World Applied Sciences Journal 32 (11): 2303-2308, 2014
ISSN 1818-4952
© IDOSI Publications, 2014
DOI: 10.5829/idosi.wasj.2014.32.11.1354

Design of Twin Precision Multiplier with Low Power and Area Using Clock Pipelining

[1]Priya Stalin and [2]C. Arun

[1]Department of Electronics and Communication Engineering, St.Peter's University, St.Peter's Institute of higher education and Research, Chennai, India
[2]Department of Electronics and Communication Engineering, RMK College of Engineering and Technology, Chennai, India

Abstract: Design of a twin precision multiplier with less consumption of space for gates is the main objective of this paper. The existing system uses the twin precision multiplier as one n bit multiplier based on the length of the inputs. The drawback of this system is that it occupies the largest area of hardware. To overcome this drawback, a new technique to reduce the size of architecture is introduced. The results of the system will reduce the area to 50% occupied by gates using a novel method called clock pipelining.

Key words: Twin precision · Clock pipelining · Low power · Low area

INTRODUCTION

Multiplication is an essential function in basic arithmetic operations that is present in many parts of digital computers especially in Signal processing applications such as graphics and computation system [1]. Multiplication is nothing but a series of repeated additions. Multiplication process includes multiplicand and the product. The number that is to be added is called multiplicand and result of it is the product. The number of times that the multiplicand is added is called as multiplier [2]. In most of the digital systems such as computers, the number of bits will be the same in the given operands. The length of the result between two integers will be double the operand length generally which saves the informations [3].

Steps in a Multiplication Process: The multiplication process is always the product of two binary numbers. The process involves an addition operation and a shifting in each addition also which takes place in one multiplier bit [4]. Arithmetic circuits form an important class of circuits in digital systems. Among arithmetic operations, multiplier requires more hardware resources in digital systems [5]. The steps involved in the multiplication process are signed and unsigned. It can be explained through some algorithms [6].

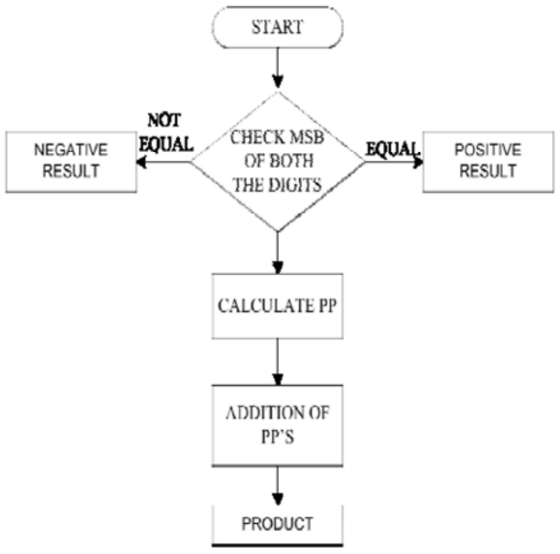

Fig. 1: Flow Diagram of Multiplier

Unsigned and Signed Multiplication Algorithm: Unsigned Multiplication can be done using serial addition. At every step of the multiplication process that starts from the LSB of the multiplier, a single bit multiplier is functioned to give a partial product [7]. The result of this multiplier is further shifted right by single bit before adding to the next bit multiplication. Then, the carry from the result is used to provide the next bit [8]. At every

Corresponding Author: Priya Stalin, Department of Electronics and Communication Engineering, Sree Sastha Institute of Engineering and Technology, Chennai, India. E-mail: priyastalin72@gmail.com.

stage, the LSB of the addition becomes one bit of the result similar to unsigned multiplication process, signed multiplication can be done using serial addition [9]. For an unsigned process, the carry out provides the last fifth bit. But this bit is not suitable for signed addition if the sign of the two numbers are different [10]. Therefore, to ensure that a 4-bit signed addition gives the correct 5 bit result. The input numbers must first be sign extended to form a 5-bit signed number before the addition [11].

Multiplier and its Types: Multiplication is a complex arithmetic operation, which possesses relatively high signal propagation delay, high power dissipation and large area requirement. When choosing a multiplier for a digital system, the bit width of the multiplier is required to be at least as wide as the largest operand of the applications that has to run on that digital system. The bit width of the multiplier is, therefore, often much larger than its operands, which leads to excessive power dissipation and long delay. Several multipliers with the same bit width could also be used in a Single Instruction Multiple Data fashion, in order to increase throughput, thus reducing total execution time for multiplication-intensive applications. Multipliers can be classified based on which data words are accessed, namely:

- Serial form,
- Parallel form,
- Serial-parallel form.

Depending on the application requirements multipliers can be classified since the multiplication can be performed at a faster rate and others concentrate on less hardware and moderate speed. There are different types of multipliers based on architecture, function, power consumption etc. But basically there are three types of multipliers. They are

- Binary Multiplier
- Array multiplier
- MAC Multiplier

In the binary number system the digits, called bits are limited to the set [0, 1]. The result of multiplying any binary number by a single binary bit is either 0 or the original number. This makes forming the intermediate partial-products simple and efficient. The sum of partial-products is the only way to consume time for binary multipliers. The only one approach is to produce the

```
                 1 1 0 1       Multiplicand
             x   0 1 0 1       Multiplier
             -------------------------------
         1 1 1 1 1 1 0 1       PP1
         0 0 0 0 0 0 0         PP2
         1 1 1 1 0 1           PP3
    +      0 0 0 0 0           PP4
  ------------------------------------------
       1 1 1 1 1 0 0 0 1=-15   Product
```

Fig. 2: Multiplication process

single partial-products for a single time and sum up them to get the result. Often these are implemented through software on processors but that do not have a hardware multiplier, this technique works correctly, but it is slow because a single machine cycle is required to sum up each additional partial product.

Multiplier can be implemented directly in hardware where the above approach does not provide good performance.The two main classifications of binary multiplication are signed and unsigned numbers. Multiplication of digits include a series of shifting the bits and series of adding the bits. The two numbers are multiplicand and multiplier which then are combined to produce the result. Considering the bit representation of the multiplicand x= x_n-1…..x_1 x_0 and the multiplier y = y_{n-1}…..y_1y_0 in order to produce the product, up to n shifted operations of the multiplicand are to be added for unsigned multiplication. The whole process can be classified into three steps, such as generation of partial product, partial product reduction and final addition.

Array Multiplier: The Fast power efficient circuit block switch-off scheme proposes a double switch circuit block switch scheme capable of reducing power dissipation during down time by shortening the settling time after reactivation. The drawbacks of this scheme are the necessary of two independent virtual rails and two additional transistors for switching each cell. A power scalable pipelined Booth multiplier design which has scalable pipelined architecture presents a multiplier using Dynamic Range Detection unit. This unit is used to select the input operand with a smaller effective dynamic range to yield the Booth codes.

There are three separate Wallace architectures of trees for the 4X4, 8X8 and 16X16 multiplications. This will certainly induce area and capacitance penalties. Under a CMOS technology, this design can obtain a twenty percent power reduction over the conventional

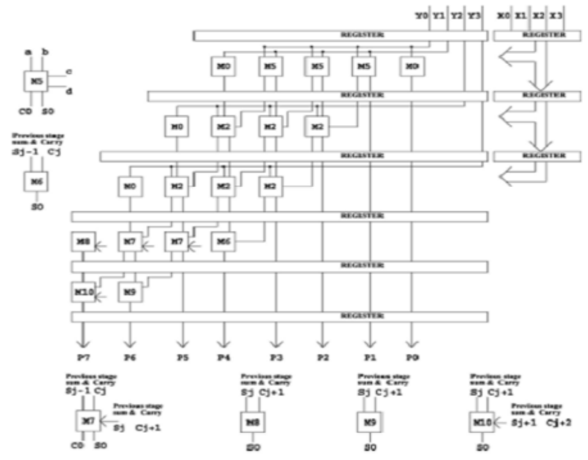

Fig. 3: Array Multiplier

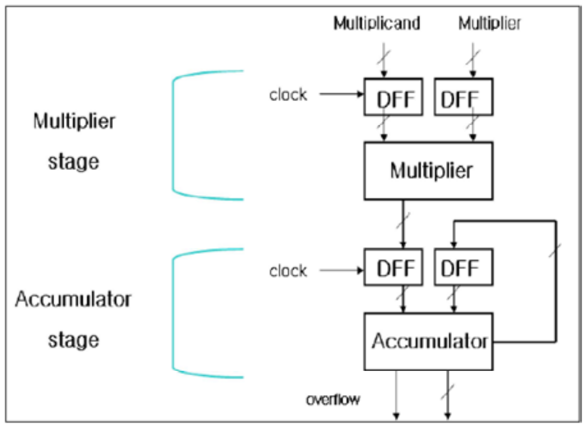

Fig. 4: Simple MAC Architecture

multipliers. But there is a forty four percent area overhead. Partially guarded computation divides the arithmetic units into two main categories such as adders and multipliers, turns off the unusable part in order to reduce the power consumption.

Multiplier and Accumulator: The inputs for the MAC are to be fetched from memory location and fed to the multiplier block of the MAC, which performs multiplication and gives the result to the adder which accumulates the result and then stores the result into a memory location. This entire process is to be achieved in a single clock cycle (Weste and Harris, 3rd Ed). The architecture of the MAC unit which has been designed in this work consists of one 16 bit register, one 16-bit Modified Booth Multiplier, 32- bit accumulator. A Modified Booth multiplier is used instead of conventional multiplier because it increases the MAC unit design speed and reduce multiplication complexity.

SPST Adder is used for the addition of partial products and a register is used for accumulation. The operation of the designed MAC unit is as in Equation 2.1. The product of Ai X Bi is always fed back into the 32-bit accumulator and then added again with the next product Ai x Bi. This MAC unit is capable of multiplying and adding with the previous product consecutively up to as many times as required.

Twin Precision Multiplier: In high performance digital systems such as microprocessors, FIR filters and digital signal processors etc., the multiplier is one of the key hardware blocks. So the design of multipliers stand challenging with advancement in technology. Many researchers have tried and are trying to design multipliers which offer either of the following- high speed, low power consumption, regularity of layout and hence less area or even combination of them, thereby making them suitable for various compact, low power and high speed VLSI implementations. However area and speed are two conflicting constraints. So, improving speed results in larger area and vice versa. Hence we try to find out the best trade off solution amongst them.

In recent trends the column compression multipliers are popular for faster computations due to their higher speeds [1-2]. The first column compression multiplier was introduced by Wallace in 1964 [3]. In 1965, Dadda altered the approach of Wallace by starting with the exact placement of the (3,2) counters and (2,2) counters in the maximum critical path delay of the multiplier [4]. In 2006, H. Eriksson along with his research team presented HPM reduction tree structure that has an ease of layout compared to Dadda's approach [5]. Compared to Dadda, HPM is slightly faster and consumes lesser power while area remains the same. So implementation of the multiplier design using HPM is done.

The total delay of the multiplier can be split up into three parts: 1. The Partial Product Generation (PPG) 2. The Partial Product Summation Tree (PPST) and 3.The Final Adder [6]. Of these the dominant components of the multiplier delay are due to the PPST and the final adder. The relative delay due to the PPG is small. Therefore significant improvement in the speed of the multiplier can be achieved by reducing the delay in the PPST and the final adder stage of the multiplier. Here the PPST delay is reducewd using faster multiplication technique of performing the N-bit multiplication by '4' N/2-bit multiplications running in parallel and by using ahybrid adder the final adder delay is reduced.

The bit width of the multiplier is same as that of the bit width of the largest operand of the application that the processor executes. But most of the times the operands do not occupy the maximum width and utilizes the resources unnecessarily which results in power loss. In the year 2005 Magnus Sjalander explored this idea to reduce this type of power consumption by using operand guarding technique and named it as Twin Precision Technique [6]. Now in this paper utilizing the same property for reducing the power and operator isolation is being performed using clock gating technique. The remaining paper is organized as follows: Section II describes the design of the faster multiplier structure. Section III describes the design of hybrid adder and clock gating. Section IV is all about result analysis. Section V is the Conclusion. Section VI includes the bibliography.

Twin Precision multiplier is a type of unsigned multiplication process. It can be implemented using both unsigned and signed multiplication process. In an unsigned binary multiplication each bit of one of the operands called the multiplier is multiplied with the second operand called the multiplicand. Thus one row of partial products is generated. Each row of partial product is shifted according to the position of the bit of the multiplier forming what is commonly called the partial-product array. Finally, partial products that are in the same column are summed together, forming the final result. But the unsigned multiplication is of limited use. It can be implemented using signed multiplication process through algorithm called Baugh-Wooley(BW).

The BW algorithm is a direct way of doing signed multiplications. The creation of the reorganized partial-product array comprises of three steps: *i)* the most significant partial product of the first N- 1 rows and the last row of partial products except the most significant has to be negated, *ii)* a constant one is added to the Nth column, *iii)* the most significant bit (MSB) of the final result is negated.

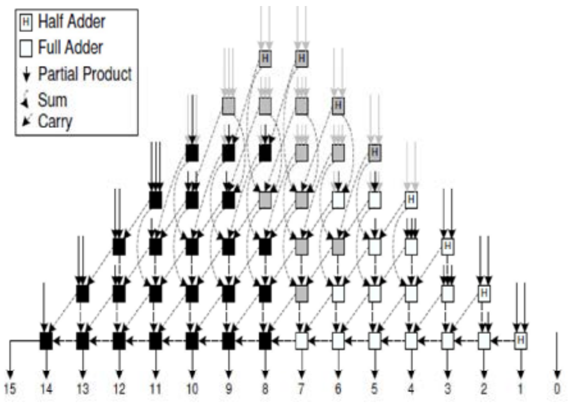

Fig. 5: Block diagram using HPM Implementation

$$
\begin{array}{cccccccccccccccc}
 & & & & & & & & y_7 & y_6 & y_5 & y_4 & y_3 & y_2 & y_1 & y_0 \\
 & & & & & & & & x_7 & x_6 & x_5 & x_4 & x_3 & x_2 & x_1 & x_0 \\
\hline
 & & & & & & & 1 & \overline{p_{70}} & p_{60} & p_{50} & p_{40} & p_{30} & p_{20} & p_{10} & p_{00} \\
 & & & & & & & \overline{p_{71}} & p_{61} & p_{51} & p_{41} & p_{31} & p_{21} & p_{11} & p_{01} & \\
 & & & & & & \overline{p_{72}} & p_{62} & p_{52} & p_{42} & p_{32} & p_{22} & p_{12} & p_{02} & & \\
 & & & & & \overline{p_{73}} & p_{63} & p_{53} & p_{43} & p_{33} & p_{23} & p_{13} & p_{03} & & & \\
 & & & & \overline{p_{74}} & p_{64} & p_{54} & p_{44} & p_{34} & p_{24} & p_{14} & p_{04} & & & & \\
 & & & \overline{p_{75}} & p_{65} & p_{55} & p_{45} & p_{35} & p_{25} & p_{15} & p_{05} & & & & & \\
 & & \overline{p_{76}} & p_{66} & p_{56} & p_{46} & p_{36} & p_{26} & p_{16} & p_{06} & & & & & & \\
 & p_{77} & \overline{p_{67}} & \overline{p_{57}} & \overline{p_{47}} & \overline{p_{37}} & \overline{p_{27}} & \overline{p_{17}} & \overline{p_{07}} & & & & & & & \\
\hline
\overline{s_{15}} & s_{14} & s_{13} & s_{12} & s_{11} & s_{10} & s_9 & s_8 & s_7 & s_6 & s_5 & s_4 & s_3 & s_2 & s_1 & s_0
\end{array}
$$

Fig. 6: Block diagram of Twin Precision Multiplier using Baugh-Wooley.

Literature Survey: Multiplication is one of the essential operations in Digital Signal Processing (DSP) applications like Fast Fourier Transform (FFT), Digital filters etc. With the advancements in technology, researches are still going on to design a multiplier that consumes less power or has high speed or occupies less area or a combination of these in a single multiplier. This makes the multipliers to be used for high speed or low power VLSI applications. The Braun's multiplier is one of the parallel array multipliers which is used for unsigned number multiplication. The dynamic power of the multiplier can be reduced by using the bypassing techniques. The delay can be reduced by replacing the ripple carry adder in the last stage by fast adders like Carry look ahead adder and Kogge stone adder. This paper presents a comparative study among different types of bypassing multipliers for 4*4, 8*8 and 16*16 bits and their architectural modifications using different FPGAs like Spartan – 3E, Virtex – 4, Virtex – 5 and Virtex – 6 Lower power using Xilinx 13.2 ISE tool from which the delay and the dynamic power and cell area reports are obtained using RTL Compiler using Cadence in 90 nm technology. Low power consumption and smaller area are some of the most important criteria for the fabrication of DSP systems and high performance systems. Optimizing the speed and area of the multiplier is a major design issue. However, area and speed are usually conflicting constraints so that improving speed results mostly in larger areas. In our work we try to determine the best solution to this problem by comparing a few multipliers. This paper presents an efficient implementation of high speed multiplier using the shift and add method, Radix_2, Radix_4, modified Booth multiplier algorithm. In this papert we compare the working of the three multipliers by implementing clock pipelining each of them separately and an in FIR filter also.The parallel multipliers like radix 2, radix 4 and modified booth multiplier does the computations using lesser adders and lesser iterative steps. As a result of which they occupy lesser space as compared to the serial multiplier. This is a very important criteria because the fabrication of chips and high performance system require components which are as small as possible. In the power consumption of all the multipliers we find that serial multipliers consume more power. So where power is an important criteria there parallel multipliers like booth multipliers instead of serial multipliers are preferred. The low power consumption quality of booth multiplier makes it a preferred choice in designing different circuits. In this paper first three different types of multipliers are designed using shift method, radix 2 and radix 4 modified booth multiplier algorithm. used Different types of adders like sixteen bit full adder in designing the multiplier are used. Then a 4 tap delay FIR filter is designed and in place of the multiplication and addition the components of different multipliers and adders are implemented. Then he working of different multipliers are compared in terms of the power consumed by each of them. The result of our paper helps us to choose a better option between serial and parallel multiplier in fabricating different systems. Multipliers form one of the most important components of many systems. So analysis of the working of different multipliers become significant. Developing an Application Specific Integrated Circuits (ASICs) costs very high, the circuits should be proved and then it would be optimized before implementation. Multiplication is the basic building block for several DSP processors, Image processing and many others. The Braun multipliers can easily be implemented using Field Programmable Gate Array (FPGA) devices. This research presents the comparative study of Spartan-3E, Virtex-4, Virtex-5 and Virtex-6 Low Power FPGA devices. The implementation of Braun multipliers and its bypassing techniques is done using Verilog HDL. The comparison of the results are done and the proposed method is effective when compared to the conventional design. There is the reduction in the resources like delay LUTs, number of slices used. Simulations are done and they are verified using the Spartan-3E, Virtex-4 and Virtex-5 devices. The Virtex-5 FPGA has shown good performance as compared to Spartan-3E and Virtex-4 FPGA devices.

Methodology: The block diagram of twin precision multiplier indicates the N-bit multiplier. Generally four multipliers will be present in a conventional multiplier circuit and now this concept explains with two multipliers. The unit is divided into two N/2 multiplier with bit selection pin. This pin determines the multiplier to be used. Each N/2 multiplier consists of clock signal and four pins. The four pins such as a1, a2, b1, b2. The inputs on a1, b1 are processed during the positive clock signal and the inputs on a2, b2 are processed during the negative clock signal.

The purpose of connecting a 2bit selector to an OR gate and an and gate is to determine the choice of multiplier that has to be selected. To select any one of the multipliers, OR gate has to function and similarly to make both the multipliers work, AND gate should be selected. The performance of the multiplier also gets increased with presence of two multipliers and also it reduces the area that is occupied by extra two multipliers.

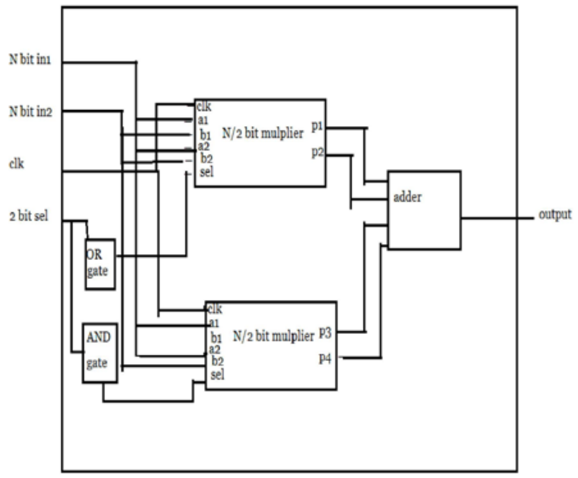

Fig. 7: Clock Pipelining

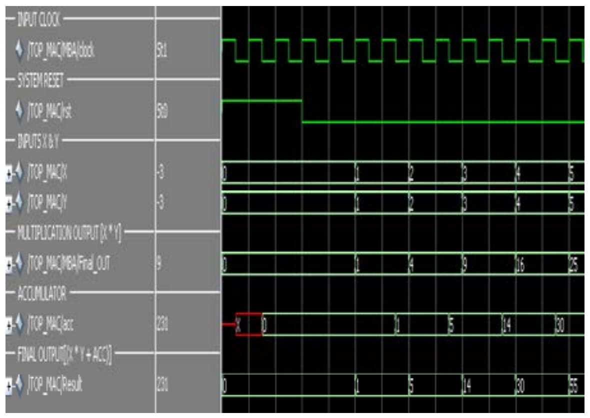

Fig. 8: Using UnSigned numbers

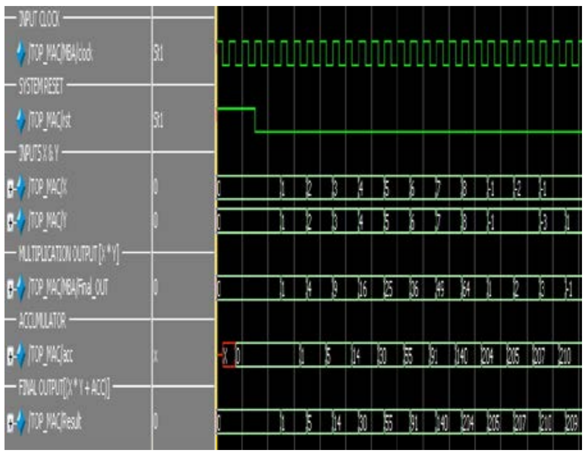

Fig. 9: Using Signed numbers

RESULTS AND CONCLUSION

In this paper, the twin precision multiplier is designed with half 50% reduced space area in hardware resources using clock pipelining process. Compared to other circuits, the twin precision multiplier has the highest operational speed and less hardware count. The basic building blocks for the twin precision unit are identified and each of the blocks is analysed for its performance. Power and delay is calculated for the blocks. 1-bit Twin unit is designed with enable pin to reduce the total power consumption based on block enable technique. Using this block, the N-bit twin precision unit is constructed and the total power consumption is calculated.

REFERENCES

1. Mead, C. and L. Conway, 1988. ‘Introduction to VLSI Systems”, Massachusetts, Addison-Wesley, 1980. T. Meng, “Asynchronous Design for Digital Signal Processing Architectures”, Ph. D. Thesis, University of California at Berkeley, November 1988.
2. Meng, T., 1989. “Design of Clock-Free Asynchronous Systems for Real Time Signal Processing”, Proceedings of IEEE ICASSP, May 1989.
3. Miller, R.E., 1965. “Switching Theory”, John Wiley and Sons, Inc., New York, 1965.
4. Noll, T.G., D. Schmitt-Landsiedel, H. Klar and G. Enders, 1986. “A Pipelined 330-MHz Multiplier”, IEEE Journal of Solid-State Circuits, SC-21, no. 3, June 1986.
5. Noice, D.C., “A Clocking Discipline for Two-Phase Digital.
6. Wen, M.C., S.J. Wang and Y.M. Lin, 2005. “Low power parallel multiplier with column bypassing,“ IEEE International Symposium on Circuits and Systems.
7. Ohban, J., V.G. Moshnyaga and K. Inoue, 2002. “Multiplier energy reduction through Bypassing of partial products”, IEEE Asia-Pacific Conference on Circuits and Systems, pp: 13-17.
8. Sung, G.N., Y.J. Ciou and C.C. Wang, 2008. “A power aware 2-dimensional bypassing multiplier using cell-based design flow”, IEEE International Symposium on Circuits and Systems.
9. Yan, J.T. and Z.W. Chen, 2009. “Low-power multiplier design with row and column bypassing,” IEEE International SOC Conference, pp: 227-230.
10. Muhammad, H. Rais, 2010. Hardware Implementation of Truncated Multipliers Using Spartan-3AN, Virtex-4 and Virtex-5 FPGA Devices, Am. J. Engg. and Applied Sci.
11. Anitha, R. and V. Bagyaveereswaran, 2011. “Braun’s Multiplier Implementation using FPGA with Bypassing Techniques”, International Journal of VLSI Design and Communication Systems (VLSICS) Vol. 2, No. 3, September, 2011.