Volume 4, Issue 7, July 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Design Analysis of Optimized Self-Testing Adder

Suyash Jain
Department of Electronics and Communication LNCT Bhopal, India

Rita Jain
Department of Electronics and Communication LNCT Bhopal, India

Abstract— Very large scale integration technology integrates a large system into a single chip. Self checking scheme is becoming an important design technique to full fill the requirements of modern computer systems with full reliability. The paper proposes a analysis of design to implement low cost self testing full adder using duplicated code scheme differential XOR gate for sum and "sharing" transistor technique for carry. The duplicated scheme has the advantage to be totally self-checking for single faults. The designed adder will be self checking for primary inputs and no extra checking circuitry is needed this will reduce the area, hardware and will increase the speed of the adder.

Keywords— self Checking, full adder, differential XOR gate, sharing transistor, CMOS logic.

I. INTRODUCTION

Aggressive new chip design technologies frequently affect chip reliability during functional operation. Self Checking Circuit Designing is a suitable approach to the design of complex VLSI IC's to cope with the growing difficulty of on line and off line testing[12]. Self checking circuits are class of circuits in which occurrence of fault can be determined by observation of the outputs of the circuits. An important subclass of these self-checking circuits is known as totally self-checking (TSC) circuits[12]. Self-checking is a design philosophy which assures on-line testability of a circuit. Error detecting codes add extra bits to original output bits so that errors in the output bits are detectable. The validity of the output code words is verified by a checking circuitry. [13]

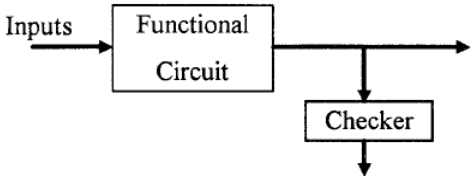

Figure 1 Basic Block diagram of self checking logic circuit

The checking circuit maps code word inputs to code word outputs and non code word inputs to non-code word outputs. By observing the output of the checking circuit it is possible to determine whether there is any fault in the functional or checking circuit.

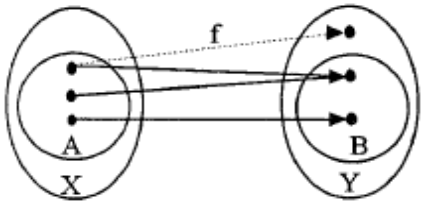

Figure 2 Input output mapping in self checking logic circuit

In fig 2 a Solid line represents a mapping between an input code word and an output code word whereas a dotted line represents a mapping in a presence of fault f [13]. The mapping results in a non code word output.TSC circuits are used to detect errors concurrently with normal operation. These circuits operate on encoded inputs to produce encoded outputs. TSC checkers are used to monitor the outputs to indicate error when a non-code word is detected. The concept of TSC circuits generalized in [15] as follows.

Definition 1. **:** A circuit is self-testing if for each fault f, present in the circuit, there exists at least an input code word that produces a non code word output
Definition 2. **:** A circuit is fault-secure if for any fault, f present in the circuit, the output for a code word input will be the correct output or a non valid output code word
Definition 3**:** A circuit is *totally self-checking* if it is fault secure and self-testing.
Definition 4**:** A circuit is *code disjoint* if it always maps code word inputs into code word outputs and non code word inputs into non code word outputs.

***Definition* 5:** A circuit is a *totally self-checking checker* if it is self-testing and code-disjoint.
A totally self-checking (TSC) functional block satisfies the two following properties:

- For any valid input code word and any single fault, the circuit, either produces an invalid code word on the output, or (fault secure) i.e. does not produce the error on the output.
- Any single fault is detectable by some valid input code word (self-testing property)

II. CONVENTIONAL CMOS STYLE FULL ADDER

The CMOS design style is not area- efficient for complex gates with larger fan-ins. Thus, care must be taken when a static logic style is selected to realize a logic function. The CMOS structure combines PMOS pull-up and NMOS pull-down networks to produce considered outputs. In this style all transistors are arranged in completely separate branches, each may consist of several sub-branches. Mutually exclusiveness of pull-up and pull-down networks is of a great concern. Figure 3 shows the conventional CMOS 28-transistor adder [3, 5].

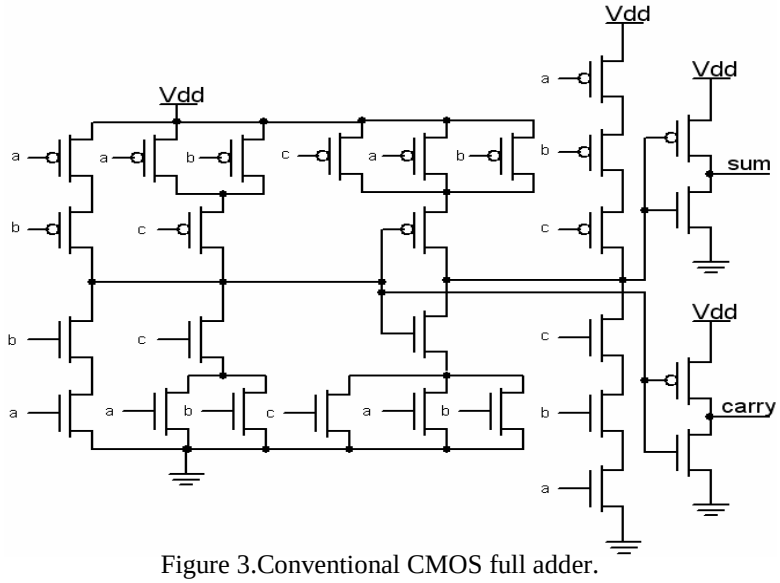

Figure 3.Conventional CMOS full adder.

III. DUAL DUPLICATED CODE SCHEME XOR AND XNOR

The exclusive-OR (XOR) and exclusive-NOR (XNOR) are fundamental components in full adders[4 , 6 , 9, 11] and in larger circuits such as multipliers, parity checkers etc.[2 , 7] The performance of these larger circuits is affected by the individual performance of the included XOR/XNOR gates.The XOR gate can be implemented using AND, OR and NOT CMOS gates. However, this solution requires large hardware overhead. On the other hand, pass transistor logic is attractive as fewer transistors are needed to implement important logic functions, smaller transistors and smaller capacitances are required, and it is faster than conventional CMOS.
Many different pass-transistor logics have been proposed. These structures are generally, composed by an NMOS pass-transistor network to realize the logic function followed by a suitable level-restoring circuit. A novel differential XOR designed in CMOS pass transistor logic is presented in Fig 4(b)[1]. This gate has dual inputs and generates dual outputs. XOR and XNOR functions are performed with only four transistors.

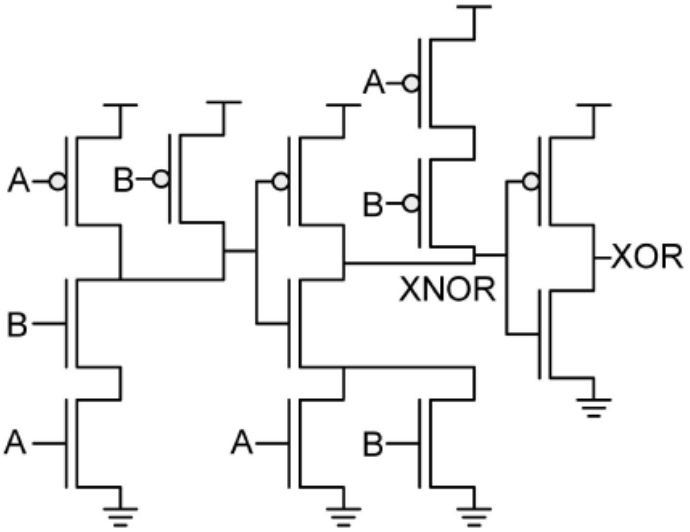

Figure 4(a) CMOS XOR (12 gates)

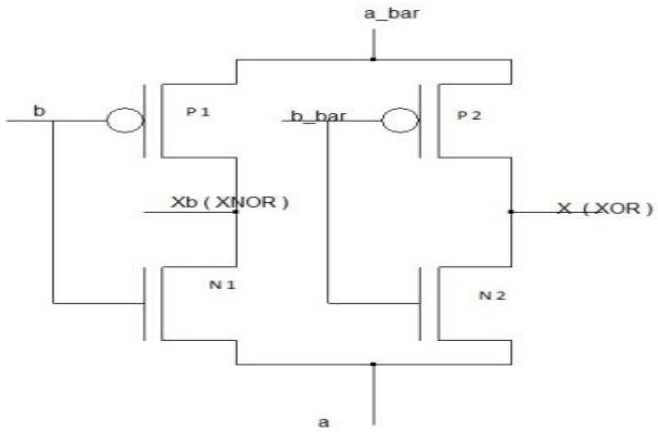

Figure 4(b) Self Checking Differential XOR Schematic using pass transistor logic (4 gates)[1]

In figure 4 (a) we can see that it requires 12 transistors for designing of CMOS XOR and XNOR gate[2] while figure 4(b) shows that we need only 4 transistors for the designing of self checking differential XOR.[1].The gate has dual inputs and generates dual outputs.

IV. DIFFERENTIAL XOR (TOTALLY SELF CHECKING PROPERTY) ANALYSIS

The highest level of protection is offered by the combination of fault secure and the self testing properties.[8 , 14].The behavior of the differential XOR Gate of figure 4(b) is analyzed in terms of fault secure and self testing properties which includes logical stuck at faults and stuck open faults. For inputs, we consider the logical *stuck-at fault model (gate* stuck-at 0 and gate stuck at 1 fault).a and a~, b and b~ are normally complementary inputs. Table 1 shows the fault secure property for primary inputs[1].

TABLE 1
PRIMARY INPUTS FAULT SECURE PROPERTY

Inputs				**Outputs**		**Conclusion**
a	a~	b	b~	X	Xb	
0	0	0	0	0	0	Multiple Fault Detected
0	0	0	1	0	0	Single Fault Detected
0	0	1	0	0	0	Single Fault Detected
0	0	1	1	0	0	Multiple Fault Detected
0	1	0	0	1	1	Single Fault Detected
0	***1***	***0***	***1***	***0***	***1***	***OK (valid inputs)***
0	***1***	***1***	***0***	***1***	***0***	***OK (valid inputs)***
0	1	1	1	0	0	Single Fault Detected
1	0	0	0	0	0	Single Fault Detected
1	***0***	***0***	***1***	***1***	***0***	***OK (valid inputs)***
1	***0***	***1***	***0***	***0***	***1***	***OK (valid inputs)***
1	0	1	1	1	1	Single Fault Detected
1	1	0	0	1	1	Multiple Fault Detected
1	1	0	1	1	1	Single Fault Detected
1	1	1	0	1	1	Single Fault Detected
1	1	1	1	1	1	Multiple Fault Detected

The self testing property of differential XOR can be proved by using the fault equivalence principle, for single transistor stuck open faults. A PMOS transistor open is equivalent to a PMOS whose gate is stuck-at 1, and a NMOS transistor open is equivalent to a NMOS whose gate is stuck-at 0. [14]
To prove the self-testing property, we will show that for each fault there is ***at least one input vector***, occurring during the circuit normal operation that detects it.

- ***Transistor P1 stuck-open***

P1 gate receives the signal b.
P1 stuck-open ⇔ b stuck-at 1. This fault, as shown in the table 1, is detectable by the input vectors (a a~ **b** b~) = (01**0**1) and (a a~ **b** b~) = (10 **0**1).

- ***Transistor N1 stuck-open***

N1 gate receives the signal b.
N1 stuck-open ⇔ b stuck-at 0. The input vector (a a~ b b~) = (0110) or (10 10) detects this fault

- ***Transistor P2 stuck-open***

P2 gate receives the signal b~.
P2 stuck-open ⇔ b~ stuck-at 1. The input vector (a a~bb~) = (0110) or (10 10) detects this fault

- ***Transistor N2 stuck-open***

N2 gate receives the signal b~.
N2 stuck-open ⇔ b~ stuck-at 0. The input vector (aa~bb~) = (0101) or (10 01) detects this fault
By observing the input vectors we see that all these vectors belong to the set of valid input codes. Consequently, the differential XOR is self-testing for all single transistor stuck-open faults.

V. PROPOSED SELF CHECKING ADDER.

Hardware redundancy is the best way to increases the reliability of *TSC* adder. The simplest hardware redundancy approach is to designing a *TSC* logic circuit using duplication. The design provides two copies of the circuit output. The second copy produces output values complementing the value of the first copy, and a (TRC) checkers makes a bitwise comparison of the outputs, Whenever the natural and complementary outputs configurations differ from each other, or whenever a fault affects one of the self-checking *TRC* checkers, the error signal reports the presence of fault.

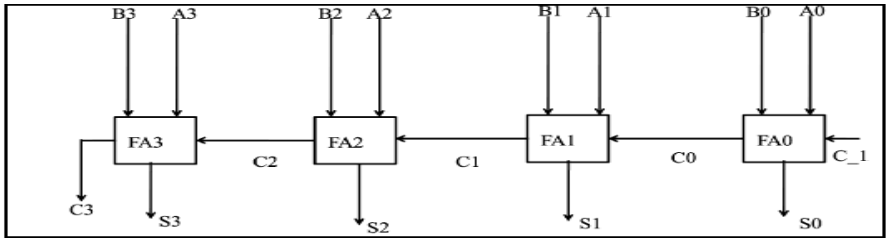

Figure 5. Simple 4 bit adder

Figure 5 shows a 4 bit adder consisting of 4 full adders. Each full adder have 3 inputs namely A,B (which are primary inputs) and C (previous carry), and two outputs S (sum) and C (carry) .We will implement our full adder by totally self checking XOR whose self checking property has been proved above which has been designed using duplicated code, as XOR is the basic component in designing of adder circuits.
Figure 6 shows the schematic of sum function which has been designed using self testing XOR. The output function generates two outputs one is a XOR b XOR c and other one is the compliment of the same. TRC(Two Rail Checker) checker checks the output of both and generates an error signal in the presence of fault. The advantage of designing the sum logic using self checking XOR makes it totally self checking for all single faults. The sum logic requires 8 transistors and 6 transistors are required for complimentary inputs. logic expression for sum is

Sum = a XOR b XOR c

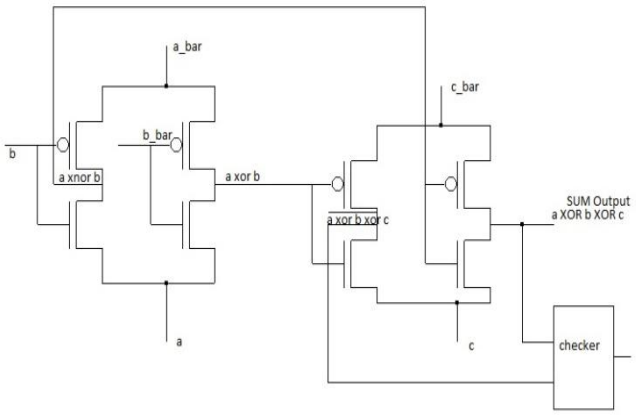

Figure 6. Three inputs Xor (sum function) circuit. (Using self checking xor)

Figure. 7 illustrate a sharing transistor implementation of carry function. Its logic expression is as follows

Carry = ab + bc + ca

The transistors in circles (Fig. 7) are sharing transistors. "Sharing transistors" provide the possibility of sharing the transistors of different paths to create a new path from supply lines to an output. These transistors help the circuit to reduce delay. These transistors must be arranged in such a way that not only validate the correctness of the circuit, but also preserve pull-up and pull-down networks mutually exclusive. The circuit in figure below is also designed using duplication scheme in which the output from pull up network is compared with the output of pull down network using a checker which generates an output high in presence of fault. Thus we can see that we need 10 transistors to implement the carry function.

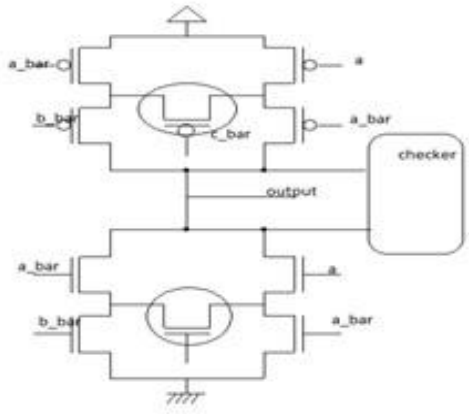

Figure 7 Carry generator circuit. (Using sharing transistor)

Figure 8 Block Diagram of a Checker (y = x_bar)

TABLE 2
ERROR DETECTING CAPABILITY OF CHECKER.

X	Y = X_bar	O/P
0	0	Error
0	1	Ok
1	0	Error
1	1	Ok

A checker is basically an equality comparator gate. Which compares the input and gives the output high in presence of fault .Table 2 shows the error detecting property of a checker. A total of 18 transistors (8 for sum and 10 for carry) are required for implementing our design (generally complimentary inputs are present) which are less then conventional one (which requires 28 transistors) and also making our adder totally self checking. The improvement in speed over conventional CMOS adder is achieved due to less delay in the path due to sharing transistor, which leads to a better power-delay product. A complete fast full adder can be built by placing the sum circuit of Fig. 6 and the carry generator of Fig. 7 together as a whole circuit and these full adder cells can be placed to make a complete fast self testing adder with less number of transistors as well.

VI. SIMULATION RESULT

The Layout of the proposed sum and carry has been designed in Microwind and simulation has been performed in CMOS 120 nm technology with V_{dd} 1.2 V. Temperature has been kept 27^{o}C. Default Values of nmos and pmos has been taken.A level restoring circuit is used while simulating sum.

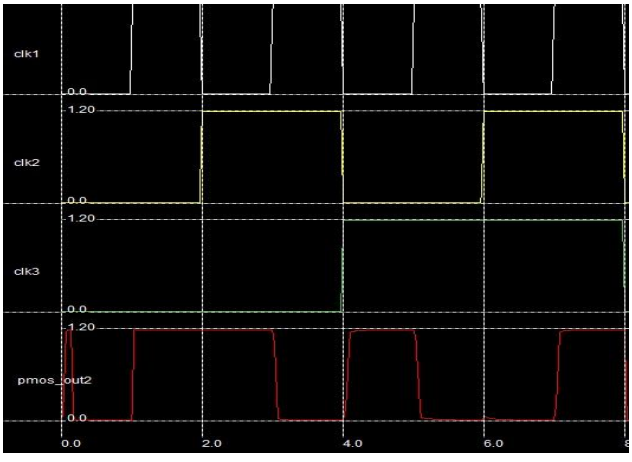

Figure 9 Waveform of Sum for Proposed Adder

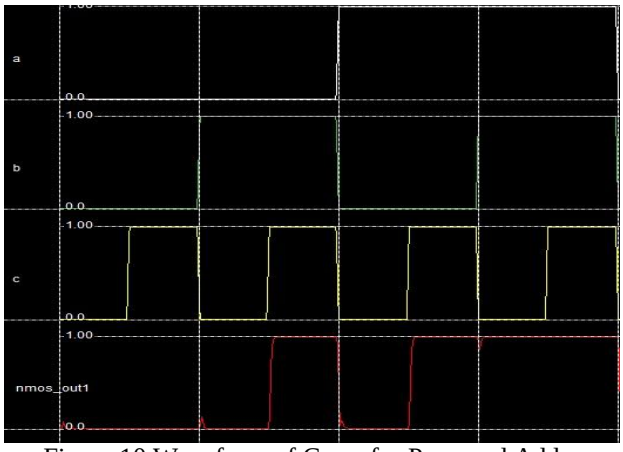

Figure 10 Waveform of Carry for Proposed Adder

VII. CONCLUSION

The conventional full adder implemented using CMOS requires a large area and having a delay so the adders (4 bit ,8 bit , 16 bit etc)circuit implemented using this CMOS technique requires larger area and less speed yet not self testing which will increase the hardware as well as cost of the circuit While an efficient optimized design of this self checking full adder cell for an adder will certainly, improve the performance (delay , speed , power delay product, self testing) and reduce the hardware overhead and also making the adder circuitry totally self checking.

REFERENCES

[1] BelgacemHamidi ,KhedriChiraz , FradiAymen , TaukriRached "Four Transistors Self Checking Differential XOR" Signals, Circuits and Systems (ISSCS), 2011 10th International Symposium IEEE July 2011.

[2] Wonhak Hong, RajashekharModugu, and Minsu Choi "Efficient Online Self-Checking Modulo 2n + 1 Multiplier Design", IEEE Transaction on Computers Sept 2011.

[3] KeivanNavi , OmidKavehie , MahnoushRouholamini "A Novel CMOS Full Adder" 20th International Conference on VLSI Design , 2007 IEEE

[4] J. Wang, S. Fang, W. Feng, New Efficient Designs for XOR and XNOR functions on the Transistor Level. IEEE Journal of Solid State Circuits,Vol. 29, No. 7, 1994, pp. 780-786

[5] R. Zimmermann, W. Fichtner, "Low-power logic styles CMOS versus pass-transistor logic," IEEE J. Solid-StateCircuits, Vol. 32, pp. 1079–1090, July 1997

[6] Tien Bui, Y. Wang, Y. Jiang, Design and Analysis of Low-Power 10 Transistor Full Adders Using Novel XOR-XNOR Gates, IEEE Trans.on circuits and system-II, Analog and Digital Signal Processing, Vol.49, No. 1, January 2002, pp. 25-30.

[7] Michael NICOLAIDIS, Ricardo O. DUARTE, Salvador MANICH, Joan Figueras, Fault-Secure Parity Prediction Arithmetic Operators,IEEE Design & Test of computers, Vol. 14, Apr 1997, pp. 60-71.

[8] B. Hamdi, H. Bederr, M. Nicolaidis, A tool for automatic generation of self-checking data paths, in: Proceeding of 13th IEEE VLSI TestSymposium, Proceedings (VTS'95), 30 Apr-3 May 1995, pp 460 – 466

[9] S .R. Chowdhury, A. Banerjee, A. Roy, H. Saha, "A High Speed 8 Transistor Full Adder Design using Novel 3 Transistor XOR Gates". International Journal of Electronics, Circuits and Systems II, 2008, pp. 217-223.

[10] A. P. Chandrakasan, S. Sheng and R. W. Brodersen, Low-Power CMOS Digital Design. IEEE Journal of Solid State Circuits, Vol. 27, No. 4,April 1992, pp. 473–484..

[11] Kuo-Hsing Cheng, Chih-Sheng hang, The Novel Efficient Design of XOR/XNOR function for Adder applications, in: Proceeding of the 6th IEEE International Conference on Electronics, Circuits and SystemsICECS'99, Vol. 1, Pafos, Cyprus, Sep. 5-8, 1999, pp.29-32.

[12] Michael Nicolaidis, On-line testing for VLSI: state of the art and trends,Integration, the VLSI Journal, Volume 26, Issues 1-2, 1 December1998, pp. 197-209

[13] Fadi Y Busaba "VHDL Description Of Self Checking Circuits " 1996 IEEE

[14] ParagK. Lala, Fault-tolerant &Fault Testable Hardware Design, Prentice-Hall,1985

[15] D. A. Anderson and G. Metze, Design of totally self-checking check circuits for m-out-of-n codes. IEEE Trans. on Computers, vol. 22, No.3, March 1973, pp. 263-269

