Indian J.Sci.Res. 17(2): 99-104, 2018

ISSN: 0976-2876 (Print)
ISSN: 2250-0138(Online)

AN INVESTIGATION ON COMPARATIVE ANALYSIS OF KSA USING CMOS AND GDI DESIGN

[1]Syed Kareemsaheb, [2]MVV SatyanarayanaChowdary, [3]S. Aisha
[1]Department of Electronics and Communication Engineering, ANU College of Engineering & Technology, Guntur
[2]Department of Electronics and Communication Engineering ,Teegala Krishna Reddy Engg. College, Meerpet, Medbowli
[3]Department of Electronics and Communication Engineering NallaMalla Reddy Engg. College, JNTUH, Hyderabad

Abstract:Adders forms a major part in various arithmetic logical operations. Parallel Prefix Adder have been built up as the most essential and efficient circuit for binary addition. Their Particular structure and execution performance are very attractive for VLSI implementation. . Further to reduce power consumption, instead of adders, 3:2, 4:3, 5:3, 6:3 and 7:3 compressors are used for the addition of partial products in multipliers. Further to reduce power consumption in the adders, compressors and multipliers using a new technique called Gate Diffusion Input (GDI) instead of Complementary Metal Oxide Semiconductor (CMOS). In these papers, we describe the design and performance of the Kogge Stone Parallel Prefix Adders and implemented using different design technique. CMOS (Complementary Metal Oxide Semiconductor) and GDI (Gate Diffusion Input) are the different design technique used. The design and simulation of logic gates is performed on CADENCE Design Suit 6.1.6 using virtuoso and ADE Environment at GPDK 180nm technology. The execution measurement considered for the performance of the KSA is delay, number of gate count/Transistor Count (area) and power. Simulation studies are done for 4-bit, 8-bit and 16-bit input data

Keywords: Gate Diffusion Input (GDI), Complementary Metal Oxide Semiconductor (CMOS), Digital Signal Processing (DSP).

I.Introduction

With the advances in Very Large Scale Integration (VLSI) technology, arithmetic operations are penetrating into more and more applications. The basic operation found in most arithmetic components is the binary addition and Multiplication. Computations needs to be performed using low-power, area-efficient circuits operating at greater speed. Addition is the most basic arithmetic operation; and adder is the most fundamental arithmetic component of the processor. In addition, each of the resulting output bits are depending on its corresponding inputs. It is very important operation because it involves a carry ripple step i.e the carry from the previous bits addition should propagates to next bits of addition.

Multiplication is an operation that occurs frequently in digital signal processing and many other applications.[1] Multipliers occupy more area so that it consumes large delay when compared to Adders. Therefore several techniques are being proposed to speed up the computation while maintaining the less or reasonable area. A multiplier can be divided into three stages: The first stage is Partial products generation stage, second is partial products addition stage, and the final is addition stage. In the first stage, the multiplier and the multiplicand are multiplied bit by bit to generate the partial products. The second stage is more complicated and it determines the speed of the overall multiplier. In this stage the partial products generated by the previous stage is added by using Adders or compressors depending on the technique used for designing multipliers.

The present development in processor designs aim is design of low power multiplier. So, the need for low power multipliers has increased. Generally the computational performance of DSP processors is affected by its multipliers performance. This results in circuit delay because to perform the addition of next bits, it should wait until the completion of addition from the previous bits. So that it propagates carry to the next stage Designers of VLSI have several options to reduce the power dissipation in the various design stages. Recently, the requirement of portability and the moderate improvement in battery performance indicate that the power dissipation is one of the most critical design parameters. The three most widely accepted metrics to measure the quality of a circuit or to compare various circuit styles are area, delay and power still demands high computational speeds. The architecture of Adders like Ripple Carry Adder, Carry Look Ahead Adder, Kogge-Stone Adder and Brent-Kung Adders have advantages with respect to power, area and complexity. The architecture of Multipliers like Braun and Wallace Tree Multipliers are efficient and easy to design when compared to other multipliers. Further to reduce power consumption, instead of adders, 3:2, 4:3, 5:3, 6:3 and 7:3

[1]**Corresponding Author**

compressors are used for the addition of partial products in multipliers. Further to reduce power consumption in the adders, compressors and multipliers, a new technique called Gate Diffusion Input (GDI) is used instead of Complementary Metal Oxide Semiconductor (CMOS).

Gate diffusion input (GDI) — A new technique of low-power digital combinational circuit design. This technique allows reducing power consumption, propagation delay, and area of digital circuits while maintaining low complexity of logic design. Pass-transistor logic has been presented for NMOS. They are based on the model, where a set of control signals is applied to the gates of NMOS transistors. Another set of data signals are applied to the sources of the n-transistors. Some of the main advantages of PTL over standard GDI design are

High speed, due to the small node capacitances

- Low power dissipation, as a result of the reduced number of transistors
- Lower interconnection due to a small area.
- However, most of the PTL implementations have two basic problems. They are:
- Since the "high input voltage level at the regenerative inverters is not VDD, the PMOS device inthe inverter is not fully turned off, and hence Direct-path static power dissipation could be significant

A new low-power design technique that allows solving most of the problems mentioned above - GDI technique. The GDI approach allows implementation of a wide range of complex logic functions using only two transistors. This method is suitable for design of fast, low-power circuits, using a reduced number of transistors (as compared to GDI and existing PTL techniques), while improving logic level swing and static power characteristics and allowing simple top-down design by using small cell library.

The GDI method is based on the use of a simple cell as shown in Figure..1 At first glance, the basic cell reminds one of the standard GDI inverter, but there are some important differences.[1]. The GDI cell contains three inputs G(common gate input of NMOS and PMOS), P (input to the source/drain of PMOS), and N (input to the source/drain of NMOS).

Bulks of both NMOS and PMOS are connected to N or P (respectively), so it can be arbitrarily biased in contrast with a GDI inverter.

Bulks of both NMOS and PMOS are connected to N or P (respectively), so it can be arbitrarily biased in contrast with a GDI inverter.

It must be remarked that not all of the functions are possible in standard p-well GDI process but can be successfully implemented in twin-well GDIor silicon on insulator (SOI) technologies. Table 1 shows how a simple change of the input configuration of the simple GDI cell corresponds to very different Boolean functions. Most of these functions are complex (6-12transistors) in GDI, as well as in standard PTL implementations, but very simple(onlytwo transistorsper function) in the GDI designmethod

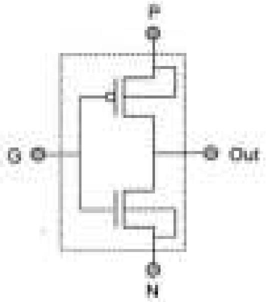


Fig.1: GDI Basic Cell [[1]

Table.1: Various Logic Functions of GDI cell

for different Input configurations[1]

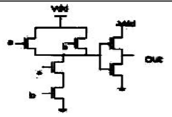
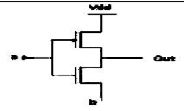
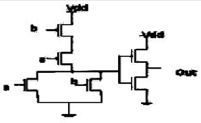
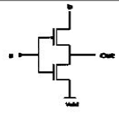
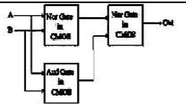
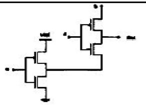
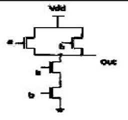
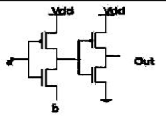
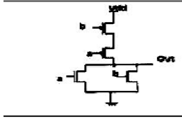
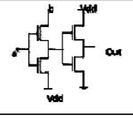
N	P	G	OUT	FUNCTION
0	B	A	A'B	F1
B	1	A	A'+B	F2
1	B	A	A+B	OR
B	0	A	AB	AND
C	B	A	A'B+AC	MUX
0	1	A	A'	NOT

III. CMOSVs GDI Structure

Complementary metal–oxide–semiconductor (CMOS) is a technology for constructing integrated circuits. CMOS technology is used in microcontroller, microprocessor, static RAM and other digital logic circuits. CMOS technology is also used for several analog circuits such as image sensors (CMOS sensors), data converters, and highly integrated transceivers for many types of communication. Two CMOS is also sometimes referred to as complementary-symmetry metal–oxide–semiconductor (or COS-MOS).[1] The words "complementary-symmetry" refer to the fact that the typical digital design style with CMOS uses complementary and symmetrical pairs of p-type and n-type metal oxide field effect transistors (MOSFETs) for logic functions. Important characteristics of CMOS devices are high noise immunity and low static power consumption. CMOS circuits are constructed in such a way that all PMOS transistors should have either an input from the voltage source or from another PMOS transistor. Similarly, all NMOS transistors should have either an input from ground or from another NMOS transistor. The composition of a PMOS transistor provides low resistance between its source and drain contacts when a low gate voltage is applied and high resistance when a high gate voltage is applied. But on the other hand, the composition of an NMOS transistor provides high resistance between source and drain when a low gate voltage is applied and low resistance when a high gate voltage is applied. CMOS achieves current reduction by complementing every NMOSFET with a PMOSFET and connecting both gates and both drains together. [2] A high voltage on the gates will result to the condition that NMOSFET will conduct and the PMOSFET will not conduct while a low voltage on the gates causes the reverse. This technique greatly reduces power consumption and heat production. However, during the switching time both MOSFETs conduct briefly as the gate voltage goes from one state to another. GDI resembles standard CMOS inverter cell with the only difference is that CMOS inverter has only one input and two supply voltages VDD and VSS. But GDI cell can have three inputs G (common gate input of NMOS and PMOS), VDD supply is replaced by P (input to the source/drain of PMOS), VSS supply is replaced by N (input to the source/drain of NMOS).

IV. Implementation

Table 2 implementation of logic gates using GDI technique and CMOS [2]

A.Ripple Carry Adder

This is also known as parallel adder. In this all the bits are given simultaneously. For a 4-bit adder its consists of 4 stages and each stage consists of one full adder. At each stage carry is calculated starting from lower bit. This carry is propagated to the next stage. The main advantage of this RCA is simplicity. The main disadvantages of this RCA are that the carry should be propagated through each stage to get the final output which takes large amount of time. [4]The below figure represents the Logic diagram of the 4-bit Ripple Carry Adder with a two input busses each of 4-bit length namely A<3:0> , B<3:0> and C_{in} as Carry input and sum<3:0> as output bus of 4-bit length and C_{out} as carry out.

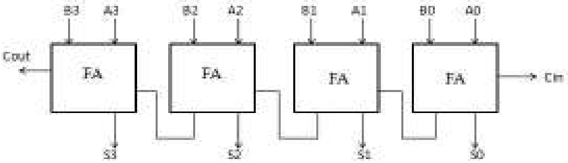


Fig 2: Logic diagram of 4-bit Ripple Carry Adder

By cascading two 4-bit RCA we will obtain a 8-bit RCA Logic diagram of the 8-bit Ripple Carry Adder with a two input busses each of 8-bit length namely A<7:0>, B<7:0> and C_{in} as Carry input and sum<7:0> as output bus of 8-bit length and C_{out} as carry out is shown in the below figure 3

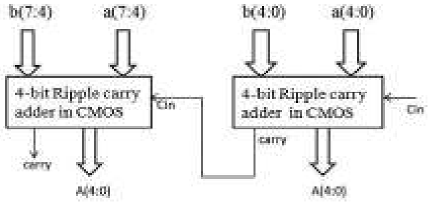

b(7:4)
a(7:4)
b(4:0)
a(4:0)
4-bit Ripple carry adder in CMOS
Cin
4-bit Ripple carry adder in CMOS
Cin
carry
carry
A(4:0)
A(4:0)

Fig 3: Logic diagram of 8-bit RCA

B.Carry Look Ahead Adder (CLA)

It is faster especially in adding large number of bits. A carry-look ahead adder improves speed by reducing the amount of time required to determine carry bits. By using a carry look ahead carry generator can easily construct a 4-bit parallel adder. Each sum requires two exclusive-OR gates. The output of first exclusive-OR generates Pi, and the AND gate generates Gi. The carriers are generated using look ahead carry generator and applied as inputs to the second exclusive-OR gate. Other input to exclusive-OR gates is Pi. Thus second exclusive-OR gate generates sum outputs. Each output each generated after a delay of two levels gates. Thus, outputs S0 through S3 have equal propagation delay times. It can be contrasted with the simpler, but usually slower, ripple carry adder for which the carry bit is calculated alongside the sum bit, and each bit must wait until the previous carry has been calculated to begin calculating its own result and carry bits The carry-look ahead adder calculates one or more carry bits before the sum, which reduces the wait time to calculate the result of the larger value bit. Carry look ahead adder is able to generate carries before the sum is produced using propagate and generate logics to make addition much faster. From the full adder circuit derived two functions carry generator and carry propagate logic.

Generate, Gi = Ai AND Bi

Propagate, Pi = (Ai XOR Bi)

The output sum and carry can be expressed as

Sum, Si = Pi xorCi

Carry, Ci+1 = Gi + PiCi

Gi is called a carry generate and it produces on carry when both Ai and Bi are one, regardless of the input carry. Pi is called a carry propagate because it is term associated with the propagation of the carry from Ci to Ci+1. Now the Boolean functions for the carry output of each stage can be written as follows,

C2=G1+P1C1

C3 = G2+P2C2 = G2+P2 (G1+P1C1) = G2+P2G1+P2P1C1

C4=G3+P3C3

From the above equations C4 does not have to wait for C3 and C2 to propagate, in fact C4 is propagated at the same time as C2 and C3. The Boolean functions for each output carry are expressed in sum of product form, thus they can be implemented using AND-OR logic. By combining multiple carry look ahead adders even larger adders can be implemented. This can be used at multiple levels to make even larger adders

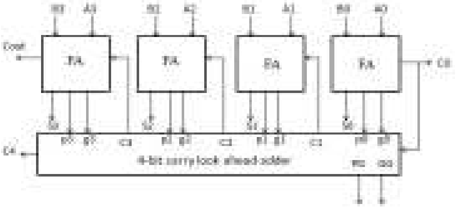


Fig 4: Logic diagram of 4-Bit Carry Look Ahead Adder

Logic diagram of the 8-bit Carry Look Ahead Adder with a two input busses each of 8-bit length namely A<7:0> , B<7:0> and C_{in} as Carry input and sum<7:0> as output bus of 8-bit length and C_{out} as carry out is shown in the figure 5

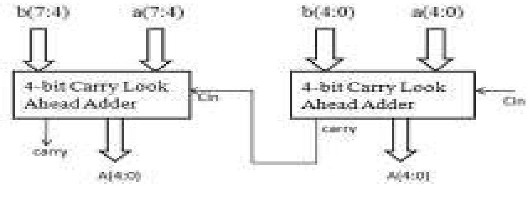


Fig 5: Logic diagram of 8-Bit CLA Adder

V. Implementation Of 8-Bit Parallel Prefix Adders

The operations involved in this figure are given as.

CP0=Pi and Pj

CG0= (Pi and Gj) or Gi

Where Pi,Gi are present bits and Pj,Gj are the previous bits

It is the fastest adder used in industries. The parallel prefix addition is done in 3 steps. They are:

1. Pre-processing stage
2. Carry generation network
3. Post processing stage

In this Pre-processing stage we compute, the generate and propagate signals are used to generate carry input of each adder. A and B are inputs. These signals are given by the equation 1&2. [7]

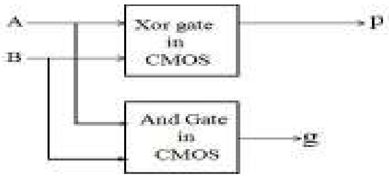


Fig 6: Pre-processing stage

In this Carry generation network stage we compute carries corresponding to each bit. Execution is done in parallel form. After the computation these. carry operator contain two AND gates, one OR gate. It uses propagate and generate as intermediate signals which are given by the equations 3&4.

P(i:k) =P(i:j) . P(j-1:k)

G(i:k) =G(i:j) +(G(j-1:k) . P(i:j))

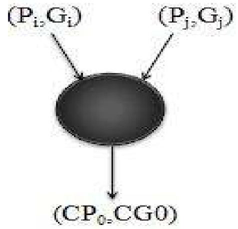


Fig 7: Carry generation network of Kogge-Stone Adder

A. Kogge-Stone Adder

Kogge-Stone prefix adder is a fast adder design. Kogge-Stone adder has best performance in VLSI implementations. It has large area with minimum fan-out and it is also known as a parallel prefix adder that performs fast logical addition. [6]Kogge-Stone adder is used for wide adders because of it shows the less delay among other architectures. Each vertical stage produce Propagate and Generate bits. Generate bits are produced in the last stage and these bits are XORed with the initial propagate after the input to produce the sum bits. Logic diagram of the 8-bit Kogge-Stone Adder with a two input busses each of 8-bit length namely A<7:0> ,

B<7:0> and C_{in} as Carry input and sum<7:0> as output bus of 8-bit length and C_{out} as carry out are shown in the figure 4.7.

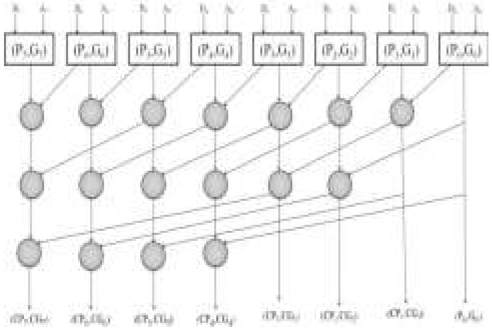


Fig 8: Logic diagram of 8-Bit Kogge-Stone Adder[6]

VI. Braun Multiplier

An nxn-bit Braun Multiplier requires n(n-1) adders and n^2AND gates. This makes Braun multipliers ideal in VLSI and ASIC realization. Each of the X_iY_j product bits is generated in parallel with the AND gates. Each partial compressors and adders are employed such that minimum number of outputs is generated.[10] For example in column 5, there are 7 partial products to be added. These could be added using a 4-3 compressor and a full adder thereby generating five output bits. But instead of this a 7-3 compressor has been used which will generate only.

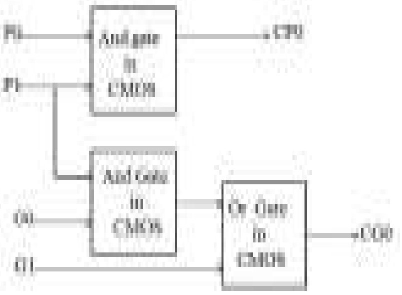


Fig9.Schematic diagram of 8x8 Braun Multiplier

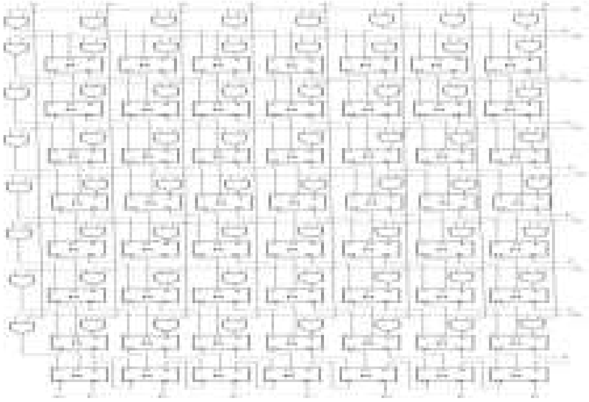
VII. Wallace Tree Multiplier Using Compressors

A. Wallace Tree Multiplier

The partial-sum adders can also be rearranged in a tree like fashion, reducing both the critical path and the number of adder cells needed.[9] The tree multiplier realizes substantial hardware savings for larger multipliers. The propagation delay is reduced as well. In fact, it can be shown that the propagation delay through the tree is equal to O (log3/2 (N)). While substantially faster than the carry-save structure for large multiplier word lengths, the Wallace multiplier has the disadvantage of being vary irregular, which complicates the task of an efficient layout design.

B. Compressors

Different Compressor logic based upon the concept of counter of full adder. Compressor is defined as single bit adder circuit that has more than three inputs as in full adder and less number of outputs.[7] Compressors can efficiently replace the combination of several half adders and full adders, thereby enabling high speed performance of the processor which incorporates the same. The schematics of 4:3, 5:3, 6:3,7:3 compressors are shown in the figures 12,13,14,15 respectively.



VIII. Conclusion And Future Work

In this work, different Adders like RCA, CLA Adder, KSA, BKA and different multipliers like Braun Multiplier and Wallace tree multipliers designed both in CMOS and GDI logics in Tanner tool. The results are compared in terms of power consumption, delay and number of transistors between CMOS logic designs and GDI logic designs. The results showed that when compared to CMOS logic based designs GDI logic based designs consumed 60% less power, 68% less delay and 80% less no. of Transistors. When the results are compared only in GDI logic. They showed that In Adders, CLA adder consumes less power, less delay and RCA consume less number of transistors. In Multipliers, Braun Multiplier consumes less power and Less delay and number of transistors.

In present work, the Adders and Multipliers are designed by using GDI. In future the work goes on designing ALU and MAC units and other applications of DSP by using the implemented Adders and Multipliers.

References

[1] Jashanpreetkaur, NavdeepKaur and Amit Grover presented ―A review on Gate Diffusion Input (GDI)‖ international journal of advance research in electronics, electrical & computer science

applications of engineering & technology volume 2, issue 4, July 2014, pp 385-391.

[2] Madhusudhandangeti presented ―minimization of transistors count and power in an embedded system using gate diffusion input technique‖ uniascit, vol 2 (3), 2012, 308-313.

[3] ArkadiyMorgenshtein, Alexander Fish, and Israel A. Wagner presented ―Gate-Diffusion Input (GDI): A power-efficient method for digital combinatorial circuits‖ ieee transactions on very large scale integration (VLSI) systems, vol. 10, no. 5, october 2002.

[4] Balakrishna.Batta, Manohar.Choragudi, Mahesh Varma.D presented ―Energy Efficient full-adder using GDI technique‖ ijairissn: 2278-7844.

[5] P.ChaitanyaKumari, R.Nagendra presented ―Design of 32 bit Parallel Prefix Adders‖ iosr journal of electronics and communication engineering (iosr-jece) e-issn: 2278-2834,p- issn: 2278-8735. Volume 6, issue 1 (may. - jun. 2013).

[6] Dharani.A, Dr.M.Jagadeeswari presented ―Design of 16-bit Carry-Look Ahead adder and 8-bit Kogge-Stone adder using gate diffusion input logic‖ international journal of research in computer applications and robotic vol.2 issue.4, pg.: 136-144 s.

[7] S.Karthick, S.Karthika, S.Valarmathy presented ―Design and Analysis of Low Power Compressors‖ international journal of advanced research in electrical, electronics and instrumentation engineering, vol. 1, issue 6, December 2012.

[8] Sanjeev kumar1, Manoj Kumar presented ―Low Power high speed 3-2 compressor‖ international journal of electrical, electronic and mechanical controls issn(online).

[9] Ravi Nirlakalla, ThotaSubbarao, TalariJayachandra Prasad presented ―Performance evaluation of high speed compressors for high speed multipliers‖ serbian journal of electrical engineering vol. 8, no. 3, November 2011, 293-306.

[10] R.Naveen, K.thanushkodi, C.Saranya presented ―Low Power Wallace tree multiplier using gate diffusion input based full adders‖ international journal of electronics & communication engineering research (ijecer) vol. 1 issue 3, august – 2013.