International Journal of Engineering Trends and Applications (IJETA) – Volume 5 Issue 2, Mar-Apr 2018

RESEARCH ARTICLE OPEN ACCESS

A Modified Partial Product Generator for Redundant Binary Multipliers

A.Hazarathaiah [1], K.Anusha [2]
Professor [1], UG Scholar [2]
Dept. of Electronics and Communication Engineering
Narayana Engineering College, Gudur, Nellore District
India

ABSTRACT

Because of its high modularity and carry-free addition, a redundant binary (RB) portrayal can be utilized when outlining superior multipliers. The traditional RB multiplier requires an additional RB fractional item (RBPP) push, on the grounds that a mistake adjusting word (ECW) is created by both the radix-4 Modified Booth encoding (MBE) and the RB encoding. This brings about in an additional RBPP aggregation organize for the MBE multiplier.

In this project, another RB adjusted halfway item generator (RBMPPG) is proposed; it evacuates the additional ECW and thus, it spares one RBPP aggregation organize. Hence, the proposed RBMPPG creates less halfway item pushes than a regular RB MBE multiplier. Recreation comes about demonstrate that the proposed RBMPPG based outlines essentially enhance the territory and power utilization when the word length of every operand in the multiplier is no less than 32 bits; these diminishments over past NB multiplier plans cause in a humble postpone increment (roughly 5 percent). The power-defer item can be diminished by up to 59 percent utilizing the proposed RB multipliers when contrasted and existing RB multipliers.

Keywords:- RBPP, RB

I. INTRODUCTION

A binary multiplier is an electronic circuit utilized as a part of advanced gadgets, for example, a PC, to duplicate two binary numbers. It is fabricated utilizing binary adders. An assortment of PC number juggling strategies can be utilized to execute an advanced multiplier. Most systems include registering an arrangement of partial products, and afterward summing the partial products together. This procedure is like the technique educated to elementary school kids for leading long multiplication on base-10 whole numbers, however has been adjusted here for application to a base-2 (binary) numeral framework.

Multiplication process has three principle steps :

1. Partial product generation.

2. Partial product decrease.

3. Last addition.

An effective multiplier ought to have following attributes:-

- Accuracy:- A great multiplier should give redress result.
- Speed:- Multiplier ought to perform activity at rapid.
- Area:- A multiplier ought to possesses less number of cuts and LUTs.
- Power:- Multiplier ought to devour less power.

An ordinary binary (NB) multiplication by advanced circuits incorporates three stages.

In the first step, partial products are created; in the second step, every partial product are included by a partial product diminishment tree until the point when two partial product lines remain. In the third step, the two partial product lines are included by a quick carry spread snake. Two strategies have been utilized to play out the second step for the partial product decrease. A first strategy utilizes four-two compressors, while a moment technique utilizes redundant binary (RB) numbers [6], [7]. The two strategies permit the partial product lessening tree to be diminished at a rate of 2:1.The redundant binary number portrayal has been acquainted by Avizienis [2] with perform marked digit math; the

RB number has the capacity to be spoken to in various ways. Quick multipliers can be outlined utilizing redundant binary addition trees [3], [4]. The redundant binary portrayal has likewise been connected to a floating-point processor and executed in VLSI [5]. Elite RB multipliers have turned out to be prevalent because of the profitable highlights, for example, high measured quality and without carry addition.

II. LITERATURE SURVEY

Addition is the most widely recognized and frequently utilized number-crunching activity on microchip and computerized flag processor, particularly advanced PCs. Likewise it fills in as building hinder for orchestrates all number juggling activity. In this manner, in regards to the productive execution of a number juggling unit, the binary adder structure turns into an exceptionally basic equipment unit.

In any book on PC number-crunching, somebody looks that there exists countless circuit number-crunching with various execution attributes and generally utilized as a part of training. Albeit numerous investigates managing binary adder structure have been done, the examinations in view of the execution investigation are just few. Among countless adders , Verilog code for swell ,carry select and carry look ahead to stress the regular execution properties have a place with their classes. The top notch comprises of the moderate swell carry adder with the littlest territory. In the below average carry select and carry skip adders with numerous levels have little territory necessitiesfurthermore, short end calculation times . From the second rate class carry look ahead adder and the fourth class, the parallel prefix adder speak to speediest addition plans with the biggest territory complexities. The last advance in finishing the augmentation technique is to include the last terms in the last adder. This is typically called "Vector-blending " adder. The decision of the last adder relies upon the structure of the gathering cluster . Following is the rundown of quick adders which are ordinarily utilized.

1. Carry look-ahead adder

2. Simple Carry skip adder

3. Carry select adder

4. Carry spare adder

III. EXISTING METHOD

A.Modified Booth Encoding:

Modified Booth Encoding (Radix-4) can adequately be connected to lessen the quantity of incomplete item columns to half in parallel multipliers. This is performed by gathering three adjoining multiplier bits (B = bn-1 bn-2… b0) to select one of the marked products as appeared in Table 2.2.1. The side bits of each gathering are covered with the two adjoining gatherings. The last piece in each gathering is alluded as reference bit. The primary gathering is coded by including "0" as reference bit preceding minimum huge piece position i.e., (b1, b0, 0).Depending on these select signs, the fractional item pushes are produced by selecting one of the mix {-2A,- A, 0, A, 2A} of the multiplicand (A = a 1an-2… a0). The twice of multiplicand (2A) in Table I is gotten by left moving the multiplicand by one piece position and refutation task is accomplished by upsetting each piece of multiplicand (i.e. one's complement)and including "1" to its minimum critical piece position.

TABLE I

Modified Booth's Encoding Unit for Radix-4

x_{i+1}	x_i	x_{i-1}	Y	Comments
0	0	0	0	String of Zeros
0	0	1	1.A	End of 1's
0	1	0	1.A	A single 1
0	1	1	2.A	End of 1's
1	0	0	-2.A	Beginning of 1's
1	0	1	-1.A	A single 0
1	1	0	-1.A	Beginning of 1's
1	1	1	0	String of ones

B.Partial Product Generator:

A product shaped by increasing the multiplicand by one digit of the multiplier when the multiplier has in excess of one digit. Halfway products are utilized as middle of the road ventures in ascertaining bigger products. Halfway product generator is intended to create the product by duplicating the multiplicand A by 0, 1, - 1, 2, - 2, - 3,- 4, 3, 4. For product generator, duplicate by zero means the multiplicand is increased by "0".Multiply by "1" implies the product still continues as before as the multiplicand esteem.

Duplicate by "- 1" implies that the product is the two's supplement type of the number. Duplicate by "- 2" is to move left one piece the two's supplement of the multiplicand esteem and increase by "2" implies simply move left the multiplicand by one place. . Increase by "- 4" is to move left no good the two's supplement of the multiplicand esteem and duplicate by "2" implies simply move left the multiplicand by two place. Here we have an odd various of the multiplicand, 3Y, which isn't quickly accessible. To create it we have to play out this past include: 2Y+Y=3Y. In any case, we are planning a multiplier for particular reason and along these lines the multiplicand has a place with a formerly known arrangement of numbers which are put away in a memory chip. We have endeavored to exploit this reality, to facilitate the bottleneck of the radix-8 engineering, that is, the age of 3Y. In this way we endeavor to accomplish a superior general increase time, or if nothing else practically identical to the time we could acquire utilizing radix-4 engineering (with the additional favorable position of utilizing a less number of transistors). To produce 3Y with 8-bit words we just need to include 2Y+Y, that is, to include the number with a similar number moved one position to one side.

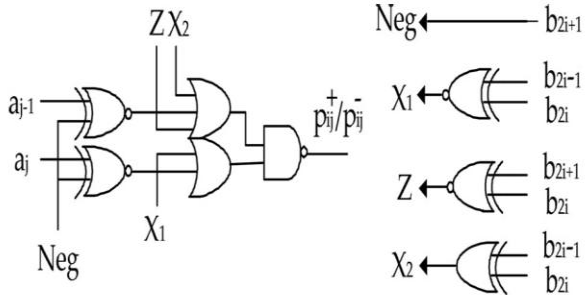

Fig 1 An Encoder and Decoder of the MBE scheme

By utilizing this, we create partial product columns i.epij + and pij-. Both MBE and RB coding plans present blunders and two redress terms are required:

1) when the NB number is changed over to a RB organize, - 1 must be added to the LSB of the RB number;

2) when the multiplicand is increased by - 1 or - 2 amid the Booth encoding, the number is reversed and +1 must be added to the LSB of the partial product. A solitary ECW can remunerate mistakes from both the RB encoding and the radix-4 Booth recoding. The customary partial product design of a 8-bit MBE multiplier [6],[7] is appeared in Fig.2.4 , where b_p speaks to the bit position, p+ij or pij-is created by utilizing an encoder and decoder in fig 2.3.

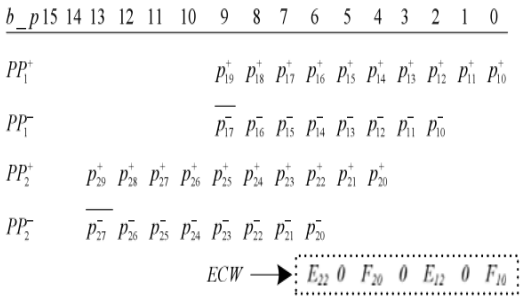

Fig .2 Conventional RBPP architecture for an 8-bit MBE multiplier.

Generally ECW takes the form as follows:

$$ECW = E_{(N/4)2}0F_{(N/4)0}.....0E_{i2}0F_{i0}.....0E_{12}0F_{10} \quad \ldots\ldots\ldots(1)$$

where i represents the i^{th} row of RBPP , $E_{i2} \in \{0,1\}$ and $F_{i0} \in \{0,\bar{1}\}$; In F_{i0}, a -1 correction term is always required by RB coding. If F_{i0}also corrects the errors from the MBE recoding, then the correction term cancels out to 0. That is to say that if the multiplicand digit is inverted and added to 1, then $F_{i0}$is 0,otherwise F_{i0} is -1.. The error-correcting digit $E_{i2}$is determined only by the Booth encoding:

$$E_{i2} = \begin{cases} 0 & nonegativeencoding \\ 1 & negativeencoding \end{cases} \quad \ldots\ldots\ldots\ldots(2)$$

p^+_{19} and p^+_{18} are sign extension bits, so

$$p^+_{19} = \overline{p^+_{18}},$$

$$p^+_{18} = \overline{b_1}\,\overline{b_0}\cdot 0 + \overline{b_1}b_0\cdot a_7 + b_1\overline{b_0}\cdot\overline{a_7} + b_1b_0\cdot\overline{a_7}$$
$$= \overline{b_1}b_0\cdot a_7 + b_1\overline{a_7}. \quad \ldots\ldots(3)$$

Power consumption is one of the basic parameters of any kind of Integrated Circuit (IC). Power and performance are always traded off to meet the system requirements. Power has a direct impact on system cost. If an IC is consuming more power than a better cooling mechanism would be required to keep the circuit in normal conditions. Otherwise its performance is degraded and on continuous use it may be permanently damaged.

Clock gating is one of the power-saving techniques used on many synchronous circuits including the Pentium 4 processor. To save power, clock gating refers to adding additional logic to a circuit to prune the clock tree, thus disabling portions of the circuitry where flip flops do not change state .
Pipelining is a technique used along with voltage scaling to reduce the power consumption. The idea is to insert registers after some appropriate distance in the circuit. The system response becomes faster than before. In order to maintain the delay, the supply voltage is reduced which reduces the power consumption.

Disadvantages of the Existing Method

- power dissipation :
  - Power dissipation is also more.
- More complex :
  - As the number of gates increases complexity also increases.
- More delay.
- number of accumulation stages are more.

Applications

- Used to synthesize any arbitrary Boolean functions.

IV. PROPOSED METHOD

A. BLOCK DIAGRAM

The block diagram of 32-BIT RB multiplier consists of 3 stages:

- Booth encoder and partial product generator stage (BEPPG stage)
- Redundant binary adder summing tree stage (RBA summing stage)
- Redundant binary to NB conversion stage (RB-to-NB stage)

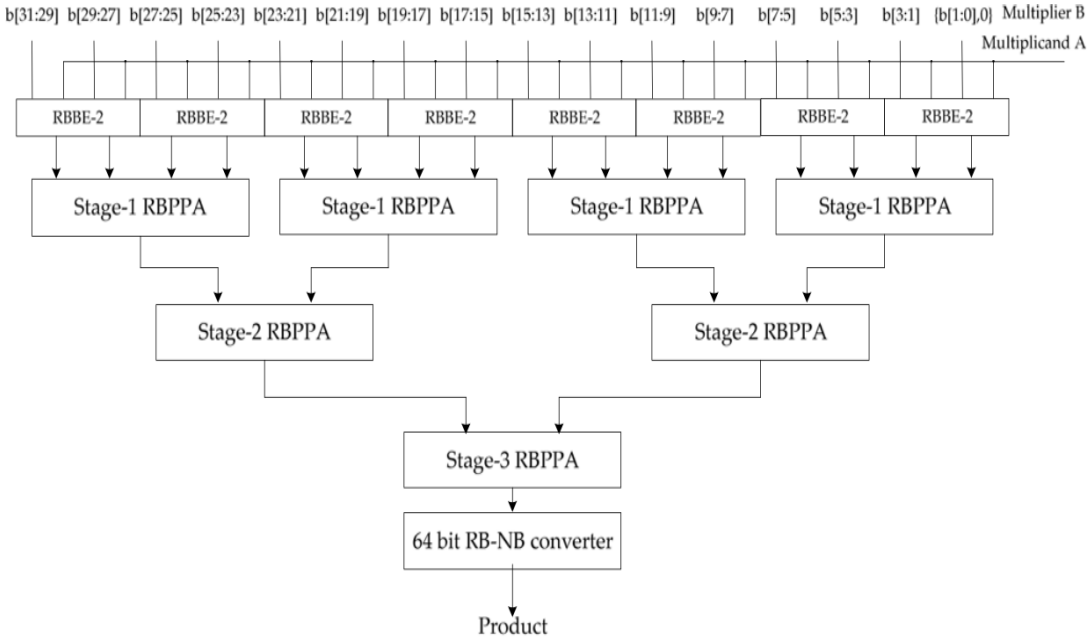

Fig 3 Block diagram of a 32-bit RB multiplier using RBMPPG-2.

Step 1: Booth Encoder and Partial Product Generator stage (BEPPG stage):
Booth encoder and partial product generator affect the efficiency of the partial product generation. The number of partial products that can be saved by this stage impacts the cost, performance, and power consumption of the RB summing tree and the multiplier as a whole. In the first stage, 16 CRBBE-3 slices are used to generate the control signals from the multiplier. The hard multiple 5X is generated. The multiplicand bits are shifted and selected into 16 rows of RB partial products in 16 slices of RBPPG.

Step 2: Redundant Binary Adder summing tree stage (RBA summing stage):
Here all the partial products generate 64 bits. These bits are added by the Redundant Binary

Adders (RBA)denoted as1, 2, 3. RBA had divided into sub blocks with 64 bits.

Step 3: Redundant binary to NB conversion stage (RB-to NB stage):

An RB-to-NB converter converts the final accumulation result to NB representation. Due to the unequal delay profile of the final RB result bits, the conversion can be carried out in uneven groups of consecutive digits according to their arrival time. The carry generation of the next group of digits can be evaluated with a carry-look ahead adder as they do not depend on the final summation results in the RBA tree stage as shown in fig 3.1.

ECW_1 can be generated by PP_1 and is expressed as

$ECW_1 = 0\ E_{12}0\ F_{10}$……..(4)

ECW_2 can be generated by PP_2 (an extra ECW) is left as the last row and is expressed as

$ECW_2 = 0\ E_{22}0\ F_{20}$.....(5)

To eliminate a RBPP accumulation stage, ECW_2 needs to be incorporated into PP_1 and PP_2.

A modified radix_4 Booth encoding and a decoding circuit for the partial product PP^+_2 are proposed in fig 3.2; an extra three inputOR gate is then added to the design of [10] Fig. 2.2.1. The three inputs of the additional OR gate are $\bar{b}_5$, $\bar{b}_4$ and $\bar{b}_3$. when $b_5b_4b_3$ =111,it is clear that $\bar{b}_5\bar{b}_4\bar{b}_3$ =000, p^+_{2i}=1, and PP^+_2 is set to all ones. So, E_{22} and F_{20} in $ECW_2$are now determined by $b_7b_6b_5$without b_4 , b_3. Although the complexity is slightly increased compared with the previous design Fig. 2.3, the delay stage remains the same. Here E_{22} and F_{20} can be represented by E_2 ,$\bar{q}_{2(-2)}$ and $\bar{q}_{2(-1)}$ as follows:

$$E_2 = \begin{cases} E_{22}, & F_{20} = 0 \\ E_{22} - 1, & F_{20} = -1, \end{cases} \quad \text{….(6)}$$

$$q^-_{2(-2)} = q^-_{2(-1)} = \begin{cases} 0, & F_{20} = 0 \\ 1, & F_{20} = -1. \end{cases}$$

……(7)

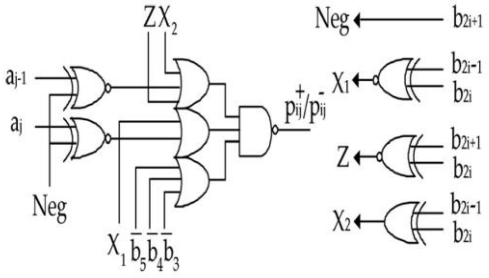

Fig 4.modified radix-4 Booth encoding and decoding scheme

also $\bar{q}_{2(-2)}$ and $\bar{q}_{2(-1)}$ can be expressed as follows

$$q^-_{2(-2)} = q^-_{2(-1)} = \bar{b}_5.$$

……(8)

E_2 can be incorporated into the modified partial products Q^+_{19} , Q^+_{18} , Q^-_{21} and Q^-_{20} by replacing p^+_{19}, p^+_{18} and p^-_{21} , p^-_{20}

$$\begin{aligned} Q^+_{18} = {} & \overline{\bar{b}_1 b_0 a_7 + b_1 \bar{a}_7} \cdot \left(\bar{b}_5 \cdot \overline{b_7 + b_6 a_0 + b_6 a_1} + b_5 b_7 \bar{b}_6 \bar{a}_1 \bar{a}_0\right) \\ & + \left(\bar{b}_1 b_0 a_7 + b_1 \bar{a}_7\right) \cdot \left[\bar{b}_5 \cdot \left(b_7 + b_6 a_0 + b_6 a_1\right)\right. \\ & \left. + b_5 \cdot b_7 \bar{b}_6 \bar{a}_1 \bar{a}_0\right], \end{aligned}$$

……(9)

$$\begin{aligned} Q^+_{19} = {} & \overline{\bar{b}_1 b_0 a_7 + b_1 \bar{a}_7} \left(\overline{\bar{b}_5 \cdot \overline{b_7 + b_6 a_0 + b_6 a_1}}\right) \\ & + \left(\bar{b}_1 b_0 a_7 + b_1 \bar{a}_7\right) \cdot b_5 \cdot b_7 \bar{b}_6 \bar{a}_1 \bar{a}_0, \end{aligned}$$

……(10)

$$Q^-_{20} = \bar{b}_6 a_0 + \bar{b}_5 \bar{a}_0.$$

……..(11)

$$\begin{aligned} Q^-_{21} = {} & \bar{b}_5 \cdot \left(\bar{b}_7 \cdot \overline{b_6 a_0 \bar{a}_1 + b_6 \bar{a}_0 a_1} + b_7 \cdot \overline{\bar{b}_6 a_0 + b_6 a_1}\right) \\ & + b_5 \cdot \left(\bar{b}_7 \cdot \overline{\bar{b}_6 \bar{a}_1 + b_6 \bar{a}_0} + b_7 \cdot \overline{\bar{b}_6 + \bar{a}_1 \bar{a}_0 + a_1 a_0}\right), \end{aligned}$$

……..(12)

Sign extension problem and solution

In signed multiplication, the sign bit of a partial product row would have to be extended all the way to the MSB position which would require the sign bit to drive that many output loads. This makes the partial product rows unequal in length as shown in Figure 3.3. The first row spans 16 bits (pp_{00} to the leftmost pp_{80}), the second row 14 bits (pp_{01} to the leftmost pp_{81}), the third row 12 bits(pp_{02} to the

leftmost pp_{82}), and the fourth row 10 bits (pp_{03} to the leftmost pp_{83}).

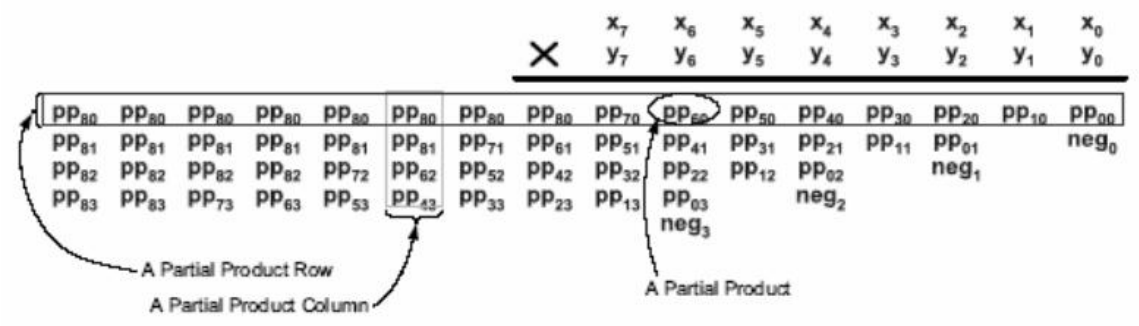

Fig5.Multiplication with sign extension

In Interleaving method, the circuit parts are copied and the recurrence of activity is divided for the two bits of the circuit. One half works on positive clock edge and alternate works on the negative clock edge. A multiplexer is utilized to select the outcome one by one from the two parts of the circuit. The throughput of the framework is multiplied by along these lines, which gives a period edge. This time edge can be utilized for supply voltage lessening to decrease control utilization.

Clock gating is one of the power-sparing procedures utilized on numerous synchronous circuits including the Pentium 4 processor. To spare power, clock gating alludes to adding additional rationale to a circuit to prune the clock tree, consequently crippling bits of the hardware where flip lemon don't change state .

Pipelining is a strategy utilized alongside voltage scaling to lessen the power utilization. The thought is to embed enlists after some fitting separation in the circuit. The framework reaction turns out to be speedier than previously. Keeping in mind the end goal to keep up the postponement, the supply voltage is decreased which lessens the power utilization.

A.ALGORITHM

Step-1:Start

Step-2: Assign inputs A 31:0

B 31:0

Step-3: Perform operations to obtain partial product rows

Step-4: Partial product accumulation stages

Step-5: A 64-bit RB-NB converter

Step-6: Final Product

Step-7: Stop

These are the basic steps involved in the RB multiplier using the proposed RBMPPG-2.

B.FLOW CHART

The flowchart of the RB multiplier using the proposed RBMPPG-2 is as shown below.

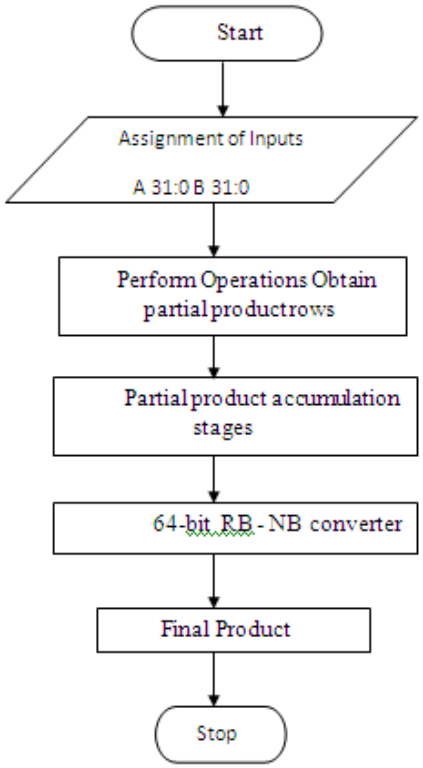

Fig 6 : Flow chart for proposed method

V. RESULT & DISCUSSIONS

1.Block Diagram

Both the multipliers have been simulated using a common test setup . A variety of test vectors are applied at the input ports and the resulting outputs are analyzed using Spectre waveform window. The power consumption and delay is measured in that window. The power is calculated by finding the average current and multiplying it with the supply voltage.

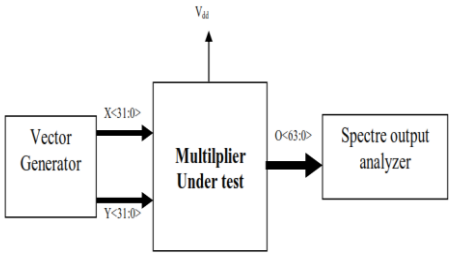

Fig 7.Simulation setup

The block diagram shows the modified partial product generator for redundant binary multipliers. Here inputs are A and B with 32 bits each. The output will be of 64 bits.

Fig 8.Schematic Block Diagram

The output waveform of the proposed RBMPPG is shown in fig 5.3. as follows

2.Discussions

The performance of various 2^n-bit RB multipliers using the proposed RBMPPG-2 is assessed; the results are compared with

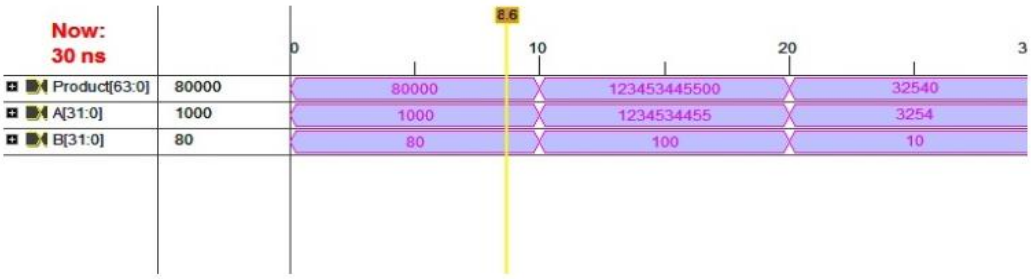

Fig 9. Output Waveform

NBBE-2, CRBBE-2 and RBBE-4 [11] multipliers that are the latest and best designs found in the technical literature.

All designs of RB multipliers use the RBFA and RBHA of [8]. An RB-NB converter is required in the final stage of the RB multiplier to convert the summation result in RB form to a two's complement number. It has been shown that the constant-time converter in [8] does not exist [12], [13]. However, there is a carry-free multiplier that uses redundant adders in the reduction of partial products by applying on-the-fly conversion in parallel with the reduction and generates the product without a carry-propagate adder .

A hybrid parallel-prefix/carry-select adder is used for the final RB-NB converter. The NBBE-2 multiplier design uses the same encoder and decoder as shown in Fig. 2.2.1. Four-two compressors [14], [15], [16] are used in the partial product reduction tree. The extra ECW in the NB multiplier designs is also modified as proposed in [10].The multiplier designs are described at gate level in Verilog HDL and verified by Synopsys VCS using randomly generated input patterns; the designs are synthesized by the Synopsys Design Compiler using the NanGate 45 nm Open Cell Library. In the simulation of each design, a supply voltage of 1.25 V and room temperature are assumed. Standard buffers of a 2X strength are used forboth the input drive and the output load. The option for logic structuring is turned off to prevent the tool from changing the structure of the unit cells. The average power consumption is found

using the Synopsys Power Compiler with back annotated switching activity files generated from 2,500 random input vectors. The device utilization summary is tabulated as below table 5.1

- The delay of this proposed 32-bit partial product generator is 1.57ns
- The power consumed during this is 3,090µW.
- The power delay product is reduced up to 59 percent

TABLE II
Device Utilization

Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice LUTs	1902	2400	79%
Number of fully used LUT-FF pairs	0	1902	0%
Number of bonded IOBs	128	132	96%

VI. CONCLUSION

Another modified RBPP generator has been proposed in this paper. This outline wipes out the additional ECW that is presented by past plans. Thusly, a RBPP collection arrange is spared because of the end of ECW. The new RB partial product age procedure can be connected to any 2n-bit RB multipliers to diminish the quantity of RBPP lines from N/4+1 to N/4. Recreation comes about have demonstrated that the execution of RB MBE multipliers utilizing the proposed RBMPPG-2 is enhanced altogether as far as postponement and region. The proposed plans accomplish critical decreases in territory and power utilization when the word length is no less than 32 bits. PDP is a generally utilized metric for consolidated execution as far as deferral and power utilization. Here PDP can be decreased by up to 59 percent utilizing the proposed RB multipliers when contrasted and existing RB multipliers. Thus, the proposed RBPP age strategy is an exceptionally helpful method when planning territory and PDP proficient energy of-two RBMBE multipliers.

VII. FUTURE SCOPE

In future RBR technique is used for Digital Signal Processing, Fast Fourier Transformations and multimedia applications.

REFERENCES

[1]Naveen Kumar, Manu Bansal, Navnish Kumar" VLSI Architecture of Pipelined Booth Wallace MAC unit" International Journal of Computer Application(0975-8887)

[2] A. Avizienis, "Signed-digit number representations for fast parallel arithmetic," IRE Trans. Electron.Comput., vol. EC-10, pp. 389–400, 1961.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] Y. Kim, B. Song, J. Grosspietsch, and S. Gillig, "A carry-free 54b_54b multiplier using equivalent bit conversion algorithm," IEEE J. Solid-StateCircuits, vol. 36, no. 10, pp. 1538–1545, Oct. 2001.

[9] W. Yeh and C. Jen, "High-speed booth encoded parallel multiplier design," IEEE Trans. Comput., vol. 49, no. 7, pp. 692–701, Jul. 2000.

[10]S.Kuang,J.Wang,andC.Guo,"ModifiedBoothmultiplierwitharegular partial product array," IEEE Trans. Circuits Syst. II,vol.56,no.5, pp. 404–408, May 2009.

[11] 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.

[12] Y. Kim, B. Song, J. Grosspietsch, and S. Gillig, "Correction to 'a carry-free 54b_54b multiplier using equivalent bit conversion algorithm'," IEEE J. Solid-State Circuits, vol. 38, no. 1, p. 159, Jan. 2003.

[13] W. Rulling, "A remark on carry-free binary multiplication," IEEE J. SolidState Circuits, vol. 38, no. 1, pp. 159–160, Jan. 2003.

[14] S. Hsiao, M. Jiang, and J. Yeh, "Design of high-speed low-power 3–2 counter and 4–2 compressor for fast multipliers," Electron.Lett., vol. 34, pp. 341–343, Feb. 1998.

[15] C. 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.