Design and Analysis of Approximate Redundant Binary Multipliers

[1] N.SWETHA, B. Tech, M.Tech. KATRVARH GOVIND [2], LAKAVATH SRIKANTH [3], CHUNCHU PAVAN KUMAR [4], CH BANU KARTHIKEYA [5].

[1] ASSISTANT PROFESSOR OF ECE IN MALLA REDDY INSTITUTE OF TECHNOLOGY & SCIENCE, MAISAMMAGUDA, MEDCHAL (M), HYDERABAD-500100, T. S.

[2, 3, 4, 5.] FINALYEAR STUDENTS FROM DEPT OF ECE IN MALLA REDDY INSTITUTE OF TECHNOLOGY & SCIENCE, MAISAMMAGUDA, MEDCHAL (M), HYDERABAD-500100, T. S.

ABSTRACT:

This paper presents the design redundant Binary multiplier for 32*32bit number multiplication. Modern computer system is a dedicated and very high speed unique multiplier. Therefore, this paper presents the design a Redundant Binary multiplier. The proposed system generates M, N and interconnected blocks. By extending bit of the operands and generating an additional product the proposed system multiplier is obtained. Multiplication operation is performed by the Proposed system is efficient with the less area and it reduces delay i.e., speed is increased.

Keywords:

Redundant binary, modified booth encoding, partial products, proposed level logic unit.

INTRODUCTION:

Multiplication is a fundamental operation in most signal processing algorithms. Multipliers have large area, long latency and consume considerable power. Therefore low-power multiplier design has been an important part in low-power VLSI system design. There has been extensive work on low-power multipliers at technology, physical, circuit and systems.

A system's performance is generally determined by the performance of the multiplier because the multiplier is generally the slowest element in the system. Furthermore, it is generally the most area consuming. Hence, optimizing the speed and area of the multiplier is a major design issue. However, area and speed are usually conflicting constraints so that improving speed results mostly in larger areas.

As a result, a whole spectrum of multipliers with different area-speed constraints has been designed with fully parallel. The high speed multipliers and pipelined multipliers are used for digital signal processing (DSP) applications such as for multimedia and communication systems. High speed DSP computation applications such as Fast Fourier transform (FFT) require additions and multiplications. The conventional modified Booth encoding (MBE) generates an irregular partial product array because of the extra partial product bit at the least significant bit position of each partial product row.

Therefore papers [4] presents a simple approach to generate a regular partial product array with fewer partial product rows and negligible overhead, thereby lowering the complexity of partial product reduction and reducing the area, delay, and power of MBE multipliers. But the drawback of this multiplier is that it functions only for signed number operands. The Existed system of RB multiplier is popular because of its high modularity and carries free addition. This RB multiplier mainly consists of four stages MBE, RB Encoding, Error correction word. In this existed system Column bypass technique can be used but this one occupied more area. So we propose the Dual logic level technique.

EXISTED METHOD:

Introduction to Redundant Binary Multiplier:

The existed technique is design a Redundant Binary Multiplier by means of the usage of Column Bypass Multiplier. The block diagram of conventional method is proven in below.

It has four stages.

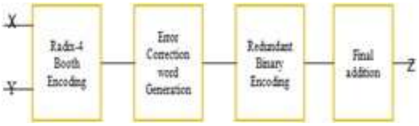
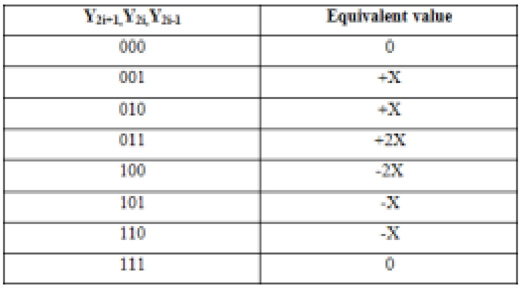


Fig 1 Block diagram of Redundant Binary Multiplier.

Radix-4 Booth Encoding:

Radix-4 Booth Encoding scheme is introduced for lessen the partial product with the aid of half. This technique calls as Modified Booth Encoding. The more speed floating factor processors are designed by usage of this approach. Multiplication system includes two operands of bits. Consequently X= xn-1 xn-2. 2 x1 x0 stands for multiplicand bits, and Y= yn-1 yn-2.... y2 y1 y0 stands for multiplier bits. From that the multiplier bits are combined into three and produce the only multiplier bit in Radix-4 Booth Encoding, therefore the partial product row reduced by half. From the under table definitely explain a grouping of the multiplier bits, where x describes the multiplicand bit. The operation includes -1,-2 means 1's and 2's complement of the multiplicand bits. The obtained partial products rows are shifted by 2-position and then placed. The height of the partial product array is (N/2).



$Y_{2i+1,} Y_{2i,} Y_{2i-1}$	Equivalent value
000	0
001	+X
010	+X
011	+2X
100	-2X
101	-X
110	-X
111	0

Table 1 Radix-4 booth encoding

Error Correcting Word:

The mistake correcting word is formed through modified booth encoding and redundant binary coding. In redundant binary coding the mistake is fashioned at each stage of converting normal binary into redundant binary numbers.

While the multiplier bit has the value of -1 or -2, it requires the error value to produce the correct value. The mistake correcting word is usually written in the shape of

ECW= E 0 F 0 E 0 F. . . .

Where, E represents the error due to booth coding and F term represents the error due to redundant binary coding. The E term has two values 0 and 1.when the multiplier bit is -1 or -2, the error bit is 1 otherwise the value is 0. The F term also has two values 0 and -1. When converting the normal binary into redundant binary the error value is -1 otherwise the value is 0. Because of the error correcting word the accumulation stage is accelerated. The accumulation stage is finding by way of power of two word length (2n bit). The variety of RBPP accumulation stage (NRBPPAS)

NRBPPAS = [log2 (N/4+1)]
= n-1, if N=2n

For 8-bit multiplication the accumulation stage is
NRBPPS = n-1; N=23
= 3-1=2

2 accumulation levels are required for 8-bit multiplication. Similarly 32-bit multiplication contains 4 accumulation stages. If the mistake correcting word is removed from the multiplication then the accumulation stage is reduced to few and 25% of memory is saved. The 24X24 and 54X54 bit multiplication does not have an error correcting word; the ECW is present only on normal matrix.

RB Encoding:

The RB coding is used for the generation of RB partial product. Each digit should be transformed to the sign and accurate value with using OR-gate and Exclusive-OR-gate that increase the hardware and delay time. In our RB structure, two NB wide variety creates the RB partial product simplest with the aid of inverting one of the pair. No additional hardware is needed.

The two normal binary bits are used to represents one RB digit. The RB digit most effective consists of three numbers they are (0, 1, and -1). If both binary words are zero or one then the RB digit is zero. If the binary word has alternative value the RB digit is has the value of 1 or -1. The -1 is nothing but a complement operation.

X	Y	RB Digit
0	0	0
0	1	-1
1	0	1
1	1	0

Table. 2 RB Encoding

Ripple Carry Adder:

Logical circuit with a couple of full adders may be used for including N-bit numbers and every full adder inputs a Cin, which is the Cout of the preceding adder. Such type of adder is known as Ripple Carry Adder, since every carry bit "ripples" to the following full adder. So ripple carry adder in virtual electronics is that circuit which produces the arithmetic sum of binary numbers which can be built with full adders related in cascaded with the carry output from every full adder related to the carry input of the subsequent full adder inside the chain. From which it could be noticed that bits a0and b0 within the figure represent the least significant bits of the numbers which is to be added and sum in form of output represented by the bits. In this we can use the column bypass multiplier in existed system. This technique occupies the more area so we can use the proposed system that is dual logic level technique.

PROPOSED SYSTEM:

The gates in the proposed system multiplier are always active regard of input logics. In, proposed system multiplier design is proposed in which the operations are disabled if the corresponding bit in the multiplicand is 0. Fig. 1 shows a N×N Proposed system multiplier, it can be seen that the M0, M1…Mn done their operations and the outputs are passed to interconnected Block and N-Block simultaneously. Depends on the preference of operation the proposed level logic gives the N-block output to interconnected block and vice versa. Therefore, the output of the adders in both diagonals is 0, and the output sum bit is simply equal to the third bit.The above fig. 1 shows the 4*4 high performance Proposed system multiplier reduced the timing waste occurring in traditional circuits that use the critical path cycle as an execution cycle period. The basic concept is to execute a shorter path using proposed logic. Since most paths execute in a cycle period that is much smaller than the critical path delay. The same architecture is extended up to 32*32 bits. Proposed system widely been adopted in multipliers since it can reduce the number of partial product rows to be added, thus reducing the size and enhancing the speed of the reduction tree. The least significant bit position of each partial product row encoding, leading to an irregular partial product array and a complex reduction tree. Therefore, the proposed system multipliers with partial product array produce a very high speed.

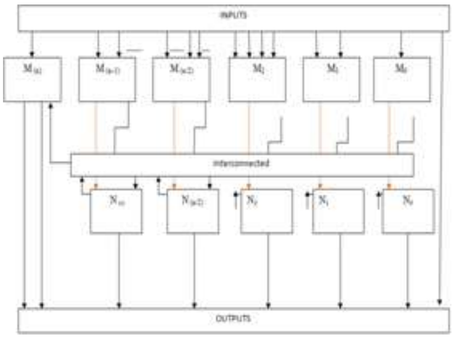


Fig. 2. 4 x 4 High Performance Proposedsystem Multiplier

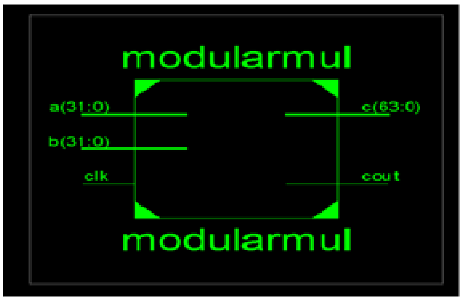


Fig. 3 R.T.L Schematic

ISSN: 2278-4632
Vol-13, Issue-11, November 2023

The above fig. 2 shows the R.T.L schematic of high performance Proposed system multiplier and fig. 3 shows the technical schematic one of the LUT block of high performance Proposed system multiplier.

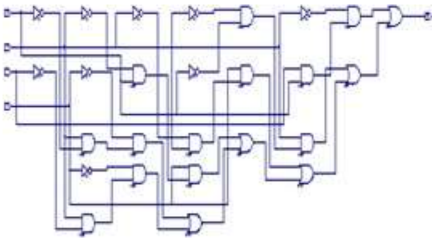


Fig. 3 LUT in technical Schematic

The below figure 4 shows the output waveform of 32*32 bit Proposed system multiplier.

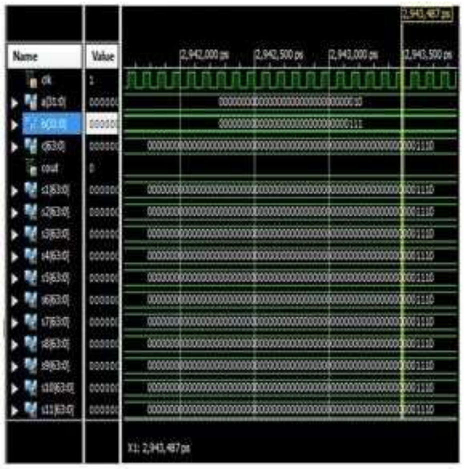


Fig. 4 OUTPUT Waveforms

The below table 1 shows the comparison of existed system and proposed system with area and delay

System/parameter	Area[kb]	Delay[ns]
Existed System	382256	410
Proposed system	304820	138

Table.1 comparison table

CONCLUSION

The proposed system generates M, N and interconnected block. The each block consists of gates and the row in the architecture is lesser than existed multiplier. By generating a product with Proposed system multiplier is obtained. Multiplication operation is performed by the Proposed system unit is better performance than existed multiplier. The required hardware and the chip memory reduces and it reduces delay i.e., speed is increased.

FUTURE SCOPE: Many of an RBR's properties differ from those of regular binary representation systems. Most importantly, an RBR allows addition without using a typical carry.[1] When compared to non-redundant representation, an RBR makes bitwise logical operation slower, but arithmetic operations are faster when a greater bit width is used.[2] Usually, each digit has its own sign that is not necessarily the same as the sign of the number represented. When digits have signs, that RBR is also a signed-digit representation. An RBR is a place-value notation system. In an RBR, digits are pairs of bits, that is, for every place, an RBR uses a pair of bits. The value represented by a redundant digit can be found using a translation table. This table indicates the mathematical value of each possible pair of bits.

REFERENCES

[1] R. Vattikonda, W. Wang, and Y. Cao, “Modelling and minimization of pMOS NBTI effect for robust naometer design,” in Proc. ACM/IEEEDAC, Jun. 2004, pp. 1047–1052.

[2] H. Abrishami, S. Hatami, B. Amelifard, and M. Pedram, “NBTI-aware flip-flop characterization and design,” in Proc. 44th ACM GLSVLSI, 2008, pp. 29– 34.

[3] S. V. Kumar, C. H. Kim, and S. S. Sapatnekar, “NBTI-aware synthesis of digital circuits,” in Proc. ACM/IEEE DAC, Jun. 2007, pp. 370–375.

[4] A. Calimera, E. Macii, and M. Poncino, “Design techniques for NBTI tolerant power-gating architecture,” IEEE Trans. Circuits Syst., Exp.Briefs, vol. 59, no. 4, pp. 249–253, Apr. 2012.

[5] K.-C. Wu and D. Marculescu, “Joint logic restructuring and pin reordering against NBTI-induced performance degradation,” in Proc. DATE, 2009, pp. 75–80.

[6] Y. Lee and T. Kim, “A fine-grained technique of NBTI-aware voltage scaling and body biasing for standard cell based designs,” in Proc. ASPDAC, 2011, pp. 603–608.

The above comparison table shows that area of the proposed system is less than existed system and delay is also efficient.

[7] M. Basoglu, M. Orshansky, and M. Erez, “NBTIaware DVFS: A new approach to saving energy and increasing processor lifetime,” in Proc.ACM/IEEE ISLPED, Aug. 2010, pp. 253–258.

[8] K.-C. Wu and D. Marculescu, “Aging-aware timing analysis and optimization considering path sensitization,” in Proc. DATE, 2011, pp. 1–6.

[9] K. Du, P. Varman, and K. Mohanram, “High performance reliable variable latency carry select addition,” in Proc. DATE, 2012, pp. 1257–1262. [10] A. K. Verma, P. Brisk, and P. Ienne, “Variable latency speculative addition: A new paradigm for arithmetic circuit design,” in Proc. DATE, 2008, pp. 1250–1255.