# CHAPTER 4

# THE ADDER

The adder is one of the most critical components of a processor, as it is used in the Arithmetic Logic Unit (ALU), in the floating-point unit and for address generation in case of cache or memory access (John Rabaey (2003)).Increasing demand for mobile electronic devices such as cellular phones and laptop computers requires the use of power efficient VLSI circuits. The power consumption of a CMOS digital circuit is given in equation (3.1).

A new full adder called Static Energy Recovery Full(SERF) adder uses only 10 transistors shown in figure 4.9 which has the least number of transistors and is reported to be the best in power consumption (Shalem.R et al (1999)).Many low power adders using various pass transistors, such as the SERF(Shalem.R et al (1999)) when compared to the complementary static CMOS adders, have the problem of threshold loss,i.e., the logic value 1 is not the value of $V_{dd}$ and the logic value 0 may not be the value of 0.This kind of threshold loss logic gates may not be used as widely as the complementary static CMOS gates.However,they are certainly useful in building up larger circuits such as multiple-bit input adders and multipliers. Since the aim of this research work is to reduce the threshold loss problem which exists in earlier designs, a new approach to designing many 14 transistor full adders is proposed. However, the proposed

NEW adder cells are useful for designing larger circuits despite increase in transistor count by four per cell.

## 4.1 THE BINARY ADDER

The full adder operation can be stated as follows: Given the three 1-bit inputs A, B, and Cin, it is desired to calculate the two 1-bit outputs Sum and Carry, where

Sum = (A xor B) xor Cin (4.1)

Carry = A and B + Cin (A xor B) (4.2)

Table 4.1. Adder Truth Table

| INPUT | | | OUTPUT | |
|---|---|---|---|---|
| CIN | A | B | SUM | CARRY |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

In this research work, the following 8 different adders are designed.

1. 14 Transistor Full Adder
2. 20 Transistor Full Adder

3. 28 Transistor Full Adder
4. Conventional Full Adder
5. Transmission Functional Full Adder
6. Transmission Gate Full Adder
7. Static Energy Recovery Full Adder
8. NEW Full Adder

### 4.1.1 14 Transistor Full Adder

The 14 transistors full adder has 14 numbers of transistors to perform the full adder function. The structure is given in the figure 4.1. The 14T full adder cell implements the complementary pass logic to drive the load.

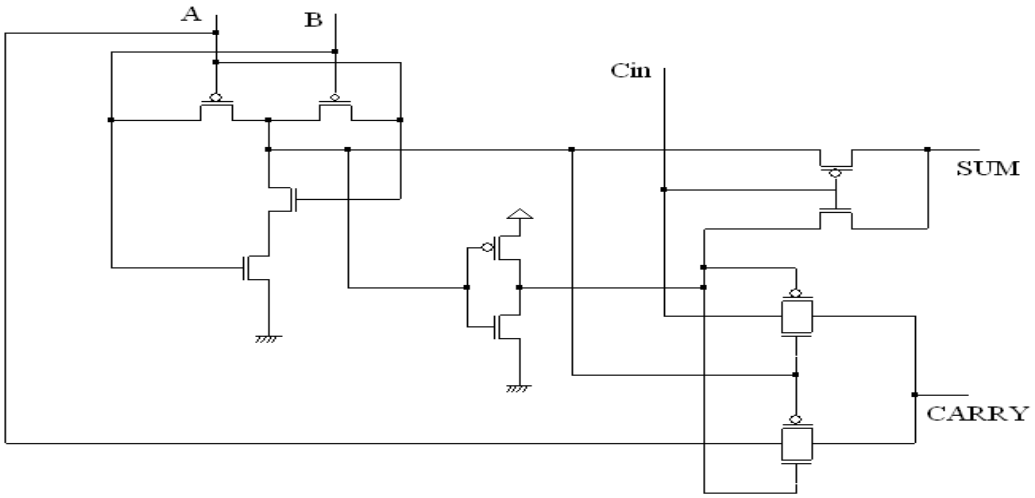


Figure 4.1 Structure of 14 Transistors Full Adder

### 4.1.2 20 Transistor Full Adder

The 20 transistors full adder has 20 numbers of transistors to perform the full adder function. The structure is given in the figure 4.2

### 4.1.3 28 Transistors Full Adder

One way to implement the full adder circuit is to take the logic equation (4.1) and equation (4.2) and translate them directly into complementary CMOS circuit. Some logic manipulations can help to reduce the transistor count. For instance, it is advantageous to share some logic between the sum and carry – generation sub circuits, as long as this does not slow down the carry generation, which is the most critical part as stated previously. The following is an example of such as reorganized equation set:

$$\text{CARRY} = A.B + B.C_{in} + A.C_{in} \quad (4.3)$$

$$\text{SUM} = A.B.C_{in} + \text{CARRY}\,(A + B + C_{in}) \quad (4.4)$$

The equivalence with the original equations is easily verified. The corresponding adder design, using complementary static CMOS, is shown in figure 4.3 and the gate level implementation is shown in figure 4.4. It requires 28 transistors. In addition to consuming a large area, this circuit is slow.

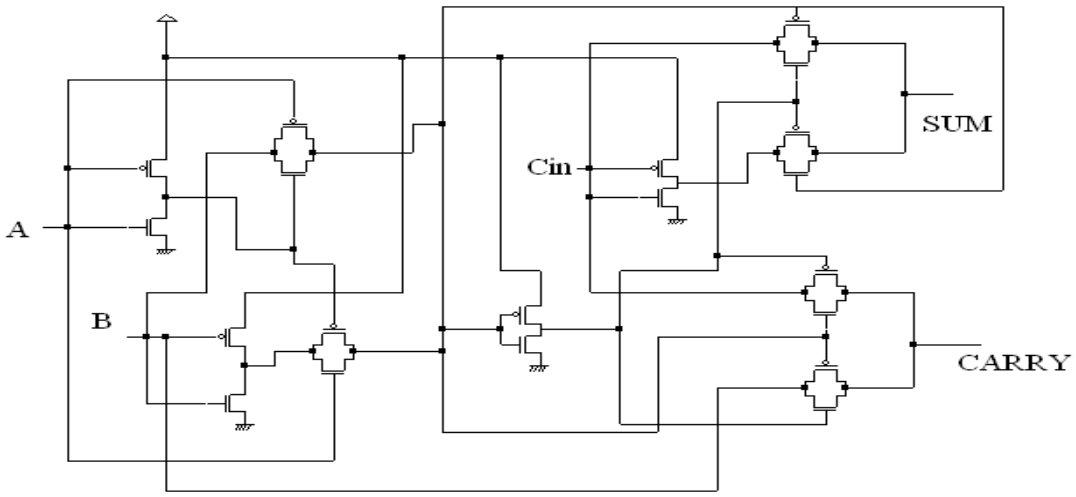


Figure 4.2 Structure of 20 Transistors Full Adder

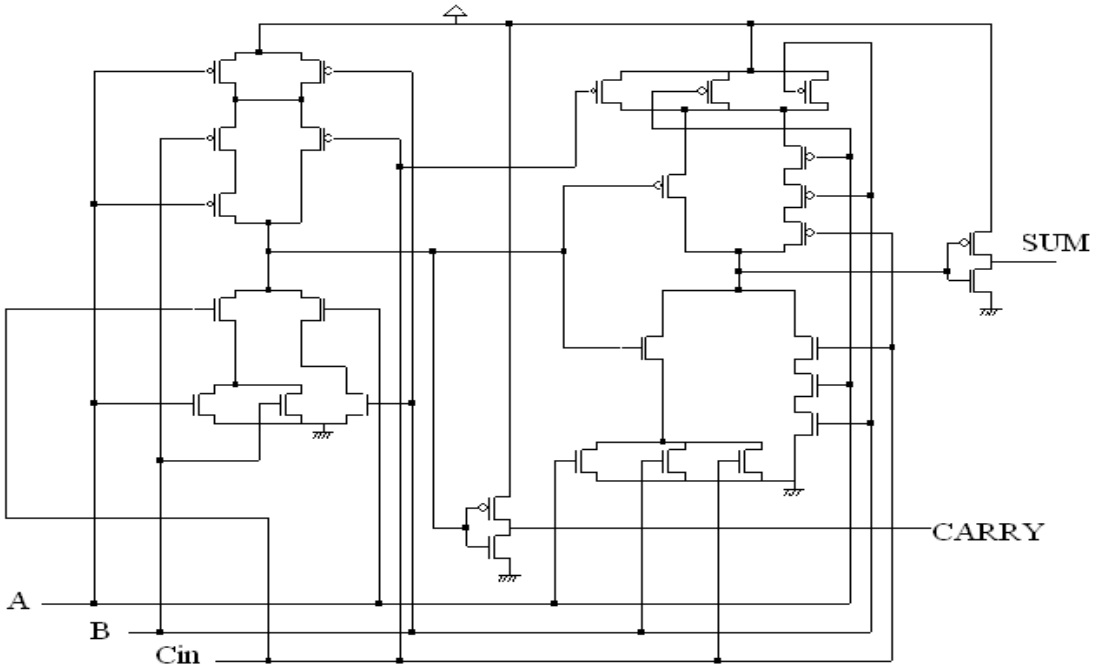


Figure 4.3 CMOS implementation of 28 Transistor full adder

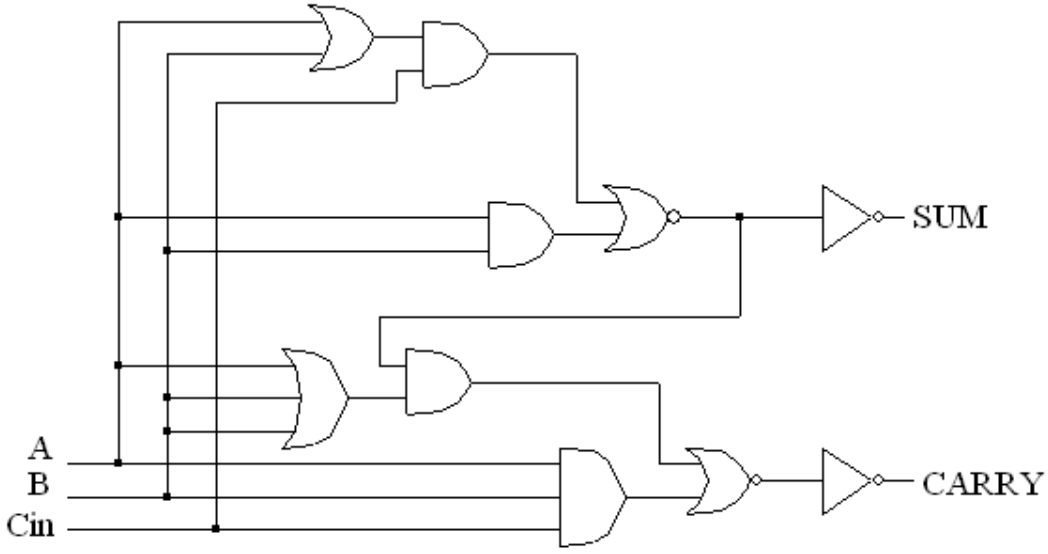


Figure 4.4 Gate level implementation of 28 Transistor full adder

### 4.1.4 Conventional Full Adder

Probably the simplest approach to designing an adder is to implement gates to yield the required majority logic functions. From the table [Table 2.1] these are:

$$SUM = A.B.C_{in} + A.B.C_{in} + A.B.C_{in} + A.B.C_{in} \quad (4.5)$$

Equation (2.3) may be factored as follows:

$$SUM = C_{in} (A.B+A.B) + C_{in} (A.B+A.B) \quad (4.6)$$

$$= A \text{ (XOR) } B \text{ (XOR) } C_{in} \quad (4.7)$$

$$CARRY = A.B + A.\ C_{in} + B.\ C_{in} \quad (4.8)$$

Equation (4.6) may be factored as follows:

$$CARRY = A.B + C_{in} (A+B) \quad (4.9)$$

The gate schematic for the direct implementation of equation (4.7) and equation (4.9) is shown in figure 4.5. This implementation uses a 3-input XOR

gate. A transistor level implementation is shown in figure 4.6. This uses a total of 32 transistors.

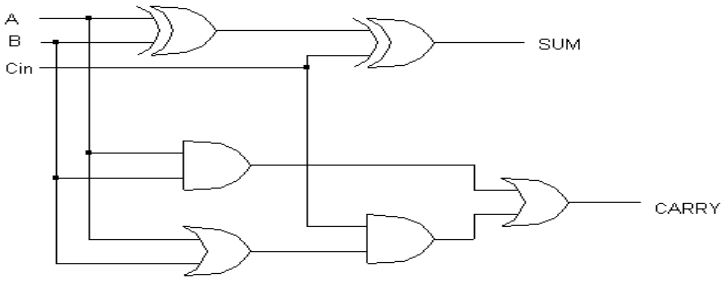


Figure 4.5 Gate schematic for conventional full adder

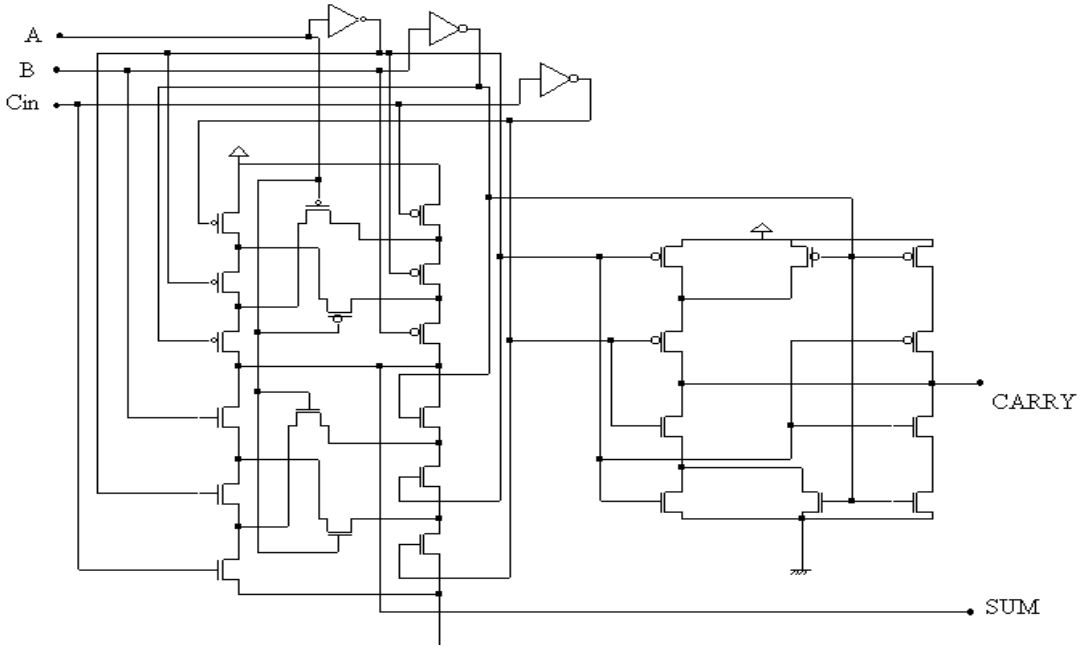


Figure 4.6 Transistor level schematic for conventional full adder

### 4.1.5 Transmission-Functional Full Adder

The transmission function full adder, which uses 16 transistors, for the realization of the circuit, is shown in figure 4.7. For this circuit there are two possible short circuits paths to ground. This design uses pull-up and pull-down logic as well as complementary pass Logic to drive the load. It gives the same delay for sum and carry.

### 4.1.6 Transmission-Gate Full Adder

A rather different implementation of an adder uses a novel exclusive –or (XOR) gate. The schematic for this XOR gate is shown in figure 4.8.It is to be noted that the switch-level simulators have problems with this gate. The operation of the gate is explained as follows:

- When signal A is high, -A is low. Transistor pair $P_1$ and $N_1$ thus acts as an inverter, with –B appearing at the output. The transmission gate formed by transistor pair $P_2$ and $N_2$ is open.
- When signal A is low, -A is high. The transmission gate ($P_2$, $N_2$) is now closed, passing B to the output. The inverter ($P_1$, $N_1$) is partially disabled (level reduced B passed to output by $P_1$, $N_1$).

Thus this transistor configuration forms a 6-transistor XOR gate. By reversing the connections of A and _A, and an exclusive-nor (XNOR) gate is constructed. By using four transmission gates, four inverters, and two XOR gates, an adder may be constructed according to the figure 4.7. A (XOR) B and the complement are formed using the Transmission Gate (TG) XOR gate shown in figure 4.8. The SUM (A (XOR) B (XOR) C) is formed by a multiplexer controlled by A (XOR) B (and complement). Examining the adder truth-table reveals that

CARRY= C when A (XOR) B is true. When A (XOR) B is false, CARRY=A (or B). This adder has 24 transistors.

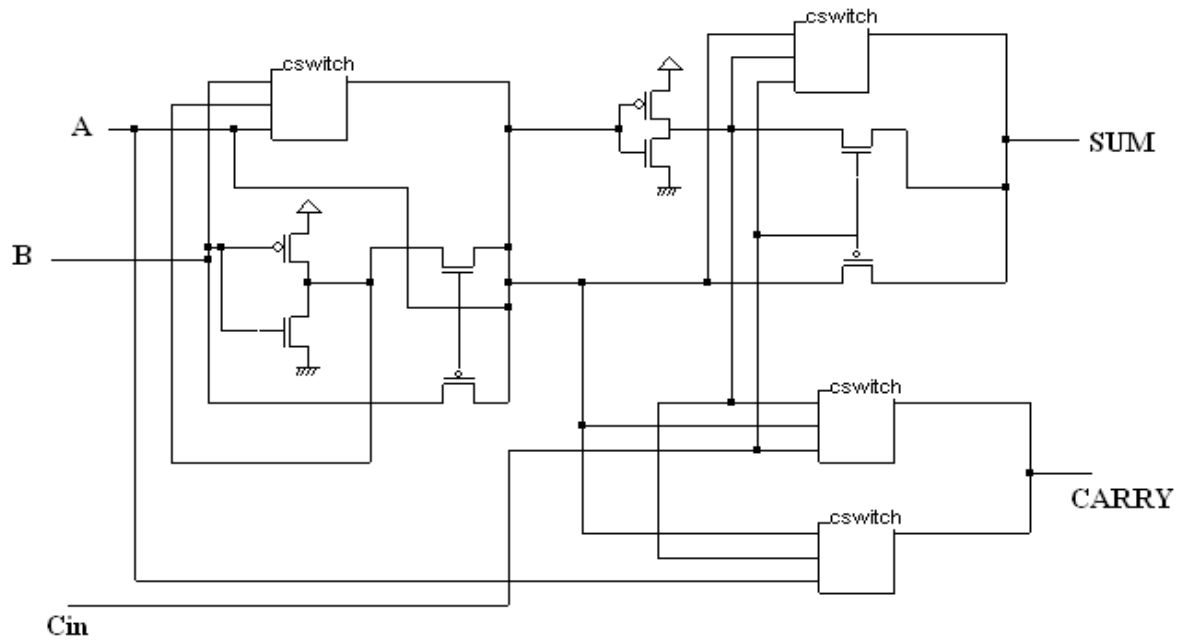


Figure 4.7 Structure of Transmission Functional Full Adder

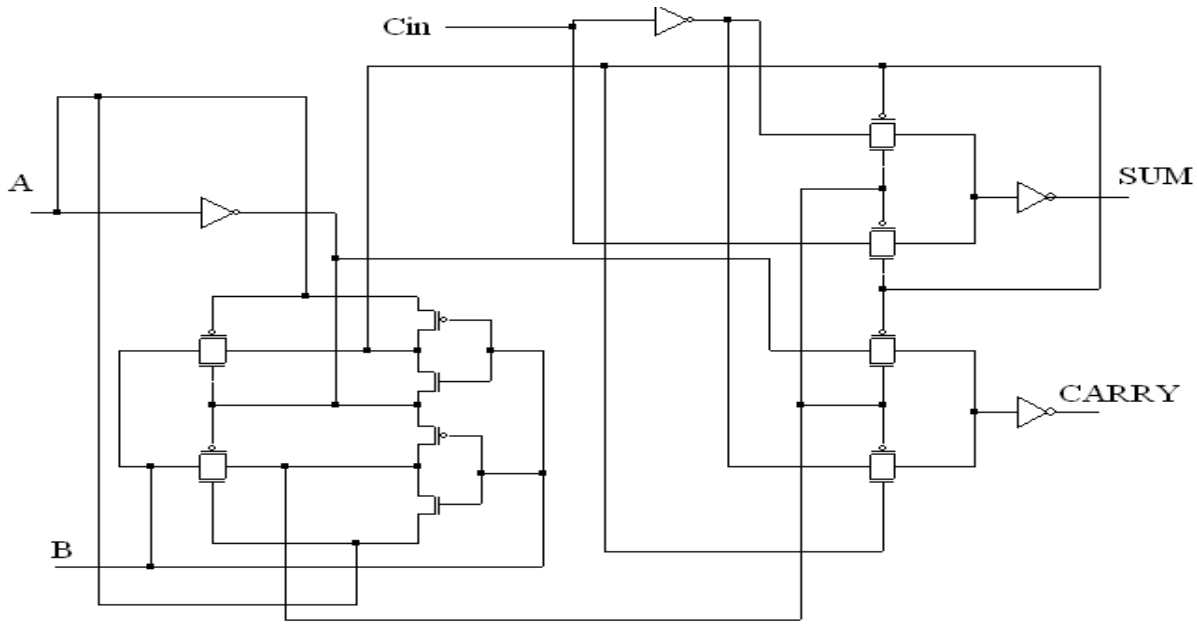


Figure 4.8 Structure of Transmission Gate Full Adder

### 4.1.7 Static Energy Recovery Full Adder

The Static Energy Recovery Full Adder (SERF) uses only 10 transistors to implement the full adder function. The design was inspired by the XNOR gate full adder design. The structure of SERF is given in figure 4.9.In non-energy recovery design the charge applied to the load capacitance during the logic level high is drained to ground during logic level low. It should be noted that the new SERF adder has no direct path to the ground. The elimination of a path to ground reduces power consumption, removing the short circuit from the power equation (3.1). The charge stored at the load capacitance is reapplied to the control gates, the combination of not having a direct path to ground and re-application of the load charge to the control gate makes the energy – recovering full adder an energy efficient design but it has the threshold loss problem (Shalem.R et al (1999))

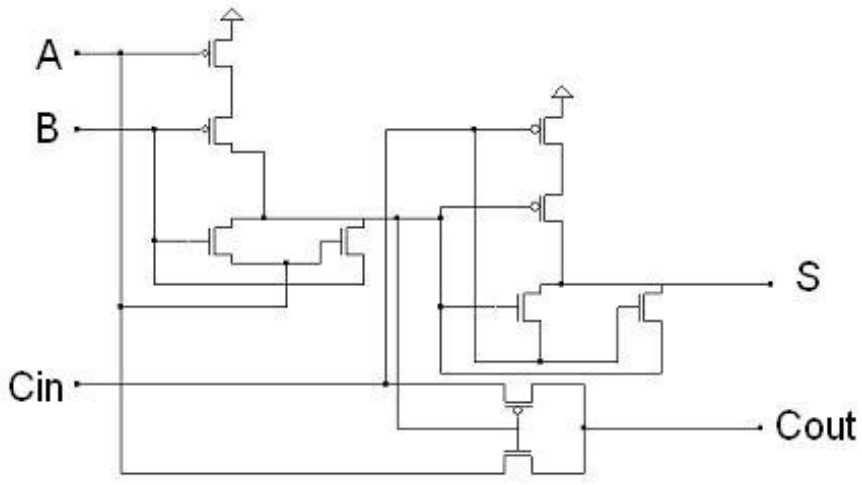


Figure 4.9 Structure of SERF Adder

### 4.1.8 NEW Full Adder

The new improved 14T adder cell requires only 14 transistors to realize the adder function shown in figure 4.10. It produces the better result in threshold loss, speed and power by sacrificing four extra transistors per adder cell. Even though the transistor count increases by four per adder cell, it reduces the threshold loss problem, which exists in the SERF by inserting the inverter between XOR Gate outputs to form XNOR gate.

The newly proposed adder implement the Sum using XNOR-XNOR and Cout using PMOS – NMOS We can also build to produce Cout using NMOS-NMOS and PMOS-PMOS. But the delay and power dissipation of PMOS-NMOS is better than other two kinds of producing Cout .The proposed XNOR gate is designed by putting inverter at the output of the XOR gate in order to improve the threshold loss problem, which exists in the SERF adder. Out of the three methods, PMOS-NMOS based Cout gives the better result in power, speed and threshold loss problem.

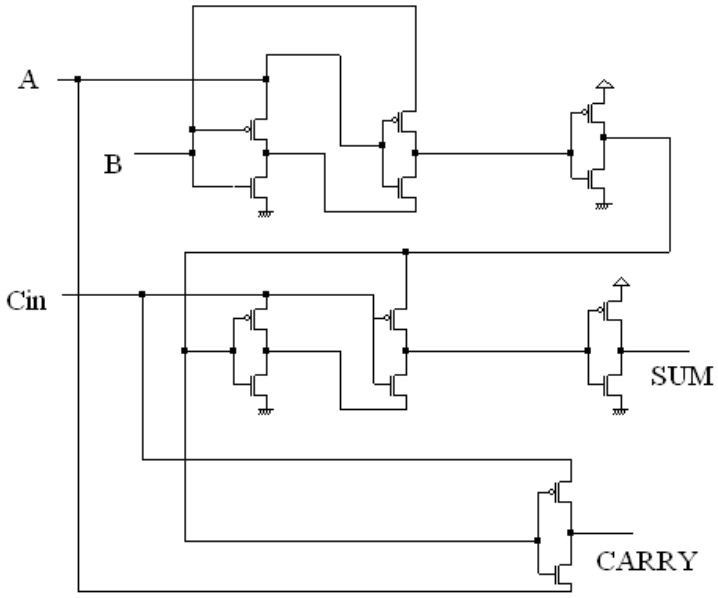


Figure 4.10 Structure of NEW Full Adder

Totally eight adders including the SERF adder are taken for comparison with the newly proposed adder. These adders are compared with respect to their power consumption and total delay by providing all the possible input vector combinations. The results proved that the newly proposed adder is efficient as it consumed the least power and eliminated the threshold loss problem. The present research work has presented a new improved 14T adder cell to construct full adders using only 14 transistors. Based on our extensive simulations, the new improved 14T adder cell consume considerably less power in the order of micro watts and has 48% higher speed and reduces 50% threshold loss problem compared to the previous different types of transistor adders.