

A Novel Carry Select Adder: Topology and Performance Evaluation

¹Romana Yousuf and ²Najeeb-ud-din

¹*Electronics and Communication Engineering Department, IUST Awantipora (J&K) India*

²*Electronics and Communication Engineering Department, NIT Srinagar (J&K) India*

¹Corresponding author: rozi13@rediffmail.com

Abstract— With the technology scaling to deep sub-micron, the speed of a circuit increases rapidly. At the same time, the power density per chip also increases significantly due to the increasing density of the chip. The thermal consideration becomes a major challenge in the VLSI circuit design, which in turn puts constraints on the speed improvement of circuit. In processors, most commonly used arithmetic operation is the addition operation which is being performed by adder. Thus, it is the adder delay that determines the maximum frequency of operation of the chip. Different topologies have been put forward, each providing trade-off between delays and other characteristics such as area and energy dissipation and as such no design is considered as superior. Carry select adder is considered to be best in terms of speed but to a lesser extent at the cost of its area.

The goal of this work is to synthesize and optimize the delay and area of carry select adder in 65 nm technology using FPGA's. This work proposes a new high-speed, low area design methodology for carry select adder. Further this proposed logic has been optimized in order to reduce the delay by making use of universal NAND gates. For the introduction of Asynchronism, a customized circuit has been developed to generate various handshaking signals. Experimental results show that the worst case delay of 128-bit proposed-optimized carry select adder is 7.739 ns based on 65 nm technology.

Index Terms— Asynchronism, Carry Select Adder, Delay, FPGA

I. INTRODUCTION

DUE to rapidly growing system-on-chip industry, not only the faster units but also smaller area and less power has become a major concern for designing VLSI circuits, i.e. demand for high speed is continuously increasing. Therefore, in realizing modern VLSI circuits, low power, less area and high-speed are the predominant factors, which need to be considered in design flow.

Arithmetic and logical unit (ALU) in computers perform the arithmetic and logical functions. An arithmetic unit (processor) is a VLSI system that performs operation on digital arithmetic (DA) numbers: on fixed point and floating point numbers. The performance parameters that a designer has to keep in mind while designing the arithmetic unit are the precision, the time required to execute each operation, the cost of processor, area occupied and its energy consumption. Depending on the operation, which includes addition, subtraction, division, comparison, square root, change of sign etc., an arithmetic processor operates on one, two or more operands. Among all these operations, addition is considered as the most elementary arithmetic operation. Implementation of other operations such as subtraction, multiplication, division, and address computation can be done with the help of adders only. Thus adder becomes a speed-limiting block and it is the adder delay that defines the delay of the chip. It is a high-speed adder can increase the overall performance of computers, although speed is not the only criterion in judging an adder performance. Other criteria that are important in determining the performance of any digital system includes power consumption and layout area. An area efficient adder is useful in decreasing the silicon area, and consequently reduces the cost of the chip. Therefore, careful optimization of the adder parameters is of utmost importance. The optimization can proceed either at the logic level or at circuit level. Typical logic optimization involves rearranging the Boolean equations so that a faster circuit is obtained.

There are various types of adders which provide trade-off between delays and other parameters such as layout area and energy dissipation. As such no design is considered as efficient, but they provide alternatives from which to choose in a specific context within a set of specific requirements and constraints. Different techniques have been proposed in order to improve the speed of large bit-width adders, but the objective is always to compute the input carry more quickly for the high order bits. After all it is the carry that determines the speed of the arithmetic unit.

Hybrid adders have been designed and are considered to be better in terms of performance, power consumption, Power-delay Product (PDP), area, and noise margin etc.

Conventional Carry Select Adder was initially proposed by Bedrij [1], in which two ripple carry adder chains were used. Main disadvantage of this methodology was that it consumes more power due to doubling of Si area with respect to the ripple carry adder. To get through this disadvantage, Chang proposed a new topology [2]. An improvement in speed was further made by R. Hashemain, who proposed a new design for high-speed and high density Carry Select Adder [3]. Further reduction in area with negligible speed penalty was made by Y. Kim and L.S. Kim [4]. Another architecture for Carry Select Adder was proposed by B. Amelifard, F. Faklllah and M. Pedram [5], which was based on the concept of sharing of two adders and hence called as carry select adder with sharing (CSAS). Hybrid Carry select Adders were also proposed, where Carry Look-ahead Adder architecture was combined with Carry Select Schemes in order to increase the speed [6]. Dual Transition Skewed Logic (DTSL) can also be used to design a low power and high performance carry select adder [7]. M. Alioto, G. Palumbo and M. Poli proposed a gate-level design of carry select adder [8]. Research on adders has revealed that pipelining is used to increase throughput of the computational unit. Based on this fact, Y. Kim, K. H. Sung and L. S. Kim proposed pipelined carry select adder architecture [9]. Another improvement in Carry Select Adder has been made by introducing reconfigurable CSA with fixed and variable sized blocks [10]. In fixed sized blocks, the operand is partitioned into slices of equal size unlike variable sized blocks where slices are of different size. K. Rawat, T. Darwish and M. Bayoumi proposed a modified carry select adder with reduction in area and power with minimum speed penalty [11].

II. CARRY SELECT ADDER

While designing any digital circuit, the performance parameters that are to be considered are power, delay and area. Carry Select Adder is considered to be the good design choice in terms of high speed and reasonable area. Such adders are considered to be a compromise between Ripple Carry Adders (RCA) and Carry Look-ahead Adder (CLA). Carry Select Adder differs from Ripple Carry Adder in the sense that in a Ripple Carry Adder the carry has to ripple through full-adder blocks. Output of every next full-adder cell is dependent on the carry generated from the previous block of full adder, which in turn increases the delay in the circuit. But in the case of a Carry Select Adder the carry has to pass through a single multiplexer.

The basic principle of Carry Select Adder is that, at each stage we have two adder units. Each unit implements the addition operation in parallel. The first unit implements the addition assuming a carry-in of '0', generating the sum and carry-out bit. The second unit performs the same operation assuming the carry-in of '1' i.e. two sums are computed in parallel, one for 1-carry and other for 0-carry. Finally one among them is selected with the help of multiplexer, provided, the carry of previous stage is available, as shown in figure 1 [12]. Carry-out of a particular stage is generated with the help of AND and OR gates.

The multiplexer in figure 1 gives us the desirable sum as:

$$(C^0, S^0) = ADD(A, B, C_0 = 0) \quad \dots (1)$$

$$(C^1, S^1) = ADD(A, B, C_0 = 1) \quad \dots (2)$$

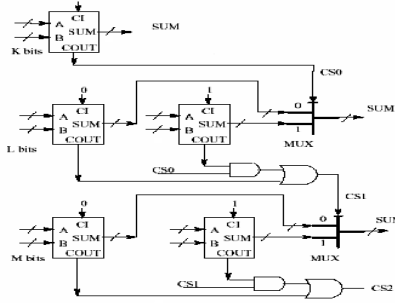


Fig. 1: Carry Select Adder using multiplexer

Thus, when the carry-in of the group is known, one of the above two functions is selected as:

$$(C_o, S) = \begin{cases} (C^0, S^0) & \text{if } C_0 = 0 \\ (C^1, S^1) & \text{if } C_0 = 1 \end{cases} \quad \dots (3)$$

Various research groups have proposed different methodologies for the implementation of carry select adders but a new methodology proposed here has improved the speed and reduced the area of CSA to a larger extent.

III. PROPOSED TOPOLOGY

Our aim here was to synthesize a carry select adder, which is optimized for delay, power and area, and hence increases the performance of the adder. A new logic was proposed for the Carry Select Adder that reduces the delay in comparison with the existing Carry Select Adder. This methodology shows better result than that of the already existing ones. It is based on the logic that sum for carry-in of '0' is compliment of the sum for carry-in of '1' i.e. $(A \oplus B)'$. Based on this, the logic of the circuit shown in the figure 2 has been developed. Sum for carry-in of '0' is given as:

$$Sum^0 = (A \oplus B) \quad \dots (4)$$

And for carry-in of '1' we have: multiplexers

$$Sum^1 = (A \oplus B)' \quad \dots (5)$$

Both Sum^0 and Sum^1 are fed to the multiplexer whose strobe is the previous carry. The multiplexer

will give us the desired sum. Carry-out is obtained from multiplexer whose inputs are 'A' and previous carry 'C_{in}' with a strobe of Sum.

The above logic has been developed by using the following equations

$$Sum^0 = (A \oplus B) \quad \dots (6)$$

$$Sum^1 = (Sum^0)' \quad \dots (7)$$

$$C_{out} = A \text{ when } Sum^0 = 0$$

$$= C_{in} \text{ when } Sum^0 = 1 \quad \dots (8)$$

$$Sum = Sum^0 \text{ when } C_{in} = 0$$

$$= Sum^1 \text{ when } C_{in} = 1 \quad \dots (9)$$

Because of the basic gate latencies there will be the mismatch of arrival time between the carry select signal and the sum signal to the multiplexer, as is evident from the figure 2. This mismatch has been eliminated by making use of concept of blocks. For the efficient design of any adder, block size should be designed to optimally match the signal arrival time at the multiplexer input to the delay time of the carry-in signal.

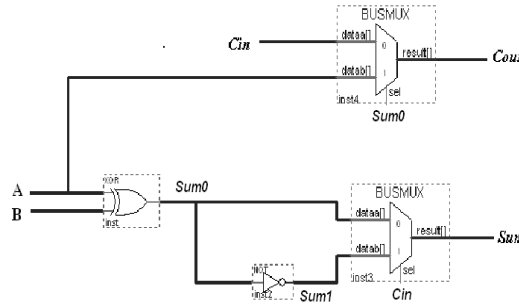


Fig. 2: Proposed Basic Carry Select Adder Cell

One of the ways to reduce the delay and hence power dissipation of CSA is to make use of the universal NAND gate. Therefore, changing the circuit of figure 2 to NAND equivalent circuit has reduced the delay by a considerable amount. Since NAND gates are in the category of the universal gates and these gates offer small delay as compared to that of the other gates. Thus the use of NAND gates further reduces the delay of the proposed logic to a larger extent. The circuit diagram of the optimized logic is shown in figure 3.

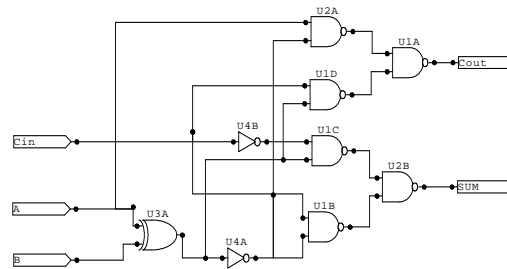


Fig. 3: Optimized Carry Select Adder Cell

Clocking is the most challenging task in the successful design of the synchronous circuits. Clock not only consumes a large portion of the chip's power budget, but also is continually active and is connected virtually to every part of the chip. This means power is being consumed by logic whether it is processing data or not. Another problem associated with synchronous circuits is the noise. Since, here clock synchronizes all activity on the chip, keeping most of the signal transitions in the lock step. This creates the absolute worst environment from suppressing noise. One more problem associated with the synchronous circuits is latency penalty. With asynchronous circuits, the total time required to complete logic operations is roughly equal to the sum of the time required for each activity. On the other hand, in a synchronous system the total time is the time for the slowest operation, since each discrete activity is tied to all others through the global clock. To overcome the disadvantages of synchronous circuits, asynchronous circuits are used. Asynchronous logic is essentially any circuit style that uses a sequencing mechanism other than centralized global clock. Sequencing the system without clock still requires a strict ordering of events as they happen within the system, but logic function activation is based on the availability of valid data from previous stages, rather than clock signal. Therefore, following the RISC philosophy of "making the common case fast," asynchronous datapaths have the potential to outperform synchronous designs on average inputs.

The logic diagram shows the internal structure of the CVin to CVout conversion. It features four 74VHC00 inverters (U3D, U7C, U7A, U4A) and three 74VHC04 inverters (U7B, U7C, U7A). The inputs are CVin, START, and Strobe. The output is CVout. The circuit implements a 3-to-1 multiplexer logic where CVin is selected when START is high, and the Strobe signal is selected when START is low.

Fig. 5: Generation of **start signal**

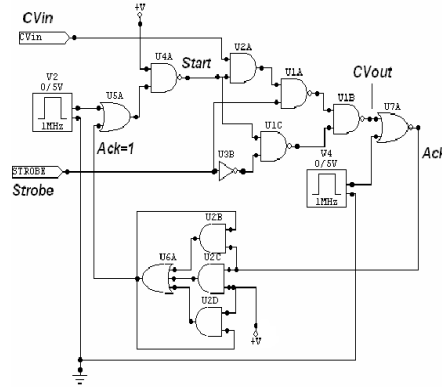


Fig. 6: Asynchronous logic

V. SIMULATION RESULTS

Here results of the existing, proposed and proposed optimized Carry Select Adders are discussed. Results of proposed logic are then compared with the results of the existing logic. Since it is not possible to compare two adder topologies, unless their technology node is same. So it became essential, to implement the adders in the same technology node for comparison. Here the adders were implemented in Xilinx simulator, having 65 nm technology. Different graphs have been developed from the simulation results. These graphs are plotted between the Delay and number of bits for different bit lengths. Figure 7 shows the graphical result of the delay in the existing carry select adder. Figure 8 shows delay in proposed Carry Select Adder with block size of fixed length.

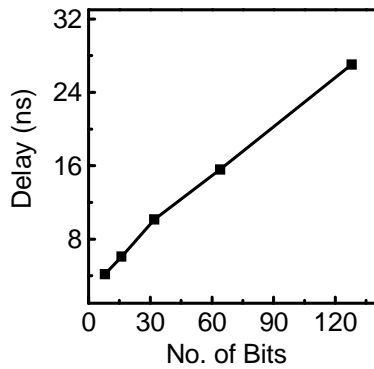


Fig. 7: Delay in Existing Carry Select Adder

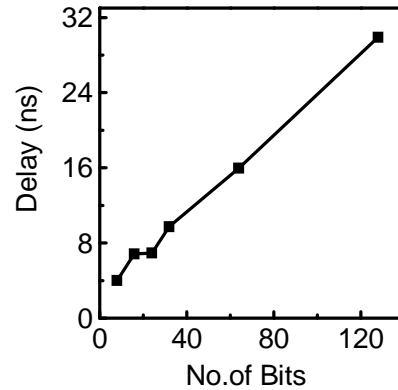


Fig. 8: Delay in proposed Carry Select Adder with Fixed Size Blocks

However, there was a slight improvement of this topology over the existing one. Further improvement in the proposed carry select adder was introduced by replacing every logic element in a basic cell by NAND gates. Results of such configuration are shown in Figure 9. Here blocks of fixed size were taken into consideration and the results for the variable block size are shown in figure 10.

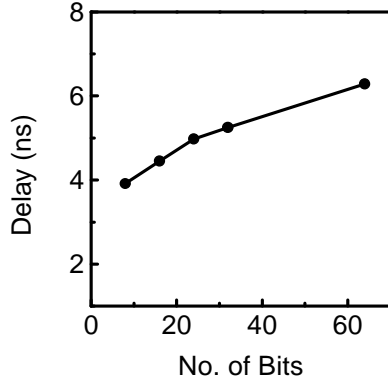


Fig. 9: Delay in Proposed-Optimized CSA with fixed size blocks

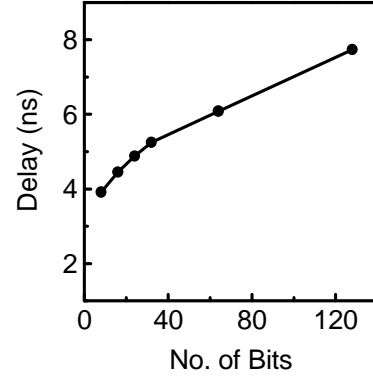


Fig. 10: Delay in Proposed-Optimized CSA with Fixed Block Size

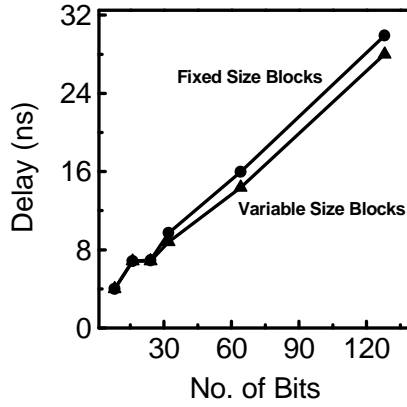


Fig. 11: Comparison of delays in Proposed logic for Fixed and variable size Blocks

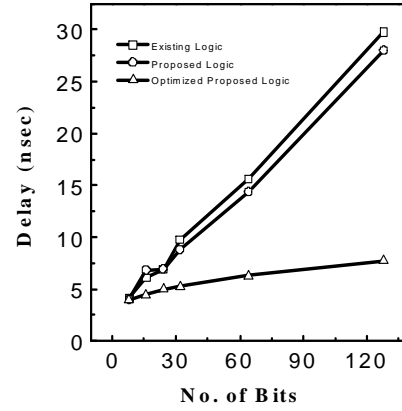


Fig. 12: Comparison of delays in Existing, Proposed and Proposed-Optimized CSA

Comparison of delays in proposed-optimized logic for fixed and variable size blocks is shown in figure 11. From the plot it is evident that, for lower bits, the delay is almost same for fixed and variable size blocks.

This implies that for lower bits the use of fixed blocks or variable blocks does not make any difference in performance. However for higher bits the delay is less for CSA based on variable sized blocks, although the difference being very small. Figure 12 shows the performance of three different topologies i.e existing logic, proposed logic and the optimized proposed logic of CSA in terms of their delay parameter. From the plot of figure 12 it is evident that, for lower bits i.e. approximately up to 24-bits the delay of the existing and the proposed logic almost coincides with each other. But for higher bits the delay of the proposed logic is lesser than that of the existing logic. Further from graph of figure 12, we can see that the delay of the optimized proposed logic is very small as compared to the other two. Thus from this graph we can conclude that proposed topology reduces the delay of the CSA and by optimization of this proposed logic we can increase the speed of the CSA to a larger extent than that of the existing logic.

VI. CONCLUSION

This paper has presented and verified a new topology for CSA. In this methodology, sum and carry-out was calculated by making use of XOR gate, an inverter and a multiplexer. Optimization of the proposed logic was made by replacing each logic element of the proposed logic with the universal NAND gates. Simulations have been performed for various configurations for different bits. Concept of blocks of fixed and variable size has also been introduced and the result has been analyzed for different bit pattern. An improvement in the proposed logic has reduced the delay to a larger extent. A customized circuit has been developed and designed for the introduction of Asynchronism, to generate various handshaking signals. The functionality of the circuit has been verified in a circuit simulator. Experimental results show that the worse case delay of 128-bit proposed-optimized carry select adder is 7.739 ns as compared to 27.033 ns in case of the existing topology based on 65 nm technology. This reveals that this proposed logic is better than the existing carry select logic. This result was also verified analytically by making use of the method of Logical Effort.

REFERENCES

- [1] J. O. Bedrij, "Carry Select Adder," *IRE Trans. Electronic Computers*, vol.11, pp. 340-346, 1962.
- [2] T.Y.Chang and M.J.Hsiao, "Carry Select Adder using single Ripple Carry Adder," *Electronic Letters*, vol. 34, pp. 2101-2103, 1998.
- [3] R. Hashemian, "A new design for high speed and high density carry select Adders." *Proceedings of IEEE Midwest Symposium on Circuits and Systems*, pp.1300-1303, 2000.
- [4] Y.Kim and L.S.Kim, "A low power Carry Select Adder with Reduced Area, *Proceedings of 2001 IEEE Symposium on Circuits and Systems*, vol.4, pp. 218-21, May 2001.
- [5] B. Amelifard, F. Fallah and M. Pedram, "Closing the gap between Carry Select Adder and Ripple Carry Adder: A new class for low power high performance Adders, *6th International Symposium on Quality of Electronic Design*, pp. 148-152, March 2005.
- [6] O. Kwon, E. Swartzlander and K. Nowka, "A Fast hybrid Carry Lookahead Carry Select Adder Design," *Proceedings of the 11th Great Lakes Symposium on VLSI*, pp.149-152, West Lafayette, Indiana, USA, 2001.
- [7] W. Jeon, Kasich Roy, and Cheng. K. Koh, "High performance Low-Power Carry Select Adders using Dual Transistor Skewed Logic, *Proceedings of 2001 IEEE Symposium. on Solid State Circuits*, pp.145-148, 2001.
- [8] M. Alioto, G. Palumbo and M. Poli, "A Gate level strategy to design Carry Select Adders," *Proceedings of 2002 IEEE Symposium on Circuits and Systems*, vol. 2, May 2004, pp. 465- 468.
- [9] Y. Kim, K. H. Sung and L. S. Kim, "A 1.67 GHz 32-bit pipelined Carry Select Adder using the complementary scheme, *Proceedings of 2002 IEEE Symposium on Circuits and Systems*, vol. 1, pp. 461-464, May 2002.
- [10] Jin. F. Li, Yao. C. Kuo, Chao. D. Huang, Tsu. W. Tseng, and Chin. L.Wey, "Design of Reconfigurable Carry Select Adders, *IEEE Asia-Pacific Conference of Circuits and Systems*," pp. 825-828, 2004.
- [11] Kuldeep Rawat, Tarek Darwaish and Magdy Bayoumi, "A Low Power and Reduced Area Carry Select Adder," *Proceedings of 2002 IEEE Symposium on Circuits and Systems*, vol. 1, pp.467-469, Aug 2002.
- [12] Milos D. Ercegovic and Tomas Lang, "Digital Arithmetic," Morgan Kaufmann.