International Journal of Research
Available at https://pen2print.org/index.php/ijr/
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 23
December 2018

Design of High Speed Pre-Encoded Multiplier Based On NR4SD Encoding Using Han-Carlson Adder

G. Naga Bhavani[1] & E.Jagadeeswara Rao[2]
[1]PG Scholar, Department of ECE, Aditya College of Engineering and Technology, Surampalem, Andhra Pradesh, India.
[2]Assistant Professor, Department of ECE, Aditya College of Engineering and Technology, Surampalem, Andhra Pradesh, India.

Abstract:

In this paper, we introduce architecture of high speed Non-Redundant radix-4 Signed-Digit (NR4SD) encoding technique, which uses the Han-Carlson adder (HCA), is used at the addition of partial products and proposed leading to a multiplier design with less area at final sum stage and high speed implementation compared to NR4SD multiplier using Carry Save Adder (CSA). NR4SD multiplier is also area efficient compared to MB multiplier, which uses the digit values {-2,-1,0,+1} or {-1,0,+1,+2} that leads to fast multiplication than the conventional Modified Booth Multiplier. These pre-encoded multipliers are based on off-line encoding of generating less number of coefficients for Digital Signal Processing (DSP) applications. The Proposed Multiplier is fast requires less area with efficient power usage.

Keywords

NR4SD, HCA, CSA.

1. Introduction

One of the arithmetic operation multiplications is tiresome process, so multipliers are main components in arithmetic, signal and image processors. Signal processing and image processing consists of multiplying functions like multiply, accumulate, convolution and filtering. The operation rate of multiplier unit impacts the execution time of particular process. Multiplication takes more time compared to other operations in DSP algorithms, so critical delay path calculated for complete operation based on delay required for multiplication unit and it measures algorithm performance. Most widely used operations in computer arithmetic are addition and multiplication in case of approximate computing full-adder calls extensively analyzed for addition operation [1-3].

All DSP algorithms would need some form of the Multiplication and Accumulation Operation. Most of the DSP algorithms need multiplication and accumulation operation. Multiply and Accumulator (MAC) consists of adder, multiplier and accumulator. Usually DSP adders are RCA and CSA. Generally multiplier multiplies the input values and passes result to the adder, then adder adds to the previous result of accumulator.

For the primary approach there is hardware implementation of MAC unit that will incorporate the multiplier as well as accumulator within a solo hardware unit. This type of modeling is implemented by well-known corporate company Motorola for manufacturing the DSP processor named as DSP5600x. in another approach there is individual hardware units for multiplier and accumulator. One of such kind is TIDSP320C5X that has multiplier whose output is preserved in product register. Whatever the content that is placed in the product register is further summed to accumulator register which is present in central processing unit. In the above discussed procedure whatever the implementation the result of operation is finished in one clock cycle.

Organization of this paper is described as follows. Section II is a review for Modified Booth Multiplier (MBM) with different schemes. The designs of Existing Pre-Encoded NR4SD multiplier are presented in Section III. Introduction HCA are given in Section IV and proposed Multiplier Scheme are given in Section V. Simulation results for multipliers with the CSA and HCA is provided in Section VI and Section VII concludes the manuscript.

2. Literature Survey

A New algorithm invented by Booth Donald (1950) [4] for multiplying two unsigned (or signed) numbers which reduces partial PP generation compare to WM, VM and RPM and it is a low power and area efficient suitable for Bio-medical signal processing applications. Chang Yeh and Chein Wei Jen (2000) [5] designed a new modified Booth Encoding Scheme (MES) in which increase the speed 25% compare to conventional MBM. Which strive to improve performance of the traditional MBE [5].

Fayez Elguibaly (2000) [6] designed MBM with parallel multiply accumulate hardware unit. [6] In

International Journal of Research
Available at https://pen2print.org/index.php/ijr/
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 23
December 2018

which was three times faster than the other standard multiply and Accumulate Unit (MAC) unit. Zhou Shun et al. (2007) [7] proposed a radix-4 MBM with multi precision reconfigurable scheme which can be cascaded to comply with the different input length. [7] The delay and area reduce compare to parallel multiply accumulate hardware based MBM [6].

Shiann Rong Kuang and Jiun Ping Wang (2010) [8] proposed low power configurable MBM. In which developed a novel dynamic range detector is used to detect the range of two inputs and the detector helps to reduce the switching activities and power consumption of the Multiplier, they also developed some extra circuits such as correcting vector generators, sign bit generator etc. Because of the extra overhead circuit, the purposed multiplier is more complex than the regular multipliers but their power consumption is significantly reduced.

Ravindra P Rajput and M.N Shanmukha Swamy (2012) [9] proposed a high speed signed-unsigned MBM which uses CSA and CLA. The circuit is expected to reduce the power consumption and cost of the multiplier. Kostas Tsoumanis et al. (2014) [10] Proposed increase the performance of MBM with fused add multiply (FAM). In this technique improve the performance with reduce the power and delay.
K. Tsoumanis and N. Axelos (2016) [11] developed a new MBM hardware with pre-encoded scheme, in which reduce the area at encoding stage also reduce delay and power consumption compare to traditional MBM [4].

3. Existing Pre-Encoded NR4SD Multiplier

Fast multiplication operation is required in DSP and generally used processors. it is most important requirements in DSP. To increase this speed by using fast multipliers. High speed multiplication is achievable by way of series of additions, subtractions and shift operations. Required many frequent of additions for multiplication. The number to be incorporated is the multiplicand, the condition that it is incorporated is the multiplier, and the result is the item. Every step in this addition produces a partial product. In may PC's, the operand having same length of bits. Exactly when the operands are deciphered as entire numbers, the thing is generally twofold the length of operands with a particular true objective to spare the information content. This frequent addition method that is proposed by the arithmetic particulars. It is delayed process , so it replace number of times by different types of algorithms that makes use of positional illustration. It is conceivable to depreciate multipliers into two sections. . The initial segment is devoted to produce the partial products and the second one gathers and adding them. The basic multiplication principle is two overlap i.e. evaluation of partial products and aggregation of the shifted partial products. It is performed by the accompanying increments of the sections of the moved partial products matrix. Multiplicand provides the proper bit to the multiplier for shift operation. The deferred, gated example of the multiplicand should all be in the equivalent segment of shifted partial product matrix. After added to shaping the product bit for the correct frame. Multiplication is Increase is thusly a multi operand operation. To stretch out the increase to both marked and unsigned.

Hear, driven to an adder and then the input X and the sum Y=A+B are drives to the multiplier for obtaining the Z. the main imitation in the adder is major delay in the difficult path .This path depends on width of the input bits. The radix-4 Booth encoding table and Gate level representation is shown in Fig. 1 (a) and 1 (b).

TABLE I
MODIFIED BOOTH ENCODING TABLE.

Binary				MB Encoding			Input Carry
y_{2j+1}	y_{2j}	y_{2j-1}	y_j^{MB}	sign=s_j	×1=one_j	×2=two_j	$c_{in,j}$
0	0	0	**0**	0	0	0	0
0	0	1	**+1**	0	1	0	0
0	1	0	**+1**	0	1	0	0
0	1	1	**+2**	0	0	1	0
1	0	0	**-2**	1	0	1	1
1	0	1	**-1**	1	1	0	1
1	1	0	**-1**	1	1	0	1
1	1	1	**0**	1	0	0	0

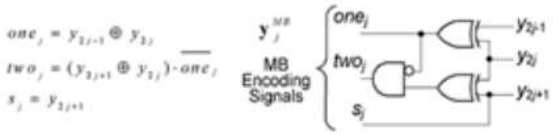

Fig.1 (a) Boolean equations and (b) Gate level representation for the execution of the MB encoding signals. [11]

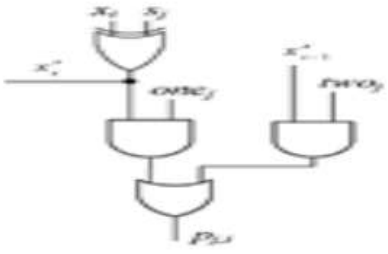

Fig. 2 Production of the i^{th} Bit $P_{j,i}$ of PP_j for conventional MB Multiplier. [11]

For the calculation of the least and the most significant bits of the partial product we consider and correspondingly. Note that in the event that that , the quantity of the resulting partial products is and the most significant MB digit is shaped in light of sign

International Journal of Research
Available at https://pen2print.org/index.php/ijr/
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 23
December 2018

augmentation of the underlying 2's complement number and shown in Fig. 2.

After this generated partial products are added and correctly weighted, through a Carry-Save Adder (CSA) tree with the Correction Term (CT) which is given by the accompanying equations:

$$Z = X \cdot Y = CT + \sum_{j=0}^{k-1} PP_j \cdot 2^{2j}$$

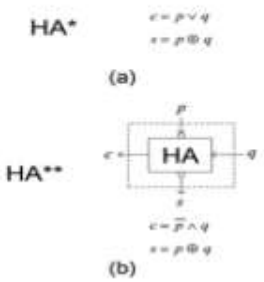

Fig.3 Boolean conditions and schematics for marked (a) HA* (b) HA** [11]

At long last, Output of the CSA tree is given to a quick Carry Look Ahead (CLA) adder to frame the ultimate result is Z = X . Y as appeared in Fig.3 (a). The Conventional Pre-Encoded NR4SD Multiplier is shown in Fig. 4

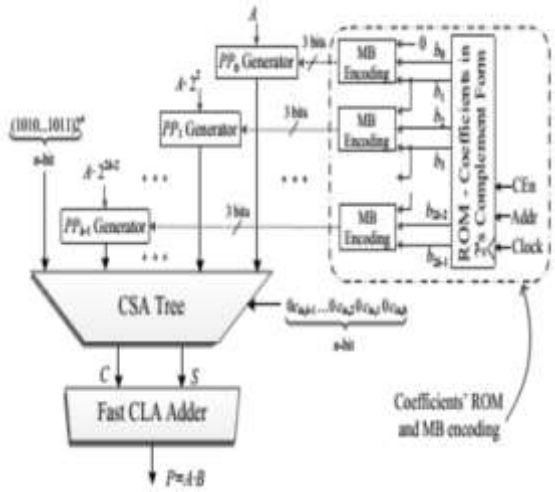

Fig. 4 Conventional Pre-Encoded NR4SD Multiplier.

The system design for the pre-encoded NR4SD multipliers is displayed in Fig. 4. Two bits are now stored in ROM: n2j+1, n+2j(Table 2) for the NR4SD or n+2j+1, n2j (Table 3) for the NR4SD^{+}form. Such a way, we decrease the memory necessity to +1 bits for every coefficient while the relating memory required for the pre-encoded MB algorithm is 3n/2 bits for each coefficient. Along these lines, the measure of put away bits is equivalent to that of the conventional MB configuration, aside from the most significant digit that needs an additional bit as it is MB encoded. Appeared differently in relation to the pre-encoded MB multiplier, where the MB encoding units are disposed of, the pre-encoded NR4SD multipliers require extra equipment to make the indications of (6) and (8) for the NR4SD and NR4SD+ algorithm. Respectively. Every partial product of the pre-encoded NR4SD and NR4SD+ multipliers is implemented based on below Fig. 8c and 8d, respectively, excluding the PP_{k1} that relates to the most significant digit. As this digit is in MB form, we utilize the PPG of Fig. 8b applying the change specified in Section 4.2 for the s j bit. The partial products, legitimately weighted, and the rectification term (COR) are encouraged into a CSA tree. The input carry cin;j is calculated as cin;j = twoj_ onej and cin;j = onej for the NR4SDand NR4SD+pre-encoded multipliers, respectively ,based on Tables 2 and 3. The convey spare yield of the CSA tree is at long last summer utilizing a quick CLA added. Give us a chance to consider the increase of 2's supplement numbers X and Y with each number having of n=2k bits .The multiplicand Y can be represented in MB.

NR4SD $^{-}$ Algorithm:

Step 1: Consider the initial values j = 0 and C0=0.

Step 2: Determine the carry C2j+1 and the sum $n2j$ + of a Half Adder (HA) with inputs $b2j$ and $c2j$ (Fig. 5a).

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

Step 3: Determine the positively signed carry $c2j$+1(+) and the negatively signed sum $n2j$+1 − (−) of a Half Adder* (HA*) with inputs $b2j$+1(+) and $c2j$+1(+) (Fig. 6a). The outputs $c2j$+1and n2j+1of the HA* narrate to its inputs as follows:

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

The subsequent Boolean equations review the HA* operation:

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

Step 4: Determine the value of the NR_j^{NR-} digit.

$$\mathbf{b}_j^{NR-} = -2n^{-}_{2j+1} + n^{+}_{2j}.$$

Step 5: j := j + 1. Step 6: If (j < k-1), go to Step 2. If (j = k-1), encode the MSB depends on the MB algorithm and allowing for the three consecutive bits to be $b2k-1$, $b2k-2 and\ c2k-2$ (Fig. 5b). If (j = k), Table 2 shows the formation for NR4SD_digits. Equations (6) show how the NR4SD^{-} encoding signals one_j^{+}, one_j^{-} and two_j^{-} of Table 2 are gererated.

$$one_j^{+} = \overline{n^{-}_{2j+1}} \wedge n^{+}_{2j},$$
$$one_j^{-} = n^{-}_{2j+1} \wedge n^{+}_{2j},$$
$$two_j^{-} = n^{-}_{2j+1} \wedge \overline{n^{+}_{2j}}.$$

International Journal of Research
Available at https://pen2print.org/index.php/ijr/
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 23
December 2018

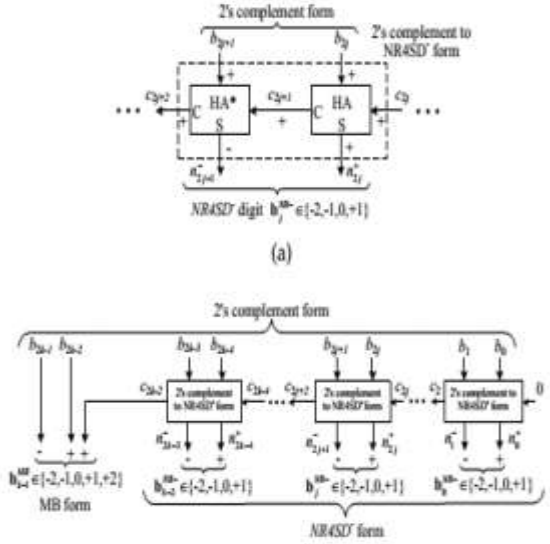

Fig. 5 Block Diagram of the NR4SD- Encoding Scheme at the (a) Digit and (b) Word Level. [11]

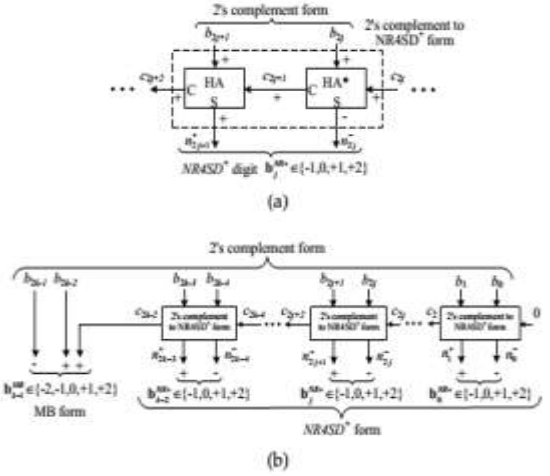

Fig. 6 Block Diagram of the NR4SD+ Encoding format at the (a) Digit and (b) Word Level. [11]

TABLE 2
NR4SD⁻ Encoding

2'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

TABLE 3
NR4SD⁺ Encoding

2'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	1	0	+2	0	0	1
1	0	0	0	1	0	+2	0	0	1
1	0	1	1	0	1	-1	0	1	0
1	1	0	1	0	1	-1	0	1	0
1	1	1	1	0	0	0	0	0	0

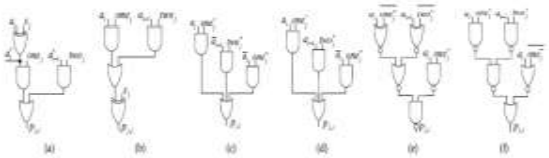

Fig. 7 Generation of the i^{th} Bit $p_{j;i}$ of PP_j for a) Conventional, b) Pre-Encoded MB Multipliers, c) NR4SD$^+$, d) NR4SD$^+$ Pre-Encoded Multipliers, and e) NR4SD$^-$, f) NR4SD$^-$ Pre-Encoded Multipliers after reconstruction. [11]

4. Han-Carlson Adder

The formula for binary addition problem is generation of n-4, sum $s=s_{n-1}$, s_{n-2}.so n-bit augends $A=a_{n-1}$, a_{n-2}….. a_0 and n-bit addend $B=b_{n-1},b_{n-2}$……b_0.we get c_i is the carry out of the i^{th} bit. The computation of whole piece s_i and carry c_i is:

$$s_i=a_i \oplus b \oplus c_{i-1} \quad (1)$$

$$c_i=a_ib_i+a_ic_{i-1}+b_ic_{i-1} \quad (2)$$

International Journal of Research
Available at https://pen2print.org/index.php/ijr/
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 23
December 2018

The stages of prefix addition to register are sum pre-processing, prefix processing and post processing. The generate g_i and propagate p_i signals are:

$$G_i=a_i \bullet b_i \quad (3)$$

$$P_i=a_i \oplus b_i \quad (4)$$

If $g_i=1$ means carry is generated at bit I and when $p_i=1$, a carry is propagated through bit i. The concept of generate and propagate extended to a block of contiguous bits, from bit k to I with necessary condition k<i.

$$g[i:k]=\begin{cases} gi & if\ i=k \\ g[i:j]+p[i:j]g[l:k] & otherwise \end{cases} \quad (5)$$

$$p[i:k]=\begin{cases} pi & if\ i=k \\ p[i:j]p[l;k] & otherwise \end{cases} \quad (6)$$

Where: i≥l≥j≥k

A carry is generated in block k-1 if condition g[i:k]meets and a carry is propagated through the block if condition meets p[i:k].The expression for any bit i the carry c_i is

$$C_i=g[i:0]+p[i:0]c_{-1} \quad (7)$$

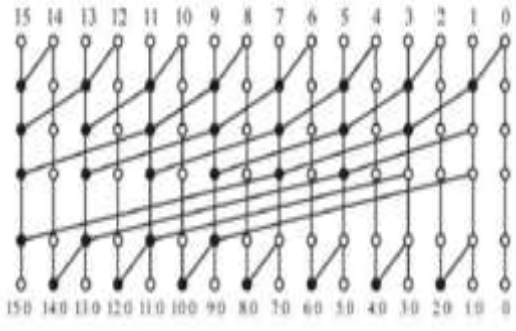

Fig. 8. HA Topology n=16 [12]

Where C_{-1} is the information convey of the n-bit adder. In the accompanying, for straight forwardness, we accept that $C_{i-1}=0$, so that above equation 4 follows as: $C_i=g[i:0]$

The prefix-preparing phase of the adder consists of block generate and propagate terms. So (g[i,k],p[i:k]) couples communicated with support of prefix operator shown below.

$$(g[i:k],p[i:k])=(g[i:j],p[i:j]) \bullet (g[l:k],p[l:k])=(g[i:j]+p[i:j]g[l:k],p[i:j]p[l:k]) \quad (8)$$

5. Proposed Multiplier

The Proposed NR4SD Multiplier design is shown in given Fig. 9 In this diagram instead of CSA place, we can use Han-Carlson Adder (HCA). The HCA is a high speed adder compare to CSA. Generally the HCA reduce the critical path delay of carry propagation so the final sum we can get only less time. In designing of proposed multiplier consists of three steps i.e., 1.Encoding Stage, 2. Partial Product Generation Stage and 3. Partial Product Addition stage. So, We can modified the Final Partial product sum stage and place HCA then we can design of the multiplier then we can very less delay, area compare to existing NR4SD Multiplier.

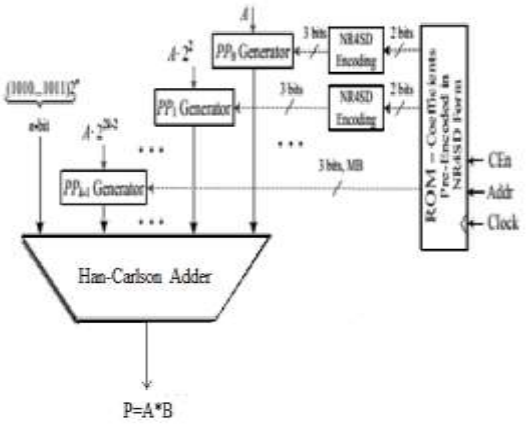

Fig. 9 System Architecture of the Proposed NR4SD Multiplier Using HCA

6. Results and Discussion

The design was synthesized on Xilinx 14.7 ISE and the functional verification of Existing Pre-Encode NR4SD Multiplier and proposed Multiplier was done on Xilinx 14.7. The targeted device is of Spartan-3e of Spartan family. The grade speed of the design is set to -5. The following section contains the results obtained by synthesizing the design in Xilinx ISE. Table 3 represents the results of the performance analysis of Existing and proposed Multiplier Techniques.

Table 3
Comparison Table between Existing NR4SD Multiplier and Proposed Multiplier

	AREA			DELAY (ns)
	SLICES	LUTS	FFS	
NR4SD Multiplier	105	181	8	19.22 ns
Proposed NR4SD Multiplier	1	1	0	4.04 ns

International Journal of Research
Available at https://pen2print.org/index.php/ijr/
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 23
December 2018

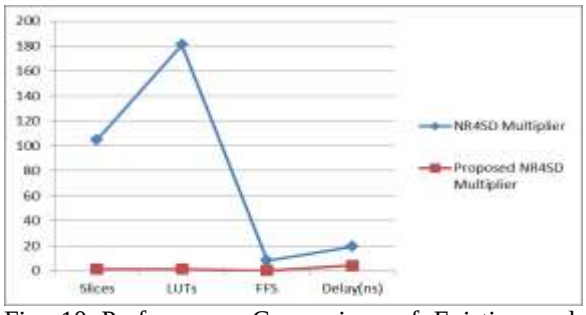

Fig. 10 Performance Comparison of Existing and proposed Multiplier

The performance analysis graph is shown in Fig. 10. Once observe the graph delay and area reduce the proposed technique compare existing one. The simulating output of proposed 8-Bit multiplier is shown in Fig. 11.

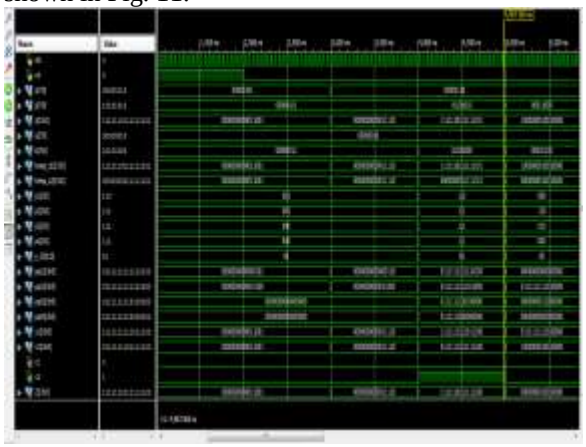

Fig 11 Simulation output of proposed 8-Bit Multiplier

7. Conclusion

Novel design of pre-encoded multipliers is investigated by off-line encoding the standard coefficients along with system memory and also proposed a technique encoding these coefficients in the Non-Redundant radix-4 Signed-Digit (NR4SD) with Han-Carlson Adder. The proposed pre-encoded NR4SD multiplier is less area as well as time efficient compare to Existing NR4SD multiplier, Usual and pre-encoded MB multipliers. In-depth experimental analysis ensures the gains of the proposed Multiplier pre encoded multipliers in provisions of less area complexity relative to the NR4SD multiplier and the proposed multipliers achieve a very fast response compared to the NR4SD multipliers. Delay in NR4SD Multiplier is 19.22 ns.This delay is reduced by utilizing proposed Multiplier; here we are achieving 4.04 ns delay with same output then compare to Existing NR4SD Multiplier as well as area all so reduced in Proposed Multiplier.

8. References

[1] V. Gupta, D. Mohapatra, S. P. Park, A. Raghunathan, and K. Roy, "IMPACT: IMPrecise adders for Low-Power Approximate Computing", Proc. of Int. Symp. On Low Power Electronics and Design (ISLPED). 1-3 Aug. 2011

[2] S. Cheemalavagu, P. Korkmaz, K.V. Palem, B.E.S. Akgul, and L.N. Chakrapani, "A Probabilistic CMOS Switch and its Realization by Exploiting Noise," in Proc. IFIP-VLSI SoC, Perth, Australia, Oct 2005

[3] H.R. Mahdiani, A. Ahmadi, S.M. Fakhraie, C. Lucas, "Bio-Inspired Imprecise Computational Blocks for Efficient VLSI Implementation of Soft-Computing Applications", IEEE Trans. on Circuits and Systems I: Regular Papers, Vol. 57, No. 4, pp. 850-862, April 2010

[4] A.D. Booth, "A Signed Binary Multiplication Technique," *Jour. Of Mech. Appl. Math.*, vol. 4, pp. 236-240, Oxford University Press,1951.

[5] Wen-Chang Yeh and Chein-Wei Jen, "High-Speed Booth Encoded Parallel Multiplier Design*",IEEE Trans. on Compt.*, vol. 49, no. 7, pp.692-701, July, 2000.

[6] Fayez Elguibaly, "A Fast Parallel Multiplier-Accumulator Using the Modified Booth Algorithm", *IEEE Trans. Circuits and Sys. II, Analog and Digital Signal Processing*, vol. 57, no. 9, pp. 902-909, Sept., 2000.

[7] Zhou Shun, Oliver A. Pf¨ander, Hans Jorg Pfleiderer and Amine Bermak, "A VLSI architecture for a Run-time Multi-precision Reconfigurable Booth Multiplier", *14th IEEE Int. Conf. on Electronics, Circuits and Systems*, pp. 975-978, Dec., 2007.

[8] Shiann-Rong Kuang and Jiun-Ping Wang, "Design of Power-Efficient Configurable Booth Multiplier", *IEEE Trans., Circuits and Sys. II Regular Papers*, vol. 57, no. 3, pp. 568-580, March, 2010.

[9] Ravindra P Rajput and M.N Shanmukha Swamy, "High speed Modified Booth Encoder multiplier for signed and unsigned numbers*", 14th IEEE Int. Conf. on Modelling and Simulation*, pp. 649-654, 2012.

[10] Kostas Tsoumanis, Sotiris Xydis, Constantinos Efstathiou, Nikos Moschopoulos and Kiamal Pekmestzi, "An Optimized Modified Booth Recoder for Efficient Design of the Add-Multiply Operator", *IEEE Trans., Circuits and Systems I, Regular Papers*, vol. 61, no. 4, pp. 1133-1143, April, 2014.

[11] K. Tsoumanis, N. Axelos, N. Moshopoulos, G. Zervakis and K. Pekmestzi,"Pre-Encoded Multipliers Based on Non-Redundant Radix-4 Signed-Digit Encoding", IEEE Trans. On Computers, vol. 65, pp. 670-676,Feb. 2016.

[12] S. M. Sudhakar, K. P. Chidambaram and E. E. Swartzlander, "Hybrid Han-Carlson Adder", *Int. Midwest Symposium on Circuits and Syst. (MWSCAS)*, Boise, Idaho, pp. 818-821, Aug. 2012.