Dissertation

on

Design & Performance Analysis of MAC Unit

Submitted in partial fulfilment of the requirements

for the award of the degree of

Master of Technology

in

VLSI Design and Embedded System

Submitted by:

Mohammad Rizwan Uddin Shaikh

(Roll No. 2K12/VLS/21)

Under the Guidance of

Dr. S. Indu

(Associate Professor)

Department of Electronics and Communication Engineering

Delhi Technological University

Main Bawana Road Delhi-110042

2012-2015

Dissertation

on

Design & Performance Analysis of MAC Unit

Submitted in partial fulfilment of the requirements

for the award of the degree of

Master of Technology

in

VLSI Design and Embedded System

Submitted by:

Mohammad Rizwan Uddin Shaikh

(Roll No. 2K12/VLS/21)

Under the Guidance of

Dr. S. Indu

(Associate Professor)

Department of Electronics and Communication Engineering

Delhi Technological University

Main Bawana Road Delhi-110042

2012-2015

Department of Electronics and Communication Engineering

Delhi Technological University

Main Bawana Road Delhi-110042

CERTIFICATE

This is to certify that the dissertation titled "**Design & Performance Analysis of MAC Unit**" is a bonafide record of work done by **Mohammad Rizwan Uddin Shaikh**, Roll No. **2K12/VLS/21** at Delhi Technological University in partial fulfilment of the requirements for the award of degree of Master of Technology in VLSI Design and Embedded System. This project was carried out under my supervision and has not been submitted elsewhere, either in part or full, for the award of any other degree or diploma to the best of my knowledge and belief.

Date: ________

Dr. S. Indu

(Assistant Professor)

Department of Electronics and Communication Engineering

Delhi Technological University Delhi

ACKNOWLEDGEMENTS

I would like to express my deep sense of respect and gratitude to my project supervisor **Dr. S. Indu**, Associate Professor, Department of Electronics and Communication Engineering, Delhi Technological University Delhi for providing the opportunity of carrying out this project and being the guiding force behind this work. I am deeply indebted to her for the support, advice and encouragement she provided without which the project could not have been a success. I am also grateful to **Prof. Prem R. Chadha**, HOD, Department of Electronics and Communication Engineering, DTU for his immense support.

A special thanks to **Dr. Neeta pandey**, Associate Professor, Electronics and Communication Engineering Department, DTU **Dr. Ashok Bhattacharya**, Retired Professor, Electronics and Communication Engineering Department, DTU for giving me valuable guidance and support. Their enormous knowledge and investigation has helped me unconditionally to solve various problems..

I would also like to acknowledge Delhi Technological University for providing the right academic resources and environment for this work to be carried out.Last but not the least I would like to express sincere gratitude to my parents and my colleagues for constantly encouraging me during the completion of work.

Mohammad Rizwan Uddin Shaikh
University Roll no: 2K12/VLS/21
M.Tech (VLSI Design and Embedded System)
Department of Electronics & Communication Engineering
Delhi Technological UniversityDelhi

ABSTRACT

ALU is a central block in the computing devices, especially Digital Signal Processor (DSP). Basic ALU consists of arithmetic unit, logic unit and control unit.In order to achieve high performance MAC unit is incorporated in the design of ALU.MAC unit performsmultiplication and accumulation process. Basic MAC unitconsists of multiplier, adder, and accumulator.In the existing MAC unit designed using Dadda Multiplier and adder as Carry Save Adder (CSA). The proposed MAC unit designed using Dadda Multiplier and adder as Carry Increment Adder (CSA).However in the proposed model all traditional full adders are replaced by improved full adder. The performance analysis of MAC unit models in terms of area, delay and power is done.Various MAC unit models are designed using Verilog HDL. Simulation and synthesis are done using Xilinx ISE 12.2 for Virtex-6 family 40nm technology device. The power is calculated using Lattice Diamond Design suite software.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

ALU	Arithmetic Logic Unit
MAC	Multiply Accumulate
ADD	Adder
RCA	Ripple Carry Adder
CSkA	Carry Skip Adder
CIA	Carry Increment Adder
CLAA	Carry Look Ahead Adder
CSA	Carry Save Adder
CS***l***A	Carry Select Adder
CBA	Carry Bypass Adder
AM	Array Multiplier
WTM	Wallace Tree Multiplier
RCAM RB	Ripple Carry Array Multiplier with Row Bypass Technique
DM	Dadda Multiplier
VM	Vedic Multiplier
Acc	Accumulator
HA	Half Adder
FA	Full Adder
TFA	Full Adder
LOFA	Logically Optimized Full Adder
ISE	Integrated System Environment
HDL	Hardware Description Language

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

The basic MAC unit comprises Multiplier, Adder and Accumulator. This unit computes the running sum of products, which is at the heart of algorithms such as the FIR and FFT. The general form of MAC operation is characterized by below equation.

$$M = (a * b) + acc \qquad (1)$$

Where a and b are two binary numbers of length N bits. M and acc are with the length of 2N bits or more to store the result of addition. The MAC unit can be alienated into two main blocks, the multiplier and accumulator. The multiplier also further divided into several blocks such as partial product generator, reduction of partial products, etc. The partial addition block comprises of summation tree and final adder. The summation block is key part of the MAC unit; this block consumes most of the circuit power and delay and occupies most of the area as well. Several MAC unit models can be obtained by replacing the multiplier unit with various architectures. The simple MAC unit block diagram [3] is shown in Fig.1.1.

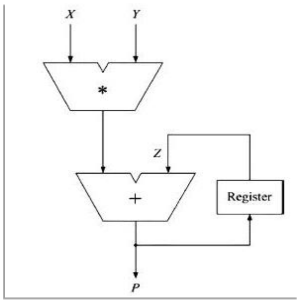


Fig.1.1.MAC Unit Block Diagram

The devoted MAC unit comprises multiplier followed by an adder and an accumulator register that stores outcome. The output of register is fed back to the one of the input of the adder, so that each clock cycle, the output of the multiplier is added to the register. And the MAC unit is controlled by means of the reset and clock.

1.2 MOTIVATION

Research and development in the field of digital electronics has revolutionized human life. Because of developments in IC technology isolated circuits are almost obsolete. The need of scaling down of chip area is increasing very rapidly. It seems that high speed; less complexity; high density; low power consumption; price etc. will be the major factor of apprehension in the ultramodern device designing. These motivate us to do research on most basic and essential building block of processor. Since MAC unit computes the running sum of products which is the most widely executed operation [4]. Therefore we have chosen the MAC unit for optimization.

1.3 APPLICATIONS

In order to design an ALU for high performance applications like Digital Signal Processor (DSP), it is needed to incorporate Multiply-Accumulate (MAC) unit. Digital signal processors have diverse architectures and improved features than general purpose processors. This makes multiplier-and-accumulator (MAC) an essential elements of the digital signal processing such as filtering, convolution, and inner products. Most digital signal processing methods use nonlinear functions such as discrete cosine transform (DCT) or discrete wavelet transform (DWT). Because they are basically accomplished by repetitive application of multiplication and addition, the speed of the multiplication and addition arithmetic determines the execution speed and performance of the entire calculation. The capability to calculate with a fast MAC unit is essential to attain high performance in many DSP algorithms.MAC unit are also indispensible element of digital filters.

1.4 OUTLINE OF THE PROJECT

MAC unit is a basic building block of ALU. In order to achieve high performance applications, MAC unit is incorporated in the design of ALU. This unit computes the running sum of products, which is at the heart of algorithms such as the FIR and FFT. The capability to calculate with a fast MAC unit is essential to attain high performance. Thus performance of MAC unit majorly affects the speed of overall system.

MAC unit consists of multiplier, adder and accumulator. In the existing MAC unit is designed using Dadda Multiplier (DM) and adder as Carry Save Adder (CSA) [4]. A new MAC unit model is proposed. Then, the performance analysis of proposed MAC unit models such as area, delay and power are done. Both MAC unit models are designed using Verilog HDL. Then, the complete simulation and verification is performed using Xilinx ISE 12.2. Lattice Diamond is used for synthesis and calculation of parameters for lattice FPGA families.

1.5 OPTIMIZATION FLOW

- Half Adders are replaced by Full adder wherever possible.
- Full adders are replaced by logically optimized full adder proposed by R.Uma and P.Dhavachelvan [5].
- In existing MAC unit Carry Save adder is replaced by Carry Increment adder because this topology provides better performance for 16 bit application [5].

1.6 ORGANIZATION OF THE THESIS

The thesis is organized as follows: Chapter 1 deals with the introduction about Multiply-Accumulate (MAC) unit. In chapter 2, the literature survey and the existing models of Multiply-Accumulate unit are discussed. Chapter 3 deals with proposed models of MAC unit. In chapter 4, simulation results of adders, Dadda multiplier, existing & proposed MAC unit models are discussed. Conclusion and future work are discussed in chapter 5.

CHAPTER 2

LITERATURE SURVEY

This chapter provides in detail about the previously work done in this domain. As concern to this research topic, during research session the references of many important involvements were very helpful during research.

2.1 ADDERS

Adders are the elementary building blocks in Digital Design and are indispensable part of digital signal processing applications. Numerous advanced blocks like subtractor, multiplier, divider, and address calculator are obtained from Adders. As Addition is the base of all these operations. First three adders discussed here are single bit adders as they can perform addition of single bit numbers only therefore they are called single bit adders. After which only important parallel N bit adders namely Carry Save Adder (CSA), Carry Increment Adder (CIA), Ripple Carry Adder (RCA) are discussed in this section. Apart from this other parallel adder topologies are also available but we have not included them because their delay is higher than Ripple Carry Adder and there area is also same or higher [6].

Table 2.1 Performance Comparison of Various Adders for 8 bit application

S.No.	Design	Area (LUT's)	Area (Slices)	Delay (ns)
1.	Ripple Carry Adder (RCA)	8	5	2.191
2.	Carry Skip Adder (CSkA)	8	6	2.267
3.	Carry Increment Adder (CIA)	8	5	1.907
4.	Carry Look Ahead Adder (CLAA)	10	5	2.266
5.	Carry Save Adder (CSA)	13	9	1.433
6.	Carry Select Adder (CS*l*A)	8	5	2.588
7.	Carry Bypass Adder (CBA)	12	6	3.160

Table 2.2 Performance Comparison of Carry Save Adder and Carry Increment adder for 16 bit application

S.No.	Design	Area (Slices)	Delay (ns)
1	Carry Increment Adder (CIA)	22	14.32
2	Carry Save Adder (CSA)	23	19.8

It is found that for 8 bit addition applications Carry Save Adder provides the least delay at cost of increase in area by roughly 50% whereas Carry Increment adder provides good speed without compromising with area. Whereas for 16 bit addition applications Carry Increment adder is better than Carry Save Adder [5]. The same results can be verified from the tables given above. Table 2.1 shows performance comparison of various adders for 8 bit application [6]. While performance comparison of Carry Save Adder and Carry Increment adder for 16 bit application [5] is described in Table 2.2.

2.1.1 Half adder

The half adder adds two single binary digits A and B. It has two outputs, sum (S) and carry (C). The carry signal represents an overflow into the next digit of a multi-digit addition. The simplest half-adder design incorporates an XOR gate for S and an AND gate for C. The half adder adds two input bits and generates a carry and sum. The logic diagram and truth table for the half adder are shown in figure 2.1. The characteristic equations for half adder are as follows:

$$\text{Sum} = A \oplus B;$$

$$\text{Carry} = A.B$$

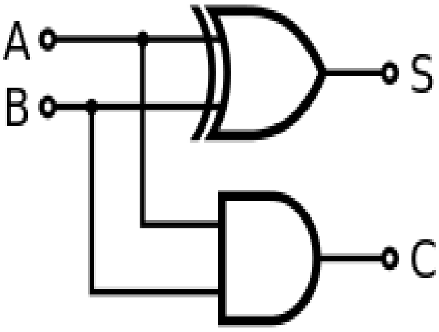


Fig. 2.1: Half Adder

Inputs		Outputs	
A	B	C	S
0	0	0	0
1	0	0	1
0	1	0	1
1	1	1	0

Table 2.3: Half Adder Truth Table

2.1.2 Full adder

A full adder adds three one-bit numbers, often written as A, B, and C_{in}; A and B are the operands, and C_{in} is a bit carried in from the previous less significant stage. The circuit produces a two-bit output, output carry and sum typically represented by the signals C_{out} and S. Here P and G are internal signals termed as propagate and generate signal respectively. The logic diagram and truth table for the full adder are shown in figure 2.2.The characteristic equations for traditional full adder are as follows:

Propagate Signal	$P = A \oplus B$;
Generate Signal	$G = A.B$;
Carry Out	$C_{out} = A.B + (A \oplus B).(C_{in})$;
Sum	$S = A \oplus B \oplus C$

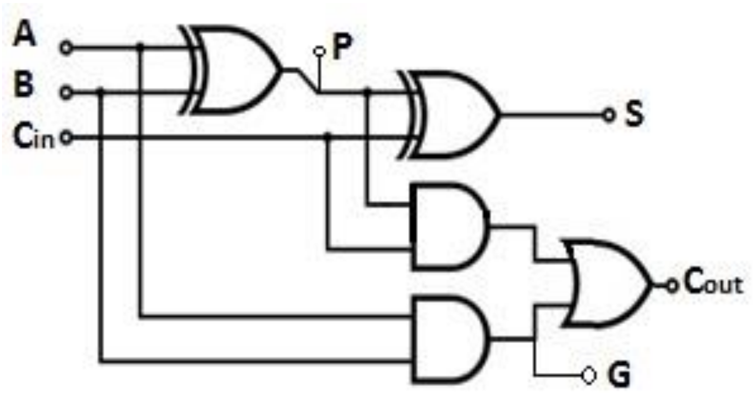


Fig. 2.2: Full Adder

Inputs			Outputs			
A	B	C_{in}	G	P	C_{out}	S
0	0	0	0	0	0	0
0	0	1	0	0	0	1
0	1	0	0	1	0	1
0	1	1	0	1	1	0
1	0	0	0	1	0	1
1	0	1	0	1	1	0
1	1	0	1	0	1	0
1	1	1	1	0	1	1

Table 2.4: Full Adder Truth Table

2.1.3 Logically Optimized Full adder

In [5] R.Uma and P.Dhavachelvan proposed a logically optimized full adder. This adder incorporates two XOR gates and one 2X1 Multiplexer. They have simulated 20 different Boolean expressions for the full adder operation. The performance of all the full adders has been analysed in terms of delay, transistor count and power dissipation. It is observed that adder designed with XOR and MUX has the least delay, transistor count and power dissipation when compared to other combinations of gate. So the adder realized with MUX and XOR is considered to be the optimized adder in terms of delay, transistor count and power dissipation. The logic diagram for this full adder is shown in figure 2.3.The characteristic equations for Logically Optimized Full adder are as follows:

Sum $S = A \oplus B \oplus C;$

Carry Out $C_{out} = \overline{(A \oplus B)}\,.B + (A \oplus B).C$

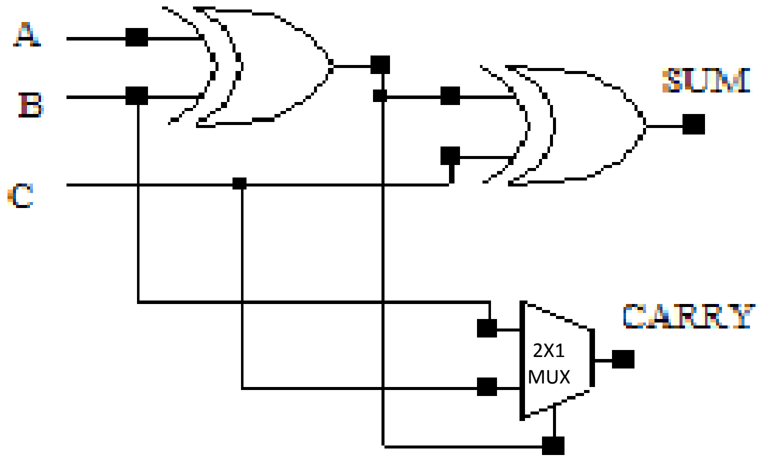
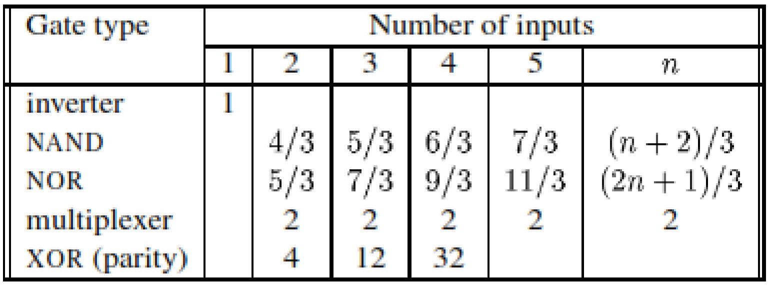


Fig. 2.3: Logically Optimized Full Adder

Reason: This Full Adder architecture uses 2X1 Multiplexer for carry computation instead of two AND and one OR gate which optimizes adder in terms of delay, transistor count and power dissipation. Because logical effort of Multiplexer is 2 while logical effort of replaced carry circuit is higher.

The logical effort of a logic gate tells how much worse it is at producing output current than is an inverter, given that each of its inputs may contain only the same input capacitance as the inverter. Reduced output current means slower operation, and thus the logical effort number for a logic gate tells how much more slowly it will drive a load than an inverter would. Equivalently, logical effort is how much more input capacitance a gate presents to deliver the same output current as an inverter [1].



Gate type	Number of inputs					
	1	2	3	4	5	n
inverter	1					
NAND		4/3	5/3	6/3	7/3	$(n+2)/3$
NOR		5/3	7/3	9/3	11/3	$(2n+1)/3$
multiplexer		2	2	2	2	2
XOR (parity)		4	12	32		

Table 2.5: Logical effort for inputs of static CMOS gates

It is interesting but not surprising to note from Table 1.1 that more complex logic functions have larger logical effort. Moreover, the logical effort of most logic gates grows with the number of inputs to the gate. Larger or more complex logic gates will thus exhibit greater delay.

2.1.4 Ripple Carry Adder (RCA)

Half Adders can be used to add two one bit binary numbers. It is also possible to create a logical circuit using multiple full adders to add N-bit binary numbers. Each full adder inputs a **Cin**, which is the **Cout** of the previous adder. This kind of adder is a **Ripple Carry Adder (RCA) [7]**, since each carry bit "ripples" to the next full adder. RCA contains series structure of Full Adders (FA); each FA is used to add two bits along with carry bit. *Only the first full adder can be substituted by a half adder*. The carry generated from each full adder is given to next full adder and so on. Hence, the carry is propagated in a serial computation. Hence, delay is more as the number of bits is increased in RCA. The 8bit RCA is shown in figure 2.4:

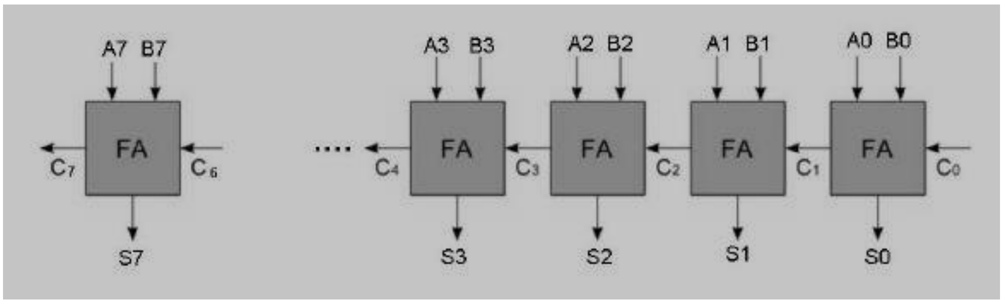


Fig.2.4: Ripple Carry Adder (RCA)

2.1.5 Carry Increment Adder (CIA)

The design of Carry Increment Adder (CIA) consists of RCA's and incremental circuitry [8]. The incremental circuit can be designed using HA's in ripple carry chain with a sequential order. The addition operation is done by dividing total number of bits in to group of 4bits and addition operation is done using several 4bit RCA's. The architecture of CIA is shown in Fig 2.3.

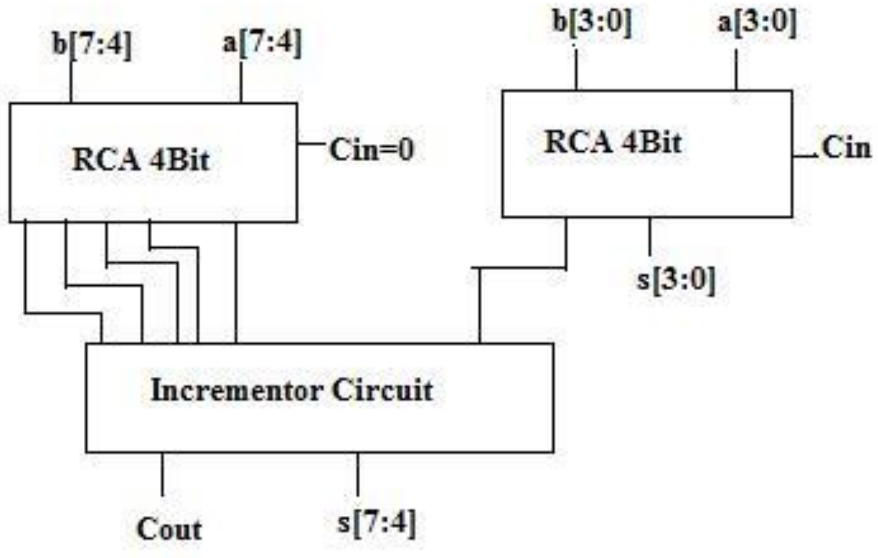


Fig.2.5: Carry Increment Adder (CIA)

2.1.6 Carry Save Adder (CSA)

The carry-save adder reduces the addition of 3 numbers to the addition of 2 numbers. The propagation delay is 3 gates regardless of the number of bits. The carry-save unit consists of n full adders, each of which computes a single sum and carries bit based solely on the corresponding bits of the three input numbers. The entire sum can then be computed by shifting the carry sequence left by one place and appending a 0 to the front (most significant bit) of the partial sum sequence and adding this sequence with RCA produces the resulting n + 1-bit value. This process can be continued indefinitely, adding an input for each stage of full adders, without any intermediate carry propagation. The main application of carry save algorithm is, well known for multiplier architecture is used for efficient CMOS implementation of much wider variety of algorithms for high speed digital signal processing. In this scheme, the carry is not propagated through the stages. Instead, carry is stored in present stage, and updated as addend value in the next stage. Hence, the delay due to the carry is reduced in this scheme. The architecture of CSA is shown in Fig 2.4.

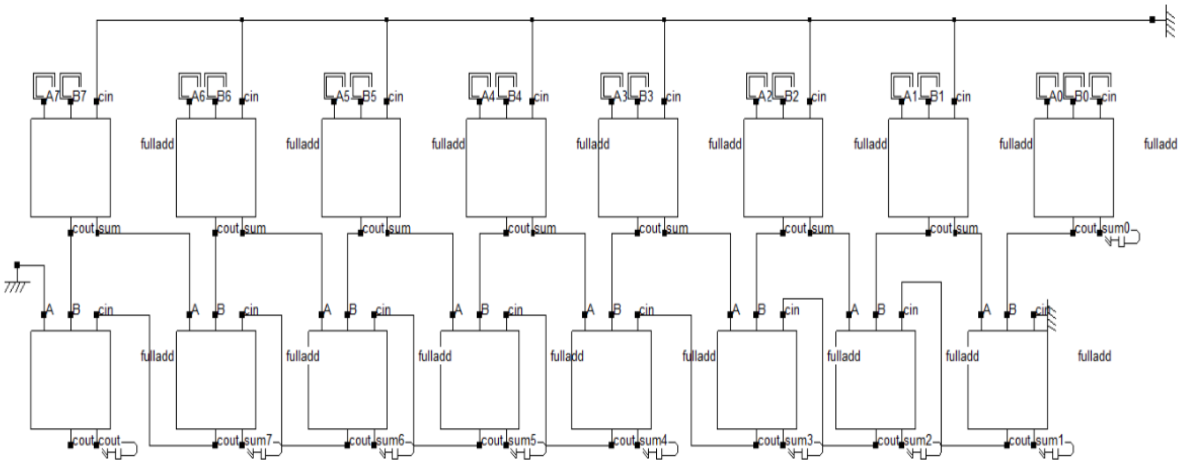


Fig.2.6: Carry Save Adder (CSA)

X		1	0	0	1	1
Y		1	1	0	0	1
S		0	1	0	1	0
C	1	0	0	0	1	
SUM	1	0	1	1	0	0

Table 2.6: Carry Save Adder Computation Flow

2.2 MULTIPLIERS

Multiplication is one of the simple functions which are used in digital signal processing applications. Multipliers need more hardware resources and processing time compared to that of adders. In order to attain the high speed and low power demand, the various multipliers has to design to meet requirements of current VLSI industry needs. Multipliers are not merely used in processor, but also used in other part of processor designs such as various data path units. In general, two numbers such as multiplier and multiplicand are multiplied and generate a product value. All multipliers architectures are built with basic blocks such as Half Adder (HA), Full Adder (FA), and various complex adder architectures. In recent years, many researchers developed several multipliers for the current needs of VLSI industry. Here, a brief description of some traditional multipliers such as Array Multiplier (AM), Ripple Carry Array Multiplier using Row Bypass Technique (RCAM RB), Wallace Tree Multiplier (WTM), Dadda Multiplier (DM) and are discussed. Table 2.7 describes Performance Comparison of Various Multipliers [10] for 8 bit multiplication applications

Table 2.7 Performance Comparison of Various Multipliers

S.No.	Multiplier (8 Bit)	Area (LUT's)	Delay(ns)
1.	Array Multiplier (AM)	79	8.369
2.	Ripple Carry Array Multiplier with Row Bypassing (RCAM RB)	74	6.417
3.	Wallace Tree Multiplier (WTM)	80	6.285
4.	Dadda Multiplier (DM)	86	3.862
5.	Vedic Multiplier (VM)	100	7.406
6.	Modified Radix-2 Booth Multiplier (MRBM)	108	7.627

From above performance comparison table, it is observed that Dadda Multiplier (DM) has optimized performance in terms of Area and Delay.

2.2.1 Array Multiplier (AM)

Array multiplier is one of the basic multiplier which comprises of partial products generated by AND Logic [11]. All partial products are added by the Half Adder (HA) and Full Adder (FA) [9] depending on the number of input bits. Architecture of array multiplier [4] is shown in Fig.2.7.

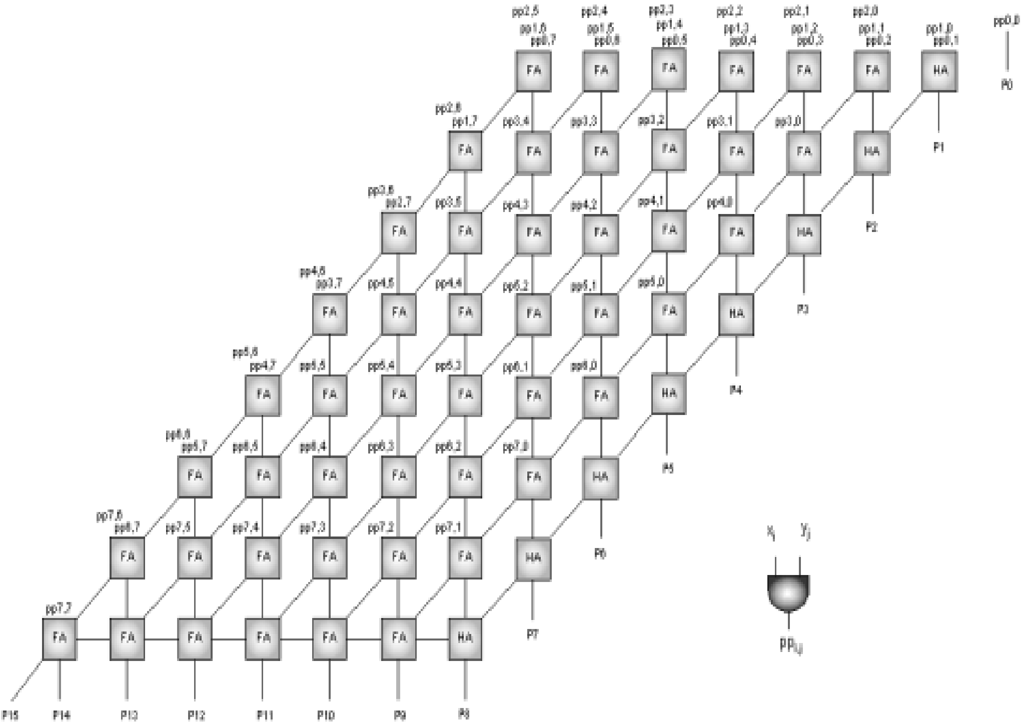
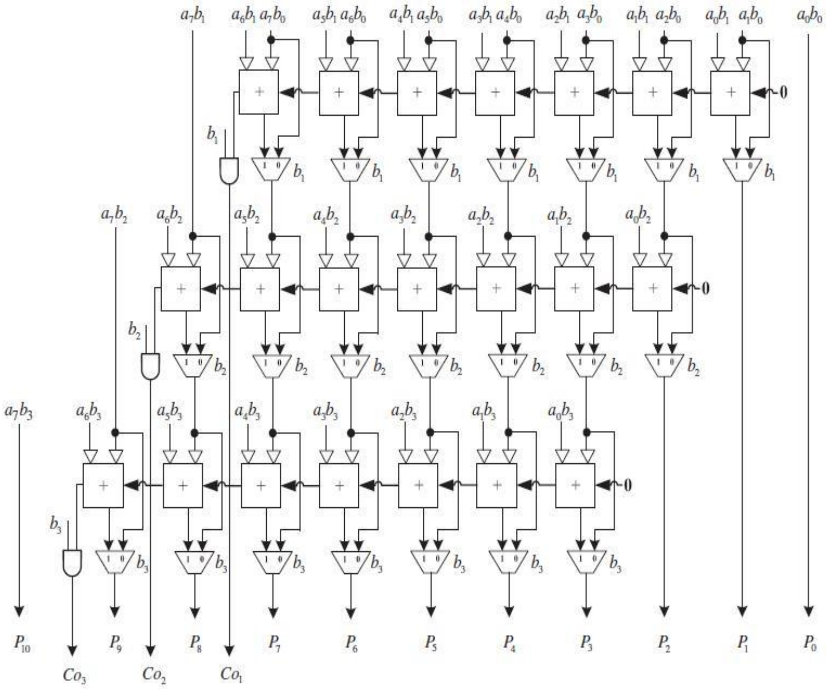


Fig.2.7.Array Multiplier

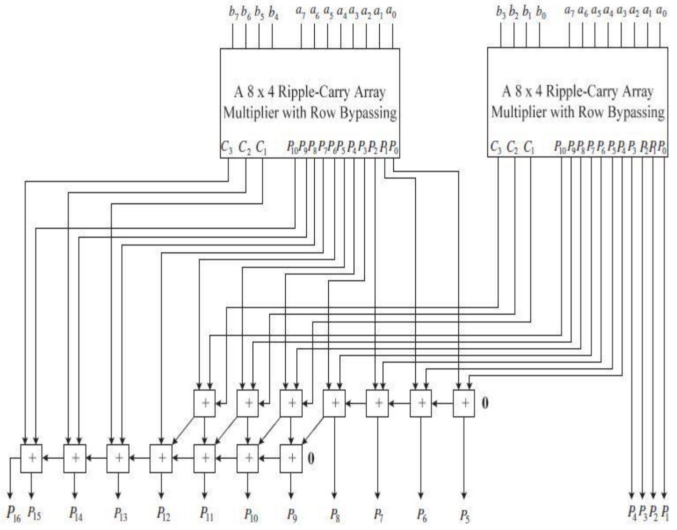
2.2.2 Ripple Carry Array Multiplier with Row Bypassing Technique (RCAM RB)

In ripple carry array multiplier with row bypassing technique [12], the multiplication method is similar to the array multiplier. But the partial product stages are bypassed from previous state to next state depending upon the carry value obtained in adder stage. An 8x8 Multiplier as shown in Fig.2.9 requires two 8x4 RCM multipliers and the architecture is shown in Fig.2.8.



a_7b_1
a_6b_1 a_7b_0
a_5b_1 a_6b_0
a_4b_1 a_5b_0
a_3b_1 a_4b_0
a_2b_1 a_3b_0
a_1b_1 a_2b_0
a_0b_1 a_1b_0
a_0b_0
0
b_1
a_7b_2 a_6b_2 a_5b_2 a_4b_2 a_3b_2 a_2b_2 a_1b_2 a_0b_2
b_2
a_7b_3 a_6b_3 a_5b_3 a_4b_3 a_3b_3 a_2b_3 a_1b_3 a_0b_3
b_3
1 0
P_{10} P_9 P_8 P_7 P_6 P_5 P_4 P_3 P_2 P_1 P_0
Co_3 Co_2 Co_1

Fig.2.8.Structure of 8x4 Ripple Carry Array Multiplier with Row Bypassing



$b_7 b_6 b_5 b_4$ $a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0$
A 8 x 4 Ripple-Carry Array Multiplier with Row Bypassing
C_3 C_2 C_1 $P_{10}P_9P_8P_7P_6P_5P_4P_3P_2P_1P_0$
$b_3 b_2 b_1 b_0$ $a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0$
A 8 x 4 Ripple-Carry Array Multiplier with Row Bypassing
C_3 C_2 C_1 $P_{10}P_9P_8P_7P_6P_5P_4P_3P_2P_1P_0$
0
0
P_{16} P_{15} P_{14} P_{13} P_{12} P_{11} P_{10} P_9 P_8 P_7 P_6 P_5
$P_4P_3P_2P_1$

Fig.2.9.Structure of 8 bit Ripple Carry Array Multiplier (RCM) with Row By passing

2.2.3 Wallace Tree Multiplier (WTM)

In Wallace tree multiplier, the carry save adder scheme is used to add partial products generated in each stage [13]. Hence, carry generated in the present state is saved and added in the next state. Hence the delay due to carry will be reduced in a greater extent. The design of Wallace Tree Multiplier [14] is shown in Fig 2.10.

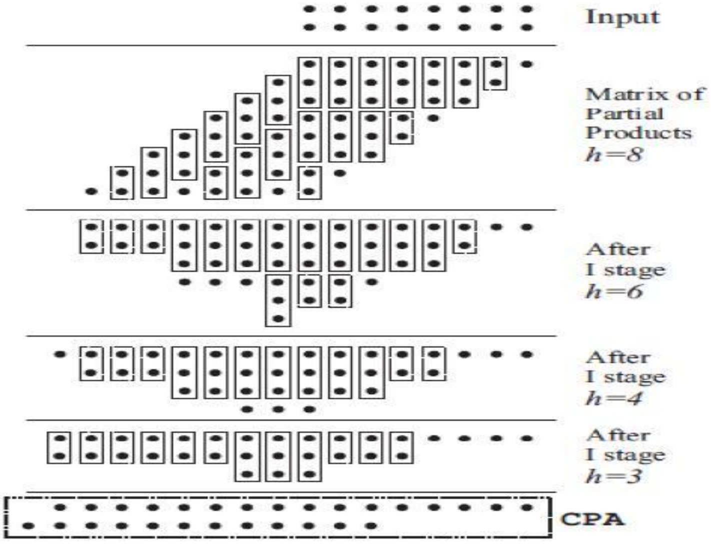


Fig.2.10.Wallace Tree Multiplier (WTM)

2.2.4 Dadda Multiplier (DM)

Dadda multipliers are the refinement of parallel multipliers first presented by Wallace in 1964. In contrast to the Wallace reduction Dadda multiplier perform the least reduction at each stage [15]. The maximum height of each stage is determined by working back from final stage which consists of two rows of partial products. The height of each stage should be in the order 2, 3, 4, 6, 9, 13, 19, 28, 42, 63 etc. An 8 bit Dadda multiplier reduction is shown in Fig 2.11. For Dadda multipliers the required number of full adders and half adders are depend on the value of N. An 8 bit Dadda multiplier reduction is shown in Fig 5. For Dadda multipliers the required number of full adders and half adders are depend on the value of N.

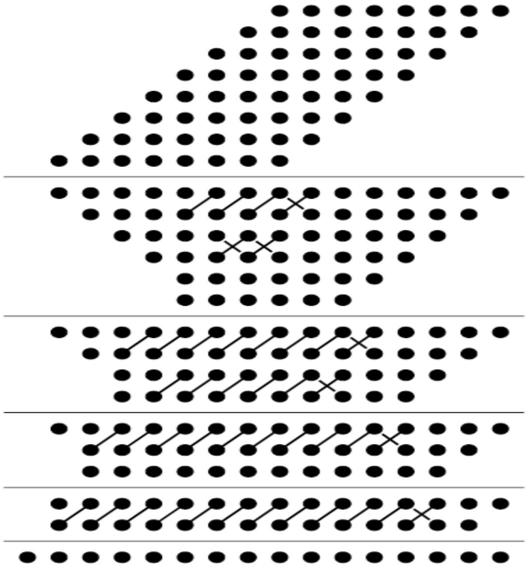
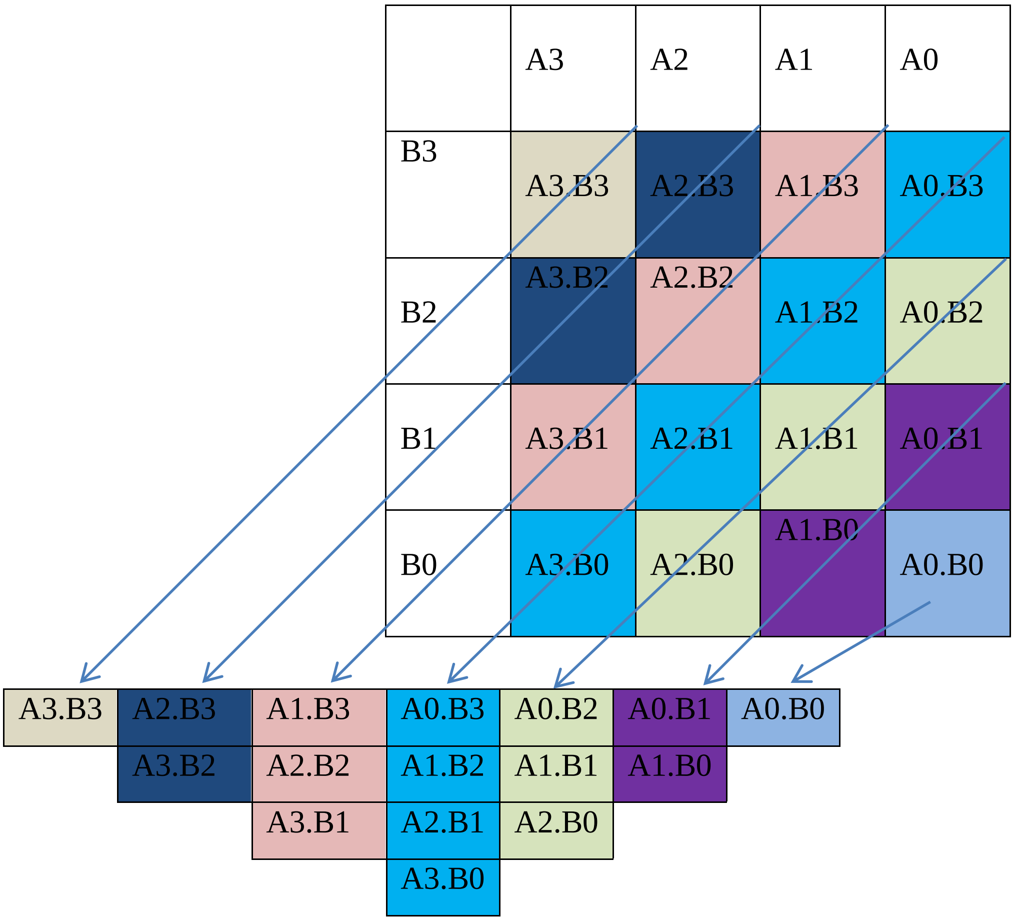


Fig.2.11.Dadda Multiplier Reduction

The principle behind Dadda Multiplier is discussed with the help of 4 bit multiplication example given below. Suppose we have to multiply tow 4 bit numbers A & B then the following algorithm is used in Dadda Multiplier. Figure 2.12 describes the algorithm used by Dadda Multiplier.



	A3	A2	A1	A0
B3	A3.B3	A2.B3	A1.B3	A0.B3
B2	A3.B2	A2.B2	A1.B2	A0.B2
B1	A3.B1	A2.B1	A1.B1	A0.B1
B0	A3.B0	A2.B0	A1.B0	A0.B0

A3.B3	A2.B3	A1.B3	A0.B3	A0.B2	A0.B1	A0.B0
	A3.B2	A2.B2	A1.B2	A1.B1	A1.B0	
		A3.B1	A2.B1	A2.B0		
			A3.B0			

Fig.2.12.Dadda Multiplier Algorithm

2.3 EXISTING MULTIPLY ACCUMULATE (MAC) UNIT MODEL

The basic MAC unit contains multiplier, adder and accumulator. A typical n-bit MAC unit contains an n-bit multiplier, 2n-bit adder, and 2n-bit accumulator. In [4] Various MAC unit models are developed by replacing the multiplier unit with various architectures. And they had found that MAC unit model, with multiplier unit is Dadda multiplier and adder as by Carry Save Adder to be optimized. The internal hardware architecture of existing MAC unit model is given in Fig 2.12.

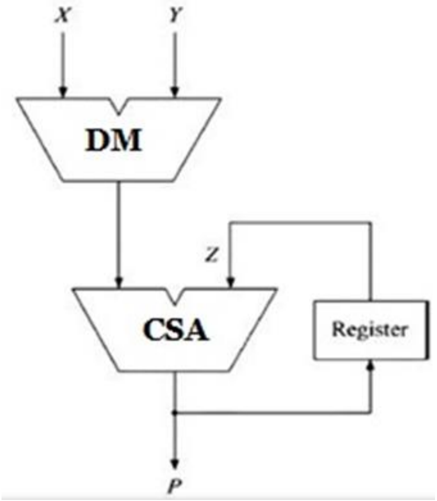


Fig.2.12.Existing MAC unit model (DM + CSA + Accumulator)

2.3.1 DRAWBACKS IN EXISTING MAC UNIT MODEL

- In existing MAC unit Carry Save adder is used while for 16 bit application Carry Increment adder is a better option
- Traditional Full adders are used while logically optimized full adder proposed by R.Uma and P.Dhavachelvan [5] is a better option.
- Half Adders can be replaced by Full adder at some appropriate instances.

2.4 CONCLUSION

After thoroughly studying various research work we have reached to following conclusion

- Use of Half adder at possible instances could minimize the area, delay & power.
- Use of logically optimized full adder instead of Traditional Full adders at all possible instances could minimize the area, delay & power as well [5].
- Use of Carry Save adder instead of for 16 bit application Carry Increment adder could minimize the area, delay & power as well [5].
- Dadda Multiplier is best suitable for 8-bit MAC unit compared to other general purpose multiplier Architectures [4].

This table summarizes the previously done research work in similar domain.

S. No.	**Title /Year**	**Author**	**Proposed work**
1	A New VLSI Architecture of MAC Based on Radix-2 Modified Booth Algorithm 2010	Young-Ho Seo & Dong-Wook Kim	They designed MAC unit model (MRBM+CSA) and found that it achieves better speed without compromising with area [3]. They performed pipelining for speed optimization.
2	32-BIT MAC UNIT DESIGN USING VEDIC MULTIPLIER 2013	Vaijyanath Kunchigi, Linganagouda Kulkarni, Subhash Kulkarni	They designed Multiply and Accumulate (MAC) unit design using Vedic Multiplier with emphasis on efficieny [17]. Vedic Multiplier is based on Urdhva Tiryagbhyam Sutra.
3	Design and Performance Analysis of MAC Unit 2014	Maroju SaiKumar, D.Ashok Kumar, Dr.P.Samundiswary	They designed various MAC unit model and found that MAC unit model (DM+CSA) achieves better performance in terms of area and delay compared to that of existing model [4]. However, there is a slight increase in the power. But they didn't performed pipelining for speed optimization.
4	**This Dissertation**	M. Rizwan Uddin Shaikh	MAC unit model (DM+CIA) is proposed. We are expecting higher speed from previous work [4].

Table 2.7: Summarized detail of Literature Survey

CHAPTER 3

PROPOSED MODELS

3.1 RCA PROPOSED MODEL

RCA proposed model is developed by incorporating logically optimized full adder in the traditional RCA architecture. The full adder at the least significant place is also replaced by Half Adder.

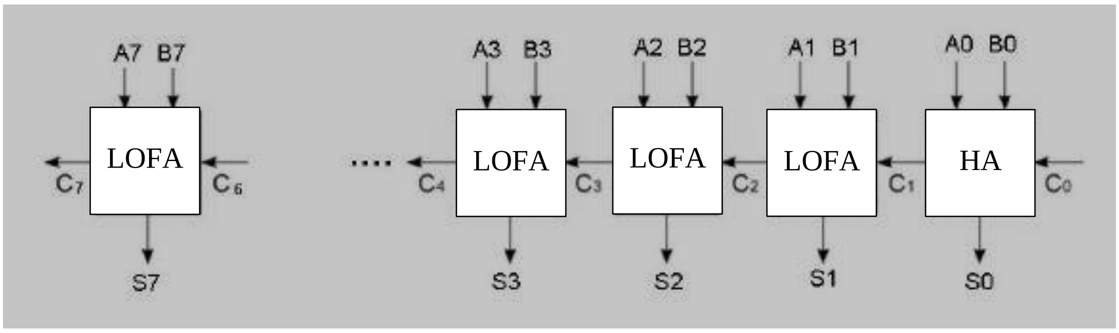


Fig.3.1: Proposed Ripple Carry Adder (RCA) 8 bit Architecture

Only the first full adder can be substituted by a half adder because when we are adding least significant bit of two numbers carry in will be always zero. Also Traditional Full adders are replaced by logically optimized full adder to minimize the area, delay & power.

3.2 CIA PROPOSED MODEL

CIA proposed model is developed by incorporating Proposed Ripple Carry Adder in the traditional CIA architecture. The first stages of carry increment adder consist of a number of RCAs therefore we can replace these traditional RCAs with the proposed RCA architecture.

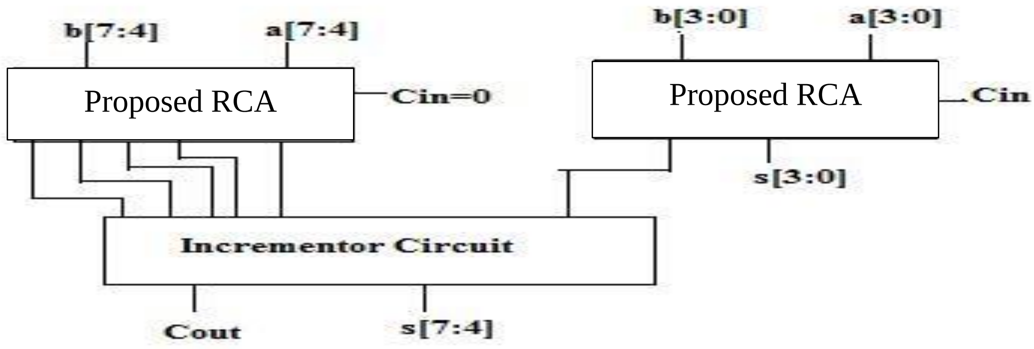


Fig.3.2: Proposed Carry Increment Adder (CIA) Architecture

3.3 CSA PROPOSED MODEL

CSA proposed model is developed by incorporating logically optimized full adder in the traditional RCA architecture.

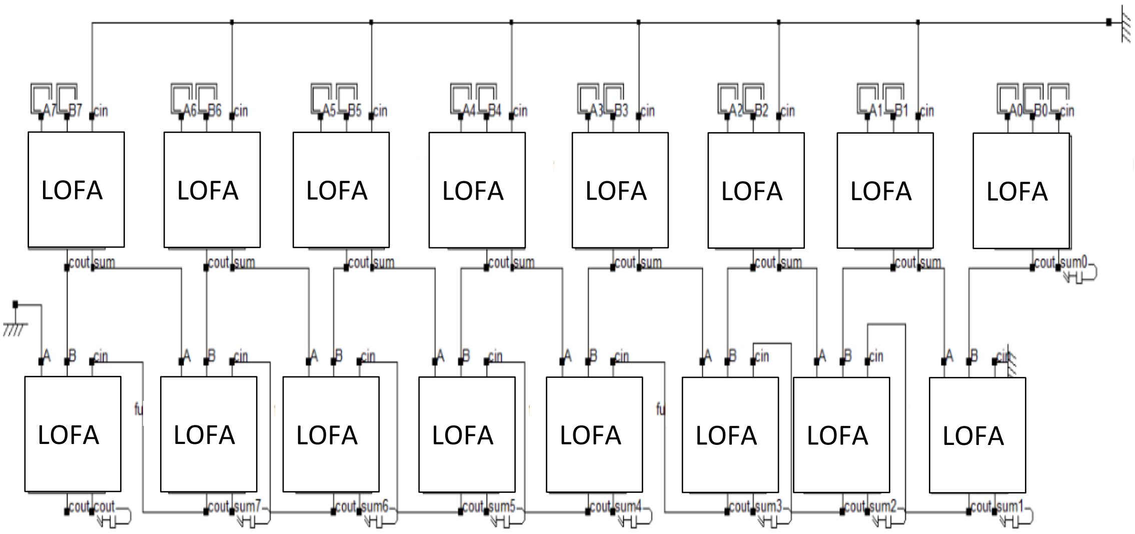


Fig.3.3: Proposed Carry Save Adder (CSA) Architecture

3.4 DADDA MULTIPLIER PROPOSED MODEL

DM proposed model is developed by replacing all traditional full adders with logically optimized full adders in the existing DM architecture. Half Adder is used at instances where we have to calculate addition of two bits only in order to minimize the area, delay & power.

3.5 MAC UNIT PROPOSED MODEL 1

MAC proposed model 1 is developed by incorporating Proposed Carry Save Adder and Proposed Dadda multiplier in the existing MAC architecture.

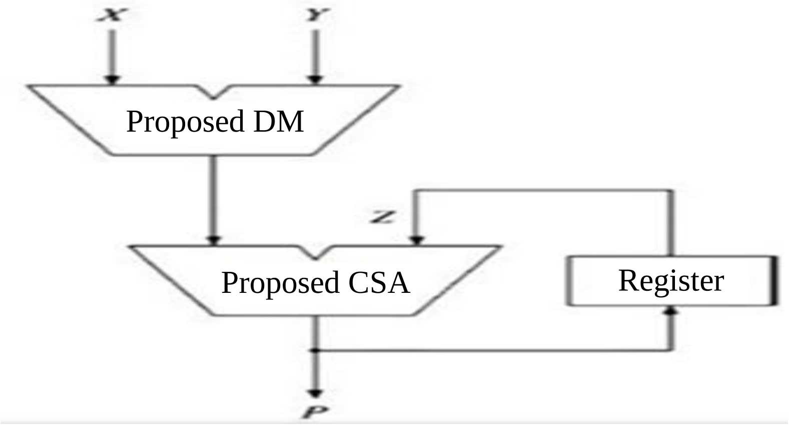


Fig.3.4: MAC Unit proposed model 1

In the existing MAC unit model, with multiplier unit is Dadda multiplier and adder as by Carry Save Adder. Traditional Full Adder is used in both but we found that use of logically optimized full adder could minimize the area, delay & power as well. Therefore we have replaced Traditional Full adders with logically optimized full adder.

3.6 MAC UNIT PROPOSED MODEL 2

MAC proposed model is developed by incorporating Carry Increment Adder in the existing MAC architecture.

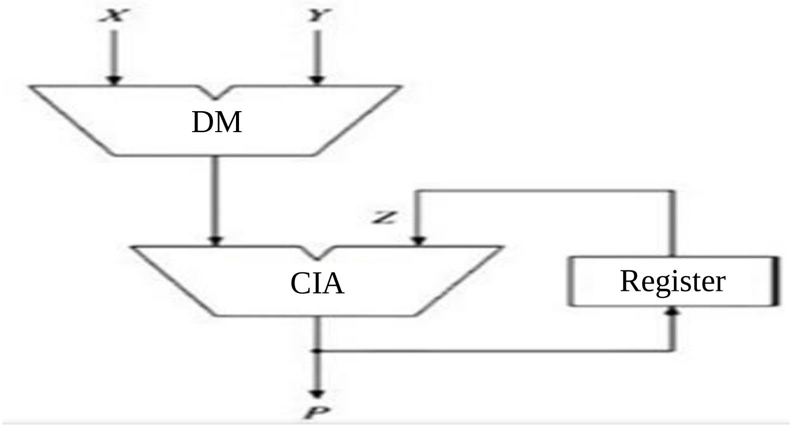


Fig.3.5: MAC Unit proposed model 2

In the existing MAC unit model, with multiplier unit is Dadda multiplier and adder as by Carry Save Adder but we have found that use of Carry Increment adder instead of Carry Save adder for 16 bit application could minimize the area, delay & power as well. Therefore we have replaced Carry Save Adder with Carry Increment adder.

3.7 MAC UNIT PROPOSED MODEL 3

MAC proposed model is developed by incorporating Proposed Carry Increment Adder in the existing MAC architecture.

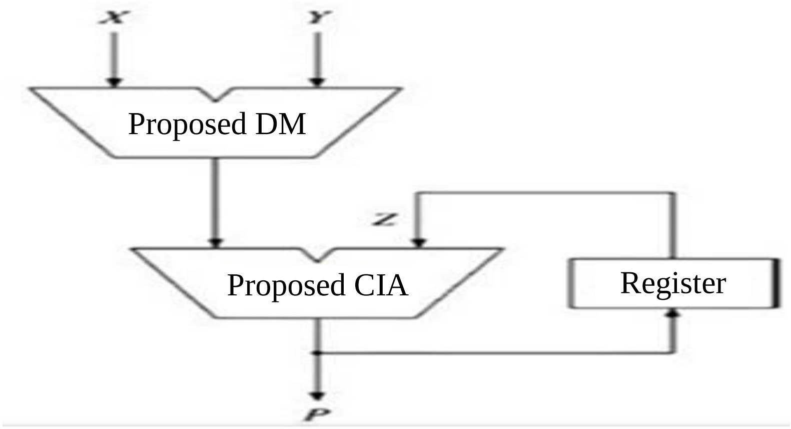


Fig.3.6: MAC Unit proposed model 3

In the existing MAC unit model, with multiplier unit is Dadda multiplier and adder as by Carry Save Adder but we have found that use of Carry Increment adder instead of Carry Save adder for 16 bit application could minimize the area, delay & power as well. Therefore we have replaced Carry Save Adder with Carry Increment adder. We also found that use of logically optimized full adder could minimize the area, delay & power as well. Therefore we have replaced Traditional Full adders with logically optimized full adder.

CHAPTER 4
SIMULATION RESULTS

Various MAC unit models are designed using Verilog HDL. Simulation and synthesis [18] are done using Xilinx ISE 12.2 for Virtex-6 family 40nm technology device. The power is calculated using Lattice Diamond Design suite software. In simulation results, technology view [19] is a schematic representation of the design in terms of logic elements optimized to the target Xilinx device or technology in terms of Look Up Tables (LUTs) [20], carry logic, I/O buffers, and other technology-specific components. A LUT is basically just a small memory. A 4-input, 1-output LUT, can generate any 4-input Boolean function (AND / OR / XOR / NOT / combinations of these / etc.). When FPGA has to be configured, it is required to configure the contents of the LUT, and thus the function will be implemented.

RTL view is a schematic representation of the pre-optimized design in terms of generic symbols that are independent of the targeted Xilinx device in terms of adders, multipliers, counters, AND gates, and OR gates. Timing waveform [19] is generated by writing test bench program which contains the set of input test vectors applied to design.

4.1SIMULATION RESULTS OF ADDERS

4.1.1 Half Adder

Technology view and RTL view of Half Adder are given in Fig 4.1.

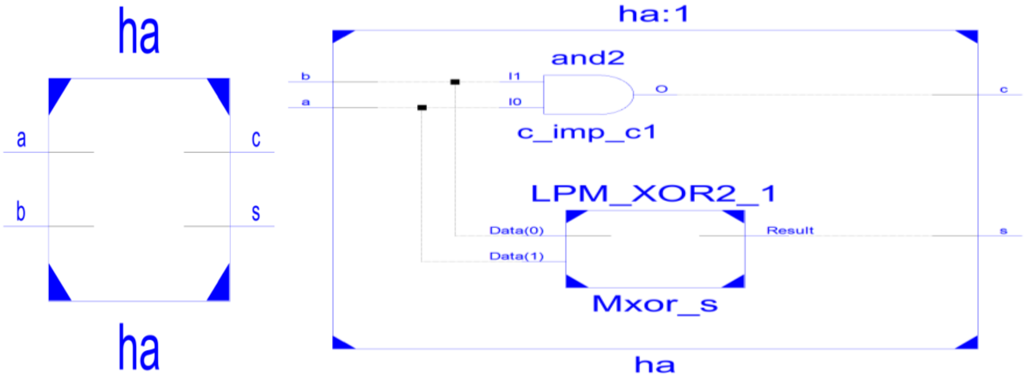


Fig.4.1.Technology View and RTL View of Half Adder

And, the timing waveform of Half Adder is illustrated in Fig 4.2.

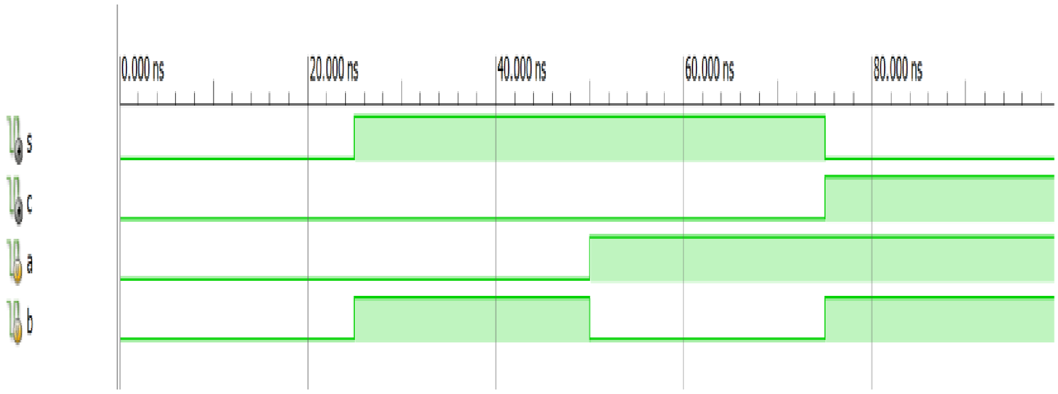


Fig.4.2.Timing Waveform of Half Adder

Table 4.1 Device Utilization Summary for Half Adder

Logic Utilization	**Used**
Number of Slice LUTs	1
Number of occupied Slices	1
Number of bonded IOBs	4
Average Fanout of Non-Clock Nets	4.00
Maximum combinational path delay in nanosecond	0.770
Power in milliwatts (mW)	90.6

4.1.2 Traditional Full Adder

Technology view and RTL view of Traditional Full Adder are given in Fig 4.3.

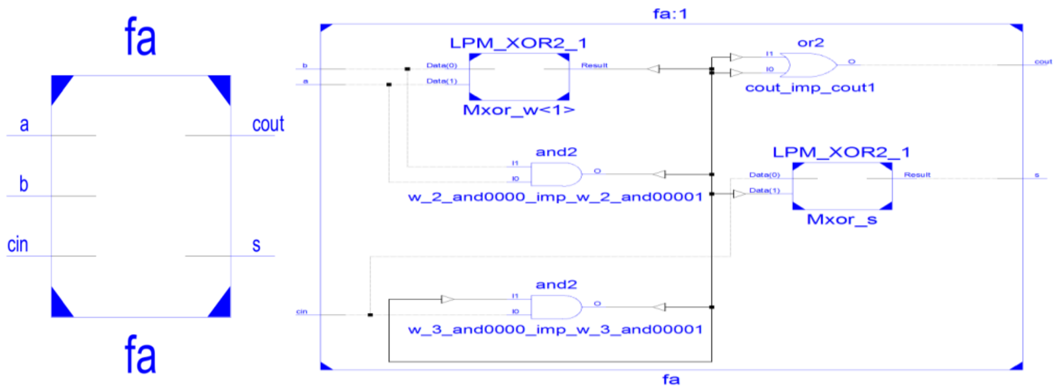


Fig.4.3.Technology View and RTL View of Traditional Full Adder

And, the timing waveform of Traditional Full Adder is illustrated in Fig 4.4.

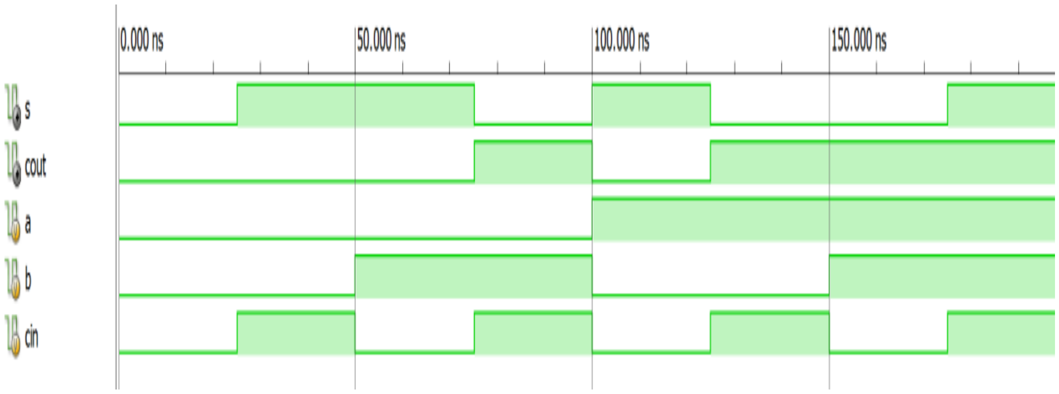


Fig.4.4.Timing Waveform of Traditional Full Adder

Table 4.2 Device Utilization Summary for Traditional Full Adder

Logic Utilization	**Used**
Number of Slice LUTs	1
Number of occupied Slices	1
Number of bonded IOBs	5
Average Fanout of Non-Clock Nets	3.50
Maximum combinational path delay in Nanosecond	0.923
Power in milliwatts (mW)	90.6

4.1.3 Logically Optimized Full Adder

Technology view and RTL view of Logically Optimized Full Adder are given in Fig 4.5.

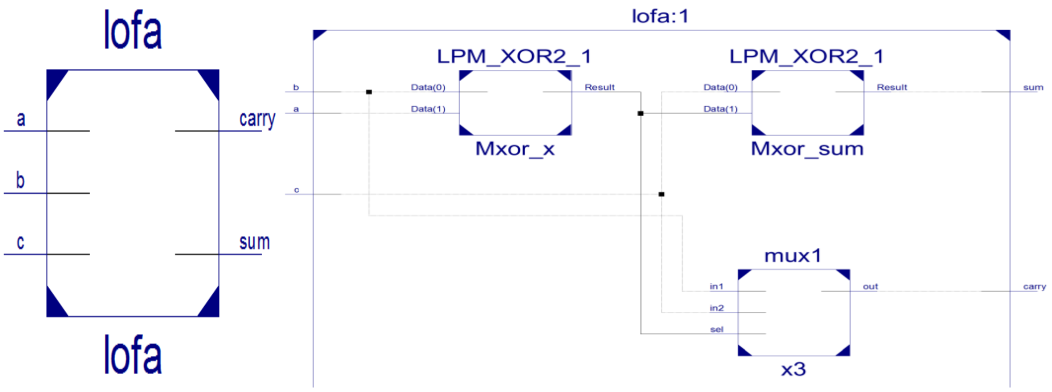


Fig.4.5.Technology View and RTL View of Logically Optimized Full Adder

And, the timing waveform of Logically Optimized Full Adder is illustrated in Fig 4.6.

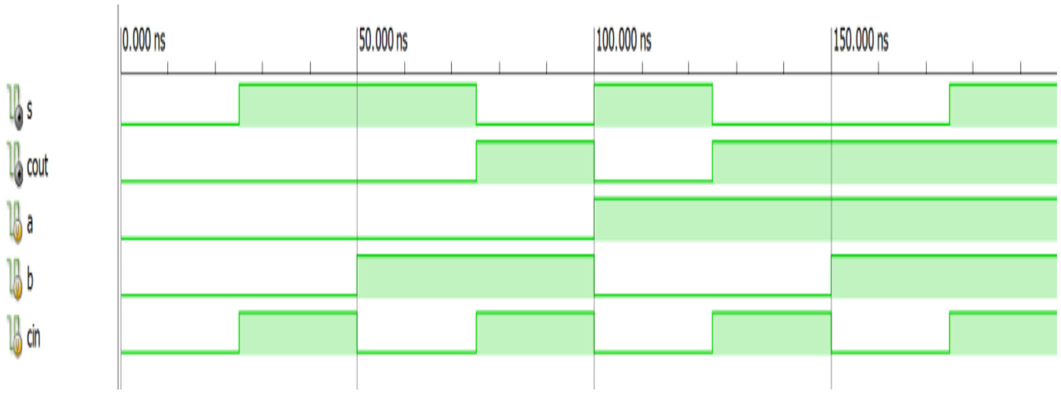


Fig.4.6.Timing Waveform of Logically Optimized Full Adder

Table 4.3 Device Utilization Summary for Logically Optimized Full Adder

Logic Utilization	**Used**
Number of Slice LUTs	1
Number of occupied Slices	1
Number of bonded IOBs	5
Average Fanout of Non-Clock Nets	3.50
Maximum combinational path delay in Nanosecond	0.776
Power in milliwatts (mW)	90.6

4.1.4 Ripple Carry Adder

Technology view and RTL view of Ripple Carry Adder are given in Fig 4.7.

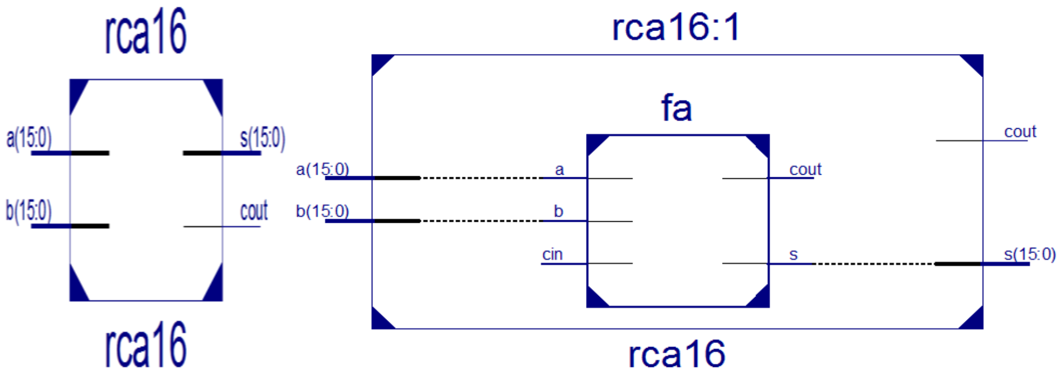


Fig.4.7.Technology View and RTL View of Ripple Carry Adder

And, the timing waveform of Ripple Carry Adder is illustrated in Fig 4.8.

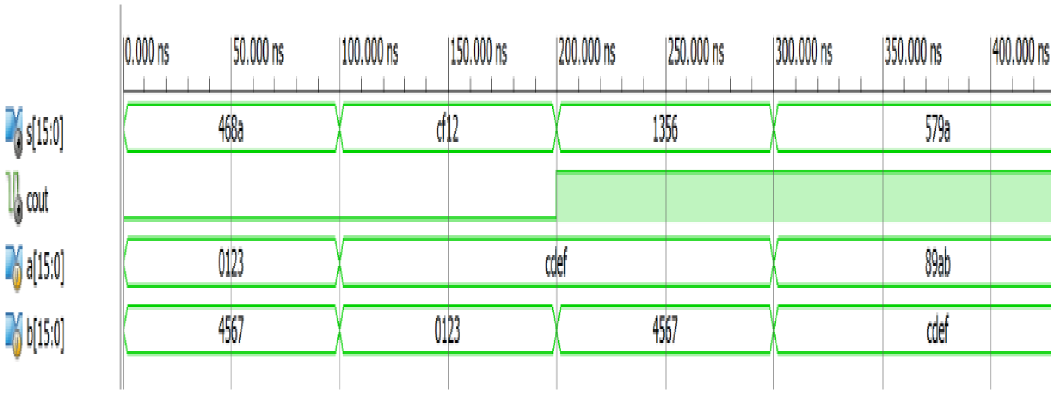


Fig.4.8.Timing Waveform of Ripple Carry Adder

Table 4.4 Device Utilization Summary for Ripple Carry Adder	
Logic Utilization	**Used**
Number of Slice LUTs	16
Number of occupied Slices	12
Number of bonded IOBs	49
Average Fanout of Non-Clock Nets	1.75
Maximum combinational path delay in Nanosecond	4.594
Power in milliwatts (mW)	95.2

4.1.5 Proposed Ripple Carry Adder

Technology view and RTL view of Proposed Ripple Carry Adder are given in Fig 4.9.

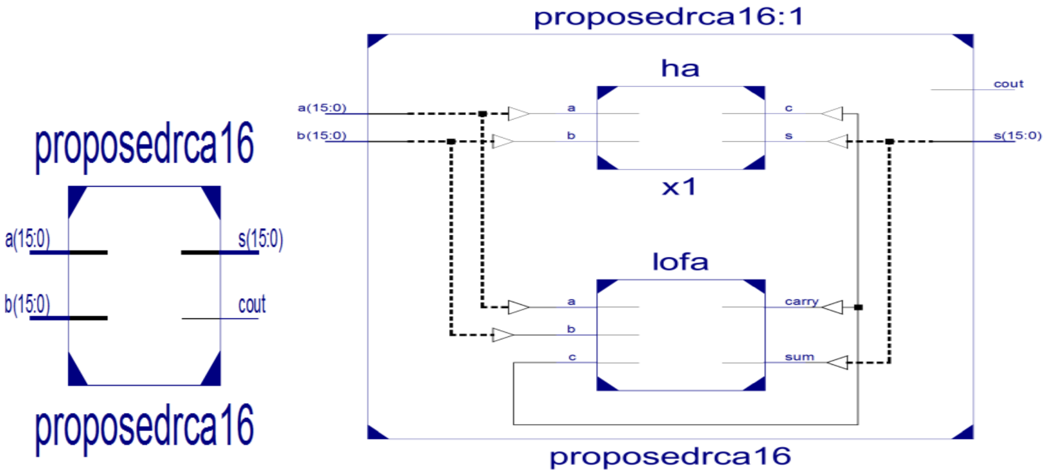


Fig.4.9.Technology View and RTL View Proposed Ripple Carry Adder

And, the timing waveform of Proposed Ripple Carry Adder is illustrated in Fig 4.10.

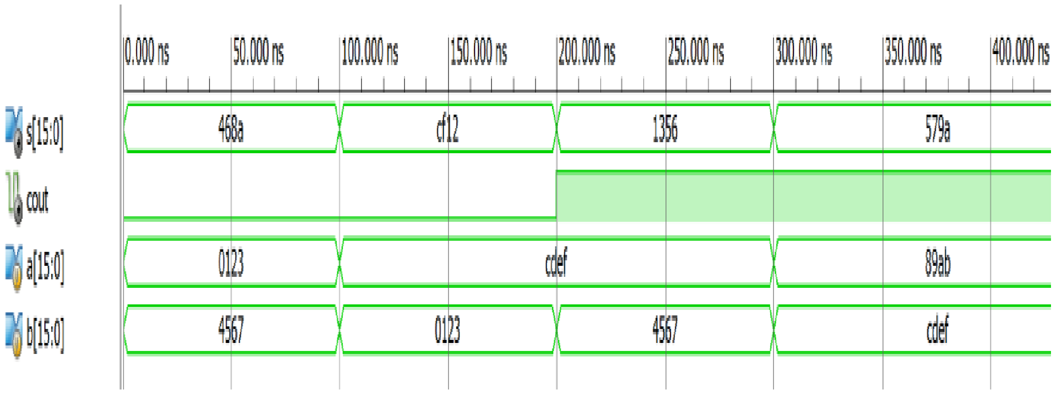
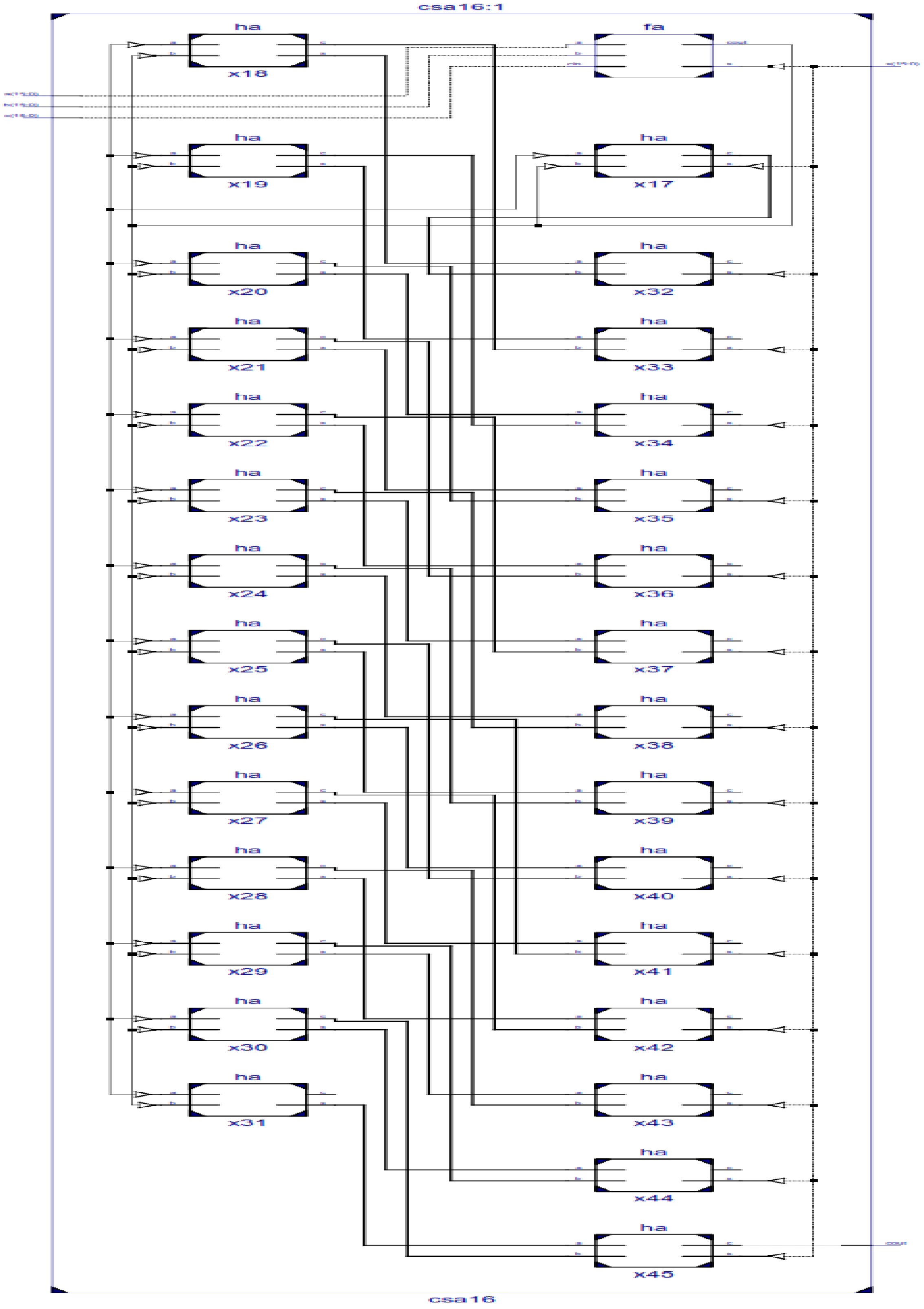


Fig.4.10.Timing Waveform of Proposed Ripple Carry Adder

Table 4.5 Device Utilization Summary for Proposed Ripple Carry Adder	
Logic Utilization	**Used**
Number of Slice LUTs	16
Number of occupied Slices	12
Number of bonded IOBs	49
Average Fanout of Non-Clock Nets	1.75
Maximum combinational path delay in Nanosecond	3.872
Power in milliwatts (mW)	96.8

4.1.6 Carry Save Adder

RTL View and Technology View of Carry Save Adder are given in Fig 4.11.



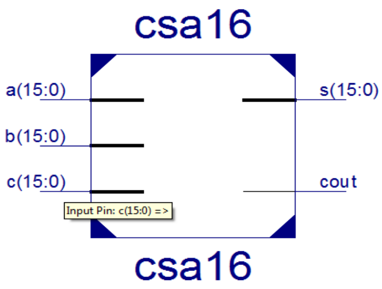


Fig.4.11. RTL View and Technology View of Carry Save Adder

And, the timing waveform of Carry Save Adder is illustrated in Fig 4.12.

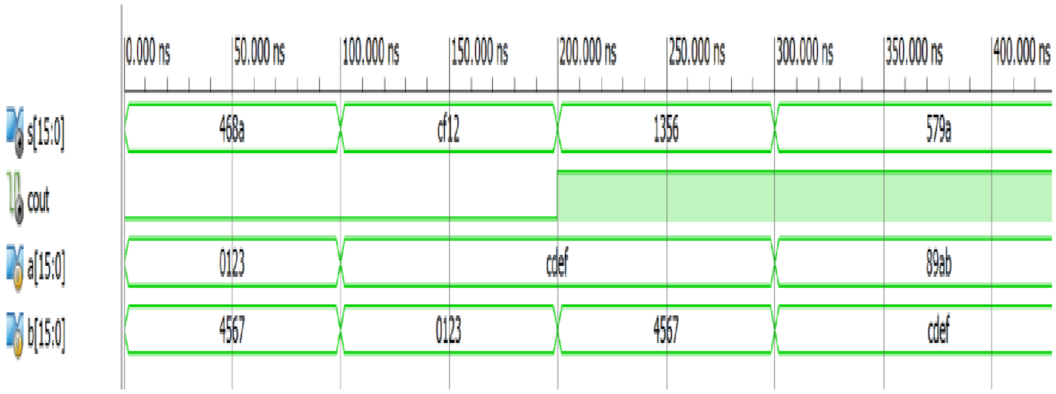
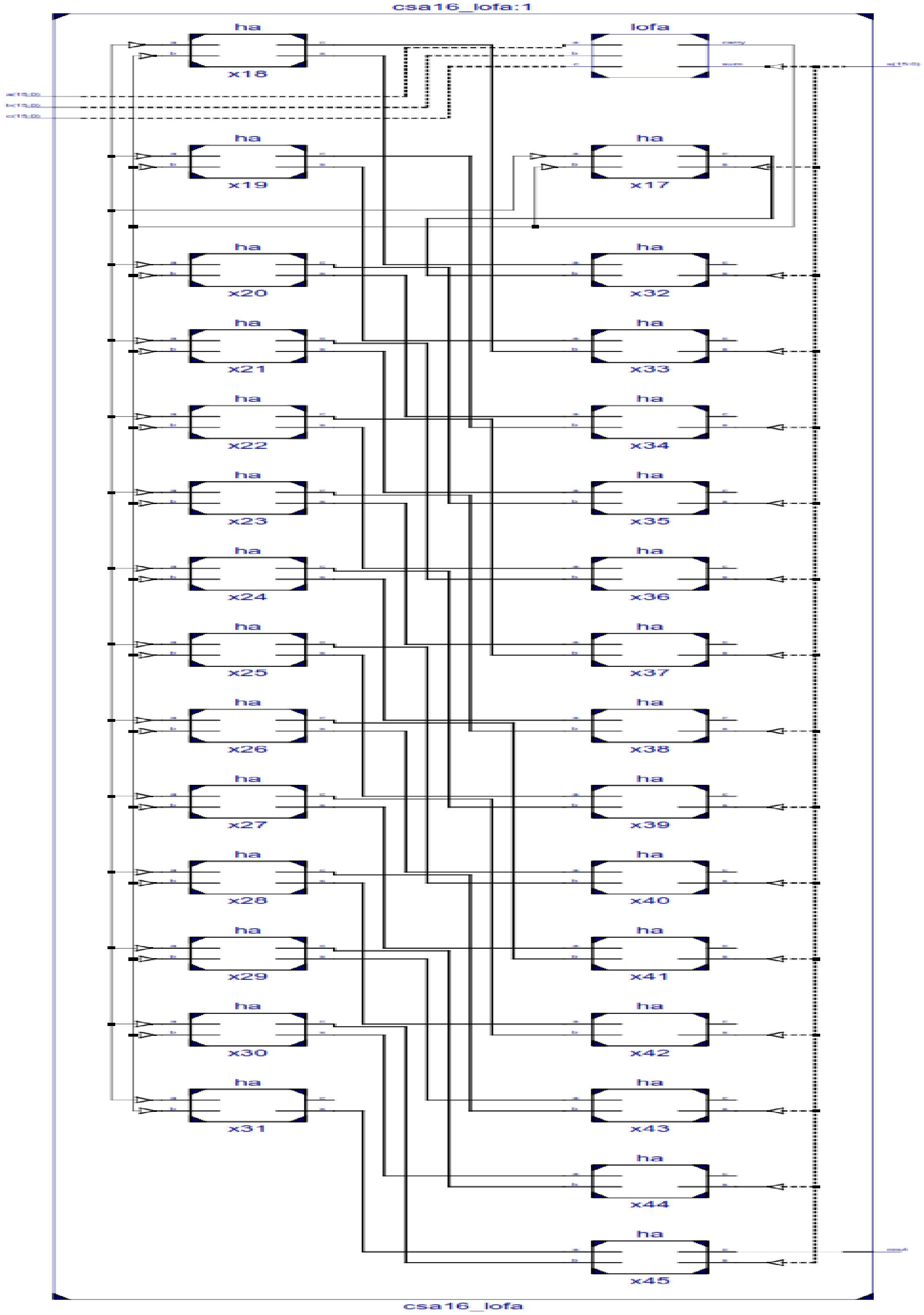


Fig.4.12.Timing Waveform of Carry Save Adder

Table 4.6 Device Utilization Summary for Carry Save Adder	
Logic Utilization	**Used**
Number of Slice LUTs	24
Number of occupied Slices	13
Number of bonded IOBs	65
Average Fanout of Non-Clock Nets	1.67
Maximum combinational path delay in Nanosecond	1.868
Power in milliwatts (mW)	87.5

4.1.7 Proposed Carry Save Adder

RTL View and Technology View of Proposed Carry Save Adder are given in Fig 4.13.



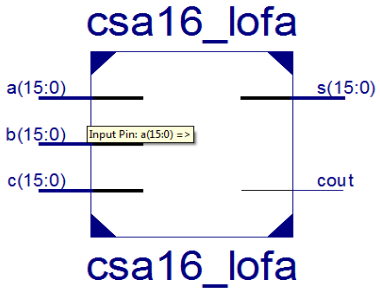


Fig.4.13.RTL View and Technology View of Proposed Carry Save Adder

And, the timing waveform of Proposed Carry Save Adder is illustrated in Fig 4.14.

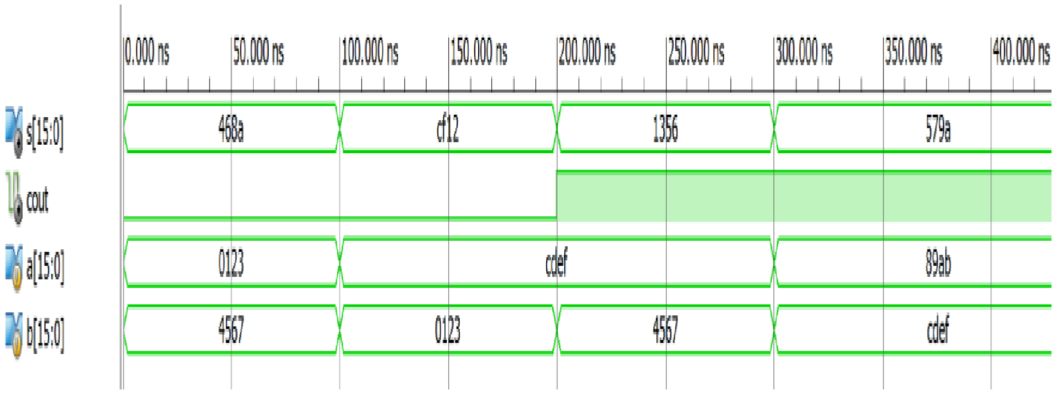


Fig.4.14.Timing Waveform of Proposed Carry Save Adder

Table 4.7 Device Utilization Summary for Proposed Carry Save Adder	
Logic Utilization	**Used**
Number of Slice LUTs	24
Number of occupied Slices	13
Number of bonded IOBs	65
Average Fanout of Non-Clock Nets	1.67
Maximum combinational path delay in Nanosecond	1.586
Power in milliwatts (mW)	90.5

4.1.8 Carry Increment Adder

Technology view and RTL view of Carry Increment Adder are given in Fig 4.15.

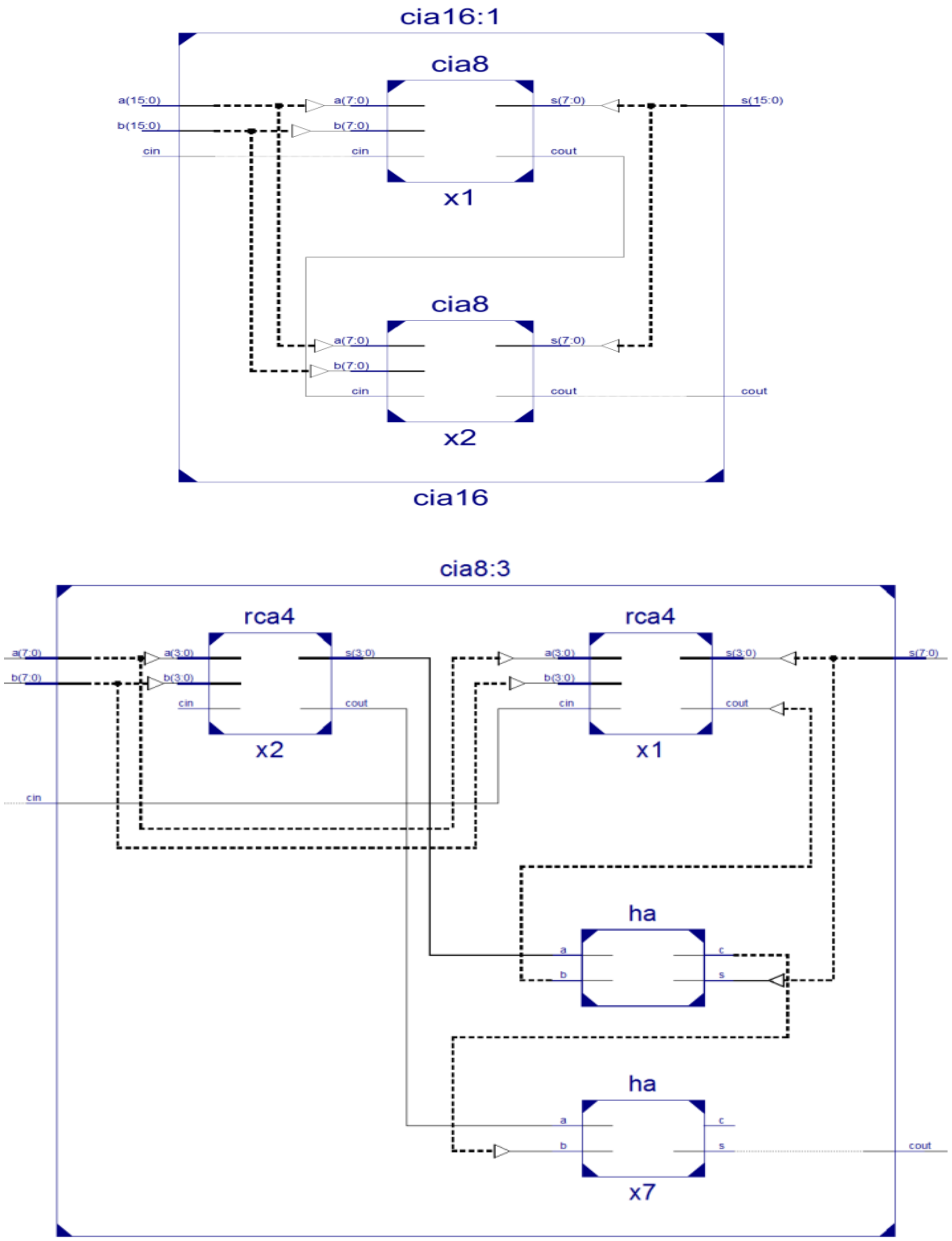


Fig.4.15.Technology View and RTL View of Carry Increment Adder

And, the timing waveform of Carry Increment Adder is illustrated in Fig 4.16.

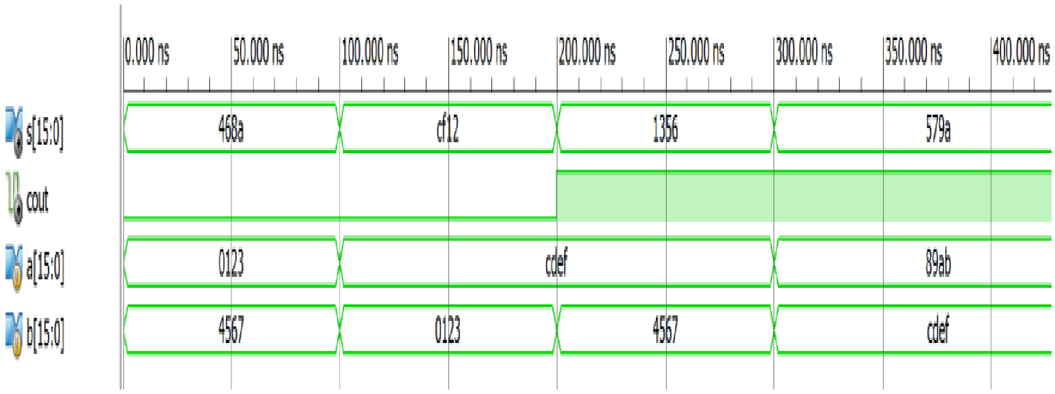


Fig.4.16.Timing Waveform of Carry IncrementAdder

Table 4.8 Device Utilization Summary for Carry Increment Adder	
Logic Utilization	**Used**
Number of Slice LUTs	24
Number of occupied Slices	15
Number of bonded IOBs	50
Average Fanout of Non-Clock Nets	1.75
Maximum combinational path delay in Nanosecond	4.254
Power in milliwatts (mW)	95.2

4.1.9 Proposed Carry Increment Adder

Technology view and RTL view of Proposed Carry Increment Adder are given in Fig 4.17.

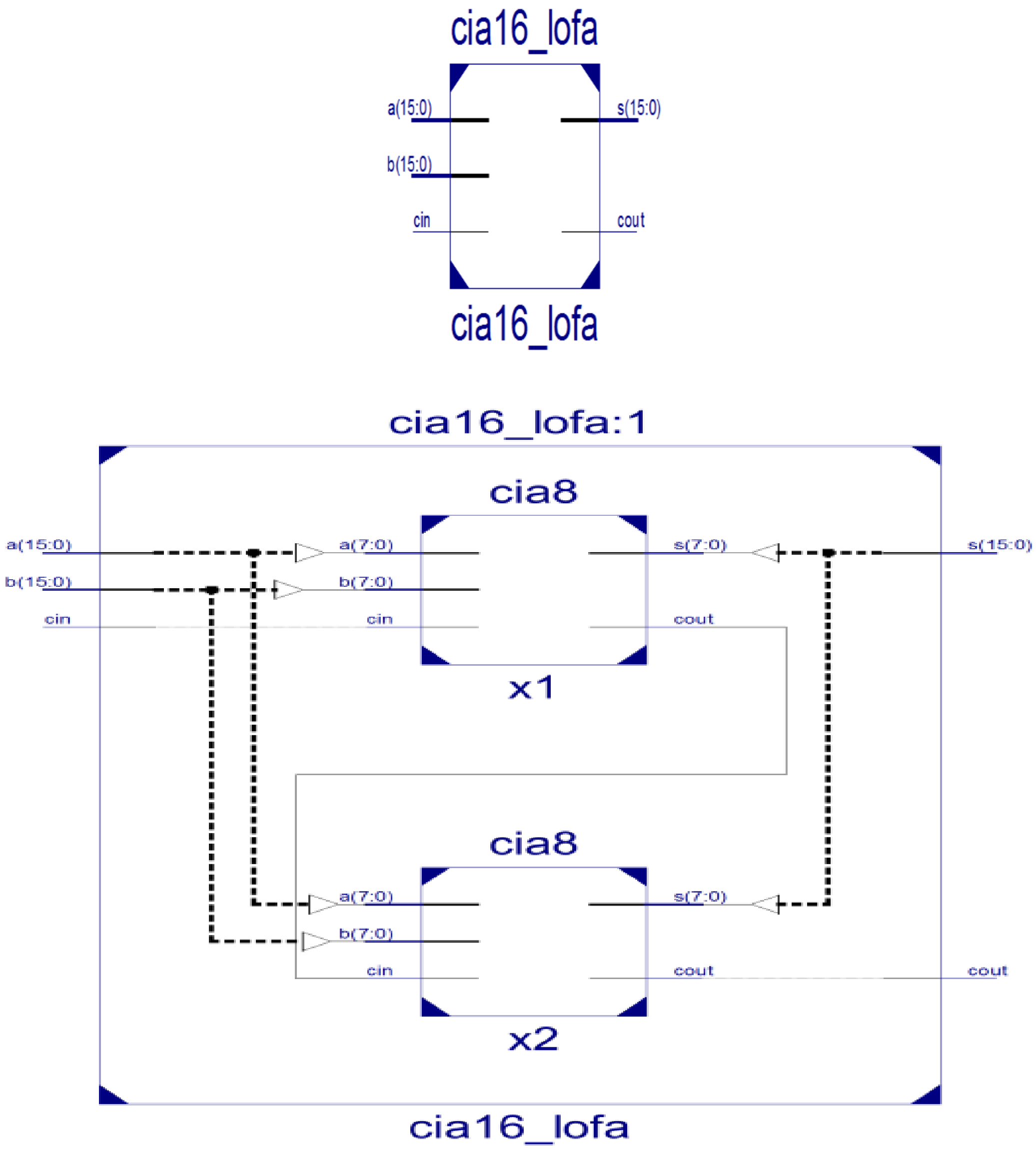


Fig.4.17.Technology View and RTL View Proposed Carry Increment Adder

And, the timing waveform of Proposed Carry Increment Adder is illustrated in Fig 4.18.

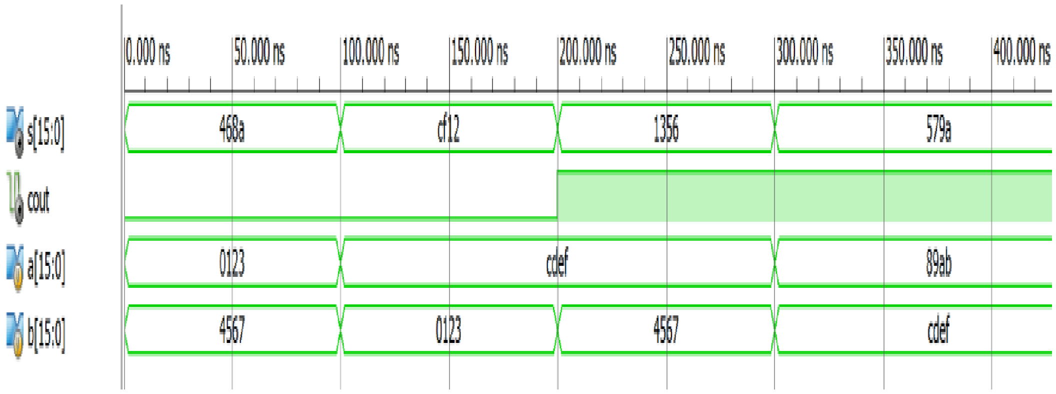


Fig.4.18.Timing Waveform of Proposed Carry Increment Adder

Table 4.9 Device Utilization Summary for Proposed Carry Increment Adder	
Logic Utilization	**Used**
Number of Slice LUTs	24
Number of occupied Slices	15
Number of bonded IOBs	50
Average Fanout of Non-Clock Nets	2.33
Maximum combinational path delay in Nanosecond	3.873
Power in milliwatts (mW)	98.5

4.2 SIMULATION RESULTS OF MULTIPLIERS

4.2.1 Dadda Multiplier (DM)

Technology view and RTL view of Dadda Multiplier are given in Fig 4.19.

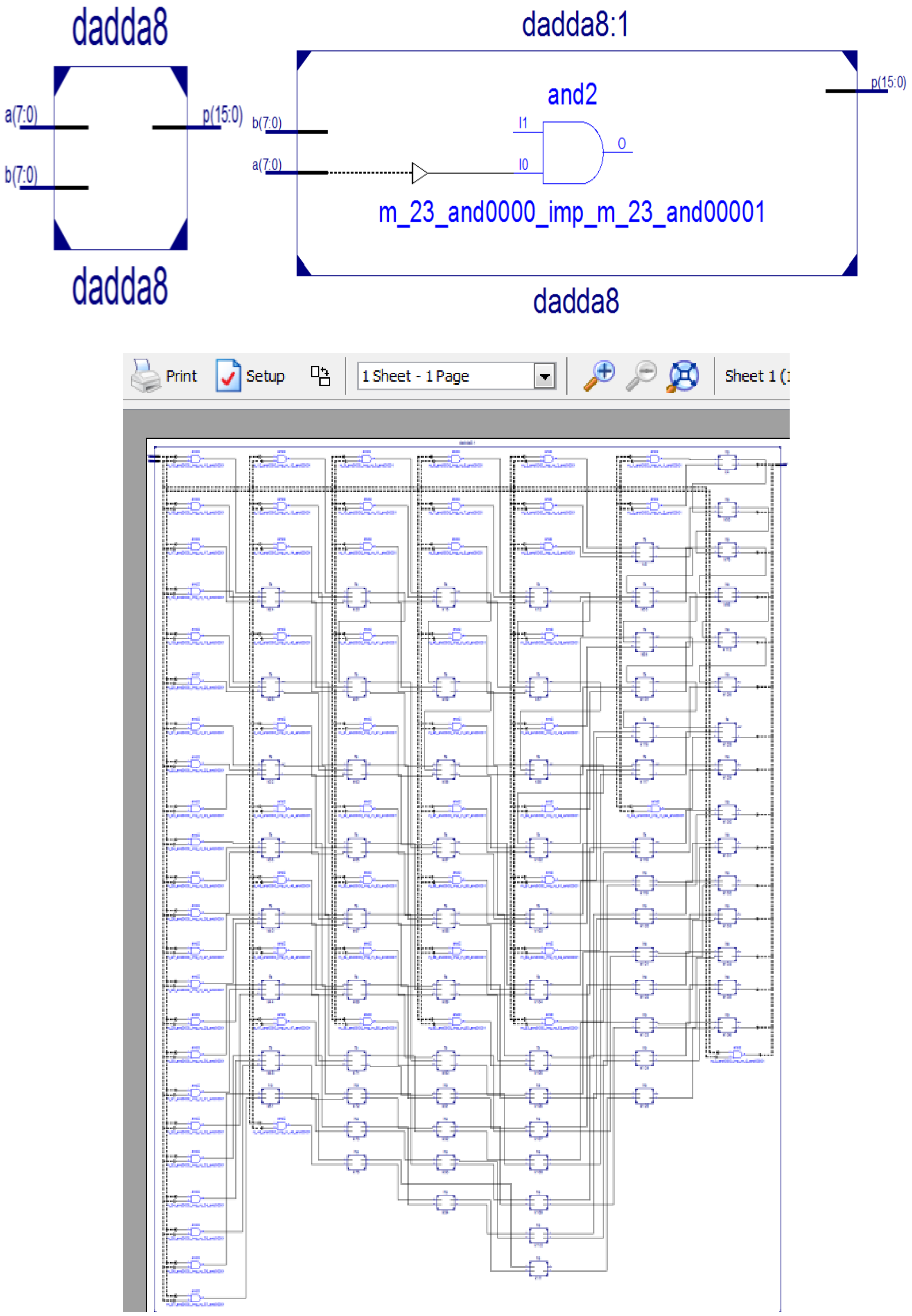


Fig.4.19.Technology View and RTL View of Dadda Multiplier

And, the timing waveform of Dadda Multiplier is illustrated in Fig 4.20.

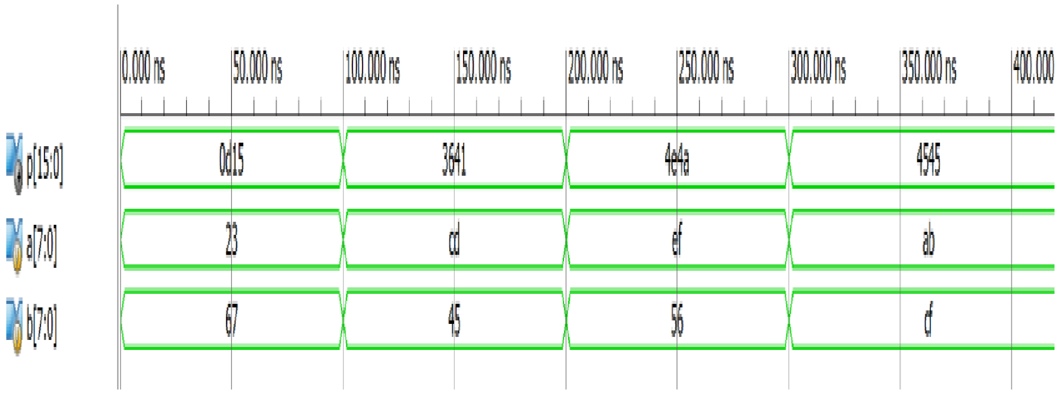


Fig.4.20.Timing Waveform of Dadda Multiplier

Table 4.10 Device Utilization Summary for Dadda Multiplier	
Logic Utilization	**Used**
Number of Slice LUTs	86
Number of occupied Slices	34
Number of bonded IOBs	32
Average Fanout of Non-Clock Nets	3.93
Maximum combinational path delay in Nanosecond	4.395
Power in milliwatts (mW)	96.8

4.2.2 Proposed Dadda Multiplier

Technology view and RTL view of Proposed Dadda Multiplier are given in Fig 4.21.

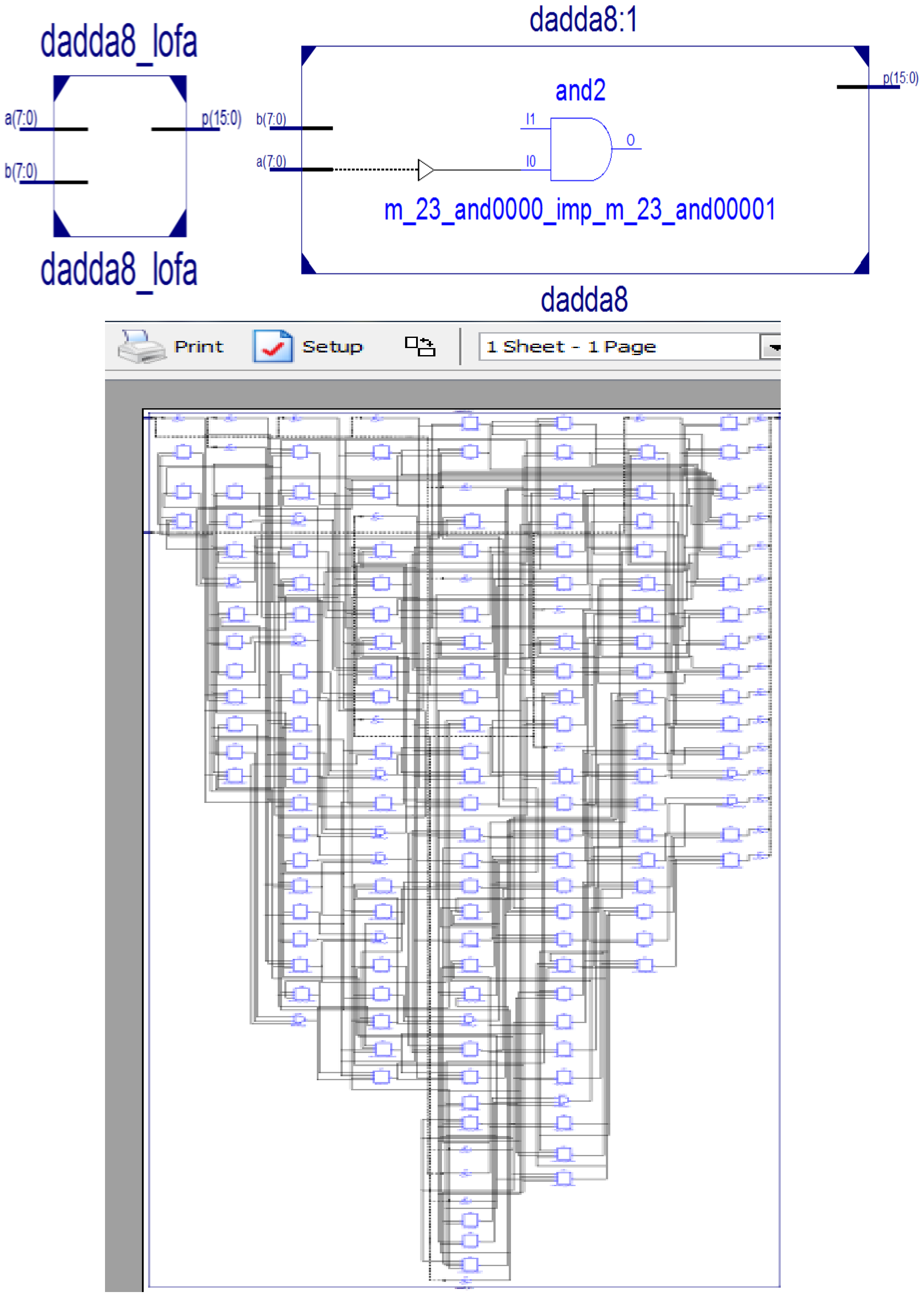


Fig.4.21.Technology View and RTL View of Proposed Dadda Multiplier

And, the timing waveform of Proposed Dadda Multiplier is illustrated in Fig 4.22.

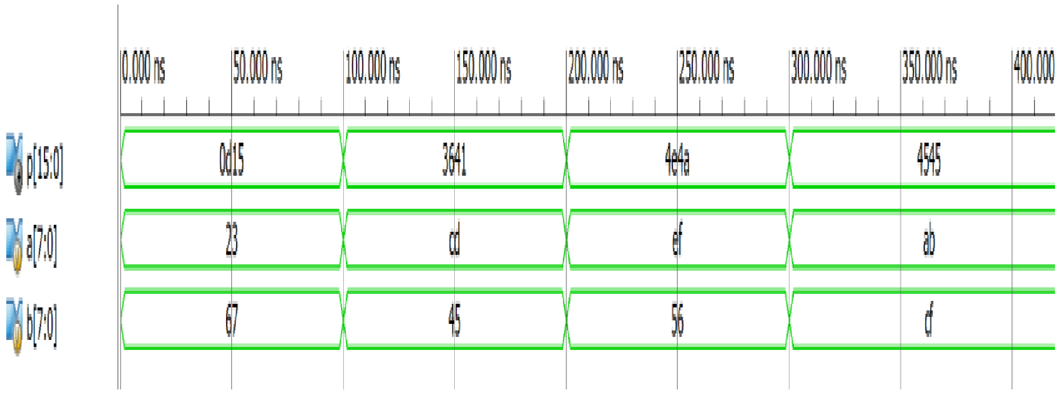


Fig.4.22.Timing Waveform of Proposed Dadda Multiplier

Table 4.11 Device Utilization Summary for Proposed Dadda Multiplier	
Logic Utilization	**Used**
Number of Slice LUTs	86
Number of occupied Slices	34
Number of bonded IOBs	32
Average Fanout of Non-Clock Nets	3.93
Maximum combinational path delay in Nanosecond	3.511
Power in milliwatts (mW)	95.1

4.3 SIMULATION RESULTS OF MAC UNIT

4.3.1 Existing MAC Unit (DM+CSA)

Technology view and RTL view of Existing MAC unit model are given in Fig 4.23.

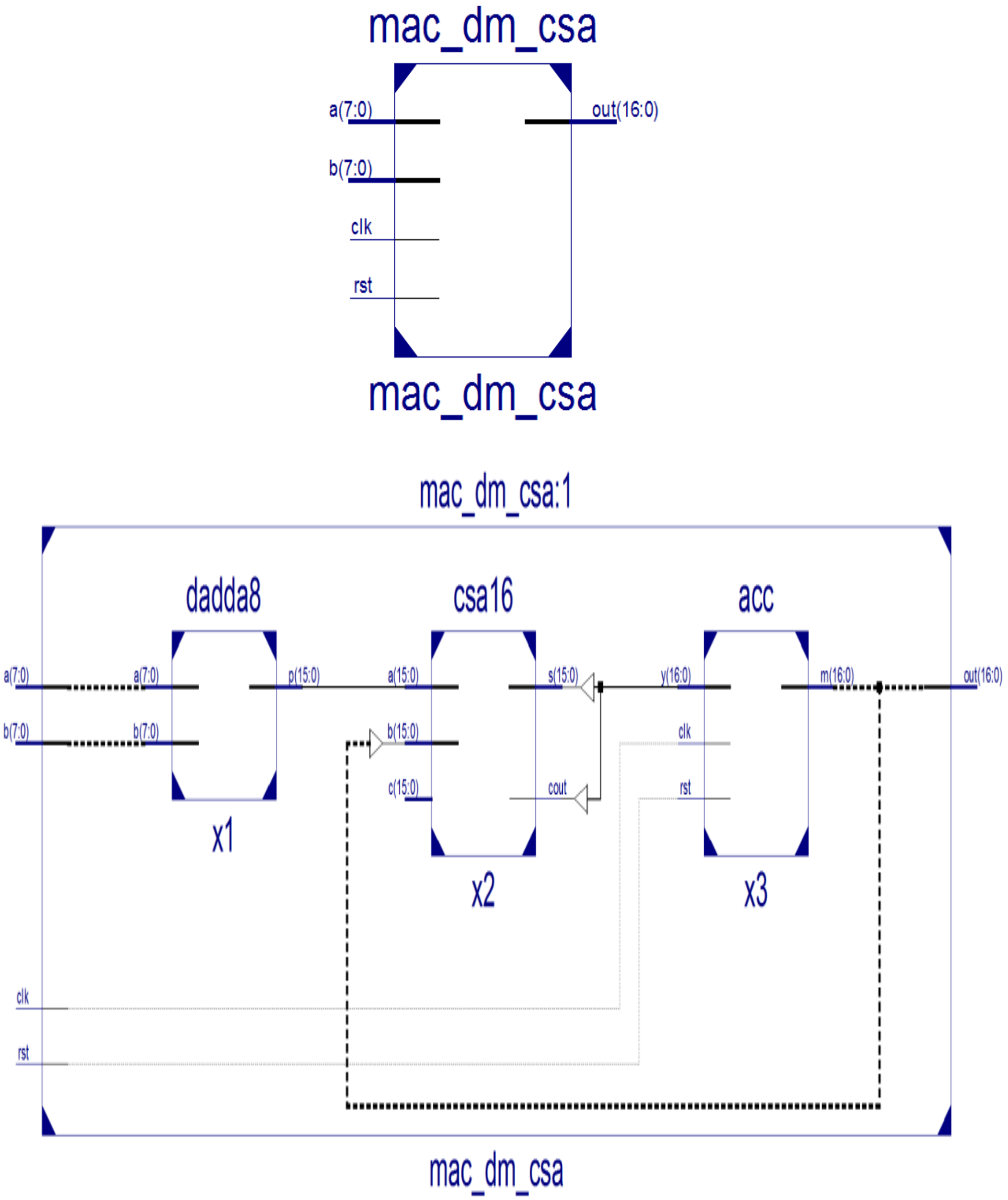


Fig.4.23.Technology View and RTL View of Existing MAC unit model (DM+CSA+Accumulator)

And, the timing waveform of Existing MAC unit model is illustrated in Fig 4.24.

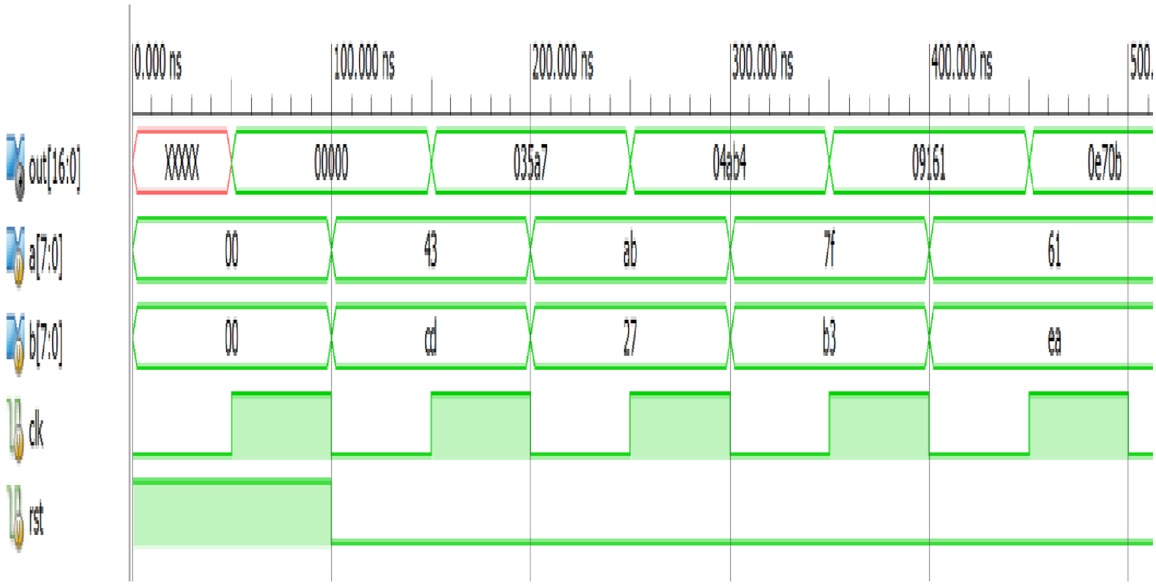


Fig.4.24.Timing Waveform of Existing MAC Unit Model (DM + CSA + Accumulator)

Table 4.12 Device Utilization Summary for Existing MAC Unit Model	
Logic Utilization	**Used**
Number of Slice LUTs	102
Number of occupied Slices	28
Number of bonded IOBs	35
Average Fanout of Non-Clock Nets	4.30
Power in milliwatts (mW)	197.1
Maximum Frequency (MHz)	1102.353
Total delay in Nanosecond	4.649

4.3.2 Proposed MAC Unit Model 1 (PDM+PCSA)

Technology view and RTL view of Proposed MAC unit model 1 are given in Fig 4.25.

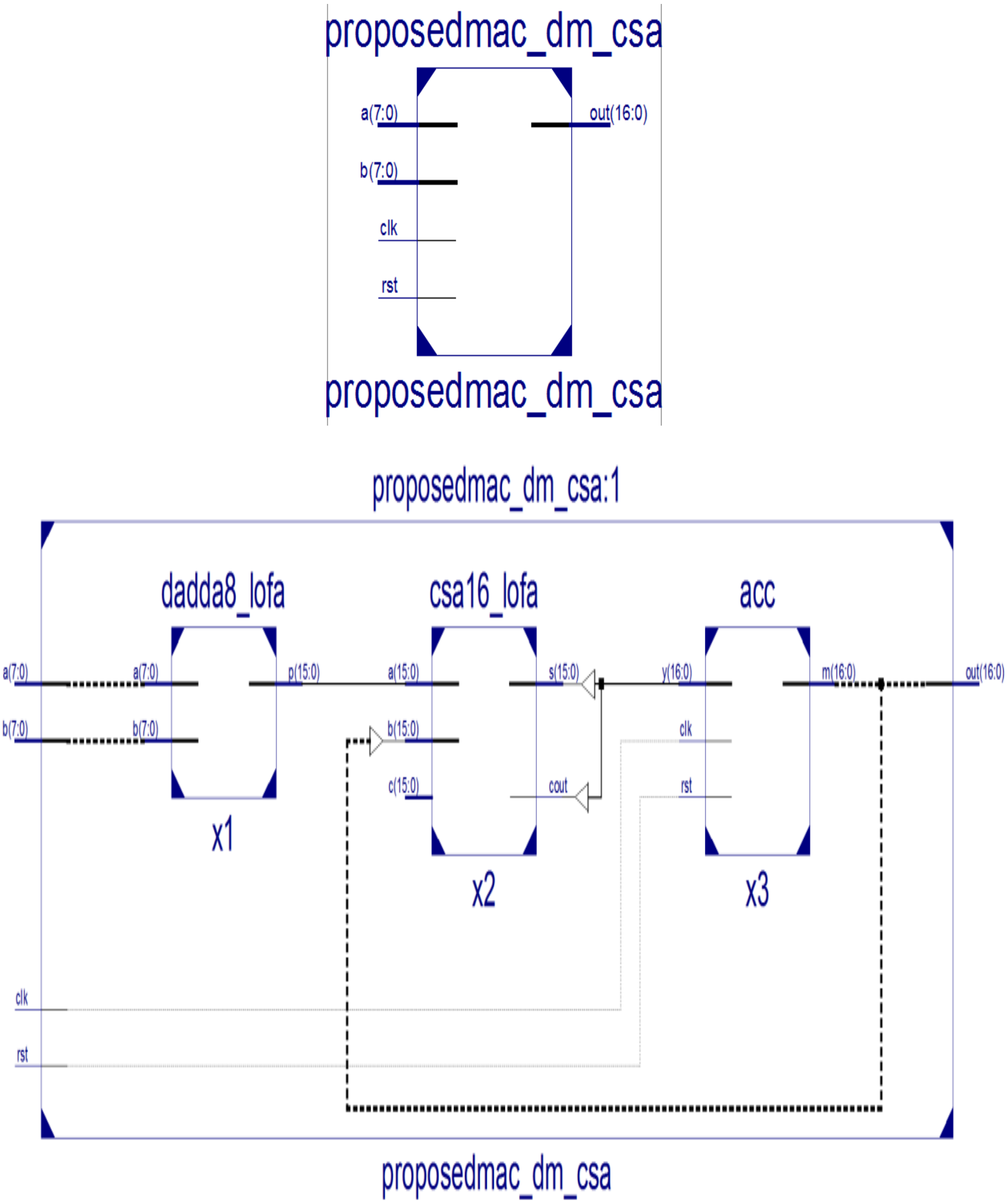


Fig.4.25.Technology View and RTL View of Proposed MAC unit model 1 (PDM+PCSA+Accumulator)

And, the timing waveform of Proposed MAC unit model 1 is illustrated in Fig 4.26.

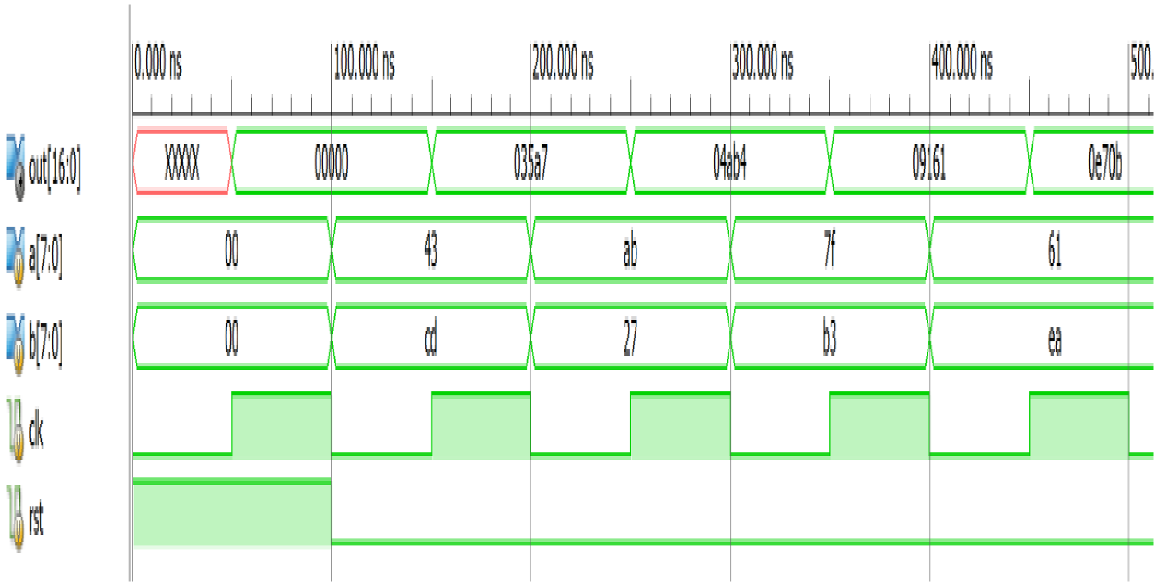


Fig.4.26.Timing Waveform of Proposed MAC Unit Model 1 (PDM + PCSA + Accumulator)

Table 4.13 Device Utilization Summary for Proposed MAC Unit Model 1	
Logic Utilization	**Used**
Number of Slice LUTs	102
Number of occupied Slices	31
Number of bonded IOBs	35
Average Fanout of Non-Clock Nets	4.30
Power in milliwatts (mW)	193.5
Maximum Frequency (MHz)	1281.558
Total delay in Nanosecond	4.111

4.3.3Proposed MAC Unit Model 2 (DM+CIA)

Technology view and RTL view of Proposed MAC unit model 2 are given in Fig 4.27.

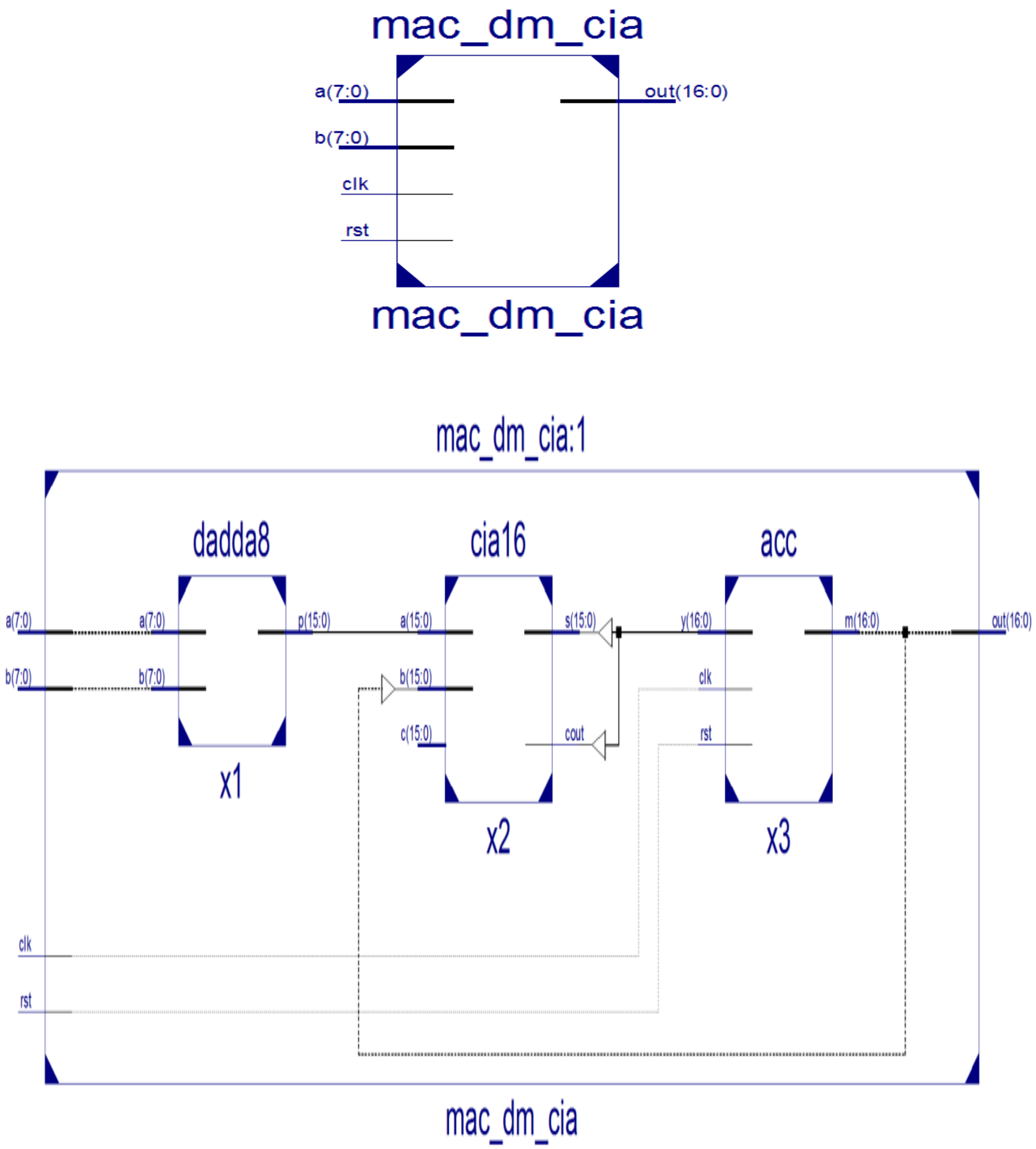


Fig.4.27.Technology View and RTL View of Proposed MAC unit model 2 (DM+CIA+Accumulator)

And, the timing waveform of Proposed MAC unit model 2 is illustrated in Fig 4.28.

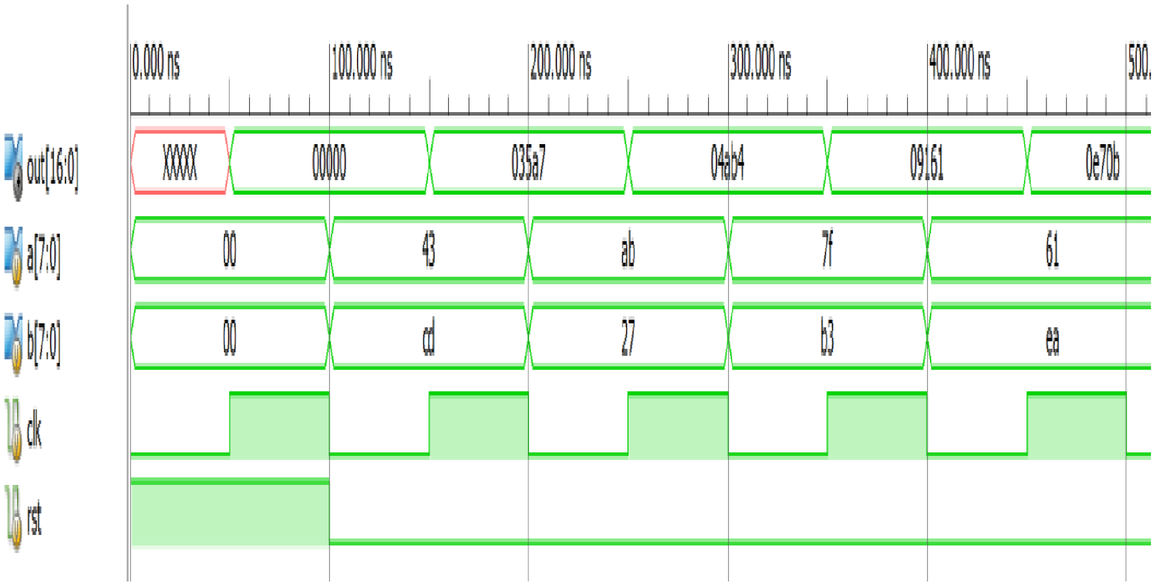


Fig.4.28.Timing Waveform of Proposed MAC Unit Model 2 (DM + CIA + Accumulator)

Table 4.14 Device Utilization Summary for Proposed MAC Unit Model 2	
Logic Utilization	**Used**
Number of Slice LUTs	130
Number of occupied Slices	37
Number of bonded IOBs	35
Average Fanout of Non-Clock Nets	4.30
Power in milliwatts (mW)	193.7
Maximum Frequency (MHz)	448.511
Total delay in Nanosecond	5.764

4.3.4Proposed MAC Unit Model 3 (PDM+PCIA)

Technology view and RTL view of Proposed MAC unit model 3are given in Fig 4.29.

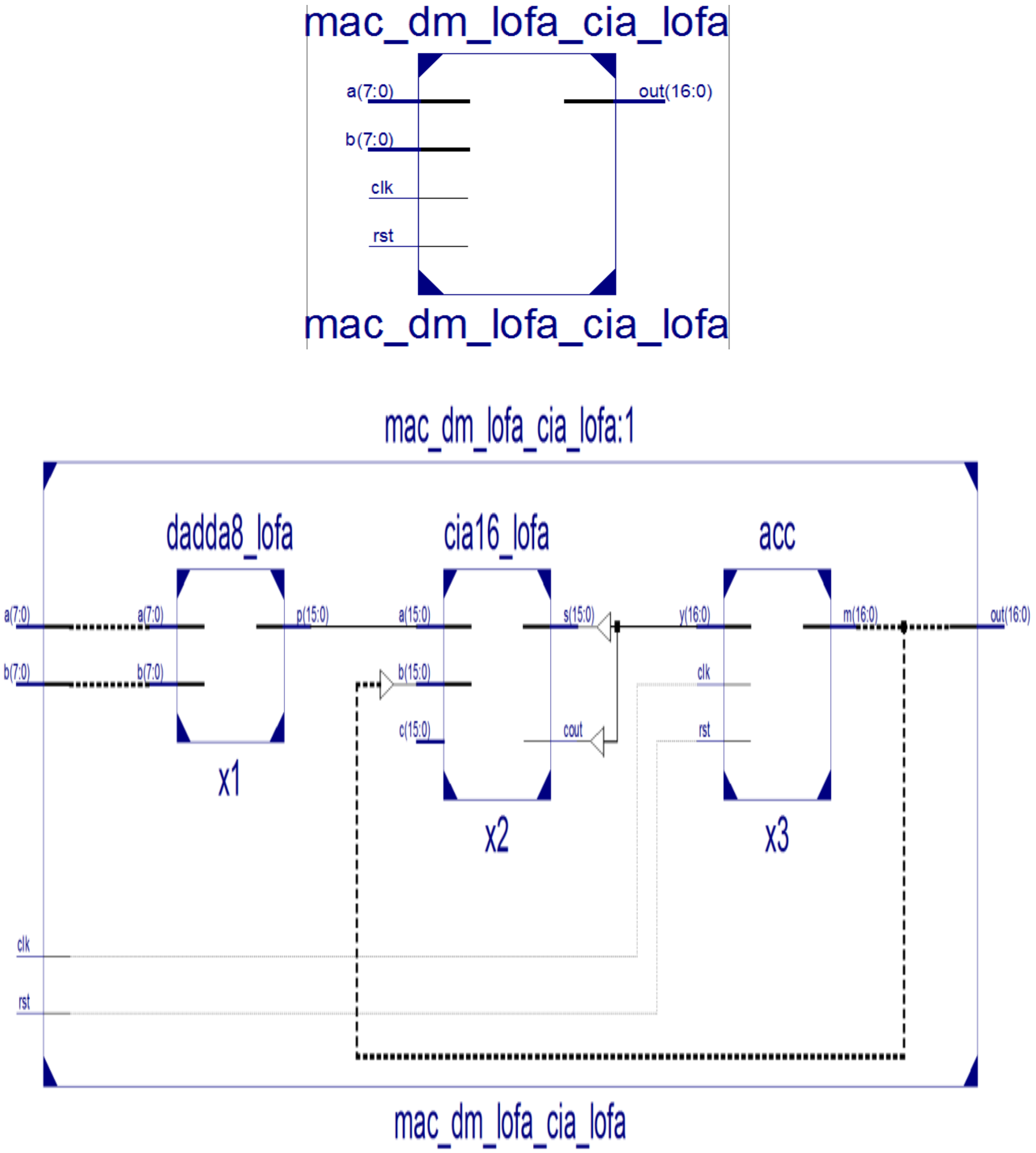


Fig.4.29.Technology View and RTL View of Proposed MAC unit model 3(PDM+PCIA+Accumulator)

And, the timing waveform of Proposed MAC unit model3 is illustrated in Fig 4.30.

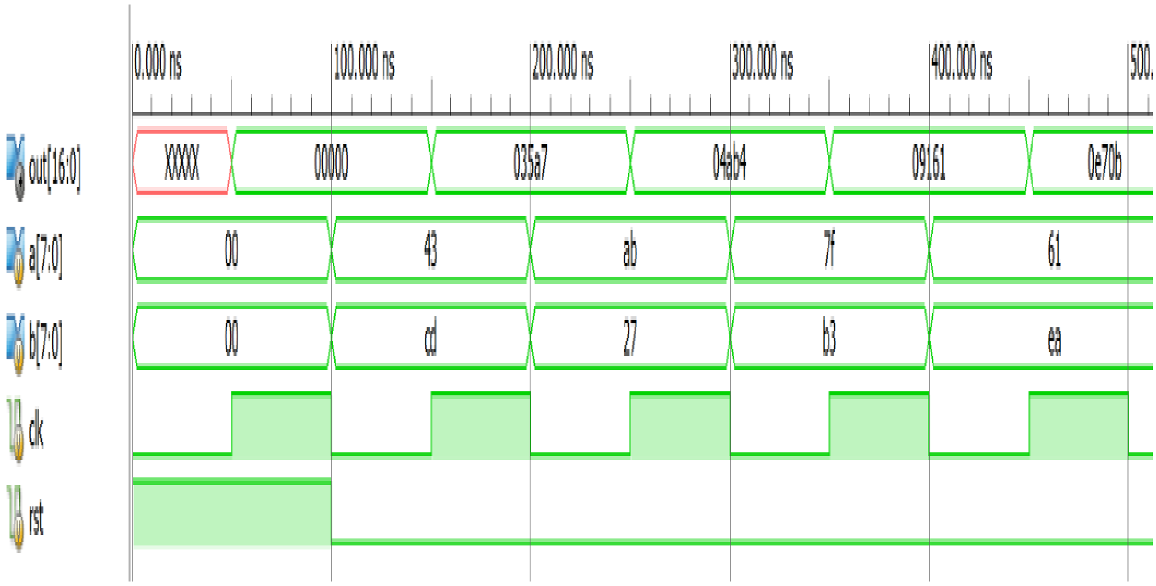


Fig.4.30.Timing Waveform of Proposed MAC Unit Model 3 (PDM+PCIA+Accumulator)

Table 4.15 Device Utilization Summary for Proposed MAC Unit Model 3	
Logic Utilization	**Used**
Number of Slice LUTs	130
Number of occupied Slices	35
Number of bonded IOBs	35
Average Fanout of Non-Clock Nets	4.30
Power in milliwatts (mW)	195.5
Maximum Frequency (MHz)	530.729
Total delay in Nanosecond	4.911

4.4 PERFORMANCE ANALYSIS OF ADDERS

The performance comparison of various adders with respect to the performance metrics such as area, delay and power are given in Table 4.16 & Table 4.17.

Table 4.16 Performance Analysis of Single Bit Adders

S.No.	Design	Area (LUT's)	Area (Slices)	Delay (ns)	Power
1.	Half Adder	1	1	0.770	90.6
2.	Traditional Full Adder	1	1	0.923	90.6
3.	Logically Optimized Full Adder	1	1	0.776	90.6

From above performance analysis table, it is observed that, LOFA having better performance in terms of area (LUT's and Slices, delay and Power.

Table 4.17 Performance Analysis of Parallel Adders

S. No.	Design	Area (LUT's)	Area (Slices)	Delay (ns)	Power (mW)
1.	Ripple Carry Adder	16	12	4.594	95.2
2.	Proposed Ripple Carry Adder	16	12	3.872	96.8
3.	Carry Save Adder	24	13	1.868	87.5
4.	Proposed Carry Save Adder	24	13	1.586	90.5
5.	Carry Increment Adder	24	15	4.254	95.2
6.	Proposed Carry Increment Adder	24	15	3.873	98.5

From above performance analysis table, it is observed that, Proposed Carry Save Adder having better performance in terms of area (LUT's and Slices), power and delay.

4.5 PERFORMANCE ANALYSIS OF MULTIPLIERS

The performance comparison of both existing & proposed Dadda Multiplier Architecture with respect to the performance metrics such as area, delay and power are given in Table 4.3.

Table 4.18 Performance Analysis of existing & proposed Dadda Multiplier

S. No.	Design	Area (LUT's)	Area (Slices)	Delay (ns)	Power (mW)
1.	Existing Dadda Multiplier	86	34	4.395	96.8
2.	Proposed Dadda Multiplier	86	34	3.511	95.1

From above performance analysis table, it is observed that, PDM having better performance in terms of delay and power without compromising with area.

4.6 PERFORMANCE COMPARISON OF MAC UNIT MODELS

The performance comparison of both existing & proposed MAC Unit with respect to the performance metrics such as area, delay and power are given in Table 4.4.

Table 4.19Performance Analysis of existing & proposed MAC Unit Models

S. No.	Design	Area (LUT's)	Area (Slices)	Delay (ns)	Power (mW)
1.	Existing MAC Unit	102	28	4.649	197.1
2.	Proposed MAC Unit Model 1	102	31	4.111	193.5
3.	Proposed MAC Unit Model 2	130	37	5.764	193.7
4.	Proposed MAC Unit Model 3	130	35	4.911	195.5

From above performance analysis table, it is observed that, Delay of Proposed MAC Unit Model 1 is 0.538 Nanosecond lower than Existing MAC Unit. Also the power consumption of Proposed MAC Unit Model 1 is 3.6 mW lower than Existing MAC Unit. Hence Proposed MAC Unit Model 1 having optimized performance in terms of area (LUT's and Slices), power and delay compared to that of other topologies.

4.7 PERFORMANCE COMPARISON OF MAC UNIT MODELS IN TERMS OF AREA

The performance analysis of existing & proposed MAC Unit Models in terms of area is shown in bar graph in Fig 4.31.

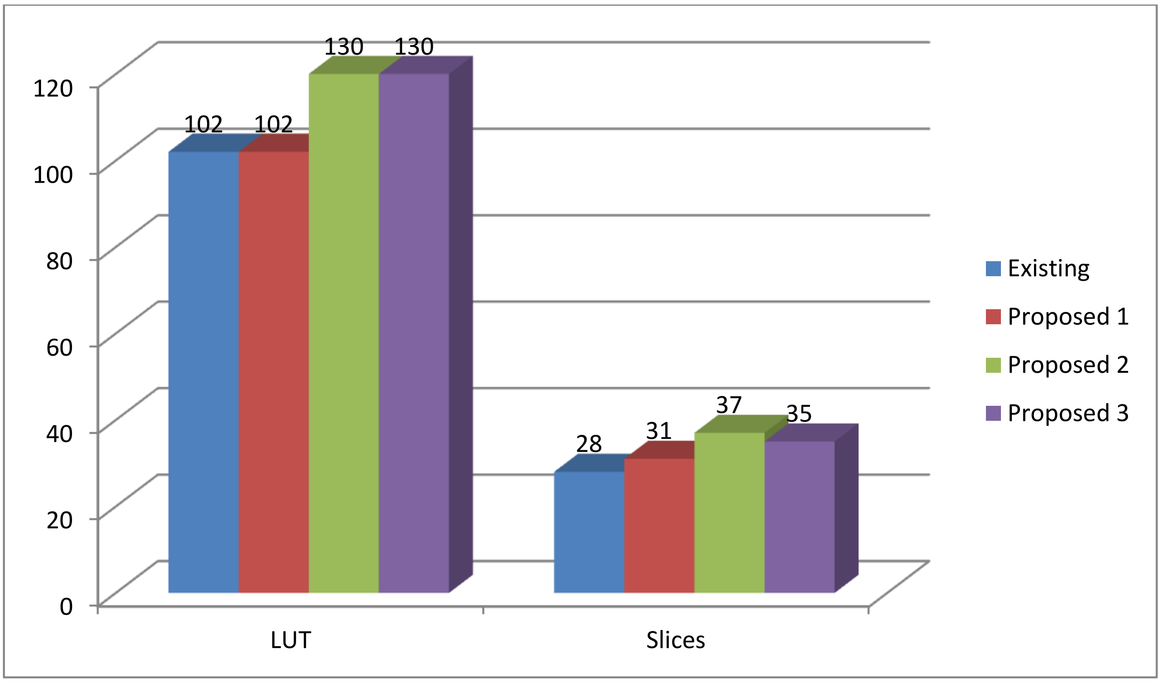


Fig.4.31.Performance Analysis of existing & proposed MAC Unit Models in terms of Area

It is observed through the Performance analysis of MAC Unit models shown in Fig 4.31 that:

- Proposed MAC Unit Model 1 requires less area compared to that of Proposed MAC Unit Model 2 & 3.
- Proposed MAC Unit Model 1 requires same area as of Existing MAC Unit.

4.8 PERFORMANCE COMPARISON OF MAC UNIT MODELS IN TERMS OF POWER

The performance analysis of existing & proposed MAC Unit Models in terms of power is shown in bar graph in Fig 4.32.

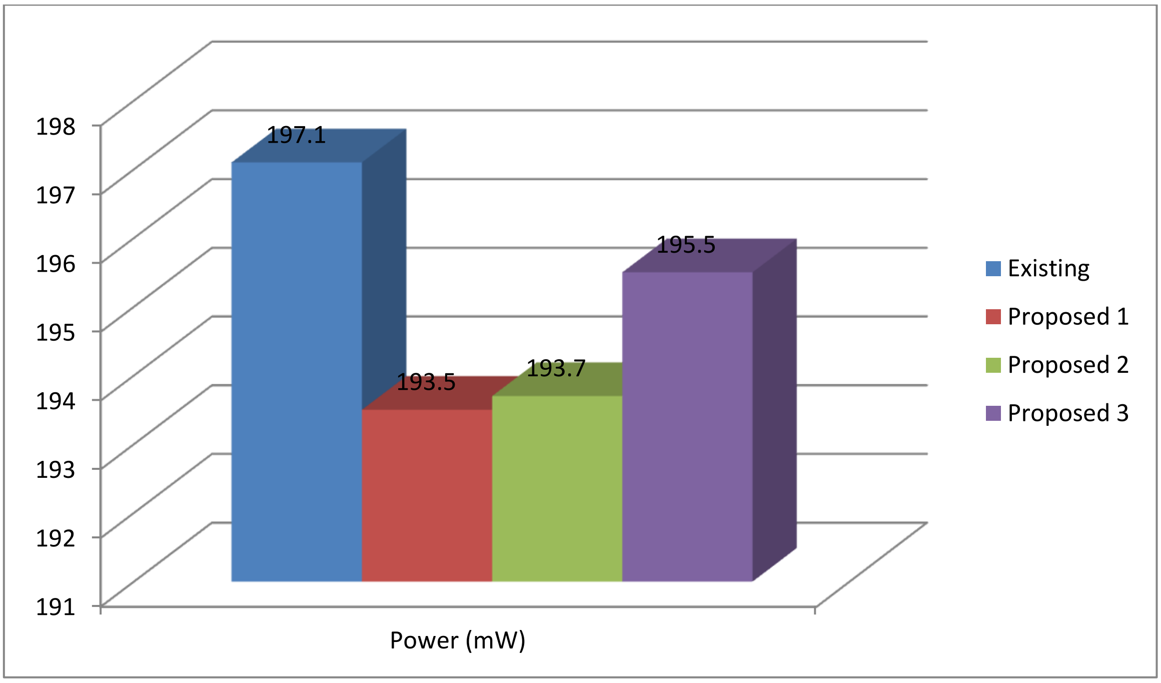


Fig.4.32.Performance Analysis of existing & proposed MAC Unit Models in terms of power

It is observed through the Performance analysis of MAC Unit models shown in Fig 4.32 that:

- Proposed MAC Unit Model 1 consumes less power compared to other MAC Unit models.

4.9PERFORMANCE COMPARISON OF MAC UNIT MODELSINTERMS OF DELAY

The performance analysis of existing & proposed MAC Unit Models in terms of delay is shown in bar graph in Fig 4.33.

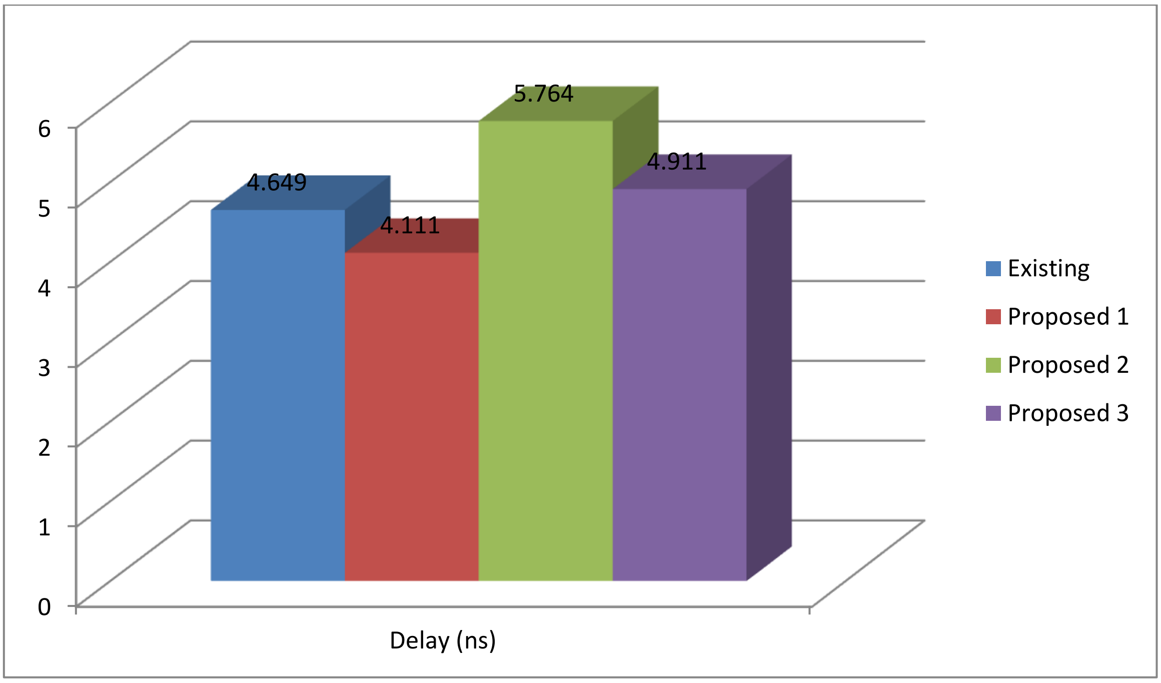


Fig.4.33.Performance Analysis of existing & proposed MAC Unit Models in terms of delay

It is observed through the Performance analysis of MAC Unit models shown in Fig 4.33 that:

- Proposed MAC Unit Model 1 has lower delay compared to other MAC Unit models.

Conclusion: Proposed MAC Unit Model 1 having better area, delay and power optimization compared to other MAC Unit models.

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1CONCLUSION:

Various MAC unit models are designed using Verilog HDL. Simulation and synthesis are done using Xilinx ISE 12.2 for Virtex-6 family 40nm technology device. The power is calculated using Lattice Diamond Design Suite software. In the existing MAC unit designed using Dadda Multiplier and adder as Carry Save Adder (CSA). The proposed MAC unit Models are designed using Dadda Multiplier and adder as Carry Increment Adder (CIA) or Carry Save Adder (CSA). However in the proposed model all traditional full adders are replaced by improved full adder. It is observed that, the Proposed MAC Unit Model 1 having better area, delay and power optimization compared to other MAC Unit Models.

5.2FUTURE WORK:

In future work, it is required to design MAC unit architecture with low area, delay and power in order to meet the needs of current VLSI Industry. Further, this work can be extended by designing of MAC unit with higher number of bit sizes such as 16, 32 and 64 and also for designing applications like ALU, filters etc.And these models can be widely used in the design of high performance DSP applications such as FFT, FIR, and IIR. Further, these models can be designed using ASIC technology for the specific application purpose. Furthermore, this work can be extended by designing the various register sets, instruction set, bus architectures which will leads to the design of complete processor.

REFERENCES

[1] Ivan E. Sutherland, Bob F. Sproull, David L. Harris "Logical Effort Designing Fast CMOS Circuits" Published by Morgan Kaufmann, May 1998.

[2] Samir Palnitkar "Verilog HDL A Guide to Digital Design and Synthesis" Published by Prentice Hall, March 1996.

[3] Young-Ho Seo, Dong-Wook Kim, "A New VLSI Architecture of Parallel Multiplier-Accumulate Based on Radix-2 Modified Booth Algorithm", IEEE Transactions on VLSI Systems, vol.18, no.2, pp.201-207, February 2010.

[4] Maroju SaiKumar, D.Ashok Kumar, Dr.P.Samundiswary "Design and Performance Analysis of MAC Unit" IEEE International Conference on Circuit, Power and Computing Technologies pp.1084-1089, March 2014.

[5] R.Uma and P.Dhavachelvan "Logic optimization using technology independent mux based adders in FPGA" International Journal of VLSI design & Communication Systems (VLSICS) Vol.3, No.4, pp.135-149, August 2012

[6] Maroju SaiKumar, Dr. P. Samundiswary "Design and Performance Analysis of Various Adders using Verilog" IJCSMC, Vol. 2, Issue. 9, September 2013, pp.128 – 138

[7] Padma Devi, Ashima Girdher, and Balwinder Singh, "Improved Carry Select Adder with Reduced Area and Low Power Consumption", International Journal of Computer Applications, vol.3, no.4, pp.14-18, June 2010.

[8] R.Uma, Vidya Vijayan, M.Mohanapriya, and Sharon Paul, "Area, Delay and Power Comparison of Adder Topologies", International Journal of VLSI Design & Communication Systems, vol.3, no.1, pp.153-168, February 2012.

[9] Prathibadevi Tapashetti, A.S. Umesh, Ashalatha Kulshrestha, "Design and Simulation of Energy Efficient Full Adder for Systolic Array", International Journal of Soft Computing and Engineering, vol.1, no.6, pp.356-360, Jan 2012.

[10] Maroju SaiKumar, P.Samundiswary, "Design and Performance Analysis of Various Multipliers using Verilog HDL", CiiT International Journal of Programmable Device Circuits and Systems, vol.5, no.9, pp.391-398, Sep 2013.

[11] K.M K.M.Prabhakaran, A.Karthika, “Low Complexity and High Accuracy Fixed Width Modified Booth Multiplier”, International Journal of Scientific and Research Publications, vol.3, no.3, pp.1-4, March 2013.

[12] Nithya J, Sathiyabama G, Revathi K “Comparative Study of Low Power Low Area Bypass Multipliers for Signal Processing Applications” Int. Journal of Engineering Research and Applications Vol. 5, Issue 1(Part 2), pp.95-98, January 2015.

[13] Ron S. Waters and Earl E. Swartzlander, Jr., "A Reduced Complexity Wallace Multiplier Reduction,” IEEE Transactions On Computers, vol. 59, no. 8, pp.1134-1137, August 2010.

[14] Jasbir Kaur, Kavita, “Structural VHDL Implementation of Wallace Multiplier”, International Journal of Scientific & Engineering Research, vol.4, issue.4, pp.1829-1833, April 2013.

[15] Anju S, M Saravana, “High Performance Dadda Multiplier Implementation using High Speed Carry Select Adder”, International Journal of Advance Research in Computer and Communication Engineering, vol.2, issue.3, pp.1572-1575, March 2013.

[16] Devika Jaina, Kabiraj Sethi and Rutuparna Panda, “Vedic Mathematics based Multiply Accumulate Unit”, Proceedings of IEEE International Conference on Computational Intelligence and Communication Systems, Gwalior, India, DOI 10.1109/CICN.2011.167, pp.754-757, July 2011.

[17] Vaijyanath Kunchigi, Linganagouda Kulkarni, Subhash Kulkarni “32-bit MAC unit design using vedic multiplier” International Journal of Scientific and Research Publications, Volume 3, Issue 2, February 2013.

[18] Xilinx 12.4, “ISim User Guide”, UG660 (v 12.4), December 14, 2010.

[19] Xilinx13.4, “Synthesis and Simulation Design Guide”, UG626 (v13.4) January 19, 2012.

[20] Xilinx 13.1, “RTL and Technology Schematic Viewers Tutorial”, UG685 (v13.1), March 1, 2011.

[21] Lattice Diamond User Guide, June 2010.