High Performance 128 Bits Multiplexer Based MBE Multiplier for Signed-Unsigned Number Operating at 1GHz

Ravindra P Rajput
Department of Electronics and Communication Engineering
JSS Research Foundation, Mysore University, Mysore, Karnataka, India

M N Shanmukkha Swamy
Department of Electronics and Communication Engineering
JSS Research Foundation, Mysore University, Mysore, Karnataka, India

Abstract - In this paper we proposed the High Performance 128 Bits Multiplexer based Modified Booth Encoder (MMBE) Multiplier for Signed Unsigned Number Operating at 1 GHz. This multiplier circuit consists of the design of MMBE for Partial Product Generator (PPG) using 16 transistors in Hybrid CMOS (Complementary Metal Oxide Semiconductor) logic. The Hybrid CMOS logic consists of CMOS and Complementary Pass Transistor Logic (CPTL). The MMBE is designed with critical path delay of 0.023 ns/bit, area of 2.52 µm^2, and power consumption of 0.48 µW. The design of full adder for Vertical Column Adder (VCA) and SCGP (Sum Carry Generate and Propagate) circuit using 10 transistors with the delay of 0,018 ns, area of 1.57 µm^2, and power consumption of 0.24 µW. And the design of 8-bit Carry Lookahead Carry Select Adder (CLCSA) for CPA (Carry Propagate Adder) using 180 transistors with delay of 0.045 ns, area of 28.35 µm^2, and the power consumption of 4.37 µW. Comparison of results shows that our proposed MMBE multiplier delay, area and power consumption has been improved by 53 %, area is reduced by 75 % and power dissipation is saved by 62 % respectively.

.

Keywords – PPG, MMBE, CPTL, VCA, PPRT, CLCSA, Hybrid CMOS logic, Supercomputer.

I. INTRODUCTION

Modern supercomputers and vector processors require dedicated and high performance 128 bits multipliers for integer number multiplication of signed and unsigned operands. Since, multiplication hardware is the most time critical, maximum area and power consuming operation, the specialized design of multipliers for least delay, minimum in area and lowest in power consumptions are essential. All the high speed parallel multiplication operation in hardware consists of three phases as follows.

1. Partial Product Generator (PPG).
2. Partial Product Reduction Tree (PPRT).
3. 3. Carry Propagate Adder (CPA).

Since, the performance of the multipliers can be enhanced by designing high speed PPG circuits, many recent advanced papers [1]-[3] have published. Since, the performance of the multipliers can also be enhanced to the most extent by designing high speed PPRT, many high performance papers [4] – [6] have published. And finally since, the maximum speed of the multiplier depends on the performance of the Carry Propagate Adder (CPA), various high speed CLA techniques have published in papers [7]-[10]. The complete literature review of existing PPG, PPRT and CPA is explained in the following sections. Consider the multiplication of two *n*-bit integer numbers *'a'* and *'b'*. Let $a = a_{n-1}\ a_{n-2}\ a_{n-3}$ - - - - - $a_2a_1a_0$ be the multiplicand and $b = b_{n-1}\ b_{n-2}\ b_{n-3}$ - - - - - $b_2b_1b_0$ be the multiplier. The multiplicand *a* and multiplier *b* in twos complement form can be written as follows.

$$a = -a_{n-1}\cdot 2^{n-1} + a_{n-2}\cdot 2^{n-2} \ldots\ldots a_1\cdot 2^1 + a_0\cdot 2^0$$

$$b = -b_{n-1}\cdot 2^{n-1} + b_{n-2}\cdot 2^{n-2} \ldots\ldots b_1\cdot 2^1 + b_0\cdot 2^0$$

Above equations can be written as follows.

$$a = -a_{n-1}\,2^{n-1} + \sum_{k=0}^{n-2} a_k\,2^k$$

$$b = -b_{n-1}\,2^{n-1} + \sum_{k=0}^{n-2} b_k\,2^k \qquad (1)$$

Since, the MBE technique uses 3-bits for the encoding of the multiplier operand *b*, equation (1) can be written as given in equation (2).

$$b = \sum_{k=0}^{\frac{n}{2}-1} (-2b_{2k+1} + b_{2k} + b_{2k-1})\,2^k \qquad (2)$$

Where $b_{-1} = 0$. In equation (2) the terms in the brackets indicates the encoding of three bits to obtain *0, a, 2a, - a, -2a* to generate the partial product rows as shown in table I. The final product using MBE technique is obtained using the following expression.

$$p = a \times b = \sum_{k=0}^{\frac{n}{2}-1} (-2b_{2k+1} + b_{2k} + b_{2k-1})\,a\,2^{2k}$$

With reference to equation (2) references [1] – [3] have presented the design of MBE architecture to generate a partial products in parallel. Reference [1] presented the design of PPG as shown in Figure 1. This PPG has implemented with 68 transistors in CMOS logic, and its delay, area and power consumption measured has 0.033 ns/bit, 7.83 μm^2/bit and 1.81 μW /bit respectively. Reference [2] presented the design of PPG as shown in figure 2. This PPG has implemented with 56 transistors in CMOS logic, and its delay, area and power consumption measured has 0.029 ns/bit, 0.13 μm^2/bit and 1.62 μW /bit respectively. Reference [3] presented the design of PPG as shown in figure 3.This PPG has implemented with 56 transistors in CMOS logic, and its delay, area and power consumption measured has 0.045 ns/bit, 0.12 μm^2/bit and 1.65 μW /bit respectively.

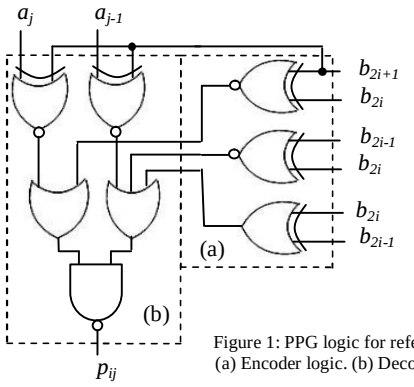


Figure 1: PPG logic for reference [1]
(a) Encoder logic. (b) Decoder logic.

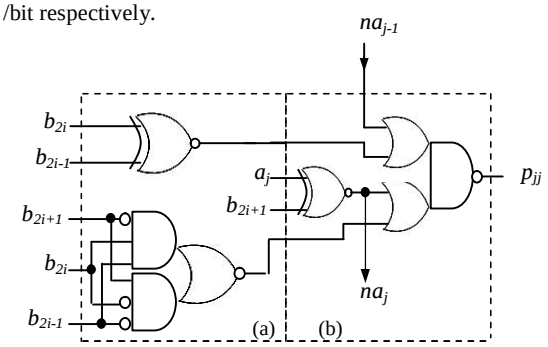
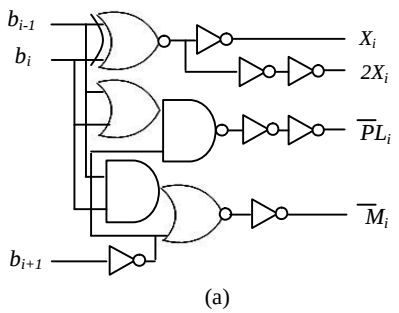


Figure 2: PPG logic for reference [2]
(a) Encoder logic. (b) Decoder logic.



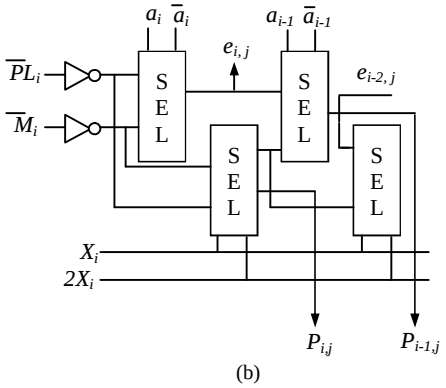


Figure 3: PPG logic for reference [3] (a) Encoder logic (b) Selector logic.

The second stage of the multiplier PPRT. The Function of the PPRT circuit is to reduce the *n* number of partial products to two only. This section briefs about various existing PPRT. Reference [4] presented the design of a PPRT using Three Dimensional Minimization (TDM) Method as shown in figure 4. In case of TDM all the bits of the column and carry bits from the previous column has added to produce a sum bit and the number of carry bits. The TDM of Figure 4 has implemented with 98 transistors in CMOS logic, and its delay, area and power consumption measured has 0.06 ns, 30.87 μm^2 and 26.43 μW respectively. Reference [1] has used the concept of reference [4] as PPRT. Reference [5] presented the design of 4:2 and 5 :2 compressors. Figure 5 shows the architecture of 4:2 compressor, this takes 5-inputs and produces 3 outputs namely two carry and a sum. The 4:2 compressor has implemented with 60 transistors in CMOS logic, and its delay, area and power consumption measured has 0.047 ns, 18.9 μm^2 and 20.67 μW respectively. The architecture of 5:2 compressor is as shown in Figure 6. This takes 7-inputs and produces 4 outputs namely three carry and a sum. The 5:2 compressor has implemented with 90 transistors in CMOS logic, and its delay, area and power consumption measured has 0.06 ns, 28.35 μm^2 and 24.3 μW respectively. Reference [6] presented the design of Wallace tree for the addition of 7-bits of the PPRT, and the number of transistors, delay, area and power consumption has same as the reference [6].

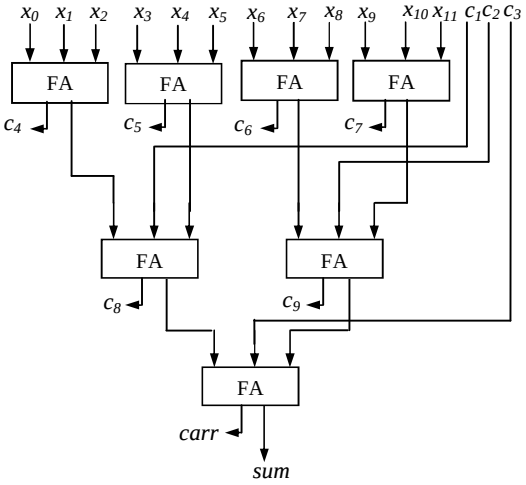


Figure 4: Vertical compression slice of TDM PPRT.

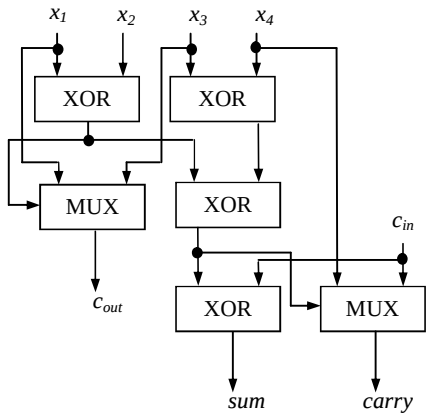


Figure 5: Architecture of 4:2 compressor.

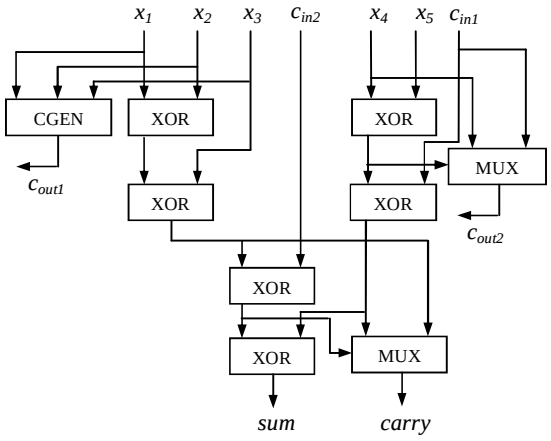


Figure 6: Architecture of 5:2 compressor.

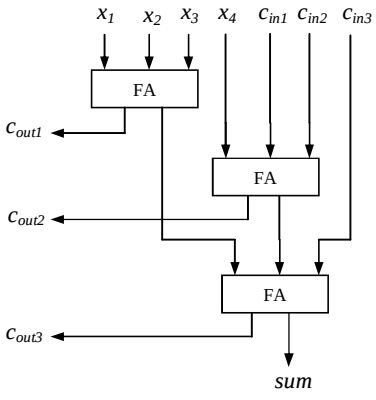


Figure 7: Wallace tree addition of 7 bits.

The final stage is the CPA, the fastest of all the CPA is the CLA. Reference [1] presented the design of multiple-level conditional-sum adder (MLCSMA) as the final stage adder for high speed operation. It uses the combined

effect of conditional-sum adder (CSMA) and conditional-carry adder (CCA). CSMA was proposed for performance and CCA was proposed to save area. References [2] - [3] has used the concept of references [7] - [10] presented the design of CPA for high speed, small area, and low power consumption. Reference [7] presented the design of high performance and low power 64 bits adder as shown in Fig. 8. Here, two pre-sums are computed by assuming carry-in is at logic-0, and the other assuming carry-in is at logic-1. The global carry network (GCN) generates an intermediate carry signals that select the appropriate 8-bit pre-sums, and the final carry output. This circuit requires additional logic circuits for the implementation of GCN.

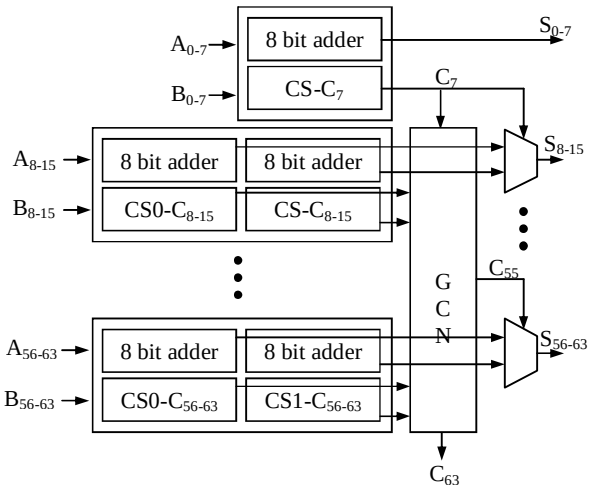


Figure.8. Architecture of 64-bit adder based on selection of 8-bit pre-sums

Thus, we proposed 128 bits multiplier with MMBE implemented using 16 transistors by CPTL. The VCA for the addition of 13-bits of the column is implemented in CMOS with 70 transistors. And the CLCSA for 8-bit operation is implemented in CMOS with180 transistors.

II. PROPOSED DESIGN OF MULTIPLIER

We proposed a 128 bits multiplier based on Multiplexer based MBE technique, the VCA, and CLCSA for high performance, very less area, and low power consumption. The requirement of modern supercomputer which can compute multiplication operation on matrix data can be fulfilled by this multiplier. The design of proposed multiplexer based MBE, the design of VCA for PPRT and the design of CLCSA for CPA is explained in the following section.

A. Proposed Multiplexer based MBE–

Figure 9 shows the block diagram of MMBE multiplier. Its operation is based on the concept of 4 to 1 multiplexer, and this is called as 1-bit partial product generator. The MMBE produces all the partial products in parallel. Table I shows the truth table of proposed MMBE scheme. From table I equations (1) – (5) are obtained.

Table I: Truth table of MMBE scheme

b_{i+1}	b_i	b_{i-1}	s_{i+1}	s_i	n_i	p_{ij}
0	*0*	*0*	*0*	*0*	*0*	*+0*
0	*0*	*1*	*0*	*1*	*0*	*+a*
0	*1*	*0*	*1*	*0*	*0*	*+a*
0	*1*	*1*	*1*	*1*	*0*	*+2a*
1	*0*	*0*	*1*	*1*	*1*	*-2a*
1	*0*	*1*	*1*	*0*	*1*	*-a*
1	*1*	*0*	*0*	*1*	*1*	*-a*
1	*1*	*1*	*0*	*0*	*0*	*-0*

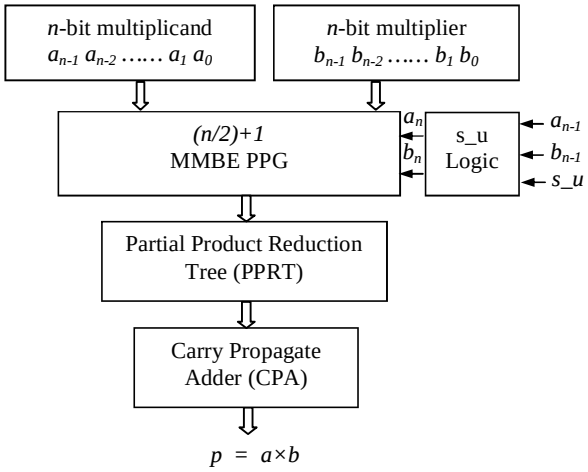


Figure 9. Block diagram of proposed multiplier.

$$p_{ij} = x_i \overline{s_{i+1}}\, s_i + x_i\, s_{i+1}\, \overline{s_i} + x_{i+1}\, s_{i+1}\, s_i \quad (1)$$

$$s_{i+1} = b_i \oplus b_{i+1} \quad (2)$$

$$s_i = b_{i-1} \oplus b_{i+1} \quad (3)$$

$$x_{i+1} = b_{i+1} \oplus a_{i+1}, \quad x_i = b_{i+1} \oplus a_i \quad (4)$$

$$n_i = b_{i+1}\,(\overline{b_{i-1}\, b_i}) \quad (5)$$

For the Equations (1) to (4) MMBE is implemented as shown in Figure 10. This is called as the 1-Bit partial product generator (PPG). This is implemented in hybrid CMOS logic using 16 transistors as shown in Figure 11. According to the input multiplier operand *b*, the MMBE logic selects *0, a, 2a, −a, −2a* to generate the partial product rows in parallel. In equation (1) when $s_{i+1} = 1$, $s_i = 1$ MMBE selects $-2a$ or $+2a$. And when $s_{i+1} = 0$, $s_i = 1$ and $s_{i+1} = 1$, $s_i = 0$ the MMBE selects $-a$, $+a$. The negate operation is achieved by one's complimenting each bit of *a* and then adding $n_i = 1$ to the least significant bit. The negate operation is implemented using equation (5) as shown in Figure 12.

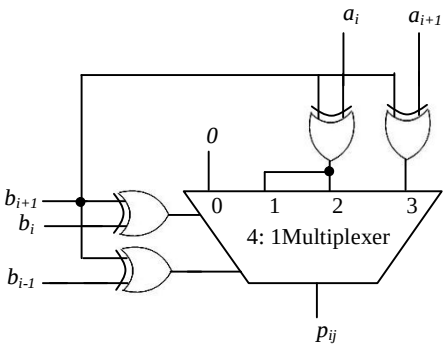


Figure 10. Logic diagram of MMBE PPG

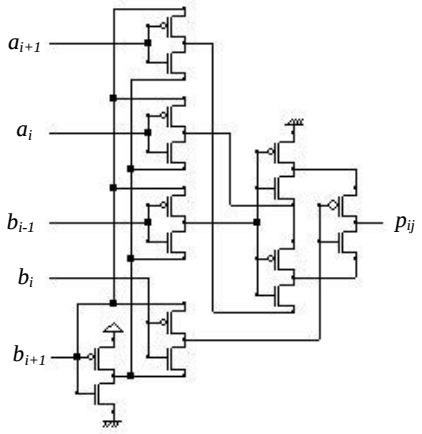


Figure 11. Circuit diagram of MMBE PPG

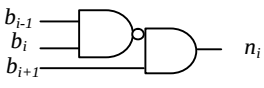


Figure 12. Negate bit generator logic.

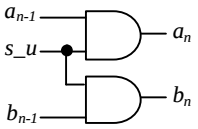


Figure 13. Sign converter logic.

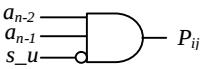


Figure.14. Logic diagram for final bit of final row of PPG.

Figure 13 shows the logic diagram of sign converter. A mode signal called signed-unsigned (*s_u*) is used to indicate whether the multiplication operation is for signed or unsigned number. When *s_u = 0*, unsigned number multiplication operation is performed and when *s_u = 1*, signed number multiplication operation is performed. When the operation is unsigned multiplication, the sign extended bit of both multiplicand and multiplier should be extended with 0's as given in equation (6), and when the operation is signed multiplication the sign extended bit depends on whether the multiplicand operand is negative or the multiplier operand is negative or both the operands are negative. For this when the multiplicand operand is negative and multiplier operand is positive the sign extended bits should be generated as given in equation (7). And when the multiplicand operand is positive and multiplier operand is negative the sign extended bits should be generated are as given in equation (8). Finally, since the multiplier has to multiply both signed and unsigned number, the MSB bit of the final row should be computed as given by the equation (9).

$$a_n = a_{n+1} = b_n = b_{n+1} = 0 \quad (6)$$

$$s\_u = 1,\ a_{n-1} = 1,\ b_{n-1} = 0,\ a_n = a_{n+1} = 1, \text{ and } b_n = b_{n+1} = 0 \quad (7)$$

$$s\_u = 1,\ a_{n-1} = 0, ,\ b_{n-1} = 1,\ a_n = a_{n+1} = 0, \text{ and } b_n = b_{n+1} = 1 \quad (8)$$

$$P_{ij} = \overline{s\_u}\ a_{n-1}\ a_{n-2} \quad (9)$$

Where $i = n-1, j = n-1$.

B. Proposed VCA for PPRT–

Our proposed Vertical Column Adder (VCA) is based on the concept of references [4], [6] which presented the design of a PPRT with minimum delay. In this method, each column partial product bits of that column and carry bits generated by the previous column has been added to produce a sum bit and the number of carry bits. The carry bits from the previous column have been fed as input to the full adder so that the delay of the VCA has been the minimum. Reference [4] PPRT consists of full adders only, but our proposed PPRT consists of full adders and the Sum Carry Generate and Propagate (SCGP) logic. The SCGP logic circuit produces the Sum, Carry Generate term and Carry Propagate term, which are essential for the CLA operation. The design of high performance full adder has been implemented using the equations (10) through (11).

$$s_i = x_{i+1} \oplus x_{i+2} \oplus c_i \quad (10)$$

$$c_{i+1} = (x_{i+1} \oplus x_{i+2})c_i + (\overline{x_{i+1} \oplus x_{i+2}})x_{i+1} \quad (11)$$

The logic diagram of full adder is shown in Figure 15 (a) and its circuit diagram is shown in Figure 15 (b). This is implemented in CMOS logic using only 10 transistors. The required logic for SCGP are derived from the equation (11) are given by the equations (12) and (13). Where cp_i is called carry propagate term, and cg_i is called carry generate term.

$$cp_i = x_{i+1} \oplus x_{i+2} \quad (12)$$

$$cg_i = (\overline{x_{i+1} \oplus x_{i+2}})\ x_{i+1} \quad (13)$$

Figure 15 (c) shows the circuit diagram of SCGP logic, this is the final cell of each VCA. This is designed to perform operations such as sum, carry generate and carry propagate terms so as to save the extra hardware for carry generate and carry propagate terms and is implemented in CMOS logic using only 10 transistors. The carry generate and propagate terms are fed as input to the 8- bit CLA circuit shown in Figure 17 (a).

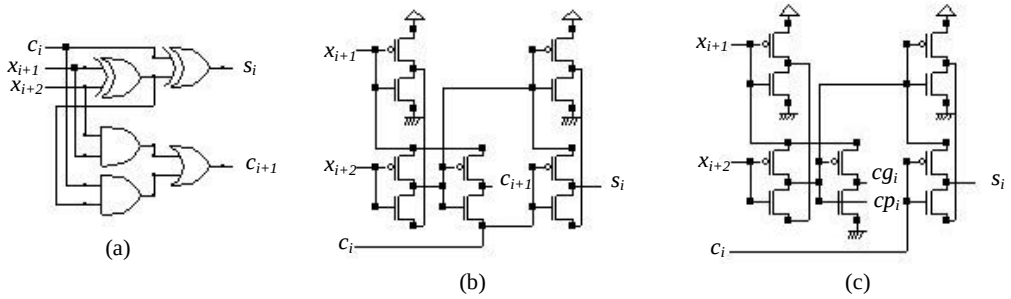


Figure 15. Architecture of full adder. (a). Logic diagram.
(b) Circuit diagram. (c) Circuit diagram of SCGP.

C. Proposed CLCSA for CPA–

The final adder which combines the effect of Carry Lookahead Adder and Carry Select Adder (CLCSA) is as shown in Fig. 16. The 8-bit CLA adder is designed and is used in cascade through carry select adder technique for high performance. All the 8-bit CLA adders produce carry in parallel and there are two such 8-bit CLA's in each stage with 0 and 1 as the initial carry input. If the final carry output from the previous stage of 8 bit CLA is 1 then the output selected by the 2:1 multiplexer is the output of the CLA adder with 1- input as the initial carry. Carry expressions for 8-bit CLA adder's are as follows.

$c_1 = g_0 + p_0 c_0$
$c_2 = g_1 + p_1 g_0 + p_1 p_0 c_0$
$c_3 = g_2 + p_2 g_1 + p_2 p_1 g_0 + p_2 p_1 p_0 c_0$
$c_4 = g_3 + p_3 g_2 + p_3 p_2 g_1 + p_3 p_2 p_1 g_0 + p_3 p_2 p_1 p_0 c_0$
$c_5 = g_4 + p_4 g_3 + p_4 p_3 g_2 + p_4 p_3 p_2 g_1 + p_4 p_3 p_2 p_1 g_0 + p_4 p_3 p_2 p_1 p_0 c_0$
$c_6 = g_5 + p_5 g_4 + p_5 p_4 g_3 + p_5 p_4 p_3 g_2 + p_5 p_4 p_3 p_2 g_1 + p_5 p_4 p_3 p_2 p_1 g_0 + p_5 p_4 p_3 p_2 p_1 p_0 c_0$
$c_7 = g_6 + p_6 g_5 + p_6 p_5 g_4 + p_6 p_5 p_4 g_3 + p_6 p_5 p_4 p_3 g_2 + p_6 p_5 p_4 p_3 p_2 g_1 + p_6 p_5 p_4 p_3 p_2 p_1 g_0 + p_6 p_5 p_4 p_3 p_2 p_1 p_0 c_0$
$c_8 = g_7 + p_7 g_6 + p_7 p_6 g_5 + p_7 p_6 p_5 g_4 + p_7 p_6 p_5 p_4 g_3 + p_7 p_6 p_5 p_4 p_3 g_2 + p_7 p_6 p_5 p_4 p_3 p_2 g_1 + p_7 p_6 p_5 p_4 p_3 p_2 p_1 g_0 + p_7 p_6 p_5 p_4 p_3 p_2 p_1 p_0 c_0$

Equations c_1 through c_8 are implemented as shown in Figure 17 (a). Inputs g_0 through g_7 has been provided from the SCGP circuit of Fig. 15 (c). The inputs n_1 through n_8 are the outputs of NAND gates, where n_1 is the output of 2-inputs NAND gate, n_2 is the output of three inputs NAND gate, and $n_3, n_4, n_5, n_6, n_7, n_8$ are outputs of 4, 5, 6, 7, 8 and 9 inputs NAND gates respectively. The two input and three input NAND gate circuit diagram is as shown in Figure 17 (b) and Figure 17 (c) respectively. The circuit diagram of *2:1* multiplexer logic is shown in Figure 17 (d). This has been implemented in CPTL with only 2 transistors. The delay of *2:1* multiplexer measured is *0.006 ns*.

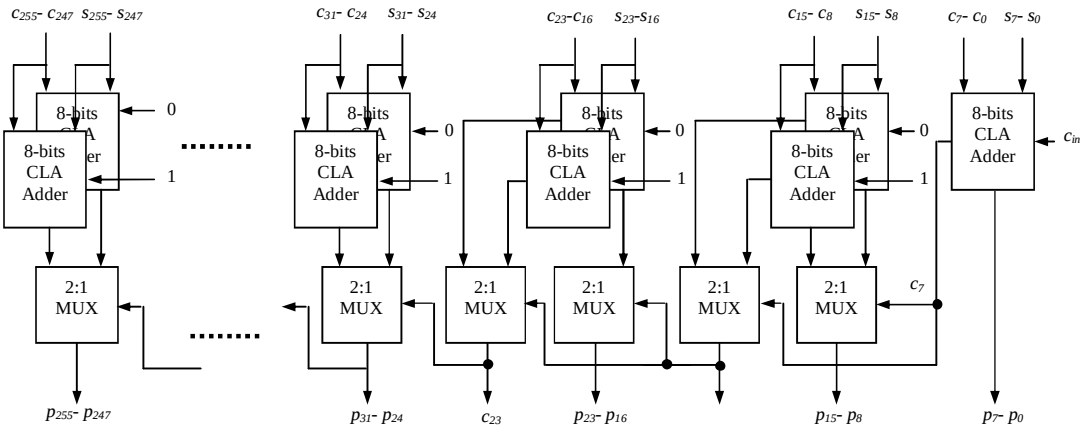


Figure 16. Architecture of CLCSA for 128-bits multiplier

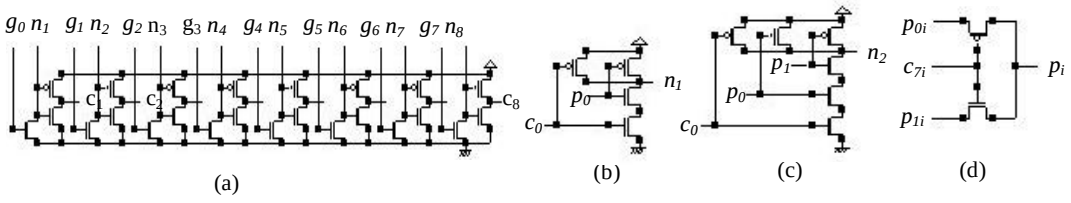


Figure 17. Circuit diagram of (a) 8-bit CLA. (b) Input at n1. (c) Input at n2. 2:1 multiplexer. (d) 2:1 Multiplexer.

III. EXPERIMENTAL RESULTS

The 45nm Microwind tool is used to measure the critical path delay, the chip area and the power dissipated for 128 ×128-Bit signed-unsigned multiplier. This multiplier is divided into MMBE (PPG), PPRT and CLCSA unit cell. Each unit is implemented and synthesized and measured critical path delay, area, and power consumption as listed in Table 1 and Table 2. Comparison of results shows that our proposed MMBE multiplier delay, area and power consumption has been improved by 53 %, area is reduced by 75 % and power dissipation is saved by 62 % respectively.

Table – 1 Comparison of PPG

References	Number of transistors	Delay (ns)	Area (μm^2)	Power (μW)
Reference [1]	68	0.033	7.83	1.99
Reference [2]	56	0.044	7.14	1.56
Reference [3]	46	0.045	6.18	1.29
Proposed	16	0.023	2.52	0.48

Table – 2 Comparison of multipliers

Multiplier Size	References	Number of transistors	Delay (ns)	Area (μm^2)	Power (μW
128×128	Reference [1]	436280	2 .15	114524	4806.0
	Reference [2]	406176	2 .36	104648	4206.0
	Reference [3]	386360	2.34	90416	4007.7
	Proposed	172224	1.00	28446	1819.5

IV. CONCLUSION

Our proposed MMBE implemented using 16 transistors by the hybrid CMOS logic compared to the reference paper of 68, 56, 62, and 46 transistors respectively. This shows MMBE circuit occupies very small portion of the total area compared to the area required by the PPRT and CLA. Comparison of results shows that for the proposed MMBE based multiplier delay, area and power consumption has been improved by 53 %, area is reduced by 75 % and power dissipation is saved by 62 % respectively. Since the Hybrid CMOS logic uses both the CMOS logic and CPTL, it requires buffers and which may increase 2% of delay, area, and power consumption.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the Chairman and members of J S S Research foundation, SJCE Campus, Mysore, for all the facility provided for this research work.

REFERENCES

[1] W. –C. Yeh and C. –W. Jen, "High Speed Booth encoded Parallel Multiplier Design," IEEE transactions on computers, vol. 49, no. 7, pp. 692-701, July 2000.

[2] Shiann-Rong Kuang, Jiun-Ping Wang, and Cang-Yuan Guo, "Modified Booth multipliers with a Regular Partial Product Array," IEEE Transactions on circuits and systems-II, vol 56, No 5, May 2009.

[3] Gensuke Goto, Atsuki Inoue, Ryoichi Ohe, Shoichiro Kashiwakura, Shin Mitarai, Takayuki Tsuru, and Tetsuo Izawa, "A 4.1 ns Compact 54×54-b Multiplier Utilising Sign-Select Booth Encoders", IEEE journal of solid-state circuit, VOL. 32, NO. 11, November 1997.

[4] Vogin G. Oklobdzija, David Vileger, Simon S. Liu, "A Method for Speed Optimized Partial Product Reduction and Generation of Fast Parallel Multipliers Using an Algorithmic Approach",IEEE transaction on computers, Vol. 45, No. 3, pp. 294-306, March 1996.

[5] C.-H. Chang, J. Gu, and M. Zhang, "Ultra low-voltage low-power CMOS 4–2 and 5–2 compressors for fast arithmetic circuits," *IEEE Trans.Circuits Syst. I, Reg. Papers*, vol. 51, no. 10, pp. 1985–1997, Oct. 2004.

[6] C. S. Wallace, "A Suggestion for a Fast Multiplier", IEEE Transaction on Electronic Computers, pp. 14-17, February 1964.

[7] Radu Zlatanovici, Sean Kao, and Borivoje Nikolic´ "A 240 ps 64 b carry-lookaheadadder in 90 nm CMOS," IEEE journal of solid-state circuit,, VOL. 44, NO. 2, pp.569-583, February 2009.

[8] Yuke Wang, C. Pai, and Xiaoyu Song,"The Design of Hybrid Carry Lookahead/Carry Select Adders, IEEE Transactions on circuits and systems-II, VOL. 49, NO. 1, JANUARY 2002, pp. 16-24.

[9] J. J. Kim, R. Joshi, C.-T. Chuang, and K. Roy, "SOI-optimized 64-bit high-speed CMOS adder design," in *Symp. VLSI Circuits Dig. Tech. Papers*, pp. 122–125, 2002.

[10] Amaury Nève, Helmut Schettler, Thomas Ludwig, Denis Flandre,*"Power-Delay Product Minimization in High-Performance 64-bit Carry Select Adders", IEEE Trans. Very Large Scale Integr. Systems*, VOL. 12, NO. 3, pp.235-244, MARCH 2004.