A Fault-Tolerant Full Adder in Double Pass CMOS Transistor

Abdelmonaem Ayachi, Belgacem Hamdi

Abstract—This paper presents a fault-tolerant implementation for adder schemes using the dual duplication code. To prove the efficiency of the proposed method, the circuit is simulated in double pass transistor CMOS 32nm technology and some transient faults are voluntary injected in the Layout of the circuit. This fully differential implementation requires only 20 transistors which mean that the proposed design involves 28.57% saving in transistor count compared to standard CMOS technology.

Keywords—Semiconductors, digital electronics, double pass transistor technology, Full adder, fault tolerance.

I. INTRODUCTION

Addition is one of the fundamental arithmetic operations. It is used extensively in many VLSI systems such as microprocessors and some transient faults are voluntary injected in the Layout of the circuit. This fully differential implementation requires only 20 transistors which mean that the proposed design involves 28.57% saving in transistor count compared to standard CMOS technology.

In this paper, we present a self-checking full adder based on two-rail encoding scheme. To prove the efficiency of the proposed method, the circuit is simulated in double pass transistor CMOS 32nm technology and some transient faults are voluntary injected in the Layout of the circuit. The proposed design involves 28.57% saving in transistor count compared to standard CMOS technology.

The paper is organized as follows. In Section II, we describe the proposed design. Section III shows the simulation results in 32 nm double pass transistor process technology. Conclusions are given in Section VI.

II. PROPOSED DESIGN

The blooming development of Computer Science has led to the growth of integrated circuit (IC) devices. Most of the Very Large Scale IC (VLSI) applications, such as digital-signal processing and microprocessors, use arithmetic operations extensively [30]. In addition, among these widely used operations, subtraction and multiplication are most commonly applied. The 1-bit full adder is the building block of these operation modules.

A Full Adder is a three-input two-output block, where the inputs are the two bits to be summed, a and b, and the carry input bit (Cin), which derives from the calculations of the previous digits. The outputs are the result of the sum operation, Sum, and the resulting value of the carry output bit (Cout) [31].

Many full adders have been designed and published in literature. They are built upon different logic styles [32]. In this paper, we present a self-checking full adder based on the double pass transistor technology.
A. Double Pass-Transistor Logic (DPL)

The basic difference of pass transistor logic compared to the CMOS logic style is that the source side of the logic transistor networks is connected to some input signals instead of the power lines. In the Double Pass Transistor Logic (DPL) style [33]-[35], both NMOS and PMOS logic networks are used in parallel.

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. However, the pass transistor gates generate degraded signals, which slow down signal propagation. This situation will be more critical when the output signals should be propagated to next stage as is the case for the carry gate in ripple carry adder. To avoid this signal degradation, inverters are added in the outputs of the circuit.

The schematic of the proposed static DPL logic circuit for a full adder is shown in Fig. 1.

Inverters are added to restore degraded signals generated by the differential SUM and carry gate. This fully differential implementation requires only 20 transistors which mean that the proposed design involves 28.57% saving in transistor count compared to standard CMOS technology.

B. Self-Checking Design

Self-checking circuits are increasingly becoming a suitable approach to the design of complex VLSI circuits, to cope with the growing difficulty of on-line and off-line testing [36]. They are class of circuits in which occurrence of fault can be determined by observation of the outputs of the circuits.

Self-checking circuits are based on an appropriate coding of the inputs and outputs of the circuit. Code checkers are used to monitor whether the circuit responses are within the output code space. As long as this condition is fulfilled, the output is assumed to be correct. If the code checker reveals a non code word, an error is detected [37].

The checker determines whether the output of the circuit is a valid code word or not. It also detects a fault occurring within itself [38]. Double-rail checker is based on the dual
duplication code as shown in Fig. 2. It compares two input words X and Y that should normally be complementary (\( \bar{y} = \bar{x} \)) and delivers a pair of outputs coded in dual-rail.

![Fig. 2 Dual rail checker cell](attachment:fig2.png)

A self-testing dual-rail checker can be designed as a parity tree where each XOR gate is replaced by a dual rail checker cell. The resulting checker is also an easily testable circuit since only four code inputs are needed to test a dual rail checker of any length [39]. This checker is important in self-checking design since it can be used to check dual blocks (and duplicated blocks by inverting the outputs of one of them). However, its more significant use consists on the compaction of the error indication signals delivered by the various checkers of a complex self-checking circuit. Each checker delivers a pair of outputs coded in dual-rail. Thus, the dual-rail checker can compact the dual-rail pairs delivered by the various checkers of the system into a single dual-rail pair. This pair delivers the global error indication of the system.

III. SIMULATION RESULTS

The full adder circuit is implemented in full-custom 32nm DPL technology [40]. SPICE simulations of the circuit extracted from the layout, including parasitic, are used to demonstrate that this adder has an acceptable electrical behaviour. The SPICE simulation of the differential full adder is as shown in Fig. 3.

As it is shown in Fig. 3 (b), the differential outputs are complementary which proves that the circuit is fault free.

In order to verify the circuit’s capability with realistic circuit defects, we simulate the adder in the presence of faults. Faults are voluntarily and manually injected into the physical layout of the circuit. In this case, the fault is injected in the primary input: \( \bar{a} \). The SPICE simulations are shown in Fig. 4.
In order to show the importance of the dual rail checker in the detection of faults, we simulate a two bit of the full adder of Fig. 4. Fig. 5 gives an example of these simulations.

Simulations show that when the fault is injected into the primary input (a), the duplicated outputs sum (Sum1/Sumb1 and Sum2/Sumb2) and carry (Cout1/Coutb1 and Cout2/Coutb2) do not remain complementary so that checkers (Sum checker and/or carry checker) indicate a non-valid code. In this case, the fault is detected by the outputs of the carry checker. We can see that fCout and fCoutb are not complementary and indicate a non-valid code.
IV. CONCLUSION

In this paper, a fault-tolerant full adder is proposed. The circuit is simulated using the double pass transistor logic. This technique involves 28.57% saving in transistor count compared to standard CMOS technology.

The presence of faults in the proposed design is detected using a double rail checker. In the presence of any fault a non-valid code word is provided as input to the checker yielding a non-valid output code word, hence the fault is detected.

REFERENCES


