ISSN:2348-2079

Volume-8 Issue-4

International Journal of Intellectual Advancements and Research in Engineering Computations

To deploy partial-carry save radix-8 booth multipliers in dathpaths using combined arithmetic logic level synthesis solution

Indujaa.R.S[1], Dr.Sharmilee.K[2]

[1]Student, Department of Electronics and Communication Engineering, Nandha Engineering College, Erode.

[2]Professor, Department of Electronics and Communication Engineering, Nandha Engineering College, Erode.

ABSTRACT

This paper presents plan of making use of multiplier of booth of radix 8, the delay in it is deduced. The quantity of fragmentary products was deduced in order to formulate a multiplier of an enhanced speed. Now there are minimal operands are there for addition & so a methodology which has high throughput can be enacted in the summing of fragmentary products. To formulate a multiplier an efficient algorithm for the complementary procedural of 2's is played in figures which are signed as MBE can be deployed for both unsigned & signed figures. In this document a research on absorption of area & delay is described. LUT are the main components that determine delay & by minimizing them delay will also be deduced.

Index Terms: Multiplier, Booth, Radix-8, Partial carry save.

INTRODUCTION

Integrated circuits have made the experimental possibility which shows that semiconductor devices could perform vice verse of the vacuum tubes, and in mid-20th-century advanced technology in semiconductor device fabrication is established. The usage of large numbers of tiny transistors into a small chip cause huge improvement over the manual association of circuits using discrete electronic components. For the rapid adoption of standardized ICs in place of design using discrete transistors the integrated circuits made mass production capability, reliability, and building -block approach to circuit design. In which cost and performance are the advantages for the ICs over discrete circuits. There is no need for construction of transistor at a time because chip with their all components are printed as unit by using photolithography so that the cost is low. Since the component can switch quickly and consume low power because the size of the component is small and also close together which covers the area range from few square mm to around 250 mm2 which improves the performance.

ADVANCES IN INTEGRATED CIRCUITS

To control everything from computers to cellular phones to digital microwave ovens using most advanced integrated circuits are the microprocessor. An another family of integrated circuit called digital memory chip which is mainly required for the modern information society. Meanwhile the cost is quite high for designing and developing a complex ICs, when spread across typically millions of production units the individual IC cost is minimized. The small size which allows short traces in turn allows a low power logic such as cmos to be used at fast switching speed which improves the ICs performance high. Over the year size of the ICs

Author for correspondence:
Department of Electronics and Communication Engineering, Nandha Engineering College, Erode

consistently migrated to smaller feature size which allows more circuitry packed on each chips. That makes improvement in cost per unit and power consumption go down with speed goes up. However, by introducing the high-k dielectrics ICs with nanometer-scale device with leakage current problem will be solved or at least ameliorated .to use finer geometrics among the manufacturer there is competition since speed and power consumption gain are apparent to the end user. The international technology roadmap for semiconductor, or ITRS described the expected progress for few years. IC design productivity depends on the efficiency with which the design may be converted from concept to a physical layout. A good design strategy with a good design system will be provided for consistent descriptions in various abstraction levels. The role of good design strategies is to reduce complexity, increase productivity, and assure working product. Design is a continuous trade-off to achieve adequate results for:

- Performance which includes speed, power, function, flexibility
- Die size (hence die cost)
- Time taken for designing
- Ease for generation of test and testability.

LITERATURE REVIEW

The simplest multiplication operation is to obtained the product of two numbers by hand. There is three steps in this procedure: partial product generation, partial product reduction and the final addition. For further specification let us calculate the product for 2 twos complement number, for example, 11012 (−310) and 01012 (510), when computing the product by hand, which can be described. The first operand is the multiplicand and the second the multiplier. The intermediate products are called partial products and the final result is called the product. However, the multiplication process, when this method is mapped directly to hardware.

Radix-4 Booth Multiplier

Radix-4 booth multiplier is possible to reduce the number of partial products by half, by using the technique of radix 4 booth recoding. To obtain the same results we can take every second column, and multiply by ±1, ±2 , or 0. Radix 4 booth encoder performs the process of encoding the multiplicand based on multiplier bits. It will compare 3 bits at a time with overlapping technique. Grouping starts from the LSB, and the first block uses two bits of the multiplier and assumes a zero for the third bit. The functional operation of Radix-4 booth encoder is shown in the Table1.1. It consists of eight different types of states and during these states we can obtain the outcomes, which are multiplication of multiplicand with 0,-1 and -2 consecutively.

Table 1: Booth recoding table for radix-4

Multiplier Bits Block			**Recoded 1-bit pair 2 bit booth**			
i+1	**i**	**i-1**	**i+1**	**I**	**Multiplier Value**	**Partial Product**
0	0	0	0	0	0	Mx0
0	0	1	0	1	1	Mx1
0	1	0	1	-1	1	Mx1
0	1	0	1	0	2	Mx2
1	0	0	-1	0	-2	Mx-2
1	0	1	-1	1	-1	Mx-1
1	1	0	0	-1	-1	Mx-1
1	1	0	0	0	0	Mx0

Radix-4 Booth algorithm is given below:

(1) To ensure that n is even extend the sign bit 1

position if necessary.
(2) To the right of the LSB of the multiplier append a 0.
(3) Based on the value of each vector, each Partial Product will be 0, +y, –y, +2y or –2y. The negative values of y are made by taking the 2's complement. Carry look ahead (CLA) fast adders are used for addition and the negative values of y are made by taking the 2s complement in this paper. By shifting one bit to the left multiplication of y is done. Thus in designing n-bit parallel multiplier, only n/2 partial products are generated some of the advantages of this method is the halving of the number of partial products which is important in circuit design as it relates to the propagation delay in the running of the circuit, and the complexity and power consumption of its implementation.

Booth Multiplier Using Radix-4

By enhancing parallelism which decrease the number of subsequent calculation stages that gives one of the solution for realizing high speed multipliers. The original version of the booth algorithm (radix-2) with two drawbacks. That are:
(i) By designing parallel multipliers the number of add subtract operations and the number of shift operations becomes variable and becomes inconvenient.
(ii) When there are isolated 1's the algorithm becomes inefficient. By using modified radix 4. booth algorithm which scan strings of the three bits is given below:
1) To ensure that n is even extend the sign bit 1 position if necessary.
2) To the right of the LSB of the multiplier append a 0.
(3) Based on the value of each vector, each Partial Product will be 0, +y, –y, +2y or –2y. The negative values of y are made by taking the 2's complement. Carry look ahead (CLA) fast adders are used for addition and the negative values of y are made by taking the 2s complement in this paper. By shifting one bit to the left multiplication of y is done. Thus in designing n-bit parallel multiplier, only n/2 partial products are generated some of the advantages of this method is the halving of the number of partial products which is important in circuit design as it relates to the propagation delay in the running of the circuit, and the complexity and power consumption of its implemention. According to the following rule the partial products are calculated
$Zn = -2 \times Bn+1 + Bn + Bn-1$ Where, B is the multiplier.

PROPOSED RADIX-8 BOOTH MULTIPLIER

Making use of multiplier of booth of radix 8 & 4, the delay in it is deduced. The quantity of fragmentary products was deduced in order to formulate a multiplier of an enhanced speed. Now there are minimal operands are there for addition & so a methodology which has high throughput can be enacted in the summing of fragmentary products. To formulate a multiplier an efficient algorithm for the complementary procedural of 2's is played which are signed as MBE can be deployed for both unsigned & signed figures. In this document a research on absorption of area & delay is described. LUT are the main components that determine delay & by minimizing them delay will also be deduced.
Also speed & performance are proportional to each other in an IC. A multiplexer of 4 radix of booth is deployed for an enhanced performance.

A navigator of project of Xilinx is deployed for the formulation of code & design of VHDL. To produce the form of wave a modelsim 5.4 is assigned for this task. All the attributes of kit of FPGA XC3S5E is accounted.

The absorption of power is the main characteristic that determines the performance of an IC & for this a software is formulated that will compute the absorption of power.

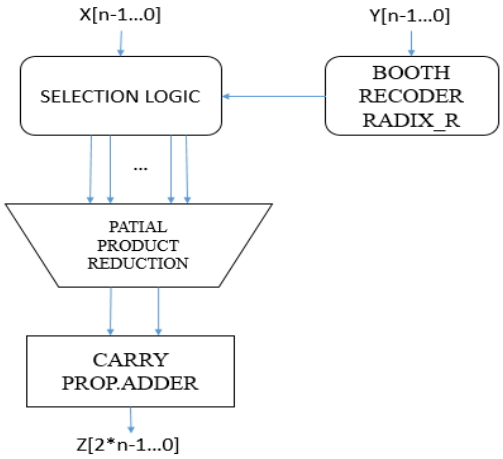

Figure 1: Block diagram for radix-R multiplier

Radix-8 carry select booth encoder

A radix 8 Booth recoder processes the multiplier operand Y in groups of 4-bits in parallel, overlapping of most significant bit of every group with the less significant of the following one, and produces the selection bits and the sign bits. The sign bit is used to compose the negative partial products. This bit is '0' when the partial product has a positive weight and '1' otherwise. The original recoder works directly with the multiplier Y However, in order to deal with partial carry-save multipliers we apply the encoding to V, with Y = V + B. Using k-bit fragments, the real carry-in signals to every fragment are unknown till they have been computed by the Y Real Carry Calculation unit. Two recoders working in parallel are required then: one considering a '0' carry-in (B8_0) and another considering a '1' carry-in (B8_1). Hence, our recoder shall be named Radix 8 Carry Select Booth Encoder (B8CSBE).

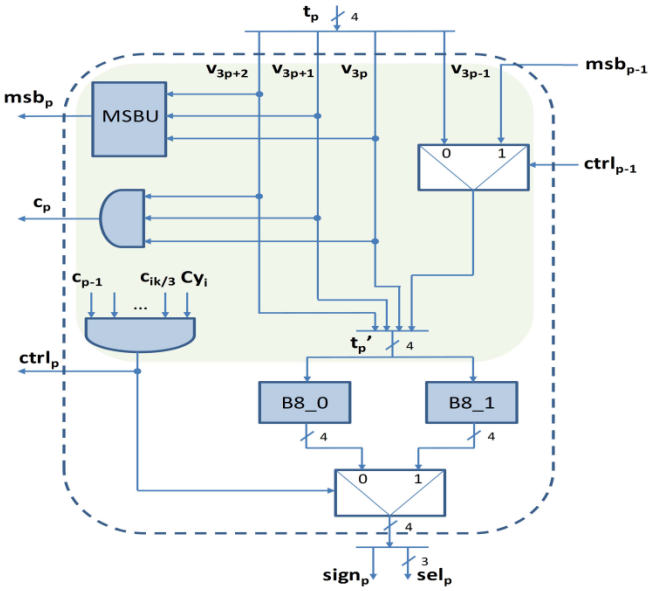

Figure 2: Radix 8 carry select booth encoder (B8CSBE) cell

Figure 2 depicts the structure of the generic cell responsible for generating the B8CSBE recoding for a given tuple tp = v3_p+2v3_p+1v3_pv3_p−1. In addition to both recoders, some extra units are necessary to properly complete the encoding process. As indicated in, first is necessary to

generate the actual Booth tuple (t_ p), as this may present a different least significant bit (lsb) because the B8_1 recoding of the prior cell may generate a different most significant bit (msb). The calculation of the actual most significant bit (msbp) is handled by the Most Significant Bit Unit (MSBU). Moreover, there can also be internal carries (cp) within every fragment due to the B8_1 recoding.

Algorithm 1 Reducer(n,k,U,A)
Input: 2U, U, 2A,A
Output: UJ,AJ
UJ<---------U, AJ <----------A
i<---------1
while i < n/k do
3to2Counter(i*k, UJ, AJ)
3to2Counter(i*k+,1, UJ, AJ)
j <---------2
while j < k do
2to2Counter(i*k+ j, UJ, AJ)
j<-------- j+1
end while
i ← i + 1
end while
2to2Counter(n + 1, UJ, AJ)

Algorithm 2 How to Deploy Partial Carry Save Datapaths
Input: G, RS, R=2r, k G is a DFG, RS is the resource set
Output: Partial carry-save datapath
H ← hard Multiples(R)
M SU ← ms Units(H, k) D Multispeculative units for H
RS ← RS ∪ MSU
D High Level synthesis part
for each product p ∈ G do
for each hard multiple h ∈ H do
Insert Hard Multiple Node(h, p, G)
end for
end for
D Arithmetic part
for each multiplier m ∈ RS do
CSE L(m.x, k) D CSEL technique for the multiplicand
for each hard multiple h ∈ H do
CSE L(m.h, k) D m.h is a hard multiple in partial carry-save
end for
CSB E(m.y, k) D CSBE recoding for the multiplier
end for
D High Level synthesis again

SIMULATION RESULTS

Modelsim software is a hardware simulation and debug environment which is primarily targeted at smaller ASIC and FPGA design. That combines simulation performance and capacity with the code coverage and debugging capabilities required to simulate multiple blocks and systems and attain ASIC gate-level sign-off. By the support of Verilog system, Verilog for design, VHDL, and System C that provides a solid foundation for single and multi-language design verification. Modelsim software is very easy to use and unified debug and for simulation environment provide FPGA designers both the advanced capabilities that they are growing to need and the environment with work productive. It is also known as verification and simulation tool for Verilog System, VHDL, Verilog and mixed language designs.

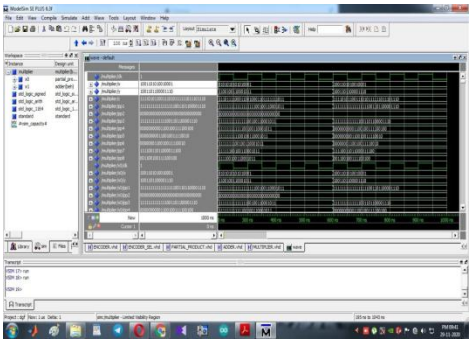

Figure 3: Simulation waveform of radix-8 booth multiplier

Xilinx ISE (Integrated Software Environment) is a software tool which is used for synthesis and analysis of HDL designs, which enables the developer to compile their designs, performs timing analysis, examine RTL diagrams, simulate a designs reaction to different stimuli, and configure the target device with the programmer. To overcome the power challenges associated with shrinking technologies Xilinx ISE constantly innovates new that provides the HDL and schematic editors, logic synthesizer, fitter, and bit stream generator software and XS tools from XESS provide utilities for downloading the bit stream into the FPGA on the XSA board. Xilinx understands, Xilinx power tool helps us to perform power estimation and analysis for given design and FPGA power consumption is one of the large concern of FPGA users. For increase in FPGAs local capacity and performance power estimation and analysis become more important.

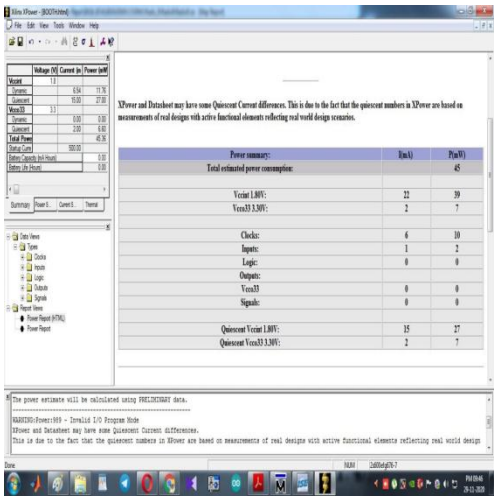

Figure 4: Power Analysis of Radix-8 booth Multiplier

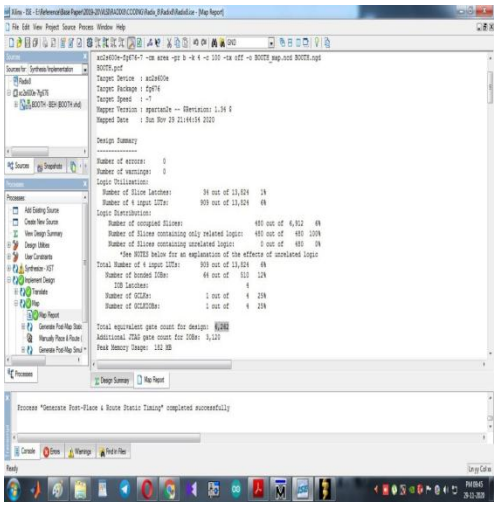

Figure 5: Area Analysis of Radix-8 booth Multiplier

FPGAs total power is the sum of two components:

Static power - When the transistor is logically "OFF" the leakage current either will be from source to drain or through the gate oxide and exists even then the static power results primarily from transistor leakage current in the device.

Dynamic power - Dynamic power is assembled with design activities and switching events to the core of the device or I/O ports of the device.

Dynamic power is obtained by node capacitance, supply voltage, and switching frequency.
The two primary components that shows the accuracy of the Xilinx power tools:

- Device data models and device characterization integrated into the tool.
- The user enters the accurate inputs into the tools. When using the Xilinx Power Tools, XPE is used for pre-design power estimation and XPA is used for post-implementation design power optimization. Since the Xilinx Power Tools include different stages of the design flow, the tools also can be used for
- Selection of part
- Designing of board
- System reliability
- Estimation of power consumption for a specific design

The Power Tools can be used for power optimization as well. You can identify which parts in the design are responsible for the most power consumption. You can then find tradeoffs to design for power. This can also be coupled with power optimization algorithms available in synthesis and implementation within ISE.

CONCLUSION

In this project a partial carry-save radix 8 Booth-based multiplier has been presented. The starting point of this multiplier is a Dataflow Graph where the 3X hard multiple has been decoupled and inserted as an independent operation. Then, the radix 8 Booth recorder has been modified to efficiently encode the partial carry save multiplier operand.

Table 2: Compression of Power Analysis

CONSTRAINTS	EXISTING METHOD	PROPOSED METHOD
Number of slices latches	53	34
Total estimated power consumption	95mW	43mW
Delay	0.723ns	0.683ns
PDP	0.0672pWs	0.0369pWs

Our experiments have studied the tradeoffs obtained with several fragment sizes. From the synthesis results we can conclude that the On-The Fly Correcting Radix 8 Booth Multi speculative Multipliers outperform their prior version with radix 4 [26] as they are more efficient in terms of energy comparison with radix 4 and radix 8 implementations.

REFERENCES

[1]. Alberto A. Del Barrio, Member, IEEE, roman Hermida, Senior Member, IEEE, and Seda Ogrenci-Memik, Senior Member, IEEE, 2019.
[2]. I. Hatai, I. Chakrabarti, and S. Banerjee, “A computationally efficient reconfigurable constant multiplication architecture based on CSD decoded Vertical–Horizontal common sub-expression elimination algorithm,” IEEE Trans. Circuits Syst. I, Reg. Papers, 65(1), 2018, 130–140.
[3]. G. D. Licciardo, C. Cappetta, L. Di Benedetto, and M. Vigliar, “Weighted partitioning for fast multiplierless multiple-constant convolution circuit,” IEEE Trans. Circuits Syst. II, Exp. Briefs, 64(1), 2017, 66–70.
[4]. A. A. Del Barrio, R. Hermida, S. O. Memik, “A partial carry-save on-the fly correction multispeculative multiplier,” IEEE Trans. Comput., 65(11), 3251–3264.
[5]. H. Jiang, J. Han, F. Qiao, and F. Lombardi, “Approximate radix-8 booth multipliers for low-power and high-performance operation,” IEEE Trans. Comput., 65(8), 2016, 2638–2644.
[6]. K. Tsoumanis, S. Xydis, C. Efstathiou, N. Moschopoulos, and K. Pekmestzi, “An optimized modified Booth recoder for efficient design of the add-multiply operator,” IEEE Trans. Circuits Syst. I, Reg. Papers, 61(4), 1133–1143.
[7]. A. A. Del Barrio, N. Bagherzadeh, and R. Hermida, “Ultra-low-power adder stage design for exascale floating

point units," ACM Trans. Embedded Comput. Syst., 13(3), 2014, 105-1–105-24.

[8]. S. Belloeil-Dupuis, R. Chotin-Avot, and H. Mehrez, "Exploring redundant arithmetics in computer-aided design of arithmetic datapaths," Integr., VLSI J., 46, 104–118.

[9]. F. de Dinechin and B. Pasca, "Designing custom arithmetic data paths with FloPoCo," IEEE Design Test Comput., 28(4), 2011, 18–27.

[10]. G. Quan, J. P. Davis, S. Devarkal, and D. A. Buell, "High-level synthesis for large bit-width multipliers on FPGAs: A case study," in Proc. 3rd IEEE/ACM/IFIP Int. Conf. Hardw./Softw. Codesign Syst. Synth. (CODES+ISSS), 2005, 213–218.

[11]. S. Gupta, A. Nicolau, N. D. Dutt, and R. K. Gupta, SPARK: A Parallelizing Approach to the High-Level Synthesis of Digital Circuits. Norwell, MA, USA: Kluwer, 2004.

[12]. S. Gupta, A. Nicolau, N. D. Dutt, and R. K. Gupta, SPARK: A Parallelizing Approach to the High-Level Synthesis of Digital Circuits. Norwell, MA, USA: Kluwer, 2004.