ISSN- 2394-5125 VOL 06, ISSUE 05, 2019

Using the Verilog language, Design a low-power and area-efficient Carry Select Adder

Chittaluri Bhaskar[1], Kota Nageswara Rao[2]
Assistant Professor[1,2]
Department Of ECE
MREC

Abstract- In carry propagation adder design, the carry select approach has been found to be a suitable compromise between cost and performance. However, due to the dual ripple carry adder construction, the conventional carry select adder (CSLA) still consumes a lot of space. The high area overhead of a typical carry select adder (CSLA) makes it undesirable, however this can be avoided by using an add-one circuit. This paper proposes a space-efficient modified CSLA approach based on a new first-zero detection logic. The gate count in a 32-bit modified CSLA can be drastically decreased, and the design suggested in this study was created in the VERILOG language and synthesised in the XILINX13.2 version.

Index Terms- CSLA, RCA, area-efficient, low power, propagation delay.

• Introduction

With the rapid advancement of the electronics industry, the need for low-power devices has skyrocketed. As we use more portable gadgets, we require a battery that can power the devices for an extended period of time. To achieve these goals, we must give a large-capacity battery, but this will require a major increase in battery size, which is not achievable in many cases? Another option is to minimise the device's power usage, which will reduce our net power consumption and, in turn, our carbon footprint in the environment.

Many cascaded single-bit full-adders make up the carry-ripple adder. The circuit architecture is straightforward and space-saving. However, because each full-adder may only start operation once the previous carry-out signal is ready, the calculation pace is slow. The N bits adder is divided into M sections in the carry select adder. Each portion of the adder is made up of two carry ripple adders, cin 0 and cin 1. We can select the right output result based on the logic state of the carry-in signal using the multiplexer. Because the current adder stage does not have to wait for the preceding stage's carry-out signal, the carry-select adder can calculate faster. Because the summation result is ready before the carry-in signal comes, all we have to do is wait to acquire the right computation result.

The carry choose adder, on the other hand, has a duplicated adder, which results in a larger area and higher power consumption. By sharing the same Boolean logic term, we developed an area-efficient carry choose adder in this paper. When compared to the carry ripple adder, the carry select adder can cut the carry propagation delay by M times. The carry choose adder, on the other hand, has a duplicated adder, which results in a larger area and higher power consumption. We suggested an area-efficient carry choose adder in this paper by sharing a common Boolean logic term. The duplicated adder cells in the typical carry choose adder can be removed after Boolean simplification. Alternatively, in each single bit adder cell, we generate duplicate carry-out and sum signals. The multiplexer is used to pick the channels.

ISSN- 2394-5125 VOL 06, ISSUE 05, 2019

Literature Survey

On microprocessors, digital signal processors, and especially digital computers, addition is the most common and widely utilised arithmetic operation. It also acts as a foundation for the creation of all subsequent mathematical operations. As a result, the binary adder structures become a fundamental hardware unit for the efficient implementation of an arithmetic unit.

Anyone looking through a book on computer arithmetic will see that there are a plethora of various circuit architectures with varying performance characteristics that are frequently employed in practise. Although many studies have been done on binary adder structures, only a few studies have been done on their comparative performance analyses.

The qualitative evaluations of the classified binary adder architectures are presented in this research. We wrote VHDL (Hardware Description Language) code for Ripple-carry, Carry-select, and Carry-look ahead among the massive members of the adders to underline the shared performance attributes belonging to their classes. A brief explanation of the examined adder architectures is given in the next section. The binary adder architectures can be classified into four basic types based on asymptotic delay time and area complexity. The table shows the maximum exponent term of the exact formulas, which is highly complicated due to the high bit lengths of the operands.

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 Coutof 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.

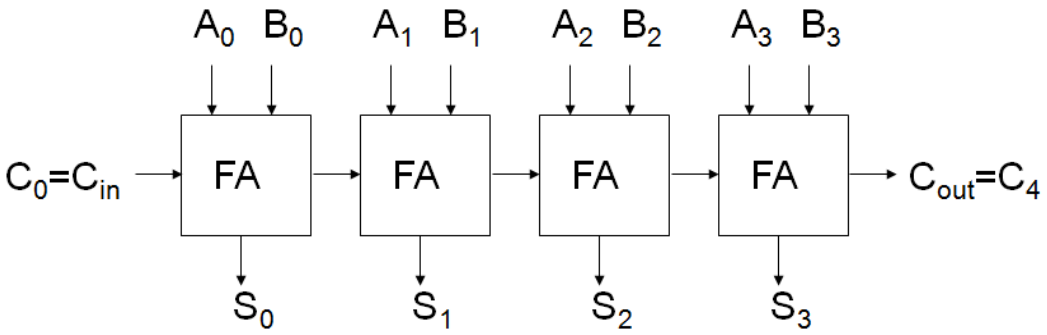


Fig.1.1 Ripple Carry Adder

One of the most serious drawbacks of this adder is that the delay increases linearly with the bit length.The worst-case delay of the RCA is when a carry signal transition ripples through all stages of adder chain from the least significant bit to the most significant bit, which is approximated by:

$$T = (n-1)\,tc + ts$$

ISSN- 2394-5125 VOL 06, ISSUE 05, 2019

Delay :

The latency of a 4-bit ripple carry adder can be derived by considering the worst-case signal propagation path. We can thus write the following expressions:

TRCA-4bit = TFA(A0,B0→Co)+T FA (C in→C1)+TFA (Cin→C2)+ TFA (Cin→S3)

And, it is easy to extend to k-bit RCA:

TRCA-4bit = TFA(A0,B0→Co)+(K-2)* TFA (Cin→Ci)+ TFA (Cin→Sk-1).

Drawbacks :

Delay increases linearly with the bit length and Not very efficient when large bit numbers are used.

Carry Look-Ahead Adder

Lookahead carry algorithm speed up the operation to perform addition, because in this algorithm carry for the next stages is calculated in advance based on input signals. In CLA, the carry propagation time is reduced to O(log2(Wd)) by using a tree like circuit to compute the carry rapidly. Fig.2.3 shows the 4-bit Carry Look-Ahead Adder.

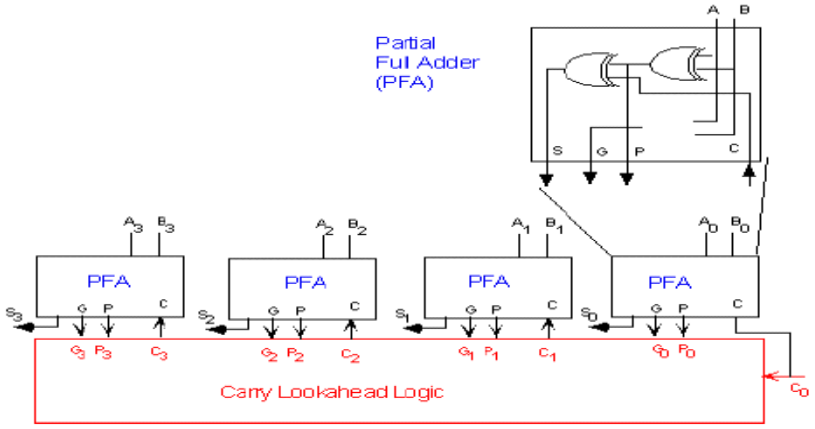


Fig.1.2.4-bit Carry Look Ahead Adder

The CLA exploits the fact that the carry generated by a bit-position depends on the three inputs to that position. If 'X' and 'Y'are two inputs then if X=Y=1, a carry is generated independently of the carry from the previous bit position and if X=Y= 0, no carry is generated. Similarly if X ≠ Y, a carry is generated if and only if the previous bit-position generates a carry. 'C' is initial carry, "S" and "Cout" are output sum and carry respectively, then Boolean expression for calculating next carry and addition is:

Pi = Xi xor Yi -- Carry Propagation
Gi = Xi and Yi -- Carry Generation
Ci + 1 = Gi or (Pi and Ci) -- Next Carry
Si = Xi xor Yi xor Ci -- Sum Generation

Thus, for 4-bit adder, we can extend the carry, as shown below:

C1 = G0 + P0 · C0
C2 = G1 + P1 · C1 = G1 + P1 · G0 + P1 · P0 · C0
C3 = G2 + P2 · G1 + P2 · P1 · G0 + P2 · P1 · P0 · C0
C4 = G3 + P3 · G2 + P3 · P2 · G1 + P3 · P2 · P1 · G0+ P3 · P2 · P1 · P0 · C0

ISSN- 2394-5125 VOL 06, ISSUE 05, 2019

As with many design problems in digital logic, we can make tradeoffs between area and performance (delay). In the case of adders, we can create faster (but larger) designs than the RCA. The Carry Look ahead Adder (CLA) is one of these designs (there are others too, but we will only look at the CLA).

Drawbacks :

For long bit length, a carry look-ahead adder is not practical, but a hierarchical structure one can improve much. The disadvantage of CLA is that the carry logic block gets very complicated for more than 4-bits. For that reason, CLAs are usual implemented as 4-bit modules and are used in a hierarchical structure to realize adders that have multiples of 4-bits.

Proposed Work

32-bit carry select adder

The entire conv CSLA is separated by distinct blocks in general. According to the SQRT approach, block size and number are determined by the size of conv CSLA. From the second block onwards, each block has three levels: the first is a ripple carry adder with input carry zero, the second is a ripple carry adder with input carrier one, and the third is a multiplexer that selects one of the ripple carry adders outputs.

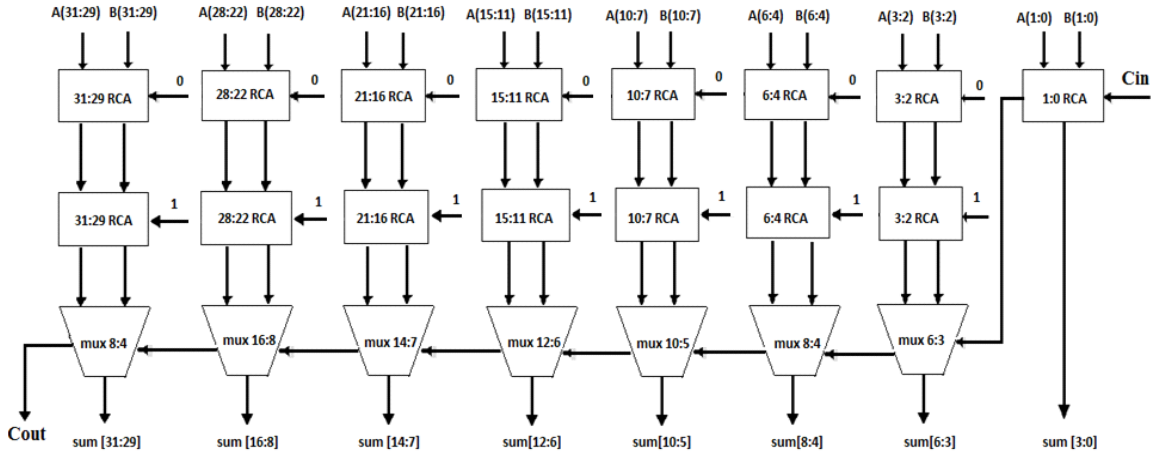


Fig.3.1.Internal Structure of 32-bit select adder

The downside of conventional CSLA is that it requires more space because it employs two levels of RCAs. In the improved CSLA, the Binary to Excess-1 Converter (BEC) is used instead of the RCA to improve space efficiency. An n+1 bit BEC In is required to replace a n bit RCA. Figure 2 shows the second block of the 32-bit mod CSLA with BEC logic. The output of the first level RCA is one input of the third level multiplexers, while the BEC output is the other. The two possible partial outcomes are produced in simultaneously, and the multiplexer is utilised to select the BEC output or the direct inputs based on the control signal Cin.

Modified 32-bit Carry Select Adder

Instead of using dual RCAs, a Modified Carry Select-Adder (MCSLA) design is proposed, which uses a single RCA and a Binary to Excess-1 Converter (BEC) to reduce area and power

ISSN- 2394-5125 VOL 06, ISSUE 05, 2019

consumption with a small speed penalty. The proposed design is based on the fact that the number of logic gates used in BEC is less than in RCA. To minimise the space and power consumption of the standard CSA, BEC replaces the RCA with Cin=1 instead of using two RCAs. When constructing MCSA for a large number of bits, the importance of BEC logic stems from the huge silicon area reduction. To demonstrate, the gate computations for the 4-bit BEC and 4-bit RCA areas are shown below. For RCA 4-bit Four FAs are used in 4-bit RCA.

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

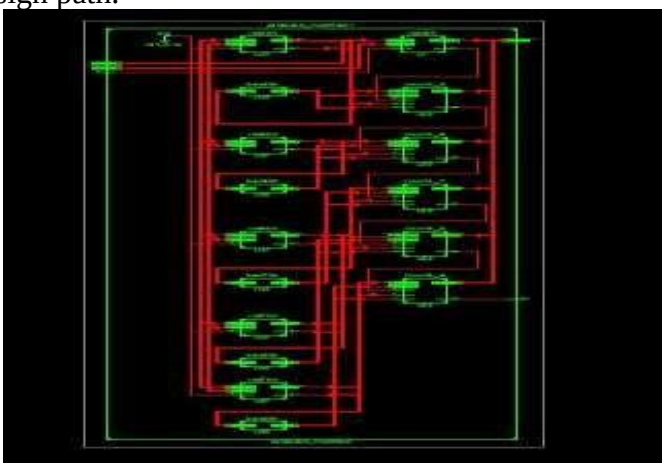
And	9
Or	3
Inverter	7

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

And	28
Or	16
Inverter	16

- **Results and discussion**

In Xilinx ISE Navigator 13.2, the various adders are constructed using the Verilog language. And the Xilinx I Sim simulator is used for all simulations. The same value changed dump (VCD) file is generated for each word size of the adder for all conceivable input conditions and fed into Xilinx ISE 13.2 Power Analysis to execute the power simulations. Both the conventional and modified CSLAs follow a similar design path.



Group	Delay	Area
Group 2	11	57
Group 3	13	87
Group 4	16	117
Group 5	19	147

Fig.4.1.32-bit MCSLA internal block diagram

ISSN- 2394-5125 VOL 06, ISSUE 05, 2019

```
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
```

Fig.4.2.Delay chart of 32-bit MCSLA

Project File:	cla_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%	

Fig.4.3.Area chart of 32-bit MCSLA

CONCLUSION

In this study, a simple solution is proposed to minimise the space and power of traditional CSLA architecture. The reduced number of gates in this study results in a significant reduction in both area and total power. The modified CSLA has a somewhat longer delay, but its area and power are greatly reduced when compared to the 32-bit modified CSLA. 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.

For VLSI hardware implementation, the modified CSLA design is low area, low power, simple, and efficient.

References

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

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

ISSN- 2394-5125 VOL 06, ISSUE 05, 2019

[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 FasterCarry 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", Electronicsletters, vol.34, pp.2101-2103, October 1998.

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