

Available online at <http://www.mecspress.net/ijem>

# Design Of A Optimized Parallel Array Multiplier Using Parallel Prefix Adder

K.KalaiKaviya<sup>a</sup>, D.P.Balasubramanian<sup>b</sup>, S.Tamilselvan<sup>c</sup>

<sup>a,b</sup> PG Student, S.M.K Fomra Institute of technology, Chennai

<sup>c</sup> Asst Professor, S.M.K Fomra Institute of technology, Chennai

---

## 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 (Brent kung Adder) in place of Ripple Carry Adder. The architecture of standard Braun Multiplier is modified in this work for reducing the area and delay due to Ripple Carry Adder and performing faster multiplication of two binary numbers. The design is implemented using Microwind1, digital schematics (DSCH)

**Index Terms:** Array multiplier, carry save adder (CSA), Kogge stone Adder, parallel prefix adder ripple carry adder, Microwind, DSCH.

© 2013 Published by MECS Publisher. Selection and/or peer review under responsibility of the Research Association of Modern Education and Computer Science.

---

## 1. Introduction

For scientific computations, Multiplication is an important and predominance in all digital signal processing (DSP) applications and its subfields. Application Specific Integrated Circuits (ASICs) utilizes as special purpose processor for DSP algorithms [1]. Multiplication is a Repeated addition of n bits will give the solution for the multiplication. i.e. Multioperand addition process. The multioperand addition process needs two n – bit operands. It can be realized in n- cycles of shifting and adding. If the multiplicand is given be.

\* Corresponding author. Tel.: 00966; 533273146  
E-mail address: salem\_farh@yahoo.com

$$A = a_{n-1} \dots a_1 a_0 \text{ ---} \quad (1)$$

Multiplier is given by

$$B = b_{n-1} \dots b_1 b_0, \text{ ---} \quad (2)$$

Then product will be

$$P = P_{2n} P_{2n-1} \dots P_1 P_0 \quad (3)$$

Over the years the computational complexities of algorithms used in Digital Signal Processors (DSPs) have gradually increased. This requires a parallel multiplier to meet the performance demands.

### 1.1. Parallel Multipliers

Three important criteria to be considered in the design of multipliers are the chip area, speed of computation and power dissipation. Most advanced digital systems incorporate a parallel multiplication unit to carry out high speed mathematical operations. a microprocessor requires multipliers in its arithmetic logic unit and a digital signal processing system requires multipliers to implement algorithms such as convolution and filtering. Today, high speed parallel multipliers with much larger areas and higher complexity are used extensively in reduced Instruction set computers, digital signal processing and graphics accelerator. There are many different design methods of multipliers such as the serial-parallel multiplier, the baugh-wooley multiplier and wallace tree multiplier. to implement a multiplier a relatively simpler form of adders is the braun array.

### 1.2 Parallel Prefix adders

Parallel prefix adder is the most flexible and widely used for binary addition. Parallel Prefix adders are best suited for VLSI implementation. Number of parallel prefix adder structures have been proposed over the past years intended to optimize area, fan out, logic depth and inter connect count. Binary addition is the most fundamental and frequently used arithmetic operation. A lot of work on adder design has been done so far and much architecture have been proposed. When high operation speed is required, tree structures like parallel prefix adders are used [2] [6] In [2], Sklansky proposed one of the earliest tree prefix is used to compute intermediate signals. In the Brent Kung approach [4], designed the computation graph for area optimization. The KS architecture [3] is optimized for timing. The LF architecture [11], is proposed, where the fan out of gates increased with the depth of the prefix computation tree.

## 2. Research Methodology

The architecture of Braun Multiplier is modified by replacing the Ripple Carry Adder with a Brent Kung Adder. The reason for selecting Brent Kung Adder is that it is the Fastest adder and low area consumption adder.

### 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. 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.

### 2.1.1 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. 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.

### 2.1.2. 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.1 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  $D$ , then total propagation delay of ripple carry adder will be  $T$ .

### 2.2. Architecture:

An  $n \times n$  bit Braun multiplier [5],[6] 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

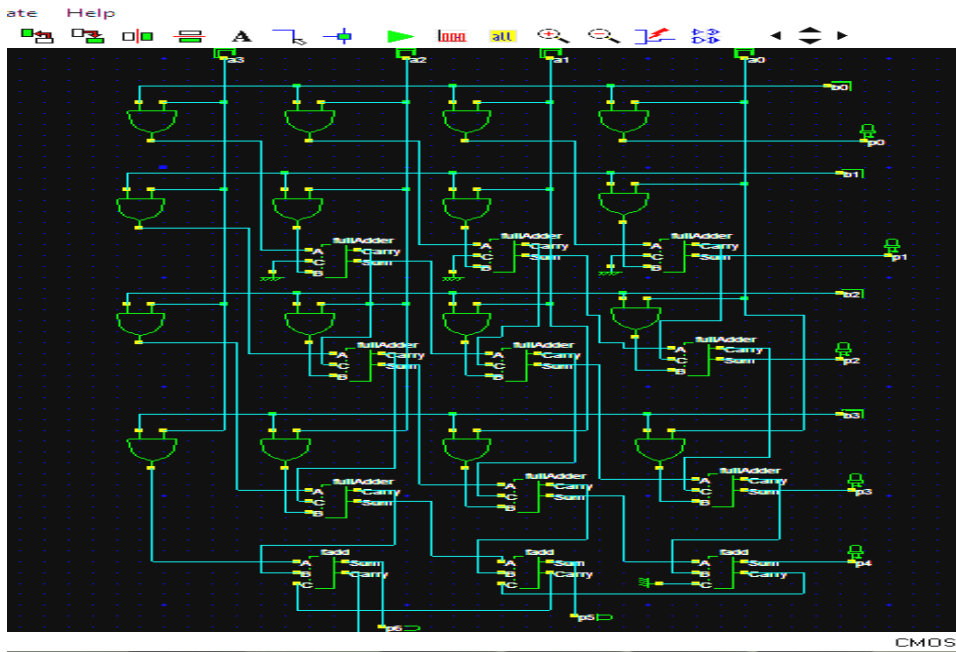


Fig 1: 4x4 Braun Multiplier Architecture

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) [6] and the Ripple carry adder should be used for the final stage of the output. Braun multiplier [5] 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.

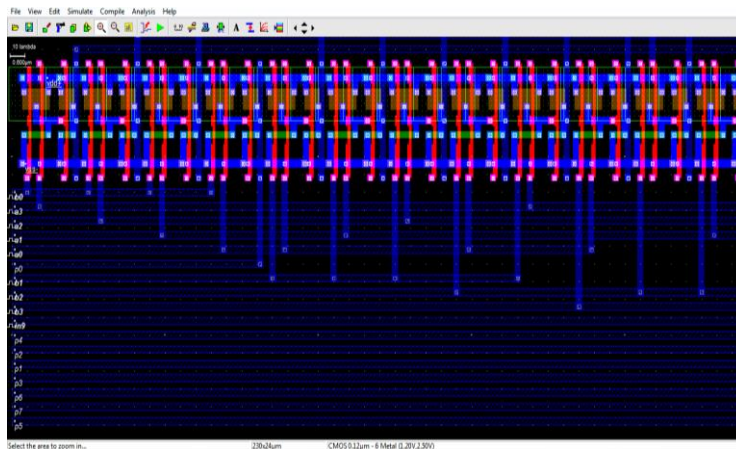


Fig 2 : Layout of Braun Multiplier using Ripple carry adder using Micro wind

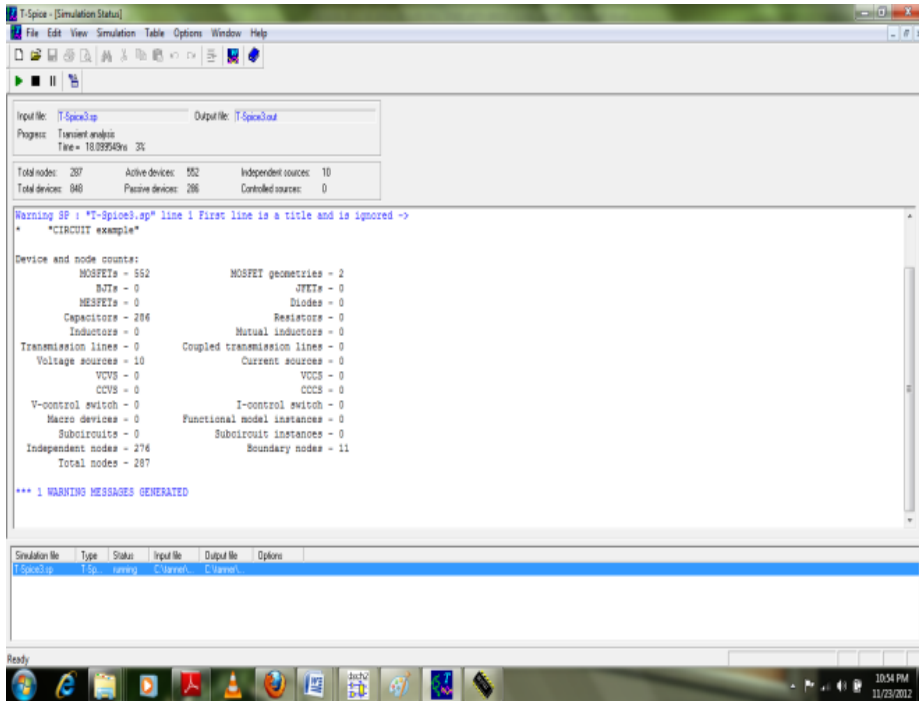


Fig 3: Transistor count using Tanner EDA

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 [2] “Brent kung adder” which is a type of carry look Head Adder.

### 3. Proposed Work - Design Of Braun Multiplier With Brent Kung Adder

The objective of this research is to design a new architecture of Braun Multiplier with Brent kung Adder which gives fast multiplication and less area.

#### 3.1 Architecture

The proposed multiplier’s block diagram is shown in fig.4 .The architecture of Braun multiplier with brent kung adder is shown in fig 5. where we have used a 3 bit Brent kung Adder in 4th stage of Braun multiplier [8].

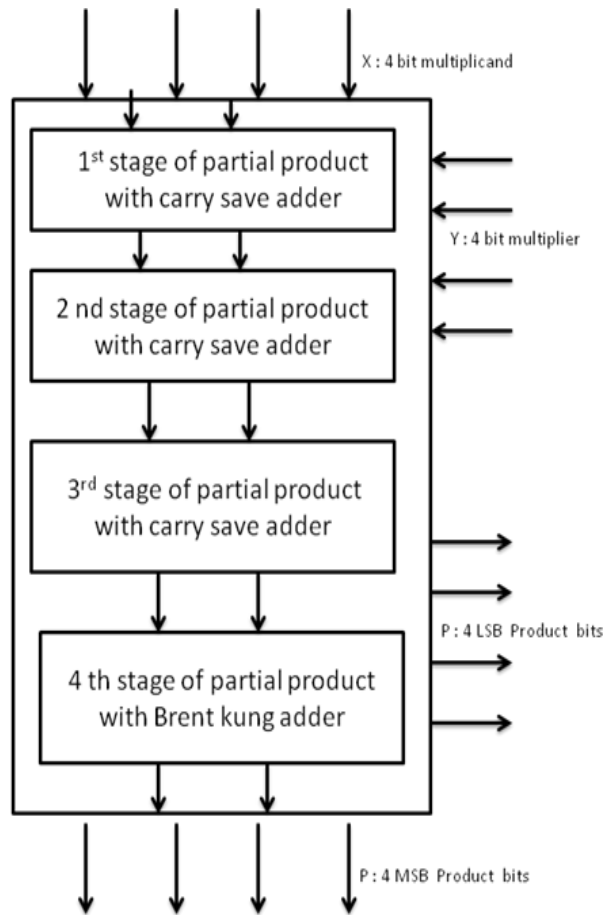


Fig 4: Block diagram of Braun multiplier with Brent Kung adder

### 3.2 Brent Kung Adder

In the following section, a structure known as the Brent Kung Structure which was first proposed by Brent and Kung in 1982 and which uses the logarithmic concept is discussed. This structure used an operator known as the dot (.) operator, which is explained in the architecture, for its basic blocks

#### 3.2.1. Brent Kung Architecture

The Brent Kung architecture is divided into three separate stages

1. Generate/Propagate Generation
2. The Dot (.) Operation
3. Sum generation

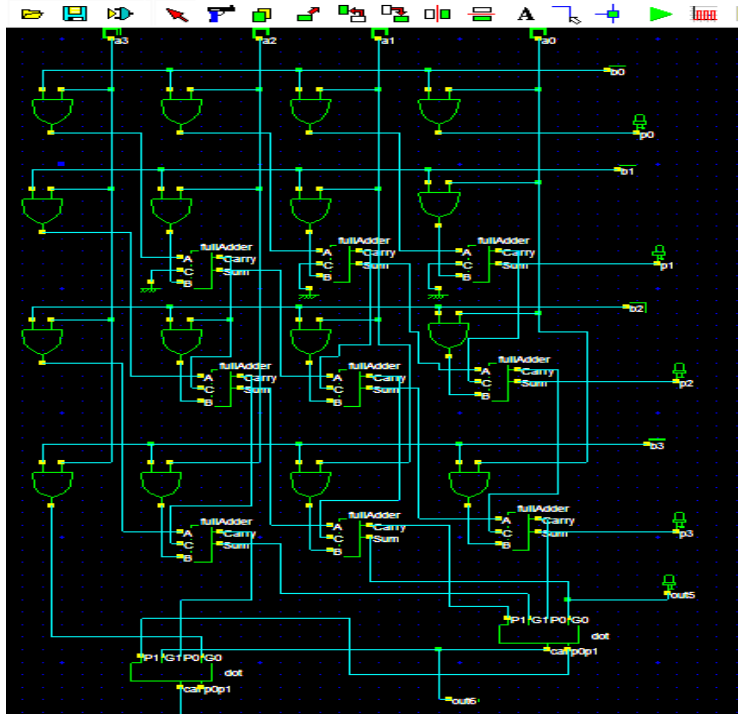


Fig 5: Architecture of Braun Multiplier with Brent kung adder

### 3.2.1.1 Generate/Propagate Generation

If the inputs to the adder are given by the signals A and B, then the generate and propagate signals are obtained according to the following equations.

$$G = A \text{ and } B \quad (4)$$

$$P = A \text{ xor } B \quad (5)$$

### 3.2.1.2 The Dot ( $\cdot$ ) Operation

The most important building block in the Brent Kung Structure is the dot ( $\cdot$ ) operator. The basic inputs to this structure are the generate and propagate signals obtained in the previous stage. The operator is a function that takes in two sets of inputs-- (g, p) and (g', p')--and generates a set of outputs-- (g + pg', pp'). These building blocks are used for the generation of the carry signals in the structure. For the generation of the carry signals, the carry for the kth bit from the carry look-ahead concept is given by.

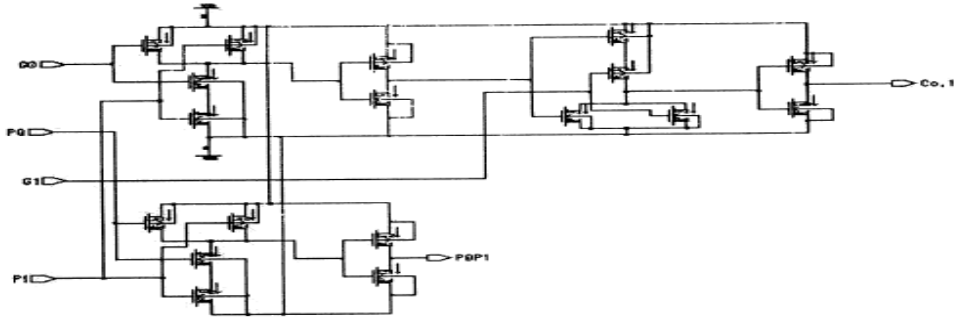


Fig 6: Design of the dot (•) operation

$$C_{o,k} = G_k + P_k(G_{k-1} + P_{k-1} + P_{k-1} + \dots + P_1(G_0 + P_0 C_{i,o})) \quad (6)$$

Using the dot operator explained above the Equation 3.3 can be written for the different carry signals as

$$C_{o,0} = G_0 + P_0 C_{i,0} = a(G_0, P_0) \quad (7)$$

$$C_{o,1} = G_1 + P_1 C_{o,0} = a((G_1, P_1), e(G_0, P_0)) \quad (8)$$

where  $a$  is a function defined in order to access all the tuples. The 8-bit Brent Kung Structure is shown in Figure 7.

This figure shows all the carry signals generated at different stages in the structure. In the structure, two binary tree organizations are represented – the forward and the reverse trees. The forward binary tree alone is not sufficient for generation of all the carry signals. It can only generate the signals shown as  $C_{o,0}$ ,  $C_{o,1}$ ,  $C_{o,3}$  and  $C_{o,7}$ . The remaining carry signals.

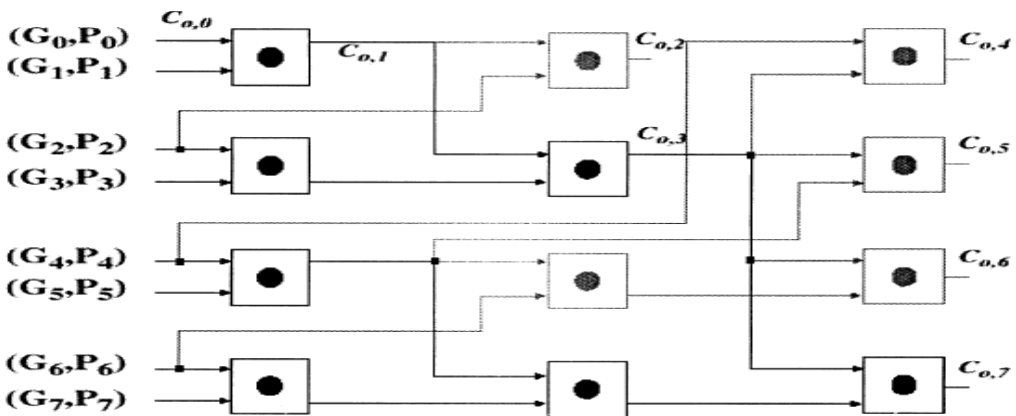


Fig 7: 8-bit Brent Kung Structure



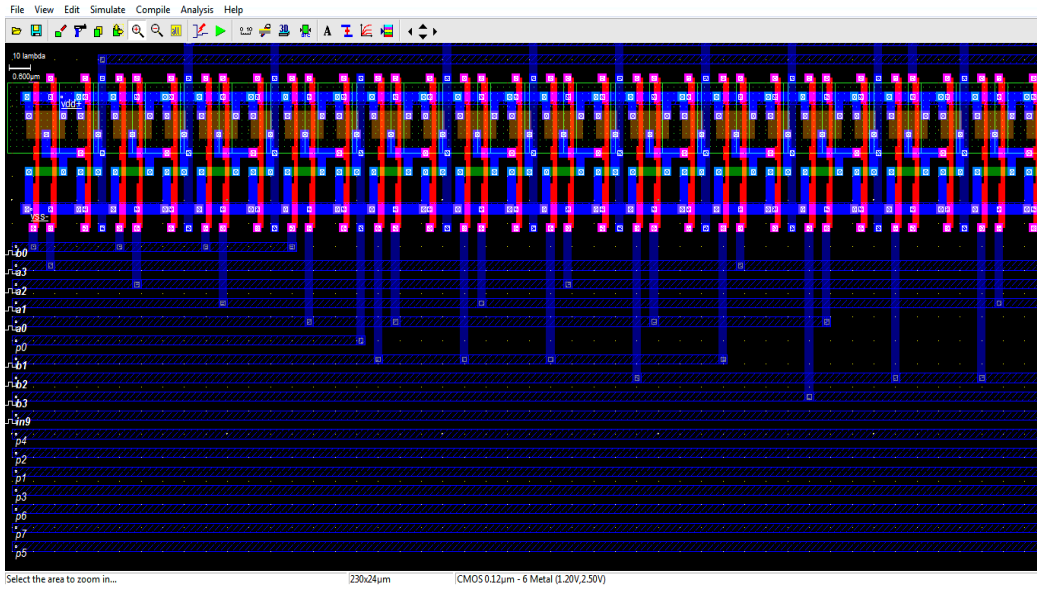


Fig 8: Layout of proposed adder using Microwind

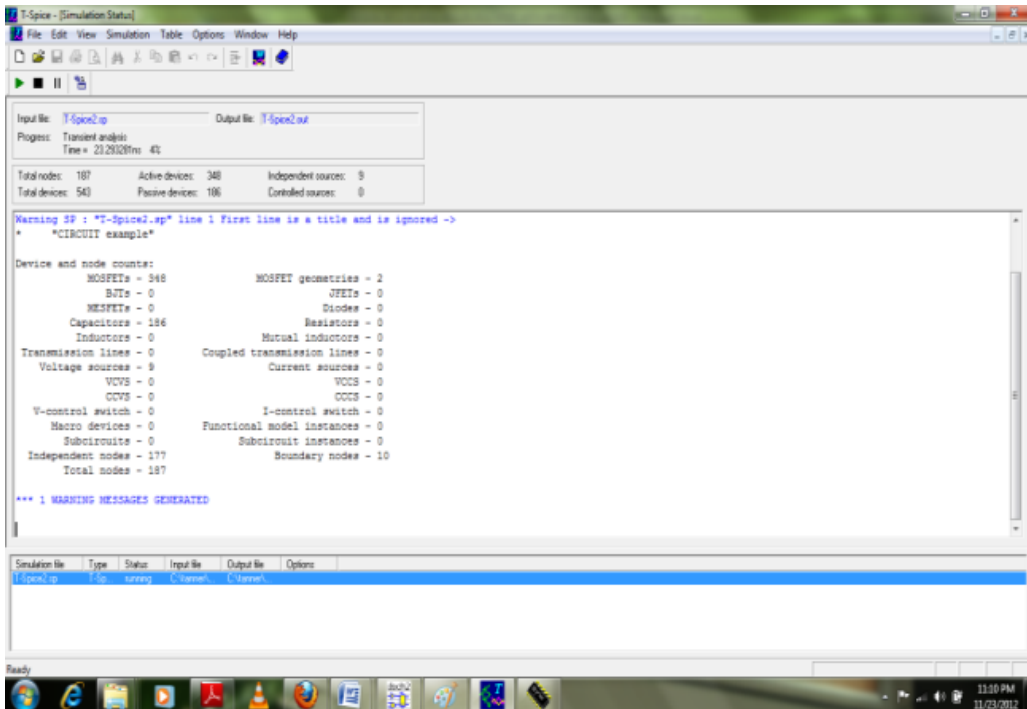


Fig 9: Transistor Count of proposed adder using Tanne

Table 1. Result analysis

S.no	Parameter	Ripple adder multiplier $\mu\text{m}^2$	Brent kung adder multiplier $\mu\text{m}^2$
1.	Area	5520	3580
2.	Transistor count	552	348

#### 4. Conclusion

The results shown in Table 1 are for the new Braun Multiplier which is designed for two inputs 'A' and 'B' each of width 4 bits. Hence, we can say that for the multiplier applications the proposed new design of Braun Multiplier using Brent Kung adder would provide the result in effectively lesser area than compared with Ripple Carry Adder we have a reduction in area of  $194 \mu\text{m}^2$ .

#### Reference

- [1] Muhammad H. Rais, "Hardware Implementation of Truncated Multipliers Using Spartan-3AN, Virtex-4 and Virtex-5 FPGA Devices", Am. J. Engg. & Applied Sci., 3 (1): 201-206, 2010.
- [2] Sklansky "conditional sum additions logic" IRE Transactions, Electronic Computers, vol, EC – 9, pp, 226 - 231, June 1960.
- [3] Kogge P, Stone H, "A parallel algorithm for the efficient solution of a general class Recurrence relations", IEEE Trans. Computers, vol.C-22, No.8, pp 786-793, Aug.1973.
- [4] Brent R, Kung H, "A regular layout for parallel adders". IEEE Trans, computers, Vol.C-31, no.3, pp 260-264, March1982 Academy of Science, Engineering and Technology.
- [5] 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.
- [6] Seng, Yeo Kiat and Roy, Kaushik (2009). "Low Voltage, Low Power VLSI Subsystems", TMC.
- [7] Wanhannar, Lars (May 2008). "DSP Integrated Circuits", Academic Press.
- [8] Weste, Neil, Harris, David and Banerjee, Ayan (2009). "CMOS VLSI Design: A circuits and system perspective", Pearson education.
- [9] 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.
- [10] P.Ramanathan and.P.T.Vanathi, "Hybrid Prefix Adder Architecture for Minimizing the Power Delay Product", 2009, World Academy of Science, Engineering and Technology.
- [11] Ladner R, Fischer M," Parallel prefix computation ", J.ACM, vol.27, no. 4, pp 831-838, Oct.1980.

**Authors Biography:**

**Er. S.Tamil Selvan** received the M. Tech degree in VLSI Design from Sathyabama University in 2011. He is currently working as Assistant Professor in Department of Electronics and Communication Engineering at SMK Fomra College of Engineering and technology in Chennai. Previously working at lecture in KCG College of technology in Chennai. His main research interests are in High speed digital VLSI circuits, Low power, Area reduction, synthesis and Simulation of digital circuits and FPGA Implementation.



D.P.Balasubramanian received his B.E. degree in Electronics and communication engineering at k.v.c.e.t under Anna university, Chennai. Currently pursuing his M.Tech Degree in S.M.K Fomra institute of technology. Area of interest is reversible logic, quantum electronics and RFID has presented papers in various national and international conferences.



K.Kalaikaviya received her B.E. degree in Electronics and communication engineering at Periyar Maniammai college of technology for women under Anna university, Tiruchirapalli. Currently pursuing her M.Tech Degree in S.M.K Fomra institute of technology. Area of interest is digital electronics, vlsi design, Testing of vlsi.

**How to cite this paper:** K.KalaiKaviya,D.P.Balasubramanian,S.Tamilselvan,"Design Of A Optimized Parallel Array Multiplier Using Parallel Prefix Adder", IJEM, vol.3, no.2, pp.40-50, 2013.DOI: 10.5815/ijem.2013.02.03