ISSN: 2249-7196
IJMRR/July 2022/ Volume 12/Issue 3/Article No-1/01-8
Capt. Mr. D. Veeranna / International Journal of Management Research & Review

Design and Implementation of CVNS Based Low Power 64-Bit Adder

Mr. D. Veeranna, Dr. K. Amit Bindaj, Dr. A. Venkateswarlu, N. Srinivas Rao, Mrs. K. Yojana
Design and Implementation of CVNS Based Low Power 64-Bit Adder
dev.veer57@gmail.com, karpurapu.gavasj@gmail.com, wenkateswarlu@gmail.com, nomula09@gmail.com
yojanak5@gmail.com

Abstract— **In this project, design of a mixed-signal 64-bit adder based on the Continuous Valued Number System (CVNS) is presented. The 64-bit adder is generated by cascading four 16-bit Radix-2 CVNS adders. Truncated Summation of the CVNS digits reduced the number of required interconnections in the system, which in turn reduces design complexity, power and hardware costs. This adder can perform one 64-bit, two 32-bit, four 16-bit additions on demand for media signal processing applications. The compact and low-power design of the CVNS adder is suitable for multimedia applications. This system implements an algorithm for Digital Systems which requires less number of interconnections due to truncation summation. The synthesize 64-Bit CVNS adder using Cadence RTL Encounter has a timing slack 7ps, power consumption of about 98.55 fW with the core area of 3995 µm^2.***(Abstract)***

Index Terms— **Computer arithmetic, Analog digits, continuous valued number system (CVNS), media signal processing, mixed signal adder, reconfigurable adder, and 64-bit adder.**

I. INTRODUCTION

The Adders are of fundamental importance in a wide variety of digital systems. Fast addition is an essential arithmetic function for most advanced digital systems, which heavily impacts the overall performance of digital systems. Various adder structures can be used to execute addition such as serial and parallel structures, but adding fast using low area and power is still challenging. Adders are used in every arithmetic operation, and also for computing the physical address in most modern CPUs. Adders are also used in many other digital systems including telecommunications systems in places where a full - fledged CPU would be superfluous. Many types of adders exist. Ripple adders are smaller but the design computation is very slow. Carry - select adders are very fast but they larger and consume much more power than ripple or carry - skip adders.

Multimedia signal processing has received major attention due to increasing demand on multimedia devices such as cellular phones, digital cameras, and video devices [1]. To design efficient signal processing units for these types of applications reconfigurable adders are required which are capable of processing data with varying lengths without adding too much to design complexity. An efficient adder design is important for the development of reconfigurable architectures and it can usually add one 64-bit, two 32-bit, four 16-bit and eight 8-bit operations [2]. Generally adding reconfigurability property to the adder increases the cost of implementation in terms of worst case delay and power consumption [3].The continuous valued number system (CVNS) representation is a new approach in computer arithmetic. It is a continuous number system with non integer digits, and has been used successfully in developing high performance and efficient arithmetic units like adders.

II. CONTINUOUS VALUED NUMBER SYSTEM

CVNS [4] stands for "Continuous Valued Number System". CVNS is a novel continuous (analog) digit representation and arithmetic system. This number system performs arithmetic operations by applying digit-level modular reduction operation on continuous real values. Some of the important and known features of the CVNS are given. These are the general arithmetic features of the CVNS, and do not consider actual system design issues of arithmetic units based on this number system. These features can be obtained by the mathematical expressions for a feasible design of a reconfigurable adder.

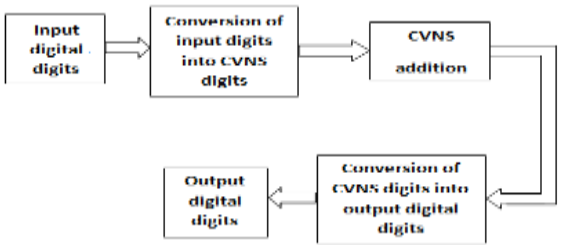

Fig. 1 Block Diagram of CVNS

A. CVNS Digits

Any value within the boundary such as $|X| \leq M$ from a positional number system with radix-B can be mapped to a set of CVNS digits to a set of CVNS digits in radix-β. The CVNS values, ((x)) are an group of CVNS digits, and can be written as a vector as follows. Where ($-k \leq i \leq n$)

$$((x)) => \{((x))_n, ((x))_{n-1}, \ldots, ((x))_0 | ((x))_{-1}, \ldots, ((x))_{-k}\} \quad (1)$$

The main characteristic of the CVNS digits is that they do not have a grid and can take continuous values. The CVNS digits are obtained by applying a basic modular reduction operation, in parallel, as follows:

$$((x))_i = \left(\frac{x}{M} . \beta^{n-i+1} \right) mod\beta \quad (2)$$

Where modβ is the modulo operation on any real value and (a) modβ=a+I.β where I is an integer and M is the maximum range of representation.

Each CVNS digit consists of two parts an integer and a non-integer part which overlaps with less informed digits. The relation between any two adjacent CVNS digits is given by

$$. ((x))_i = \lfloor ((x))_i \rfloor + \frac{((x))_{i-1}}{\beta} \quad (3)$$

Where $\lfloor . \rfloor$ represents floor function and $\lfloor ((x))_i \rfloor = 0,1, \ldots, \beta - 1$ is an associated integer of $((x))_i$ and $\frac{((x))_{i-1}}{\beta}$ is the non integer part of CVNS digit.

B. CVNS Addition

Addition in the CVNS is by summation of the digits without intercommunication. There are no carries in the accepted sense in the CVNS theory, and the circuitry associated with the digit generation and the manipulation shares the information at the digit level.

Considering two values, x and y, where x, $y < |M|$ digit wise CVNS addition is as follows:

$$((z))_i = ((x+y))_i$$
$$= (((x))_i + ((y))_i) mod\beta \quad (4)$$

The modular reduction operation in (4.4) ensures that the CVNS digits of the summation are always within the allowable range of [0,β). Therefore, if an overflow occurs in the lower informed digits, the more informed digit is not affected. The overflow is embedded within the CVNS digits.

Example: CVNS addition of two arbitrary values x = 58.34 and y = 72.89 shown in table 1. Maximum range of representation is considered M = 100, n = 2, and k = 2.

TABLE 1

CVNS Addition between two arbitrary values

i	2	1	0	-1	-2
$((x))_i$	0.5834	5.834	8.34	3.4	4
$((y))_i$	0.7289	7.289	2.89	8.9	9
$((z))_i$	1.3123	3.123	1.23	2.3	3

The CVNS Digits of ((z)) are {1.3123, 3.123, 1.23, 2.3, 3}, which is equivalent to z = 131.23 = 58.34 + 72.89.

III. CONTROL SIGNALS

CVNS is implemented by VLSI analog circuits. To reduce the design time of the adder, a regular architecture has been employed throughout the 64 - bit adder. The 64 - bit reconfigurable adder is divided into four 16 - bit adders, and each adder is divided into four uniform blocks. Inputs and Outputs of the system are in the binary form. In regular mode of operation, the system performs four parallel 16 - bit additions. To change the mode of operation of the 64 - bit adder two signals are used, namely: *part*1 and *part*2 as shown in the Table2.

. TABLE 2

Adder operation for different word lengths controlled by part1 and part2 signals

Part1	*Part2*	Adder configuration
0	0	Byte(8-bit)
0	1	Half-Word(16-bit)
1	0	Word(32-bit)
1	1	Double word(64-bit)

To partition the 16 - bit adder whenever 8 - bit operation is required, control signal breaks down the size of each of the 16 - bit adders to 8 - bits. This mode of operation is controlled by a signal denoted as ctrl8 which is generated as follows.

$$Ctrl\ 8 \equiv (part\ 1\ V\ part\ 2) \quad (5)$$

Each two of the 16 bit adders are combined to perform 32 - bit addition. The information is exchanged between the two adders, from the less informed adder to the more informed adder, not only in the 32 - bit but also in the 64 - bit mode of operation. From the table 2 by examining, ctrl32 signal is equivalent to

$$Ctrl\ 32 \equiv (part\ 2'\ V\ part\ 1) \quad (6)$$

All four 16 - bit adders are combined to work as a 64 - bit adder when both part1 and part2 signals are one. Therefore, control signals for these configurations are setup as follows:

$$Ctrl\ 64 \equiv (part\ 1\ \Lambda\ part\ 2) \quad (7)$$

These control signals are used to adjust the size and resolution of the adder on-demand.

IV. MATHEMATICAL ANALYSIS

A. Radix-2 16-Bit Addition

The operations of 16-bit CVNS adder operations are converting binary data to the CVNS, adding the CVNS digits, and converting final results back to binary. The maximum range of CVNS and positional number system is as follows:

:

$$B^{m+1} = \beta^{n+1} = M \quad (8)$$

Here, 'm' is the max index value in CVNS representation. 'n' is the max index value in positional number system.

Input digits into each of the 16-bit CVNS adders, x_{i+t}, represent an integer value, x^t, as follows:

$$x^t = \sum_{i=0}^{15} x_{i+t}.2^i \quad (9)$$

Where, t= 0,16,32,48.

By placing the previous term in (2), a direct relation between the binary and the corresponding CVNS digits for each of 16-bit CVNS adders is obtained as follows:

$$((x))_{j+t} = \left(\frac{x^t}{2^{16}}.2^{16-j}\right) mod2$$

$$= \left(\sum_{i=0}^{15} x_{i+t}.2^{i-j}\right) mod2, 0 \le j \le 1 \quad (10)$$

The previous summation represents a direct relationship between the input binary digits, and the CVNS digits. To obtain some insight on the previous expression, we expand this summation for two CVNS digits:

When i=0, $((x))_{j+t}$ becomes

$$((x))_0 = \left(\sum_{i=0}^{15} x_i.2^i\right) mod2$$

$$= (x_0 + 2x_1 + 4x_2 + \cdots) mod2 = x_0 \quad (11)$$

And for i=1

$$((x))_1 = \left(\sum_{i=0}^{15} x_i.2^{i-1}\right) mod2$$

$$= (2^{-1}x_0 + x_1 + 2x_2 + \cdots) mod2$$

$$= 2^{-1}x_0 + x_1 \quad (12)$$

Therefore, the general expression given by expression (10) can be modified to a pre congruent form to eliminate the modular reduction operator (mod2) as follows:

$$((x))_{j+t} = \sum_{i=0}^{j} x_{j+t}.2^{i-j} \quad 0 \le j \le 15 \quad (13)$$

From the system design point of view, this alteration not only simplifies the design and reduces the complexity, but it also decreases the delay of the adder. In this form, every CVNS digit is obtained directly from the input binary digits. In general, since

the delay of D/A conversion is constant, and typically is less than the delay of a modular reduction circuit, the designed system tends to be faster. Using the previous expression, the digit wise summation of two CVNS numbers ((x)), and ((y)) is

$$((z))_{j+t} = \left(((x))_{j+t} + ((y))_{j+t}\right)$$
$$= \left(\sum_{i=0}^{j}(x_{i+t} + y_{i+t})\, 2^{i-j}\right) mod 2 \qquad (14)$$

The modular reduction unit should be included in the previous summation, since added value of two CVNS digits may exceed the radix range. Fig. 2 & 3 shows the block level representation of the improvements that were made in the CVNS adder. Fig. 2 shows the original form of the CVNS addition between two binary values when digits are generated based on, while Fig. 3 shows the improvements made by eliminating the continuous modular reduction operator.

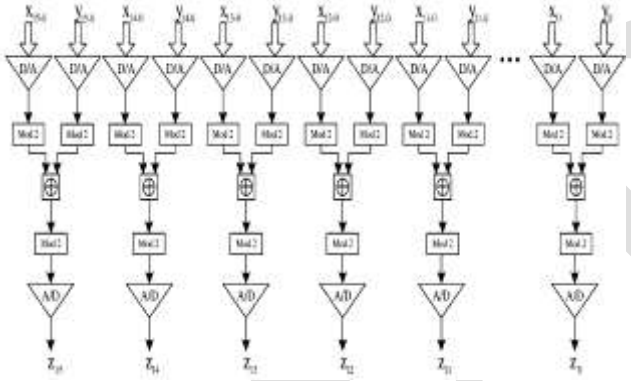

Fig. 2 Original Form of CVNS Addition

In this form, addition between two CVNS values requires conversion from binary to the CVNS representation, addition between the CVNS values, applying the modular reduction on the CVNS summation to adjust the values, and conversion back from the CVNS to binary representation. We are going to eliminate the modular reduction operation from the addition by manipulating the mathematical expressions to speed up the addition.

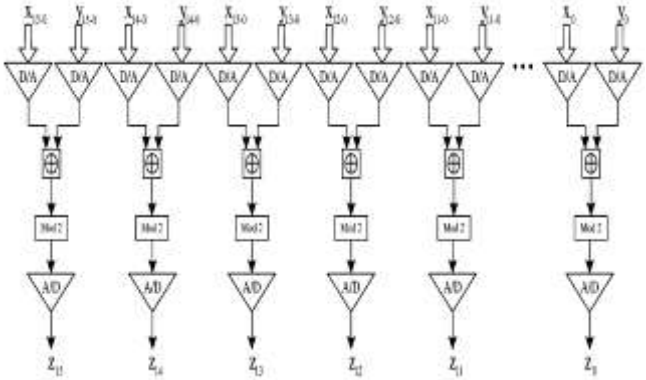

Fig. 3 Improvements made by manipulating the mathematical analysis of the CVNS addition, eliminating one gate in the addition

A CVNS digit as shown by the (10) is composed of two parts; an integer part and a non-integer part, which is shared with its less informed digits. By applying the (10) in the previous expression, the previous summation term can be expressed as follows:

$$((z))_{j+t} = \left(x_{j+t} + y_{j+t} + 2^{-1}\sum_{i=0}^{j-1}(x_{i+t} + y_{i+t})\, B^{i-j}\right) mod 2 \qquad (15)$$

Or $$((z))_{j+t} = \left(x_{j+t} + y_{j+t} + 2^{-1}((z))_{j+t-1}\right) mod 2 \qquad (16)$$

The term $((z))_{j+t}$ are in the CVNS form and have to be converted back to binary form. For values of $((z))_{j+t} \in [0,1)$ binary outcome digit, $((z))_{j+t}$, is equal to 0, and for $((z))_{j+t} \in [1,2)$ it is equal to 1. At this stage, low radix of the CVNS allows us to remove modular reduction operation (mod2) and CVNS to binary conversion with a simple XOR, and to generate the binary outcome of the adder as follows:

$$z_{j+t} = x_{j+t} \oplus y_{j+t} \oplus xy_{j+t} \qquad (17)$$

Where $\oplus$ denotes the logical XOR function, and

$$xy_{j+t} = \begin{cases} 0, & if\ 0 \le 2^{-1}((z))_{j+t-1} < 1 \\ 1, & if\ 0 \le 2^{-1}((z))_{j+t-1} < 2 \end{cases} \qquad (18)$$

Therefore, by manipulating the mathematical expressions of the CVNS addition, we have been able to reduce the adder design complexity, and simplified the system design. In this form, addition of two binary values is performed in a mixed-signal format. While analog values provide carry information locally, hence reducing the number of interconnections, final outcome is computed by digital circuits. Fig. 4 shows the block level representation of the mixed-signal CVNS adder.

In the resulted mixed signal from of the CVNS addition, the modular reduction, and analog-to-digital (A/D) conversion are replaced with a simpler XOR gate. There is, however, another important design issue that has to be addressed before attempting to develop the adder circuitry. The CVNS representation is an analog (continuous) number system, and is therefore implemented by analog circuits. In order to implement the CVNS adder, a high resolution analog environment, here equivalent to 14 bits is required. This condition cannot be satisfied in most targeted analog technologies.

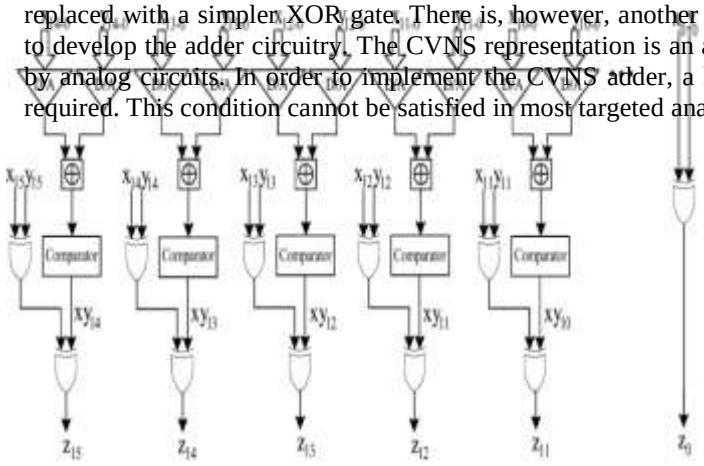

Fig. 4 Block level representation of the mixed-signal CVNS addition

If analog environment distinguishes 2^{φ} different levels, in order to obtain a reliable value for the $((z))_{j+t+1}$,digit generation and addition has to be modified by performing the summation over a fixed-size group of bits of length ψ' with ψ >1.This parameter is technology dependent, limited by the maximum reliable resolution of environment. In this paper, we have chosen it to be equal to ψ=4, which not only can be easily implemented by reliable current-mode analog circuits in our target technology, but also simplifies the partitioning scheme. Addition over a group of digits called Truncated Addition [5], for 16-bit summation with is ψ=4 .

$$((z))_{j+t-1} = \left(\sum_{i=4\left\lceil \frac{j+2}{4} \right\rceil}^{j-1} (x_{i+t} + y_{i+t})\, 2^{i-j} + 2^{4\left\lceil \frac{j+2}{4} \right\rceil + 6 - j} Tr_{j+t+1} \right) mod\beta \quad (19)$$

Where$Tr_{14-j+t}$is called the *truncation signal* and is equal to

$$\mathrm{Tr}_{j+t+1} = \left(\begin{matrix} \mathrm{gt}_{(k+t)} + \mathrm{rt}_{(k+t)} \mathrm{gt}_{(k+t-1)} + \cdots \\ + \mathrm{rt}_{(k+t)} \mathrm{rt}_{(k+t-1)} \cdots \mathrm{rt}_{(2+t)} \mathrm{rt}_{(1+t)} C_{\mathrm{in}_t} \end{matrix} \right) \quad (20)$$

Where *gt* and *rt* signals are

$$\mathrm{gt}_{(k+t)} = \begin{cases} 1, & \text{if } \sum_{i=4k-4}^{4k-1} (x_{i+t} + y_{i+t}) 2^{i-4k+4} \geq 2 \\ 0, & \text{otherwise} \end{cases} \quad (21)$$

$$rt_{(k+t)} = \begin{cases} 1, & \text{if } \sum_{i=4k-4}^{4k-1} (x_{i+t} + y_{i+t}) 2^{i-4k+4} \geq 1.875 \\ 0, & \text{otherwise} \end{cases} \quad (22)$$

And *k* is an integer equal to $\left\lceil \frac{(11-j)}{4} \right\rceil$ and, *t* = 0, 16, 32, 48. In this approach, the required resolution of the analog environment is reduced to 4 bits; however, the arithmetic unit is able to process data with much longer word length. In fact, this form of the CVNS addition is similar to the carry-look-ahead concept in digital circuits, except that the addition is performed in analog domain. The truncation signals provide an estimate of the analog digit.

The resulted 64-bit adder is a mix of both classical binary circuits such as XOR for generating the output and CVNS style circuits for evaluating terms such as (24) and (25). Analog circuits in this design are the front circuits, and are used for processing a group of input bits with higher speed and fewer interconnections. These analog blocks detect the existence of the truncation signals within a group of binary inputs. Digital circuits are at the output stage of each adder and provide the required driving capability for various interconnection loadings in different adder Configurations.

Equation (23) indicates that only a limited number of truncation signals are required within the 16-bit adders. Because of the low number of interconnections, control and partitioning of the adders is performed with less complexity. The number of truncation

signals depends on the chosen length of the groups. Based on the chosen group length in this design (*ψ=4),* in each of the 16-bit adders, the three truncation signals are as follows:

$$Tr_{t+4:t+7} = gt_{(1+t)} + rt_{(1+t)} C_{in_t} \quad (23)$$

$$Tr_{t+8:t+11} = ctrl8.in_{8+t} + ctrl8' \begin{pmatrix} gt_{(2+t)} + rt_{(2+t)} gt_{(1+t)} \\ + rt_{2+t} rt_{(1+t)} C_{in_t} \end{pmatrix} \quad (24)$$

$$Tr_{t+12:t+15} = gt_{(3+t)} + rt_{(3+t)} \times \left(ctrl8' \times \begin{pmatrix} ctrl8.in_{8+t} + \\ gt_{(2+t)} + rt_{(2+t)} gt_{(1+t)} + \\ rt_{2+t} rt_{(1+t)} C_{in_t} \end{pmatrix} \right) \quad (25)$$

Where C_{in} is the carry input to each of the adders. The expression for this signal is derived in the next section. The three truncation signals given by the above equations are the only signals that are passed from the second layer of the adder to the third. The carry out of the adder is generated in the same style.

B. Radix- 2 64- bit Addition

The reconfigurable 64-bit adder is generated by cascading four 16-bit radix-2 CVNS adders. This CVNS adder has a uniform design, making it suitable for reconfigurable media processing and SoC applications. In this adder, information is generated locally for addition hence; the number of required interconnections is reduced. The only global information is the carries out of each 16-bit adder. Between the four adders, there are eight truncation signals, which are similar to the *gt* and *rt* signals inside each adder, as follows:

$$gt_{(t:t+15)} = (gt_{(t+12:t+15)} + rt_{(t+12:t+15)} gt_{(t+8:t+11)} + rt_{(t+12:t+15)} rt_{(t+8:t+11)} gt_{(t+4:t+7)} + rt_{(t+12:t+15)} rt_{(t+8:t+11)} \times rt_{(t+4:t+7)} gt_{t:t+3}) \quad (26)$$

$$rt_{(t:t+15)} = rt_{(t+12:t+15)} rt_{(t+8:t+11)} \times rt_{(t+4:t+7)} rt_{(t:t+3)} \quad (27)$$

Where t =0, 16, 32, 48

The propagation of these signals is controlled by *Ctrl* 32 and *Ctrl* 64. Input carry into each of the 16-bit adders, which is shown as *Cin* $_t$, is as follows:

$$cin_{16} = crtl32' \left(gt_{(0:15)} + rt_{(0:15)} . in_0 \right) + ctrl32 . in_{16} \quad (29)$$

$$cin_{32} = crtl64' . in_{32} + crtl\ 64 \begin{pmatrix} gt_{(16:31)} + \\ rt_{(16:31)} gt_{(0:15)} \\ + rt_{(16:31)} rt_{(0:15)} . in_0 \end{pmatrix} \quad (30)$$

$$cin_{48} = crtl32' . in_{48} + crtl\ 32 \begin{pmatrix} gt_{(32:47)} + \\ rt_{(32:47)} (crtl64' . in_{32} \\ + crtl\ 64 \begin{pmatrix} gt_{(16:31)} + \\ rt_{(16:31)} \\ gt_{(0:15)} \\ + rt_{(16:31)} \\ rt_{(0:15)} \\ . in_0 \end{pmatrix} \end{pmatrix} \quad (31)$$

V. RESULTS

Cadence analysis tools were used for simulating and synthesize RTL schematic diagram for the 64-Bit addition the proposed design Fig. 5 & Fig. 6. Comparing the ripple carry adder with CVNS adder, less power consumption is observed in CVNS adder Fig.7. The Table 3 shows Comparison between 64-bit Ripple Carry Adder and 64-bit CVNS Adder synthesis reports like area, timing and power.

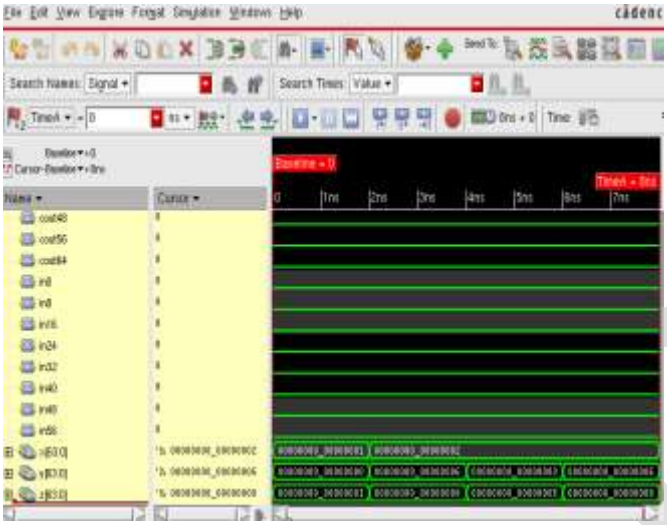

Fig. 5 Simulated output of 64-bit mode

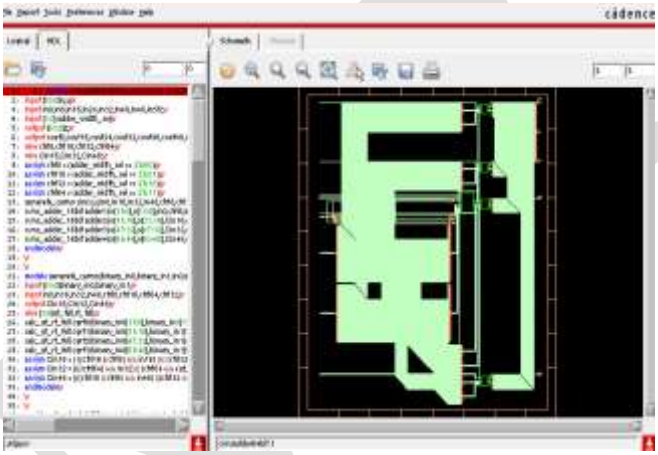

Fig. 6 schematic diagram of 64-bit addition

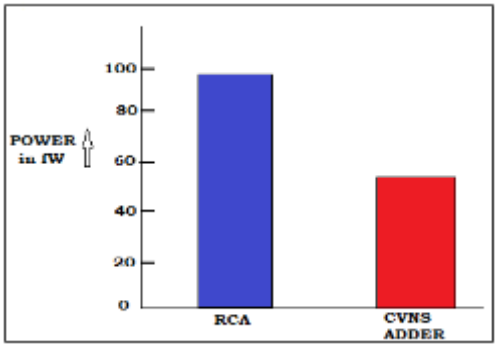

Fig. 7 Comparison of Power Consumption

TABLE 3

Comparison between 64-bit Ripple Carry Adder and 64-bit CVNS Adder synthesis reports

Synthesis report	Area(μm^2)	Power(fW)
RCA	4558	98.55
CVNS	3995	59.87

VI. CONCLUSION

The CVNS adder has been implemented in Verilog language. Area and power consumption of the 64-bit adder is analyzed using 64-bit CVNS adder and ripple carry adder and also compared the results. The proposed Adder has been found advantageous over ripple carry adder for its low power dissipation. Work in future can consider area reduction techniques along with the above parameter optimization by using 45nm or less technology.

REFERENCES

[1] A. Farooqui, V. Oklobdzija, and F. Chechrazi, "Multiplexer based adder for media signal processing, "in Proc IEEE Int Symp VLSI Tech, Syst, Appl, pp 100–103, 1999.

[2] S. Perri, P. Corsonella, and G. Cocorullo, "A high speed energy efficient 64-bit reconfigurable binary adder, "IEEE Trans Very Large Scale Integer (VLSI) Syst, vol 11, no 5, pp 939–943, May 2003.

[3] A. Saed, M. Ahmadi, and G. A. Jullien, "A number system with continuous valued digits and modulo arithmetic," IEEE Trans Comput, vol 51, no 11, pp 1294–1304, Nov 2002.

[4] M. Mirhassani, M. Ahmadi, and G. A. Jullien, "Digital multiplication using continuous valued digits, " in Proc IEEE Int Symp Circuits Syst (ISCAS), pp 3263–3266, 2007

[5] J.-F. Li, J.-D. Yu and Y.-J. Huang, "A design methodology for hybrid carry-lookahead/carry select adders with reconfigurability, " in Proc IEEE Int Symp Circuits Syst (ISCAS), vol 1, pp 77–80, 2005.

[6] M. Mirhassani, M. Ahmadi, and G. A. Jullien, " 16-bit binary multiplication using high radix analog digits, in Proc IEEE Asilomar Conf Sig, Syst, Comput, vol 9, pp 332–336, 2006.