A Design of 16 ×16-bit Redundant Binary MAC Using 0.25 ㎛ CMOS Technology

Tae-Min Kim*·Gun-Soon Shin**

이 논문은 2002 년도 금오공과대학교 학술연구비와 IDEC 의 CAD TOOL 을 지원 받았음

ABSTRACT

In this paper, a 16×16-bit Multiplier and Accumulator (MAC) is designed using a Redundant Binary Adder (RBA) circuit so that it can make a fast addition of the Redundant Binary Partial Products (RB_PP's) by using Wallace-tree structure. Because a RBA adds two RB numbers, it acts as a 4-2 compressor, which reduces four inputs to two output signals. We propose a method to convert the Redundant Binary (RB) representation into the 2's complement binary representation. Instead of using the conventional full adders, a more efficient RB number to binary number converter can be designed with new conversion method.

요 약

승산기의 부분곱을 가장 빠르게 연산하기 위해 Wallace-tree 방법이 사용된다. Redundant Binary Adders (RBAs)을 이용한 Wallace-tree 연산은 두개의 Redundant Binary Number (RBN)을 캐리 전송 없이 빠르게 더하여 하나의 RBN 를 만든다. Redundant Binary Partial Products (RB_PPs)을 합하는 RBA 가 승산기 면적의 대부분을 차지할 뿐만 아니라 동작 속도를 결정한다. 본 논문에서는 제안된 Redundant Binary (RB) 덧셈 알고리듬을 이용하여 가장 적은 트랜지스터 개수를 갖는 RBA 가 설계되고 제안된 RBA 을 사용하여 16 ×16-bit Multiplier and Accumulator (MAC)을 구현한다.

I. INTRODUCTION

Multipliers are fundamental components of digital hardware. They occupy a relatively large portion of the overall chip area and have often been the limiting factor in terms of speed. A simplicity of design and low transistor count are also required for the multipliers because of the increase in the bit size of the number to be calculated. The Wallace-tree method is commonly used to realize high speed because it is theoretically the fastest method [1]. Generated Redundant Binary Partial Products (RB_PP's) are added up by the Wallace-tree of Redundant Binary Adders (RBA's). Because an RBA adds two RB numbers to make one RB number, four inputs are reduced to two output signals. The RBA acts as a 4-2 compressor. The array of a RBA tree can increase operating speed by use of high speed RBA. Many algorithms and circuits have been reported for RBA [2-3].

The Wallace-tree of a RBA array adds the partial products until the final RB number is obtained. Then the final RB number must be converted to a NB number, that is the product, by an Redundant Binary to Normal Binary (RB-to- NB) converter. We present a new method for

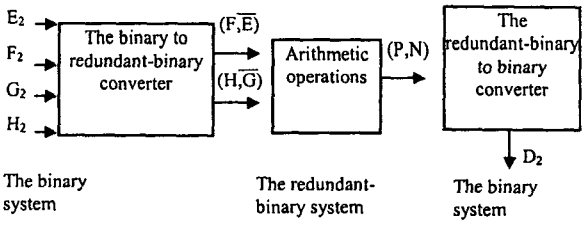


Fig. 1. Conversion between the binary and the redundant-binary systems [6].

**금오공과대학교 전자공학부

*구미 1 대학 정보통신과

접수일자 : 2002. 12. 18

converting the RB on to the 2's complement binary representation. The method requires less VLSI chip area and takes less conversion time than the conventional method [2-4].

The sign extension can be reduced by the modified sign extension with no additional circuit. There is no need to consider sign extension for partial products addition.

In the next section the details of this architecture are explained. To realize higher speed RB multiplier than conventional Normal Binary (NB) multipliers, the architecture must be optimized to CMOS circuit by taking advantage of RB number. In section Ⅲ, the design and simulation of a 16×16-bit MAC employing this architecture are described. In section Ⅳ, this paper is concluded.

Ⅱ. RB ARCHITECTURE

A. *Partial Products Generation*

RB architecture uses the well known method as shown in reference [5]. This is done according to the following consideration. The addition of A to B is expressed as

$$\begin{aligned} A+B &= A-(-B)_{\bar{2}} \\ &= A-(\bar{B}+1) \\ &= (A-\bar{B})-1 \\ &= (-1)(a_{n-1}-\bar{b}_{n-1})\cdot 2^{n-1}+\sum_{k=0}^{n-2}(a_k-\bar{b}_k)\cdot 2^k-1 \\ &= \sum_{k=0}^{n-1} d_k\cdot 2^k-1 \end{aligned}$$

A = 0 0 1 1 0 1 0 1 (53)
B = 1 0 0 1 1 1 0 1 (-99)
A×B = -5247

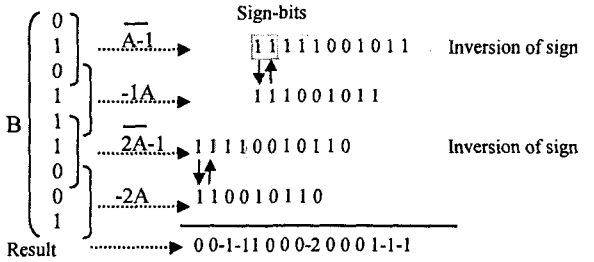


Fig. 2. An example of the RB_PP's multiplication scheme without the sign extension

$$= \sum_{k=0}^{n-1} d_k\cdot 2^k+(0,1) \qquad (1)$$

Where $\bar{B}$ is the inversion of B. Then the subtraction $A-\bar{B}$ becomes one of the following four forms whose values are 1, 0, or -1 :

$$(1,0)=1\ (1,1)=(0,0)=0 \text{ and } (0,1)=-1 \qquad (2)$$

Thus, the RB partial product, that is equal to the sum of two NB partial products, is generated by inverting one of the two partial products and adding (0,1) to the lowest digit.

The drawback of the Booth algorithm is the sign extension. The use of 2's complement arithmetic to produce negative values of the multiplicand causes the MSB sign bits to be spanned on the entire bit-width when a signal changes sign. Fig. 2 exemplifies the RB_PP' multiplication scheme without the sign extension.

B. *Addition in Redundant Binary Representation*

One important property of the redundant binary representation is its carry propagation free addition. This is the reason that RB adders usually are used to realize fast arithmetic operations. Another important property of the RB representation is that it does not use the 2's complement method to handle negative numbers, therefore, multiplication and division operations can be performed easily using the representation. However, for the RB representation, the conversion between the binary system and the RB system has to be performed. This is shown in Fig. 1. In the RB representation, the algebraic value of a n-bit number

$$B=(b_{n-1}, b_{n-2}, \cdots, b_1, b_0) \qquad (3)$$

2's complement B may be rewritten as follows :

$$\begin{aligned} B &= -b_{n-1}\cdot 2^{n-1}+\sum_{k=0}^{n-2} b_k\cdot 2^k \\ &= 2\times\left[-b_{n-1}\cdot 2^{n-1}+\sum_{k=0}^{n-2} b_k\cdot 2^k\right]-\left[-b_{n-1}\cdot 2^{n-1}+\sum_{k=0}^{n-2} b_k\cdot 2^k\right] \\ &= -2\times b_{n-1}\cdot 2^{n-1}+b_{n-1}\cdot 2^{n-1}+\left[\sum_{k=0}^{n-2} b_k\cdot 2^{k+1}-\sum_{k=0}^{n-2} b_k\cdot 2^k\right] \\ &= -b_{n-1}\cdot 2^{n-1}+\left[\sum_{k=0}^{n-2} b_k\cdot 2^{k+1}-\sum_{k=0}^{n-2} b_k\cdot 2^k\right] \\ &= \left[b_{n-2}-b_{n-1}\right]\cdot 2^{n-1}+\sum_{k=0}^{n-3}\left[b_k-b_{k+1}\right]\cdot 2^{k+1}+\left[0-b_0\right] \\ &= \left[b_{n-2}-b_{n-1}\right]\cdot 2^{n-1}+\sum_{k=0}^{n-2}\left[b_{k-1}-b_k\right]\cdot 2^k ;\quad (b_{-1}=0) \\ &= d_{n-1}\cdot 2^{n-1}+\sum_{k=0}^{n-2} d_k\cdot 2^k \\ &= \sum_{k=0}^{n-1} d_k\cdot 2^k \end{aligned} \qquad (4)$$

Where d_k is a binary signed digit ($d_k \in \{=1,0,-1\}$). Therefore, the conversion from an n-bit 2's complement binary representation into its RB representation based on new converter algorithm.

Let us consider four numbers E, F, G and H represented in 2's complement form and their RB product D expressed by (1) and (4).

$$E = -e_{n-1}\cdot 2^{n-1} + \sum_{k=0}^{n-2} e_k \cdot 2^k \quad (5.a)$$

$$F = -f_{n-1}\cdot 2^{n-1} + \sum_{k=0}^{n-2} f_k \cdot 2^k \quad (5.b)$$

$$G = -g_{n-1}\cdot 2^{n-1} + \sum_{k=0}^{n-2} g_k \cdot 2^k \quad (5.c)$$

$$H = -h_{n-1}\cdot 2^{n-1} + \sum_{k=0}^{n-2} h_k \cdot 2^k \quad (5.d)$$

$$\begin{aligned} D &= E + F + G + H + (c_0^+ + c_0^-) \\ &= (F, \overline{E}) + (H, \overline{G}) + (c_0^+ + c_0^-) - 2 \end{aligned} \quad (6)$$

By substituting (5) into (6) and using 2's complement number characteristics, D becomes

$$\begin{aligned} D &= (F - \overline{E}) + (H - \overline{G}) + (c_0^+ + c_0^-) - 2 \\ &= (-f_{n-1}\cdot 2^{n-1} - \textstyle\sum_{k=0}^{n-2} \overline{e}_k \cdot 2^k + \overline{e}_{n-1}\cdot 2^{n-1} + \sum_{k=0}^{n-2} f_k \cdot 2^k \\ &\quad -h_{n-1}\cdot 2^{n-1} - \textstyle\sum_{k=0}^{n-2} \overline{g}_k 2^k) + \overline{g}_{n-1}\cdot 2^{n-1} + \sum_{k=0}^{n-2} h_k \cdot 2^k + (c_0^+ + c_0^-) - 2 \\ &= ([2\cdot c_{n-1}^- + s_{n-1}^+] \cdot 2^{n-1} + \textstyle\sum_{k=0}^{n-2} [2\cdot c_k^- + s_k^+] \cdot 2^k) \\ &\quad + \overline{g}_{n-1}\cdot 2^{n-1} + \sum_{k=0}^{n-2} h_k \cdot 2^k + (c_0^+ + c_0^-) - 2 \\ &= c_{n-1}^- \cdot 2^n + \left[c_{n-2}^- + s_{n+1}^+ + \overline{g}_{n-1}\right] \cdot 2^{n-1} \\ &\quad + \textstyle\sum_{k=0}^{n-3} \left[c_k^- + s_{k+1}^+ + h_{k+1}\right] \cdot 2^{k+1} + s_0^+ + h_0 + (c_0^+ + c_0^-) - 2 \end{aligned}$$

case	$(F_k, \overline{E}_k)$ $(H_k, \overline{G}_k)$	C_{k-1}^-	Carry Sum (C_k^+, C_k^-) (C_{k-1}^+, S_k^-)
1	(0,0) (1,1) (1,0) (0,0) (0,0) (0,1)	1	(0,0) (c_{K-1}^+,1)
		0	(0,0) (c_{K-1}^+,0)
2	(1,0) (0,0) (1,1) (1,0) (0,0) (1,0) (1,0) (1,1)	1	(0,0) (c_{K-1}^+,0)
		0	(1,0) (c_{K-1}^+,1)
3	(1,0) (1,0)	1	(1,0) (c_{K-1}^+,1)
		0	(1,0) (c_{K-1}^+,0)
4	(0,1) (0,0) (1,1) (0,1) (0,0) (0,1) (0,1) (1,1)	1	(0,1) (c_{K-1}^+,0)
		0	(1,1) (c_{K-1}^+,1)
5	(0,1)(0,0) (1,1) (1,0) (1,1) (1,1)	1	(1,1) (c_{K-1}^+,1)
		0	(1,1) (c_{K-1}^+,0)
6	(0,1) (0,1)	1	(0,1) (c_{K-1}^+,0)
		0	(0,1) (c_{K-1}^+,1)

Fig. 3. Rule for RB Addition. Immediate Carry and sum are shown for the six cases of input $(F_k, \overline{E}_k)$ and $(H_k, \overline{G}_k)$.

$$\begin{aligned} &= c_{n-1}^- 2^n + \left[2\cdot c_{n-1}^+ + s_{n-1}^-\right]\cdot 2^{n-1} \\ &\quad + \textstyle\sum_{k=0}^{n-3} \left[2\cdot c_{k+1}^+ + s_{k+1}^-\right]\cdot 2^{k+1} + 2\cdot c_0^+ + s_0^- + (c_0^+) - 2 \\ &= \left[c_{n-1}^+ + c_{n-1}^-\right]\cdot 2^n + \textstyle\sum_{k=0}^{n-2} \left[c_k^+ + s_{k+1}^-\right]\cdot 2^{k+1} + (c_0^+ + s_0^-) - 2 \\ &= d_n \cdot 2^n + \textstyle\sum_{k=0}^{n-1} d_k \cdot 2^k - 2 \\ &= \textstyle\sum_{k=0}^{n} d_k \cdot 2^k - 2 \end{aligned} \quad (7)$$

Using this Eq.(4), we define four numbers, c^-, c^+, s^- and s^+ as given in (7).

$$f_k - \overline{e}_k - \overline{g}_k = 2\cdot c_k^- + s_k^+ \quad (8)$$

By substituting (8) into (6), d becomes

$$d = 2\cdot c_k^- + s_k^+ + h_k + (c_{k-1}^+ + c_{k-1}^-) - 2 \quad (9.a)$$

$$c_{k-1}^- + s_k^+ + h_k = 2\cdot c_k^+ + s_k^- \quad (9.b)$$

By substituting (9.b) into (9.a), d becomes

$$\begin{aligned} d &= 2\cdot c_k^+ + 2\cdot c_k^- + c_{k-1}^+ + s_k^- - 2 \\ &= 2(c_k^+ + c_k^-) + (c_{k-1}^+ + s_k^-) - 2 \\ &= 2(c_k^+ + c_k^-) + d_k - 2 \end{aligned} \quad (10)$$

We obtain RB representation as (4) (where, $d_k \in \{-1, 0, 1\}$) Finally, the RB number can be expressed as

$$D = \sum_{k=0}^{N} d_k \cdot 2^k \quad (11)$$

C. Improved RBA

Generated RB partial products are added up by the Wallace-tree of RBA's. Because a RBA adds two RB numbers to make one RB number, four inputs are reduced to two output signals. The RBA acts as a 4-2 compressor. We consider the addition of the *k*th digit of two redundant

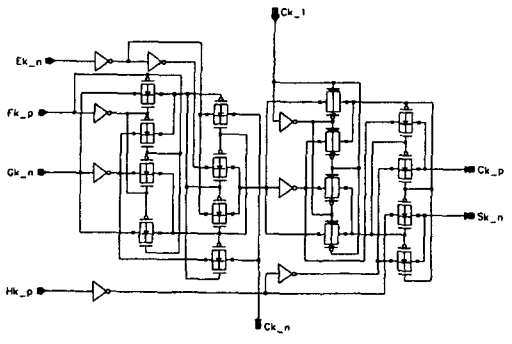
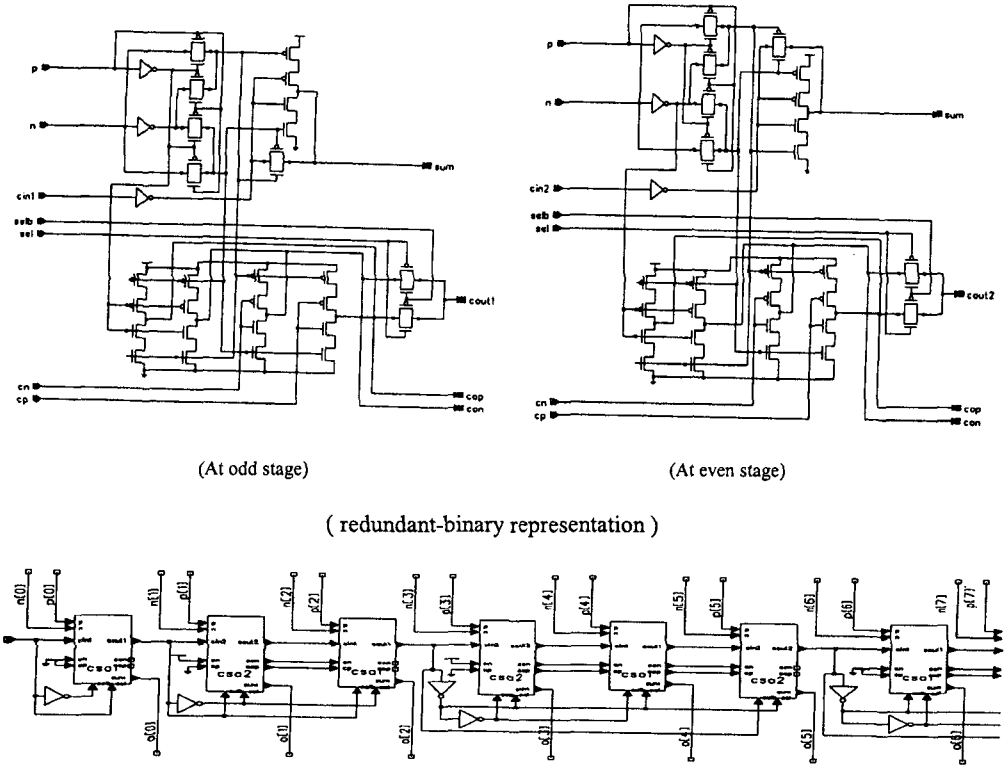


Fig. 4. CMOS circuit diagram of proposed RBA



(At odd stage) (At even stage)

Fig. 6. CMOS circuit diagram of the new redundant-binary to binary converter.

binary numbers $(F_k, \overline{E_k})$ and $(H_k, \overline{G_k})$ expressed by the definition (7) (10).

$$(F_k, \overline{E_k}) + (H_k, \overline{G_k}) + (C_{k-1}^{+} + C_{k-1}^{-}) = 2(C_k^{+} + C_k^{-}) + (d_k^{+}, d_k^{-}) \qquad (12)$$

where (d_k^{+}, d_k^{-}) is the sum. To simplify the consideration, we assume that both the inputs $(F_k, \overline{E_k})$ and $(H_k, \overline{G_k})$ take one of the four states (0,1), (0,0), (1,1) and (1,0). By this assumption, there are sixteen kinds of combination in the sum of $(F_k, \overline{E_k})$ and $(H_k, \overline{G_k})$. They are classified into the six cases by the different results of the addition as shown in Fig. 3. The Fig. shows the intermediate sum

(d_k^{+}, d_k^{-}) and carry (C_k^{+}, C_k^{-}) for each case. The carry is added to the sum of the higher digit.

Fig. 4 shows the CMOS circuit diagram of RBA that realizes the above expressions (7). This consists of inverters and Transmission Gate circuits (TG's) only.

D. RB-to-NB Conversion

The partial products are added in parallel in the Wallace tree of an RBA array until to be a final RB number. The final RB number is converted to an NB number by the final converter part. In the RB representation, the algebraic value of a n-digit number$[d_{n-1}, d_{n-2}, \ldots, d_1, d_0]$ ($d_k \in \{-1, 0, 1\}$) is equal to $\sum_{k=0}^{N-1} d_k \cdot 2^k$, which is the same as the unsigned binary representation. Therefore, the conversion from the RB represetation into its n-bit 2's complement binary number $[b_{n-1}, b_{n-2}, \ldots, b_1, b_0]_2$ ($b_k \in \{0, 1\}$), representation can be realized by using new conversion algorithm Eq.(14). In the 2's complement representation, the algebraic value of an n-digit number B = ($[b_{n-2} - b_{n-1}]$, $\cdots$,$[b_0 - b_1]$, $[0, -b_0]$) written in RB (See Eq. 4).

$$B = \left[b_{n-2} - b_{n-1} \right] \cdot 2^{n-1} + \left[\sum_{k=0}^{n-3} b_k - \sum_{k=0}^{n-3} b_{k+1} \right] \cdot 2^{k+1} - b_0 \cdot 2^0 \qquad (13)$$

B may be rewritten as follows :

$$
\begin{aligned}
B &= \left[b_{n-2} - b_{n-1}\right]\cdot 2^{n-1} + \sum_{k=0}^{n-3}\left[b_k - b_{k+1}\right]\cdot 2^{k+1} + \left[0 - b_0\right] \\
&= \left[b_{n-2} - b_{n-1}\right]\cdot 2^{n-1} + \sum_{k=0}^{n-3}\left[b_k - b_{k+1}\right]\cdot 2^{k+1} + \left[0 - b_0\right] \\
&\quad - \left(b_{n-1}\cdot 2^{n-1} + \sum_{k=0}^{n-3} b_{k+1}\cdot 2^{k+1} + b_0\right) \\
&\quad + \left(b_{n-1}\cdot 2^{n-1} + \sum_{k=0}^{n-3} b_{k+1}\cdot 2^{k+1} + b_0\right) \\
&= \left[b_{n-2} + b_{n-1}\right]\cdot 2^{n-1} + \sum_{k=0}^{n-2}\left[b_{k-1} + b_k\right]\cdot 2^k + \left[0 + b_0\right] \\
&\quad - 2\cdot\left(b_{n-1}\cdot 2^{n-1} + \sum_{k=0}^{n-2} b_k\cdot 2^k + b_0\right) \\
&= -b_{n-1}\cdot 2^n + \sum_{k=0}^{n-1}\left[b_{k-1} + b_k - b_{k-1}\right]\cdot 2^k + \left[0 + b_0\right] \\
&= -b_{n-1}\cdot 2^n + \sum_{k=0}^{n-1} b_k\cdot 2^k \qquad (14)
\end{aligned}
$$

TABLE I The conversion rules in stage *k* shown in Fig. 6.

Input Signal				Output signal	
Redundant binary			Input	Sum	Borrow
D_k	$(d_k^+$,	$d_k^-)$	c_k^-	b_k	c_{k+1}
0	0	0	0	0	0
0	0	0	1	1	1
-1	0	1	0	1	1
-1	0	1	1	0	1
1	1	0	0	1	0
1	1	0	1	0	0
0	1	1	0	0	0
0	1	1	1	1	1

In this equation, the term in brackets is in the set {0, 1}. Therefore, the conversion from an n-bit RB representation into its 2's complement binary representation based on new converter algorithm. It is noted that the new converter does not need any extra operation required by the conventional converter. The conversion is carried out by the addition of d_k^+ to d_k^-, that is the addition of two NB numbers, therefore the carry propagation occurs from lowest to highest digit. For each stage *k*, given the input redundant digit (d_k^+, d_k^-) and the input variable c_k^-, we obtain the binary output B_k and the output variable C_{k+1}. The conversion rules are shown in Table I . It is noted that the new converter does not need any extra operation required. From the conversion rules shown in Table I , we have the Boolean equations for b_k and c_{k+1}

$$b_k = (d_k^+ \text{ nand } d_k^-) \text{ xor } c_k \qquad (15.a)$$

$$c_{k+1} = c_g \text{ or } (c_p \text{ and } c_k) \qquad (15.b)$$

$$c_g = c_s \text{ and } d_k^- \qquad (15.c)$$

$$c_s = (d_k^+ \text{ xor } d_k^-) \qquad (15.d)$$

$$c_p = (d_k^+ \text{ xnor } d_k^-) \qquad (15.e)$$

An additional approach to increase the speed of a parallel adder that expends area in favor of speed is to use a carry-select adder. Usually, two ripple-carry-adder structures are built, one with a zero carry-in and the other with a one carry-in. Fig. 6. shows a CMOS circuit diagram of the carry propagation circuit in this architecture. This is a kind of carry select

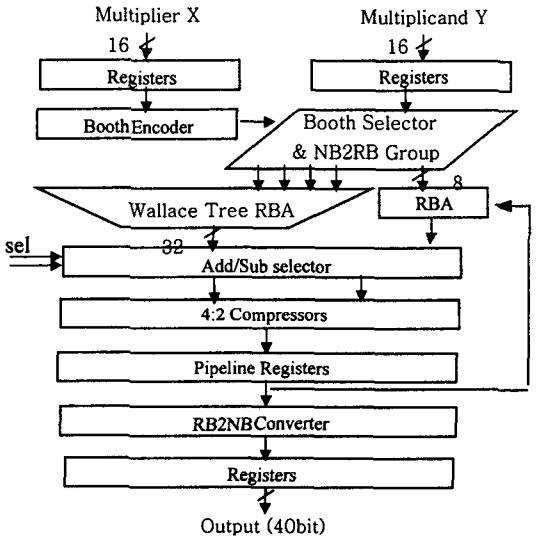


Fig. 7. Block diagram of MAC

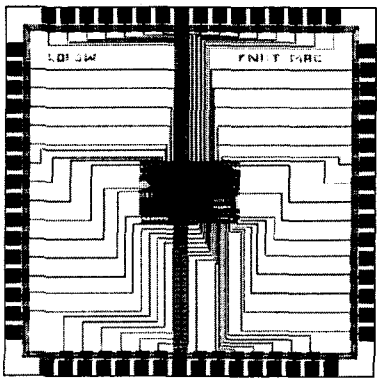


Fig. 8 Layout of MAC

Ⅲ. 16×16-BIT MAC DESIGN

A 16×16-bit MAC is designed with this RB Architecture. Fig. 7 shows the block diagram and Fig. 8 shows the layout pattern of 16×16-bit MAC.

In the RBPP generation stage, modified Booth's algorithm [7] is used to halve the number of partial products. The pairs of two adjacent partial products, in which one is a non-inverted and the other is an inverted partial products, become RB partial products. The additional bits of all partial products that arise from the inversion of the sign are added by each RBA , FS (Full Subtractor) and RB-to-NB converter. The additional bit (+1) of RBPP8 is converted to it's complement form -1 and added in parallel by the FS's (Fig. 5). In the RB-to-NB conversion stage, the final RB number with 40 digits is converted to an NB number with the same number of digits. A CMOS circuit diagram in this architecture is showed by Fig 6. This is a kind of carry select mode. The conversion speed of this stage depends on the conversion time of 40-digits. The number of stages of the MUX circuits is 10 in the critical path of the 40-bit converter. We evaluated the operating speed of the multiplier using the HSPICE simulation. We used the parameter of 0.25 ㎛ CMOS in the simulation. The supply voltage is 2.5V. The simulation is carried out for the critical path extracted from the whole multiplier circuit.

TABLE II Features of 16×16-bit MAC

Multiplier, Multiplicand	16-bit (2's complement)
Instruction	3 instructions X × Y (00) ACC + X × Y (01) ACC - X × Y (10)
Supply voltage	2.5 V
Operation frequency	200 MHz
Active area size	0.526 ×0.284 ㎟
Transistor count	15,172
Density of transistors	100 k/㎟
Process	0.25 ㎛ 5- metal CMOS
Data types	fixed point, Integer
Operation	16 × 16 + 40 bits (2's complement)
Accumulator	Carry Select Adder & RB2NB converter
Architecture	2-stage pipelined architecture

Ⅳ. CONCLUSION

A 16×16-bit MAC is designed using the high speed RB architecture. The RB_PP's are added up by an array of the improved RBA's, that is less transistor count than the conventional 4-2 compressors. A simple and high speed RB-to-NB converter is developed. The process technology of 0.25 ㎛ CMOS with double metal will be used to fabricate the multiplier. The supply voltage is 2.5V. The number of transistors is 15,172.

REFERENCES

[1] C. S. Wallace, "A Suggestion for Fast Multiplier." *IEEETrans. Electron. Comput* vol.EC-13,pp.14-17,Feb. 1964.

[2] Y. Harata, Y. Nakamura, H. natase, M. Takigawa, and N. Takagi, "A high speed multiplier using a redundant binary adder tree," *IEEE J. Solid-state Circuits,* vol. SSC-22, pp. 28-34, Feb. 1987.

[3] H. Edamatsu, T. Taniguchi, T. Nishiyama, and S. Kuninobu, "A 33 MFLOPS floating point processor using redundant binary representation," *Dig. Tech. Papers of 1988 ISSCC,* pp. 152-153, Feb. 1988.

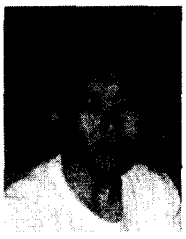
[4] M. Tonomura, "High-speed digital circuit of discrete cosine transform," *SP97-41, DSP94-66, Tech. Rep. IEICE* Japan, pp. 39-46, 1994.

[5] MAKINO et al, "An 8.8-ns 54 x 54-bit multiplier with high speed architecture" *IEEE jouranl of solid-state circuits*. Vol. 31. No 6. June 1996.

[6] Sung-ming Y, Chi-sung L, Chin-hsing C, and Jau-yien L, "An efficient redundant-binary number to binary number converter." *IEEE Journal of solid-state circuits.* Vol. 27, No. 1, January 1992.

[7] A. D. Booth: "A Signed Binary Multiplication Technique," *Quart. J. Mech. Appl.* Math., Vol. 4. Pt. 2, pp. 236-240, 1951.

저자소개



김태민(Tae-min Kim)

1994 년 금오공과대학교 전자공학과 공학사

1998 년 금오공과대학교 전자공학과 공학석사

2002 년 금오공과대학교 전자공학과 (공학박사)

현재 구미 1 대학 정보통신과 겸임교수

관심분야 : 디지털통신, ASIC 등.

신건순(Gun-soon Shin)

1972 년 한양대학교 전기공학과 공학사

1983 년 전북대학교 전기공학과 공학석사

1989 년 전북대학교 전기공학과 공학박사

1993 년 미국 Ohio state university, 전기공학과 solid-state microelectronice lab. 객원교수

현재 금오공과대학교 전자공학부 교수

관심분야 : ASIC 및 아날로그 IC 설계 등.