Original Research Paper Engineering

FPGA IMPLEMENTATION OF THE CARRY SELECT ADDER WITHOUT USING MULTIPLEXER

Pabbathi Suvarna M.Tech VLSI Design, Anurag Group of Institutions (Autonomous), Hyderabad, Telangana, India.

M. Murali krishna Assistant Professor, Dept of ECE, Anurag Group of Institutions (Autonoums), Hyderabad, Telangana, India.

ABSTRACT The processor speed of operation is depends on the adder. In digital world the processor operation is main design consideration is high performance adder with less area and small power consumption is important for design a digital adder. The proposed carry select adder is designed for the advanced processors. This paper proposes the fpga implementation high performance carry select adder without using the multiplexer for the final selection of the sum. The approach for design a CSA we implemented by using the high speed parallel prefix adder and then first finding zero logic. By removing the multiplexer in the final stage as a result the area as well as power consumption is reduce. For the selection of the high speed carry generator we use the Kogge stone parallel prefix adder. It will generate the fast carry for intermediate stages of the adder. The adder is designed by using the verilog HDL in VIVADO IDE and implemented on the Zynq Board. The proposed carry select adder is consume less power and occupies less area compared to conventional carry select adder with ripple carry adder.

KEYWORDS : Carry Select Adder, Multiplexer, Kogge Stone, FPGA, RCA, Parallel prefix adder.

INTRODUCTION

Basically Adder is a digital circuit that performs addition of digital bits. In many computers and other kinds of processors adders are used in the arithmetic logic units. They are also utilized in other parts of the processor, where they are used to calculate addresses and similar applications. Although adders can be constructed for many number representations, the most common adders operate on binary numbers. Carry select adder (CSLA) is generally the combination of two ripple carry adders (RCA) and multiplexer. Area of the carry select adder is not efficient because it uses more number of pair of ripple carry adder to produce the sum and carry through supposing Cin='1' and '0' and final output is selected by multiplexer. In CSA, requirement of producing two adders and final selection for multiplexers as a result it require more area, even though carry propagation delay is reduced much.

The Parallel Prefix Adder (PPA) is one of the fastest types of adder that had been created and developed and can also be used to reduce the de-lay. Several examples of such adders have been published and there are many efficient implementations. The kogge-stone adder has an ideal depth of $\log_2 n$.The kogge-stone adder is considered the fastest adder.

There are many carry select adder approaches available but most of them use ripple carry adders to implement the adder. CSA implemented with parallel prefix adder approach can give better delay and area performance.

This paper illustrates the two different approaches of carry select adder implementation to achieve minimum delay and reduced area without increasing the fan-out or lateral wires and covers the comparison of inbuilt CSA and Kogge Stone (KS) adder with multiplexer and without multiplexer.

Parallel prefix Adder:

There is so many approaches are there to implement a carry select adder. The carry select adder is designed using the ripple carry adder is one of the method. But it will take more area and consume more power. To overcome by that problem we need a high performance adder in place of the ripple carry adder. In intermediate stage the speed of the adder depends on the fast carry propagation, the high speed carry generation parallel prefix adder is needed for intermediate stage. The kogge stone adder satisfy the aim of these paper.the kogge stone adder is generates the carry adder and propagates the fast carry for intermediate stages.The kogge stone adder is a high speed for carry propagation adder compared to any other parallel prefix adders and take less area. . Some papers suggested using add one circuit to eliminate the second adder required for the CSA with cin=1 condition. We have shown that Carry select adder

Implemented with parallel prefix adder approach can give better delay and area performance. This paper illustrates the two different approaches of carry select adder design to achieve less delay and small area without increasing the hardware components. we discuss the two different methods used for realizing the carry select adder using Kogge Stone tree, one with multiplexer and the other without multiplexer.

Kogge stone adder:

The carry propagation of KS adder shows the generation of carry for intermediate stages. The carry propagation delay of the kogge stone adder is proportional to $\log_2(n)$ and logic elements used is proportional to $n\log_2(2n)$, where n is the number of bits used in ks adder addition. It is shows that for KS adder, area require is very high, even though it reduces the delay by a large amount.

$C_1 = G_0 + G_0 C_{in}$

$C_2 = (G_1 + P_1 G_0) + P_1 P_0 C_{in}$

$C_3 = (G_2 + P_2 G_1) + P_2 P_1 C_1$

$C_4 = (G_3 + P_3 G_2) + P_3 P_2 (G_1 + P_1 G_0) + P_3 P_2 P_1 P_0 C_{in}$

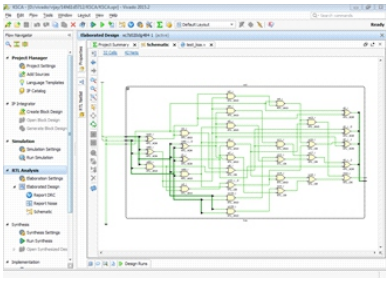
$C_5 = (G_4 + P_4 G_3) + P_4 P_3 (G_2 + P_2 G_1) + P_4 P_3 P_2 P_1 C_1$

$C_6 = (G_5 + P_5 G_4) + P_5 P_4 (G_3 + P_3 G_2) + P_5 P_4 P_3 P_2 C_2$

$C_7 = (G_6 + P_6 G_5) + P_6 P_5 (G_4 + P_4 G_3) + P_6 P_5 P_4 P_3 C_3$

$C_8 = (G_7 + P_7 G_6) + P_7 P_6 (G_5 + P_5 G_4) + P_7 P_6 P_5 P_4 [(G_3 + P_3 G_2) + P_3 P_2 (G_1 + P_1 G_0)] + P_7 P_6 P_5 P_4 P_3 P_2 P_1 P_0 C_{in}$

Figure 1: 4-bit Kogge stone Adder



Carry select adder with ks adder and with mux for final selection:

In the first method adder is implemented by considering cin=0 KS adder first and then cin=1 adder is generated by using an Excess1

adder as shown in the Fig. 5. Finally sum is selected by using a MUX.
Carry equations for cin=0 adder are
c1 = g0
c2 = (g1+p1g0)
c3 = (g2+p2g1) + p2p1c1
c4 = (g3+p3g2) + p3p2 (g1+p1g0)

As shown in the figure it considerably reduces the logic elements required for implementing the second adder.

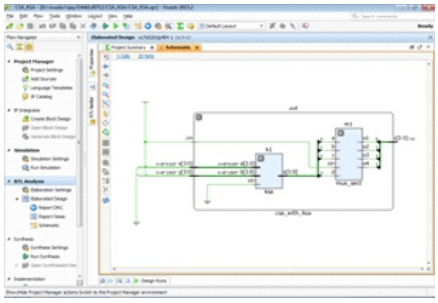


Figure 2:csa with ks adder and final selection for Mux

Proposed Carry select adder without multiplexer:
Here instead of using simple Excess 1adder, a first zero finding logic is used. If cin=0, logic will select the KS adder output as the final output. If cin=1, logic will invert the KS adder output from LSB to MSB until it finds the first occurrence of a zero. As the carry input is propagating through the AND network, the resulting delay will be slightly greater compared to CSA with MUX. But the proposed adder uses lesser logic elements compared to CSA with MUX.

In order to reduce the delay of carry propagation through multiplexers; a fast carry network is implemented using Kogge Stone tree. So as shown in the Fig. 7 we can achieve the same delay as that of Kogge Stone adder with this modification. For the second adder MUX is not required as it directly realizes the result using the carry input. Certain product terms required for the fast carry logic is generated in the CSA adder. Every 4 bit generates its corresponding product terms and is fed to the fast carry logic.

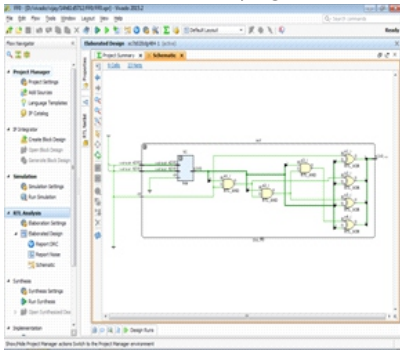
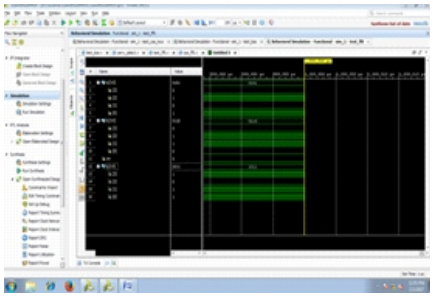


Figure 3: proposed carry select adder without using multiplexers.

RESULTS:
Figure 4: Simulation result of proposed carry select adder without using multiplexers.



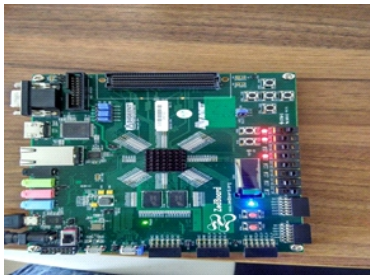


Figure 5: zynq board implementation result of CSA using without multiplexers

Comparison among the carry select adder with multiplexer and without multiplexer:

	CSA with Mux	CSA without MUX
Power(mW)	6.79	5.107
Delay(nS)	9.41	9.045

CONCLUSIONS
In this paper we have shown the design of carry select adder without multiplexers for final selection of sums implemented with Kogge Stone tree using two different approaches. One approach uses Excess1 adder and the other uses first zero finding logic to realize the carry select adder. Both the adders are implemented on ZYNQ Board device and the performance is compared. CSA with MUX performs better in terms of delay and CSA without MUX performs better in terms of area. Due to this area and power is also reduced. CSA without MUX performs slightly better compared to CSA with MUX when the area-delay product is taken. FPGA inbuilt adder is also used for comparison. This is to show the closeness of the performance of the optimized adders with FPGA Adder.

REFERENCES:

[1] O. J. Bedrij, "Carry-select adder," IRE Trans. Electron. Computer., pp.340–344, 1962.
[2] J. Sklansky, "Conditional-Sum Addition Logic" IRE. Transactions on Electronic Computers, vol. EC-9, pp. 226-231, 1960.
[3] P.M. Kogge, H.S. Stone; "A Parallel Algorithm for the Efficient Solution of a General Class of Recurrence Equations" IEEE Trans., C- 22(8):786-793, Aug. 73.
[4] R.E. Ladner, M.J. Fischer; "Parallel Prefix Computation" JACM, 27(4):831-838, Oct. 80.
[5] R.P. Brent, H.T. Kung; "A Regular Layout for Parallel Adders" IEEE Trans., C-31(3):260-264, March 82.
[6] T. Han, D.A. Carlson; "Fast Area-Efficient VLSI Adders" 8th IEEE Symp. Computer Arithmetic, Como Italy, pp. 49-56, May 87.
[7] J. Kowalczuk, S. Tudor, D. Mlynek; "A New Architecture for Automatic Generation of Fast Pipelined Adders" ESSCIRC, Milano Italy, pp. 101- 104, Sept 91.
[8] A. Beaumont-Smith, N. Burgess; "A GaAs 32-bit Adder" 13th Symp. Computer Arithmetic, hilomar California, pp. 10-17, June 97.
[9] S. Knowles, "A family of adders," Proceedings of the 14th IEEE Symposium on Computer Arithmetic, April 14-16 1999, adelaide, Australia.
[10] Feng Liu et.al "A Comparative Study of Parallel Prefix Adders in FPGA Implementation of EAC" Proceedings of the 12thEuromicro conference on digital system design 2009