COMPARATIVE ANALYSIS OF DIFFERENT LNS DESIGN

Ujjayant Lamba [1], Mohammad Umar [2]
Electronics and Communication Engineering Department,
Satya college of engineering and technology, palwal, Haryana, India,

***Abstract*- The logarithmic number system can be used in various ways to represent the data in efficient way which is used in various special purpose VLSI processors. The LNS exploits the property to reduce the basic arithmetic function of division, roots multiplication and powers to binary subtraction, addition and left and right shifts respectively. Furthermore, the LNS also provides additional benefit as it gives us freedom to choose the logarithmic base. In this paper, basic architecture of LNS adder/subtractor consist of adder, multiplexer and 2 look up tables(LUT) which are named as addition LUT and subtraction LUT. Power dissipation is done by partitioning the LUT's into sub-lut's and further dividing into 4 lut's. Partitioning of LUT is done to create parts in circuits to help in achieving power dissipation. only one of sub-lut is activated upon on each operation which depends on MSB or LSB and also on the sign of operands. Power dissipation can further be reduced by using D-FF since they allow more than one MSB to break the lut's. The performance has been measured in terms of delay, power, area with logical effort and Xilinx ISE 14.4(Verilog HDL).**

***Keywords:* HDL, LNS, LUT, Xilinx, MSB, LSB.**

I. INTRODUCTION

A logarithmic number system (LNS) is an arithmetic system used for representing real numbers in computer mostly for digital signal processing. In LNS, a number, X, is represented by the logarithm, x of its absolute value as follows:
X {s,x = logb(|X|)}

Where s is a bit denoting the sign of X (s = 0 if X > 0 and s= 1 if X < 1).The number x is represented by a binary word which usually is in the two's complement format. LNS can be evaluated as a floating-point number with the significant being always equal to 1. This formulation make the operations of division, powers, multiplication and roots very simple, since they are reduced down to division, addition, subtraction, and multiplication respectively[1].On the other hand, the operations of subtraction and addition are more complicated and they are calculated by the formula:

logb(|X| + |Y|) = x + sb(z)
logb(|X| - |Y|) = x + db(z)

Where z = y-x is the difference between the logarithms of the operands, the "sum" function is sb(z) = logb(1+bz), and the "difference" function is db(z) = logb(1–bz). These functions of sb(z) and db(z), depicted in the fig.1 is known as Gaussian logarithms. The simplification of division, multiplication, powers and roots is counterbalanced by the cost of evaluating these functions for addition and subtraction. If m and n are positive numbers, a is a positive number other than 1 and p is any real number, then the following laws hold true:
Law for Logarithms of Products: loga(mn) = loga m +loga n
Law for Logarithms of Quotients: loga(m/n) = logam-logan
Law for Logarithms of Powers: loga mp = p loga m

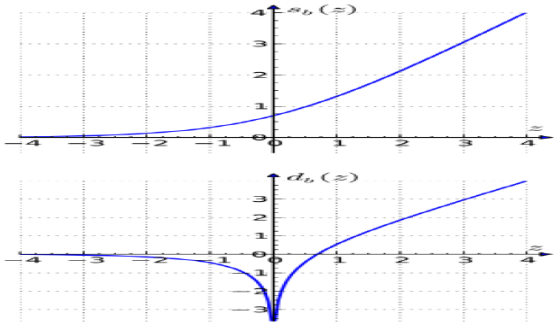

Fig 1 Graph of sb(z) and db(z)

1.1 Logarithmic Identities

Several important formulas, sometimes called logarithmic identities or log laws, relate logarithms to one another. The logarithm of a product is the sum of the logarithms of the numbers being multiplied; the logarithm of the ratio of two numbers is the difference of the logarithms. The logarithm of the *p*-th power of a number is *p* times the logarithm of the number itself; the logarithm of a *p*-th root is the logarithm of the number divided by *p*. Each of the identities can be obtained after substitution of the logarithm definitions x = blogb(x), and/or y = blog b(y), in the left hand sides[2].

Operation	Formula
Product	$Log_b(xy) = log_b(x) + log_b(y)$
Quotient	$Log_b(x/y) = log_b(x) - log_b(y)$
Power	$Log_b(x^p) = plog_b(x)$
Root	$Log_b\ (p\sqrt{x}) = log_b(x)/p$

Table 1 Logarithm identities

II. LNS DESIGN

The basic idea in LNS is to use logarithmic to represent data, since the logarithm of negative number is not real. In LNS the sign information is used as a separate bit Sx and in combination with logarithm of magnitude of number to represent the signed number. Furthermore the logarithm of zero is not a finite number so an additional single bit flag Zx is used to denote that the number is zero[4].let us assume that X denote the number and x denote the logarithm of absolute value of |X|. Xlns is a triplet having sign bit, x and zero bit. In LNS ,a number X is represented as triplet

Xlns = (Zx,Sx,x)

Where Zx is asserted in that case X is zero.Sx is the sign of X and x = logb(|X|),if X is not zero with b being base of logarithm also called base of representation. Due to basic properties of logarithm,the multiplication of Xlns and Ylns is reduced to computation of of triplet Zlns

Zlns = (ZZ,SZ,Z) where ZZ = Zx ∨ ZY, Sz =SX (+)SY , Z = X + Y

Similarly, the division reduces to binary subtraction

The derivative of logarithm a of the sum A of two triplets is more involved as it relies on computation of

a = max{x,y} + log2(1+ b-|x-y|)
=max{x,y}+øa(d)……(1)

Where øa(d) = log2(1+ b-d) and d = |x-y|

Similarly ,the derivation of the difference of two number, requires the computation of

c = max{x,y} + log2(1- b-|x-y|)
=max{x,y}+øs(d)……(2)

Where øs(d) = log2(1- b-d) and d = |x-y|

The basic organisation of an adder/subtractor is shown in fig.2.

The parallel subtractions

s1 = x-y……...(3)
s2 = y-x……(4)

are implemented followed by a multiplexer which computes d according to the rule

d = |x-y| = { s1 , s1 > 0
{s2,otherwise…… (5) }

The choice of sign of either (1) or (2) as a select signal for multiplexer. The same signal is used to select the maximum of x and y required for computation of (1) and (2).The complexity of LNS circuitary arises from the fact that the values of functions øa and øs should be computed by the LNS addition/subtraction circuit hardware for all required values of d. There are two main approaches to implement the evaluation of functions namely storage of all required values in an LUT and the hardware implementation of an approximation algorithm or offline precomputation.[5].The former approach is preferred for smaller word length i.e in low precision application where size of required LUT is moderate, while the latter approach is generally used for high-precision applications

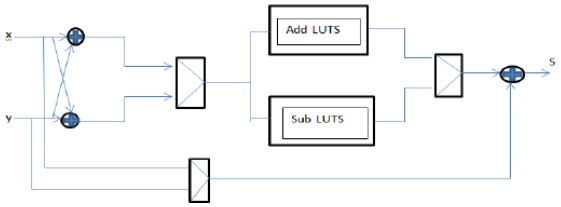

Fig2 The organisation of an LNS adder/subtractor.

2.1 Low Power Design of LNS Adder/subtractor

Low power LNS architecture for addition and subtraction are discussed. The memory structure is organised as a collection of LUTs and is most complex part of LNS adder/subtractor. We have designed two designs or we have two choices i.e first the choice of using either latches or D-flip-flops(DFFs) to freeze the address of inactive sub-luts either based of MSB of d in (5)

In the proposed design, power dissipation reduction is sought by partitioning the particular LUTs into smaller LUTs known as sub-LUTS, only one of which is vital per operation. To gurantee that no dynamic power is dissipated in the inactive sub luts, the corresponding sub-lut addressed are latched and remain constant through out a particular operation .The organisation of LNS adder/subtractor comprises N sub-luts per operation [6]. The upper Sub-lut system corresponds to function Qa (d) required for LNS addition ie addition of operands having same

sign,while the lower sub-lut system is used for LNS subtraction ie addition of operands of different signs.

2.2 Implementation of LNS Adder/Subtractor

2.2.1 LNS addition/subtraction with latches

Fig 3 depicts an architecture using one bit MSB selection for LUT segregation. latches are connected to the inputs of the sub-luts. sign s and d0 are used to generate a signal that enables the latches at the input of the sub-lut required to be activated for particular computation. The MSB should reach the latch fast enough, considering the additional delay of |x-y| to avoid the violation of timing constraints. To achieve the required minimum delay the computation of is delayed with buffers.
Furthermore the computation of is performed by parallel prefix adder , to minimize the delay between the MSB generation and LSB generation. In case of using a ripple-carry adder,more additional delay has to be added to avoid timing violations at the latches. The power consumption is considerably lower compared to implementations that do not latch the sub-LUT inputs, despite the additional power consumed by the introduced buffers, required for the additional delay MSB-based LUT activation is not efficient in the case of a latch based architecture, since significant amount of delay should be introduced in order to avoid the timing violations.

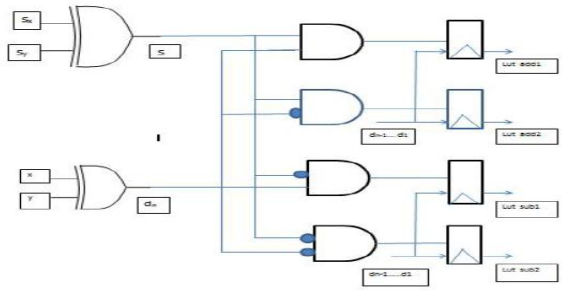

Fig 3 Latch organisation using MSB as selection bit

Simulation Result of Latch organisation

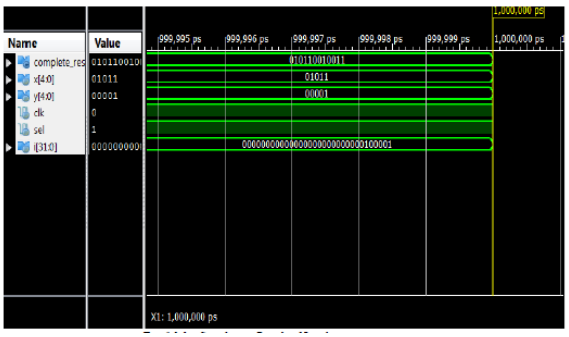

Fig 4 Simulation Result of Latch organization

2.2.2 LNS addition/subtraction with D-flip-flop

An alternative implementation of LNS addition/subtraction is studied, based on the use of D flip-flops (DFF) with clock and reset instead of level-sensitive latches. Significant power consumption reduction is achieved when using D-flip-flops instead of latches. The advantage of the adopting flip-flop based selection is that one or more MSBs can be used to break up the LUTs. Fig.5 summarizes the organization of the D-flip-flop based architecture. Signals clock and reset are not shown for clarity [7].It is concluded that MSB-based architecture creates simpler sub-LUTs. Since the utilization of the MSB for LUT selection is not effective for a latch-based design, a solution based on D flip-flops is preferable. It is noted that a latch based gated clock is used for DFF since additional signals are used to enable the corresponding flip flop.

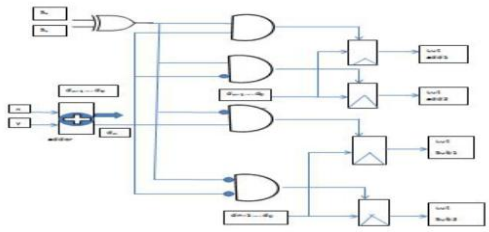

Fig 5 D-FF organisation using MSB for LUT selection

Simulation Result of D-FF organisation.

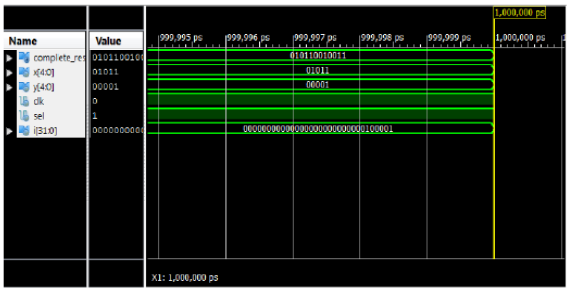

Fig 6 Simulation Result of D-FF organisation.

2.2.3 Look up table

The Look-Up Table and Related Logic: A lookup table is used to obtain the values of the two functions –: $\log_2(1 + 2\text{-}D)$ and $\log_2(1 - 2\text{-}D)$. The values are stored in two different tables which are selected by a signal named LU. The number D is the address input to both tables. The range of D (which is zero to MAX, where MAX = +2+64) is divided into a number of smaller intervals. This is helpful in reducing the total memory required. For most intervals, a separate ROM is used with a PLA. There

is an identical breakup of the interval for the function $\log_2(1- 2\text{-}D)$. The first entry is defined over a range of D in which either of the functions $\log_2(1 + 2 \text{-}D)$ or $\log_2(1 \text{-}2\text{-}D)$ has a value that is less than the resolution of the system (called essential zero). Hence, both function values are known a priori and are equal to zero. The rationale behind choosing a multiple ROM scheme and the choice of breakpoints over the range of D is in a later section[8]. There are three hardware parts to the lookup table called the LU logic, the interval determinator, and the table subsystem (composed of ROM, PLA, and multiplexers) itself.

2.2.4 Look up Memory Reduction Techniques

The lookup table is used to evaluate the two functions $\log_2(1 + 2\text{-}D)$ and $\log_2(1 - 2\text{-}D)$ (also referred to as F1 and F2,respectively), where D = ex – ey. The lookup table could be a 19 bit wide address input D and provide a 19 bit wide output. The ―brute force‖ memory requirements are extremely bit intensive. However, the reported LNS chip is predicated on the use of data compression techniques[8s].First, we ensure that D is always positive by comparing ex to ey to form other words D = max (ex ,ey) - min (ex ,ey). In the proposed embodiment, this is achieved by evaluating ex – ey and ey – ex in parallel and choosing the appropriate output.

3 Proposed LNS MAC Architecture

MAC stands for multiply accumulate where a multiplier is followed by a adder and accumulator that stores the result. The output of register is fed back to one input of adder so that at each clock cycle the output of multiplier is added to register.. The LNS equivalent to single MAC architecture is given in fig 7 where binary multiplier has been replaced by an adder, binary adder is mapped to an LNS adder/subtractor. The LNS adder/subtractor is supplement with saturation circuits and exploits a zero flag to avoid unnecessary activation of lut partition and further reduce power dissipation. The employed LNS MAC uses MSB based architecture for LUT partitioning and use DFF for address latching. The word length of 12 bit was selected for LNS MAC. The LNS MAC was found to be more efficient then its binary counterpart.

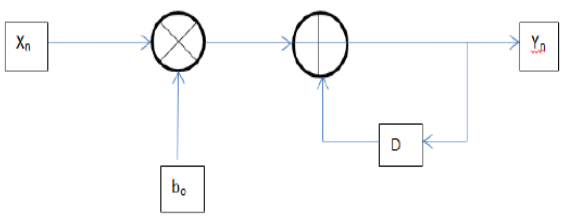

Fig 7 Organisation of single MAC architecture

Simulation Result of BINARY MAC

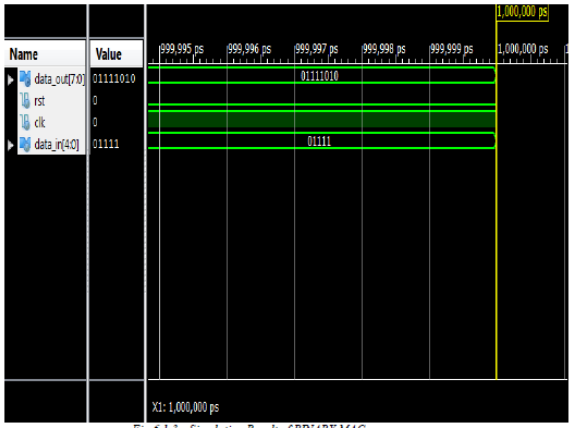

Fig 8 Simulation Result of BINARY MAC

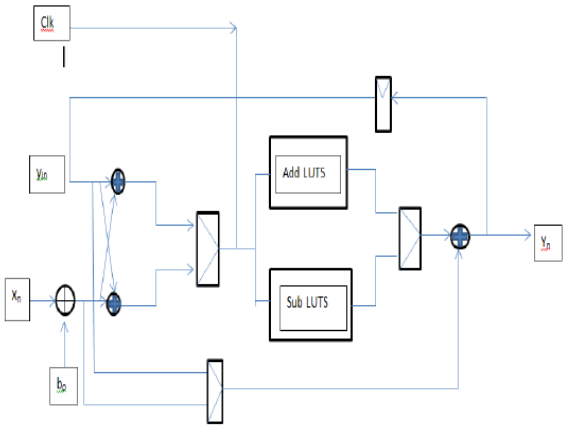

Fig 9 LNS MAC Unit

Simulation Result of LNS MAC

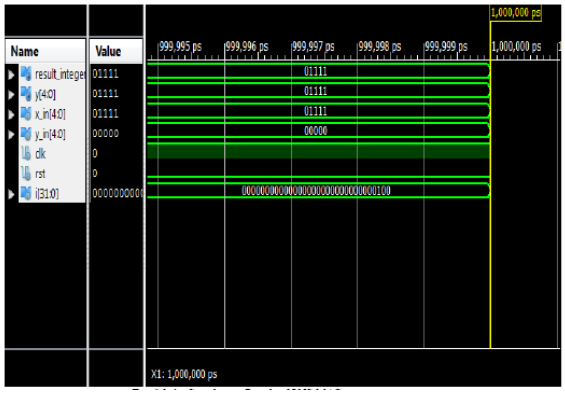

Fig 10 Simulation Result of LNS MAC

The LNS equivalent to a single MAC architecture is depicted in fig where binary multiplier has been replaced by an adder and binary adder is mapped to lns adder/subtractor.The LNS adder/subtractor is elevated with saturation circuitary and exploits a zero flag to avoid unnecessary activation of LUT partition and further reduce power dissipation.

III. RESULTS

By exploiting optimal selection of LNS representation (base and word length) combined with proper LUT partitioning, it is found that substantial power dissipation savings are obtained at no performance penalty, as shown by simulation results. The LNS adder/subtractor with D-FF was found to be more efficient than latch enabled LNS adder/subtractor. While LUT partitioning is employed to create parts in the circuit that can be independently activated.

Comparision Of Results

S.no	LNS adder/subtractor	Delay(ns)	Slice lut	No. of Bounded I/O	Power(mW)
1.	LATCH	11.004	143	25	47.14
2.	D-FF	11.004	149	24	48.64
3.	BINARY MAC	No Path	8	15	190.57
4.	LNS MAC	7.054	38	21	47.35

Table2 Comparision Of Results

IV. Conclusion

The design techniques and quantitative performance analysis of LNS MAC units presented in this paper, show that LNS can offer a viable solution for low-power signal processing systems with moderate word length requirements. This paper shows that the assumption of LNS can lead to very efficient circuits for digital filtering applications when appropriately selecting the logarithmic base and the word length.

References

[1] T.Stouraitis and V.Paliouras, ―Considering the Alternatives in Low-Power Design,‖ IEEE Circuits and Devices, vol. 17, no. 4,pp. 23-29, July 2001.

[2] P.E. Landman and J.M. Rabaey, ―Architectural Power Analysis:The Dual Bit Type Method,‖ IEEE Trans. Very Large Scale Integration Systems, vol. 3, no. 2, pp. 173-187, June 1995.

[3] M.G. Arnold, T.A. Bailey, J.R. Cowles, and M.D. Winkel,―Applying Features of the IEEE 754 to Sign/Logarithm Arithmetic,‖ IEEE Trans. Computers, vol. 41, pp. 1040-1050, Aug. 1992.

[4] I. Orginos, V. Paliouras, and T. Stouraitis, ―A Novel Algorithm for Multi-Operand Logarithmic Number System Addition and Subtraction using Polynomial Approximation,‖ Proc. IEEE Int'l Symp. Circuits and Systems (ISCAS '95), pp. III.1992-III.1995,

[5] V. Paliouras and T. Stouraitis, ―A Novel Algorithm for Accurate Logarithmic Number System Subtraction," Proc. IEEE Symp.Circuits and Systems (ISCAS '96), vol. 4, pp. 268-271, May 1996.

[6] D. Chandra, ―Error Analysis of FIR Filters Implemented Using Logarithmic Arithmetic," IEEE Trans. Circuits and Systems II:Analog and Digital Signal Processing, vol. 45, no. 6, pp. 744-747, June 1998.

[7] V. Paliouras and T. Stouraitis, ―Low-Power Properties of the Logarithmic Number System," Proc. 15th Symp. Computer Arithmetic(ARITH), pp. 229-236, June 2001.

[8] S. Collange, J. Detrey, and F. de Dinechin, ―Floating-Point or LNS: Choosing the Right Arithmetic on an Application Basis," Proc. Ninth Euromicro Conf. Digital System Design (DSD '06), pp. 197-203, 2006

Ujjayant lamba is currently pursuing M.TECH from satya college of engineering & technology, palwal . He did his B.TECH in 2012 in Electronics and telecommunication Engineering from same college. His area of interest include Electronics, Analog and Digital Communication and optical communication.

Mohammad Umar is currently working as an asst. professor in Satya college of engineering and technology, palwal. He did his B.TECH in 2008 in Electronics and Communication Engineering from UPTU and M. Tech in 2011 in Nanotechnology from the Jamia Millia Islamia. He has over 8 years of teaching experience. His area of interest include Digital Electronics, Digital system Design and low power digital ckt.

