International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2014): 5.611

Fast Computation Using High Radix Signed Digit Number with Different Adders

Mahendra Prasad Sharma

Ph.D Scholar, Department of CSE, IIMT College of Engineering, Greater Noida- India

Abstract: ***The proposed system is an efficient implementation of 16-bit Multiplier- Accumulator using Radix-8 and Radix-16 Modified Booth Algorithm and seven different adders (SPST Adder, Parallel Prefix Adder, Carry Select Adder, Error Tolerant Adder, Hybrid Prefix Adder, Modified Area Efficient Carry Select Adder, Parallel Binary Adder) are using VHDL. This proposed system provides low power, high speed and less delay. The comparison between the power consumption (mw) and estimated delay (ns) of both Booth multipliers is calculated. The application of digital signal processing like fast Fourier transform, finite impulse response filters and convolution requires high speed and low power MAC (Multiplier and Accumulator) units to construct an adder. Speed of operation can be improved and dynamic power can be reduced by reducing the glitches (1 to 0 transition) and spikes (0 to 1 transition). The adder designed using SPST avoids the unwanted glitches and spikes, minimizing the switching power dissipation and hence the dynamic power. The speed can be improved by reducing the number of partial products to half by grouping of bits from the multiplier term. The proposed Radix-8and Radix-16 Modified Booth Algorithm MAC with SPST reduces the delay with less power consumption as compared to array MAC.***

Keywords: Radix-8 modified booth algorithm Radix -16 modified booth algorithm, Digital Signal Processing, VHDL

1. Introduction

Multiplication is an important operation in digital signal processing algorithms. It needs more area, and consumes considerable power. Therefore, there is requirement of designing low power Booth Algorithm is a multiplication algorithm that multiplies two signed binary numbers in two's complement notation.

Both multiplication is a technique that allows for a smaller, faster multiplication circuit, by recording the numbers that are multiplied. It is a standard technique that used in chip design and provides significant improvements over the long multiplication technique.

The performance of the multiplier depends on the type of adder which is using in the MAC. By combining the multiplication with the accumulation and development of a hybrid type of adders like Parallel prefix adder and carry save adder, performance has improved. Then the accumulator having the greatest delay parallel prefix adder as compared to carry save adder but the overall performance was high. Several commercial processors have selected the radix-8 multiplier architecture to increase their speed of operation, thereby reducing the number of partial products in the multiplication terms. Radix-8 encoding reduces the digit number length in a signed digit representation as compared to Radix-2 Multiplication. Its performance bottleneck is the generation of the term 3X (Multiplicand), also referred to as hard multiple. The proposed MAC accumulates intermediate results in the kind of sum and carry bits instead of the output of the final adder, which has optimized the pipeline system to improve performance.

Digital multiplication is an obligatory and critical element in microprocessors, signal processing and arithmetic based systems.

The modified Booth's algorithm based on a radix-8, generally called Booth-2, is the most popular approach for implementing fast multipliers using parallel encoding. With the recent rapid advances in multimedia and communication system devices, real-time signal processing like audio signal processing, video/image processing, or large-capacity data processing are increasingly being demanded. Multiplication and addition arithmetic determines the execution speed and performance of the entire calculations.

Because the multiplier requires the longest delay among the basic operational blocks in digital systems, the critical path is determined by the multiplier; in general Multi-operand addition is a part of many complex arithmetic algorithms, such as multiplication and certain DSP algorithms.

One of the popular multi-operand adders is the carry-save adder capable of adding more than two operands at a time. The objective of this paper is to introduce the flexibility of adding three-input operands to a regular adder, thereby eliminating the need of a special adder to do the same.

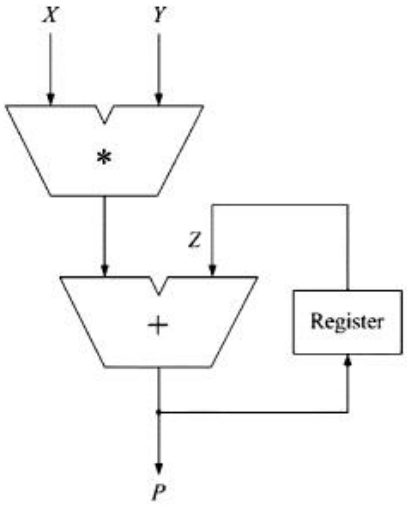

Figure 1: Hardware Architecture General MAC Array Multiplier

Here in this designing, the VHDL designing used the M o d e l S i m 6.5c software. General architecture of MAC is shown in figure 1. This executes the multiplication operation by multiplying the multiplier and the multiplicand. Multiplier is considered as X and multiplicand is Y. This is added to the previous multiplication result Z as the accumulation step.

2. Types of Adders

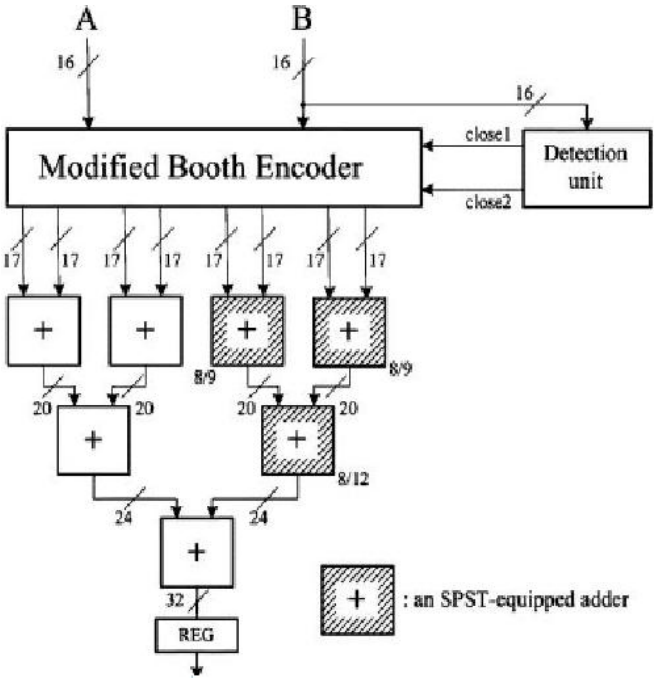

Figure 2: Proposed Low Power SPST Equipped Multiplier

Logic gates are used to implement the latches and the sign extension circuits in order to reduce the additional overhead as for as possible. Low power adder / Subtractor consists of the above blocks, Figure 2. shows the Proposed Low Power SPST Equipped Multiplier which has the following parts:

1. Latch
2. Detection logic
3. Sign extension logic

All the arithmetic operations can be implemented using

Low-power VLSIsystem design, where the fundamental operation in the signal processing.

2.2 Kongge Stone Adder

The design of sparse adders relies on the use of a sparse parallel-prefix (Kongge-Stone) carry computation unit and carry-select (CS) blocks. Only the carries at the boundaries of the carry-select blocks are computed in the adder, saving considerable amount of area in the carry-computation unit. A 32-bit adder with 4-bit sparseness is shown below. The block which is used to select carry, computes two sets of sum bits corresponding to the two possible values of the incoming carry. When the actual carry is computed, it selects the correct sum without any delay overhead. The Inter Adder structure and the computation unit is shown in figure 3 and 4.

2.1 SPST ADDER

In this Adder, the 16-bit adder / Subtractor is divided into MSP (Most Significant Part) and LSP (Least Significant Part) between the 8th and 9th bits. In which the MSP of the original adder is modified to include the detection logical circuits, data controlling circuits, sign extension circuits, latch and clock circuits, logic for calculating carry-in and carry-out signals.

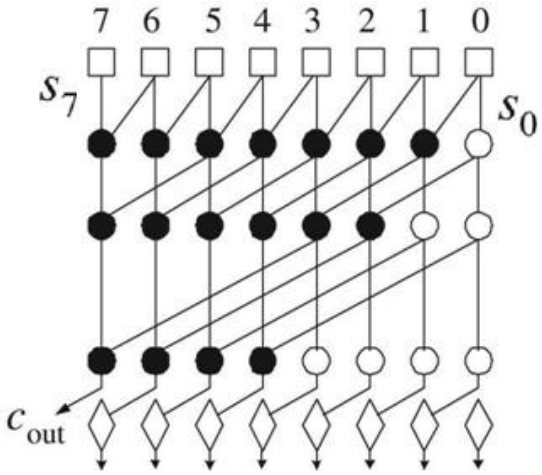

Figure 3: Parallel-prefix structures for integer adders-Kongge-Stone

2.3 Carry Select Adder

We investigate design methods to minimize the power-delay product of 16-bit adders in partially depleted (PD) silicon-on-insulator (SOI) technology. Addition is used as a benchmark here since it is one of the important tasks performed by the CPU, considering that adders are needed in the Arithmetic and Logic Units, for the memory address generation and for floating point calculations. The improvement of the power-delay product will be performed at the different hierarchical

Levels of the design: circuit design style, cell decomposition, and global architecture.

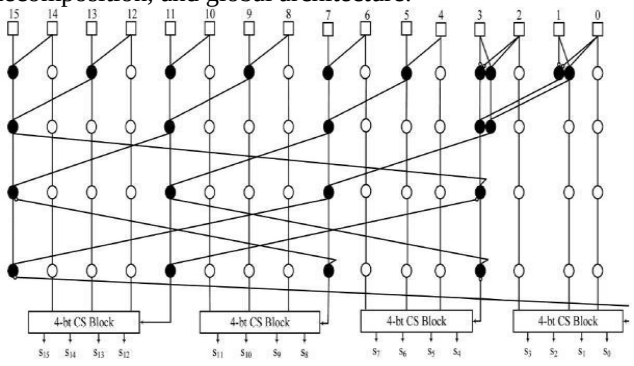

Figure 4: Using a sparse carry computation unit

In this study, we concentrate on static design styles, since the performance advantage of both dynamic logic styles and pass-gate design is expected to decrease in future deep-submicron technologies. The features of lower dynamic power consumption and higher noise margin make static CMOS particularly attractive.

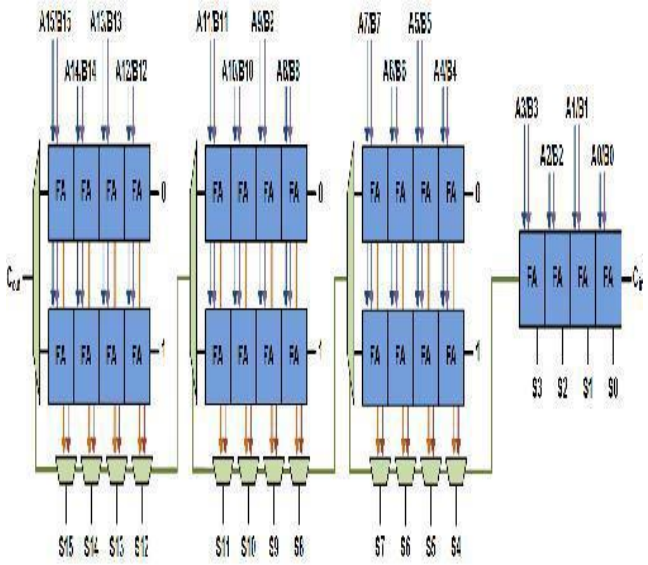

Figure 5: Regular Fixed Size CSLA

A 16-bit carry-select adder with a uniform block size of 4 can be created with three of these blocks and a 4- bit ripple carry adder which is show in fig 5. Since carry-in is known at the beginning of computation, a carry select block is not needed for the first four bits. The delay of this adder will be four full adder delays, plus three MUX delays.

A 16-bit carry-select adder with variable size (Figure 6) can be similarly created. Here we show an adder with block sizes of 2-2-3-4-5. This break-up is ideal when the full-adder delay is equal to the MUX delay, which is unlikely. The total delay is two full adder delays, and four multiplexer delays.

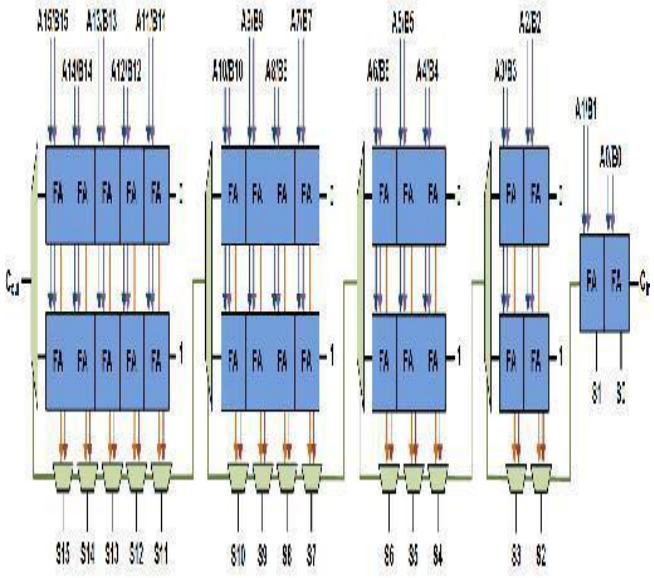

Figure 6: Variable Sized CSLA

Moreover, the activation of the parasitic bipolar transistor in PD SOI is reported to result in fatal erroneous states in dynamic logic and to make circuit design with pass-gates more difficult. The renewed interest in static design styles like pseudo-NMOS and rationed CMOS shows that alternative design styles are investigated in SOI in order to reduce the power dissipation while still maintaining high-speed performance.

2.4 Error Tolerant Adder

A1CSA: An Energy-Efficient Fast Adder Architecture for Cell-Based VLSI Design is Error Tolerant Adder. In modern VLSI technology, the occurrence of all kinds of errors has become inevitable. By adopting an emerging concept in VLSI design and test, Error Tolerance (ET), a novel Error-Tolerant Adder (ETA) is proposed.

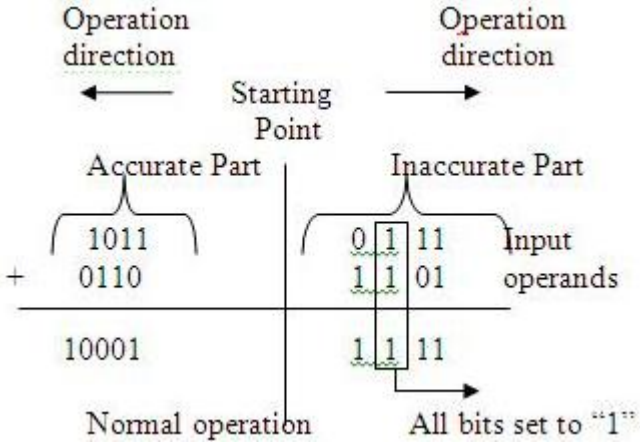

Figure 7: Arithmetic Procedure of Error Tolerant

Adder

The ETA is able to ease the strict restriction on accuracy and at the same time achieve tremendous improvements in both the power consumption and speed performance.

When compared to its conventional counterparts, the proposed ETA is able to attain improvement in the Power-Delay Product (PDP).

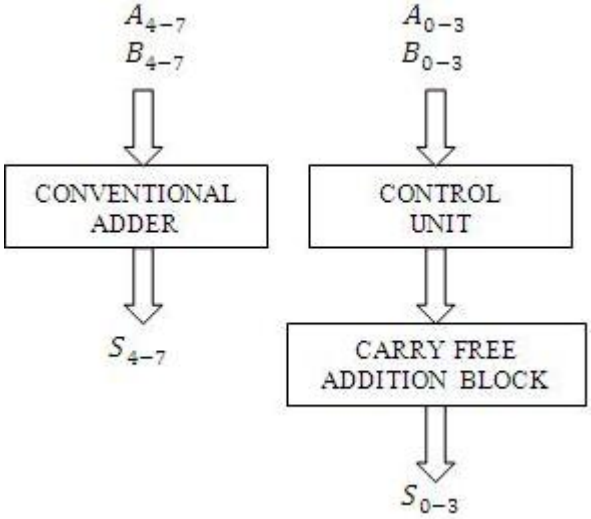

Figure 8: Block Diagram of Error Tolerant Adder

One important potential application of the proposed ETA is in digital signal processing systems that can tolerate certain amount of errors. The delay and power are compared for various adders like RCA and CLA. ETA has high speed and less power compared to its counterparts.

2.5 Hybrid Prefix Adder

Parallel Prefix addition is a technique for improving the speed of binary addition. Due to continuing integrating

intensity and the growing needs of portable devices, high performance and high performance designs are of prime importance.

The classical parallel prefix adder structures presented in the literature over the years optimize for depth of logic, area, fan-out and interconnect count of logic circuits. A new architecture for performing 8-bit, 16-bit and 32-bit Parallel Prefix addition is proposed. The proposed prefix adder structures is compared with several classical adders of same bit width in terms of power consumed, delay calculated and number of computational nodes. The results reveal that the proposed structures have the least power delay product when compared with its peer existing Prefix adder structures.

2.6 MODIFIED AREA EFFICIENT CARRY SELECT ADDER Carry Select adder (CSLA) is an adder which computes n+1

bit sum of two n bit numbers. When compared to earlier Ripple Carry Adder and Carry Look Ahead Adder, Regular CSLA(R-CSLA) is observed to provide optimized results in terms of area. From the architecture of Modified CSLA it is observed that there is a possibility of reducing the area further .Regular CSLA uses dual Ripple Carry Adder to perform addition operation. Modified CSLA (M-CSLA) uses BEC as add one circuit which reduces the area furthermore, such that the total gate count is reduced subsequently.

For 16bit addition, it is proposed to simple gate level modification which significantly reduces the area. It is known as Modified Area Efficient Carry Select Adder (MA-CSLA), shown in figure 9. The strategic work in MA-CSLA reduces the area using the modified XOR gates.

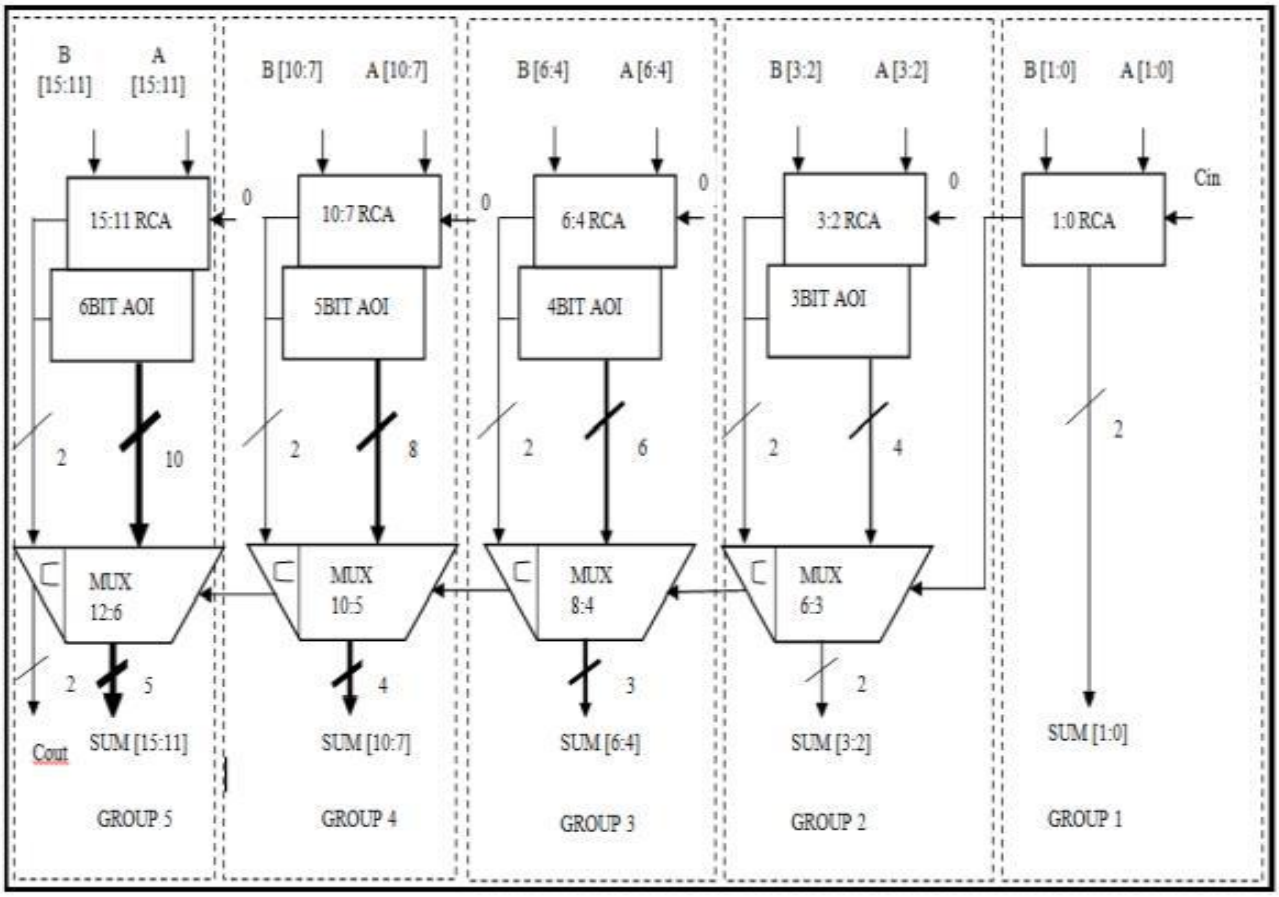

Figure 9: Modified Area Efficient Carry Select Adder (MA-CSLA)

The result analysis shows that the Modified area efficient CSLA is better than the M-CSLA for low power applications like digital signal processing and ALU.

2.7 Parallel Binary Adder

The goal of this paper is to present architectures that provide the flexibility within a regular adder to augment/decrement the sum of two numbers by a constant which is considering in the addition process.

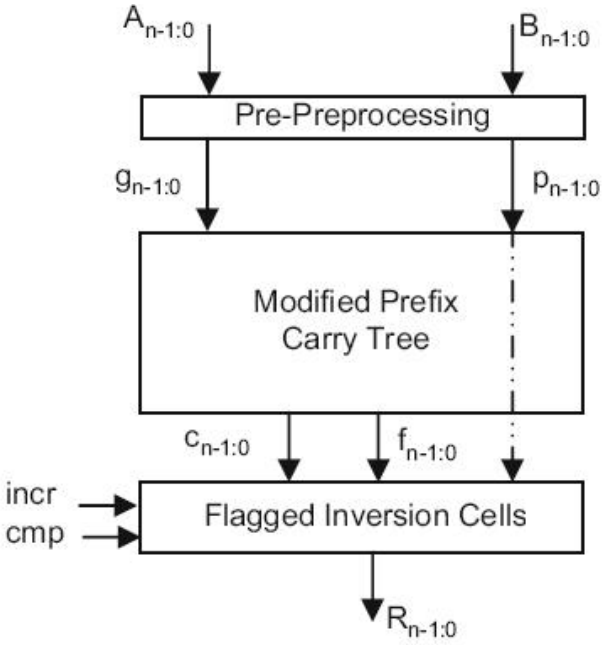

Figure 10: Modified PBA block diagram

This flexibility adds to the functionality of a regular adder, which achieving a comparable performance to conventional designs, therefore eliminating the need of having a dedicated adder unit to perform similar tasks. In this adder if the third operand is a constant, design to accomplish three-input addition. This is accomplished by the introduction of flag bits.

Such designs are called Enhanced Flagged Binary Adders (EFBA), shown in figure 10. It also examines the effect on the performance of the adder when the operand size is expanded from 16 bits to 32 and 64 bits. Detailed analysis has been provided to compare the performance of the new designs with carry-save adders in terms of delay, power dissipated and area consumes.

3. Implementation

Booth multiplication is a technique that allows faster multiplication by grouping the multiplier bits. The grouping of multiplier bits and Radix-8 Booth encoding reduce the number of partial products to half.

The shifting and adding for every column of the multiplier term and multiplying by 1 or 0 is commonly using. But here we take every second column, and multiply by ±1, ±2, or 0.The advantages of this method is halving of the number of partial products. In Booth encoding the multiplier bits are formed in blocks of three, such that each blocks overlap the previous block by only one bit.

Grouping is started from the LSB side, and the first lock only uses two bits of the multiplier term. Figure 3 below shows the grouping of bits from the multiplier term.

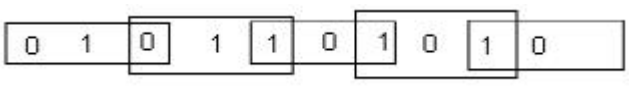

Figure 11: Grouping of bits from the multiplier term in the multiplication operation

Table 1: Operations on the multiplicand

B	Zn	Partial Product
0	0	0
1	1	1×Multiplicand
10	1	1×Multiplicand
11	2	2×Multiplicand
100	-2	-2×Multiplicand
101	-1	-1×Multiplicand
110	-1	-1×Multiplicand
111	0	0

To obtain the correct partial product each block is decoded from the grouped terms. Table 1 shows the encoding of the multiplier value Y, which using the Modified Booth Algorithm. Which generating the following five signed digits, -2, -1, 0, +1, +2. Each encoded digit in the multiplier performs a certain operation on the multiplicand X.

4. Modified Booth

Algorithm for Radix-8

The number of subsequent calculation stages can be decreased by enhancing the parallelism operation. So, one of the solutions of realizing high speed multipliers is to enhance parallelism operation.

The Radix-4 Booth multiplier is the modified version of the conventional version of the Booth algorithm (Radix-2), which has two drawbacks. They are: (i) Which is inconvenient in designing parallel multipliers because the number of add subtract operations and the number of shift operations becomes variable.

(ii) When there are isolated 1's, the algorithm becomes inefficient. These problems are overcome by sing modified Radix-8 Booth multiplier. The Booth algorithm which scans the strings of three bits is given below:

1) If necessary to ensure that n is even, then the sign bit 1 position is extend.
2) A 0 bit is appended to the right of the LSB of the multiplier.
3) Each partial products will be 0, +M,-M, +2M,-2M,-3M,+3M,-4Mor+4M.

Table 2: Radix-8 Recoding

Quartet value	Signed digit value
0	0
1	1
10	1
11	2
100	2
101	3
110	3
111	4
1000	-4
1001	-3
1010	-3
1011	-2
1100	-2
1101	-1
1110	-1
1111	0

Generation of Radix 2 and Radix 8 multiplication (referred to as a hard multiple, since it cannot be obtained via simple shifting and complementing of the multiplicand) generally requires some kind of carry propagate adder to produce. Generated carry propagate adder may increase the latency, mainly due to the long wires that are required for propagating carries from the less significant to more significant bits.

High-speed modulo multipliers using Booth encoding for partial product generation have been proposed in The Booth encoding technique reduces the number of partial products to be generated and accumulated, thereby minimizing the associated hardware. The radix-4 Booth encoding is most prevalent as all modulo-reduced partial products can be generated by mere shifting and negation. Greater savings in area and dynamic power dissipation are feasible for large word- length multipliers by increasing the radix beyond four.

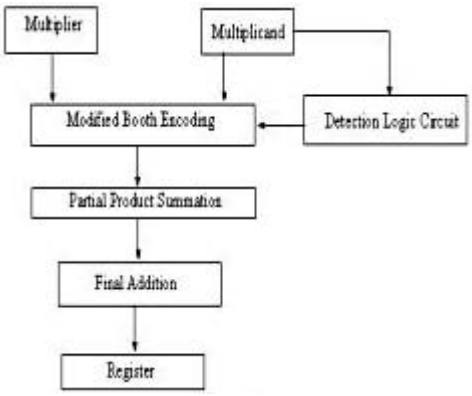

Figure 12: Block Diagram of Radix-8 MBA

In the radix-8 Booth encoding, the number of partial products is reduced by two- thirds. However this reduction in the number of partial products comes at the expense of increased complexity in their generation.

A Digital multiplier is the fundamental component in general purpose microprocessor and in DSP. Compared with many other arithmetic operations multiplication is time consuming and power hungry. Thus enhancing the performance of the circuit and reducing the power dissipation are the most important design challenges for all applications in which multiplier unit dominate the system performance and power dissipation.

The one most effective way to increase the speed of a multiplier is to reduce the number of the partial products. The number of partial products can be reduced with a higher radix Booth encoder, but the number of hard multiples that are costly to generate and which increases simultaneously.

To increase the speed of operation and performance, nowadays many parallel MAC architectures have been proposed. The technique Parallelism in obtaining partial products is the most common technique used in the above implemented architecture.

There are two common approaches that make use of parallelism to enhance the multiplication performance. The difference between the two is that the latest one carries out the accumulation by feeding back the final CSA (Carry Save Adder) output rather than the final adder results which we are obtaining. The rest of the paper is organized as follows. Section second, in which an introduction to the general MAC is given along with basic MAC algorithms. Section third, in which the entire process of parallel MAC based on radix-8 booth encodings is explained. In section four which shows implementation result and the characteristics of parallel MAC based on both of the booth encodings. At last, the conclusion will be given in section five in which provides summary of our proposed approach and discuss scope of future extensions.

5. Results

The simulation results for 16-bit Radix-2 and Radix-8 modified Booth algorithm with seven adders and MAC are trying to implement. Table below shows the synthesis report for array MAC, Radix-2 and Radix-8modified Booth algorithm with seven adders which used here in MAC.

The code is dumped onto the target device Spartan 3E (Xc3s500eft256-4), inputs (Set frequency of asynchronous nets as10MHz), signals (Set frequency of asynchronous nets as10MHz) and outputs (Set capacitive load of outputs as 28000 pf).

Table 3 shows the comparisons of power consumption and delay estimated of the Radix-2Modified Booth Algorithm with seven different adders in MAC. Table 4 shows the power dissipation and delay of Radix-8 using that same adders which used in the Radix-2 MAC. The design summary and simulation result also shown on figure 13 and 14.

6. Conclusion

Here we are compared different adders by its different criteria. They worked well in either power dissipation or in delay. So the performance of each adder is different from the other. The adders avoid the unwanted glitches and thus minimizes the switching power dissipation. Radix -2 modified booth algorithm reduces the number of partial products to half by grouping of bits from the multiplier term in the multiplication operation, which improves the speed.

7. Future Scope

Nowadays we are dealing with the modified booth algorithm which is different from the booth algorithm which we are commonly using now. Radix-2 and Radix-8 Booth Algorithm is commonly using for all multiplication process. Which reduces the number of critical path, there by reduces power consumption. In this paper, 16- bit Radix-8 Modified Booth Algorithm using spurious power suppression technique is designed. The Radix-16 MBA also can be implemented from this designed Radix-8 MBA.

Table 3: Comparison of Radix-2 MBA

Device Parameters	Array multiplier & Accumulator	SPST Adder	Parallel prefix Adder	Carry Select Adder	Error tolerant Adder	Hybrid prefix Adder	Mod. Area Efficient CSL Adder	Parallel Binary Adder
Number of 4 input LUTs	636 out of 29504	1093 out of 29504	1083 out of 29504	657 out of 9312	539 out of 9312	631 out of 9312	735 out of 9312	549 out of 9312
Number of gate count for design	4209	5987	7167	4593	3741	4425	4926	3768
Estimated delay(ns)	217.8	39.69	24.936	54.959	36.041	50.086	57.724	53.084
Power consumption (mw)	154	144	138.8	16.746	16.338	16.631	16.508	16.533

Table 4: Comparison of Radix-8 MBA

Device Parameters	*Array multiplier & Accumulator*	*SPST Adder*	*Parallel prefix Adder*	*Carry Select Adder*	*Error tolerant Adder*	*Hybrid prefix Adder*	*Mod. Area Efficient CSL Adder*	*Parallel Binary Adder*
Number of 4 input LUTs	636 out of 29504	1093 out of 29504	1083 out of 29504	1212 out of 9312	1083 out of 9312	1178 out of 9312	1257 out of 9312	1222 out of 9312
Number of gate count for design	4209	5987	7167	7875	6942	7629	7998	7155
Estimated delay(ns)	217.8	39.69	24.936	59.723	39.15	54.45	60.934	66.106
Power consumption (mw)	154	144	138.8	19.98	22.906	20.019	19.982	19.935

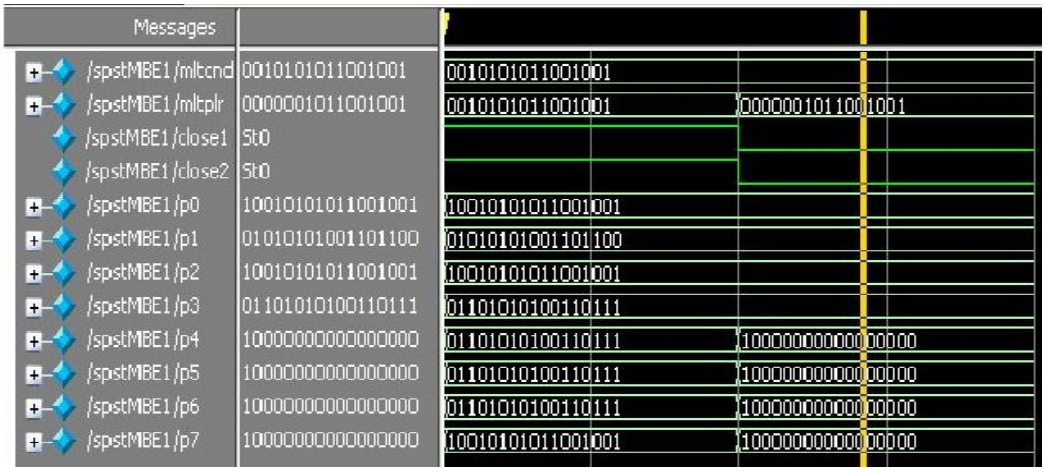

Figure 13: Simulation results for a 16-bit multiplier using radix-8 modified Booth algorithm with Parallel Prefix adder

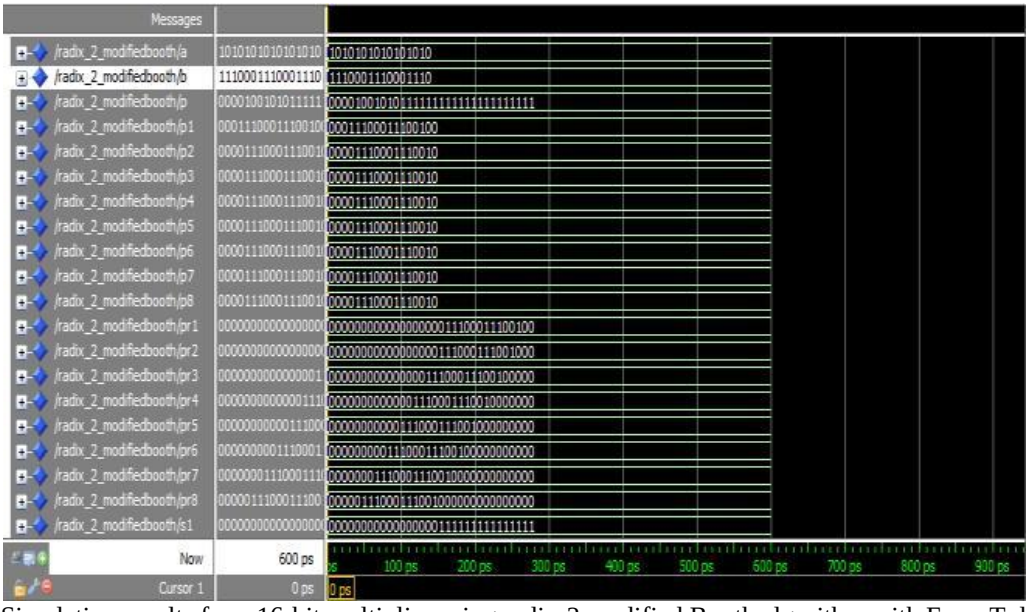

Figure 14: Simulation results for a 16-bit multiplier using radix-2 modified Booth algorithm with Error Tolerant Adder

Device Utilization Summary				
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of 4 input LUTs	1,088	29,504	3%	
Logic Distribution				
Number of occupied Slices	568	14,752	3%	
Number of Slices containing only related logic	568	568	100%	
Number of Slices containing unrelated logic	0	568	0%	
Total Number of 4 input LUTs	1,095	29,504	3%	
Number used as logic	1,088			
Number used as a route-thru	7			
Number of bonded IOBs	64	250	25%	
Total equivalent gate count for design	7,167			
Additional JTAG gate count for IOBs	3,072			

Figure 15: Design Summary of Radix-2 MBA for Parallel Prefix Adder

Device Utilization Summary				
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of 4 input LUTs	1,745	9,312	18%	
Logic Distribution				
Number of occupied Slices	902	4,656	19%	
Number of Slices containing only related logic	902	902	100%	
Number of Slices containing unrelated logic	0	902	0%	
Total Number of 4 input LUTs	1,748	9,312	18%	
Number used as logic	1,745			
Number used as a route-thru	3			
Number of bonded IOBs	83	190	43%	
Total equivalent gate count for design	10,983			
Additional JTAG gate count for IOBs	3,984			

Figure 16: Design Summary of Radix-8 MBA for Parallel Prefix Adder

The benefits of miniaturization are high packing densities, good circuit speed and low power consumption. Binary multiplier is an electronic circuit used in digital electronics such as a computer to multiply two binary numbers, which is built using binary adders. A fixed-width multiplier is attractive to many multimedia and digital signal processing systems which are desirable to maintain a fixed format and allow a minimum accuracy loss to output data.

References

[1] Young-Ho Seo and Dong-Wook Kim, (February 2010) 'A new VLSI architecture of parallel multiplier-accumulator based on radix-2 modified Booth algorithm', in IEEE Trans. on Very Large Scale Integration (VLSI) Systems, vol. 18, no. 2, pp.201-208.

[2] Z.Huang and M. D. Ercegovac, (March 2005), 'High-performance low- power left-to right array multiplier design', IEEE Trans. Comput., vol.54, no.3, pp.272–283.

[3] G. Lakshmi Narayanan a n d B.Venkataramani, (July2005), 'Optimization t e c h n i q u e s for FPGA - based wave pipelined DSP blocks', IEEE Trans. Very Large Scale Integration (VLSI) Systems, vol.13, no.7,pp. 783-792.

[4] H.K. Chen, K.C. Chao, J.I. Guo, J.S. Wang and Y.S. Chu, (2005), 'An efficient spurious power suppression technique (SPST) and its applications on MPEG- 4 AVC/H.264 transform coding design', Proc. IEEE Int. Symps. Low Power Electron. Des., pp.155–160.

[5] H. Lee, (2004) 'A power-aware scalable pipelined Booth multiplier', Proc. IEEE Int. SOC Conf., pp.123–126.

[6] J.Choi, J. Jeon and K.Choi, (2000), 'Power minimization of functional units by partially guarded computation', Proc. IEEE Int. Symp. Low Power Electron. Des., pp.131–136.

[7] J. Fadavi-Ardekani,(June1993), 'M*N Booth encoded multiplier generator using optimized Wallace trees', IEEE Trans. Very Large Scale Integration (VLSI) Systems, vol. 1, no. 2, pp. 120–125.

[8] K.H.Chen, Y.M.Chen, and Y.S.Chu, ' A versatile multimedia functional unit design using the spurious power suppression technique', in Proc. IEEE Asian Solid -State Circuits Conf ., 2006

[9] Addanki Purna Ramesh, Dr. A.V.N. Tilak and Dr. A.M. Prasad June 2012 'Efficient implementation of 16-bit Multiplier-Accumulator using Radix-2 Modified Booth Algorithm and SPST adder using Verilog' International Journal of VLSI design & Communication Systems (VLSICS) Vol.3, No.3

[10] M.Karthikkumar,D.Manoranjaitham,K.Praveen Kumar,"Implementation of efficient 16 Bit MAC Using modified booth algorithm and different addres" Volume 4, Issue 3, March 2014 1 ISSN 2250-3153.