

Design of Braun Multiplier with Kogge Stone Adder & It's Implementation on FPGA

Ms. Madhu Thakur ,Prof. Javed Ashraf

ABSTRACT

Multiplication is the basic building block for several DSP processors, Image processing and many other. Over the years the computational complexities of algorithms used in Digital Signal Processors (DSPs) have gradually increased. This requires a parallel array multiplier to achieve high execution speed or to meet the performance demands. A typical implementation of such an array multiplier is Braun design. Braun multiplier is a type of parallel array multiplier. The architecture of Braun multiplier mainly consists of some Carry Save Adders, array of AND gates and one Ripple Carry Adder.

In this research work, a new design of Braun Multiplier is proposed and this proposed design of multiplier uses a very fast parallel prefix adder (Kogge Stone Adder) in place of Ripple Carry Adder. The architecture of standard Braun Multiplier is modified in this work for reducing the delay due to Ripple Carry Adder and performing faster multiplication of two binary numbers.

This research also presents a comparative study of FPGA implementation on Spartan2 and Spartatan2E for new multiplier design and standard braun multiplier. The RTL design of proposed new Braun Multiplier and standard braun multiplier is done using Verilog HDL. The simulation is performed using ModelSim. The Xilinx ISE design tool is used for FPGA implementation. Comparative result shows the modified design is effective when compared in terms of delay with the standard design.

Key words: array multiplier, carry save adder (CSA), Kogge stone Adder, parallel prefix adder ripple carry adder, sparatan2, sparatan2E.

1 INTRODUCTION

Multiplication – an important fundamental function in arithmetic operation. It is an essential operation in Digital Signal Processing (DSP) applications such as FFT, Filtering etc., and usually contributes significantly to time delay. Both the multiplication and the DSP play a vital role in the implementation of VLSI system [12].

Multiplication – Repeated addition of n – bits will give the solution for the multiplication. i.e. Multi –operand addition process. The multi – operand addition process needs two n – bit operands. It can be realized in n- cycles of shifting and adding. This can be performed by using parallel or serial methods. If the multiplicand is given be $A = a_{n-1} \dots a_1 a_0$ and multiplier $B = b_{n-1} \dots b_1 b_0$, then product $P = P_{2n} P_{2n-1} P_{2n-2} \dots P_1 P_0$ [13] is given by:

$$P = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} a_i b_j 2^{(i+j)} \dots \quad (1)$$

Braun Multiplier in standard form has (n-1) **Carry Save Adder** stages for generating partial products and one **Ripple Carry Adder** [6] stage which give final 4 MSB Product Bits.

1.1 Carry Save Adder

Carry save adder is a digital adder which is used to compute sum of three or more n-bit binary numbers. Carry save adder is same as a full adder. It generates two outputs of equal dimensions as the inputs.

This unique dual output consists of:

- One sequence of partial sum bits
- One sequence of carry bits

One of the major speed enhancement techniques used in modern digital circuits is the ability to add numbers with minimal carry propagation.

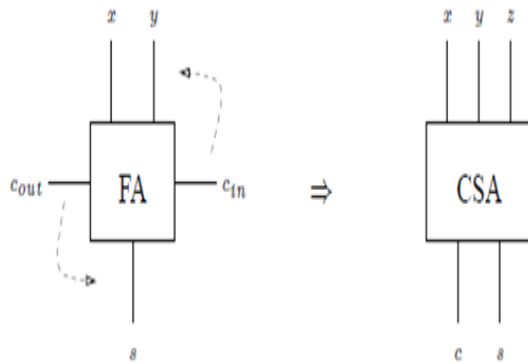


Fig1: The carry save adder block is the same circuit as the full adder.

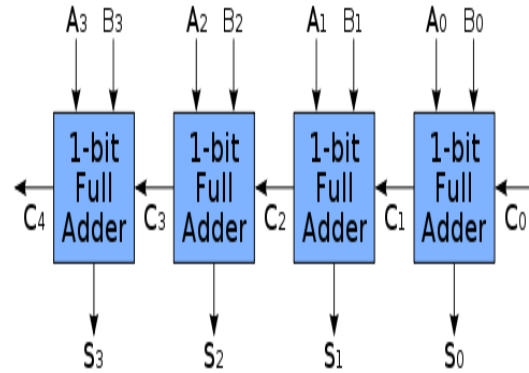


Fig2: 4-Bit Ripple Carry Adder

The basic idea is that three numbers can be reduced to 2, in a 3:2 compressor, by doing the addition while keeping the carries and the sum separate. This means that all of the columns can be added in parallel without relying on the result of the previous column, creating a two output "adder" with a time delay that is independent of the size of its inputs. The sum and carry can then be recombined in a normal addition to form the correct result. It is only the final recombination of the final carry and sum that requires a carry propagating addition.

1.2 Ripple Carry Adder

A simple **ripple carry adder** is a digital circuit that produces the arithmetic sum of two binary numbers. It can be constructed with full adders connected in cascade, with the carry output from each full adder connected to the carry input of the next full adder in the chain. Figure.2 shows the interconnection of four full adder (FA) circuits to provide a 4-bit ripple carry adder.

The main drawback of this adder is that the total propagation delay, T is directly proportional to the total number of stages of Ripple Carry Adder. If the total no. of stages are N and propagation delay of each stage is t_p , then total propagation delay of ripple carry adder will be $T = N \times t_p$.

1.3 FPGA

A **Field-programmable Gate Array (FPGA)** is an integrated circuit designed to be configured by the customer or designer after manufacturing. Field Programmable means that the FPGA's function is defined by a user's program rather than by the manufacturer of the device. A typical integrated circuit performs a particular function defined at the time of manufacture. In contrast, the FPGA's function is defined by a program written by someone other than the device manufacturer. Depending on the particular device, the program is either 'burned' in permanently or semi-permanently as part of a board assembly process, or is loaded from an external memory each time the device is powered up. This user programmability gives the user access to complex integrated designs without the high engineering costs associated with application specific integrated circuits (ASIC).

2 RESEARCH METHODOLOGY

FPGA implementation of 4 bit Braun multiplier using Ripple Carry Adder is selected [9] on which our current research work is based. The architecture of Braun Multiplier shown in figure 3 is modified by replacing the Ripple Carry Adder with a Kogge Stone Adder. The reason for selecting the Kogge Stone Adder is that it is the fastest adder but at the cost of increased area.

2.1 Braun Multiplier

It is a simple parallel multiplier generally called as carry save array multiplier. It has been restricted to perform signed bits. The structure consists of array of AND gates and adders arranged in the iterative manner and no need of

logic registers. This can be called as non – additive multipliers.

2.2 Architecture

An $n \times n$ bit Braun multiplier [10],[11] is constructed with $n(n-1)$ adders, n^2 AND gates and $(n-1)$ rows of Carry Save Adder as shown in the fig.1, where

- X:** 4 bit Multiplicand
- Y:** 4 bit Multiplier
- P:** 8 bit Product of X & Y
- P_n :** $X_i Y_i$ is a Product bit

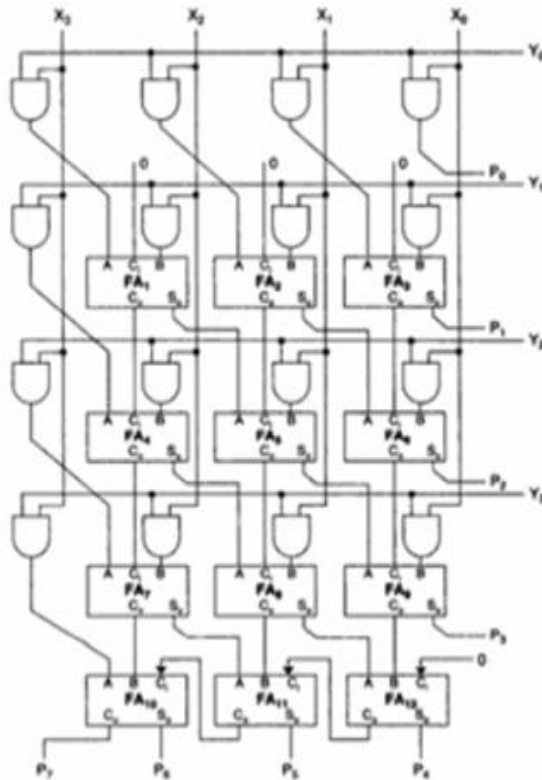


Fig3: 4x4 Braun Multiplier Architecture

Each products can be generated in parallel with the AND gates. Each partial product can be added with the sum of partial product which has previously produced by using the row of adders. The carry out will be shifted one bit to the left or right and then it will be added to the sum which is generated by the first adder and the newly generated partial product. The shifting would carry out with the help of Carry Save Adder (CSA) [4] and the Ripple carry adder should be used for the final stage of the output. Braun multiplier [13] performs well for the unsigned operands that are less than 16 bits in terms of

speed, power and area. But it is simple structure when compared to the other multipliers.

The main drawback of this multiplier is that the potential susceptibility of glitching problem due to the Ripple Carry Adder in the last stage. The delay depends on the delay of the Full Adder and also a final adder in the last stage.

Delay due to the final ripple adder can be minimized by using very fast one of a Parallel Prefix Adder [1] “KOGGE STONE ADDER” which is a type of Carry Look Head Adder.

3 PROPOSED WORK

3.1 Design of Braun Multiplier with Kogge Stone Adder

The objective of this research is to design a new architecture of Braun Multiplier with Kogge Stone Adder which gives fast multiplication results at the cost of increased area.

3.2 Architecture

The proposed multiplier’s block diagram is shown in fig.4 where we have used a 3 bit Kogge Stone Adder in 4th stage of Braun multiplier [9]. This proposed multiplier is implemented using Verilog HDL.

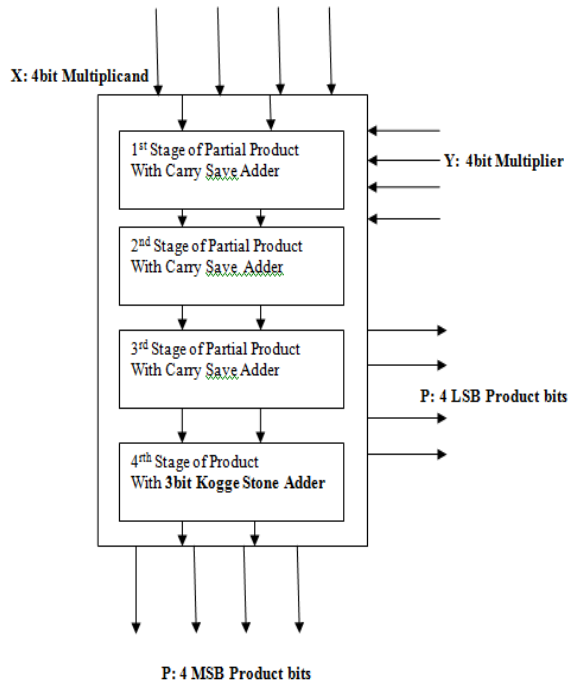


Fig4: Block diagram of Braun Multiplier with Kogge Stone Adder

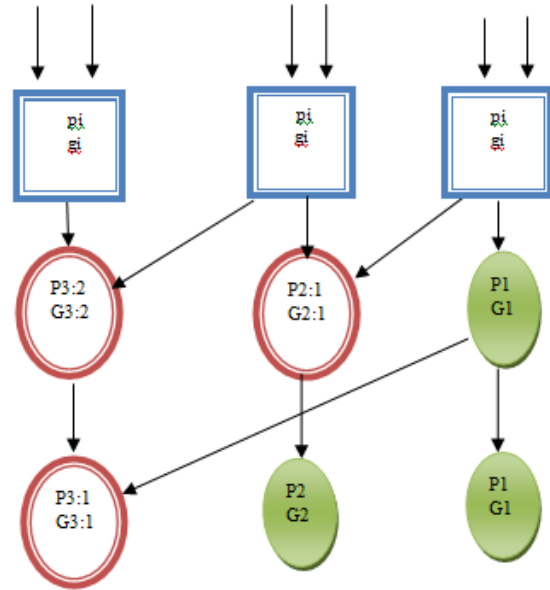


Fig5: Tree Structure of 3bit Kogge Stone Adder

3.3 KSA (Kogge Stone Adder)

KSA is a parallel prefix form carry look ahead adder. It generates carry in $O(\log n)$ time and is widely considered as the fastest adder and is widely used in the industry for high performance arithmetic circuits.

In KSA [5], carries are computed fast by computing them in parallel at the cost of increased area. A tree structure of 3 bit Kogge stone adder (KSA) [5] is shown fig 5. It has three processing stages for calculating the sum bits.

3.3.1 Complete functionality of KSA

I. Pre processing:

This step involves computation of generate and propagate signals corresponding to each pair of bits in A and B. These signals are given by the logic equations below:

$$p_i = A_i \text{ xor } B_i \quad \dots \dots \dots (2)$$

$$g_i = A_i \text{ and } B_i \quad \dots \dots \dots (3)$$

II. Carry look ahead network:

This block differentiates KSA from other adders and is the main force behind its high performance. This step involves computation of carries corresponding to each bit. It uses group propagate and generate as intermediate signals which are given by the logic equations below:

$$P_i:j = P_i:k+1 \text{ and } P_k:j \quad \text{and} \quad \dots \dots \dots (4)$$

$$G_i:j = G_i:k+1 \text{ or } (P_i:k+1 \text{ and } G_k:j) \quad \dots \dots \dots (5)$$

III. Post processing:

This is the final step and is common to all adders of this family (carry look ahead). It involves computation of sum bits. Sum bits are computed by the logic [2] given below:

$$S_i = p_i \text{ xor } C_{i-1} \quad \dots \dots \dots (6)$$

3.3.2 Illustration of KSA using an example:

A = 110 and B = 101

I Step: $p_i = A_i \text{ xor } B_i$ and $g_i = A_i \text{ and } B_i$

$$p_3 = 1 \text{ xor } 1 = 0; p_2 = 1 \text{ xor } 0 = 1; p_1 = 0 \text{ xor } 1 = 1;$$

$$g_3 = 1 \text{ and } 1 = 1; g_2 = 1 \text{ and } 0 = 0; g_1 = 0 \text{ and } 1 = 0;$$

II Step: $P_{i:j} = P_{i:k+1}$ and $P_{k:j}$ and $G_{i:j} = G_{i:k+1}$ or $(P_{i:k+1}$ and $G_{k:j})$

$P_{3:2} = p_3$ and p_2 ; 0 and 1 = 0;
 $G_{3:2} = g_3$ or $(p_3$ and $g_2)$; 1 or $(0$ and $0) = 1$;

$P_{2:1} = p_2$ and p_1 ; 1 and 1 = 1;
 $G_{2:1} = g_2$ or $(p_2$ and $g_1)$; 0 or $(1$ and $0) = 0$;

$P_{3:1} = P_{3:2}$ and p_1 ; 0 and 1 = 0;
 $G_{3:1} = G_{3:2}$ or $(P_{3:2}$ and $g_1)$; 1 or $(0$ and $0) = 1$;

III Step: $S_i = p_i$ xor C_{i-1}

$S_1 = p_1$ xor 0; 1 xor 0 = 1;
 $S_2 = p_2$ xor g_1 ; 1 xor 0 = 1;
 $S_3 = p_3$ xor G_2 ; 0 xor 0 = 0;
 $C_{out} = G_{3:1}$; 1;

Sum Bits : Cout S3 S2 S1 : 1011;

4 RESULTS

The simulation results for the design were observed on ModelSim. The figure no.6 shows the simulation of proposed design (Braun Multiplier implemented using Kogge Stone Adder). The figure no. 7 is showing the simulation result of Braun Multiplier which uses Ripple carry Adder.

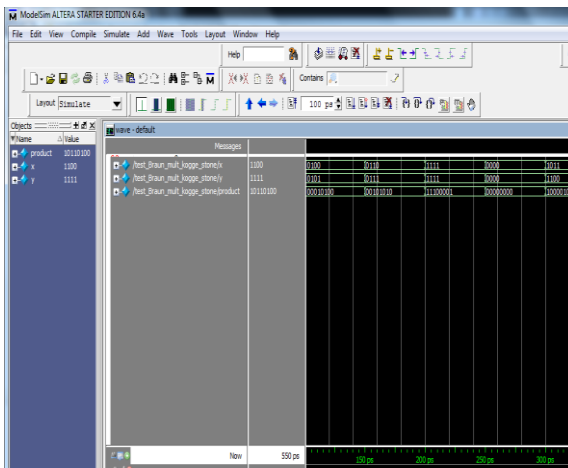


Fig6: Simulation result of New Braun Multiplier with KSA

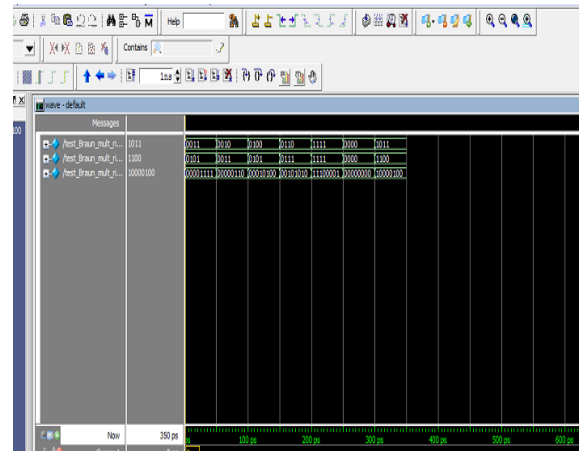


Fig7: Simulation result of Braun Multiplier with Ripple Carry Adder

4.1 FPGA Implementation using Spartan2 and Spartan2E

Kogge Stone Adder is the fastest Adder available but at the cost of area. The New Braun Multiplier using Kogge Stone Adder has less combinational delay as compared with existing Braun Multiplier (using Ripple Carry Adder) which is shown in Table 1.

The design of standard and modified braun multiplier with KSA (4x4 multiplier) are simulated using Verilog HDL and implemented on the Spartan2 (Xc2s15-6cs144 and Xc2s200-5fg256) and Sparatn2E (Xc2s100e-7ft256) FPGAs using the Xilinx ISE design tool.

Table 1:

Comparative results on FPGA implementation for Braun Multiplier using: Ripple Carry Adder and Kogge Stone Adder

S.No.	FPGA	Delay for Braun Multiplier using Ripple Carry Adder (ns)	Delay for New Braun Multiplier using Kogge Stone Adder (ns)
1.	Spartan 2 (xc2s 15-6cs144)	16.019	15.929
2.	Spartan 2 (xc2s 200-5fg 256)	18.446	18.346
3.	Spartan 2E (xc2s 100e-7ft256)	14.251	14.161

5 CONCLUSION

The results shown in Table 1 are for the new Braun Multiplier which is designed for two inputs 'X' and 'Y' each of width 4 bits. Hence, we can say that for the multiplier applications requiring inputs more than 4 bits, the proposed new design of Braun Multiplier using Kogge Stone adder would provide the result in effectively lesser time than compared with Ripple Carry Adder.

6 FUTURE WORK

In proposed design of Braun Multiplier if the Kogge Stone Adder is replaced by another Parallel Prefix Adder (Brent Kung Adder), then the design can be improved in terms of area.

REFERENCES

[1] Al-Khalili, Dr. A.J.(2006).”Parallel Prefix Adders”, Concordia University: Kostas Vitoroulis.
[2] Bhatia, Ashish and Sindhu, Anurag. “8-bit Kogge Stone Adder” , IIT Kanpur, Project Report : COURSE PROJECT, EE 619.

[3] Earle, J. G. et al, (July 12, 1965).”Latched Carry Save Adder Circuit for Multipliers” U.S. Patent 3,340,388.
[4] Gregg, Chris (2009). “Kogge-Stone Adder Applet”.
[5] Kogge, P.M., and Stone, H.S (August 1973). “A Parallel Algorithm for the Efficient Solution of a General Class of Recurrence Equations”, IEEE Transactions on Computers. Vol.C-22, No.8.
[6] Kungching, Chen (2005). “Types of adders”, M. Eng. Project.
[7] Palnitkar, Samir (1996). “Verilog HDL: A Guide to Digital Design and Synthesis”, SunSoft Press.
[8] P.Ramanathan and.P.T.Vanathi, “ Hybrid Prefix Adder Architecture for Minimizing the Power Delay Product”,2009, World Academy of Science, Engineering and Technology
[9] R ,Anitha, and V, Bagyaveereswaran (September 2011). “Braun’s Multiplier Implementation using FPGA with Bypassing Techniques”, International Journal of VLSI design & Communication Systems (VLSICS),Vol.2, No.3
[10] Seng, Yeo Kiat and Roy, Kaushik (2009). “Low Voltage, Low Power VLSI Subsystems”,TMC.
[11] Wanhannar, Lars (May 2008). “DSP Integrated Circuits”, Academic Press.
[12] Weste, Neil, Harris, David and Banerjee, Ayan (2009). “CMOS VLSI Design: A circuits and system perspective”, Pearson education.
[13] Wen, M.-C., Wang, S.-J. and Lin Y.-N.(12th May 2005). “Low-power parallel multiplier with column bypassing”,ELECTRONICS LETTERS, Vol. 41 No. 10.