

Efficient Implementation of 32 Bit PASTA for Low Area, High Speed and Low Power Applications

K. Jhansi Rani¹, G. Mary Sowjanya²

P.G. Student, Department of Electronics and Communication Engineering, VR Siddhartha Engineering College,
Kanuru, Vijayawada, India¹

Assistant Professor, Department of Electronics and Communication Engineering, VR Siddhartha Engineering College,
Kanuru, Vijayawada, India²

ABSTRACT: With modern technology, power dissipation, delay and area have become major and vital constraints in the electronic industry. Gate diffusion Input (GDI) is a technique that lowers power dissipation; delay by using less No. of transistors i.e. with low area. This paper describes the design of a 32 bit Parallel Self Timed Adder (PASTA) using GDI logic for low area, high speed and low power applications. PASTA is an asynchronous adder which works using a recursive formulation for performing multi-bit binary addition. The PASTA worked in a parallel manner for the bits which do not require any carry chain propagation. PASTA design is simple which uses multiplexers and half adder. All the carries are detected using a completion signal detection unit to terminate the recursive operation. Mentor graphics 130nm Pyxis schematic editor and Eldo simulator are used for simulation. The simulation results show that the proposed model attains better power consumption, delay and reduction in the number of transistors.

KEYWORDS: Asynchronous recursive Adder, CMOS Implementation, Gate diffusion input.

I. INTRODUCTION

Adder is considered as the heart of computational circuits and addition has been the core for many complex arithmetic circuits. In processors, adders are also used to increment program counters, calculate effective addresses, table indices and similar operations. Thus the performance of the processor is greatly influenced by the speed of the adders used by the processor. Nowadays the demand for portable devices is increasing day by day. The devices with small size, high speed and longer battery life with low power dissipation have much demand [7]. Device performance mainly depends on the performance of the adder. So in order to improve the performance of these devices we have to first improve the performance of adder circuits.

So many techniques are used to improve the performance of adder circuits. So many topologies and architectures are used for better performance of adder circuits [6]. Parallelism and pipelining concepts are introduced in adder designing. Circuits are of two types synchronous and asynchronous. So many adders are implemented for synchronous circuits, but some adders also implemented for asynchronous circuits also because of their advantages [1]. Synchronous circuits are based on clock pulse. That is synchronous systems comprise a number of subsystems that uses a clock to synchronize their operations. In asynchronous circuits, there is no global synchronization within the system. The results generated by the subsystems in an asynchronous system can be used by other subsystems as soon as they are produced without having to wait for a global clock pulse. Communication protocols are used to synchronize the Subsystems within the system locally. An asynchronous system involves request/acknowledge handshake signaling protocol [2]. The synchronous adder circuits operate at the worst rates and asynchronous ones operate only at average rates. Because of the advantages like no clock skew, lower power, and easing of global timing issues asynchronous circuits or self timed circuits are preferred for high speed applications.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 8, August 2016

This paper proposes the design of parallel self timed adder (PASTA) using GDI technique. The PASTA design is systematic and simple. This is an asynchronous, self timed adder which uses single rail pipelining. Half adder and multiplexers are used for PASTA design [4]. It requires minimum interconnection. The fabrication of PASTA of VLSI chip is very easy. PASTA works parallel for the bits which do not require any carry propagation. It is self-timed, which means that as soon as the addition is done, it will generate the completion signal to notify the completion of addition thereby overcoming the clocking limitations. The limitations of PASTA like high power consumption and usage of more No.of transistors can be overcome by implementing PASTA with GDI technique. By using GDI technique implementation the performance efficiency of PASTA is improved for low power, high speed and low area applications. In the next Section the PASTA design is explained. Section 3 describes the proposed Parallel Self Timed Adder with GDI technique and its operations. In Section 4 simulation results are discussed. Section 5 gives the conclusion.

II. DESIGN OF PASTA

A. Parallel self timed adder (PASTA)

Parallel Self Timed Adder (PASTA) is an asynchronous adder. The algorithm used in the implementation of PASTA is Cellular Automata Machine (CAM)[3]. The PASTA design is simple and regular [4]. Half adder and multiplexers are used for PASTA design. It requires minimum interconnection. In the First stage, the actual inputs are selected by the single bit half adder block of PASTA. The sum bit generated in the first stage is fed back to same half adder block and is added with the carry in from the previous bit adder, for recursive operations. The advantage of PASTA is that it is logarithmic and self timed. PASTA will generate a TERM signal when all the carry signals from individual bit adders become zero as an indication of completion of summation.

B. Architecture of parallel self timed adder

The architecture of PASTA is shown in figure 1(b). The multiplexer with 2 inputs has selection input denoted by SEL. Mux will first select the actual operands which are to be added during SEL = 0. When SEL=1 the feedback paths will be selected for recursive operation. The multiple iterations will be continued until all the carry signals will become zero values and then generates a completion signal.

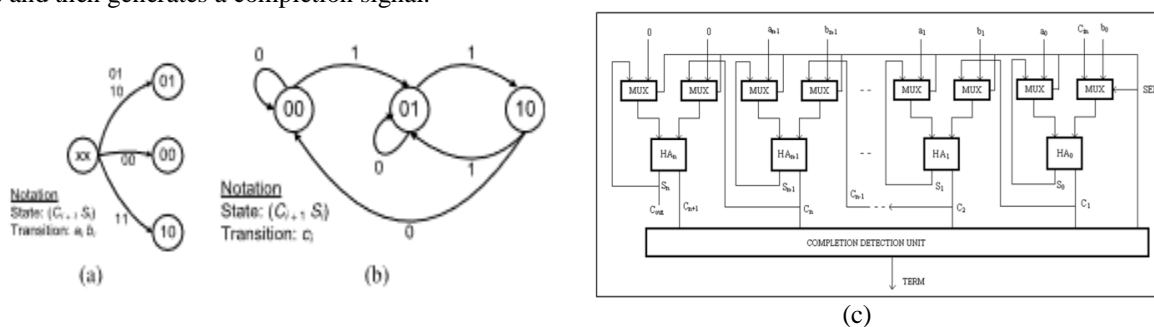


Fig 1 (a). State diagrams for PASTA. (a) Initial Phase. (b). Iterative phase. (c).General block diagram of PASTA

In Fig. 1(a), the initial phase and the iterative phase of PASTA are explained using the state diagrams. In the state diagrams every state is denoted by (C_{i+1}, S_i) in which C_{i+1} denotes carry out and S_i denotes the sum value of the i^{th} bit adder block. In the initial phase (SEL=0), the circuit will work as simple Half Adder. Because we use HAs instead of FAs, state (11) cannot appear. In the iterative phase the feedback path is started when (SEL = 1). The carry transitions (C_i) are allowed as many times as required to complete the recursive operation.

C. Recursive formula used in addition in PASTA

Let S_i^j and C_{i+1}^j denote the sum and carry, respectively, for i^{th} bit at the j^{th} iteration. The initial condition ($j=0$) for addition is formulated as follows:

$$S_i^0 = a_i \oplus b_i \tag{1}$$

The j^{th} iteration for the recursive addition is formulated by

$$S_i^j = S_i^{j-1} \oplus C_i^{j-1}, 0 \leq i < n \tag{2}$$

$$C_{i+1}^j = S_i^{j-1} C_i^{j-1}, 0 \leq i \leq n. \tag{3}$$

The recursion is terminated at k^{th} iteration when the following condition is met:

$$C_n^k + C_{n-1}^k + \dots + C_1^k = 0, 0 \leq k \leq n \tag{4}$$

D. CMOS implementation of PASTA

A CMOS implementation for the multiplexer showed in fig. 2(a), Schematics of sum module and carry module shown in fig. 2(b) and fig 2(c). The schematic of completion detection unit is shown in fig 2(d). In order to activate the high completion signal (TERM), the completion detection circuit is used. n input NOR gate is required. Pseudo NMOS design is used for the completion unit to avoid the high fan in problems by connecting all NMOS in parallel. The PMOS transistor is connected to V_{dd} in this rated design. It acts as a load register..

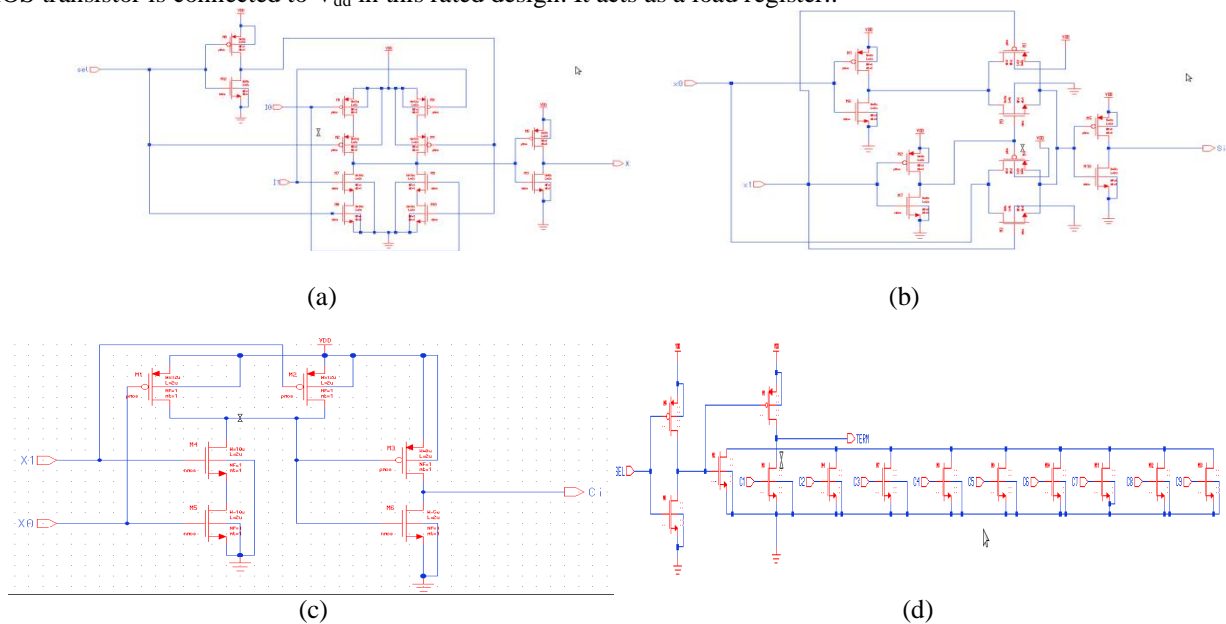


Fig 2. (a).MUX Module (b)Sum Module.(c).Carry Module (d)Completion Detection circuit for 8 bit PASTA.

The SEL bar which is the negative of SEL is connected with the complete unit to generate a TERM signal. It is for the confirmation that the completion cannot be turned on by mistake when the original inputs are initially selected. It also avoids the pull up PMOS transistor to turn on always, So that, static current will flow only when the actual computation is done.

III. PROPOSED 32 BIT PASTA DESIGN

A. Gate diffusion input technique

In modern technology power dissipation, delay and area have become major and vital constraints for high speed and low power applications. Transistor count is the major factor which decides the design complexity of the circuit. But the proposed PASTA design has the drawback of high power consumption and uses a large number of transistors which further increases are. This drawback can be overwhelmed by implementing PASTA using Gated Diffusion Input (GDI) technique [8]. The GDI method is based on the use of a simple cell as shown in Fig.3.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 8, August 2016

The basic GDI cell looks like CMOS inverter, but there are some conflicts. The main difference of GDI cell from CMOS inverter is that GDI cell has three inputs whereas CMOS inverter has single input. The GDI cell comprises of four terminals out of which three are input nodes and other is output node. The three inputs are G: The common gate input of NMOS and PMOS, P: Input to the source or the drain of PMOS, N: Input to the source or drain of NMOS. The output node: The common diffusion node of both transistors. The GDI approach uses two transistors for implementing various complex logic functions. This technique is appropriate for low power designs with minimum No.of transistors for low area, low power and high speed applications.

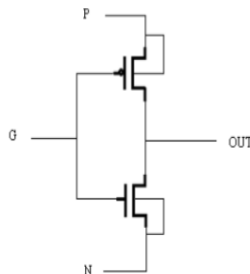


Fig 3. Basic GDI cell

By applying different inputs at different nodes we can implement so many logic functions in GDI cell. If A, B and C are inputs to G, P and N respectively, then the output function of GDI cell is given by

$$\text{Out} = AC + \bar{A}B \tag{5}$$

It is written by using Shannon expansion. Any logic function can be implemented according to this expression using this GDI cell. By observing the logic functions shown in Table 1, we can say that 3 input GDI cell is an extension of the single input CMOS inverter. With this we can say that any n input CMOS structures can be extended to (n+2) input GDI cell by adding inputs, P instead of V_{dd} and input N instead of V_{ss} . This extended implementation can be represented by the following logic expression

$$\text{out} = F(x_1, \dots, x_n)P + F((x_1, \dots, x_n)N \tag{6}$$

Where $F(x_1..x_n)$ is a logic function of n-MOS block.

Table I Basic functions using GDI CELL

N	P	G	OUTPUT	FUNCTION
0	1	A	A^1	INVERTER
0	B	A	A^1B	F1
B	1	A	A^1+B	F2
1	B	A	$A+B$	OR
B	0	A	AB	AND
C	B	A	A^1B+AC	MUX
B^1	B	A	A^1B+B^1A	XOR
B	B^1	A	$AB+A^1B^1$	XNOR

Table I describes the basic functions which are implemented using GDI Technique. Multiple input gates can be implemented by combining several GDI cells. By applying different inputs for N,P and G inputs different functions will be implemented.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 8, August 2016

B. Design of PASTA using GDI Technique

The PASTA adder designed using the GDI technique in order to reduce the power consumption, transistor count and delay compare with CMOS implementation. The different modules of PASTA are designed using GDI technique.

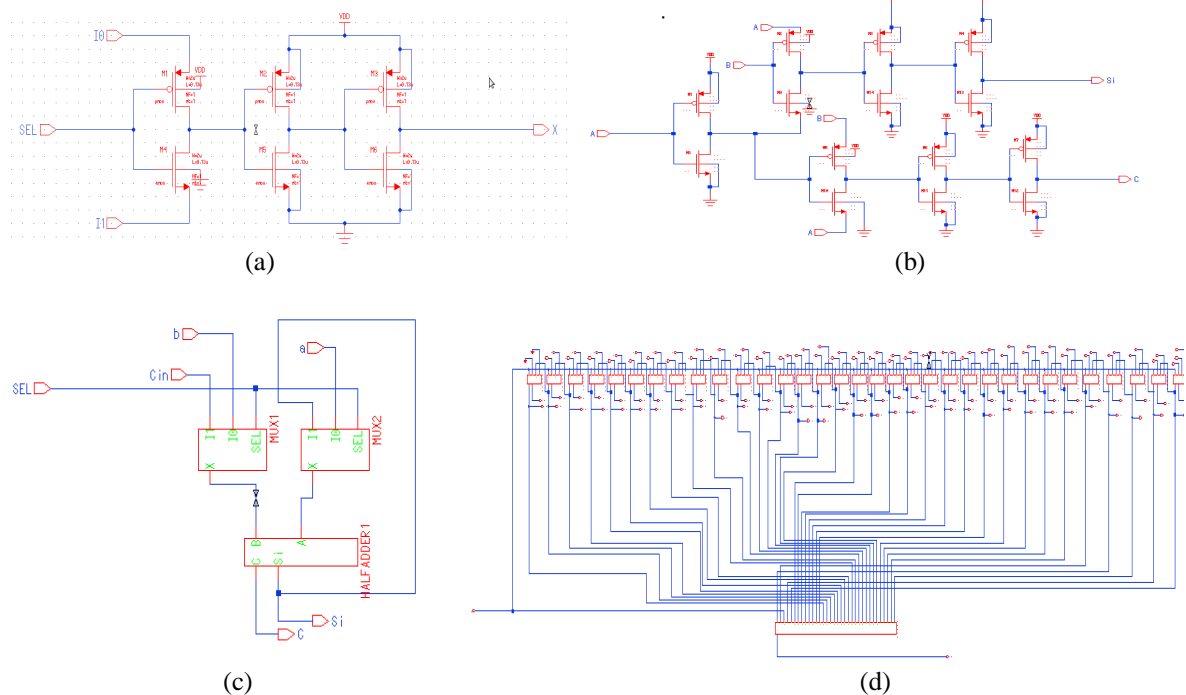


Fig 4 .(a) Schematic of mux. (b).Schematic of a half adder using GDI.(c).Schematic of half adder and mux module.(d).Schematic of proposed 32 bit PASTA using GDI Technique.

The schematics of MUX and half adder are shown in figures 4(a) and 4(b).By implementing the Half Adder using GDI logic the transistor count can be reduced. With this power consumption can be reduced. Generally, CMOS Half Adders have two separate circuits for sum module and carry module. But using GDI logic, single circuit is used for half adder implementation. The design of an HA using GDI logic is shown in figure 4(b).Half adder and mux are combined to get half adder and a mux module block shown in 4(c). This block is used in PASTA design. Schematic of 32 bit PASTA with GDI implementation is shown in fig 4(d).

IV. SIMULATION RESULTS

We can understand the superiority of the proposed 32 bit PASTA design with GDI technique by comparing the results with existing PASTA with CMOS design. Mentor Graphics 130nm technology is used. The Pyxis schematic of Mentor Graphics is used in order to draw the schematics of the circuit. The performance analysis of the PASTA using GDI design is done using the simulation results. Simulation is done using Eldo simulator. The output waveforms are viewed using an E-Z wave viewer.

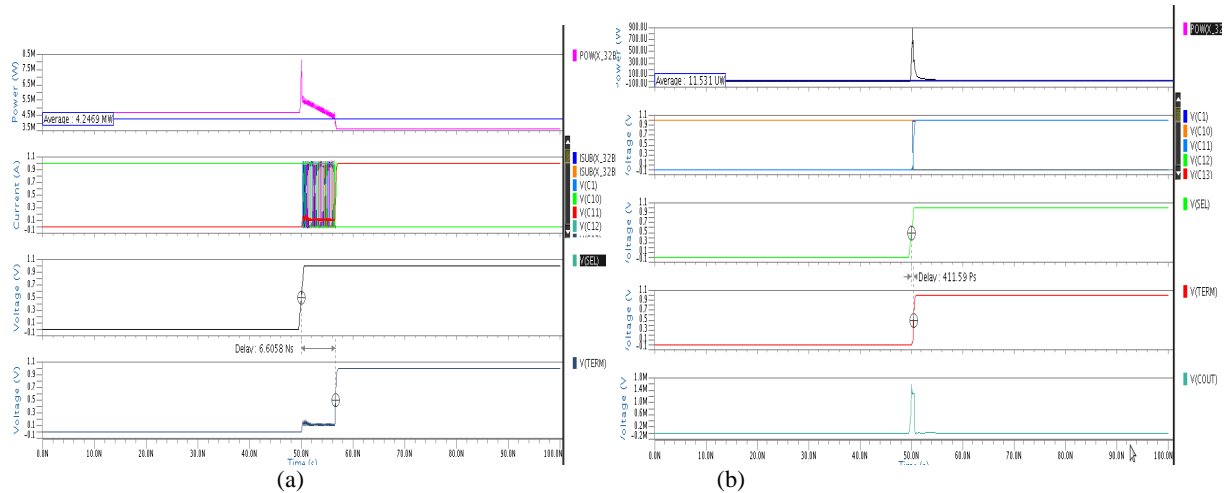


Fig5.(a)Simulation results of existing PASTA with CMOS implementation.(b).Simulation results of proposed 32 bit PASTA using GDI

The simulation results were observed for different cases of random operands. The values are tabulated in Table II. By observing the simulation results we can understand the superiority of the PASTA implemented using GDI over the PASTA using CMOS. The Power dissipation is in the order of (nw) in proposed adder while it is in the order of (uw) in existing system. The delay also reduced in the proposed system. Delay we observed in the order of (ps) in proposed PASTA while it is in the order of (ns) in existing PASTA. The no. of transistors used in the implementation of proposed PASTA are less compared to existing PASTA. By all these results we can recommend this proposed PASTA for low power, high speed and low area applications.

The input operands taken for different cases are, Best case: Addition of $(FFFF\ FFFF)_{32}$ and $(0000\ 0000)_{32}$.
 Average case: Addition of $(3F05\ 0FC0)_{32}$ and $(01300041)_{32}$. Worstcase: Addition of $(FFFF\ FFFF)_{32}$ and $(0000\ 0001)_{32}$.

Table II Comparison of existing PASTA and proposed PASTA

Cell name	Delay in CMOS technique(ns)	Delay in GDI Technique(ns)	Avg Power in CMOS (uw)	Avg power in GDI (uw)	No. of transistors in CMOS	No. of transistors in GDI
MUX	0.208	0.144	0.622	0.0017	10	6
HALF ADDER	0.264	0.371	206.9	0.783	16	14
4 BIT PASTA	2.280	0.275	2351	1.537	137	107
8 BIT PASTA	2.38	0.285	2765	1.645	245	191
32BITPASTA (best case)	0.246	0.244	4734	1.394	893	660
32BITPASTA (average case)	1.43	0.377	4000	2.254	893	660
32BITPASTA (worst case)	6.605	0.411	4246	11.531	893	660

According to simulation results the power dissipation is greatly reduced with the usage of GDI logic. Delay and transistor count also reduced. The comparisons are shown in the graphs. In the graphs we can observe that there is 40% reduction in transistor count, 70% reduction in delay and more than 80% reduction in power dissipation. So the proposed PASTA is highly suitable for low area, high speed and low power applications.

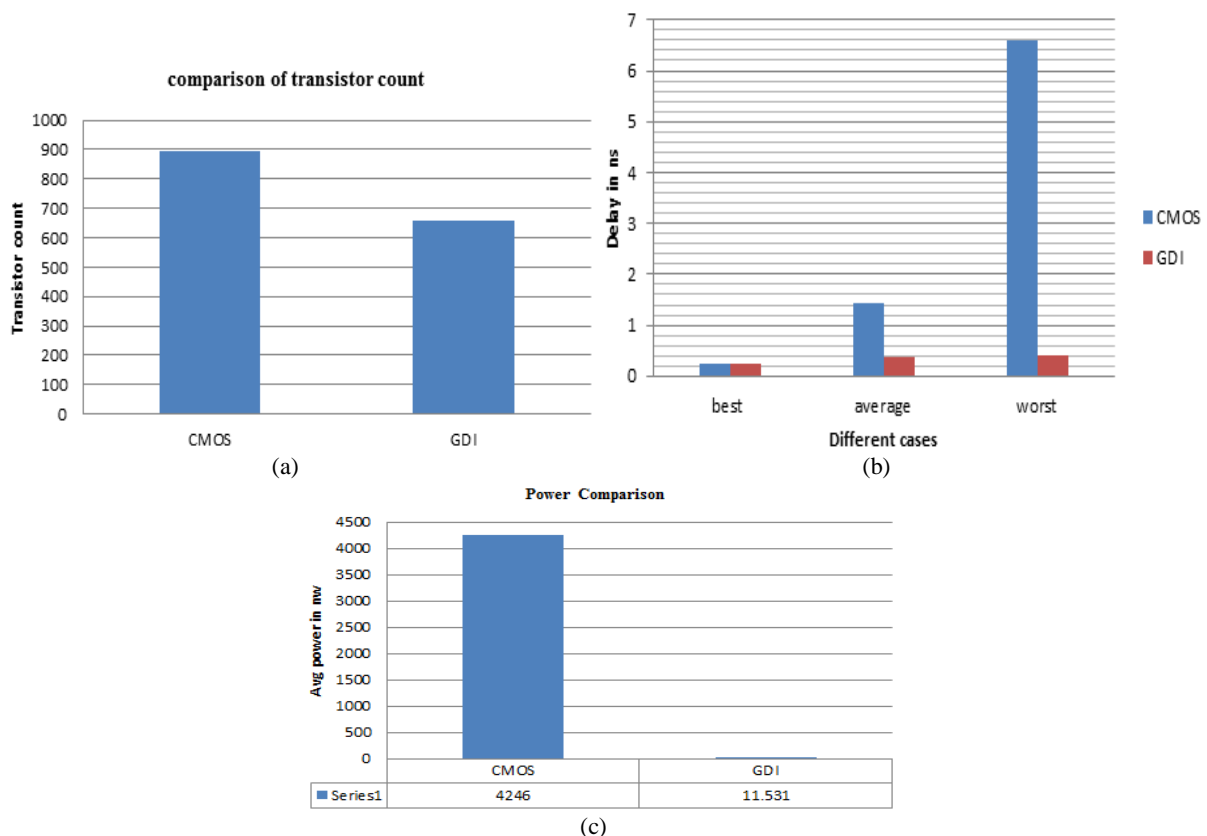


Fig 6.(a). Comparisons of transistor count.(b).Comparison of delay of 32 bit PASTA .(c).Comparison of power of 32 bit PASTA.

V.CONCLUSION

The Parallel Self Timed Adder (PASTA) design is systematic and easy. Half adder and multiplexers are used for PASTA design. The architectural design and CMOS implementation are explained. The disadvantage of PASTA is, it uses more No.of transistors compared to synchronous adders and dissipates high power. To avoid this disadvantage PASTA is implemented using Gate Diffusion Input (GDI) technique. With minimum No.of transistors the 32 bit PASTA was implemented with GDI. The 32 bit PASTA is very simple adder and is more efficient in terms of area, interconnection and power consumption wise than the previous self-timed adder. So this adder is highly suitable for low area, low power and high speed applications.

REFERENCES

- [1]D. Geer, "Is it time for clock less chips? [Asynchronous processor chips]," IEEE Comput., vol. 38, no. 3, pp. 18–19, Mar. 2005.
- [2]J. Sparsø and S. Furber, Principles of Asynchronous Circuit Design. Boston, MA, USA: Kluwer Academic, 2001.
- [3]P. Choudhury, S. Sahoo, and M. Chakraborty, "Implementation of basic arithmetic operations using cellular automaton," in Proc. ICIT, 2008, pp. 79–80.



ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 8, August 2016

- [4] "Recursive Approach to the Design of a Parallel Self-Timed Adder", Mohammed ZiaurRahman, Lindsay Kleeman, and Mohammad AshfakHabib, 1063-8210 © 2014 IEEE.
- [5] N. Weste and D. Harris, CMOS VLSI Design: A Circuits and Systems Perspective. Reading, MA, USA: Addison-Wesley, 2005.
- [6] M. Anis, S. Member, M. Allam, and M. Elmasry, "Impact of technology scaling on CMOS logic styles," IEEE Trans. Circuits Syst., Analog Digital Signal Process., vol. 49, no. 8, pp. 577–588, Aug. 2002.
- [7] A. Morgenshtein, A. Fish, and I. A. Wagner, "Gate-diffusion input (GDI) – A novel power efficient method for digital circuits: A detailed methodology," in Proc. 14th IEEE Int. ASIC/SOC Conf., Sept. 2001, pp. 39–43.
- [8] Gate-diffusion input (GDI) – A power efficient method for digital combinatorial circuits, IEEE Trans. VLSI Syst., vol. 10, pp. 566–581, Oct. 2002.
- [9] Zhanfeng Zhang and Liyuan Sheng-A new Adder Theory based on half adder and implementation in CMOS Gates. I. J. Image, Graphics and Signal Processing, 2010, 2, 11-17 Published Online December 2010 in MECS.
- [10] C. Cornelius, S. Koppe, and D. Timermann, "Dynamic circuit techniques in deep submicron technologies: Domino logic reconsidered," in Proc. IEEE ICICDT, Feb. 2006, pp. 1–4.
- [11] W. Liu, C. T. Gray, D. Fan, and W. J. Farlow, "A 250-MHz wave pipelined adder in 2- μ m CMOS," IEEE J. Solid-state Circuits, vol. 29, no. 9, pp. 1117–1128, Sep. 1994.
- [12] S. Nowick, "Design of a low-latency asynchronous adder using speculative completion," IEE Proc. Comput. Digital Tech., vol. 143, no.5, pp. 301–307, Sep. 1996.
- [13] F.-C. Cheng, S. H. Unger, and M. Theobald, "Self-timed carry-lookahead adders," IEEE Trans. Comput., vol. 49, no. 7, pp. 659–672, Jul. 2000.