International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 2 Issue 12, December - 2013

Low-Power and Area-Efficient Carry Select Adder

P. Sirish Kumar [a] S. Kiran [b] T. Lakshmana Rao [c]

a: Assistant Professor, Department of Electronics and Communication Engineering, Aditya Institute of Technology and Management, Tekkali-532201.
b: M.Tech (2nd Year), Department of Electronics and Communication Engineering, SISTAM College of Engineering, Ampolu, Sriakulam, AP-532404.
c. Assistant Professor, Department of Electronics and Communication Engineering, SISTAM College of Engineering, Ampolu, Sriakulam, AP-532404.

ABSTRACT:

D*esign of power-efficient and high-speed data path logic systems are one of the most substantial areas of research in VLSI system design. In digital adders, the speed of addition is limited by the time required to propagate a carry through the adder. The sum for each bit position in an elementary adder is generated sequentially only after the previous bit position has been summed and a carry propagated into the next position.*

The CSLA is used in many computational systems to alleviate the problem of carry propagation delay by independently generating multiple carries and then select a carry to generate the sum. However, the CSLA is not area efficient because it uses multiple pairs of Ripple Carry Adders (RCA) to generate partial sum and carry by considering carry input, then the final sum and carry are selected by the multiplexers (mux). The basic idea of this work is to use Binary to Excess-1 Converter (BEC) instead of RCA in the regular CSLA to achieve high speed and low power consumption.

Keywords: *CSLA, RCA, BEC, area-efficient, low power, propagation delay.*

1. Introduction:

Design of high speed data path logic systems are one of the most substantial research area in VLSI system design. High-speed addition and multiplication has always been a fundamental requirement of high-performance processors and systems.

The major speed limitation in any adder is in the production of carries and many authors have considered the addition problem. The basic idea of the proposed work is using n-bit Binary to Excess-1 Converters (BEC) to improve the speed of addition. This logic can be implemented with Carry Select Adder to Achieve Low Power and Area Efficiency. The proposed 32-bit Carry Select Adder compared with the Carry Skip Adder (CSKA) and Regular 32-bit Carry Select Adder.

The main advantage of this Binary to Excess Converter (BEC) is logic comes from the lesser number of logic gates than the n- bit Ripple Carry Adder (RCA). A structure of 4-bit Binary to Excess Converter (BEC) and the truth table is shown in Fig.1.1 and Table 1 respectively.

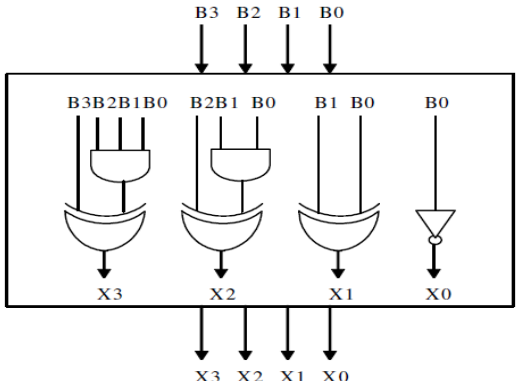


Fig: 1.1:4-bit Binary to Excess-1 Converter (BEC)

Table-1:

B[3:0]	X[3:0]
0000	0001
0001	0010
⋮	⋮
1110	1111
1111	0000

Table.1: Functional Table of 4-Bit BEC

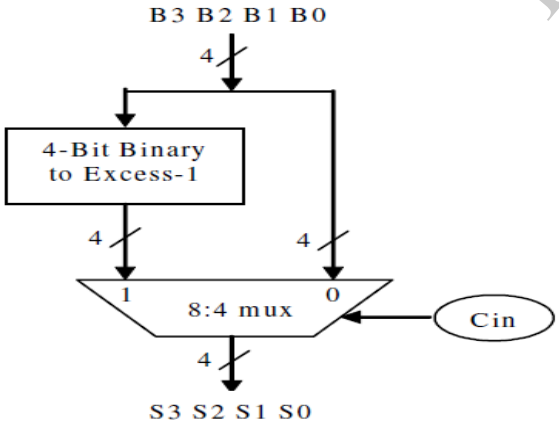


Fig.1.2. 4-b BEC with 8:4 mux

How the goal of fast addition is achieved using BEC together with a multiplexer (mux) is described in Fig.1.2, one input of the 8:4 mux gets as it input (B3, B2, B1, and B0) and another input of the MUX is the BEC Output.

This produces the two possible partial product results in parallel and the Muxes are used to select either BEC output or the direct inputs according to the control signal C_{in}.

The Boolean expressions of 4-bit BEC are listed below, (Note: functional symbols, ~ NOT, & AND, ^ XOR).

X0 = ~ B0
X1 = B0 ^ B1
X2 = B2 ^ (B0 & B1)
X3 = B3 ^ (B0 & B1 & B2)

2. Carry Select Adder:

A carry-select adder is divided into sectors, each of which – except for the least-significant –performs two additions in parallel, one assuming a carry-in of zero, the other a carry-in of one. A four bit carry select adder generally consists of two ripple carry adders and a multiplexer. The carry-select adder is simple but rather fast, having a gate level depth of O (√n). Adding two n-bit numbers with a carry select adder is done with two adders (two ripple carry adders) in order to perform the calculation twice, one time with the assumption of the carry being zero and the other assuming one.

After the two results are calculated, the correct sum, as well as the correct carry, is then selected with the multiplexer once the correct carry is known. The design schematic of Carry Select Adder is shown in Fig.2.1.

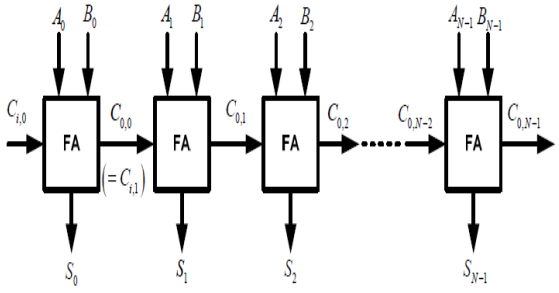


Fig.2.1: The N-bit Ripple Carry Adder Constructed by N set single bit Full-adder

In the N-bit carry ripple adder, the delay time can be expressed as:

TCRA = (N-1) Tcarry + Tsum

In the N-bit carry select adder, the delay time is:

TCSA = Tsetup + (N/M) Tcarry + MTmux + Tsum

In our proposed N-bit area-efficient carry select adder, the delay time is:

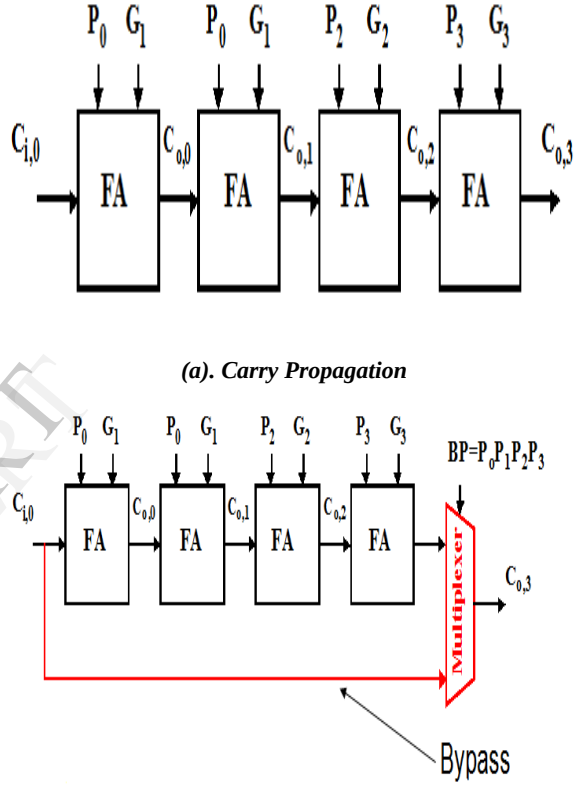
Tnew = Tsetup + (N-1) Tmux + Tsum

The carry select adder comes in the category of conditional sum adder. Conditional sum adder works on some condition. Sum and carry are calculated by assuming input carry as 1 and 0 prior the input carry comes. When actual carry input arrives, the actual calculated values of sum and carry are selected using a multiplexer.

The conventional carry select adder consists of k/2 bit adder for the lower half of the bits i.e. least significant bits and for the upper half i.e. most significant bits (MSB's) two k/bit adders. In MSB adder's one adder assumes carry input as one for performing addition and another assumes carry input as zero. The carry out calculated from the last stage i.e. least significant bit stage is used to select the actual calculated values of output carry and sum. The selection is done by using a multiplexer. This technique of dividing adder in to stages increases the area utilization but addition operation fastens.

3. Carry Skip Adder:

A carry-skip adder consists of a simple ripple carry-adder with a special speed up carry chain called a skip chain. Carry skip adder is a fast adder compared to ripple carry adder when addition of large number of bits take place; carry skip adder has O(√n) delay provides a good compromise in terms of delay, along with a simple and regular layout This chain defines the distribution of ripple carry blocks, which compose the skip adder. A carry-skip adder is designed to speed up a wide adder by aiding the propagation of a carry bit around a portion of the entire adder. Actually the ripple carry adder is faster for small values of N. However the industrial demands these days, which most desktop computers use word lengths of 32 bits like multimedia processors, makes the carry skip structure more interesting. The basic structure of Carry Skip Adder is shown in Fig.3.1.



(a). Carry Propagation

(b).Adding By Pass

Fig.3.1. Carry skip adder structure – basic concept

3.1.1. 32-bit Carry Skip Adder

A carry skip divides the words to be added in to groups of equal size of k-bits. Carry Propagate pi signals may be used within a group of bits to accelerate the carry propagation. If all the pi signals within the group are pi=1, carry bypasses the entire group as shown in Fig.3.1.1.

P = pi * pi+1 * pi+2 *… pi+k

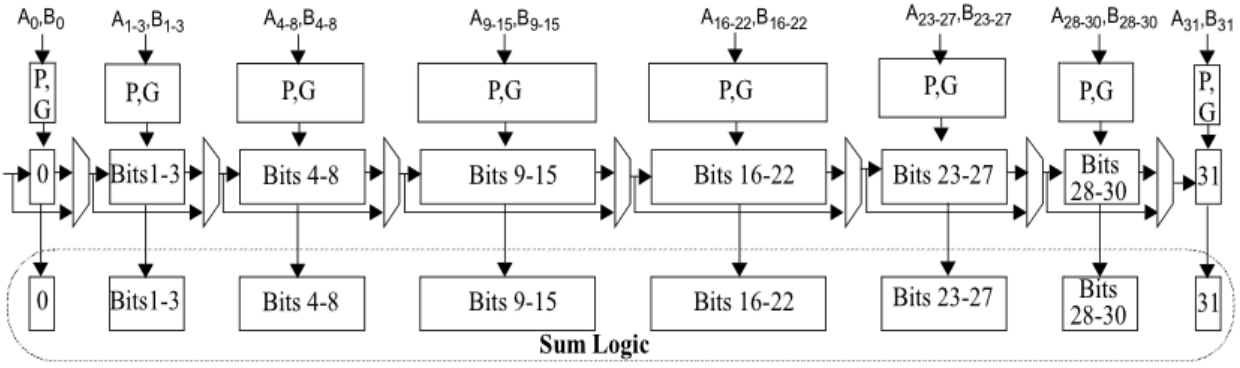


Fig.3.1.1. 32-bit Carry skip adder

In this way delay is reduced as compared to ripple carry adder. The worst-case carry propagation delay in a N-bit carry skip adder with fixed block width b, assuming that one stage of ripple has the same delay as one skip, can be derived:

TCSKA = (b -1) +0.5+ (N/b-2) + (b -1)
= 2b + N/b – 3.5 Stages

4. VLSI:

VLSI stands for "Very Large Scale Integration". This is the field which involves packing more and more logic devices into smaller and smaller areas.

- Simply we say Integrated circuit is many transistors on one chip.
- Design/manufacturing of extremely small, complex circuitry using modified semiconductor material.
- Integrated circuit (IC) may contain millions of transistors, each a few mm in size.
- Applications wide ranging: most electronic logic devices.

4.1 VLSI Design Flow

4.1.1 Digital Circuit:

Digital ICs of SSI and MSI types have become universally standardized and have been accepted for use. Whenever a designer has to realize a digital function, he uses a standard set of ICs along with a minimal set of additional discrete circuitry. Consider a simple example of realizing a function as

$$Q\,n+1 = Q\,n + (A\,B)$$

Here on, *A*, and *B* are Boolean variables, with *Q n* being the value of *Q* at the *n*th time step. Here *A B* signifies the logical AND of *A* and *B*; the '+' symbol signifies the logical OR of the logic variables on either side. A circuit to realize the function is shown in Figure. The circuit can be realized in terms of two ICs – an A-O-I gate and a flip-flop. It can be directly wired up, tested, and used.

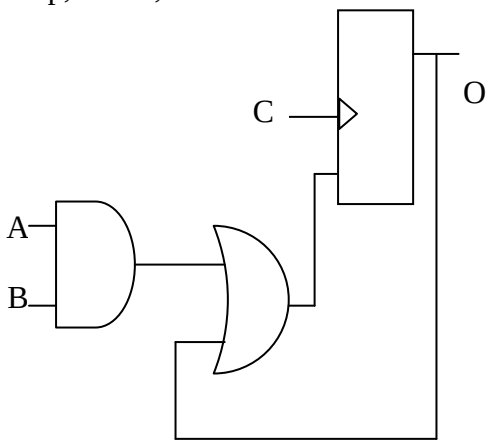


Fig 4.1.1(a): Simple digital circuit

With comparatively larger circuits, the task mostly reduces to one of identifying the set of ICs necessary for the job and interconnecting; rarely does one have to resort to a micro level design. The accepted approach to digital design here is a mix of the top-down and bottom-up approaches as follows.

• Decide the requirements at the system level and translate them to circuit requirements.

• Identify the major functional blocks required like timer, DMA unit, register file *etc*., and say as in the design of a processor.

• Whenever a function can be realized using a standard IC, use the same –for example programmable counter, mux, demux, *etc*.

• Whenever the above is not possible, form the circuit to carry out the block functions using standard SSI – for example gates, flip-flops, *etc*.

• Use additional components like transistor, diode, resistor, capacitor, *etc*., wherever essential.

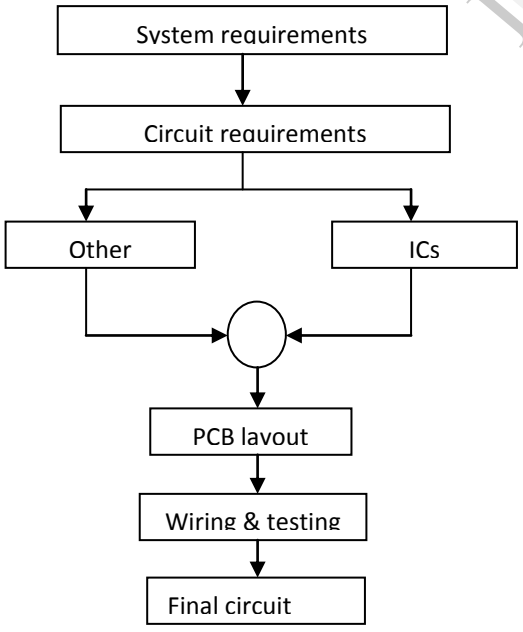


Fig 4.1.1(b): Process flowchart

Once the above steps are gone through, a paper design is ready. Starting with the paper design, one has to do a circuit layout. The physical location of all the components is tentatively decided; they are interconnected and the 'circuit-onpaper' is made ready. Once a paper design is done, a layout is carried out and a net-list prepared. Based on this, the PCB is fabricated and populated and all the populated cards tested and debugged.

5. ROLE OF VHDL:

VHDL is an acronym for Very High Speed Integrated Circuits Hardware description Language. The language can be used to model a digital system at many levels of abstraction ranging from the algorithmic level to the gate level. The complexity of the digital system being modeled could vary from that of a simple gate to a complete digital electronic system. The VHDL language can be regarded as an integrated amalgamation of sequential, concurrent, net list and waveform generation languages and timing specifications.

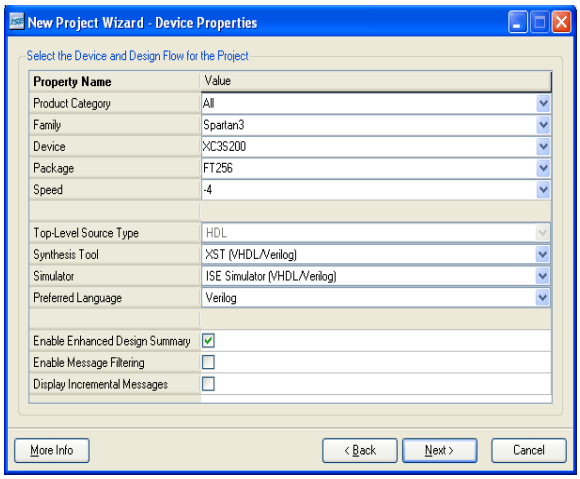
6. Software Used:
XILINX:

Xilinx software is used by the VHDL/VERILOG designers for performing Synthesis operation. Any simulated code can be synthesized and configured on FPGA. Synthesis is the transformation of VHDL code into gate level net list. It is an integral part of current design flows.

Algorithm:

Start the ISE Software by clicking the XILINX ISE icon.

Create a New Project and find the following properties displayed.



Create a VHDL Source formatting all inputs, outputs and buffers if required. Which provides a window to write the VHDL code, to be synthesized?

6. RESULTS:

CSLA WITH RCA IMPLEMENTATION SYNTHESIS REPORT:

Release 8.2i - xst I.31

Copyright (c) 1995-2006 Xilinx, Inc. All rights reserved.

--> Parameter TMPDIR set to ./xst/projnav.tmp

CPU: 0.00 / 0.32 s | Elapsed : 0.00 / 0.00 s

--> Parameter xsthdpdir set to ./xst

CPU: 0.00 / 0.32 s | Elapsed : 0.00 / 0.00 s

--> Reading design: CSLA_32_RCA.prj

POWER REPORT OF CSLA WITH BEC:

Started: "Generate Power Data".

Loading device for application Rf_Device from file 'v50.nph' in environment C:\Xilinx.

"CSLA_32" is an NCD, version 3.1, device xcv50, package bg256, speed -5

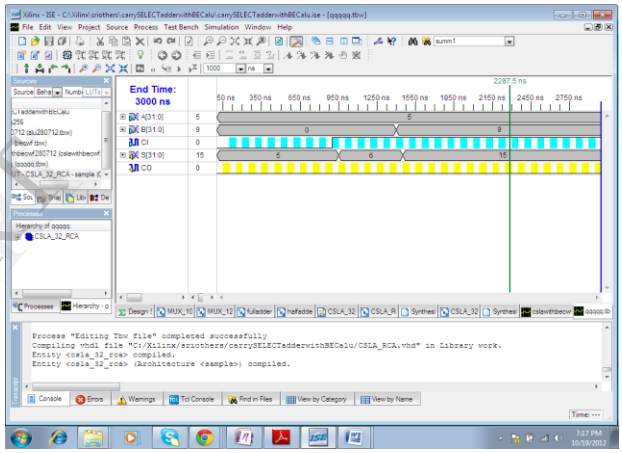
Design load 35% complete

Process "Generate Power Data" completed successfully.

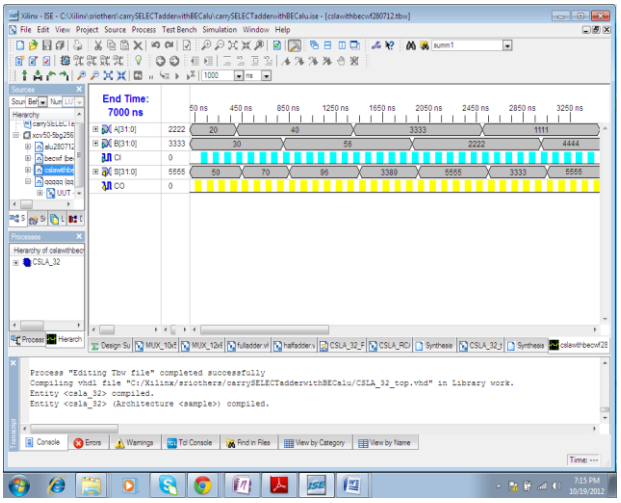
❖ **Comparison Table Between CSLA Adders with RCA and BEC in terms of timing (delay) and power by using the device :** xcv50-5-bg256 to analysis

	TIMING REPORT	POWER
CSLA WITH BEC	36.876ns	7mw
CSLA WITH RCA	51.536ns	27mw

❖ **WAVEFORM OF CSLA WITH BEC:**



❖ **WAVEFORM OF CSLA WITH RCA:**



7. CONCLUSION:

Addition is the most common and often used arithmetic operation on microprocessor, digital signal processor, especially digital computers. Also, it serves as a building block for synthesis all other arithmetic operations. Therefore, regarding the efficient implementation of an arithmetic logic unit, the adder structures become a very critical hardware unit.

In any book on computer arithmetic, someone looks that there exists a large number of different circuit architectures with different performance characteristics and widely used in the practice. Although many researches dealing with the adder structures have been done, the studies based on their comparative performance analysis are only a few.

Digital Adders are the core block of DSP processors. The final carry propagation adder (CPA) structure of many adders constitutes high carry propagation delay and this delay reduces the overall performance of the DSP processor. In this project, qualitative evaluations of the CSLA adder with and without BEC architectures are given. Among the huge member of the adders we wrote VERILOG (Hardware Description Language) code for Carry skip and carry select adders to emphasize the common performance properties belong to their classes. With respect to delay time and power consumption we can conclude that the implementation of CSLA with BEC is efficient. The main advantage of this BEC logic comes from the lesser number of logic gates than the n-bit Full Adder (FA) structure.

Now a day's Carry Select Adder (CSLA) used in many ~~da~~ta-processing processors to perform fast arithmetic functions. That's why; we have designed a configurable adder with minimal delay overhead, and power efficient. CSLA RCA can be replaced by CSLA BEC Where the speed and power are the major constraints. The proposed CSLA BEC consumes only 17mw which is very less when compare to the existing CSLA RCA which consumes 37mw.

8. REFERENCES:

[1] B. Ramkumar, H.M. Kittur, and P. M.Kannan, "ASIC implementation ofmodified faster carry save adder," *Eur. J.Sci. Res., vol. 42, no. 1, pp. 53–58, 2010.*

[2] D. Radhakrishnan, "Low-voltage low power CMOS full adder," in *Proc. IEEE Circuits Devices Syst.*, vol. 148, Feb. 2001.

[3] E. Abu-Shama and M. Bayoumi, "A new cell for low power adders," in Proc. Int. Midwest Symp. Circuits and Systems, 1995, pp. 1014–1017.

[4] T. Y. Ceiang and M. J. Hsiao, "Carry select Adder using single ripple carry Adder," *Electron. Lett.*, vol. 34, no. 22, pp. 2101–2103, Oct. 1998.

[5] J. M. Rabaey, *Digital Integrated Circuits—A Design Perspective*. Upper Saddle River, NJ: Prentice-Hall, 2001.

[6] J. Storer andR. J. Turyn, "On binary sequences," in *Proc. Amer. Math. Soc.*, vol. 12, 1961, pp. 394–399.

[7] M. J. E. Golay, "The merit factor of long low autocorrelation binary sequences," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 543–549, May 1982.

[8] J. Lindner, "Binary sequences up to length 40 with best possible auto-correlation function," *Electron. Lett.*, vol. 2, p. 507, 1975.

[9] S. Mertens, "Exhaustive search for low-autocorrelation binary se-quences," *J. Phys. A*, vol. 29, pp. L473–L481, 1996.

[10] . (2002) the Bernasconi Model. [Online]. Available: http://odysseus.nat.uni-magdeburg.de/~mertens/bernasconi/