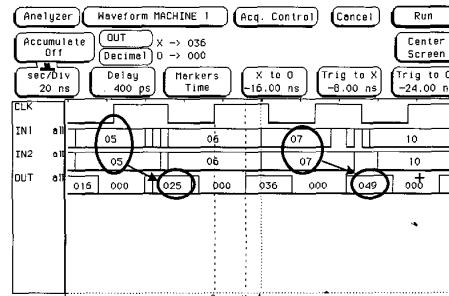
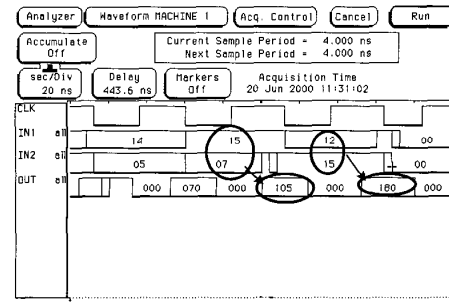


Figure 7: Testing of the real chip