Modular Design of Adders with Domino Logic 1

M.B.DAMLE 2 DR S.S.LIMAYE

Abstract- Based on 180nm CMOS technology a 4 bit domino logic adder is designed for speed optimization over ripple carry adder. The adder is designed using 4 bit slice of carry look-ahead adder. Multiple slices are may be connected in ripple carry fashion to obtain higher order adders like 8, 16, 32 and others. This result in considerable reduction in time as compared to nominal ripple carry adder. Equations of sum, generate and propagate are implemented in domino CMOS logic using TSMC 180nm library to provide energy optimization. A 64 bit adder designed using 16 slices of Carry look-ahead adder gives latency of no more than time equivalent to 33 clocks with a transistor count of 1504. Average power results are also presented in this paper with selected input vectors. Average power is 4.65 microwatt

Keywords: Domino logic, generator, propagator, 4-bit slice.

I. INTRODUCTION

Addition is one of the fundamental arithmetic operations. It is used extensively in many VLSI systems such as application-specific DSP architectures and microprocessors. In most of these systems, the adder is part of the critical path that determines the overall performance of the system. Along with improved performance of microprocessors, it is essential to improve the performance of the adder. Recently, a lot of adder structures were proposed by many other authors [1], [2]. Among various binary adder architectures, a large family of prefix adders is particularly attractive because it is easily expressed and suggests very efficient implementations, i.e. Adders based on this formulation can be attractively fast and compact when implemented in VLSI [3]. Among those factors affecting the performance, the fan-out and wire length are often the deterministic factors. As VLSI technology moves into the deep submicron domain, the wiring problem becomes even more significant and, in many cases, dominates in both area and delay optimization [4], [5].

This paper describes how higher order multiplier can be implemented using slices of lower order multiplier and as an example a 64 bit adder implemented using slices of 4 bits. The 4 bit slice is a carry look-ahead adder implemented using CMOS domino logic with TSMC 180 nm technology. Section II describes the equations used to implement the 4 bit slice and their circuit implementation in TannelTools. Section III focuses on the carry propagation chain analysis for the 64 bit adder and its design using 4 bit slices. The delay analysis is also presented in this section. Section IV shows the simulation results for the adder architecture and its blocks and conclusions are presented in the last section (V).

II. 4 BIT CARRY LOOK AHEAD ADDER WITH DOMINO CMOS LOGIC

The 4 bit slice is implemented using a carry look-ahead adder structure that contains the following blocks. The implementation of these blocks is also presented here and all are built using CMOS domino logic structures.

A) Sum circuit that calculates

Sum(i) for i = 0 -63 with the given equation

Adder Equations

sum(i) = a(i) xor b(i) xor cin(i) [Eq.1]

Fig. 1 shows the schematic of the SUM circuit. The SUM circuit is composed of two XOR gates. The XOR gate is modified from the cross-coupled version by replacing the NMOS portion with a clock gated NMOS. In this circuit, the PMOS transistors receive the input signal *A*, *B*, and *Cin*. The operation of this circuit can be divided into two phases: the IDLE PHASE and the EVALUATING PHASE. In the IDLE PHASE, the clock signal *CLK* is 'logic 1', and the output signal *SUM* will be 'logic 0'.

In the EVALUATING PHASE, the clock signal *CLK* is 'logic 0', and the corresponding output signal *SUM* will be evaluated according to the input signals *A*, *B*, and *Cin*.

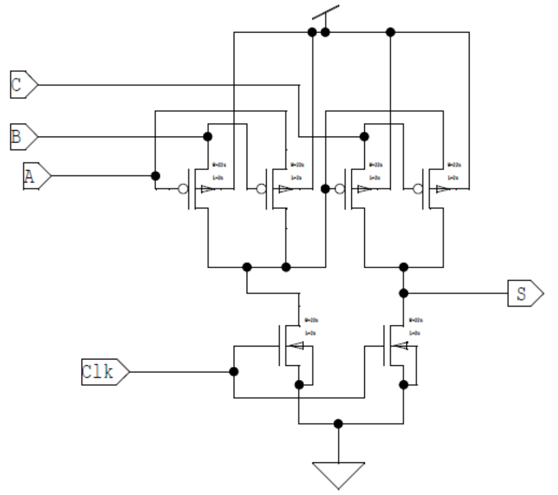

Fig. 1 Sum circuit

B) The carry out is defined as:

$c(i+1) = (cin(i) (a(i) \text{ or } b(i))) \text{ or } (a(i) b(i))$ [Eq.2]

For carry look ahead adder the generate and propagate equations [7] are

Generate $g(0) = a(0) b(0)$ [Eq.3]

Propagate $p(0) = a(0) + b(0)$ [Eq.4]

So the carry for the first block is given by:

$c(1) = g(0) + p(0)c(0)$ [Eq.5]

and in general:

$c(i+1) = g(i) + p(i)c(i)$ [Eq.6]

Hence for a 4 bit slice of carry look-ahead adder the carry block is implemented as:

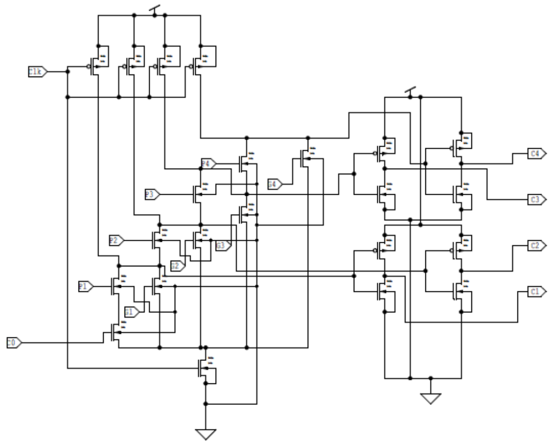

Fig.2 Carry look ahead circuit

Where the generators G1, G2, G3, G4 and propagators P1, P2, P3, P4 are implemented using the following logic equations and circuits.

C) Propagation and generation of carry across multiple bits provides low latency and hence, Propagate/Generate across multiple bits equations [7]

$P(0,1) = p(0) p(1)$ [Eq.7]

$G(0,1) = g(1) \text{ or } p(1)g(0)$ [Eq.8]

In general, any j with i<j, j+1 < k

$c(k+1) = G(i,k) + P(i,k)c(i)$ [Eq.9]

$G(i,k) = G(j+1,k) + P(j+1),G(i,j)$ [Eq.10]

$P(i,k) = P(i,j) P(j+1,k)$ [Eq.11]

So, in a nutshell the equations for propagate and generate are given by:

Carry Look-ahead Generate Logic

$g0 = a0 b0$ [Eq.12]

$g1 = a1 b1 + g0 (a1 \text{ or } b1)$ [Eq.13]

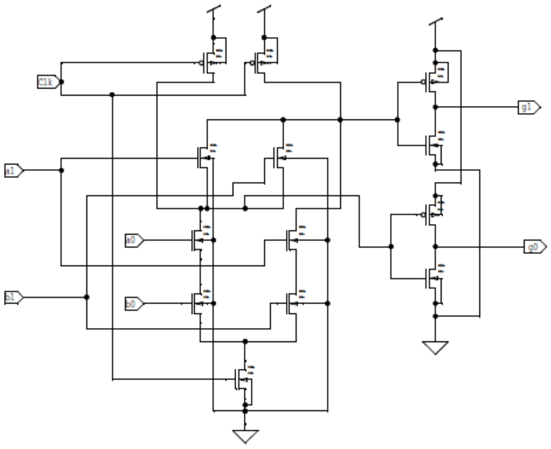

Fig.3 Generator block

Carry Look-ahead Propagate Logic

$p0 = a0 + b0$ [Eq.14]

$p(0,1) = (a1+b1) (a0 + b0)$ [Eq.15]

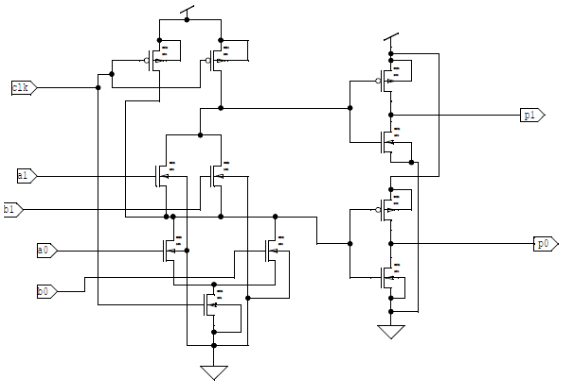

Fig.4 Propagator Block

D) Finally, a 4 bit slice is created by using the above designed modules which is capable of computing a 4 bit addition.

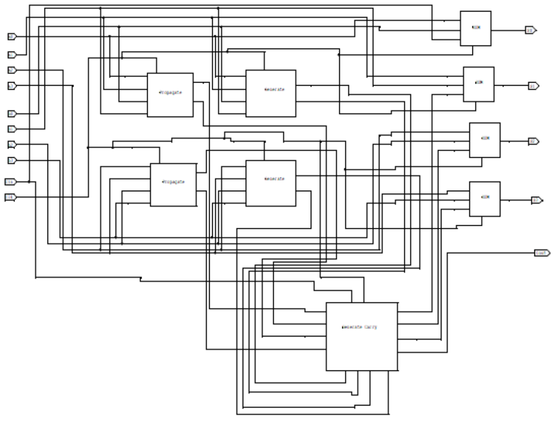

Fig.5 4-bit carry look-ahead adder

This 4 bit block takes inputs of two 4 bit numbers, a carry-in and produces a 4 bit sum and carry-out. This block can be further instantiated multiple times in order to produce 8, 16, 32, 64 bit or any higher order adder circuit. In this paper we have implemented a 64 bit adder using 16 of the above slices as presented in next section.

III. 64 BIT ADDERS CARRY CHAIN ANALYSIS

Using the 4 bit carry look-ahead adder described in previous section, we have implemented a 64 bit adder structure with ripple carry propagation between the carry look-ahead adders [6].

The comparison of the designed 64 bit adder with respect to the worst case delay is presented henceforth.

For adding two n-bit binary numbers there are a total of 2n inputs
Using r-input logic gates, according to
Spira's bound $t \geq \lceil \log r\ 2n \rceil$
For a ripple carry adder
The worst case delay for Cn is 2n+1 and the worst case delay for Sn-1 is 2n
Hence for a 64 bit ripple carry adder, n=64 and the worst case delay for Cn=129 and for Sn-1=128.

In this paper the carry propagates only between the blocks of the carry look-ahead adders and hence there are only 16 propagations. So for the designed adder,

Delay for Cn=33 and for Sn-1 delay=32.

This is 4 times faster than the worst case delay for a ripple carry adder. Following is the 64 bit adder implementation using 4 bit slices of carry look-ahead adder.

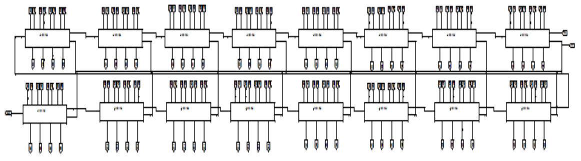

Fig. 6 64 bit domino logic adder

IV SIMULATION AND ANALYSIS

The simulation is carried out in two parts, first each of the blocks in section II are simulated by applying appropriate vectors at the input and observing the outputs for the Boolean equations for each block.

A) Sum circuit simulation

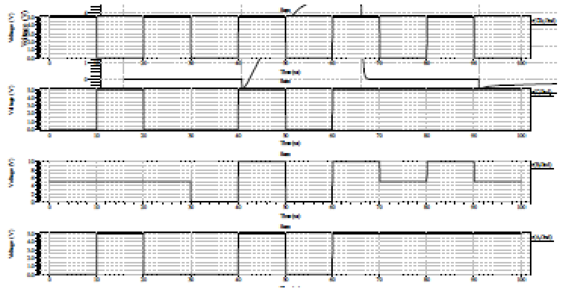

Fig.7 T-Spice simulation result for Sum

Clock input provides the synchronous operation and also is a part of the domino logic. In the above simulation inputs to sum circuit in fig.1 are provided and the equation Eq.1 is verified.

B) Carry-out simulation shown below verifies Eq.6 when sample inputs are applied to circuit in Fig.2.

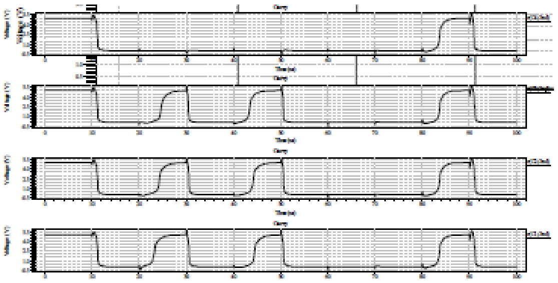

Fig.8 T-Spice simulation result for Carry look-ahead adder block

C) Simulation graphs for generate and propagate blocks are show below to verify equations Eq.12,13 and Eq.14,15 respectively.

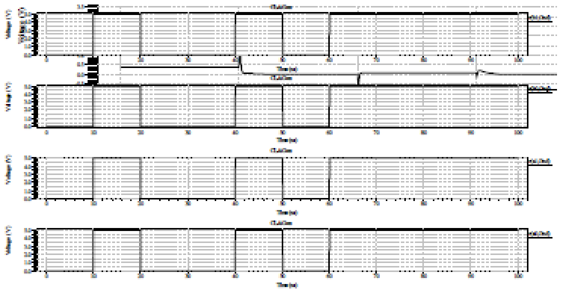

Fig.9 T-Spice simulation result for generator block

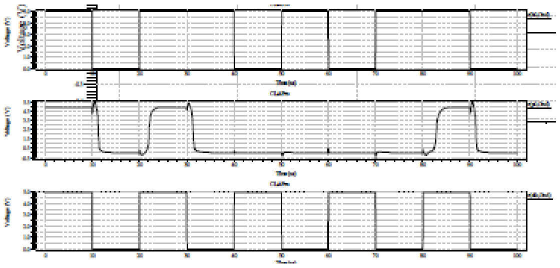

Fig.10 T-Spice simulation result for propagator block

D) The simulation result for 64 bit ripple carry adder designed in previous section is presented below for a given sample input vector

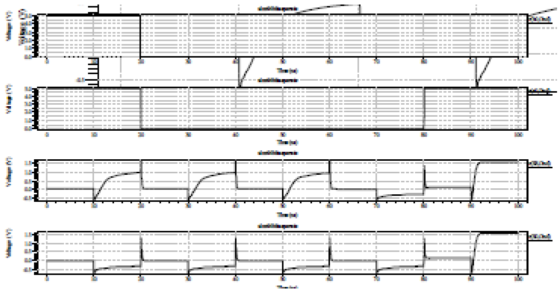

Fig.11 T-Spice simulation result for 64 bit adder

In the second part of the simulation, average power for the design is calculated over the transient analysis. The simulation report from the TannerTool T-spice gives the following result after complete simulation:
Power:
avgpower = 4.6534e-006 Watts
Transistor Count:
MOSFETs - 1504 MOSFET geometries – 2

Implemented in Tanner-13, for low power microwatts and high speed in GB **V. CONCLUSION**

In this paper, a higher order (64 bit) adder is realized using multiple carry look-ahead adders designed in CMOS domino logic. From the analysis, it is concluded that a balance between the performance, power and area can be achieved by designing the lower order adder (bit slice, 4 bit in our case) and then cascading multiple of these to form higher order adder. A direct implementation of 64 bit carry look-ahead adder would result in high transistor count accounting for larger chip area and also high power. Therefore, there are no adders that can perform the best for all applications and designers can choose the adder that satisfy their need and requirements and the design presented in this papers provides the flexibility to achieve requirements as per design.

REFERENCES

[1] R. Zimmermann and W. Fichtner, “Low-power logic styles: CMOS versus pass-transistor logic”, *IEEE J. Solid-State Circuits*, vol. 32, no. 7, pp. 1079-1090, July 1997.

[2] M. Vesterbacka, “A 14-transistor CMOS full adder with full voltage -swing nodes”, in *Proc. IEEE Workshop Signal Processing Systems*, Taipei, Taiwan, Oct. 20–22, 1999, pp. 713-722.

[3] H. T. Bui, Y. Wang, and Y. Jiang, “Design and analysis of low-power 10-transistor full adders using novel XOR-XNOR gates”, *IEEE Trans. Circuits Syst. II*, vol. 49, no. 1, pp. 25-30, Jan. 2002.

[4] C. H. Chang, J. Gu, and M. Zhang, “A review of 0.18-μm full adder performances for tree structured arithmetic circuits”, *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 13, no. 6, pp. 686-695, June 2005.

[5] J. F. Lin, M. H. Sheu, and C. C. Ho, “A novel high-speed and energy efficient 10-transistor full adder design”, *IEEE Trans. Circuits and Syst. I*, vol. 54, no. 5, pp. 1050-1059, May 2007.

[6] Gin Yee, Carl Sechen. “ Clock delayed domino for adder and combinational logic design”.

[7] Lecture notes on VLSI systems by Prof. Robert Reese, ECE dept. Mississippi State university.

[8] R.P.P. Singh, Parveen Kumar and Balwinder Singh, “Performance Analysis of Fast Adders Using VHDL”, IEEE International Conference on Advances in Recent Technologies in Communication and Computing, 189-193,2009.

[9] B.Ramkumar and Harish M.Kittur, “Low Power and Area Efficient Carry Select Adder”, IEEE Transactions on Very Large Scale Integration (VLSI) systems, accepted for publication DOI:10.1109/ TVLSI. 2010.2101621.

Authors :

M.B.Damle, Associate Professor, RCOEM, Nagpur, M.S. from BITS PILANI RAJASTHAN, SMIEEE, FIETE,CE (IE),

Dr.S.S.Limaye, Ph.D. Electronics, Principal, Zhulelal C.O.E. Nagpur. SMIEEE, FIETE,CE (IE),