Journal of Theoretical and Applied Information Technology
20th November 2013. Vol. 57 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A LOW POWER MULTIPLIER USING ENCODING AND BYPASSING TECHNIQUE

[1]PRABHU. E, [2]Dr.MANGALAM.H, [3]S.KARTHICK

[1]Assistant Professor, Department of Electronics and Communication Engineering
Amrita School of Engineering, Amrita Vishwa Vidyapeetham University, Coimbatore, India
[2]Professor, Department of Electronics and Communication Engineering
Sri Krishna College of Engineering and Technology, Coimbatore, India
[3]Assistant Professor (Sr.G), Department of Electronics and Communication Engineering,
Bannari Amman Institute of Technology, Sathyamangalam.
E-mail: [1]eprabhu0585@gmail.com

ABSTRACT

In this paper, a low power Encoding and Bypassing technique based shift-add multiplier is presented. The proposed architecture is derived from simple way to reduce power consumption and area of the multiplier in VLSI design architecture level model. The proposed architecture maximum reduces the power consumption and area compared to the other conventional multiplier. The modification to the multiplier includes proposed Encoder design for Modified Radix-4 recording rules, removal of zero partial products using bypassing technique (decoder). A decoder instead of bypass and feeder register is utilized for the removal of zeros (bypassing) and selecting the current partial product value to be stored in register. In this paper, encoder and decoder selector circuit has been used in the proposed model work. Low power consumption and low area occupied multiplier architecture model is proposed. The simulation result for the encoding process and bypassing technique using decoder generated using Xpower analyzer in Xilinx 10.1 ISE (integrated software environment) represents the dynamic power consumption is reduced to almost 50%. When the power consumed by the proposed multiplier using Spartan-2 is 6.28mW, the Virtex-4 device is 6.89mW. The proposed multiplier is mainly applicable for designing low power VLSI circuits and high speed switching techniques.

Keywords: *Modified Radix-4 Recording Rules, Encoding, Bypassing, Decoder, Switching Activity, Partial Product.*

1. INTRODUCTION

In several of the digital circuits, the multiplier is mainly implemented. To use the digital multiplier, various methods [1, 2] can be implemented. The majority of the steps involved calculating the set of partial product and summing the partial product together. The technique for multiplying decimal number is on the basis of computing the partial products, shifting them to the left and then adding them together. The first stage of majority of multipliers involves making the partial products which is an array of AND gates. For partial product generation, an n-bit by n-bit multiplier needs n^2 AND gates. The most difficult part is to get the partial products, as it involves multiplying the long number by one digit. The technique [6, 12] is very slow since it involves several intermediate additions. The multiplier bits are categorized into groups of s bits, called selection groups in s-bit selection. By multiplying the i^{th} selection group Si with the multiplicand N, the i^{th} partial product Pi is received, and shifting it to the same position x as the selection group, these multiplications takes lot of time. The sign with a separate rule generally in the 2's complement representation is the second issue. That compels the multiplication process to be adapted to manage 2's complement numbers, and that perplexes the process a little more. For the multiplication process it takes more time and it consumes high power, delay increased and switching activity also increased [19].

More and more sophisticated signal processing systems are being used on a VLSI chip, as the scale of integration remains developing. These signal processing applications not only need great computation capacity but also eat up substantial amount of energy. Power consumption has become a vital concern in today's VLSI system design, while performance and area stay to be the two key design tolls. From two important forces, the demand for low-power VLSI system wakes up. First with the stable development of operating frequency and processing power capacity of circuit. In portable devices, the low power design straightly leads to prolonged operation time. In

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

most signal processing algorithms, Multiplication is a fundamental operation. Multipliers have large area, long latency and consume substantial power. So, in low power VLSI system design, the low-power multiplier design has been a significant part. There has been widespread work on low-power multipliers at technology, physical, circuit and logic levels. Commonly, the performance of the multiplier decides the system's performance, because the slowest element in the system generally is the multiplier. And also, generally it is the most area consuming. Thus, a major design issue is the optimizing the power and area of the multiplier. Nevertheless, area and power are normally contradicting constraints so that improving power results mainly in low areas.

Two disadvantages were there in the Original version of Booth's multiplier (Radix-2). [1] Many add / subtract operations became changeable and thus became inconvenient while deciding multipliers. [2] When there are separated 1s, the algorithm becomes ineffective. By means of Radix 4 Booth's algorithm these issues are overcome which can scan strings of three bits with the encoding algorithm. In this paper, the design of low power multiplier contains encoding, partial product generators, bypassing and finally an adder. To reduce the power by decreasing the number of partial products, this low power multiplier method is implemented. In this paper, there are only four partial products generated since an 8-bit multiplier is implemented. To produce the product by multiplying the multiplicand A by 0, 1, -1, 2 or -2, the partial products generator is designed. For product generator, multiply by zero intends the multiplicand is multiplied by '0'. Multiply by '1' intends the product still keeps the same as the multiplicand value. Multiply by '-1' intends that the product is the 2's complement form of the number. Multiply by '-2' means to shift left one bit the 2's complement of the multiplicand value and multiply by '2' intends just shift left the multiplicand by one place.

In these multipliers, many techniques [6, 10, and 12] are implemented to diminish the power dissipation. Tree multiplication is implemented for this multiplication process for high speed multiplication process. In microprocessor and digital signal processors, the digital multiplier is an arithmetic unit and implemented for emerging media processors. Before and after every partial product were empty gaps without any bits to be added. It would be tough to create a common summation network for such circuits which works with all possible partial product generators. Multimedia and DSP applications are extremely multiplication intensive so that the performance and power consumption of these systems are ruled by multipliers. To create several partial products, the calculations of the multipliers manipulates two input data for addition operations, which in the CMOS circuit design needs several switching activities. Hence, switching activity within the functional unit needs for most of power consumption. For the intention of power reduction technique implemented as the kernel operator is applicable exact data path of video and audio codec technique the common multipliers can be simplified to a network of shift, adders and subtractor also decrease the switching quality of the following signal. To decrease the number of partial product in the system is one of the best ways to decrease the switching activity in resulting in less power consumption.

To decrease the switching activity of the system, the different kinds of multiplication methods are implemented. To decrease the power consumption of the system, the [15] ultra low power multiplier is implemented. It implements positive feedback charge distributing logic and 30% of power and the switching activity of the multiplier has been decreased. A power efficient 16 times 16 Configurable Booth Multiplier was presented in [14] that supported single 16-bit, single 8-bit, or twin parallel 8-bit multiplication operations. To increase the probability of partial product becoming zero, the Booth encoding is implemented, which contributes in reduction of the redundant switching activities and the total power consumed by the configurable Booth multiplier. The recoding method [12] which was introduced in Booth decoder, increased the number of zeros in multiplicand. Such that, the number of switching activity were minimized resulting in reduced power consumption. The Massey-Omura multiplier of GF(2^m) [22] exploited a normal basis with its bit parallel version being more often than not executed applying 'm' indistinguishable combinational logic blocks whose inputs were cyclically shifted from each other. But, the parallel Massey-Omura multiplier had redundancy which was overcome by modified modular multiplier architecture of lower circuit complexity.

In this paper, a low power multiplier using proposed Radix-4 encoding technique is presented for Modified Booth algorithm and also bypassing the partial products. The recording rules employed in [11] for realization of the Booth multiplier is

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

utilized in our encoding approach. The advantage of employing an encoding technique is to reduce the partial products and wherever it introduces zeros, the bypassing has been performed. If the zero will be bypassed then the number of partial product is reduced, and also the switching activity has been minimized. The encoding technique contributes largely in partial product generation. If the partial product generated is zero then it is bypassed. Otherwise, the decoder is enabled and the non-zero partial product selected is stored in the partial product register which is processed through add register. Also MBRA4 algorithm is used to reduce the multiplicand value by executing proposed encoding architecture. For this encoding method the multiplier value has been reduced. Adder has been used in this project in order to accelerate multiplication by compressing the number of partial products. There are four sign extension values generated namely sign 1E, 2E, 3E and 4E for the partial product PP1, PP2, PP3 and PP4 respectively. The arrangement of total four partial products is shown in the Figure.5. The second partial product is to be shifted left by two bits before adding to the first partial product. Hence the third has been shifted left by four whereas for fourth it will be shifted left by six. In this paper, the architecture of Modified Booth algorithm for Radix-4 multiplier [11] is presented with even more minimized switching activities which cuts down the power consumption and area required.

The paper is organized as follows: Review of related works in Section (2), Problem definition/research methodology in Section (3), proposed model in Section (4), Result and Discussion in Section (5), Conclusion in Section (6).

2. REVIEW OF RELATED WORKS

Literature presents several architectures for low power multiplier using shift-add operation and Booth multiplication. Here, some of the works presented based on Booth multipliers are reviewed. Rizwan Mudassir *et al.* [8] have presented a low power multiplication algorithm for reducing the switching activity through operand decomposition for Radix-8 Booth multiplier. The algorithm incorporated Redundant Binary Signed Digit (RBSD) Modified Booth-3 (Radix-8) encoding scheme to generate RBSD partial product rows and low power RB adder unit designed for accumulation and thereby circumventing the need to generate hard multiples and sign extension. In the experiments, result had shown that reduction of 21% in dynamic power consumption and at least 44% reduction in Energy Delay Product (EDP) with a penalty of 4% in area. Yuan-Ho Chen *et al.* [9] have presented a closed form of compensation function for fixed-width Booth multipliers utilized Generalized Probabilistic Estimation Bias (GPEB). Based on the probabilistic estimation from the truncation part, the GPEB circuit could be easily built according to the systematic steps. An 8 GPEB Booth multiplier improved more than 88% on the reduction of absolute average error compared with the traditional Direct Truncation (D-T) multiplier, and more than 32% area savings were obtained in the GPEB Booth multiplier compared with Post Truncation (P-T) Booth multiplier.

Yajuan and Chip-Hong Chang, [10] have presented the use of redundant binary (RB) arithmetic in the design of high-speed digital multipliers that was beneficial due to its high modularity and carry-free addition. The redundant binary (RB) arithmetic led to lower encoding and decoding complexity than the recently proposed RB Booth encoder. Synthesis results using Artisan TSMC 0.18-um standard-cell library showed that the RB multipliers designed with Booth encoding algorithm exhibit on average 14% higher speed and 17% less energy-delay product than the existing multiplication algorithms for a gamut of power-of-two word lengths. Nishat Bano [11] have designed the conventional Radix-2 Booth multiplier and modified Radix-4 Booth multiplier using VHDL. The delay and power dissipation of modified Radix-4 Booth multiplier was less compared to the conventional Radix-2 Booth multiplier. When implemented on FPGA, the modified Radix-4 Booth multiplier consumed 22.9% less power than the conventional Radix-2 multiplier. Also, estimated delay was less for Radix-4 multiplier.

A.S.Prabhu and V.Elakya [12] have proposed Booth multipliers for reducing the power of the multiplier circuit. The multiplier circuit was designed with conventional full adder. The power results were thus compared for the different inputs. The result shown that average power consumed by the multiplier. Booth multiplier consumes comparatively less power and hence multiplier with Booth recoding unit was designed for low power consumption. Devi, V Vidya *et al.* [13] have proposed a probabilistic estimation bias (PEB) circuit for a fixed-width 2's complement Booth multiplier. The PEB circuit was derived from theoretical computation, instead of exhaustive simulations and heuristic compensation strategies that tend to introduce curve-fitting errors and

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

exponential-grown simulation time. Consequently, the PEB circuit provided a smaller area and a lower truncation error compared with existing works. Implemented in an 8 × 8 2-D discrete cosine transform (DCT) core, the DCT core utilizing the PEB Booth multiplier improved the peak signal- to-noise ratio by 17 dB with only a 2% area penalty compared with the direct-truncated method.

Shiann-RongKuang [14] have presented a power-efficient configurable Booth multiplier (CBM) that supports single 16-b, single 8-b, or twin parallel 8-b multiplication operations. To efficiently reduce power consumption, a dynamic-range detector was developed to dynamically detect the effective dynamic ranges of two input operands. The results shown that, the multiplier was more complex than non-CBMs, but significant power and energy savings were achieved. Jiaoyan Chen *et al* [15] have presented a 16-by-16-bit Radix-4 Booth multiplier that was built based on Positive Feedback Charge Sharing Logic (PFCSL) and implemented in 45nm technology. They achieved around 30% reduction in dynamic power and 60% in static power respectively compared to the same design being implemented using static dual-rail logic. Also, the area of the multiplier was significantly smaller.

3. PROBLEM DEFINITION/RESEARCH METHODOLOGY

In [16] BZ- FAD (By pass zero, feed A directly) method and add and shift multiplier method was used. Where; the multiplier multiplies A by B, the removal of the shifting the B registers, and directs feeding of A to the adder, bypassing the zero values, using a ring counter as replacement for a binary counter and partial product shifts were removed. Encoder bus selector was used in each and every cycle of multiplication process and to use a ring counter to select the n^{th} cycle of multiplication process. The same ring counter is also used as the multiplexer. When Multiplexer multiplication value has been zero, it reduces the switching activity of adder .i.e. adder has been bypassed. The partial product was shifted to right one bit and the switching activity should be reduced for this process. So the power consumption of multiplier was reduced. BZ-FAD [16] saves power for two reasons: first, the lower half of the partial product is not shifted, and second, this half is implemented with latches instead of flip-flops.

This paper presents a low power multiplier using encoding and bypassing techniques. Modified Radix-4 Booth multiplier is a high speed multiplier that introduces parallelism which helps to reduce the number of calculation stages. The original version of the Booth multiplier (Radix-2) had two drawbacks. They are:

(i) The number of add subtract operations and the number of shift operations present.
(ii) The algorithm becomes inefficient. These problems are overcome by using modified Radix-4 Booth multiplier.

The negative values of B are made by taking the 2's complement and in this paper Carry-look-ahead (CLA) fast adders are used. The multiplication of M is done by shifting M by one bit to the left. In any case, n-bit parallel multiplier, only n/2 partial products are produced.

In [11] the modified Booth Radix-4 multiplier is designed. The Power dissipation of modified Radix-4 Booth multiplier is less as compared to conventional Radix-2 multiplier. Radix-4 Booth multiplier consumes less power than conventional Radix-2 multiplier. Also estimated delay is less for Radix-4 multiplier. In [16] it is multiplication process i.e. taken into each and every bit, and then it passes through zero bypass technique. So power consumption is increased. In [11] modified Radix-4 Booth multiplier, eliminated the limitations of conventional Radix-2 Booth multiplier but, un-bypassed partial products were limitations of its own. Hence, motivated from the multiplier architectures presented in [11] and [16], a multiplier by utilizing Modified Booth algorithm Radix-4 and bypassing technique is proposed in this paper. As far as bypassing is concerned, a decoder circuit is employed here, which eliminates partial products with zero occurrence. It is used for bypassing zeros and selecting the non-zero partial products.

4. PROPOSED MODEL

The primary intention of presenting this work is to design and develop a technique for low power multiplier. The proposed design utilizes the Modified Booth algorithm for Radix-4 (MBAR4) and the modification is done in MBAR4 with the perspective of reducing power. The research idea has come from the techniques presented recently in [11] and [16], of which one technique is focused on MBAR4 [11] and other one is focused on low power multiplier based on bypass technique of shift and add architecture [16]. The encoded bits obtained by Booth encoder are used for

Journal of Theoretical and Applied Information Technology
20th November 2013. Vol. 57 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

multiplication. Here, addition needed completely depend upon the number of encoded bits. In order to reduce the number of addition process required, bypass technique has been used this reduces the switching activities.

The proposed multiplier architecture lowers the total switching activity so that power consumption is reduced compared with the conventional architecture. The proposed architecture will be implemented using XILINX. The whole program is written in Verilog and power consumption is measured using Xpower analyzer.

4.1.Grouping Process

The multiplier is 8 bit ones and zeros format and it is grouped into four groups. Each group is comparing 3 bits Figure.1. In this encoding process it uses full subtractor and EX-OR gates to produce e_0, e_1 and e_3 which are the three encoded outputs. Note that before grouping the encoding bits one common bit is added i.e. b_{-1}, it produces value i.e. always zero. A full subtractor and the EX-OR gate is used to produce the final encoded output i.e. 8x8 combination of input taken into the multiplication process. Taking the combination process into 4 groups and each group comparing 3 bits is named as E_1, E_2, E_3, and E_4.

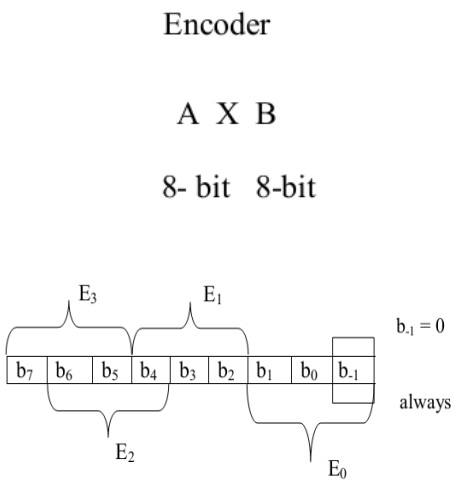

Figure.1. Encoder 3 Bit Grouping Process

4.2. Encoding Process

In encoding process let us first compare 8 x 8 multiplication processes. The encoder input is $b_{n+i,}$ b_i and $bi_{-1,}$ as represented in Figure.2. are encoded at its output as e_2, e_1, and e_0. Encoded output bit e_2 is used to assign the sign value for the multiplication i.e. +\-. The encoder input for b_{i+1}, b_i and b_{i-1} is executed by the full subtractor and it gives the encoded outputs. Using the Modified Radix-4 Recoding rule tabulated in Figure.3, the proposed encoder architecture is designed. Encoding output is taken from e_0, e_1, e_2.

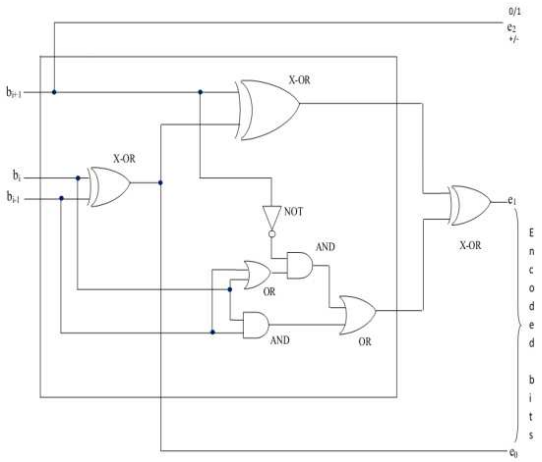

Figure.2. Proposed Encoder Architecture

b_{i+1}	b_i	b_{i-1}	E	Multiplicand	
0	0	0	0	0	BYPASS
0	0	1	+1	+M	PPA
0	1	0	+1	+M	PPA
0	1	1	+2	+2M	PP Shift 1
1	0	0	-2	-2M	PP 2'S shift 1
1	0	1	-1	-M	PP2'S A
1	1	0	-1	-M	PP 2' S A
1	1	1	0	0	By pass

Figure.3. Modified Radix-4 Recoding rule [11]

In this use of encoding process, the number of multiplication process should be reduced, taken the difference and borrowed output for the method.

$$difference = [(b_i \ XOR \ b_{i-1}) \ XOR \ b_{i+1}]$$

$$borrow = [\overline{b_{i+1}} \ AND (b_i \ OR b_{i-1})] OR (b_i \ AND b_{i-1})$$

$$e_2 = signbit = b_{i+1}$$

$$e_1 = (difference \ XOR \ borrow)$$

$$e_0 = (b_i \ XOR \ b_{i-1})$$

4.3. Proposed Multiplier Operation

In the proposed multiplier (Figure.4.) the 2:4 decoder is employed to select the partial product registers and the counter signals c_0, c_1 is used ensure the required number of addition operations.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Basically, the c_0, c_1 is used as the selector lines of 4:1 multiplexer (MUX) of the partial product registers. Also, the 2:1 multiplexer is used to select the positive and negative operation performed over register 'A' (multiplicand). Then a 1-bit shift register is included to perform multiplication of the positive or negative (2's complement) multiplicand by decimal 2. Again there is one 4:1 MUX is employed to select between the partial product values generated after performing encoded multiplication for which the e_0 and e_1 encoder output acts as select lines. Other than this, a one input bubbled AND gate is used to enable the shift operation. While the XOR gate output is used as the enable for 4:1 multiplexer (MUX) of the partial product registers and for 2:4 decoder. The four partial product registers and 16- bit Result register are represented in Figure.5.

First consider the generated encoding output of e_0, e_1, e_2 is 0, 0, 0 i.e., the encoding output of e_2 value is 0. So e_2 as a selector line selects the 'A' register value using 2:1 MUX to is stored in 'S' register. Similarly, if e_0, e_1, e_2 is 0, 0, 1, i.e., the encoding output of e_2 value will be 1. So e_2 as a selector line selects the 2's complement of 'A' register value using 2:1 MUX to is stored in 'S' register. But then, it is observed that both the e_1 and e_0 outputs are 0. It is considerable at the select lines, which select the partial product register but EX-OR gate produces the logical high signal output. Thus at decoder at active low enable pin, input is high at present and so partial product register is disabled and won't store any data. Also, at the same time the 4:1 MUX is disabled too, thereby leaving the particular partial product in its additional operation to be performed i.e. it bypasses the partial product containing zeros by saving an addition operation, ultimately minimizing the switching activities and resulting in reduced power consumption.

Next, if the output of encoder (e_0, e_1, e_2 is 0, 1, 0,) i.e., the e_2 value is 0 then, e_2 as a selector line selects the 'A' register value using 2:1 MUX to is stored in 'S' register. As the outputs of e_1 and e0 is 1 and 0 respectively i.e. the encoded value is +2 (e_2=0, e_1=1 and e_0=0) then the inverted e_0 input of the AND gate enables the 1-bit shift register which shifts the value in 'S' register by 1-bit. The select lines and enable signals for partial product and decoder are now enabled since the operation is not processing any zeros in partial products. The produced encoding e_0 and e_1 value is given through EX-OR gate enabled output i.e. one in addition of NOT gate to the output i.e. zero. So partial product has been selected and stored in the register and the value is given through the adder. The counter signal c_0, c_1 are then used to select the partial product registers output.

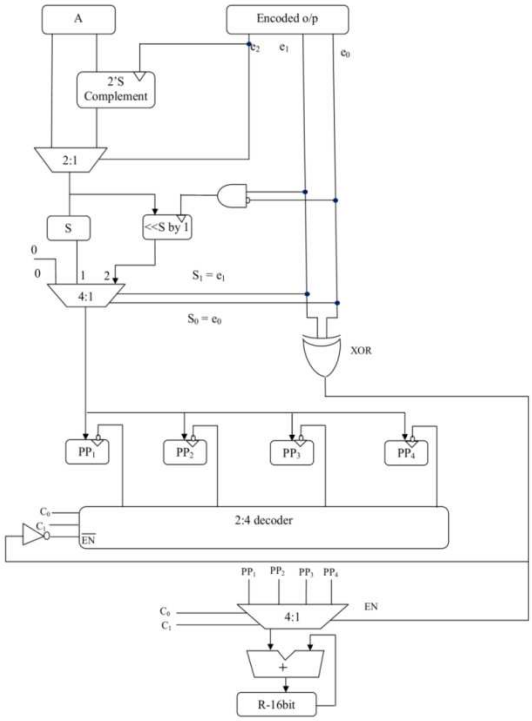

Figure.4. Proposed Multiplier Architecture

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

.

Similarly, if the output of encoder (e_0, e_1, e_2 is 0, 1, 1,) is -2, then sign bit is high (e_2 value is 1). So e_2 as a selector line selects the 2's complement of 'A' register value using 2:1 MUX to is stored in 'S' register. Now as it is known that the outputs of e_1 and e0 is 1 and 0 respectively then the one input bubbled AND gate enables the 1-bit shift register, shifting the value in 'S' register by 1-bit. The select lines and enable signals for partial product and decoder are now enabled since the operation is not processing any zeros in partial products. Therefore, the partial product register value is processed through the adder. The counter signal c_0, c_1 are then updated to select the partial product registers output.

Finally, if the output of encoder (e_0, e_1, e_2 is 1, 0, 0,) i.e., the e_2 value is 0 then, e_2 as a selector line selects the 'A' register value using 2:1 MUX to is stored in 'S' register. Similarly, if the negative encoded value (e_0, e_1, e_2 is 1, 0, 1,) is obtained i.e., the encoder output is -1 (e_2 value is 1) then, for e_2 as a selector line selects the 2's complement of 'A' register value using 2:1 MUX to is stored in 'S' register. Further, if the output of e_1=0 and e_0=1 then the shift operation is disabled as the one input bubbled AND gate generates logic low at active high enable signal of the 1-bit shift register. Then being the selector line of 4:1 MUX e_1=0 and e_0=1 selects the 'S' register value into the partial product. Since the encoder values (e_1=0 and e_0=1) are uneven then the XOR gate output which produces enable signal for the 4:1 MUX of partial product registers and the decoder, will enable the 4:1 MUX with active high signal and at the same time the decoder is enabled with active low by inverting the same XOR gate output which in-turn enables the partial product registers. Thus, the partial product register are added by enabling the adder using counter signals c_0, c_1. Thus, from the above procedure it is clearly understandable that whenever there are zeros encoded by encoder from the 8-bits of multiplier, the bypassing of partial products comes into the picture and inturn the switching activities are minimized which reduces the power consumed by the proposed multiplier architecture.

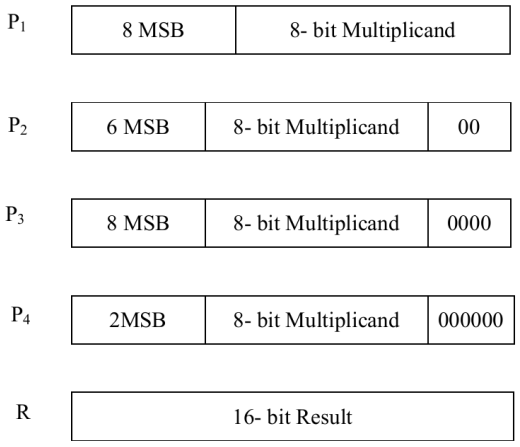

Figure.5. Structure Of Partial Product Stored Register Model

5. RESULT AND DISCUSSION

In this section, experimental result for proposed low power multiplier using encoding and bypassing technique is presented. XILINX 10.1 ISE was used for the synthesis and simulation of the proposed multiplier. The results for the system under test are generated and the obtained dynamic power is reduced compared to the existing Booth Multiplier architectures.

Device Utilization Summary				[-]
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Flip Flops	59	12,288	1%	
Number of 4 input LUTs	100	12,288	1%	
Logic Distribution				
Number of occupied Slices	70	6,144	1%	
Number of Slices containing only related logic	70	70	100%	
Number of Slices containing unrelated logic	0	70	0%	
Total Number of 4 input LUTs	100	12,288	1%	
Number of bonded IOBs	36	240	15%	
Number of BUFG/BUFGCTRLs	1	32	3%	
Number used as BUFGs	1			

Figure.6. Device Utilization Table For Virtex-4

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

.

Device Utilization Summary				[-]
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Flip Flops	58	384	15%	
Number of 4 input LUTs	128	384	33%	
Logic Distribution				
Number of occupied Slices	85	192	44%	
Number of Slices containing only related logic	85	85	100%	
Number of Slices containing unrelated logic	0	85	0%	
Total Number of 4 input LUTs	128	384	33%	
Number of bonded IOBs	35	86	40%	
Number of GCLKs	1	4	25%	
Number of GCLKIOBs	1	4	25%	

Figure.7. Device Utilization Table For Spartan-2

In Figure.6. and Figure.7. the Device utilization for the proposed multiplier architecture is represented for Virtex-4 and Spartan-2 family respectively. The results of both the device types are compared with the Radix-2 and Radix-4 Booth multipliers presented in [11]. The comparison will be done in terms No. of slice FFs utilized, No. of LUTs required, No. of Bonded I/Os, delay and Dynamic power consumed. These parameters values can be obtained from the synthesis report, except for Dynamic power which can be obtained from the XPower analyzer. The proposed multiplier requires almost 20% less no. of slice FFs for both Virtex-4 and Spartan-2 devices, and no. of LUTs required is also minimized to 10 to 20 %. While, the Delay required is approximately 30% less for Spartan-2 family and 72 % less for Virtex-4. The switching activity of the proposed multiplier is reduced compared to other existing multipliers. In this proposed method maximum power reduction has been achieved.

Table.1. Comparison Table Of Multipliers

Multiplier Types / Parameters	Radix-2 Booth Multiplier [11]	Radix-4 Booth Multiplier [11]	Proposed Radix-4 Booth Multiplier	Proposed Radix-4 Booth Multiplier
Device and Family	Spartan-2	Spartan-2	Spartan-2	Virtex-4
No. of slice FFs	77	72	58	59
No. of LUTs	140	129	128	100
No. of Bonded I/O	32	32	35	36
Delay(ns)	27.110	26.103	18.817	6.025
Dynamic Power (mW)	15	11	6.28	6.89

Table.2. Comparison Table Of The Proposed Multiplier At Different Frequencies And Devices

Proposed Multiplier @ different Frequencies		25MHz	50MHz	75MHz	100MHz
Device and Family Dynamic power (mW)	Spartan-2	15.510	31.020	46.530	62.040
	Spartan-3	0.190	0.370	0.560	0.750
	Virtex-2	1.000	2.000	3.000	4.000
	Virtex-4	7.250	7.860	8.470	9.070
Device and Family Total power (mW)	Spartan-2	24.510	40.020	55.530	71.040
	Spartan-3	24.190	24.370	24.560	24.750
	Virtex-2	25.000	26.000	27.000	28.000
	Virtex-4	260.250	260.860	261.470	262.070

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

To reduce the power of the proposed architecture, Encoder architecture for the Modified Radix-4 recording rules is implemented. Using this encoding process the switching activity is reduced highly using bypassing technique. The decoder circuit is used to bypassing zeros and it is mainly used to reduce the power consumption of the circuit. Therefore, the proposed design is synthesized in Xilinx ISE tool and using XPower Analyzer the Dynamic power consumed by the designed multiplier is calculated with respect to the device and family. Dynamic power consumption of our proposed system is reduced approximately by 50% compared to the Radix-2 and Radix-4 Booth Multiplier. Also, the proposed multiplier is implemented on different devices and their families (Spartan and Virtex families). The proposed design implemented on these devices viz; Spartan-2, Spartan-3, Vitex-2 and Virtex-4 shows that power consumption increases with increase in frequency on particular device while the power consumption for the proposed design varies with different devices. This comparison results are tabulated in Table.2.

6. CONCLUSION

In this paper a low power multiplier using encoding and bypassing technique method was proposed. The modifications to the conventional architecture included the Modified Booth algorithm for Radix-4, bypassing method, reducing the switching activities use of encoding and bypassing zeros technique. The results were shown for dynamic power and delay reduction compared to other low power multiplier design. For this proposed low power multiplier was successfully designed and synthesized using XILINX ISE. Output power for proposed design was 50% minimized and delay was minimized up to 30% compared to the existing low power Booth's multipliers.

REFERENCES

[1] Paul J.M. Havinga, Gerard J.M. Smit, "Design techniques for low power systems", *Journal of Systems Architecture*, Vol. 46, no. 1, pp. 1-21, January 2000.

[2] A. Chandrakasan and R. Brodersen, "Low-power CMOS digital design," *IEEE Journal Solid-State Circuits*, Vol. 27, No. 4, pp. 473 – 484, 1992.

[3] Wu A., "High performance adder cell for low power pipelined multiplier", *In proceedings of IEEE International Symposium on Circuits and Systems*, Vol. 4, pp. 57–6, 1996.

[4] S. F. Anderson and J. G. Earle, "The IBM system 360 Model 91: Floating-Point Execution Unit", *IBM Journal*, Vol. 11, No. 1, PP. 34-53, 1967.

[5] K. Hwang, "Computer Arithmetic: Principles, Architecture and Design", *John Wiley and Sons*, 1979.

[6] Nan-Ying Shen and Oscal T.-C. Chen, "Low-power multipliers by minimizing switching activities of partial products," *in Proc. IEEE International Symposium on Circuits and Systems*, vol. 4, pp. 93 – 96, 2002.

[7] U. Ko, P.T. Balsara, W. Lee, "Low-Power Design Techniques for High Performance CMOS Adders," *IEEE Transactions on VLSI Systems*, Vol. 3, No. 2, pp. 327-333, 1995.

[8] RizwanMudassir, MohabAnis and JavidJaffari, "Switching Activity Reduction in Low Power Booth Multiplier", *in proceedings of IEEE International Symposium on Circuits and Systems*, 2008.

[9] Yuan-Ho Chen, Chung-Yi Li, and Tsin-Yuan Chang, "Area-Effective and Power-Efficient Fixed-Width Booth Multipliers Using Generalized Probabilistic Estimation Bias", *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol.1, no. 3, pp. 277 - 288, 2011.

[10]Yajuan He, and Chip-Hong Chang, "A New Redundant Binary Booth Encoding for Fast 2n-Bit Multiplier Design", *IEEE Transactions On Circuits And Systems—I: Regular Papers*, Vol. 56, No. 6, June 2009.

[11]NishatBano, "VLSI Design of Low Power Booth Multiplier", *International Journal of Scientific & Engineering Research*, Vol. 3, no. 2, February 2012.

[12]A.S.PrabhuV.Elakya, "Design of Modified Low Power Booth Multiplier", *in proceedings of 2012 International Conference on Computing, Communication and Applications (ICCCA)*, pp. 1- 6, 2012.

[13]Dr. V.VidyaDevi ,GuruKumar.Lokku and A.Natarajan, "Fixed width Booth multiplier based on peb circuit," *International Journal of Artificial Intelligence & Applications (IJAIA)*, Vol.3, No.2, March 2012.

[14]Shiann-RongKuang, "Design of Power-Efficient ConFigurable Booth Multiplier," *IEEE*

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Transactions on circuits and systems, vol. 57, no. 3, march 2010.

[15]Jiaoyan Chen, Emanuel Popovici ,DilipVasudevan and Michel Schellekens, "Ultra Low Power Booth Multiplier Using Asynchronous Logic," *IEEE 18th International Symposium on Asynchronous Circuits and Systems*, 2012.

[16]M. Mottaghi-Dastjerdi, A. Afzali-Kusha, and M. Pedram, "BZ-FAD: A Low-Power Low-Area Multiplier based on Shift-and-Add Architecture", I*EEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 17, no. 2, pp. 302-306, February 2009 .

[17]O.T. Chen, S. Wang, and Yi-Wen Wu, "Minimization of switching activities of partial products for designing low-power multipliers," *IEEE Transactions on VLSI Systems*, vol. 11, pp. 418 – 433, June 2003.

[18]J. S. Wang, C. N. Kuo, and T. H. Yang, "Low-power fixed-width array multipliers," *in Proc. IEEE Symp. Low Power Electron.Des*.,2004, pp. 307–312.

[19]O. Chen, S.Wang, and Y.W. Wu, "Minimization of switching activities of partial products for designing low-power multipliers," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst*., vol. 11, no. 3, pp. 418–433, Jun. 2003.

[20]Z. Huang and M. D. Ercegovac, "High-performance low-power left-toright array multiplier design," *IEEE Trans. Comput*., vol. 54, no. 3, pp. 272–283, Mar. 2005.

[21]H. Lee, "A power-aware scalable pipelined Booth multiplier," *in Proc. IEEE Int. SOC Conf*., 2004, pp. 123–126.

[22]Reyhani-Masoleh, Arash, and M Anwarul Hasan. "A new construction of Massey-Omura parallel multiplier over GF (2m)." *Computers, IEEE Transactions on* 51.5 (2002): 511-520.