Implementation Of 32-Bit Sqrt Carry Select Adder Using Bec-1

ROJA.M
UG Student/Department of ECE
SNS College of Technology, Coimbatore.

NISHANA.S
UG Student/Department of ECE
SNS College of Technology, coimbatore.

SARAVANA.V.A.
UG Student/Department of ECE
SNS College of Technology, Coimbatore.

PRIYANKA.S
UG Student/Department of ECE
SNS College of Technology, Coimbatore.

ABSTRACT:

In digital circuits,addition is performed using adder circuits such as full adder,half adder,carry look ahead adder,etc. But these logic circuits requires more area,since area plays a major role. CSLA is one of the fastest adders which is used to perform arithmetic operations fastly in many data processing processors. This paper proposes the idea of efficient carry select adder,using 32-bit by apportioning the binary to excess converter(BEC). By using multiplexer, the correct output is picked out according to the summation operation using XOR gate and inverter gate as well as the carry out operation using AND gate and inverter gate. Thus the design architecture of 32 bit SQRT carry select adder(CSLA) has been formulated using BEC. The suggested architecture has reduced area and delay while comparing the normal SQRTCSLA. The result analytic thinking shows that the aimed SQRT CSLA design is good than the regular SQRTCSLA.

Keywords: *Binary to Excess -1 converter (BEC), Common Boolean Logic (CBL), Area efficient, SQRTCSLA (square root carry select adder).*

INTRODUCTION:

In VLSI system design,power and area efficient high speed data logic systems is one of the significant areas of explore. Usually addition collide with widely the overall performance of arithmetic operations and digital systems.Adders are most widely used in electronic applications and in multipliers,DSP to execute different algorithms like FIR,IIR,FFT. In digital adders .the speed of addition is limited by the time necessitated to propagate a carry via the adder. The sum bit generated sequentially only after the previous bit position has been summed and a carry propagated into the next position.

In many computational systems , CSLA is used to palliate the problem of carry propagation delay by independent generation of multiple carriers and then select a carry to generate the addition. Nevertheless, the CSLA is not area efficient since it use multiple pairs of ripple carry addesr(RCA) to generate partial sum and carry by assuming carry input as cin=0, cin=1. With the help of multiplexers,final summation and carry are selected.The carry select adders(CSLA) are categorized into squqre root carry select adder and linear carry select adder.

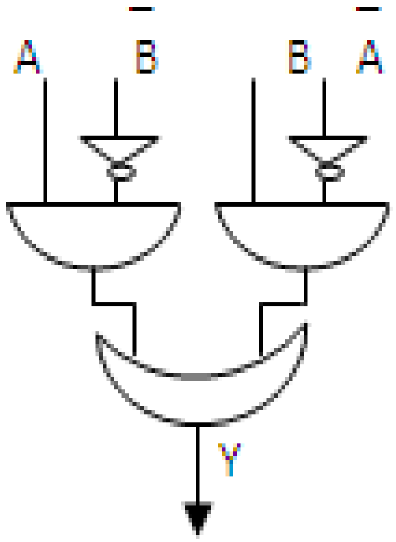


Fig: 1 area and delay evaluation

ARCHITECTURE OF CARRY SELECT ADDER:

The square root carry select adder is fabricated using the delay through two carry Chain as well as the block multiplexer.the other name of this adder is non linear carry select adder. The carry-select adder generally consists of two ripple carry adders and a multiplexer. Adding two n-bit numbers with a carry-select adder is done with two adders (therefore 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 modified SQRTCSLA uses Binary to Excess-1 converter(BEC) instead of RCA with cin=1 to reach lower delay and marginal increase in area.

DESIGN OF LINEAR ADDER:

The linear carry select adder is fabricated by chaining a number of equal length adder stages. For an n-bit adder, it could be carried out with equal length of carry select adder and so it is known to be linear carry select adder.

This paper is master minded as section II and III as proposed carry save adder and normal CSLA using RCA respectively. The part IV and V explicates the altered structure of SQRT CSLA using BEC-1 and valuation of delay, area respectively. The section VI and VII deals with the result analysis and conclusion.

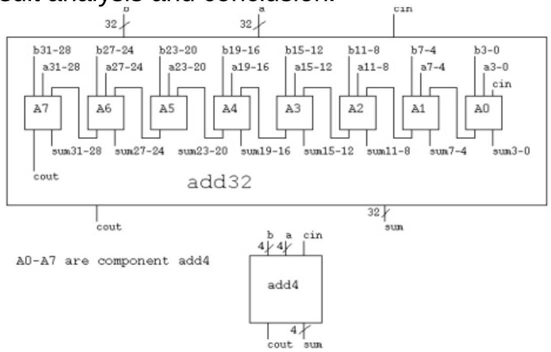


Fig:2 32 bit carry save adder (CSA)

II. PROPOSED CARRY SAVE ADDER:

In computer micro architecture for calculating the sum of three or more *n*-bit numbers we use a carry save adder in binary which is one type of digital adder. This adder is somewhat different from other digital adders because it outputs two numbers which are of same dimensions as that of inputs, in which one is a sequence of sum bits and the other is a sequence of carry bits. In carry-save unit, it comprise of *n* full adders, each of which executes a single sum and carry bit based on the related bits of the three input numbers. **a,b,c** are the three given *n* - bit numbers which produces a incomplete sum $\mathbf{Ps_i}$ and a shift-carry $\mathbf{Sc_i}$: sum is executed by Switching the carry sequence Sc_i one place to the left, adding a 0 to the foremost (most significant bit) of the partial sum sequence Ps_i. For computing these two together and bringing out the output as *n* + 1-bit value. To add three or more numbers unitedly we use a carry-save adder (CSA) accompanied by a ripple carry adder (RCA) which is more quicker than using two ripple carry adders(RCA). Since a ripple carry adder could not calculate a sum bit without waiting for the former carry bit to be brought out, and so the delay is equal to that of *n* full adders. However, a carry save adder which executes all of its output values in parallel, and has an equal delay as a single full-adder. Thus the total calculation time for both the carry-save adder and the ripple carry adder will be *n* + 1, on the otherhand for two ripple carry adders it would be 2*n*.

III. NORMAL CSLA USING RIPPLE CARRY ADDER:

In digital computers the various logic gates like AND, OR, NOR, NAND etc. performs different arithmetic operations like addition, subtraction, multiplication, division. Between all the four arithmetic operations if we could able to execute addition then it will be easy to do multiplication, subtraction or division. To add two one bit binary numbers, a half adder can be used. To add N-bit binary numbers, it is feasible to create a logical circuit using multiple full adders. Since each carry bit "ripples" to the next full adder, it is named as ripple carry adder. Instead of using full adder at first it can be replaced by a half adder. The block diagram of 32-bit SQRTCSLA using Ripple Carry Adder is shown here below: Ripple carry adder design is very uncomplicated,that allows for a very fast design time. However, the ripple carry adder is comparatively slow, because each full adder must wait for the carry bit to be computed from the preceding full adder. With the help of review of full adder the gate delay can easily be ciphered.There must be three levels of logic for full adder. Nevertheless, the CSLA is not area efficient since it use multiple pairs of ripple carry addesr(RCA) to generate partial sum and carry by assuming carry input as cin=0, cin=1. With the help of multiplexers,final summation and carry are selected. The structure of the 32-b SQRT CSLA using RCA is shown in Fig:(3).

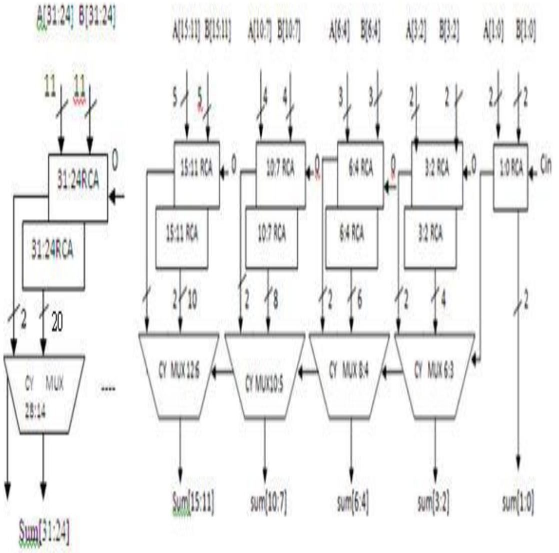


Fig: 3 32 bit CSLA using RCA

IV. ALTERED STRUCTURE OF CARRY SELECT ADDER USING BEC-1

The 32-bit SQRT CSLA using BEC structure is similar to the 32-bit SQRTCSLA using RCA. The minor change is that, we should exchange the RCA with Cin=1 between the two usable RCAs with a BEC. Binary to excess -1 converter could do the same operation like RCA with Cin=1. The figure shows the modified diagram 32-bit SQRT CSLA using binary to excess -1 converter.

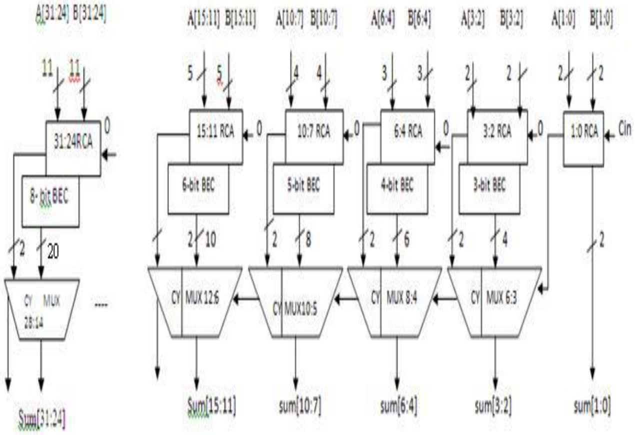


Fig:4 32 bit SQRTCSLA using BEC-1

FINAL REPORT:

```
Final Results
RTL Top Level Output File Name     : sqrtcsla.ngr
Top Level Output File Name         : sqrtcsla
Output Format                      : NGC
Optimization Goal                  : Speed
Keep Hierarchy                     : No

Design Statistics
# IOs                              : 957

Cell Usage :
# BELS                             : 5
#      GND                         : 1
#      LUT3                        : 1
#      LUT4                        : 2
#      MUXF5                    : 1
# IO Buffers                   : 47
#      IBUF                    : 5
#      OBUF                     : 42
=========================================
=====================================

Device utilization summary:
---------------------------

Selected Device : 3s500efg320-4

 Number of Slices:                      2  out of   4656
0%
 Number of 4 input LUTs:                 3  out of
9312     0%
 Number of IOs:                       957
 Number of bonded IOBs:                  47  out of
232    20%

---------------------------
Partition Resource Summary:
---------------------------

  No Partitions were found in this design.

---------------------------



=========================================
=====================================
TIMING REPORT

NOTE: THESE TIMING NUMBERS ARE ONLY A
SYNTHESIS ESTIMATE.
     FOR ACCURATE TIMING INFORMATION
PLEASE REFER TO THE TRACE REPORT
     GENERATED AFTER PLACE-and-ROUTE.

Clock Information:
------------------
No clock signals found in this design

Asynchronous Control Signals Information:
----------------------------------------
No asynchronous control signals found in this
design

Timing Summary:
---------------
Speed Grade: -4

   Minimum period: No path found
   Minimum input arrival time before clock: No path
found
   Maximum output required time after clock: No
path found
   Maximum combinational path delay: 6.557ns

Timing Detail:
--------------
All values displayed in nanoseconds (ns)

=========================================
=====================================
Timing constraint: Default path analysis
  Total number of paths / destination ports: 12 / 2
-------------------------------------------------------------------
------
Delay:              6.557ns (Levels of Logic = 4)
  Source:            a0<1> (PAD)
  Destination:       y62<1> (PAD)

  Data Path: a0<1> to y62<1>
                              Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical
Name (Net Name)
    ----------------------------------------  ------------
     IBUF:I->O            2   1.218   0.622
a0_1_IBUF (a0_1_IBUF)
     LUT4:I0->O           1   0.704   0.000
r1/f2/Mxor_s17_xo<0>21
(r1/f2/Mxor_s17_xo<0>2)
     MUXF5:I1->O          1   0.321   0.420
r1/f2/Mxor_s17_xo<0>2_f5 (y62_1_OBUF)
     OBUF:I->O                3.272
y62_1_OBUF (y62<1>)
    ----------------------------------------
    Total                    6.557ns (5.515ns logic,
1.042ns route)
                                  (84.1% logic, 15.9%
route)

=========================================
=====================================


Total REAL time to Xst completion: 13.00 secs
Total CPU time to Xst completion: 13.38 secs

-->

Total memory usage is 193340 kilobytes

Number of errors   :    0 (   0 filtered)
Number of warnings :  208 (   0 filtered)
Number of infos    :    0 (   0 filtered)
```

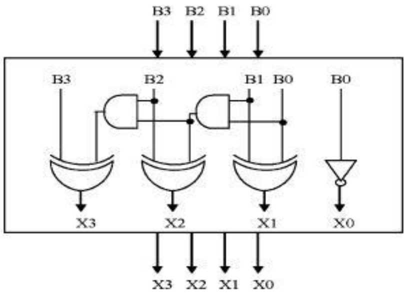


Fig:5 4 bit BEC

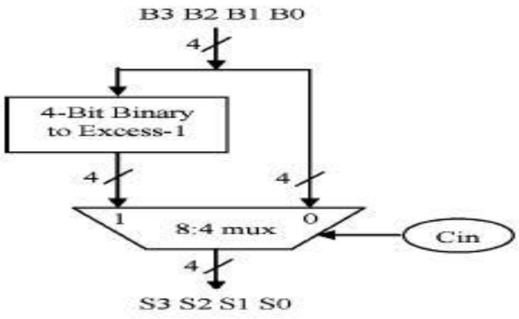
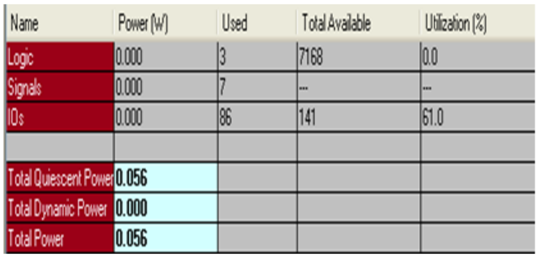


Fig:6 4 bit BEC using 8:4 mux

VI. RESULT ANALYSIS:

The proposed architecture design of 32 bit SQRT CSLA has been simulated using verilog. The output of 32-bit CSLA adder are projected and simulated using verilog and their result analysis are equated with CSA and Carry Select Adder using RCA. The different size codes are combined using Xilinx 13.2 after simulation. Thus the simulated files are analysed and the delay and area values are noted.



Name	Power (W)	Used	Total Available	Utilization (%)
Logic	0.000	3	7168	0.0
Signals	0.000	7	---	---
IOs	0.000	86	141	61.0
Total Quiescent Power	0.056			
Total Dynamic Power	0.000			
Total Power	0.056			

Power report

IX.REFERENCES:

[1]. B.Ramkumar, Harish M Kittur and P.Mahesh Kannan,European Journal of Scientific Research, vol.42, pp.53-58,2010, "ASIC implementation of Modified Faster Carry Save Adder".

[2]. Behnam Amelifard, Farzan Fallah and Massoud Pedram, Sixth International Symposium on Quality of Electronic Design, pp.148-152"Closing the gap between Carry Select Adder and Ripple Carry Adder: a new class of low-power high-performance adders". April 2005

[3]. M.Moris Mano, Pearson Education, 3rd edition, "Digital Design" 2002.

[4]. Kuldeep Rawat, Tarek Darwish. and Magdy Bayoumi, March 2002 45th Midwest Symposium on Circuits and Systems, vol.1, pp. 467-470, "A low power and reduced area Carry Select Adder".

[5]. J. M. Rabaey, New Jersey, Prentice-Hall, "Digital Integrated Circuits- A Design Perspective", 2001.

[6].Youngjoon Kim and Lee-Sup Kim,IEEE International Symposium on Circuits and Systems, vol.4, pp.218-221, "A low power carry select adder with reduced area" May 2001.

[7]. T.-Y. Chang and M.-J. Hsiao Electronicslettervolpp.21021Adder using single Ripple-Carry Adder,October 1998.

AUTHORS PROFILE:

Roja.M, pursuing **B.E.** degree in electronics and communication engineering from the SNS College of Technology, Coimbatore, Anna University, Chennai, India, in 2014. His area of interest includes digital electronics, communication theory.

Nishana.S, pursuing **B.E.** degree in electronics and communication engineering from the SNS College of Technology, Coimbatore, Anna University, Chennai, India, in 2014. His area of interest includes digital electronics, vlsi design.

Saravana.V.A,pursuing **B.E.** degree in electronics and communication engineering from the SNS College of Technology, Coimbatore, Anna University, Chennai, India, in 2014. His area of interest includes networking and digital electronics.

Priyanka.S pursuing **B.E.** degree in electronics and communication engineering from the SNS College of Technology, Coimbatore, Anna University, Chennai, India, in 2014. His area of interest includes digital electronics, vlsi design.