FPGA IMPLEMENTATION OF PRE-ENCODED MULTIPLIERS

A Dissertation Submitted in Partial Fulfillment of the Requirement for the Award of the Degree of

MASTER OF TECHNOLOGY

in VLSI Design

Submitted By

SALMA MOTLA

601562020

Under Supervision of

Ms. Sakshi

Assistant Professor

ELECTRONICS AND COMMUNICATION ENGINEERING DEPARTMENT

THAPAR UNIVERSITY, PATIALA, PUNJAB

JUNE, 2017

DECLARATION

I hereby certify that this thesis entitled, **"FPGA Implementation of Pre-encoded Multipliers"** in fulfillment for the requirements for the award of Degree of Master of Technology in VLSI Design submitted to Electronics and Communication Department, Thapar University, Patiala, is an authentic record of my own work and is carried under the supervision of Miss Sakshi. The matter submitted via this thesis report has not been submitted for the award of any other degree to the best of our knowledge.

Date: **Salma Motla**

Place: **(601562020)**

CERTIFICATE

It is certified that the work contained in the thesis titled FPGA implementation of Pre-encoded Multipliers by Salma Motla [601562020] has been carried out under my supervision and that this work has not been submitted elsewhere for any other degree.

Ms Sakshi
Assistant Professor
ECED
Thapar University ,Patiala

Date:

Place:

ACKNOWLEDGEMENT

It is my proud privilege to acknowledge and extend my gratitude to several persons who helped me directly or indirectly in completion of this report. I express my heart full indebtedness and owe a deep sense of gratitude to my teacher and guides **Ms. Sakshi Bajaj, Assistant Professor** for their sincere guidance and support with encouragement to go ahead.

I am also thankful to **Dr. Alpana Agarwal, Associate Professor and Head, ECED** for providing us with the adequate infrastructure for carrying out the work.

I am also thankful to **Dr. HemDutt Joshi**, **P.G. Coordinator, ECED and Dr. Anil Arora, Programme Coordinator, ECED** for the motivation and inspiration that triggered me for the work.

Last but not the least, I would like to thank my parents for their years of unyielding love and encouragement. They have always wanted the best for me and I admire their determination and sacrifice.

The study has indeed helped me to explore knowledge and avenues related to my topic and I am sure it will help me in my future.

Salma Motla

Roll No. 601562020

ABSTRACT

With the increasing demand of high speed multipliers either embedded or on die new technologies have to be proposed. These new technologies should accomplish the demand of high speed and lesser area. They have to be energy or power efficient and should have low manufacturing cost. Since the introduction of Pre-encoded multipliers many research efforts have been put in using it for various applications but its major scope lies in architecture with high speed.

Pre-encoded multipliers for digital signal processing applications based on off-line encoding of coefficients is proposed. To extend this, the Non-Redundant radix-4 Signed-Digit (NR4SD) encoding technique, which uses the digit values (-1, 0, +1, +2) or (-2,-1, 0, +1) is used leading to a multiplier design with less complex partial products implementation. The partial products used in this pre-encoded multiplier design is designed using primitive NAND gates. The number of gates used is reduced as compared to the previous design as well as the number of stages in Partial Product Generation unit is also reduced .Extensive experimental analysis verifies that the proposed pre-encoded NR4SD multipliers, including the coefficients memory, improve the speed of operation than previous scheme. In proposed system the delay of the architecture design is reduced by replacing the CSA and CLA adder to carry skip adder**.**

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

ASIC	Application-Specific Integrated Circuit
CDMA	Code Division Multiple Access
CLA	Carry Look Ahead
CLB	Configuration Logic Blocks
CSA	Carry Save Adder
CMOS	Complementary Metal Oxide Semiconductor Field effect Transistor
CPLD	Complex Programmable Logic Device
CSLA	Carry Select Adder
DTC	Direct Torque Control
DSP	Digital Signal Processing
FIR	Finite Impulse Response
FFT	Fast Fourier Transformation
FPGA	Field Programmable Gate Array
IFFT	Inverse Fast Fourier Transform
IDCT	Inverse Discrete Cosine Transform

LUT	Look Up Table
MB	Modified Booth
MBE	Modified Booth Encoding
MPEG	Moving Picture Experts Group
NR4SD	Non-Redundant Radix 4 Signed Digit
NGC	Net list file with constraint information
PPG	Partial Product Generation
PLI	Programming Language Interface
RAM	Random Access Memory
ROM	Read Only Memory
RTL	Register-Transfer Level
VHDL	Very High Speed Integrated Circuit Hardware Description Language
VLSI	Very Large Scale Integration

CHAPTER 1

INTRODUCTION

1.1 BACKROUND

In today's fast technologically developing world, the shift has been towards construction of small and portable devices. As the number of these battery operated, processor driven equipment increase, their performance demand is expected to be more. There is a need of increasing their speed and reducing their power dissipation. In such a consumer controlled scenario, these demands mean a serious look into the construction of the devices. These processors used for such purposes but also, major operations such as FIR filter design DTC etc. are done through multipliers. As multipliers are the major components of DSP, optimization in multiplier design will surely lead to a better operating DSP.

1.2 IMPORTANCE OF MULTIPLIERS

High presentation systems widely use Multipliers. They are used as little blocks in large digital signal processing systems, filters, microprocessors etc. Digital Signal Processors have operations like convolution operations; transform operations, filtering operations etc. All these operations need multiplication unit. So multiplier plays an important role in Digital Signal Processing. Multiplication operation is performed by recurring addition in older microprocessors although multiplication instructions must be accessible in recent microprocessors. For baseband signal transmission in communication systems multiplier is used. Filter has large number of multipliers, so for high speed applications, exploit of fast multipliers is necessary. In video processing and in 3D graphics and for displaying streamline images multipliers have many applications. For determining the performance of multipliers time taken in devious final result is extremely significant [1]. So high speed multiplier is must of need.

1.3 BASIC OPERATION OF MULTIPLIERS

As done in mathematics two numbers are multiplied, digital multipliers are operated i.e. addition and shifted partial products. Using AND gates and full adders, unsigned number multiplication are done. For generating the partial products AND gates are used and for

addition of the generated partial products full adders are used. The negative numbers is primarily transformed into its two's complement depiction for signed number multiplication, so that every bit of the partial products are positive.

Multimedia applications like audio-video coder-decoders, fast flourier transform at the time of execution of application uses a vast figure of multiplication such that coefficients do not alter. The multiplier's performance is critically affected by their architecture [2]. Canonic Signed Digit representation is used to obtain the least non-zero digits from the constant coefficient encoding [1]. The switching activity of the Canonic Signed digit multipliers are less due to the less non-zero partial product terms.

In today's scenario more than 70% of the instructions in microprocessor and mostly in digital signal processing algorithms perform multiplication and addition. The execution time is dominated by these operations. Therefore we need high speed multiplier. There is also another major issue of low power consumption in multipliers. If we reduce number of operations, the significant power consumption is also reduced and thus by reducing the dynamic power which is the major part of total power consumption. This leads us to increased need of low power and high speed multiplier. The main concentration of designer is on low power and high speed efficient design of circuit. A good multiplier's objective is to provide a physically packed together low power consumption and high speed unit. Booth multiplier, combinational multiplier, sequential multiplier, Wallace tree multiplier are the different type of multipliers. Previously the implementation of multiplication was generally by a sequence of subtraction, addition and shift operations. Multiplicand is the number that to be added, multiplier is the number of the times that is to be added and product is the result. Partial product is generated by each step of addition. The same number of bits is captured by operands in most of the computers. In order to preserve the contents of information the product becomes twice the length of the operands. The method of repeated addition is always replaced by algorithm that utilizes positional representation due to slow performance of arithmetic operations. The multipliers are decomposed into two parts. The first part is for generation of partial product terms and the function of second part is to collect and perform addition. The multiplication is twofold process that is partial product evaluation and accumulation of shifted partial product terms. Successive addition of shifted partial product term matrix columns performs this. Shifting of multiplier and gating of the appropriate bit of the multiplicand is successful. The all gated, delayed instance of shifted partial product term

matrix of the multiplicand must be in the same column. Product bit of particular form is formed by the addition of them. Therefore multiplication is defined as the operation of multi-operands. Two's complement number representation is used to extend multiplication of both signed and unsigned numbers.

1.4 Multiplier Schemes

In multiplication process there are two basic schemes. These are serial multiplication and parallel multiplication.

1.4.1 Serial Multiplication

It includes computation of set of partial product terms and then summing the partial product term together. The executions are primal with simple architectures and used when there is a lack of keen hardware multipliers.

1.4.2 Parallel Multiplication

The generation of partial product terms are simultaneously. These are used for high performance machines, but there is a need for computation latency to minimize. Parallel multipliers are better than serial multipliers. The parallel multipliers are faster than serial multipliers due to the lesser number of steps.

1.5 Different Types of Multipliers

The characteristics of efficient multipliers are accuracy, speed, area and power. Multiplication involves three steps

1. Partial product generation
2. Partial product reduction
3. Final addition

The different types of multipliers are discussed below

1.5.1 Combinational Multiplier

Combinational multipliers are used for multiplication of both signed and unsigned number. The operation of this multiplier is bit wise that every bit of multiplier and multiplicand is multiplied. In multiplier product is calculated according to the bit location and the final result is obtained by the addition of product terms. Generation of intermediate products are easy in binary multiplication which is the main advantage .combinational multipliers are generally consumes lot of area and are slow.

1.5.2 Wallace Tree Multiplier

A digital circuit that consist of efficient hardware and that multiplies two integers is Wallace tree multiplier. Carry select adder is used for the addition of partial products in where Wallace tree also reduces partial product terms.

There are mainly three steps for description of operation of Wallace tree multiplier:-

1. Multiplicand of same bit position is multiplied with the multiplier's each bit. Partial products of different weights are generated depending upon of the position of bits of multiplier.
2. Two layers of half adders as well as full adders be intended for the reduction of partial production terms.
3. Conventional adders are used for the addition of the rows of carry and sum that we get after second step.

In Wallace tree multiplier the propagation delay for each layer is O(1) and the main advantage is that there is only O(log n) reduction layers. In regular adders for addition of partial product terms the time required is O(log n2).

1.5.3 Array Multiplier

Regular structures are the main reason for the importance of array multipliers. Multiplier circuits operation is based on recurring addition as well as shifting procedure. Generation

of every partial product is due to the result of multiplication of multiplicand and multiplier digit. According to the bit sequence the shifting of partial products are done. Carry propagation adders are used for summation. The number of ladders required are N, where N are number of multiplier bits.

1.5.4 Sequential Multiplier

The multiplication of two binary numbers using a single n bit adder where multiplicand X is of n bits and multiplier Y is of m bits, a sequential circuit is build which m times cycles the circuit and processes a single partial product terms. And this circuit is known as sequential multiplier. The main advantage of this multiplier is requirement of less area. Partial products are generated at each step, and then accumulation of partial products and then this partial sum is shifted to arrange in a line the previous calculated sum by next stage partial product terms. Therefore, sequential multiplication process consists of three operations that are partial product generation, addition of partial product terms to accumulated sum and partial product shifting. To produce result number of cycle are taken by sequential multipliers due to which the latency of this multiplier may be equal to or more than that of combinational multiplier to get an output in conditions of complete time. When the numbers that we want to multiply is less, and then coding is very easy.

1.5.5 Booth Multiplier

Booth multiplication's algorithm is a method for multiplying binary integers in both signed and unsigned representation. Booth's algorithm is implemented using following steps. Let X is having m number of bits and Y is having n number of bits.

Step 1:-For Booth's Table

While making booth's table we have to acquire four columns one for multiplier and second column for previous first least significant bit of multiplier and last two (U and V) for partial product accumulator (P)[3]

1. Select multiplier X and multiplicand Y.

2. Let us take multiplicand Y two's compliment.
3. In the table load value of X
4. If value is X-1 load 0.
5. To obtain product of X & Y load 0 in U and V at the end of the process.
6. We have to make n rows for each cycle because m and n bits numbers are multiplied together.

Step 2:- Booth's Algorithm

In Booth algorithm multiplier bits are examined and the partial product (P) is shifted. In addition to shifting of the multiplicand the addition of Partial product, subtraction as of the Partial product, not here unaffected according to the rules given:

1. X_i X_{i-1}

   0 0 only shifting

   1 1 only shifting

   0 1 addition of Y to U and then shifting

   1 0 subtraction of Y from U and then shifting

2. Shifting arithmetic right U and V collectively due to which the sign bit of two's complement number is jammed. So that negative numbers as well as positive numbers leftovers negative and positive correspondingly.
3. To prevent us from use of two registers X is shifted circularly right. Repeat the same steps until n number of cycles is completed. At the end we obtain the product of X and Y.

1.6 The following characteristics are possessed by efficient multiplier

- Accurateness: Always correct result is given by good multiplier.

- Speed: Operations at high speed should be performed.

Table 1.1 Booth Table

U	V	X	Y
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	2
1	0	0	-2
1	0	1	-1
1	1	0	-1
1	1	1	0

- Area: Lesser number of slices and LUTs are occupied by good multiplier.
- Power: Less power should be consumed by multiplier.

1.7 Organization of Thesis

Organization of thesis is as follows:

Chapter 1 presents a brief overview of multipliers and description of the types and applications of the multipliers.

Chapter 2 gives a concise explanation of the research that has been reported in literature in Pre-encoded multipliers.

Chapter 3 discusses the pre-encoded schemes and algorithms.

Chapter 4 discusses the proposed work and algorithm.

Chapter 5 FPGA flow by means of Xilinx is explained in this chapter.

Chapter 6 Implemented results are explained and report is concluded.

Chapter 7 future possibilities to carry forward the research in this domain are discussed.

CHAPTER 2

LITERATURE REVIEW

Simmonds *et.al* (1993) [4] presented a system for generating subfields of sign extension and gadget for addition of signed partial products. The system as well as gadget was predominantly helpful, when engaged within combination of 3-bit customized Booth algorithm. The effect of summing the sign-extension bits mimics with correction factor is inevitable to which the partial product rows have been added. The fewer bits were contained particularly in correction factor so that the whole figure of bits was restricted in every elimination of sign extension part. Consequently fewer computer circuitries in moment was necessary intended for the summation stage of partial product procedure. The means which minimize the computer time and essential for generating the correction factor was also disclosed.

D. Villeger *et.al* (1993) [5] reported that schematic of Booth recoding is the outline of the consequential execution and fewer standard principal to an additional tricky design as well as VHDL account. The Booth method was at best equivalent in conditions of speed. In the two's complement depiction of addition without Booth encoding, the previous partial product row is generated by means of AND gate of an inverted input. In former words, the amount of gate levels is identical as in case of sign extent holder. Sign extension is needed by means of Booth encoding or without Booth encoding. The resulting two schemes are comparable, although using 4:2 compressors is more efficient due to lower number of gate levels and lesser complexity.

L.Cimineria *et.al* (1996) [6] presented n * n multiplication schemes where the one is the translation of carry-save multipliers, which needs an adder at the most recent move to adapt the carry-sum depiction of the main units. Whole of the product into a non-redundant figure is not performed. The obtained 2n-bit result is shaped. The proposed scheme has obtainable design of binary as well as two's complement $n * n$ multipliers. In the proposed multiplication schemes, the computation of the least significant bits is approved, `in parallel with the resolution of the most significant digits.

Y.T Han *et.al* (1998)[7] presented a Moving Picture Experts Group-2 audio which acts as the assistant band coding expertise for various applications of audio. In case of audio decoder of Moving Picture Experts Group-2, this document gives a semi-custom application-specific integrated circuit purposes. In this document the decoder implemented is the request of Moving Picture Experts Group -2 of international standard, as well as three parts are formed: the pre-processor, the creation filter and the multichannel processor. The VHDL (VHS1C Hardware Description Language) is used for the decoder scheme and is industrialised in 0.6,m~ CMOS semiconductor process as a solitary**.**

M.Kolluru *et.al* (2000) [8] presented a ROM for storing constants used in both the MPEG-2 and AC-3 audio decoding algorithms. These constants include (i) matrixing constants for AC-3 and MPEG audio decoding algorithms and (ii) windowing coefficients for AC-3 and MPEG audio decoding algorithms. The ROM includes (a) a first partition for storing a first set of constants which are windowing coefficients for AC-3 decoding, (b) a second partition for storing a second set of constants which are constants used for pre-IFFT and post-IFFT steps of AC-3decoding(Blkswflag=1), (c) a third partition for storing a third set of constants which are both IDCT coefficients for MPEG decoding and IFFT coefficients for AC-3 decoding,(d) a fourth partition for storing a fourth set of constants which are constants used for pre-IFFT and post-IFFT steps of AC-3 decoding (Blkswflag=0), and (e) a fifth partition for storing a fifth set of constants which are windowing coefficients for MPEG decoding. The first and fifth partitions store 256 windowing coefficients for AC-3 decoding and 256 windowing coefficients for MPEG decoding, respectively. The second partition stores 64 sine and 64 cosine coefficients for AC-3 pre- and post-IFFT steps (Blkswflag=1).The third partition stores 64 unique sine constants and 64 unique cosine coefficients. The fourth partition stores 128 sine and 128 cosine coefficients for AC-3 pre- and post-IFFT steps (Blkswflag=1).

A. Goldovsky *et.al* (2000) [9] reported the operation of two's complement parallel multiplier. To breed the partial products, the multiplier uses Booth encoders of radix-4 and well placed array of (3,2),(5,3), and (7,4) counters to diminish the sum of partial products in addition to carry. The product's extra major bits are computed from left to right by means of a modified Ercegovac-Lang converter.

Yeh *et.al*(2000)[10] proposed a methodology for parallel multiplier using booth encoding with high speed. For production of partial products and to recover the performance in final

addition of fixed Modified Booth Encoding schemes, they proposed a new system which also uses modified Booth encoding but in an improved manner. For addition in final stage, a new algorithm is urbanized in the direction of creation of conditional-sum adder of multiple-level. According to the known cell property and summary of input delay, the final adder is optimized by projected algorithm. As compared it with sum adder based on binary tree conditionally, the rapidity performance enhancement is up to 25 %. On standard, the intended development here in reduces the total delay of parallel multiplier by 8 percent. The entire scheme has been confirmed through simulation at gate stage.

N.Y.Shen *et.al* (2002) [11] observed that on minimising switching actions of partial products with the Booth algorithm using radix4,the input datum resulted with slighter efficient active range, is a method for generation of codes by Booth scheme. By imposing the active-range fortitude units to control input data path, the projected 16*16-bit low-power multipliers are independently implemented. Towards the demonstration of the low-power dissipation multiplier, the switching activities analyses of partial products be imitated. Authors compared 10 values those consumed by the conventional multipliers. The projected design preserves power by 14%, 30% and 31% correspondingly. Therefore, the design projected in this is able to extensively use for a variety of media processing to attain low-power expenditure next to fair-haired hardware expense.

J.Park *et.al* (2003) [12] presented a high speed scheme for FIR filters computation which is based on the computation multiplier that specially targets computation and used again in products of vector-scalar. The implementation of carry–save and Wallace tree multipliers in 0.35nm technology is compared and the speediness of the work is improved by 52% via carry-save multiplier and 33% via Wallace tree multiplier scheme. The power expenditure of the FIR filter by means of voltage scaling, based on computation contribution multiplier using Wallace tree multiplier, is able to compact to 41%.

Wang *et al*. (2007) [13] presented multiplication algorithm for two's complement multipliers that is sign-magnitude multi-bit overlapped hard-wired. The projected scheme adapted partial products in a reduced left edging banded matrix. Rows are extended through matrix of partial products in such a manner that the length of the partial product conditions at the right as well as left ends is insignificant. The resultant adapted matrix of the partial product can be abridged to the concluding product by means of diversity of eminent scheme. With the

projected sign-encoding technique, the design system of the multiplier is non-iterative or somewhat iterative, so that the production and the drop of the multiplication matrix be able to be implemented with significant reserves area of chip, while in comparison to the earlier known algorithms that make use of sign extensions.

G.Pastuszak *et.al* (2008) [14] offered significant compensation at the computational level as compared to the previous video compression system but complexity to the design is added. First phase of the video coder is H.264/AVC binary coder which is presented in this document. The unit of the elevated outline chains two binary coding modes: situation adaptive binary arithmetic coding and situation adaptive variable-length coding. A substantial amount of hardware is saved since two coding modes contribute to the similar judgment plus storage elements. Five versions of the arithmetic coding pathway were urbanized to revise the area and performance trade off associated in parallel representation programming. The functioning results demonstrate that the parallel representation programming allows higher effectiveness.

Kim *et.al* (2008) [15] proposed the booth encoding procedure which is a modified Booth multiplier scheme. In this the pulse-shaping filter design is used in CDMA. It is exposed that by the projected method, area and power expenditure can be abridged up to 44% and 48%, correspondingly, compared with the predictable designs. In case of 128-point radix-24 FFT, the area and power expenditure is reduced by 18% and 36%, respectively.

A. Jacobson *et.al* (2009) [16] presented the design of an extremely configurable steady flow mixed-radix (CFMR) Fast Fourier Transform (FFT) processor. It presented a runtime configurable, steady-flow, mixed-radix memory support FFT processor. FFTs and IFFTs of size16 to 4096-point are performed on 32-bit (16+16) fixed-point complex data. Elevated throughput is achieved from side to side in-place radix-2/4 butterfly computational unit. While operation at 866 MHz, the processor require 1.5 μs to compute a 1024-pt complex FFT while having an average SQNR of 74 dB.

B.Paul *et.al* (2009) [17] reported low-power applications based on 16 x16 multiplier ROM. The design uses sixteen 4*4 ROM-based multiplier block followed by carry-save adders and a last carry-select adder (all ROM-based) to obtain the 32 bit output. The ROM-based design also provide 44% fewer delay than the array multiplier with a negligible addition to (7.7%) in

power. This illustrates the low-power process of the ROM-based multiplier also at improved frequencies. The ROM-based multiplier design accessible here considerably reduces the power consumption mostly due to fewer number of switching. It was observed that while this design has considerably better performance over the array multiplier, it still maintains low power consumption as compared to previous state-of-the-art high-performance multiplier designs.

B.Ramkumar *et.al* (2012) [18] proposed one of the adders used in data-processing processors that perform rapid arithmetic functions known as Carry Select Adder. CSLA structure indicates that there is an extent for falling of area in addition to power expenditure. This effort uses well-organized gate-level alteration to considerably reduce CSLA's area and power. The projected scheme has condensed area and power compared with the normal SQRT CSLA with the expenditure of delay. The given work evaluated the presentation of the projected scheme in conditions of delay, area, power and their results with logical effort and from side to side custom plan. The results examination shows that the projected CSLA arrangement is enhanced than the ordinary SQRT CSLA

M. Bahadori *et.al* (2016) [19] presented a carry skip adder formation which has a superior speed with lower energy expenditure as compared to the conventional one. By applying concatenation as well as incrimination scheme, the speed enhancement is tackled. In accumulation, AND-OR-Invert as well as OR-AND-Invert multipart gates are intended for the skip logic. Both predetermined stage size as well as variable stage size styles are realized for the configurations. The grades are obtained with the help of HSPICE simulation, disclosed standard 44% and 38% improvement within the delay as well as energy correspondingly.

Tsoumanis *et.al* (2016) [20] introduced style of multipliers that are pre-encoded and used for applications of digital signal processing that rely upon encoded offline coefficients .The Non-Redundant radix-4 Signed-Digit encoding system, which uses the digit values (-1; 0,+1;+2)or (-2,-1,0+1) was projected to a principal of using fewer complex partial products executed multiplier design .Booth scheme alteration algorithm for the triple moduli set $\{2^n - 1, 2^n, 2^n + 1\}$. The dilemma can be summarised to a simpler residue-to-binary converter for two moduli set of RNS acquiescent for fast mixed-radix exchange. Two dissimilar version of structural design based on this new formulation are projected. Compared with the fastest repeal converter

for the identical moduli set, the projected converters are efficient in both the area and time complexity.

CHAPTER 3

PRE ENCODED MULTIPLIERS

By means of the Canonic Signed Digit illustration Constant coefficients are encoded to include the fewer nonzero digits [21]. Multiplier design was proposed in a Canonic Signed Digit representation that shares certain features intended for group of pre-determined coefficients. The extent of ROM in which the groups of coefficients are stored is drastically summarised in addition to the area along with power expenditure of the circuit. Due to generation of partial products generation unit intended particularly for a cluster of coefficients they can't be reused in a further cluster, this multiplier proposes lack flexibility. Moreover, this scheme cannot be simply extensive to bulky groups of predetermined coefficients attain at the identical time elevated effectiveness.

The amount of partial products reduces to half resulting in reduced power consumption, area and critical delay tackles the aforementioned limitations in Modified Booth (MB) encoding[12][13][14].Though, a devoted encoding circuit be necessary and generation of the partial products is more complex.

3.1 ALGORITHM OF MODIFIED BOOTH SCHEME

A Modifying Booth technique includes encoding of redundant radix-4 numbers [14][15].Let us consider the two two's complement numbers A,B and multiply them both consist of n = 2k bits, B be able to be characterized in Modified Booth form as:

$$B = (b_{n-1}....b_0)_{2's} = -b_{2k-1}2^{2k-1} + \sum_{i-0}^{2k-2} b_i 2^i$$

$$= (b_{k-1}{}^{MB})_{MB} = \sum_{j=0}^{k-1} b_j{}^{MB} 2^{2j} \qquad (1)$$

Digits $b_j{}^{MB} \in \{-2,-1,0,+1,+2\}, 0 \leq j \leq k-1$, are formed as follows:

$$b_j{}^{MB} = -2b_{j+1} + b_{2j} + b_{2j-1} \qquad (2)$$

Where $b_{-1} = 0$. Bits s is used to represent each Modified Booth figure, computation of one and two is given in table 3.1. The bit s represent s = 1 negative digit or s = 0shows positive digit. The supreme worth of a single digit generation is 1 or not one represents by one. And the two is used to represent the absolute digit value equals 2 or not. By using those bits, we compute the Modified Booth digits b_j^{MB} as given:

$$b_j^{MB} = (-1)^{sj\cdot}(one_j + 2two_j) \qquad (3)$$

TABLE 3.1 Modified Booth Encoding

b_{2j+1}	b_{2j}	b_{2j-1}	b_j^{MB}	s_j	one_j	two_j
0	0	0	+0	0	0	0
0	0	1	+1	1	0	0
0	1	0	+1	1	0	0
0	1	1	+2	0	1	0
1	0	0	-2	0	1	1
1	0	1	-1	1	0	1
1	1	0	1	1	0	1
1	1	1	-0	0	0	1

Equation (4) from the Modified Booth encoding signals

$$s_j = b_{2j+1}, one_j = b_{2j-1} \oplus b_{2j},$$
$$two_j = (b_{2j+1} \oplus b_{2j}) \wedge \overline{one_j} \qquad (4)$$

Modified Booth algorithm, leads us to $0 \leq j \leq k - 2$. As the dynamic range is needed to be covered of the two's complement representation, the most significant digit's encoding is Modified Booth encoding (i.e., b_{k-1}^{MB}-€{-2;-1; 0;+1;+2}).

To increase the speed by the reduction of half of the partial production terms the Booth multiplier scheme is used. Booth encoded operand is called multiplier and the further one is called multiplicand. For generating partial products Modified Booth encoding is used in most

of cases due to its capability of reduction of partial product by half. The functionality of Booth encoder is shown in truth table.

3.2 NON-REDUNDANT RADIX-4 SIGNED DIGIT ENCODING ALGORITHM

The sequential schemes of encoding are extended by a NR4SD encoding technique. This encoding technique uses one value from the given sets of number values: (−2, −1, 0, +1)or(+2,+1,0,−1*)*.The most significant one digit is Modified Booth while others are encoded according to NR4SD in order to wrap entire the active range of two's complement representation., The standard coefficients are pre encoded using the given encoding formula and then stored into condensed form two bits per digit are stored in ROM. As we compared it to the previous pre encoded Modified Booth multiplier where the coefficients are encoded three bits for every digit, the projected this scheme helps in size reduction of memory, as well as in comparison to the Modified Booth scheme, where five digit values(−2, −1, 0, +1,+2) are used. The given encoding scheme uses four digits. Therefore in this pre-encoded multipliers the partial products generation circuit required is less complex. The size of the coefficients of ROM plays an important role in effectiveness of the abovementioned multipliers [23][24]. As the reduction of partial products is half in Modified Booth form, during the encoding of the two's complement number *B*, digits b_j^{NR-} one of the four values is taken:*(*−1, 0, +1, +2*)*or b_j^{NR+}(−2, −1, 0, +1*)*NR4SD+ otherwiseNR4SD− algorithm correspondingly. Instead of five values as used in Modified Booth algorithm only four diverse values are used which gives $0 \leq j \leq k-2$.The most significant digit is Modified Booth encoded as the dynamic range of the two's complement representation is need to be covered (that is., $\mathbf{b}^{MB}{}_{k-1} \in \{+2,+1, 0, -1, -2\}$). The NR4SD− as well as NR4SD+ encoded algorithms are explained in point in Fig. 3.4 and 3.5correspondingly

3.2.1 NR4SD- Algorithm

Step 1: let the values be $j = 0$ as well as. $c_0 = 0$

Step 2: The carry c_{2j+1} in addition to the sum n_{2j}^{+} with input values b_{2j} with c_{2j} of a Half Adder is calculated as (Fig. 3.1).

$$c_{2j+1} = b_{2j} \wedge c_{2j}, n_{2j}^{+} = b_{2j} \oplus c_{2j}$$

Step 3: The carry c_{2j+2} (+) with positive sign and the sum n_{2j+1}^{-} (-) with negative sign with inputs b_{2j+1} (+) and c_{2j+1} (+)(Fig3.1)of a Half Adder. The outputs c_{2j+2} and n_{2j+1}^{-} of Half Adder communicate to its inputs are deliberated as given:

$$2c_{2j+2} - n^{-}{}_{2j-1} = b_{2j+1} + c_{2j+1}$$

The Half Adder operation is summarized using given Boolean equations :

$$c_{2j+2} = b_{2j+1} \vee c_{2j+1},\ n_{2j+1}^{-} = b_{2j+1} \oplus c_{2j+1}$$

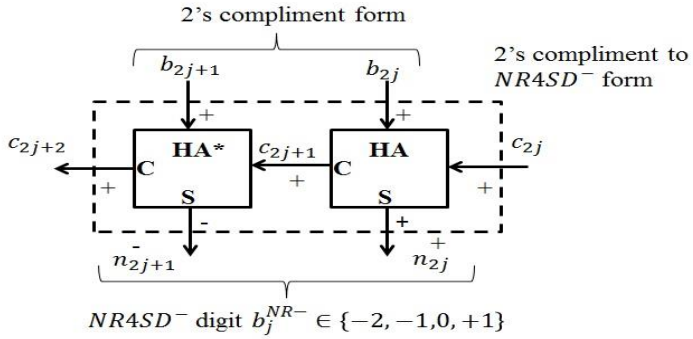
Step 4: The b_j^{NR-} digit value is calculated as follows

$$b_j{}^{NR-} = -2n^{-}{}_{2j+1} + n^{+}{}_{2j} \qquad (5)$$

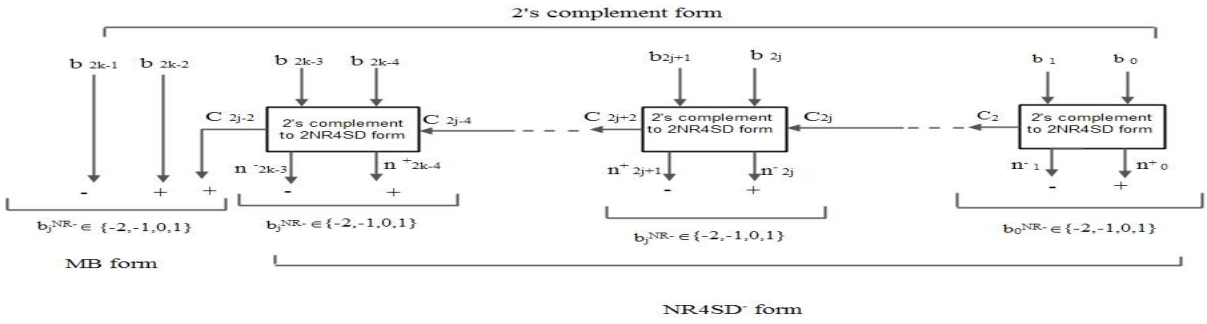
Table 3.2 NR4SD^{-} encoding

Two's complement			NR4SD^{-} form			Digit	NR4SD^{-} encoding		
b_{2j+1}	b_{2j}	c_{2j}	c_{2j+2}	$n^{-}{}_{2j+1}$	$n^{+}{}_{2j}$	b_j^{NR-}	one_j^{+}	one_j^{-}	two_j^{-}
0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	1	1	1	0	0
0	1	0	0	0	1	1	1	0	0
0	1	1	1	1	0	-2	0	0	1
1	0	0	1	1	0	-2	0	0	1
1	0	1	1	1	1	-1	0	1	0
1	1	0	1	1	1	-1	0	1	0
1	1	1	1	0	0	0	0	0	0

$$one^{+}_{j} = \overline{n^{+}_{2j+1}} \wedge n^{+}_{2j}$$
$$one^{-}_{j} = n^{-}_{2j+1} \wedge n^{+}_{2j} \qquad (6)$$
$$two^{-}_{j} = n^{-}_{2j+1} \wedge \overline{n^{+}_{2j}}$$



(a)



(b)

Figure 3.1. $NR4SD^{-}$ Programming Scheme Block diagram (a) Single Level and (b) Multi Level

Negatively signed n^{-}_{2j+1} and positively signed n^{+}_{2j} results into the equation (5).

Step 5: Now calculate the value as $j = j+1$

Step 6: If the values $j < k-1$, then set out to Step 2. And if the value $j = k-1$, then the MSB is encoded according to the Modified Booth scheme and three successive bits are considered as b_{2k-1} b_{2k-2} and c_{2k-2} (Fig. 3.1b).

Stop if $j = k$, then NR4SD$^-$ digits are formed according to Table 3.2. Generation of encoding signals of Table 3.2 NR4SD$^-$ are shown in equations (6) .

The maximum and minimum limit of the given dynamic range in the NR4SD$^-$ algorithm are $2^{n-1} + 2^{n-4} + \text{-- - -} 2^{n-6} + \text{--+} 1 > 2^{n-1} - 1$ and $-2^{n-1} - 2^{n-3} - 2^{n-5} - \text{- - -} 2 < -2^{n-1}$ .It has been monitored that NR4SD$^-$ structure have generously proportioned active variety than the two's complement representations.

3.2.2 NR4SD$^+$ Algorithm

Step 1: let primary ideals be $j = 0$ as well as $c_0 = 0$

Step 2: The carry c_{2j+2} (+) with positive sign and the sum n^-_{2j+1} (-) with negative sign with inputs b_{2j+1} (+) and c_{2j+1} (+) (Fig3.2) of a Half Adder. The outputs c_{2j+2} and n^-_{2j+1} of the Half Adder communicate to its inputs are calculated as given:

$$2c_{2j+1} - n^-_{2j} = b_{2j} + c_{2j}$$

The Half Adder outputs are calculated at gate level by following equations

$$c_{2j+1} = b_{2j} \vee c_{2j}, \quad n^-_{2j} = b_{2j} \oplus c_{2j}$$

Step 3: The carry c_{2j+1} in addition to the sum n^+_{2j} with input values b_{2j} with c_{2j} of a Half Adder is calculated as (Fig. 3.2).

$$c_{2j+2} = b_{2j+1} \wedge c_{2j+1} \quad n^+_{2j+1} = b_{2j+1} \oplus c_{2j+1}$$

Step 4: The b_j^{NR-} digit value is calculated as follows

$$b_j^{NR+} = 2n^+_{2j+1} - n^-_{2j} \qquad (7)$$

Negatively signed n^-_{2j} and positively signed n^+_{2j+1} results into the equation (7).

Step 5: Now calculate the value as $j = j+1$

Step 6: If the values $j < k-1$, then set out to Step 2. And if the value $j = k-1$, then the MSB is according to the Modified Booth scheme and the three successive bits are considered as b_{2k-1} b_{2k-2} and c_{2k-2} (Figure 3.2b).

Stop if $j = k$, then NR4SD$^-$ digits are formed according to Table 3.3.Generation of encoding signals of Table 3.3 NR4SD$^-$is shown in equations (8)

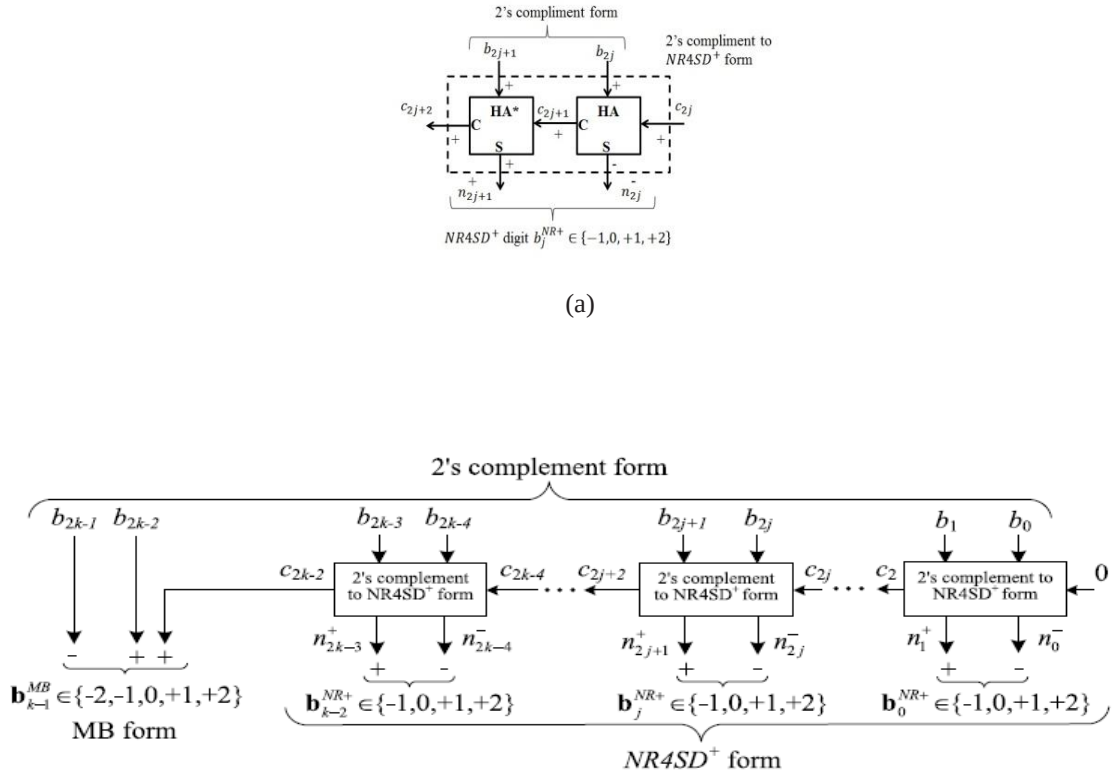
$$\begin{aligned} one^+_j &= \overline{n^+_{2j+1}} \wedge n^-_{2j} \\ one^-_j &= n^+_{2j+1} \wedge n^-_{2j} \\ two^+_j &= n^+_{2j+1} \wedge \overline{n^-_{2j}} \end{aligned} \qquad (8)$$

Table 3.3. NR4SD$^+$ Encoding

Two's complement			NR4SD$^+$ form			Digit	NR4SD$^+$ encoding		
b_{2j+1}	b_{2j}	c_{2j}	c_{2j+2}	n^-_{2j+1}	n^+_{2j}	b_j^{NR+}	one_j^+	one_j^-	two^+_J
0	0	0	0	0	0	0	0	0	0
0	0	1	0	1	1	+1	1	0	0
0	1	0	0	1	1	+1	1	0	0
0	1	1	0	0	0	+2	0	0	1
1	0	0	0	0	0	+2	0	0	1
1	0	1	1	1	1	-1	0	1	0
1	1	0	1	1	1	-1	0	1	0
1	1	1	1	0	0	0	0	0	0

The maximum and minimum restrictions of the given active variety in the NR4SD$^+$ algorithm are-2^{n-1} -2^{n-3} - 2^{n-5} - - - 2 < - 2^{n-1} as well as 2^{n-1} + 2^{n-4} + -- - - 2^{n-6} +--+ 1 > 2^{n-1} – 1and.It has

been monitored that NR4SD$^-$ form has generously proportioned active variety than the two's complement representation.



(a)

(b)

Figure 3. 2. Encoding Scheme of NR4SD$^+$ Block Diagram (a) Digit Level And (b) Word Level.

3.3 PRE-ENCODED MULTIPLIER DESIGN

Either Modified Booth or NR4SD$^-$/NR4SD$^+$ scheme is used for pre-encoding of one of the two inputs. The set of fixed coefficients is used for forming inputs. Using Modified Booth or NR4SD algorithms where co-efficient are encoded offline and the ROM is used to store the resulting bits.

3.3.1 Conventional Multiplier Design

The ROM consists of coefficients in two's complement form and the system architecture comprises of the conventional Modified Booth multiplier. Let us multiply the two numbers A x B. The coefficient of number B =$(b_{n-1} \ldots b_0)_{2's}$ have n = 2k bits which is stored in ROM in two's complement form and is determined to the Modified Booth encoding blocks. The encoding of the digits is according to the Modified Booth algorithm and multiplied by the two's complement representation of A = $(a_{n-1} \ldots a_0)_{2's}$.

The generation of K partial product term is given as follows:

$$PPj = A.b_j^{MB} = p_{j,n}2^n + \sum_{i=0}^{n-1} p_{j,i}2^i \qquad (9)$$

The generation of the ith bit $p_{j,i}$ of partial product term PPj is demonstrated at gate level in Figure3.4a. We consider $a_{-1} = 0$ and $a_n = a_{n-1}$ respectively for the calculation of the LSB and MSB of PPj.

Once the partial products are determined, further addition, correctly subjected, by a tree formed by carry save adder all along side through correction term:

$$P = A.B = COR + \sum_{j=0}^{k-1} PP_j 2^{2j} \qquad (10)$$

$$COR = \sum_{j=0}^{k-1} c_{in,j} 2^{2j} + 2^n (1 + \sum_{j=0}^{k-1} 2^{2j+1}) \qquad (11)$$

where $c_{in,j} = (one_j \vee two_j) \wedge s_j$ (table 3.2). The Cary Save output tree is further fed to a fast carry look ahead adder and the final result is obtained $P = A.B$ (fig 3.3)

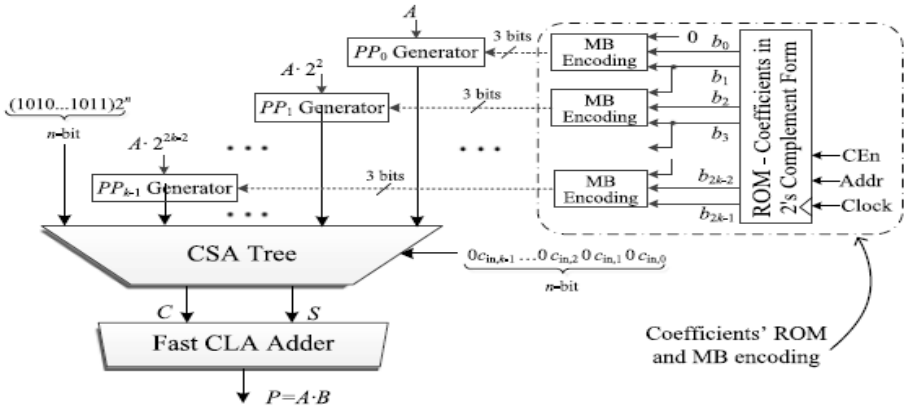


Figure 3.3 Conventional Modified Booth Multiplier System Architecture [19]

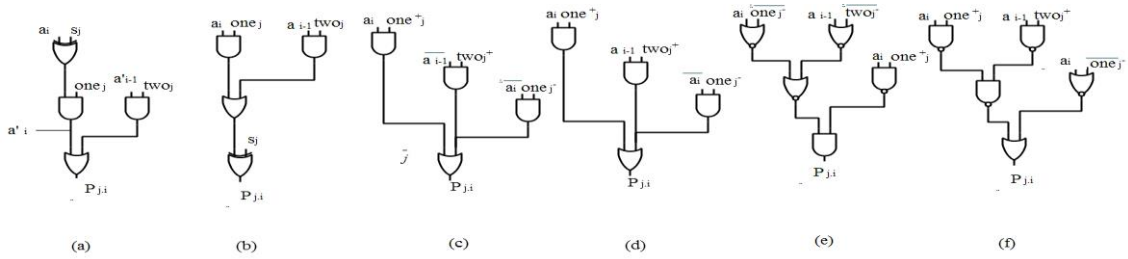


Fig.3.4. Ith Bit Generation $P_{j,I}$ Of The Ppj A) Conventional B) Pre-Encoded Modified Booth Multipliers, C) Non-Redundant Radix 4Signed Digit^{-}, D)Non-Redundant Radix 4 Signed Digit^{+} Multipliers, And E) Non-Redundant Radix 4 Signed Digit^{-}, F) Non-Redundant Radix 4 Signed Digit^{+} After Reconstruction Of Pre-Encoded Multipliers [19]

3.3.2 Design of Pre-encoded multiplier

In the pre-encoded design the encoding of coefficient B is off-line. Modified Booth multiplier scheme, according to the conventional Modified Booth structure (Table 1).Encoding signals of the result of B are kept in ROM. In Fig. 3.3 the part which is circled, enclosed the ROM with coefficients stored in two's complement representation as well as the Modified Booth encoding circuit, is now completely omitted with the ROM of Figure3.5. The Modified Booth encoding blocks of Fig. 3.3is removed. The encoding signals of B are store in new ROM and then provide them into partial product generators

on each clock put. To decline switching action, the last entry of Table 3.1 value ‘1’ of sj is replaced by ‘0’. The sign sj is now given by the following equation:

$$s_j = b_{2j+1} \oplus (b_{2j+1} \wedge b_{2j} \wedge b_{2j-1}) \quad (12)$$

The Partial Product Generation of Figure3.4a is removed by the individual of Figure3.4b. The Carry Save Adder tree is fed the COR of (11) and with properly weighted partial products. The computation of input carry $c_{in,j}$ in (11) as $c_{in,j} = s_j$ that is depends upon (12) and Table 3.3. The Carry Save output is at last combined by a speedy Carry Look Ahead Adder. The Width Of The ROM Is Increased.

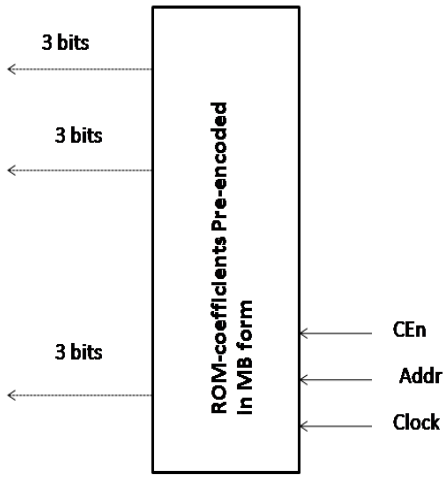


Figure 3.5 Pre-Encoded Multiplier Scheme Rom With Standard Coefficients In Modified Booth Form. [19]

3.3.3 Pre-Encoded NR4SD Multiplier Design [19]

In Figure3.6architecture of the pre-encoded Non-Redundant Radix-4Signed Digit multipliers is represented. In ROM now two bits are stored: n_{2j+1}^{-}, n_{2j}^{+} (Table 3.3) for the NR4SD^{-} or n_{2j+1}^{+}, n_{2j}^{-} (Table 3.3). The memory necessity to n + 1 bits for each coefficient is reduced. The MSB to facilitate needs an additional bit as it is encoding is Modified Booth otherwise; the sum is equivalent so as to of the conventional Modified Booth propose in case of stored bits. NR4SD multipliers required extra hardware to produce the signals of (6) as well as (8) for the given encoded multipliers. In Figure3.6 the NR4SD encoding blocks circuitrary is implemented in the Figure3.7.

Figure 3.4c and 3.4d represents the implementation of pre-encoded NR4SD^{-} as well as NR4SD^{+} multiplier's partial product correspondingly, except for the PP_{k-1} that correspond to the mainly noteworthy digit. This digit is in Modified Booth form, the Partial Product Generation of Fig. 3.4b by applying the modification as mentioned in Section 3.3.2 for the s_j bit. The properly weighted partial products and the Correction term generated by equation (11) both are embed into a Carry Save Adder tree. The calculated of input carry $c_{in,j}$ of (11) is as $c_{in,j} = two_j^{-} \vee one_j^{-}$ and $c_{in,j} = one_j^{-}$ for the Non-Redundant Radix-4Signed Digit^{-} and Non-Redundant Radix-4Signed Digit $^{+}$ pre-encoded multipliers, correspondingly, based on Tables 3 and 4. The carry-save output of the Carry Save Adder

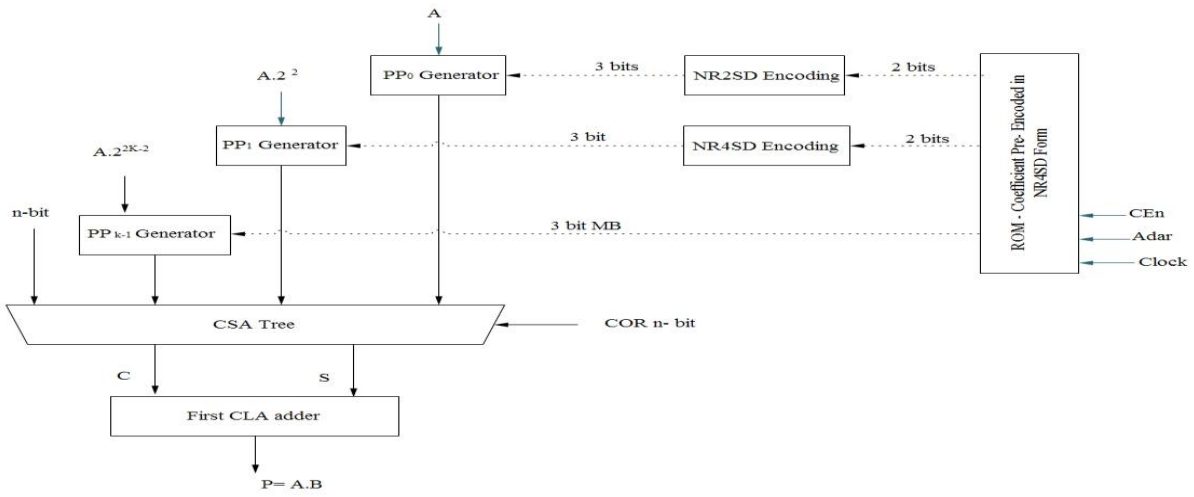


Figure 3.6 NR4SD Multiplier System Architecture [19]

tree is finally summed using a fast Carry Look Ahead adder.

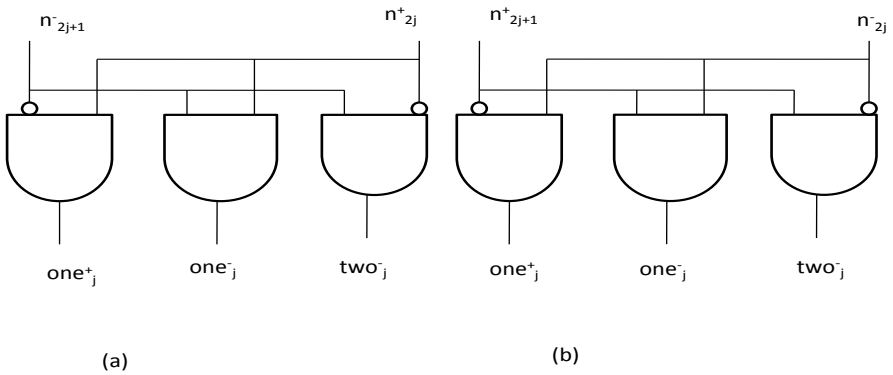


Figure 3.7. NR4SD Multipliers Extra Circuitry (A) NR4SD^{-} And (B) NR4SD^{+} Encoding.[19]

CHAPTER 4

PRE-ENCODED MULTIPLIERS USING CARRY SKIP ADDER

In the previous design we studied the pre-encoded multiplier using carry save adder and carry look ahead adder. But in proposed system, the last two stages of carry save adder and carry look ahead adder has been replaced with carry skip adder. The Partial Product Generation unit is also based upon nand gates. In proposed system the delay of the architecture design is reduced by replacing the CSA and CLA adder to carry skip adder[17][18] and use the NAND gate based PPG unit.

In Figure4.1 architecture of the NR4SD pre-encoded multipliers is represented. In ROM now two bits are saved: n_{2j+1}^{-}, n_{2j}^{+} (Table 3.3) for the NR4SD^{-} or n_{2j+1}^{+}, n_{2j}^{-} (Table 3.3). Within this way the memory necessity to n + 1 bits for each coefficient is reduced. The MSB to facilitate needs an additional bit as it is encoding is Modified Booth otherwise; the sum is equivalent to that of the conventional Modified Booth propose in case of stored bits[16]. Non-Redundant Radix-4Signed Digit multipliers required extra hardware to produce the signals of (6) as well as (8) for the pre-encoded multiplier form. In Figure3.6 the NR4SD encoding blocks circuitrary is implemented in the Figure 4.3.

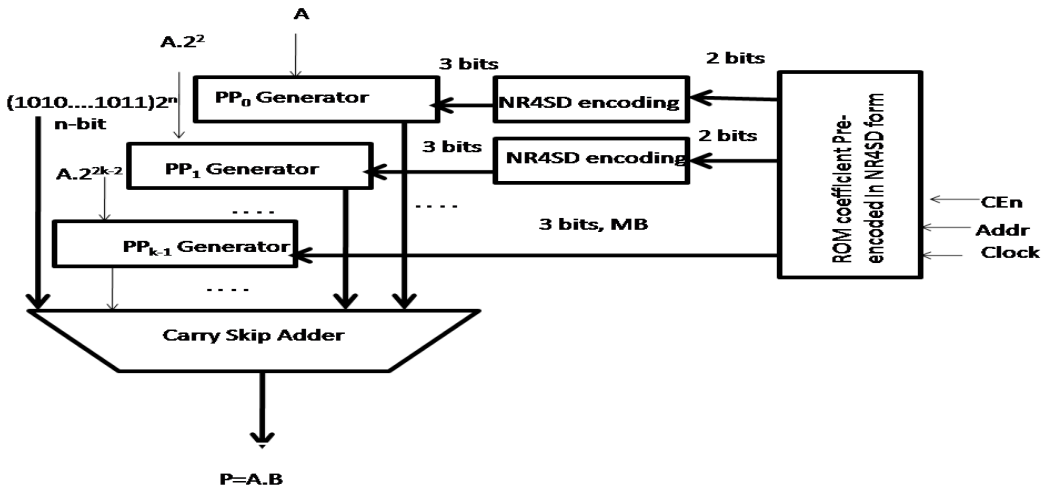


Figure 4.1: Projected NR4SD Multipliers System Architecture

Figure 9(a) and 9(b) are used to implement every partial product of the multipliers that are pre-encoded implemented correspondingly, except for the most significant digit. PP_{k-1}.Due to the encoding of this digit in Modified Booth form, the Partial Product Generation of Figure 4b applies the modification for the s_j bit describe in Section 3.3.2. The properly weighted partial products as well as Correction expression in (11) be embedded into a Carry Save Adder tree. The calculation $c_{in,j}$ carry input of (11) is as $c_{in,j} = two_j^- \vee one_j^-$ and $c_{in,j} = one_j^-$ for the pre-encoded multipliers NR4SD$^-$ andNR4SD$^+$correspondingly, based on Tables 3.2and 3.3. And finally the outputs of PPG unit are finally summed by means of carry skip adder.

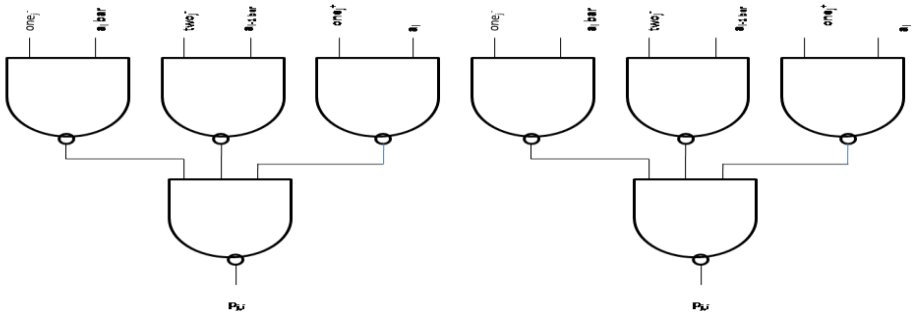


Figure 4.2: Ith Bit Generation $P_{j,I}$ Of Ppj In Proposed Design

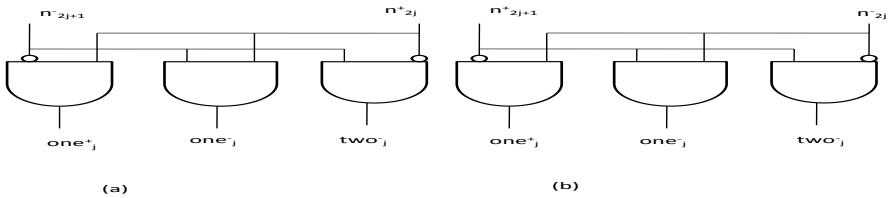


Figure 4.3: NR4SD multiplier's extra circuit the (a) NR4SD$^-$ and (b) NR4SD$^+$ encoding [19]

In proposed system reduce the delay of the architecture design by replace the CSA and CLA adder to carry skip adder and use the NAND gate based PPG unit.

CHAPTER 5
INTRODUCTION TO VLSI DESIGN

In this chapter the FPGA ideas and Synthesis Flow of FPGA are presented. Tool that comprises of large number of transistors associated to implement logical functions is known as FPGA.

5.1 WHAT IS VLSI?

VLSI acronym for "Very Large Scale Integration" involves packaging of large number of logic circuit devices into compact areas. VLSI technologies are used now and then in our daily lives whether starting from our cell phones, transport systems and whatever we are using. Different expertise sections of VLSI are discussed in the coming sections.

5.1.1 Dealing with VLSI Circuits

Implementation of different blocks like the gate system and latch system are different, although the behaviour of operation remains the same. Many new considerations are undertaken for new implementations of the circuit.

Circuit Delays: these are large complicated circuits running at very high frequencies dealing with the issue in delay of the propagation of signals through gates and wire even for very small distance areas. The operating speed of the system is so fast that they actually become comparable to the clock speeds when all the delays add up.

Power: With the increase in the operation of high range of frequencies, the consumption of power increases which however leads to more heat dissipation and faster rate of discharging of batteries. Such issues create a major threat to the circuit stability.

Layout: There are different ways to layout the circuit components. Some of them are: layout multiple layers of different materials on the same silicon, having different arrangements of the smaller parts for the same component and soon. Layout can also affect the fabrication of VLSI chips, making it either easy or difficult to implement the components on the silicon.

5.1.2 **Verilog-HDL**

- Verilog HDL is one of the standard hardware description languages which is easy to learn and use which offer many useful and unique features for hardware designing.
- Its coding syntax is similar to the C programming language which helps the C language designers to understand and use Verilog HDL language.
- Verilog HDL helps designers to learn only one language for stimulus and hierarchical design as it allows the mixing of different levels of abstraction in the same model. Thus, a designer can define a hardware model in terms of switches, gates, RTL, or behavioural code depending on the accessibility of the system.
- Verilog HDL is one of the language choice for designers as it supports most of the logic synthesis tools. Thus, to design a chip in Verilog HDL allows the widest choice of vendors for post logic synthesis simulation.
- The Programming Language Interface (PLI) interfaces the user written custom C code with the internal data structures of Verilog. Designers can customize a Verilog HDL simulator to their needs with the PLI.

With the usage of Verilog HDL, the speed and complexity of digital circuits has increased rapidly that has influenced designers to design at higher levels of abstraction. With designer assistance, CAD tools are used for the implementation details. Behavioural modelling will be used more and more as behavioural synthesis matures. Until then, RTL design will remain very popular. Formal verification techniques are also appearing on the horizon. Formal verification applies formal mathematical techniques to verify the correctness of Verilog HDL descriptions and to establish equivalency between RTL and gate level net lists. However, the need to describe a design in Verilog HDL will not go away. The net list generated at gate level during logic synthesis tools is not optimal for high speed and critical timing circuits. Therefore for achieving optimum results designers generated mix gate-level description directly into the RTL description. The high-speed designs generally uses this scheme because designers required squeezing of the last bit of timing out of circuits and CAD tools sometimes prove to be insufficient to achieve the desired results. Thus, the system-level simulation for the CPU could be up and running very quickly and long before the RTL descriptions for the graphics chip and the I/0 chip are completed.

Field-Programmable Gate Array

A field-programmable gate array (FPGA) is a semiconductor device that can be configured by the customer or designer after manufacturing.

Logic blocks known as programmable logic components rely on FPGA and allowed the blocks to be "wired together" a hierarchy of interconnects that are reconfigurable to some extent like breadboard programmable on one-chip. To perform complex combinational functions logic blocks are configured. Memory elements, which may be simply either flip-flops or complete blocks of memory are included in most of the FPGAs.

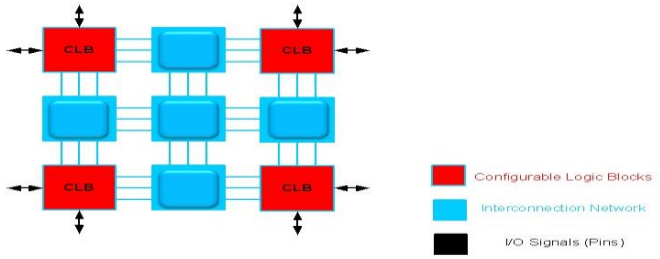


Figure 5.1: FPGA Architecture

Figure 5.1 illustrates the basic architecture of FPGA which consists a matrix of CLBs (Configurable Logic Blocks) that are interconnected by an array of switch matrices

Another basic difference between a FPGA and a CPLD is the storage of the inter connects. While CPLDs are non-volatile (that is, they make use of anti fuse, EEPROM, Flash, etc.).

Figure 5.2 : Examples of FPGA Packages

Technology Trends in FPGA

- Common development is better and quicker.
- By increasing the density of device even by less significant fabrication progression technology.
- All function is performed incomplete systems on a solitary device.
- Characters like RAM, devoted arithmetic hardware, clock management as well as transceivers are existed in main programmable logic.
- Embedded processors also make use of FPGA.

5.2 XILINX SPECIFICS

All Xilinx FPGAs consists of following basic properties

1. Configurable logic blocks.
2. Input and output blocks.
3. Random Access Memory blocks.
4. Programmable Interconnections.

5. Former properties similar to clock buffers, tri-state buffers.

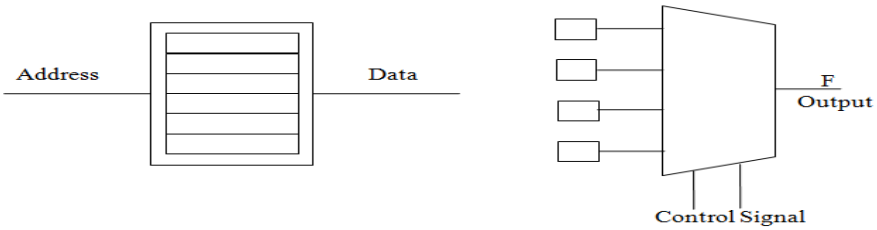


Figure 5.3: LUT Execution as (a) Memory (b) Multiplexers

5.2.1 Configurable Logic Blocks

The main building block of Configurable Logic Blocks in Xilinx is Slice. Four slices are hold by per Configurable Logic Block in Spartan III. The two function generators of four-input, carry logic as well as two storage elements are placed in every slice. The output of each function generator drives both flip-flop's D-input and the CLB output. The characteristics of storage elements and look-up tables are given below:

1) Look-Up Tables -Another key feature is logic functions implementations in FPGA. Look-up tables are carry out logical functions implemented as memory in logical blocks or and memory. These alternatives are shown together in memory contents for some basic operations as in figure 5.3.A n-bit function is implement by any 2n*1 ROM. An LUT consists of n-bit is implement as a memory of 2n*1as shown in figure 5.3(a). The2n memory locations select one as input address. User's configuration bit streams are used for loading memory locations. In figure 5.3(b) the LUT inputs are the multiplexer control inputs. As a result logic gate is formed. Any n-bit function is implemented with Look Up Tables.
2) Storage Elements-An edge-triggered D-type flip- flops or a latch that is level-sensitive can be configured as the storage elements in a slice. The function generators are used for driving D-input directly from the slice inputs or within the slice are bypassed. Both set and reset signals are synchronized as clock and clock enable signals

5.3 FPGA IMPLEMENTATION USING XILINX

For the implementation of the circuit Xilinx Spartan 6E (Family), XC3S5000 (Device) FPGA is used. Xilinx ISE 14.2i is the working tool .

5.3.1 FPGA Design Flow Overview

Complication increases as the FPGA architecture evolves. A moderately absolute set of design tools that allow automatic synthesis are provided by most of the FPGA vendors and design specifications compilation. The steps and components of typical design flow of FPGA is shown in Figure 5.4.Some explicit pairs of registers contain delays that may be inhibited. There is a wide range of FPGA devices available at FPGA vendors which are specified according to cost, diverse performance as well as power tradeoffs. There is an iterative process for assortment of target device. Since quality directly impacts the cost and performance of FPGA so there is a need to have improved synthesis tools. To check the functionality of design on actual hardware FPGA implementation is done. The design series time and implementation cost of ASICs is high for that reason the superior designs are former checkered on FPGA and if the results are satisfactory then implementation of ASIC design is done.

The diverse steps implicated within the FPGA implementation is shown in figure 5.4.Design entries are done all the way through HDL and also include logic synthesis behavior simulation, design implementation, generation of bit stream and to conclude the FPGA programming.

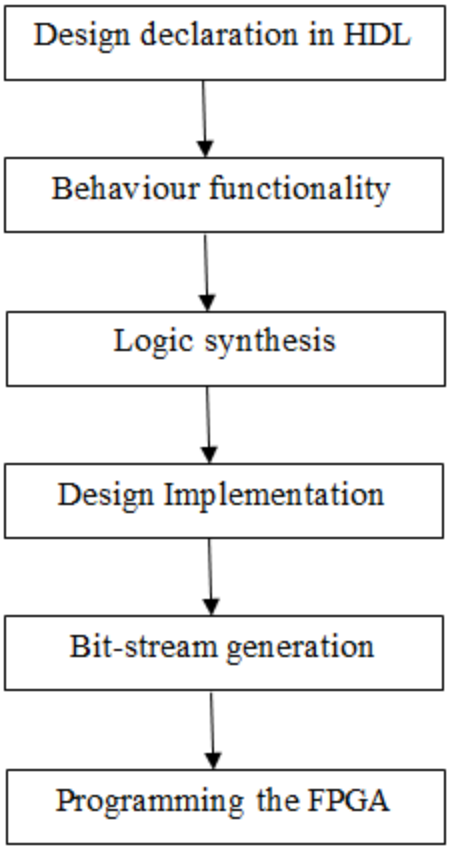


Figure 5.4: FPGA Design Flow

All the compulsory functions required for implementation of a design on FPGA is provided by Xilinx. For timing and functional simulation the ISIM simulator required is available in Xilinx ISE suite. The diverse steps implicated in FPGA execution with Xilinx are given below:

Design Entry

The fundamental structural design for the scheme is deliberated in the step that is coding in a HDL like VHDL or Verilog. The external interface to the design unit is represented by module statement. The interior explanation of the design unit-its structure, its behavior, or a mixture of both is represented by interior module.

Behavioral Simulation

To verify the Verilog code functionality of the unit a test bench is generated following the design phase using simulation software for singular inputs toward generated outputs as well as if it is verified that then headed additional, or else changes are required corrections will be completed in the Hardware Descriptive Language code and known as behavioral simulation.

Design Synthesis

The design is synthesized following the accurate simulations outcome. The operations that are performed during synthesis of the Xilinx ISE tool are: HDL Compilation: If there is any sub-modules then compilation of every sub-modules is done within the main module and then syntax is checked of the code designed for the design.

1) In design hierarchy the analysis of design is done.

2) HDL Synthesis: The procedure that converts Verilog code keen on a device net list format. If more than one sub-design exists in the design then for each design element the synthesis process generates net list for processor's implementation.

3) Advanced HDL Synthesis: The synthesized block during the Hardware Descriptive Language synthesis are additionally distinct in conditions of the low level blocks like as lookup tables and buffers. The 'net list' file (NGC file) is then generated by tool and then optimized. The ending net list production file has NGC extension. Both the constraints and design data are included in NGC file.

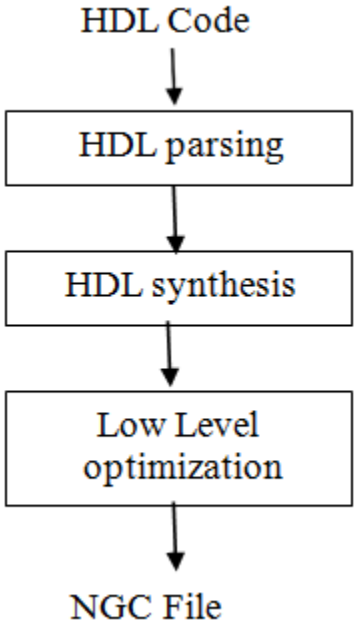


Figure 5.5: Steps In Synthesis Process

Design Implementation

The following sub processes occurs in implementation of design process

1) Translation: All the design constraints as well as input net lists are collective and a file is saved known as native generic database file. In translate process timing requirements are specified. By modifying we can change the translate properties.

2) Mapping: Mapping process is done after the translation process. The sub-blocks are arranged in mapping process. In FPGA sub-blocks the sub-blocks that are made in mapping process is fitted. A native circuit description file is then generated. In this file our design is mapped into FPGA components.

3) Place and Route: Logic blocks are formed from Sub-blocks of mapping procedure and associated in place and route stage. Input is taken as NCD file and this file is routed. Hence placement as well as block's routing is done here.

4) Bit stream Generation: For exacting Xilinx device the routing of NCD file generation for bit file is done. The output bit file contains binary bits necessary to program the device. For programming the FPGA device use bit file that is previously generated is used.

5) Functional Simulation: Functional simulation which is also known as Post-Translate be able to be execute earlier to the design mapping. For verification of the designs that have been accurately synthesized and several variations owed in the direction of the inferior stages for generalization be able to recognize this simulation method is used.

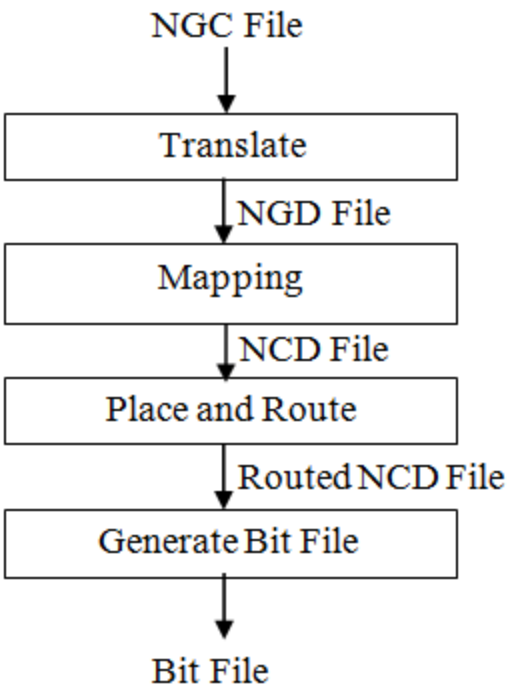


Figure 5.6: Different Files Generated In Implementation Process

CHAPTER 6

SIMULATION RESULTS AND SYNTHESIS

The implementation of Pre-encoded NR4SD multipliers using carry skip adder, and their synthesis and simulation results is discussed in this chapter. Timing constraints and actual hardware utilization of the design is described in synthesis report. The design's behavioral functionality of is described in simulation results. To relocate the design on to actual hardware design's implementation is done. The Xilinx ISE 14.5 software is intended for the simulation, synthesis as well as implementation in targeting Spartan -6E FPGA. The operational situation intended the entire designs is

- Version of tool : ISE 14.5
- Goal optimization: rapidity
- Design Strategy: Balanced
- Family of device: Spartan 6E
- Device: XC6SLX45
- Package: CSG324
- Simulator: ISIM

6.1 NR4SD MULTIPLIERS

The proposed design is implemented in Verilog using table 6.1. The PPGs for the $NR4SD^{-}$ and $NR4SD^{+}$ multiplier (Figs 3.4c and 3.4dcorrespondingly) contains a bulky figure of inverters because in case of negative digit every bit of A bits is complemented. For the reduction of the delay of Non Redundant Radix 4Signed $Digit^{-}$pre-encoded and Non Redundant Radix 4 Signed $Digit^{+}$ pre-encoded multipliers, these inverters are avoided and thus the PPGs intended for the Non Redundant Radix 4Signed $Digit^{-}$ and

Non Redundant Radix 4 Signed Digit^{+} multipliers be designed on primitive NAND as well as NOR gates and replaced by 3.4e and 3.4f correspondingly. Further to reduce the delay the Partial Product Generators for the NR4SD^{-} and NR4SD^{+} multipliers are premeditated only using primal NAND gates. The Carry Skip Adder is also designed using verilog.

Table 6.1 Comparison of Different Designs

Design		Input A	Input B encoding		ROM width
			Type	Technique	
Conventional MB		2's complement n-bit	MB	MB encoding	n- bit
Pre-encoded			MB	Fully Pre-encoded	3n/2- bit
			NR4SD^{-}	Partially pre-encoded	(n+1)- bit
			NR4SD^{+}	Partially pre-encoded	(n+1)- bit

The two's complement or coefficient's that are pre-encoded in ROM is a synchronous 512 words ROM frequently used in Digital Signal Processing system. The ROM width depends upon the architecture multiplier (Table 6.1).

6.1.2 NR4SD- Multiplier Simulation Result

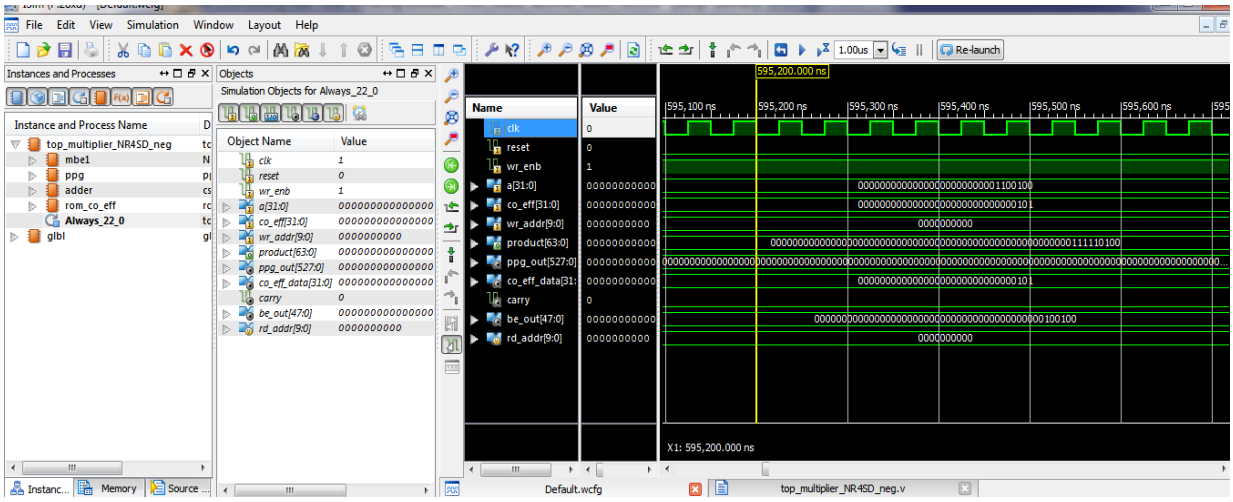


Figure 6.1 Simulation Result Of NR4SD- Algorithm

Synthesis Report

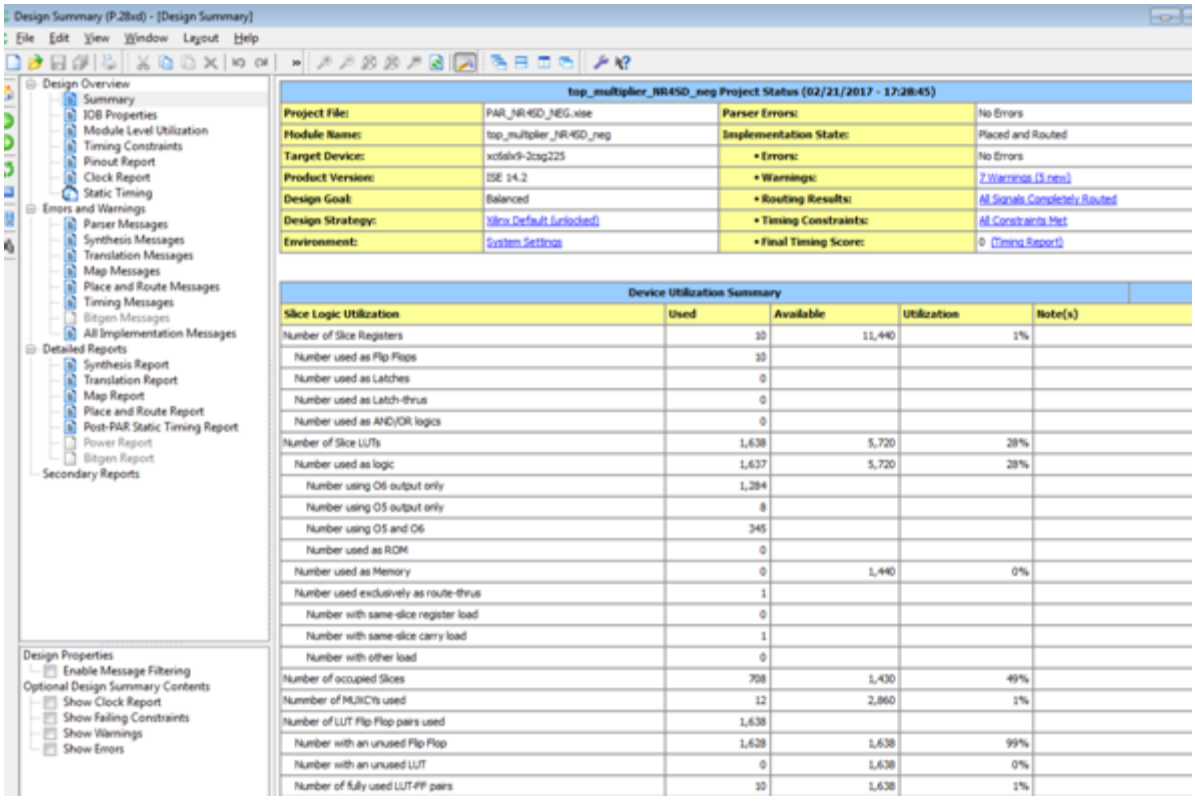


Figure 6.2 Synthesis Result Of NR4SD- Algorithm

Timing Report

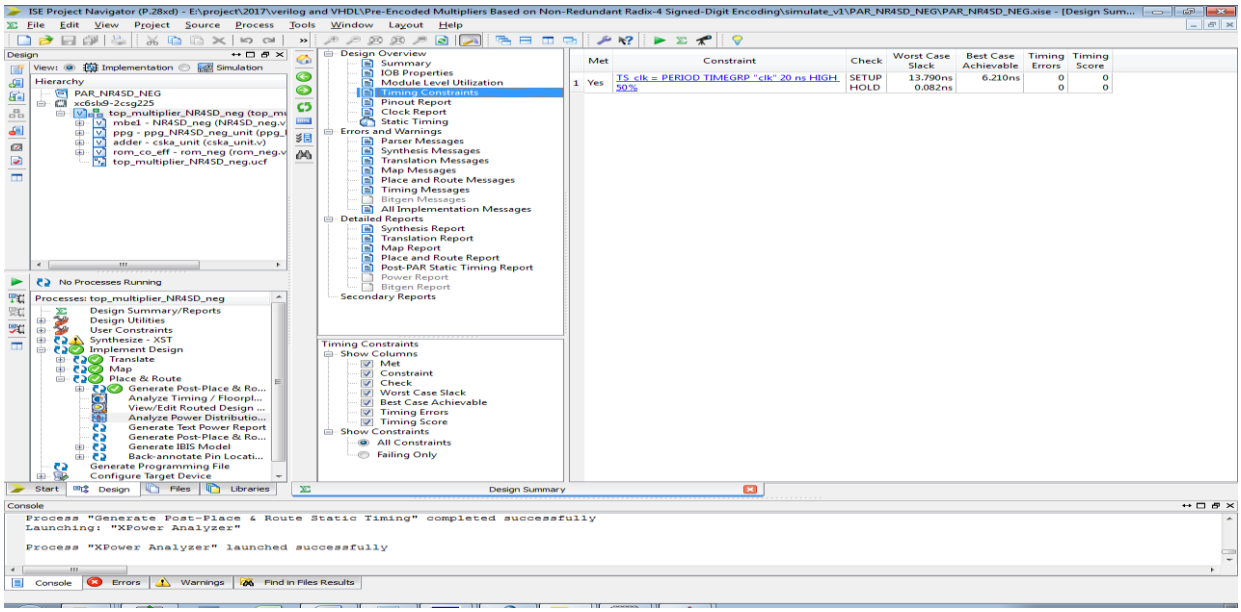


Figure 6.3 Timing Report Of NR4SD- Algorithm

Delay Report

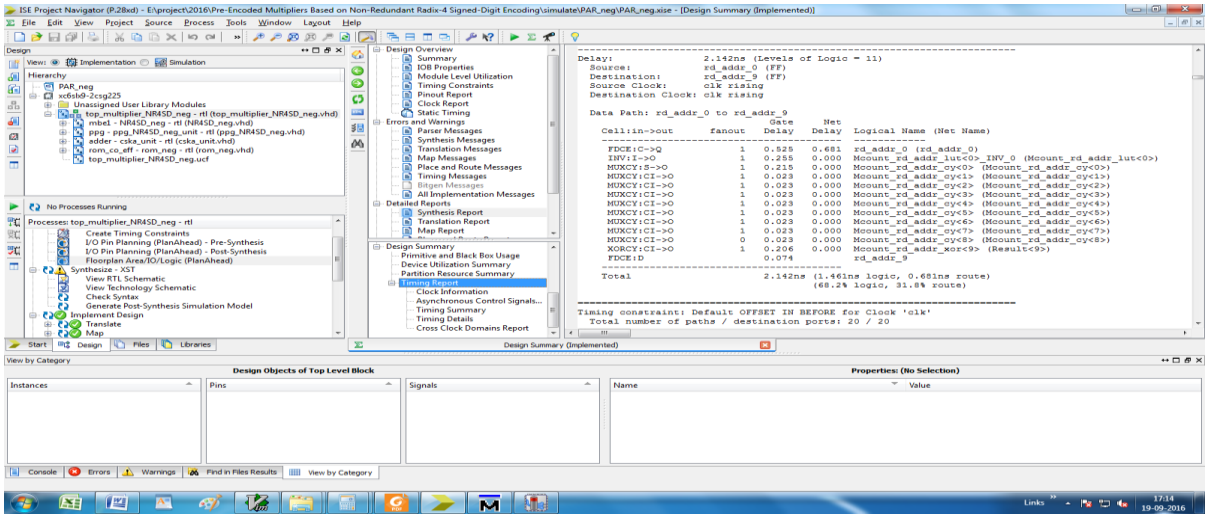


Figure 6.4 Delay Reportof NR4SD- Algorithm

RTL View

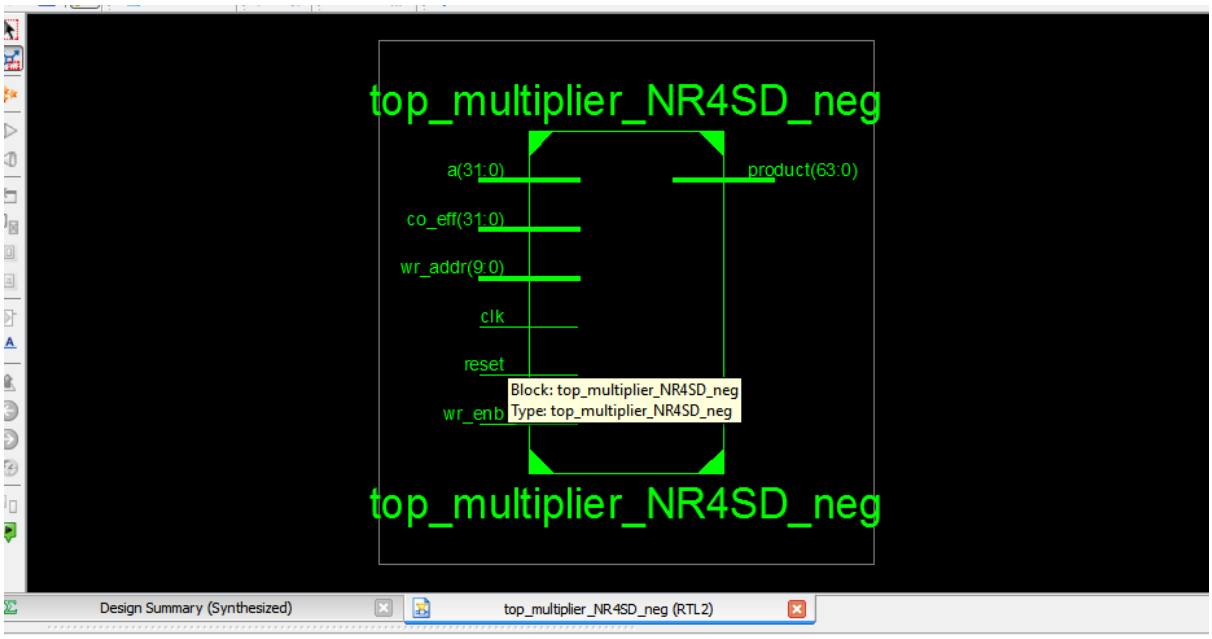


Figure 6.5 RTL View Of NR4SD- Algorithm

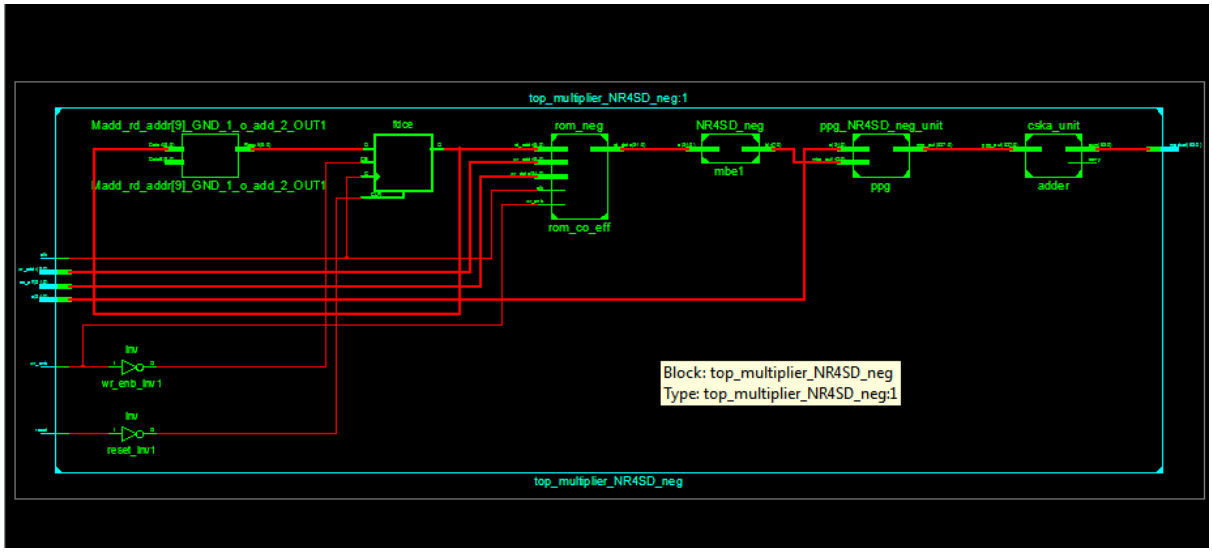


Figure 6.6 RTL View Of NR4SD- Algorithm

NR4SD^{+} Simulation Result

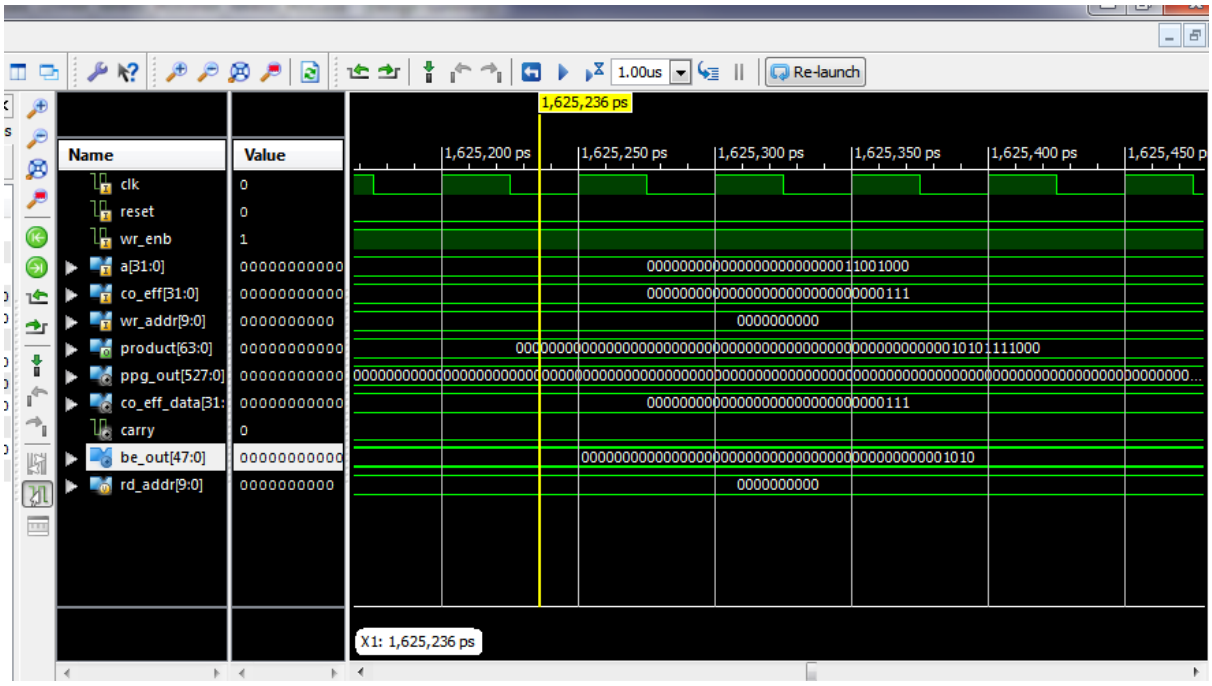


Figure 6.7 Simulation Result Of NR4SD^{+} Algorithm

Synthesis Report

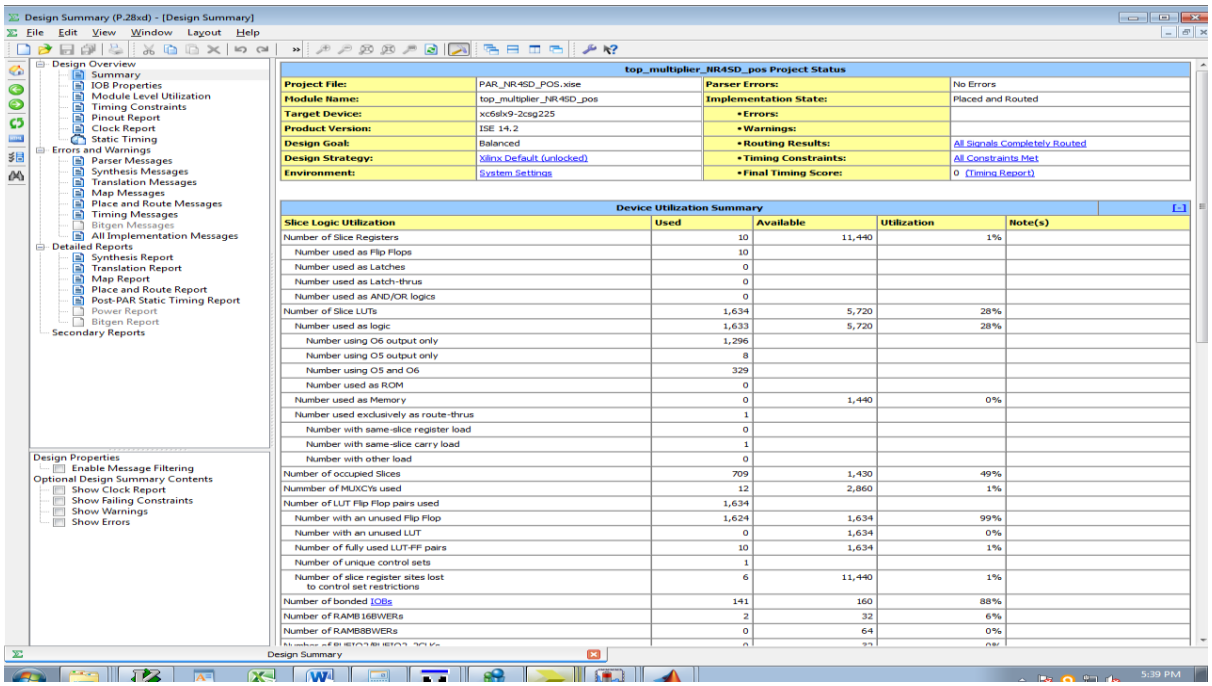


Figure 6.8 Synthesis Result Of NR4SD^{+} Algorithm

Timing Report

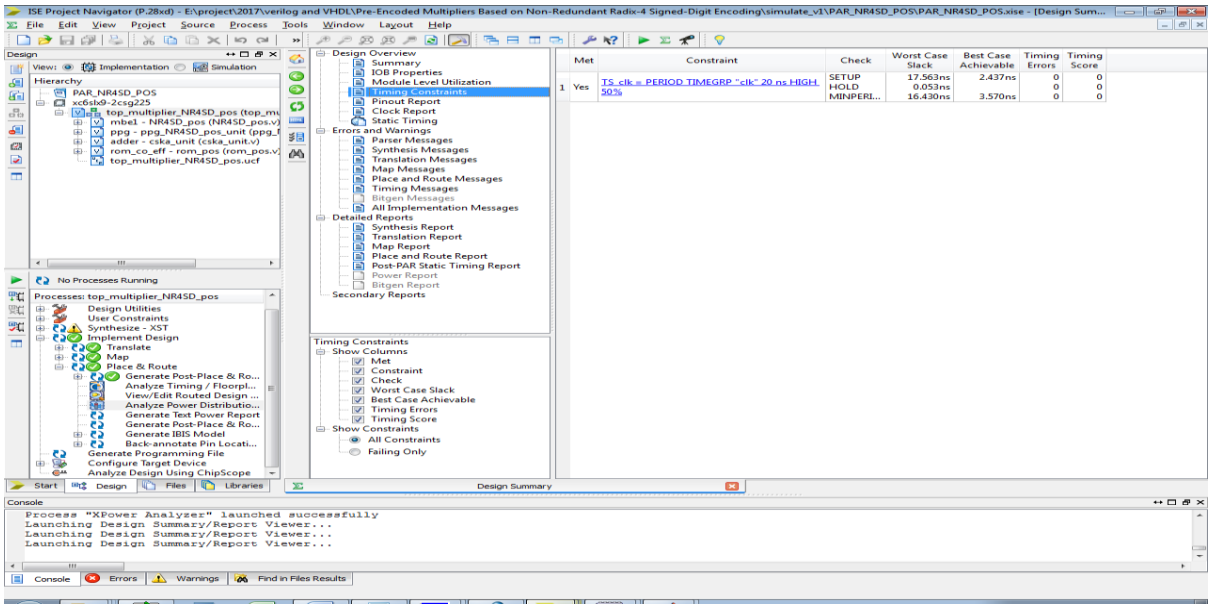


Figure 6.9 Timing Report Of NR4SD^{+} Algorithm

Delay Report

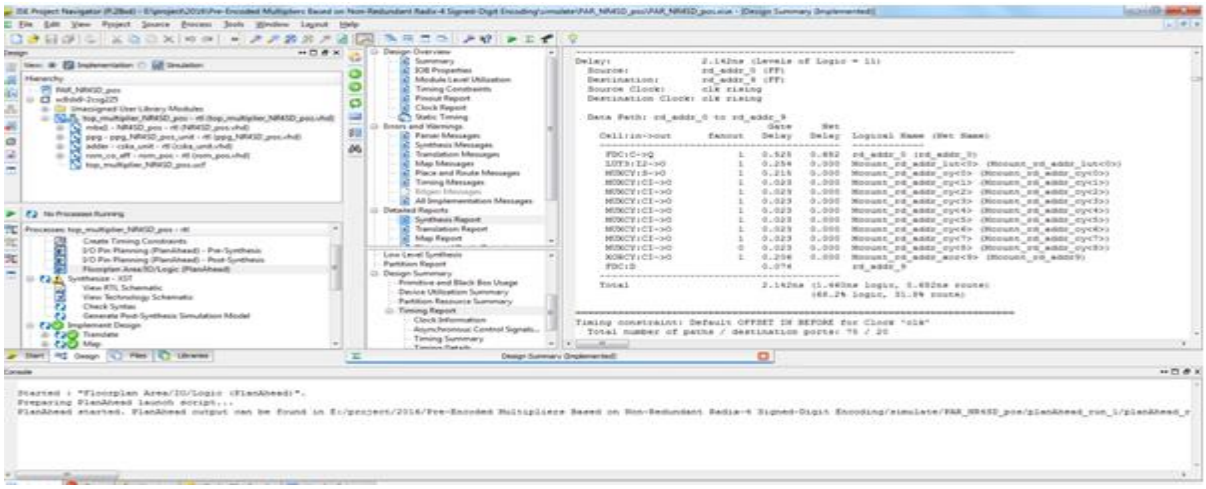


Figure 6.10 Delay Report Of NR4SD^{+} Algorithm

RTL View

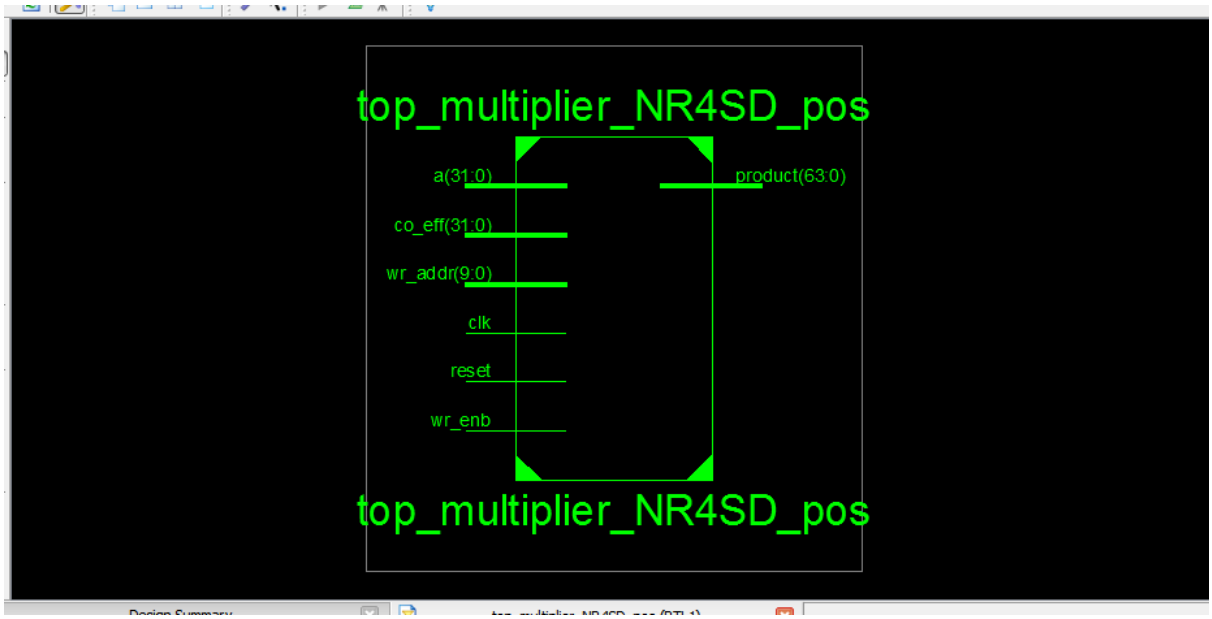


Figure 6.11 RTL View Of NR4SD^{+} Algorithm

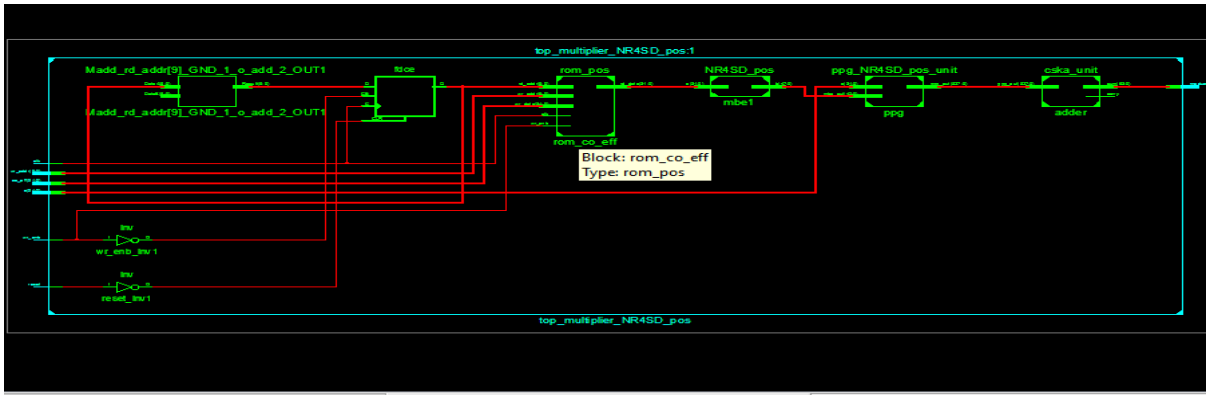


Figure 6.12 RTL View Of NR4SD^{+}Algorithm

Comparison of Results

Table 6.2 Comparison Table Between Existing And Proposed Work Done

Parameter	Existing system		Proposed system	
	NR4SD+	NR4SD-	NR4SD+	NR4SD-
Delay(ns)	2.29	2.28	2.142	2.142

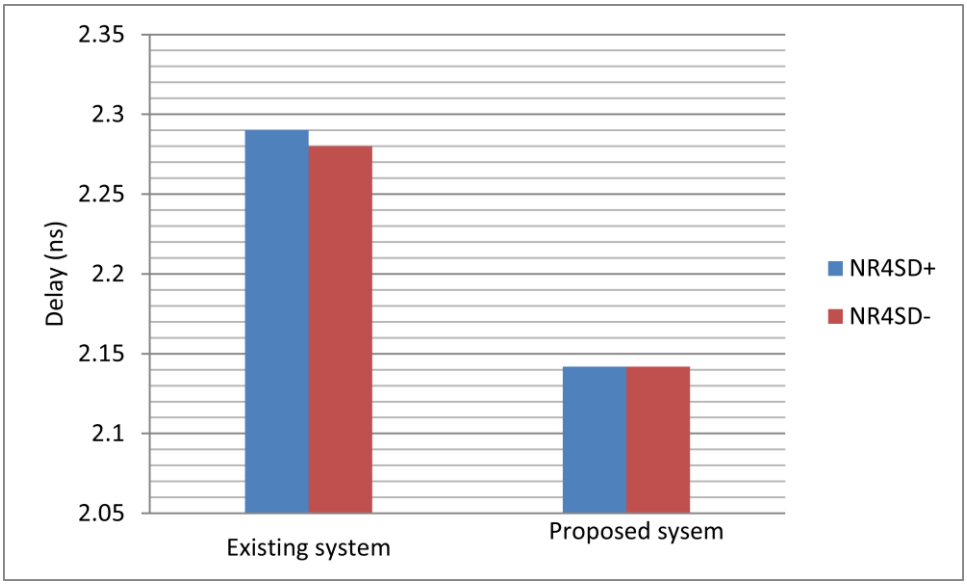


Figure 6.13 Delay Comparison

CHAPTER 7
CONCLUSION AND FUTURE SCOPE

7.1 CONCLUDING REMARKS

With the increasing demand of high speed multipliers either embedded or on die new technologies have to be proposed. These new technologies should fulfil the demand of high speed and lesser area. They have to be energy or power efficient and should have low manufacturing cost. Since the introduction of Pre-encoded multipliers many research efforts have been put in using it for various applications but its major scope lies in architecture with high speed.

The research work in the thesis presents Pre-encoded multipliers which have less complex partial product implementation. Based on the simulation results, it is concluded that these multipliers are more area and power efficient. A vast improvement in the speed of operation of the circuit is also observed which further improves the circuit performance. Also the observed area, which is an important aspect for achieving high density, is quite less as compared to previous quoted work.

7.2 FUTURE PROSPECTS

The present work on Non-Redundant Radix 4 Signed Digit Encoding using Carry Skip Adder can be extending in various instructions. Several suggestions are given as follows. Various encoding techniques be able to be analysed for optimizing the speed and area.

1. In place of Carry Skip Adder, different adders could be used to enhance the performance.
2. Different Partial Product Generation units can be used for addition of partial products as a result delay be able to be reduced.
3. In order to enhance the performance higher order compressors like 17:2 can be used to accumulate the partial products.

REFERENCES

[1] O. Macsorley, "High-speed arithmetic in binary computers," *Proc. IRE*, vol. 49, no. 1, pp. 67–91, Jan. 1961.

[2] M. D. Ercegovac and T. Lang, *"Multiplication," in Digital Arithmetic*. San Francisco: Morgan Kaufmann, 2004, pp. 181–245.s

[3] *Z.* Huang, "*High-level optimization techniques for low-power multiplier design,"* Ph.D. dissertation, Department of Computer Science, University of California, Los Angeles, CA, 2003

[4] Simmonds *et.al* "Precomputation-based sequential logic optimization for low power," in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 2, no. 4, pp. 426-436, Dec. 1993.

[5] D. Villeger and V. G. Oklobdzija, "Evaluation of Booth encoding techniques for parallel multiplier implementation," in *Electronics Letters*, vol. 29, no. 23, pp. 2016-2017, 11 Nov. 1993.

[6] L. Ciminiera and P. Montuschi, "Carry-save multiplication schemes without final addition," in *IEEE Transactions on Computers*, vol. 45, no. 9, pp. 1050-1055, Sep 1996.

[7] S. C. H. Sung-Chul Han *et al*., "An ASIC implementation of the mpeg-2 audio decoder," 1996. Digest of Technical Papers, International Conference on Consumer Electronics, 1996, pp. 192-.

[8] M. Kolluru, "Audio decoder core constants ROM optimization," Patent US 6 108 633, Aug., 2000

[9] A. Goldovsky *et.al*, "Design and implementation of a 16 by 16 low-power two's complement multiplier," 2000 *IEEE International Symposium on Circuits and Systems*. Emerging Technologies for the 21st Century. Proceedings (IEEE Cat No.00CH36353), Geneva, 2000, pp. 345-348 vol.5.

[10] W.-C. Yeh and C.-W. Jen, "High-speed Booth encoded parallel multiplier design," *IEEE Trans. Comput.*, vol. 49, no. 7, pp. 692–701, Jul. 2000.

[11] Nan-Ying Shen and O. T. C. Chen, "Low-power multipliers by minimizing switching activities of partial products," 2002 *IEEE International Symposium on Circuits and Systems. Proceedings* (Cat. No.02CH37353)*, 2002, pp. IV-93-IV-96 vol.4*.

[12] J. Park, K. Muhammad, and K. Roy, "High-performance FIR filter design based on sharing multiplication*," IEEE Trans. Very Large Scale Integr. (VLSI) Syst*., vol. 11, no. 2, pp. 244–253, Apr.2003.

[13] C. Wang, W.-S. Gan, C. C. Jong, and J. Luo, "A low-cost 256-point FFT processor for portable speech and audio applications, "*Proc. Int. Symp. on Integrated Circuits (*ISIC 2007), Sep.2007, pp. 81–84

[14] G. Pastuszak, "A high-performance architecture of the double mode binary coder for h.264.avc*," IEEE Trans. Circuits Syst. Video Technol*., vol. 18, no. 7, pp. 949–960, Jul. 2008.

[15] Y.-E. Kim, K.-J. Cho, and J.-G. Chung, "Low power small area modified Booth multiplier design for predetermined coefficients*,"* IEICE *Trans. Fundam. Electron. Commun. Comput. Sci*., vol. E90-A, no. 3, pp. 694–697, Mar. 2007

[16] A.Jacobson, D. Truong, and B. Baas, "The design of a reconfigurable continuous-flow mixed-radix fft processor," in *Proc. IEEE Int. Symp. On Circuits and Syst*. (ISCAS 2009), May 2009, pp. 1133–1136.

[17] B. Paul, S. Fujita, and M. Okajima, "Rom-based logic (rbl)design: A low-power 16 bit multiplier," *IEEE J. Solid-State Circuits*, vol. 44, no. 11, pp. 2935–2942, Nov. 2009.

[18] B. Ramkumar and H. M. Kittur, "Low-Power and Area-Efficient Carry Select Adder," in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 20, no. 2, pp. 371-375, Feb. 2012

[19] M. Bahadori *et.al* "High-Speed and Energy-Efficient Carry Skip Adder Operating Under a Wide Range of Supply Voltage Levels," in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems, v*ol. 24, no. 2, pp. 421-433, Feb. 2016.

[20] K. Tsoumanis *et.al* "Pre-Encoded Multipliers Based on Non-Redundant Radix-4 Signed-Digit Encoding," in *IEEE Transactions on Computers*, vol. 65, no. 2, pp. 670-676, Feb. 1 2016.

[21] G. W. Reitwiesner, "Binary arithmetic," *Adv. Comput*., vol. 1, pp. 231–308,

[22] Y. T. Han, J. S. Koh, and S. H. Kwon, "*Synthesis filter for mpeg-2 audio decoder*," Patent US 5 812 979, Sep., 1998.

[23] "*Dual dsp plus micro for audio applications*," Feb. 2003,TDA7503 Datasheet, STMicroelectronics.

[24] Z. Huang and M. Ercegovac, "*High-performance low-power left-to-right array multiplier design,*" IEEE Trans. Comput. , vol. 54, no. 3, pp. 272–283, Mar. 2005.

[25] H.-Y. Lin *et.al* "Combined 2-d transform and quantization architectures forh.264 video coders," in *Proc. IEEE Int. Symp. On Circuits and Syst*.(ISCAS 2005), vol. 2, May 2005, pp. 1802–1805.

[26] K.-S. Chong, B.-H. Gwee, and J. S. Chang, "A 16-channel low power non uniform spaced filter bank core for digital hearing aids," *IEEE Trans. Circuits Syst. II*, Exp. Briefs, vol. 53, no. 9,pp. 853–857, Sep. 2006.

[27] K. K. Parhi, *VLSI Digital Signal Processing Systems: Design and Implementation.* John Wiley & Sons, 2007.

[28] C. Wang, W.-S. Gan, C. C. Jong, and J. Luo, "A low-cost 256-point fft processor for portable speech and audio applications, "*Proc. Int. Symp. on Integrated Circuits* (ISIC 2007), Sep.2007, pp. 81–84.

[29] K. Yong-Eun, C. Kyung-Ju, J.-G. Chung, and X. Huang, "CSD based programmable multiplier design for predetermined coefficient groups," *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, vol. 93, no. 1, pp. 324–326, 2010.

[30] N. Weste and D. Harris, *CMOS VLSI Design: A Circuits and Systems Perspective*, 4th ed. USA: Addison-Wesley Publishing Company, 2010.

[31] C. Xu, X. Dong, N. Jouppi, and Y. Xie, "Design implications of memristor-based RRAM cross-point structures," *in Proc. Design, Automation Test in Europe Conf. Exhibition* , Mar. 2011,pp. 1–6.

601562020_salma_motla

ORIGINALITY REPORT

%20	%11	%15	%0
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Tsoumanis, Konstantinos, Nikos Axelos, Nikos Moschopoulos, Georgios Zervakis, and Kiamal Pekmestzi. "Pre-Encoded Multipliers Based on Non-Redundant Radix-4 Signed-Digit Encoding", IEEE Transactions on Computers, 2015. Publication	%4
2	www.readbag.com Internet Source	%1
3	dspace.thapar.edu:8080 Internet Source	%1
4	www.slideshare.net Internet Source	%1
5	www.ijarcce.com Internet Source	%1
6	www.sandeepani-vlsi.com Internet Source	%1
7	Paul, Bipul C., Shinobu Fujita, and Masaki Okajima. "ROM-Based Logic (RBL) Design: A	%1

Low-Power 16 Bit Multiplier", IEEE Journal of Solid-State Circuits, 2009.
Publication

8	conferenceworld.in Internet Source	<%1
9	Grzegorz Pastuszak. "A High-Performance Architecture of the Double-Mode Binary Coder for H.264.AVC", IEEE Transactions on Circuits and Systems for Video Technology, 07/2008 Publication	<%1
10	J.-G. CHUNG. "Low Power Small Area Modified Booth Multiplier Design for Predetermined Coefficients", IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences, 03/01/2007 Publication	<%1
11	Anthony T. Jacobson. "The design of a reconfigurable continuous-flow mixed-radix FFT processor", 2009 IEEE International Symposium on Circuits and Systems, 05/2009 Publication	<%1
12	S. Vassiliadis. "Hard-wired multipliers with encoded partial products", IEEE Transactions on Computers, 1991 Publication	<%1
13	Advances in Intelligent Systems and Computing, 2016.	<%1

	Publication	
14	A. Singer. "Narrow Escape, Part II: The Circular Disk", Journal of Statistical Physics, 02/2006 Publication	<%1
15	opus.bibliothek.uni-wuerzburg.de Internet Source	<%1
16	Subramaniam, Uvaraj, and Srinivasan Alavandar. "Low power and high speed computation using hybridized multiplier", 2013 Fourth International Conference on Computing Communications and Networking Technologies (ICCCNT), 2013. Publication	<%1
17	works.bepress.com Internet Source	<%1
18	ijirt.org Internet Source	<%1
19	docslide.us Internet Source	<%1
20	zenodo.org Internet Source	<%1
21	Kiamal Pekmestzi, Constaninos Efstathiou. "Design of Efficient 1's Complement Modified Booth Multiplier", 2016 Euromicro Conference on Digital System Design (DSD), 2016 Publication	<%1