

## A Multiplier Based on the Algorithm of Chinese Abacus

<sup>1</sup>Chien-Hung Lin, <sup>2</sup>Shu-Chung Yi and <sup>1</sup>Jin-Jia Chen

<sup>1</sup>Department of Electrical Engineering,  
National Changhua University of Education

<sup>2</sup>Graduate Institute of Integrated Circuit Design,  
National Changhua University of Education

The National Changhua University of Education

No.1, Jin-De Road, 50007, Changhua,

Taiwan, R.O.C

[scy@cc.ncue.edu.tw](mailto:scy@cc.ncue.edu.tw)

*Abstract*—A 4x4 and 8x8 bit multiplier is demonstrated based on the Chinese abacus. As comparing the simulation result of this work with the speed of the 4x4 and 8x8 bits Braun array multiplier, the delays of the 8-bit abacus multiplier are 14% and 7.5% less than that of Braun array multiplier with 0.35 $\mu$ m and 0.18 $\mu$ m technologies, respectively. Meanwhile, the power consumption of the 8-bit abacus multiplier is, respectively, less about 11.9% and 22.3% also.

*Key Word*—Braun array multiplier, Chinese abacus multiplier, fast multiplier.

### 1 Introduction

Multiplication is one of the most critical operations in many computational systems. Among various multiplier techniques, array-based multipliers [1][2] and tree-based multipliers [3] are the most well known techniques and are often used in the VLSI design for implementing fast multipliers. This study presents a multiplier based on the antique Chinese abacus algorithm to achieve an efficient operation with high speed and low power consumption.

The Chinese abacus is a very old and very popular invention, which has been used for centuries in China and other Asian countries to perform arithmetic functions. The basic architecture of the Chinese abacus is depicted in Fig.1, which demonstrates a decimal number of one hundred sixty-eight. Each column element has one higher bead with a weight of five and four lower beads, each with a weight of one. The key feature of the Chinese abacus is the use of one bead with weight five. This allows the user to minimize the transmission of rests. The first multiplier and adders employing the technique of the Chinese abacus are proposed by Gang et al. [4]–[6]

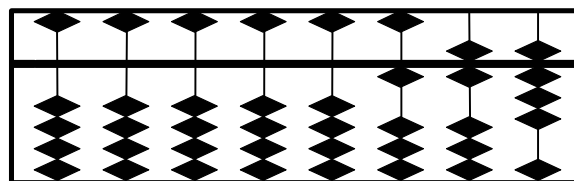


Fig.1. Basic architecture of Chinese abacus coded with a decimal number of 168.

The proposed Chinese abacus multiplier is based on an abacus adder in [9] that each column element, e.g.  $(H_2H_1H_0|M_2M_1M_0|L_2L_1L_0)_{\text{abacus}}$ , consists of three different weighted bead groups, which represent a decimal number to be

$$\begin{aligned} & (H_2H_1H_0|M_2M_1M_0|L_2L_1L_0)_{\text{abacus}} \\ &= (H_2 + H_1 + H_0) * 16 + (M_2 + M_1 + M_0) * 4 + (L_2 + L_1 + L_0) \end{aligned}$$

For instance, a multiplication operation is demonstrated in Fig.2, where  $B = (b_3b_2b_1b_0)_2 = (1101)_2 = 13$  is the multiplicand and  $A = (a_3a_2a_1a_0)_2 = (1110)_2 =$

14 is the multiplier. This multiplication is first proceeded with two partial products, i.e.,  $(a_1a_0)_2 * (1101)_2 = (001|011|011)_{abacus}$  and  $(a_3a_2)_2 * (1101)_2 = (011|001|111)_{abacus}$ , and then the binary products of these two partial products give the product of  $B \times A$ , which is denoted by a binary product to abacus (BPA),  $(011|111|001|011)_{abacus}$ , as shown in Fig.2. This binary product represents a decimal number to be  $(011|111|001|011)_{abacus} = 2 \times 4^3 + 3 \times 4^2 + 1 \times 4^1 + 2 \times 4^0 = 182$ .

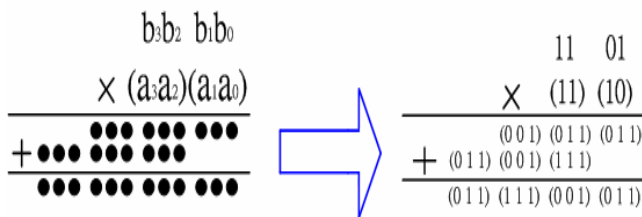


Fig.2. Example of the multiplication based on the proposed algorithm.

The block diagram of the proposed multiplier is depicted in Fig.3. The 4x4 abacus multiplier is divided into three modules. The first one is the BPA(binary product to abacus) module. The second one is the PA (parallel addition) module [9]. The third one is TB (Thermometric to Binary) [9]. These three modules are discussed in the following sections.

## 2 The Design of the Proposed Multiplier

### 2.1 The BPA Module

The block diagram of the BPA module is depicted in Fig.4. This module converts each 4x2 binary number,  $(b_3b_2b_1b_0)_2 * (a_1a_0)_2$  and  $(b_3b_2b_1b_0)_2 * (a_3a_2)_2$ , into an abacus representation  $(H_2H_1H_0|M_2M_1M_0|L_2L_1L_0)_{abacus}$ .  $(H_2H_1H_0)$  represents the three higher beads each with a weight of sixteen ( $4^2$ ).  $(M_2M_1M_0)$  represents the three middle beads each with a weight of four ( $4^1$ ), and  $(L_2L_1L_0)$  represents three lower beads each with a weight of one ( $4^0$ ).

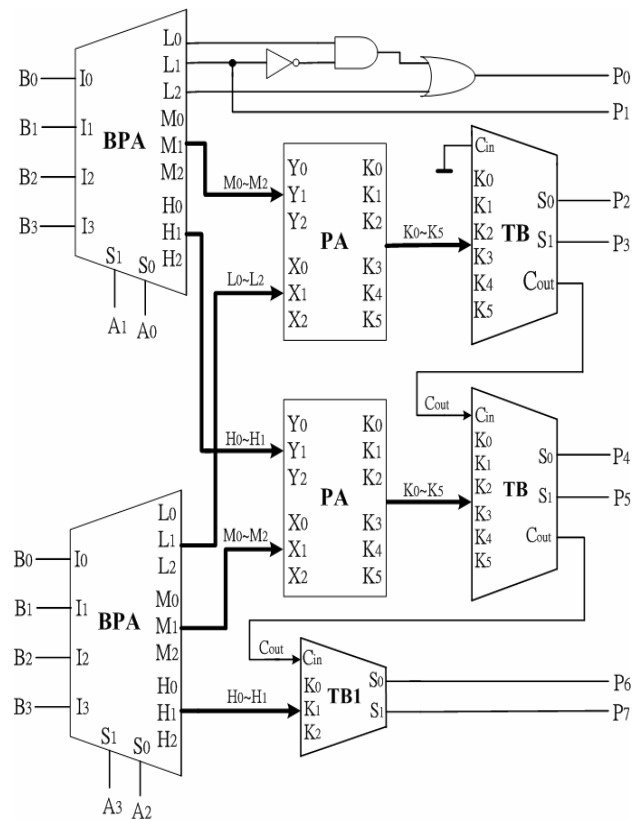


Fig.3 Block diagram of the 4x4 abacus multiplier.

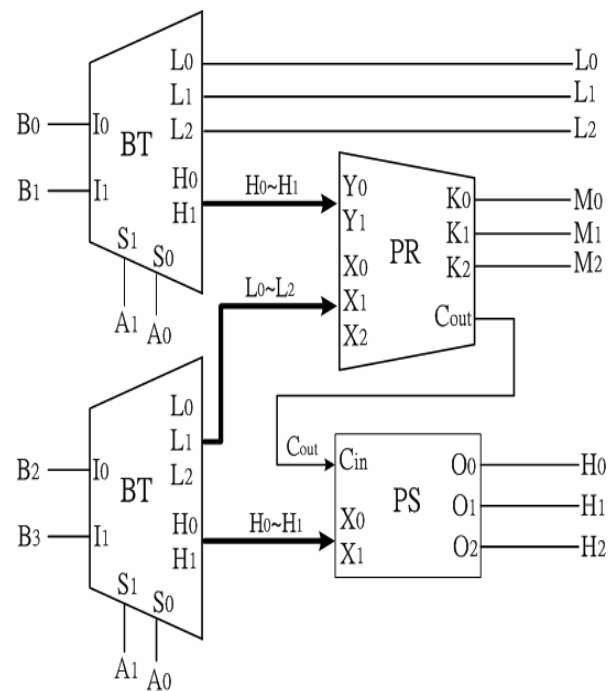


Fig.4 Block diagram of the BPA module.

As shown in Fig.4, the BPA module also consists of three different sub-modules. The behavior of the BT module is modeled in equations (1) - (5):

$$L_0 = (I_1 + I_0)(\overline{S_1} \cdot S_0) + (I_0)(S_1 \cdot \overline{S_0}) + (I_1 + I_0)(S_1 \cdot S_0) \quad (1)$$

$$L_1 = (I_1)(\overline{S_1} \cdot S_0) + (I_0)(S_1 \cdot \overline{S_0}) + (I_1 \oplus I_0)(S_1 \cdot S_0) \quad (2)$$

$$L_2 = (I_1 \cdot I_0)(\overline{S_1} \cdot S_0) + (\overline{I_1} \cdot I_0)(S_1 \cdot S_0) \quad (3)$$

$$H_0 = I_1 \cdot S_1 \quad (4)$$

$$H_1 = (I_1 \cdot I_0)(S_1 \cdot S_0) \quad (5)$$

The PR module adds the beads with the same weight and then transforms the beads into middle beads ( $K_2K_1K_0$ ). The behavior of the PR module can be modeled in equations (6) -(11).

$$f_1 = \overline{X_2} \cdot X_1 \quad (6)$$

$$f_2 = \overline{X_1} \cdot X_0 \quad (7)$$

$$C_{out} = (Y_1) f_1 + (0) f_2 + (0) \overline{X_0} + (Y_0) X_2 \quad (8)$$

$$K_0 = (\overline{Y_1}) f_1 + (1) f_2 + (Y_0) \overline{X_0} + (Y_1 + \overline{Y_0}) X_2 \quad (9)$$

$$K_1 = (\overline{Y_1}) f_1 + (Y_0) f_2 + (Y_1) \overline{X_0} + (Y_0) X_2 \quad (10)$$

$$K_2 = (\overline{Y_1} \cdot Y_0) f_1 + (Y_1) f_2 + (0) \overline{X_0} + (\overline{Y_0}) X_2 \quad (11)$$

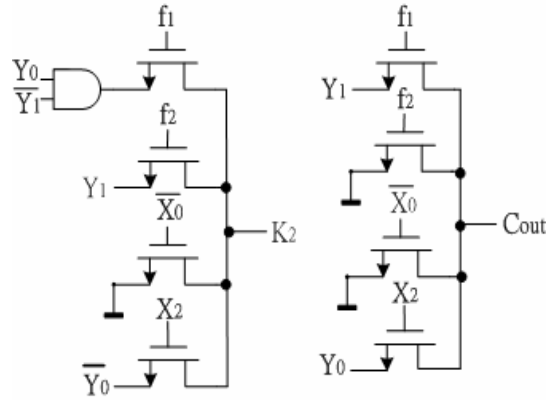
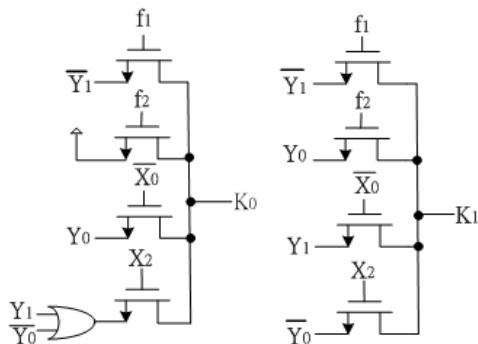


Fig.5 Detailed circuit of the PR module.

The detailed circuit of the PR module is depicted in Fig.5, The PS module transfers the previous stage to higher beads. The behavior of the PS module is modeled in equations (12) - (14).

$$O_0 = X_0 + C_{in} \quad (12)$$

$$O_1 = X_1 + C_{in} X_0 \quad (13)$$

$$O_2 = 0 \quad (14)$$

An example to demonstrate the algorithm of the BPA is shown below:

$(B_3B_2B_1B_0)_2 = (1101)_2 = 13$  and  $(A_1A_0)_2 = (10)_2 = 2$ .  
 $(1101)_2 * (10)_2 = (001|011|011)_{abacus} = (0+0+1)*16 + (0+1+1)*4 + (0+1+1)*1 = 26$ .

## 2.2 The PA (Parallel Addition) Module

This module acts similarly as a multiplexer, which can count two column elements with the same weight and then transform the sum into a thermometric representation  $K_0 \sim K_5$ , in which  $0 \leq K_i \leq K_j \leq 1$  for  $i > j$ .

As shown in Fig. 3, the number  $(X_2X_1X_0)$  is the input signal of the multiplexer, and  $(Y_2Y_1Y_0)$  is the selector used to modify the number  $(X_2X_1X_0)$ . The result gives a thermometric sum  $(K_5K_4K_3K_2K_1K_0)$ . Note that there are only four configurations for each number  $(X_2X_1X_0)$  or  $(Y_2Y_1Y_0)$ , i.e., 000, 001, 011, and 111.

The behavior of the PA module can be modeled in equations (15) – (22), and the detailed circuits, as shown in Fig.6. The PA module can count all the beads simultaneously.

$$f_1 = \overline{X_2} \cdot X_1 \tag{15}$$

$$f_2 = \overline{X_1} \cdot X_0 \tag{16}$$

$$K_0 = (1) f_1 + (1) f_2 + (Y_0) \overline{X_0} + (1) X_2 \tag{17}$$

$$K_1 = (1) f_1 + (Y_0) f_2 + (Y_1) \overline{X_0} + (1) X_2 \tag{18}$$

$$K_2 = (Y_0) f_1 + (Y_1) f_2 + (Y_2) \overline{X_0} + (1) X_2 \tag{19}$$

$$K_3 = (Y_1) f_1 + (Y_2) f_2 + (0) \overline{X_0} + (Y_0) X_2 \tag{20}$$

$$K_4 = (Y_2) f_1 + (0) f_2 + (0) \overline{X_0} + (Y_1) X_2 \tag{21}$$

$$K_5 = (0) f_1 + (0) f_2 + (0) \overline{X_0} + (Y_2) X_2 \tag{22}$$

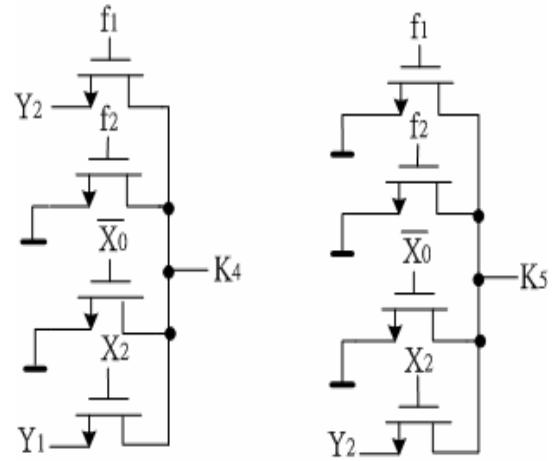
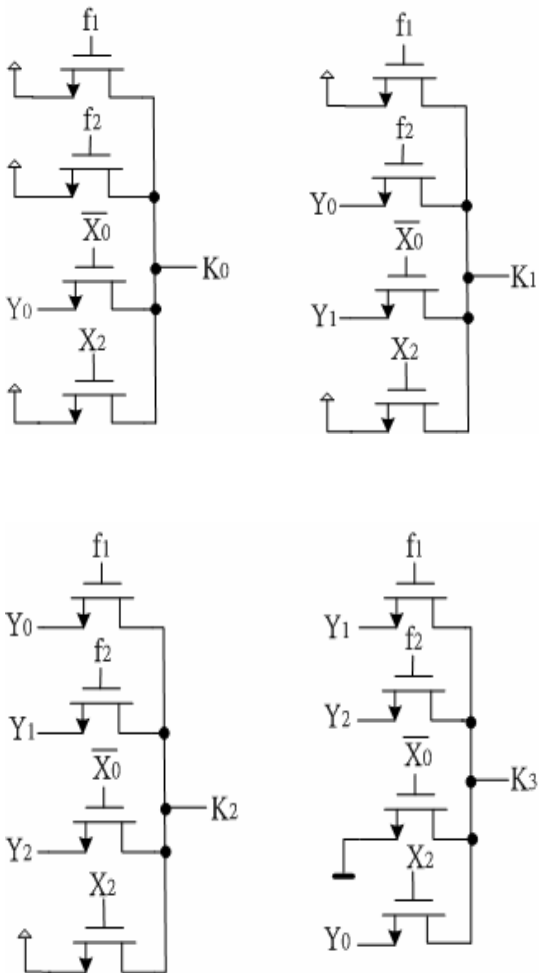


Fig.6 The circuit of (K<sub>0</sub>K<sub>1</sub>K<sub>2</sub>K<sub>3</sub> K<sub>4</sub>K<sub>5</sub>) in the PA module.



### 2.3 The TB (Thermometric to Binary) Transformation Module

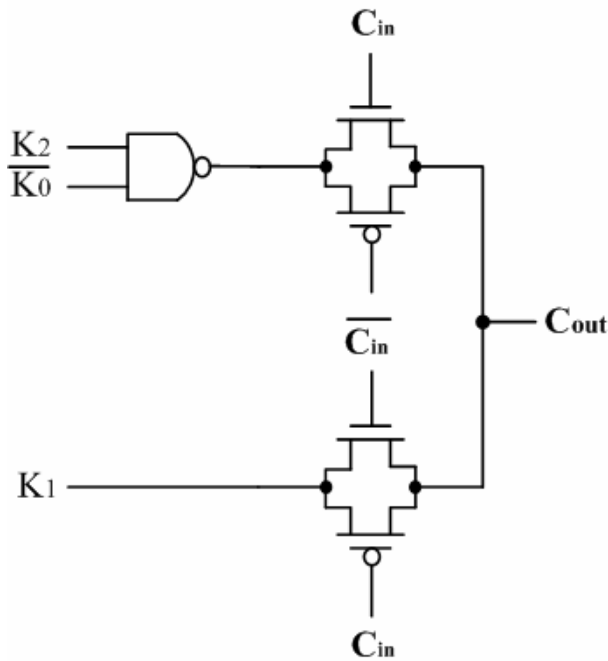
This module transforms the thermometric representation to binary numbers. It can convert the higher part or lower part numbers of K<sub>5</sub> - K<sub>0</sub> to binary numbers as shown in Fig.3. The output signals S<sub>1</sub>, S<sub>0</sub> and C<sub>out</sub> are determined using the following equations:

$$C_{out} = C_{in} K_2 + K_3 \tag{23}$$

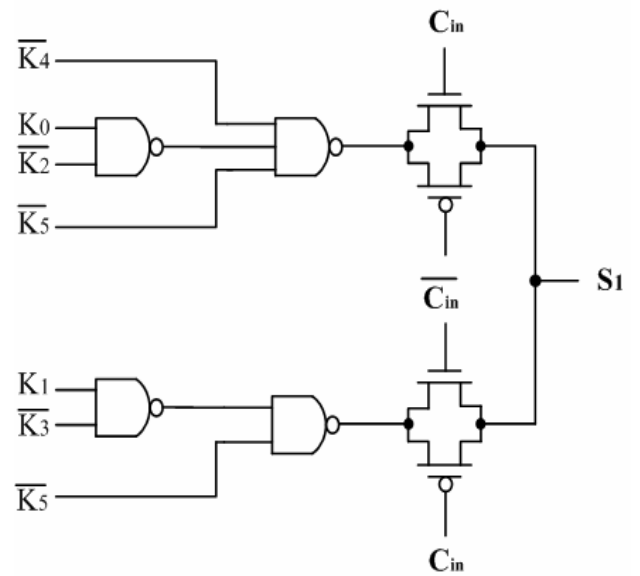
$$S_0 = \overline{C_{in}} \overline{K_1} K_0 + \overline{C_{in}} \overline{K_3} K_2 + \overline{C_{in}} \overline{K_5} K_4 + C_{in} \overline{K_0} + C_{in} \overline{K_2} K_1 + C_{in} \overline{K_4} K_3 + C_{in} K_5 \tag{24}$$

$$S_1 = \overline{C_{in}} \overline{K_3} K_1 + C_{in} \overline{K_2} K_0 + C_{in} K_4 + K_5 \tag{25}$$

The detailed circuits of the TB module are depicted in Fig.7.

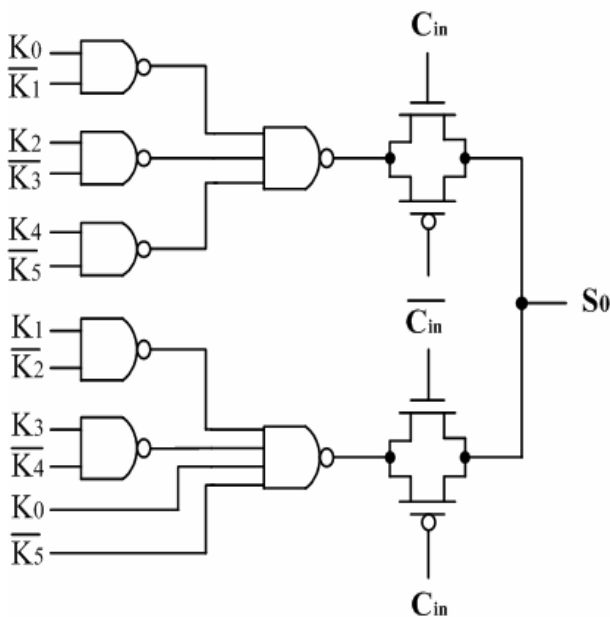


(a)



(c)

Fig. 7 (a) The circuit of  $C_{out}$  in TB block, (b) The circuit of  $S_0$  in TB block, (c) The circuit of  $S_1$  in TB block.



(b)

### 3 High-Bit Chinese Abacus Multiplier Architecture

The Chinese abacus multiplier architecture can easily be reconfigured to higher-bit multipliers (e.g.,  $8 \times 8$ ,  $16 \times 16$ ,  $32 \times 32 \dots$ ). For a  $n$ -bit ( $n=4, 8, 16, 32 \dots$ ) multipliers we need to design two BPA modules,  $n/2$  PA modules, and two  $n/2$  TB modules. Since the four  $(n/2) \times (n/4)$  BPA modules can be combined together to create the  $n \times (n/2)$  BPA module, thus the minimum size of the BPA module is  $4 \times 2$ . For instance an  $8 \times 8$  multiplier is given in Fig.8. It carries out the multiplication with three steps: (1) Binary product to abacus, (2) Parallel addition, and (3) Thermometric to Binary. The circuit includes two  $8 \times 4$  BPA modules, four PA modules, and four TB modules. These PA modules and TB modules are the same as the proposed in subsections 2.2 and 2.3, respectively. The four  $4 \times 2$  BPA modules can constitute a  $8 \times 4$  BPA module.

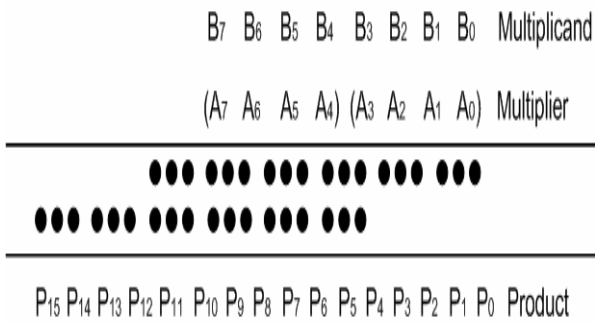


Fig.8 8x8 Chinese abacus multiplier

**3.1  $n \times (n/2)$  BPA module:**

The four  $4 \times 2$  BPA modules can be combined together to create the  $8 \times 4$  BPA module in Fig.9. In the  $8 \times 4$  BPA module, we need to design two new cells, a PTA and a PA3\_3 cells. And then, the other blocks can be composed of the PTA\_cell and PA3\_3.

**3.2 PA7\_4 Module:**

Fig.10 shows the block diagram of the PA7\_4 cell. It consists of a PA module and a PTA\_cell. The PA module counts two column elements with the same weight and then transforms them into thermometric representations  $K_0 \sim K_5$ . Then, the PTA\_cell transforms the thermometrics  $K_0 \sim K_5$  and the carry bit  $C_{in}$  into the output denoted by  $(C_{out}S_2S_1S_0)$ . The output consists of four beads, which are, respectively, weighted with a number for each bead, for example, the weighting of  $(S_2S_1S_0)$  is one, and that of  $C_{out}$  is four. Meanwhile, there are fourteen conditions for the PTA\_cell. The others are don't care that truth table shown in Table 1. The behavior of the PTA\_cell is modeled in equations (26) – (29), and detailed circuits of the PTA\_cell are depicted in Fig.11.

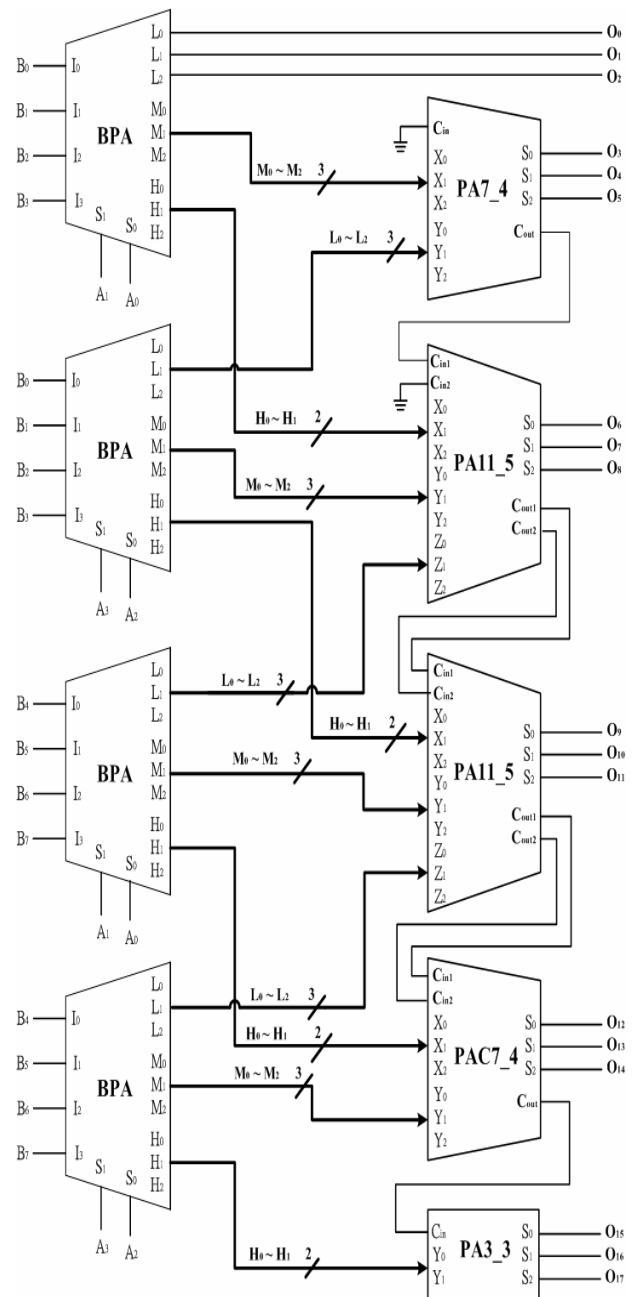
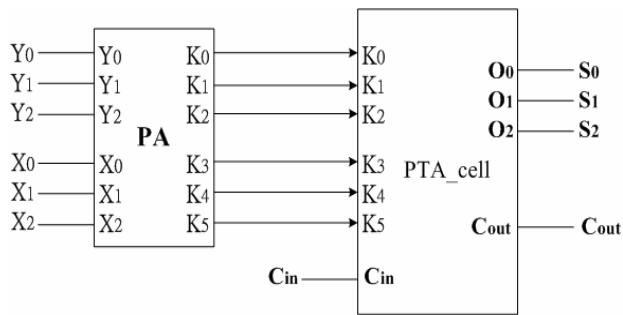


Fig.9 Block diagram of the 8x4 BPA module



$$C_{out} = K_2 C_{in} + K_3 \overline{C_{in}} \tag{26}$$

$$S_0 = (K_3 + \overline{K_2}) C_{in} + K_0 (K_4 + \overline{K_3}) \overline{C_{in}} \tag{27}$$

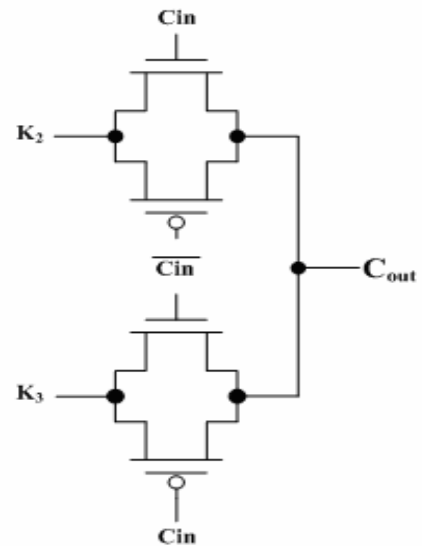
$$S_1 = K_0 (K_4 + \overline{K_2}) C_{in} + K_5 \overline{C_{in}} + \overline{K_3} K_1 \overline{C_{in}} \tag{28}$$

$$S_2 = K_1 (K_5 + \overline{K_2}) C_{in} + \overline{K_3} K_2 \overline{C_{in}} \tag{29}$$

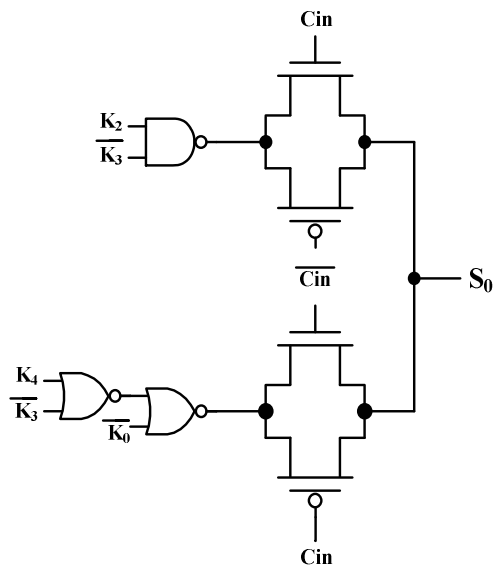
Fig.10 the block diagram of the PA7\_4 module

Table 1 Truth table of PTA\_cell

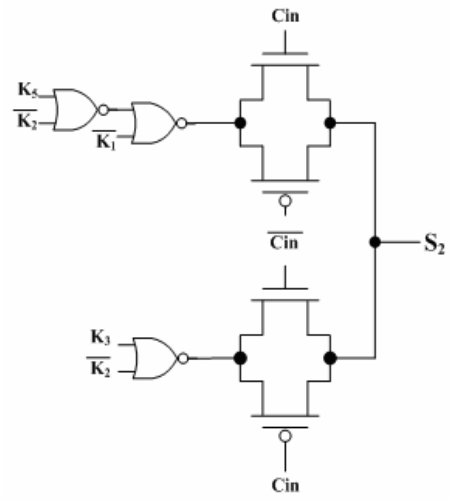
Inputs	Outputs			
$K_5, K_4, K_3, K_2, K_1, K_0, C_{in}$	$C_{out}$	$O_2$	$O_1$	$O_0$
0, 0, 0, 0, 0, 0, 0	0	0	0	0
0, 0, 0, 0, 0, 0, 1	0	0	0	1
0, 0, 0, 0, 0, 1, 0	0	0	0	1
0, 0, 0, 0, 0, 1, 1	0	0	1	1
0, 0, 0, 0, 1, 1, 0	0	0	1	1
0, 0, 0, 0, 1, 1, 1	0	1	1	1
0, 0, 0, 1, 1, 1, 0	0	1	1	1
0, 0, 0, 1, 1, 1, 1	1	0	0	0
0, 0, 1, 1, 1, 1, 0	1	0	0	0
0, 0, 1, 1, 1, 1, 1	1	0	0	1
0, 1, 1, 1, 1, 1, 0	1	0	0	1
0, 1, 1, 1, 1, 1, 1	1	0	1	1
1, 1, 1, 1, 1, 1, 0	1	0	1	1
1, 1, 1, 1, 1, 1, 1	1	1	1	1



(a)

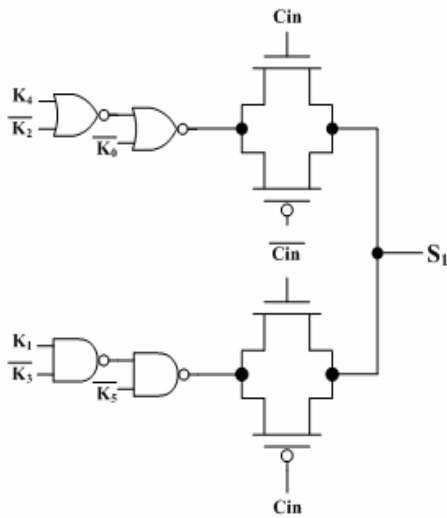


(b)



(d)

Fig.11 (a) The circuit of  $C_{out}$  in PTA block, (b) the circuit of  $S_0$  in PTA block, (c) the circuit of  $S_1$  in PTA block, and (d) the circuit of  $S_2$  in PTA block.



(c)

### 3.3 PA11\_5 Module

Fig.12 shows the block diagram of the PA11\_5 cell. It consists of two PA7\_4 cells. The input of the module consists of three column elements and two carry in bit ( $C_{in1}C_{in2}$ ) with the same weight. The output consists of five beads, each with a weight of one for ( $S_2S_1S_0$ ) and four for ( $C_{out1}C_{out2}$ ).

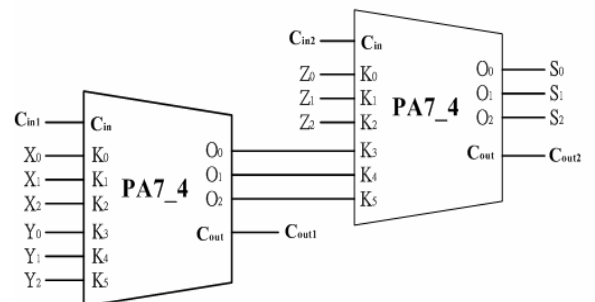


Fig.12 Block diagram of the PA11\_5 cell



### 3.4 PA3\_3 Module

PA3\_3 transforms two beads  $Y_1Y_0$  and the carry bead  $C_{in}$  into the output  $(S_2S_1S_0)$  weighted by one. The truth table for PA3\_3 is shown in Table 2. The behavior of the PA3\_3 module is modeled in the equations (30) - (32), and the detailed circuits of PA3\_3 are depicted in Fig.13.

Table 2 the truth table of PA3\_3

Inputs	Outputs		
$Y_1, Y_0, C_{in}$	$S_2$	$S_1$	$S_0$
0, 0, 0	0	0	0
0, 0, 1	0	0	1
0, 1, 0	0	0	1
0, 1, 1	0	1	1
1, 0, 0	*	*	*
1, 0, 1	*	*	*
1, 1, 0	0	1	1
1, 1, 1	1	1	0

$$S_0 = (1)C_{in} + Y_0\overline{C_{in}} \tag{30}$$

$$S_1 = Y_0C_{in} + Y_1\overline{C_{in}} \tag{31}$$

$$S_2 = Y_1C_{in} + (0)\overline{C_{in}} \tag{32}$$

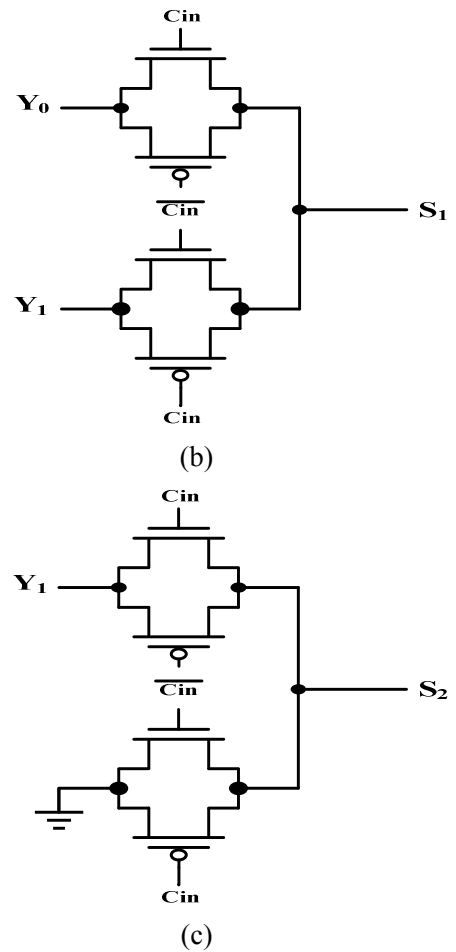
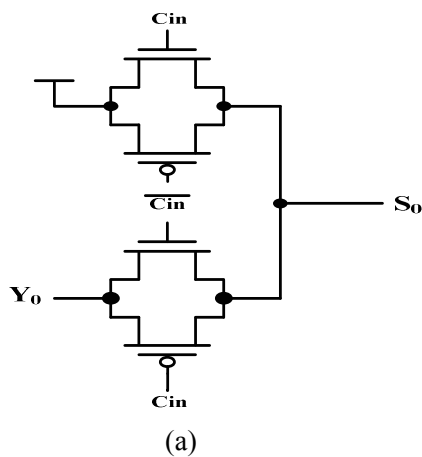


Fig. 13 (a) The circuit of  $S_0$  in PA3\_3 block, (b) the circuit of  $S_1$  in PA3\_3 block, and (c) the circuit of  $S_2$  in PA3\_3 block.

### 3.4 PAC7\_4 Module

Fig.14 shows the block diagram of the PAC7\_4 module. It consists of the PA7\_4 cell and PA3\_3 cell. This module is composed of two column elements,  $(Y_1Y_0)$  and  $(X_2X_1X_0)$ , and two carry bits  $(C_{in1}C_{in2})$  with the same weight. The output consists of four beads, each with a weight of one for  $(S_2S_1S_0)$  and four for  $(C_{out})$ .

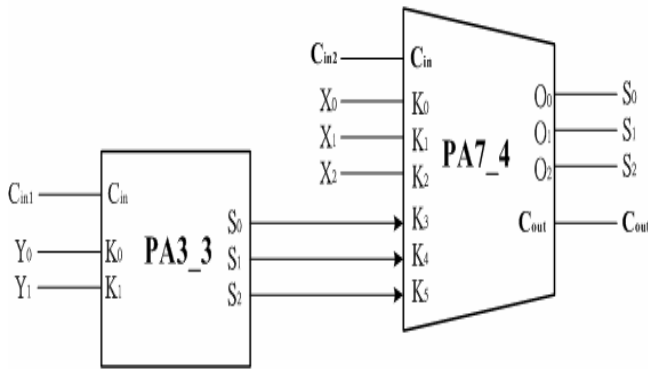


Fig.14 Block diagram of the PAC7\_4 cell

### 4 Simulations and Comparisons

In the previous sections we have designed a multiplier using the methodology of the Chinese abacus. The circuits of the prototype are simulated using HSPICE and the 0.35 $\mu$ m and 0.18 $\mu$ m TSMC CMOS technologies. For the sake of simplicity, all the lengths and widths of the transistors were set to be the smallest value allowed by each technology, and the width size of PMOS transistors was 2.5 times of that of NMOS transistors. CMOS inverters were used as loads in each output in the simulations. The simulation was performed at a frequency of 50 MHz, with 0.35 $\mu$ m CMOS technology. All of the output and input waves are shown in Fig.15. The power consumption wave is also shown in Fig.16.

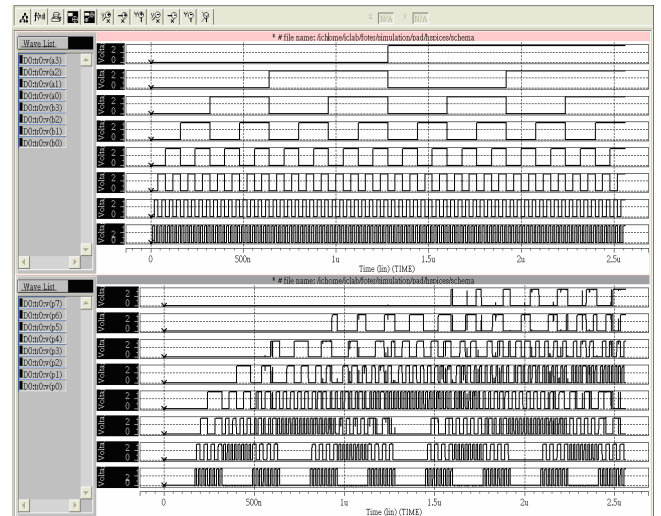


Fig.15 Simulation output and input waves of the 4-bit Chinese abacus multiplier

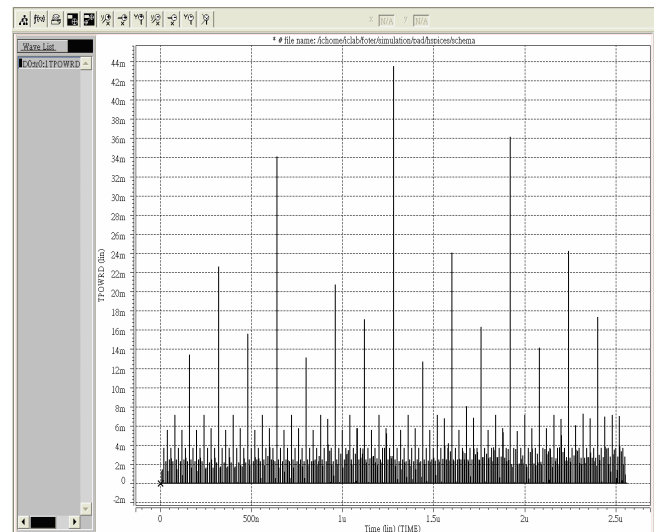


Fig.16 Simulation power consumption of the 4-bit Chinese abacus multiplier.

The simulation results are compared with those of the Braun array multiplier, as listed in Table 3 and Table 4. The delay is defined as the longest signal time from input to output with all input patterns. The 4x4 abacus multiplier results a delay of 2.83ns and 1.432ns for 0.35 $\mu$ m and 0.18 $\mu$ m TSMC CMOS technologies, respectively. These data are 19.7% and 10.6% less than those of the Braun array multiplier for the same 0.35 $\mu$ m and 0.18 $\mu$ m technologies, respectively. Also the delays of the 8-bit abacus

multiplier are 14% and 7.5% less than those of the Braun array multiplier with 0.35 $\mu$ m and 0.18 $\mu$ m technologies, respectively.

The power consumption levels of various multipliers using different methodologies is also listed in Table 3 and Table 4. The average results of the power consumption are different for all input patterns. The simulation was performed at a frequency of 50MHz. For the 4-bit abacus multiplier power consumptions were 8.7 % and 18% less than those of the Braun array multiplier with 0.35 $\mu$ m and 0.18 $\mu$ m technologies, respectively. The power consumption levels of the 8-bit abacus multiplier were 11.9% and 22.3% less than those of the Braun array multiplier with 0.35 $\mu$ m and 0.18 $\mu$ m technologies, respectively. From these results, we conclude that the abacus multiplier still has competitive with the Braun array multiplier. The chip layout of the 4x4 abacus multiplier is shown in Fig.17.

TABLE 3. Simulation results of the 4x4 array multiplier and abacus multiplier

	Tech.	[8]	Braun	Abacus	Reduction % (compare to Braun )
<b>Delay (ns)</b>	0.35	-	3.39	2.83	19.7%
	0.18	3.97	1.584	1.432	10.6%
<b>Power (<math>\mu</math>w)</b>	0.35	-	313	288	8.7%
	0.18	145.5	45.2	38.3	18%
<b>P*D (<math>\mu</math>w*ns)</b>	0.35	-	1061	815	30%
	0.18	577.6	71.6	54.8	30%

TABLE 4. Simulation results of 8x8 array Multiplier and abacus multiplier

	Tech.	[10]	Braun	Abacus	Reduction % (compare to Braun )
<b>Delay (ns)</b>	0.35	-	7.69	6.75	14%
	0.18	6.98	3.60	3.35	7.5%
<b>Power (mw)</b>	0.35	-	1.52	1.36	11.9%
	0.18	0.188	0.216	0.179	22.3%
<b>P*D (mw*ns)</b>	0.35	-	11.69	9.18	27%
	0.18	1.312	0.778	0.599	29.8%

## 4 Conclusion

We propose a multiplier based on the Chinese abacus algorithm that all the results are simulated by 0.18 $\mu$ m and 0.35 $\mu$ m TSMC CMOS technologies, respectively. As can be concluded from the simulation results, the delays level of the 8-bit abacus multiplier are 14% and 7.5% less than those of Braun array multiplier with 0.35 $\mu$ m and 0.18 $\mu$ m technologies, respectively; furthermore, the power consumption of the 8-bit abacus multiplier is about 11.9 % and 22.3% less than those of the Braun array multiplier with 0.35 $\mu$ m and 0.18 $\mu$ m technologies. Obviously, this proposed abacus multiplier can significantly reduce the delay and the power consumption that it owns the competitive ability with respect to conventional fast multipliers.

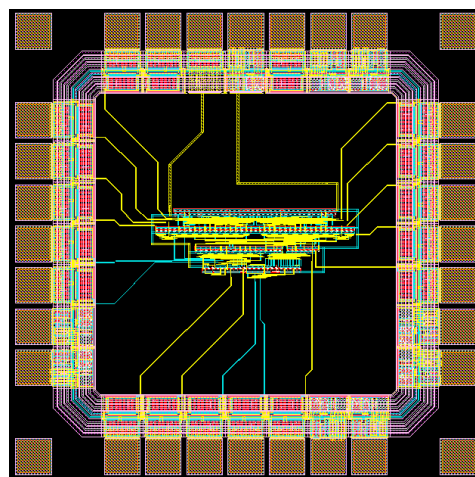


Figure 17 Chip layout of the 4x4 abacus multiplier using 0.35 $\mu$ m TSMC CMOS technology.

### References:

- [1] S. D. Pezaris, A 40-ns 17-bit by 17-bit array multipliers, *IEEE Transactions on Computers*, Vol. 20, April 1971, pp. 442-447.
- [2] K.Z. Pekmestzi, Multiplexer-based array multipliers, *IEEE Transactions on Computers*, Vol. 48, No. 1, Jan. 1999, pp. 15-23.
- [3] C. Wallace, A suggestion for a fast multiplier, *IEEE Transactions on Electronic Computers*, Vol. 13, 1964, pp. 14-17.
- [4] Franco Maloberti and Chen Gang, The Chinese Abacus method: can we use it for digital

- arithmetic, *Proceedings of the 8th Great Lakes Symposium on VLSI*, 19-21, Feb. 1998 pp. 192 – 195.
- [5] Franco Maloberti and Chen Gang, Use of the Chinese Abacus method for digital arithmetic functions, *Proceedings of the 1998 IEEE International Symposium on Circuits and Systems*, Vol. 5, 31 May - 3 June 1998 pp. 213 – 216.
- [6] Franco Maloberti and Chen Gang, Performing Arithmetic Functions with the Chinese Abacus Approach, *IEEE Transaction on circuits and systems-II: Analog and digital signal processing*, Vol. 46, No. 12, Dec. 1999, pp. 1512 – 1515.
- [7] Shu-Chung Yi., Kun-Tse Lee., Jin-Jia Chen., Chien-Hung Lin., Chuen-Ching Wang., Chin-Fa Hsieh and Chih-Yung Lu., The new architecture of radix-4 Chinese abacus adder, *IEEE International Symposium on Multiple-Valued Logic*, 17-20 May. 2006, pp. 12-15.
- [8] Vasefi, F., and Abid, Z, Low power n-bit adders and multiplier using lowest-number-of-transistor 1-bit adders, *Canadian Conference on Electrical and Computer Engineering*, May. 2005, pp. 1731 – 1734.
- [9] Shu-Chung Yi, Zi-Yi Zhao, Chien-Hung Lin, Yu-Zhi Xie, Yen-Ju Chen, Yi-Jie Lin, The novel Chinese abacus adder, *IEEE International Symposium on VLSI Design, Automation and Test (VLSI-DAT)*, Ambassador Hotel, Hsinchu, Taiwan, April 25-27, 2007, pp. 270–273.
- [10] Van, L.-D., and Chih-Chyau Yang., Generalized Low-Error Area-Efficient Fixed-Width Multipliers, *IEEE Transactions on Circuits and Systems*, Vol. 52, Aug. 2005, pp. 1608 – 1619.