INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH & DEVELOPMENT

ISSN: 2278 – 0211 (Online)

Novel 16-Bit Adder Design For Low Power, Area And Delay

Maroju SaiKumar
Department of Electronics Engineering, Pondicherry University
(A Central University), Pondicherry, India
N.Rajasekhar
Department of Electronics Engineering, Pondicherry University
(A Central University), Pondicherry, India

Abstract:

Adders are one of the widely used digital components in digital integrated circuit design. The Carry Select Adder (CSA) provides a good compromise between cost and performance in carry propagation adder design. However, conventional CSA is still area-consuming due to the dual ripple carry adder (RCA) structure. In this paper, modification is done at gate-level to reduce area and power consumption. The Novel 16-Bit Adder is designed 16-bit and then compared with conventional CSA respective architectures. Novel Adder shows reduction in Delay, Area and Power consumption in comparison with conventional CSA.

In this paper, by using Novel 16-Bit architecture to reduce Delay, Area and Power. The MCSA is designed by using single RCA and Binary to Excess-1 Converter (BEC) instead of using dual RCAs.

Key words: *Adder, RCA, CSA, Low Power VLSI Design*

1.Introduction

The main objective of the VLSI Design Engineer is to design a circuit with Low Power, Low Delay and Low Area. For this type of design, we need to focus on the low power design methodologies and techniques. In this paper, the Adder is designed such a way that it attained all goals of a VLSI Design Engineer. Adder is a Basic block which is present in ALU and which is utilized for arithmetic and logical operations. In this paper, we are going to see a modified hybrid structure of Adder with inner blocks as Ripple Carry Adder, Carry Select Adder with multiplexers and binary to excess 1 converter. In order to achieve a design with low power, area and delay we need to focus on the principles of digital electronics along with low power design methodologies.

2.Carry Select Adder

Carry Select Adders (CSA) is one of the fastest adders used in many data-processing processors to perform fast arithmetic functions. The carry-select adder partitions the adder into several groups, each of which performs two additions in parallel. Therefore, two copies of ripple-carry adder act as carry evaluation block per select stage. One copy evaluates the carry chain assuming the block carry-in is zero, while the other assumes it to be one. Once the carry signals are finally computed, the correct sum and carry-out signals will be simply selected by a set of multiplexers. The 4-bit adder block is RCA. The Figure 1 shows 16-bit conventional Carry Select adder.

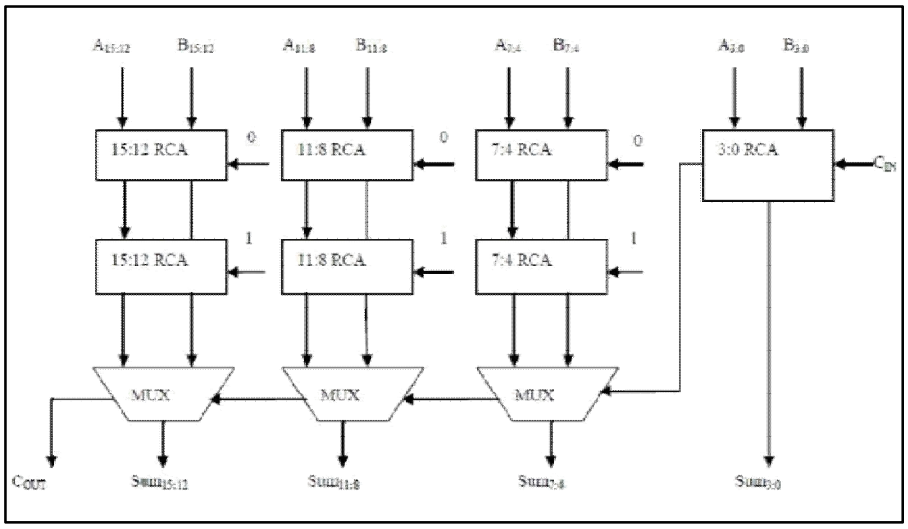

Figure 1 Conventional Carry Select Adder

3.Novel 16-Bit Adder Design

A Novel 16-Bit Adder 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. To replace the N-bit RCA, an N+1 bit BEC is required. The Novel Adder architecture for 16-bit is shown in Figure below. The importance of BEC logic comes from the large silicon area reduction when designing Novel Adder 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.

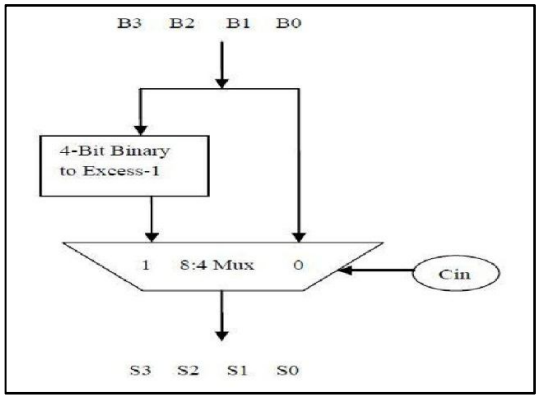

Figure 2: Binary to Excess-1 with Multiplexer

Instead of doing the two RCA adders with carry change we go for the one RCA with carry 0 and another with carry as to be increment to that so, we are designing a Binary to Excess 1 converter. To select among two, we are designing a Multiplexer with (8:4) such that depending on the Carry line the data selected and propagated to next stage. Hence, we need to develop Mux and Binary to Excess 1 Converter in order to achieve High Speed Design.

4. Simulation Results And Comparison

```
--------------------------------------------------------------------------------
Delay:               13.517ns (Levels of Logic = 11)
  Source:            b<1> (PAD)
  Destination:       cout (PAD)

  Data Path: b<1> to cout
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     IBUF:I->O             2   1.228   1.047  b_1_IBUF (b_1_IBUF)
     LUT5:I0->O            3   0.254   0.759  x3/Mmux_out321 (N23)
     LUT5:I3->O            3   0.250   0.759  x4/Mmux_out11 (y<2>)
     LUT3:I1->O            2   0.250   0.617  x7/Mmux_out211 (N6)
     LUT5:I4->O            3   0.254   0.759  x7/Mmux_out411 (N12)
     LUT5:I3->O            3   0.250   0.651  x11/Mmux_out411 (N7)
     LUT5:I4->O            3   0.254   0.759  x11/Mmux_out211 (N13)
     LUT5:I3->O            3   0.250   0.651  x15/Mmux_out311 (N8)
     LUT5:I4->O            2   0.254   0.725  x15/Mmux_out511 (N14)
     LUT3:I1->O            1   0.250   0.579  x15/Mmux_out11 (cout_OBUF)
     OBUF:I->O                 2.715          cout_OBUF (cout)
    ----------------------------------------
    Total                     13.517ns (6.209ns logic, 7.308ns route)
                                       (45.9% logic, 54.1% route)

================================================================================
```

Figure 3: CSLA Delay Calculations

```
--------------------------------------------------------------------------------
Delay:               8.559ns (Levels of Logic = 6)
  Source:            a<10> (PAD)
  Destination:       s<15> (PAD)

  Data Path: a<10> to s<15>
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     IBUF:I->O             2   1.228   1.072  a_10_IBUF (a_10_IBUF)
     LUT6:I0->O            4   0.254   0.792  x7/x3/cout1 (x7/w<2>)
     LUT3:I1->O            3   0.250   0.650  x7/x4/cout1 (y2)
     MUXF7:S->O            1   0.185   0.580  x13/Mmux_y421_f7 (N33)
     LUT2:I1->O            1   0.254   0.579  x13/Mmux_y41 (s_15_OBUF)
     OBUF:I->O                 2.715          s_15_OBUF (s<15>)
    ----------------------------------------
    Total                     8.559ns (4.886ns logic, 3.673ns route)
                                       (57.1% logic, 42.9% route)

================================================================================
```

Figure 4: Novel Adder Delay Calculations

	CSLA	**Novel Adder**
Number of Gates used	512	304
Delay	13.217ns	8.559ns

Table 1: Delay and Area Comparison

Factor	% Reduced
Delay	36%
Area	40.60%

Table 2: Comparison of Delay and Area

DESIGN	POWER
CSLA Adder 16-Bit	0.201mW
Novel Adder 16-Bit	0.173mW

Table 2: Power Comparison

5.Conclusion

In this paper, a Novel Adder is designed by using single Ripple Carry Adders (RCA) and Binary to Excess-1 Converter (BEC) instead of using dual RCAs to reduce Delay, Area and Power. The reduced number of gates of this work offers the great advantage in the reduction of area and also the total power. The Novel Adder architecture for 16-bit is designed and then compared with conventional CSA respective architecture. The Simulation Results as shown above concludes that, by using the Novel 16-Bit Adder, we can achieve the Low Delay, Lesser Area (Number of Gates) and Low Power. These all simulation results are taken for 16 Bit CSLA, Novel Adder and the design simulated by writing Verilog HDL code for design and simulated using Xilinx ISE 12.1.

6.Reference

1. Belle W.Y.Wei and Clark D.Thompson, “Area-Time Optimal Adder Design”, IEEEtransactions on Computers, vol.39, pp. 666-675, May1990.
2. Singh, R.P.P. Kumar, P.; Singh, B., “Performance Analysis of Fast Adders Using VHDL”, Advances in Recent Technologies in Communication and Computing, 2009.
3. ART Com '09. International Conference on, vol., no., pp.189-193, 27-28 Oct. 2009.
4. Richard P. Brent and H. T. Kung, “A Regular Layout for Parallel Adders”, IEEE transactions on Computers, vol.c-31, pp.260-264, March 1982.
5. D. J. Kinniment, “An evaluation of asynchronous addition”, IEEE transaction on very large scale integration (VLSI) systems, vol.4, pp.137-140, March 1996.
6. Akhilesh Tyagi, “A Reduced Area Scheme for Carry-Select Adders”, IEEE
7. International Conference on Computer design, pp.255-258, Sept 1990.