Research Inventy: International Journal of Engineering And Science
Vol.6, Issue 5 (May 2016), PP -61-66
Issn (e): 2278-4721, Issn (p):2319-6483, www.researchinventy.com

Design Of Low Power and Area Efficient Carry Select Adder (CSLA) Using Verilog Language

Guguloth Sreekanth[1,] V Harish[2,] D Mohammad Elias [3]
[1,2,3]*Department Of Electronics and Communication Engineering*
Holy Mary Institute of Technology and Science
Hyderabad, India

ABSTRACT : *Carry select method has deemed to be a good compromise between cost and performance in carry propagation adder design. However conventional carry select adder (CSLA) is still area consuming due to the dual ripple carry adder structure. The excessive area overhead makes conventional carry select adder (CSLA) relatively unattractive but this has been the circumvented by the use of add-one circuit. In this an area efficient modified CSLA scheme based on a new first zero detection logic is proposed. The gate count in 32-bit modified CSLA can be greatly reduced, design proposed in this paper has been developed using VERILOG language and synthesized in XILINX13.2 version.*

I. INTRODUCTION

With the tremendous development in electronics sector, the demand of low power device has significantly increased. As we are using more portable devices we require such a battery that will provide power to the devices for a long time. In order to do these we have to provide a battery of large capacity but that will be significantly increase its size which is not feasible in many cases? Another option is to reduce the power consumption of the device and this will reduce the net power consumption by us and eventually reduce the carbon footprint in the environment.

The carry-ripple adder is composed of many cascaded single-bit full-adders. The circuit architecture is simple and area-efficient. However, the computation speed is slow because each full-adder can only start operation till the previous carry-out signal is ready. In the carry select adder, N bits adder is divided into M parts. Each part of adder is composed two carry ripple adders with cin_0andcin_1 respectively. Through the multiplexer, we can select the correct output result according to the logic state of carry-in signal. The carry-select adder can compute faster because the current adder stage does not need to wait the previous stage's carry-out signal .the summation result is ready before the carry –in signal arrives, therefore, we can get the correct computation result by only waiting for one multiplexer delay in each single bit adder. In the carry select adder, the carry propagation delay can be reduced by M times as compared with the carry ripple adder.

However, the duplicated adder in the carry select adder results in larger area and power consumption. In this paper we proposed an area-efficient carry select adder by sharing the common Boolean logic term. In the carry select adder, the carry propagation delay can be reduced by M times as compared with the carry ripple adder. However, the duplicated adder in the carry select adder results in larger area and power consumption. In this paper, we proposed an area-efficient carry select adder by sharing the common Boolean logic term. After Boolean simplification, we can remove the duplicated adder cells in the conventional carry select adder. Alternatively, we generate duplicate carry-out and sum signal in each single bit adder cell. By utilizing the multiplexer to select the correct output according to its previous carry-out signal, we can still preserve the original characteristics of the parallel architecture in the conventional carry select adder.

II. MOTIVATION

As the scale of integration keeps growing, more and more sophisticated signal processing systems are being implemented on a VLSI chip.Thesesignal processing applications not only demand great computation capacity but also consume considerable amount of energy. While performance and Area remain to be the two major design tolls, power consumption has become a critical concern in today's VLSI system design. The need for low-power VLSI system arises from two main forces. First, with the steady growth of operating frequency and processing capacity per chip, large currents have to be delivered and the heat due to large power consumption must be removed by proper cooling techniques. Second, battery life in portable electronic devices is limited. Low power design directly leads to prolonged operation time in these portable devices.

III. POWER OPTIMIZATION

Power refers to number of Joules dissipated over a certain amount of time whereas energy is the measure of the total number of Joules dissipated by a circuit. In digital CMOS design, the well-known power-delay

product is commonly used to assess the merits of designs. In a sense, this can be shown as power × delay = (energy/delay) × delay = energy, which implies delay is irrelevant.

THE ADDERS

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 unit, the binary 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 binary adder structures have been done, the studies based on their comparative performance analysis are only a few.

In this project, qualitative evaluations of the classified binary adder architectures are given. Among the huge member of the adders we wrote VHDL (Hardware Description Language) code for Ripple-carry, Carry-select and Carry-look ahead to emphasize the common performance properties belong to their classes. In the following section, we give a brief description of the studied adder architectures. With respect to asymptotic delay time and area complexity, the binary adder architectures can be categorized into four primary classes as given. The given results in the table are the highest exponent term of the exact formulas, very complex for the high bit lengths of the operands.

The first class consists of the very slow ripple-carry adder with the smallest area. In the second class, the carry-skip, carry-select adders with multiple levels have small area requirements and shortened computation times. From the third class, the carry-look ahead adder and from the fourth class, the parallel prefix adder represents the fastest addition schemes with the largest area complexities.

RIPPLE CARRY ADDER

Half Adders can be used to add two one bit binary numbers. It is also possible to create a logical circuit using multiple full adders to add N-bit binary numbers. Each full adder inputs a Cin, which is the Cout of the previous adder. This kind of adder is a Ripple Carry Adder, since each carry bit "ripples" to the next full adder. The first (and only the first)full adder may be replaced by a half adder. The block diagram of 4-bit Ripple Carry Adder is shown here below.

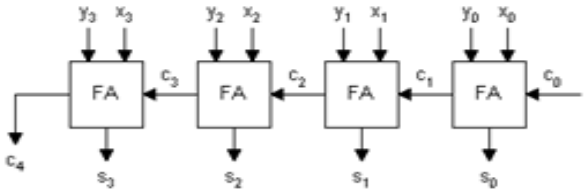


Figure 1: Block diagram of ripple carry adder

The layout of ripple carry adder is simple, which allows for fast design time; however, the ripple carry adder is relatively slow, since each full adder must wait for the carry bit to be calculated from the previous full adder. The gate delay can easily be calculated by inspection of the full adder circuit. Each full adder requires three levels of logic. In a 32-bit [ripple carry] adder, there are 32 full adders, so the critical path (worst case) delay is 31 * 2(for carry propagation) + 3(for sum) = 65 gate delays.

CONVENTIONAL 32-BIT CARRY SELECT ADDER

In general the complete conv CSLA is divided into different blocks. Block size and the number of blocks depends upon the size of conv CSLA according to the SQRT technique. From second block onwards, each block contains three different levels, first level is ripple carry adder with input carry zero, second level is ripple carry adder with input carry one and the third level is multiplexer which is used to select one of the ripple carry adders output.

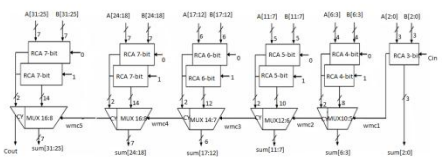


Figure 2: structure of conventional 32-bit carry select adder (CSLA)

The disadvantage in conv CSLA is more area requirement as it uses two levels of RCAs. For achieving better area efficiency Binary to Excess-1 Converter (BEC) is replaced in the place of RCA with C =1 in the modified CSLA. To replace n bit RCA an n+1 bit BEC In is required. Second block of 32-bit mod CSLA with BEC logic is shown in fig.2. One input of third level multiplexers is the output of first level RCA and another input is BEC output. This produces the two possible partial results in parallel and the multiplexer is used to select either the BEC output or the direct inputs according to the control signal Cin.

IV. MODIFIED 32-BIT CARRY SELECT ADDER

A Modified Carry Select-Adder (MCSLA) design is proposed, which make use of single RCA and Binary to Excess-1 Converter (BEC) instead of using dual RCAs to reduce area and power consumption with small speed penalty. As the base of proposed design is that the number of logic gates used in BEC is less than that of RCA. Thus BEC replaces the RCA with Cin=1 instead of using dual RCAs to reduce area and power consumption of the conventional CSA. The importance of BEC logic comes from the large silicon area reduction when designing MCSA for large number of bits. To elaborate this, the gate calculations are made for 4-bit BEC and 4-bit RCA area as under. For 4-bit RCA In 4-bit RCA, four FAs are connected in a chain. Therefore the gates require to build 4-bit RCA are shown in Table 1&2.

Table1: AND, OR and INV gates in 4-bit RCA (MCSLA).

And	9
Or	3
Inverter	7

Table 2: AND, OR and INV gates in 4-bit RCA (CSLA)

And	28
Or	16
Inverter	16

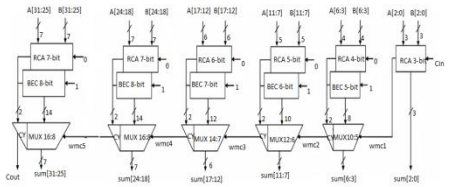


Figure 3: The structure of modified 32-bit carry select adder(MCSLA)

SIMULATION RESULTS

The various adders are designed using Verilog language in Xilinx ISE Navigator 13.2 .And all the simulations are performed using Xilinx I Sim simulator. For each word size of the adder, the same value changed dump (VCD) file is generated for all possible input conditions and imported the same to Xilinx ISE 13.2 Power Analysis to perform the power simulations. The similar design flow is followed for both the regular and modified CSLA..

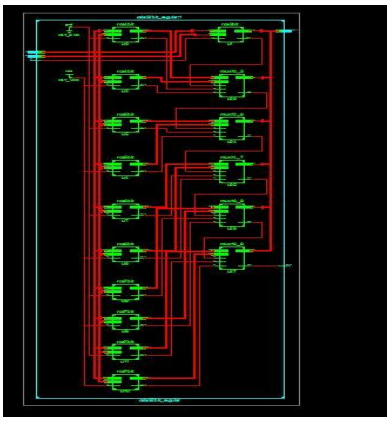


Figure4: 32-bit conv CSLA internal block diagram

```
Timing Summary:
---------------
Speed Grade: -5

   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: 41.238ns

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

=========================================================================
Timing constraint: Default path analysis
  Total number of paths / destination ports: 1153 / 33
-------------------------------------------------------------------------
Delay:               41.238ns (Levels of Logic = 34)
  Source:            b<0> (PAD)
  Destination:       s<31> (PAD)

  Data Path: b<0> to s<31>

                                Gate     Net
    Cell:in->out      fanout    Delay    Delay  Logical Name (Net
```

Figure 5: delay chart of 32-bit conv CSLA.

SQRT_RCA_CSLA_32 Project Status (04/26/2016 - 20:22:58)			
Project File:	cisa_rca.xise	**Parser Errors:**	No Errors
Module Name:	SQRT_RCA_CSLA_32	**Implementation State:**	Placed and Routed
Target Device:	xc3s1600e-5fg320	**• Errors:**	No Errors
Product Version:	ISE 12.2	**• Warnings:**	No Warnings
Design Goal:	Balanced	**• Routing Results:**	All Signals Completely Routed
Design Strategy:	Xilinx Default (unlocked)	**• Timing Constraints:**	
Environment:	System Settings	**• Final Timing Score:**	0 (Timing Report)

Device Utilization Summary				
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of 4 input LUTs	64	29,504	1%	
Number of occupied Slices	48	14,752	1%	
Number of Slices containing only related logic	48	48	100%	
Number of Slices containing unrelated logic	0	48	0%	
Total Number of 4 input LUTs	64	29,504	1%	
Number of bonded IOBs	98	250	39%	

Figure 6: Area chart of 32-bit conv CSLA

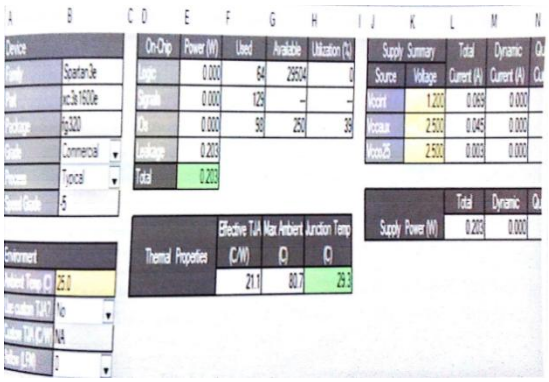


Figure 7: Power chart of 32-bit conv CSLA

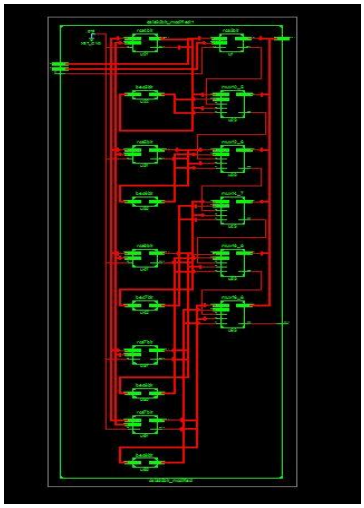
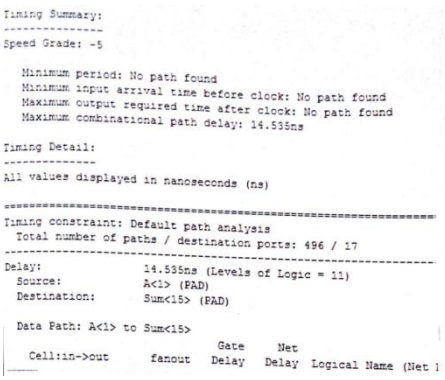
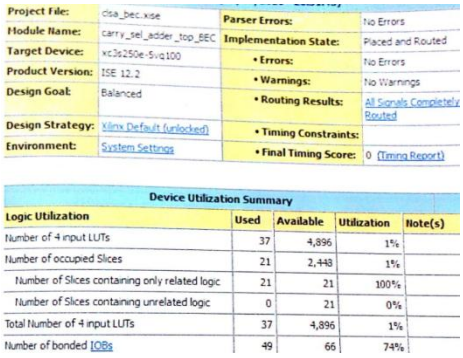


Figure 8: 32-bit MCSLA internal block diagram



```
Timing Summary:
---------------
Speed Grade: -5

   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: 14.535ns

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

=========================================================================
Timing constraint: Default path analysis
  Total number of paths / destination ports: 496 / 17
-------------------------------------------------------------------------
Delay:               14.535ns (Levels of Logic = 11)
  Source:            A<1> (PAD)
  Destination:       Sum<15> (PAD)

  Data Path: A<1> to Sum<15>
                                Gate     Net
    Cell:in->out      fanout   Delay    Delay  Logical Name (Net 1
```

Figure 9: Delay chart of 32-bit MCSLA



Project File:	csa_bec.xise	**Parser Errors:**	No Errors
Module Name:	carry_sel_adder_top_BEC	**Implementation State:**	Placed and Routed
Target Device:	xc3s250e-5vq100	**• Errors:**	No Errors
Product Version:	ISE 12.2	**• Warnings:**	No Warnings
Design Goal:	Balanced	**• Routing Results:**	All Signals Completely Routed
Design Strategy:	Xilinx Default (unlocked)	**• Timing Constraints:**	
Environment:	System Settings	**• Final Timing Score:**	0 (Timing Report)

Device Utilization Summary

Logic Utilization	Used	Available	Utilization	Note(s)
Number of 4 input LUTs	37	4,896	1%	
Number of occupied Slices	21	2,448	1%	
Number of Slices containing only related logic	21	21	100%	
Number of Slices containing unrelated logic	0	21	0%	
Total Number of 4 input LUTs	37	4,896	1%	
Number of bonded IOBs	49	66	74%	

Figure 10: Area chart of 32-bit MCSLA

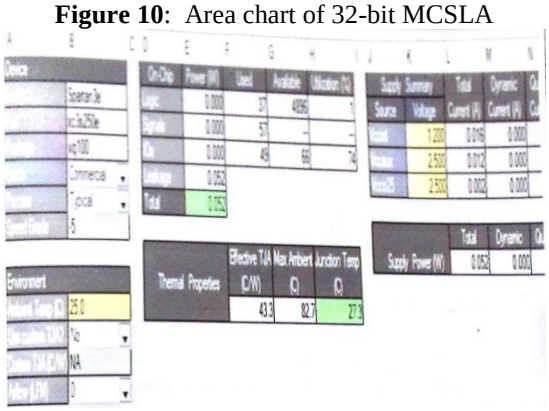


Figure 11: Power chart of 32-bit MCSLA

Table 3: comparison of area, power and Delay of conventional and MCSLA

Technique	Delay(ns)	Area(Lut)	Power(W)
CSLA	41.2	48	0203
MCSLA	21.180	21	0.052

V. CONCLUSION

A simple approach is proposed in this paper to reduce the area and power of conventional CSLA architecture. The reduced number of gates of this work offers the great advantage in the reduction of area and also the total power. The compared results show that the modified CSLA has a slightly larger delay,but the area and power of the 32-bit modified CSLA are significantly reduced . The power-delay product and also the area-delay product of the proposed design show a decrease for 32-bit sizes which indicates the success of the method and not a mere tradeoff of delay for power and area.

The modified CSLA architecture is therefore, low area, low power, simple and efficient for VLSI hardware implementation.

REFERENCES

[1]. AkhileshTyagi, "A Reduced Area Scheme for Carry-Select Adders", IEEE International Conference on Computer design, pp.255-258, Sept 1990.

[2]. KuldeepRawat, TarekDarwish. andMagdyBayoumi, "A low power and reduced area Carry Select Adder", 45th Midwest Symposium on Circuits and Systems, vol.1, pp. 467-470,March 2002.

[3]. O. J. Bedrij, "Carry-Select Adder", IRE transactions on Electronics Computers, vol.EC-11, pp. 340-346, June1962.

[4]. Youngjoon Kim and Lee-Sup Kim, "64-bit carry-select adder with reduced area", Electronics Letters, vol.37, issue 10, pp.614-615, May 2001.

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

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

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

[8]. T.-Y. Chang and M.-J.Hsiao,"Carry-Select Adder using single Ripple-Carry Adder", Electronics letters, vol.34, pp.2101-2103, October 1998.

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