

An Efficient Architecture for 32-bit Multiply-Accumulate (MAC) Unit Using Redundant Binary Multiplier

M.Ephraem¹ Mr. K.Raju²

1. M. Tech Student, Department of ECE, KKR & KSR Institute of Technology & Sciences (KITS), Guntur.
2. Associate Professor, Department of ECE, KKR & KSR Institute of Technology & Sciences (KITS), Guntur.

ABSTRACT

The multiplication and accumulation are the vital operations involved in almost all the Digital Signal Processing applications. Consequently, there is a demand for high speed processors having dedicated hardware to enhance the speed with which these multiplications and accumulations are performed. The speed of MAC depends on the speed of multiplier. Due to its high modularity and carry-free addition, a redundant binary (RB) representation can be used when designing high performance multipliers. In this paper, a new RB modified partial product generator (RBMPPG) is proposed; it removes the extra ECW and hence, it saves one RBPP accumulation stage. Here using RB modified partial product generator multiplier is used to design MAC unit. The results reveals the implementation of proposed MAC unit is efficient in terms of area and speed. Synthesis and Simulation are performed using Xilinx ISE design suit 13.2 and Modelsim respectively.

Key Words: MAC, Redundant binary, modified booth encoding, RB partial product generator, RB multiplier

I. INTRODUCTION

DIGITAL multipliers are widely used in arithmetic units of microprocessors, multimedia and digital signal processors. Many algorithms and architectures have been proposed to design high-speed and low-power multipliers [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13]. A normal binary (NB) multiplication by digital circuits includes three steps. In the first step, partial products are generated; in the second step, all partial products are added by a partial product reduction tree until two partial product rows remain. The two partial product rows are added by a fast carry propagation adder in the third step. Two methods have been used to perform the second step for the partial product reduction. A first method uses four-two compressors, while a second method uses redundant binary (RB) numbers [5], [6]. Both methods allow the partial product reduction tree to be reduced at a rate of 2:1.

By Avizienis to perform signed-digit arithmetic the redundant binary number representation has been introduced the RB number has the capability to be represented in different ways. Fast multipliers can be designed using redundant binary addition trees [2], [3]. The redundant binary

representation has also been applied to a floating-point processor and implemented in VLSI [4]. High performance RB multipliers have become popular due to the advantageous features, such as high modularity and carry-free addition [5], [6], [7], [8], [9].

A RB partial product (RBPP) generator, a RBPP reduction tree and a RB-NB converter are consisted in a RB multiplier. A Radix-4 Booth encoding or a modified Booth encoding (MBE) is usually used in the partial product generator of parallel multipliers to reduce the number of partial product rows by half [5], [6], [10], [11], [12], [13]. An N-bit conventional RB MBE (CRBBE-2) multiplier requires $\lceil N/4 \rceil$ RBPP rows and a RBPP row can be obtained from two adjacent NB partial product rows by inverting one of the pair rows [5], [6]. For both the RB and the Booth encoding [5], [6] [14]; An additional error-correcting word (ECW) is also required. Therefore, the number of RBPP accumulation stages (NRBPPAS) required by a power-of-two word-length (i.e., 2n-bit) multiplier is given by:

$$\begin{aligned} \text{NRBPPAS} &= \lceil \log_2(N/4 + 1) \rceil \\ &= n - 1, \text{ if } N = 2^n. \end{aligned} \quad (1)$$

If the additional ECW can be removed, an RBPP accumulation stage is saved, so resulting in improvements of complexity and criticalpath delay for a RB multiplier. For example, a conventional 32-bit RB multiplier has four RBPP accumulation stages; if the ECW is removed, then the number of RBPP accumulation stages is reduced to 3, i.e., the stage count is

decreased by 25 percent. Note that the problem of extra ECW does not exist in standard significant size (i.e., 24x24-bit and 54x54-bit) RB multipliers as used in floating point-arithmetic units [5], [6]. Alternatively, the number of partial products can be reduced by a high-radix Booth encoding technique. However, the number of expensive hard multiples (i.e., a multiple that is not a power of two and the operation cannot be performed by simple shifting and/or complementation) increases too [14], [15], [16]. Besli and Desmukh [16] noticed that some hard multiples can be obtained by the differences of two simple power-of-two multiplies.

A new radix-16 Booth encoding (RBBE-4) technique without ECW has been proposed in [14]; the issue of hard multiples is avoided by it. To overcome the hard multiple problem and avoid the extra ECW, A radix-16 RB Booth encoder can be used but at the cost of doubling the number of RBPP rows. Therefore, the number of radix-4 MBE rows is the same as in the radix-16 RBPP. However, based on a radix-16 Booth encoding the RBPP generator has a lower speed compared with the MBE partial product generator [10] and complex circuit structure when requiring the same number of partial products.

For designing a 2ⁿ-bit RB multiplier this paper focuses on the RBPP generator with fewer partial product rows by eliminating the extra ECW. A new RB modified partial product generator based on MBE (RBMPPG-2) is proposed. In the proposed RBMPPG-2, the ECW of each row is moved to its next neighbor row. Furthermore, the

last partial product row is combined with both the two most significant bits (MSBs) of the first partial product row and the two least significant bits (LSBs) of the last partial product row by logic simplification the extra ECW generated.

II. 32-BIT MAC UNIT USING CONVENTIONAL MULTIPLIER

A multiplier and an accumulator containing the sum of the previous successive products which are consisted in a MAC unit. From the memory location the MAC inputs are obtained and given to the multiplier block. The design consists of 32 bit modified multiplier, 64 bit ripple carry adder and a shift register. Not only in DSP applications but also in multimedia information processing and various other applications the Multiplier-Accumulator (MAC) operation is the key operation.

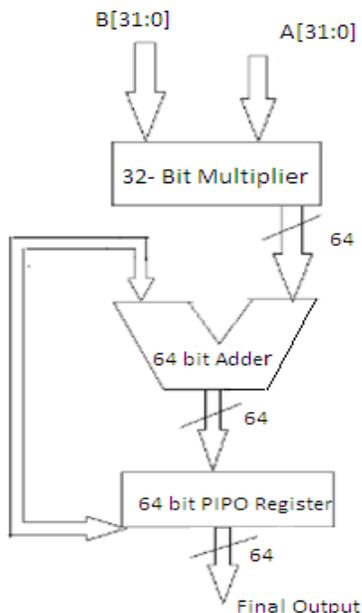


Fig 1 Conventional MAC unit

We are using the conventional multiplier in above MAC unit. The conventional multiplier of width $N \times N$ bits will generate the N number of partial products. The partial products are generated by bit wise AND in one multiplier bit with another multiplier. Hence, $2N$ -multiplications and N -Adders are used in the $N \times N$ bit multiplier in the architecture of Conventional multiplier.

III. PROPOSED DESIGN

The RBMPPG-2 in [17] can be applied to any 2^n -bit RB multipliers with a reduction of a RBPP accumulation stage compared with conventional designs. Although the delay of RMPPG-2 increases by one-stage of TG delay, the delay of one RBPP accumulation stage is significantly larger than a one-stage TG delay. Therefore, the delay of the entire multiplier is reduced. For the proposed design the improved complexity, delay and power consumption are very attractive. By using the proposed RBPP generator A 32-bit RB MBE multiplier is shown in Fig. 2. The multiplier consists of the proposed RBMPPG-2, three RBPP accumulation stages, and one RB-NB converter. Eight RBBE-2 blocks generate the RBPP (p_i^+ , p_i^-); they are summed up by the RBPP reduction tree that has three RBPP accumulation stages. RB full adders (RBFAs) and half adders (RBHAs) are contained in each RBPP accumulation block.

The 64-bit RB-NB converter converts the final accumulation results into the NB representation, which uses a hybrid parallel-prefix/carry select adder [18] (as one of the most efficient fast parallel adder designs). There are four stages in a conventional 32-

bit RB MBE multiplier architecture; however, by using the proposed RBMPPG-2, the number of RBPP accumulation stages is reduced from 4 to 3 (i.e., a 25 percent reduction).

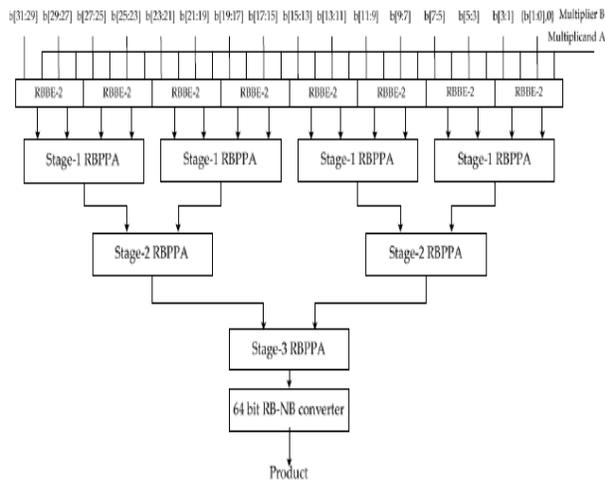


Fig 2 The block diagram of a 32-bit RB multiplier using the proposed RBMPPG-2.

These are significant savings in delay, area as well as power consumption. The improvements in delay, area and power consumption are further demonstrated in the next section by simulation.

Table I compares the number of RBPP accumulation stages in different 2^n -bit RB multipliers, i.e., 8x8-bit, 16x16-bit, 32x32-bit, 64x64-bit multipliers. For a 64-bit multiplier, the proposed design has four RBPP accumulation stages; it reduces the partial product accumulation delay time by 20 percent compared with CRBBE-2 multipliers. Although both the proposed design and RBBE-4 have the same number of RBPP accumulation stages, RBBE-4 is more complex, because it uses radix-16 Booth encoding [14].

Table I Comparison of RBPP Accumulation Stages in RBPP Reduction Tree

Methods	64x64	32x32	16x16	8x8
CRBBE-2	5	4	3	2
RBBE-4 [14]	4	3	2	1
Proposed	4	3	2	1

It starts computing value for the given 32 bit input when the input is given to the multiplier and hence the output will be 64 bits. The multiplier output is given as the input to ripple carry adder which performs addition. The output of carry save adder is 65 bit i.e. one bit is for the carry (16bits+ 1 bit). Then, the output is given to the accumulator register. In this design the accumulator register used is parallel in-Parallel out (PIPO).

Since the bits are huge and also ripple carry adder produces all the output values in parallel, PIPO register is used where the input bits are taken in parallel and output is taken in parallel. For the accumulator register the output is taken out or fed back as one of the input to the ripple carry adder. The above figure 1 shows the basic architecture of MAC unit. The figures 2, 3 & 4 shows the detailed design block diagrams of 32X32 Redundant Binary multiplier, 64 bit Ripple Carry Adder and 64 bit PIPO Shift Register.

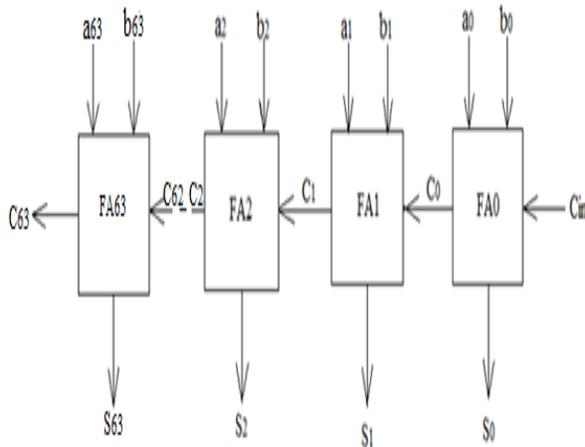


Fig 3 64-Bit Ripple Carry Adder

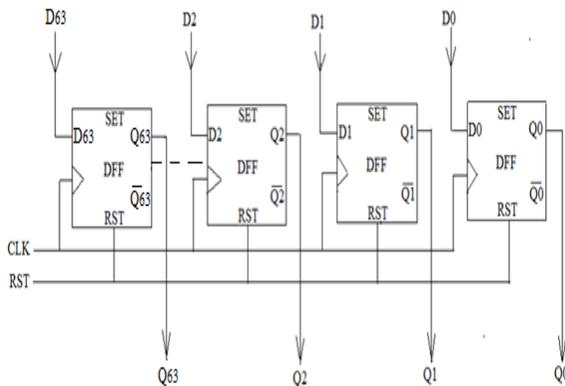


Fig 4 64-Bit PIPO Shift Register

IV. RESULTS

The Proposed MAC unit simulated and synthesized using the Xilinx Design Suit13.2 with device family as spartan3E and device Xc3s100e-5vq100. The simulation Results are verified by using Modelsim. The Figure 5 shows the Simulation Results of Proposed MAC unit

and Table II shows the comparison of conventional and Proposed MAC units.

In the below Table-II observe the number of (Look Up Tables) LUT's used in the Conventional MAC unit is 3698 which is higher than that of Proposed MAC unit. Here area occupied by the Conventional MAC unit is higher than that of Proposed MAC unit. In proposed Mac unit, the multiplier used in conventional MAC unit is replaced by Redundant binary multiplier. And also the delay produced by Conventional MAC unit is very high when comparing with proposed MAC unit.

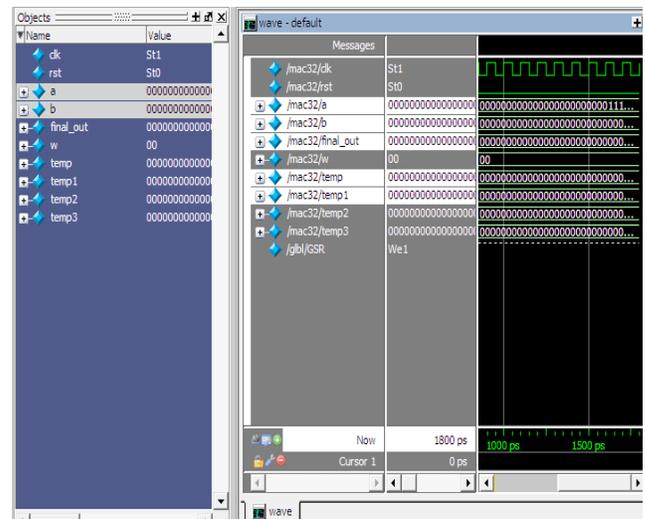


Fig 5 Simulation Results of Proposed MAC unit

Table II Comparison of Area and Delay for Conventional and Proposed MAC units

Architecture	LUT's	Area (Kb)	Delay (ns)
Conventional (32 – bit)	3698	277556	82.414
Redundant Binary (32 – bit)	2927	229364	41.575

V. CONCLUSION

The results obtained are quite encouraging. The 32-bit multiplier-accumulators (MAC) unit using redundant binary multiplier is presented in this work. For the MAC unit the basic building blocks are identified and each of the blocks is analyzed for its performance. It can be concluded that 32-bit MAC using redundant binary Multiplier is superior in all respect like speed, delay, area compared to conventional one. The application of transform algorithm includes linear filtering, Spectrum Analysis, Correlation which will further adds the field of Communication, signal & image processing and instrumentation that can also benefit future needs of wireless communications systems.

REFERENCES

[1] Avizienis, “Signed-digit number representations for fast parallel arithmetic,” IRE Trans. Electron.

Comput., vol. EC-10, pp. 389–400, 1961.

- [2] N. Takagi, H. Yasuura, and S. Yajima, “High-speed VLSI multiplication algorithm with a redundant binary addition tree,” IEEE Trans. Comput., vol. C-34, no. 9, pp. 789–796, Sep. 1985.
- [3] Y. Harata, Y. Nakamura, H. Nagase, M. Takigawa, and N. Takagi, “A high speed multiplier using a redundant binary adder tree,” IEEE J. Solid-State Circuits, vol. SC-22, no. 1, pp. 28–34, Feb. 1987.
- [4] H. Edamatsu, T. Taniguchi, T. Nishiyama, and S. Kuninobu, “A 33 MFLOPS floating point processor using redundant binary representation,” in Proc. IEEE Int. Solid-State Circuits Conf., 1988, pp. 152–153.
- [5] H. Makino, Y. Nakase, and H. Shinohara, “A 8.8-ns 54x54-bit multiplier using new redundant binary architecture,” in Proc. Int. Conf. Comput. Des., 1993, pp. 202–205.
- [6] H. Makino, Y. Nakase, H. Suzuki, H. Morinaka, H. Shinohara, and K. Makino, “An 8.8-ns 54_54-bit multiplier with high speed redundant binary architecture,” IEEE J. Solid-State Circuits, vol. 31, no. 6, pp. 773–783, Jun. 1996.
- [7] Y. Kim, B. Song, J. Grosspietsch, and S. Gillig, “A carry-free 54b_54b multiplier using equivalent bit conversion algorithm,” IEEE J. Solid-State Circuits, vol. 36, no. 10, pp. 1538–1545, Oct. 2001.
- [8] Y. He and C. Chang, “A power-delay efficient hybrid carry-look ahead carry-

- select based redundant binary to two's complement converter," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 55, no. 1, pp. 336–346, Feb. 2008.
- [9] G. Wang and M. Tull, "A new redundant binary number to 2's-complement number converter," in Proc. Region 5 Conf.: Annu. Tech. Leadership Workshop, 2004, pp. 141–143.
- [10] W. Yeh and C. Jen, "High-speed booth encoded parallel multiplier design," IEEE Trans. Comput., vol. 49, no. 7, pp. 692–701, Jul. 2000.
- [11] S. Kuang, J. Wang, and C. Guo, "Modified Booth multiplier with a regular partial product array," IEEE Trans. Circuits Syst. II, vol. 56, no. 5, pp. 404–408, May 2009.
- [12] J. Kang and J. Gaudiot, "A simple high-speed multiplier design," IEEE Trans. Comput., vol. 55, no. 10, pp. 1253–1258, Oct. 2006.
- [13] F. Lamberti, N. Andrikos, E. Antelo, and P. Montuschi, "Reducing the computation time in (short bit-width) two's complement multipliers," IEEE Trans. Comput., vol. 60, no. 2, pp. 148–156, Feb. 2011.
- [14] Y. He and C. Chang, "A new redundant binary booth encoding for fast -bit multiplier design," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 56, no. 6, pp. 1192–1199, Jun. 2009.
- [15] Y. He, C. Chang, J. Gu, and H. Fahmy, "A novel covalent redundant binary booth encoder," in Proc. IEEE Int. Symp. Circuits Syst., 2005, vol. 1, pp. 69–72.
- [16] N. Besli and R. Deshmukh, "A novel redundant binary signed-digit (RBSD) Booth's encoding," in Proc. IEEE Southeast Conf., 2002, pp. 426–431.
- [17] Xiaoping Cui, Xin Chen and Fabrizio Lombardi, "A Modified Partial Product Generator for Redundant Binary Multipliers", IEEE TRANSACTIONS ON COMPUTERS, VOL. 65, NO. 4, APRIL 2016.
- [18] G. Dimitrakopoulos and D. Nikolos, "High-speed parallel- prefix VLSI Ling adders," IEEE Trans. Comput., vol. 54, no. 2, pp. 225–231, Feb. 2005.
- [19] NM Nayeem, Md AHossian, L Jamal and Hafiz Md. Hasan Babu, "Efficient design of Shift Registers using Reversible Logic," Proceedings of International Conference on Signal Processing Systems, 2009.