CHAPTER 3
LOGIC TECHNIQUES FOR LOW TRANSISTOR COUNT FULL ADDER DESIGN

Using lower number of transistors to implement a logic function is beneficial in reducing the number of components, interconnect parasitic and reducing the chip area, resulting in lower time delay and potentially lower power consumption. However, many low count transistor adders do not operate correctly at low supply voltage in 0.18\textmu m and subsequent CMOS technologies due to threshold loss problem. Therefore, the prime goal now-a-days is to design the low number of transistors full adder that works successfully at low supply voltage, reduces degradation on the output voltage, has less delay in critical path and consumes less power. Numerous adder implementations using different techniques exist in literature whereas some of them are good for low power consumption and some takes least propagation delay. The Sum and Carry outputs [2], [4] of a 1-bit full adder generated from the binary inputs A, B, C\textsubscript{in} in its conventional form can be expressed as:

\begin{align*}
\text{Sum} &= A \oplus B \oplus C\text{in} \quad (3.1) \\
\text{Cout} &= A.B + C\text{in}. (A \oplus B) \quad (3.2)
\end{align*}

These outputs can be expressed in many different logic expressions and, thereby, determine the structure of the circuit. Based upon these different logic expressions, many full adders were implemented. In this chapter, three different 1-bit adder architectures depending upon their structure and logical expression are designed.

The basic architecture of 8T full adder [20], [22] which consists of 3 modules, is shown in Fig.3.1 and different design methodologies for C\textsubscript{out} module are shown in Fig. 3.2. Module-1 and Module-2 can be XOR or XNOR gates and Module-C\textsubscript{out} can be either multiplexer, double pMOS or double nMOS transistors. Thus, in this way a total of six designs are possible. The Sum is generated by cascading Module-1 and Module-2.
Among all the designs of $C_{out}$ module, multiplexer gives best performance as it carries the advantage of pMOS and nMOS both, therefore we have used multiplexer design for Carry output in all circuits. Table I shows the various design methodologies of 8T full adders which results into six different designs of 1-bit full adder cells all having a delay of 2T.
In 8T adder, the Sum output is based on implementing XOR operation between all inputs A, B and Cᵢⁿ and Carry output is dependent on XORing between input A and B. Thus, Sum and Carry output of 8T adder cell can be expressed as:

\[
\text{Sum} = A \oplus B \oplus Cᵢⁿ \quad (3.3)
\]

When \( A \oplus B = 0 \),
\[
C_{out} = A,
\]

When \( A \oplus B = 1 \),
\[
C_{out} = Cᵢⁿ, \quad (3.4)
\]

Fig. 3.3 shows architecture of 9T full adder based on carry logic. It consists of 4 modules:

- Module-1: 3T XOR gate,
- Module-2: A CMOS inverter to generate \( C_{out} \),
- Module-3: 2T multiplexer for Sum output and
- Module-Cout again using 2T multiplexer

<table>
<thead>
<tr>
<th>Adder</th>
<th>Module-1</th>
<th>Module-2</th>
<th>Module-Cout</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3-T XOR</td>
<td>3-T XOR</td>
<td>Multiplexer</td>
</tr>
<tr>
<td>2</td>
<td>3-T XOR</td>
<td>3-T XOR</td>
<td>Double pMOS</td>
</tr>
<tr>
<td>3</td>
<td>3-T XOR</td>
<td>3-T XOR</td>
<td>Double nMOS</td>
</tr>
<tr>
<td>4</td>
<td>3-T XNOR</td>
<td>3-T XNOR</td>
<td>Multiplexer</td>
</tr>
<tr>
<td>5</td>
<td>3-T XNOR</td>
<td>3-T XNOR</td>
<td>Double pMOS</td>
</tr>
<tr>
<td>6</td>
<td>3-T XNOR</td>
<td>3-T XNOR</td>
<td>Double nMOS</td>
</tr>
</tbody>
</table>
The name of this architecture implies that it uses only input and output carry signals to generate Sum output. The Sum output is based on implementing XOR operation between inputs A and B and transferring either $C_{out}$ or $C_{in}$ by 2T multiplexer. $C_{out}$ is implemented using another 2T multiplexer which is controlled by output of XOR gate. Thus, Sum and Carry output of an adder cell can be calculated as:

When $A \oplus B = 0$,
\[
\text{Sum} = C_{in}; \quad C_{out} = A
\]  \hspace{1cm} (3.5)

When $A \oplus B = 1$,
\[
\text{Sum} = C_{out}; \quad C_{out} = C_{in}
\]  \hspace{1cm} (3.6)

In the similar manner, another 9T full adder can be designed using inversion logic. Fig. 3.4 shows the architecture for the same. It also consists of 4 modules that are:

- Module-1: 3T XOR gate,
- Module-2: A CMOS inverter to generate XNOR function,
- Module-3: 2T multiplexer for generating Sum output, and
- Module-Cout again using 2T multiplexer
Fig. 3.4 Architecture of 9T 1-bit Full Adder based on inversion logic

Here, the Sum output is based on implementing XOR and XNOR operation between inputs B and Cin. An inverter is connected at the output of XOR gate to generate XNOR function. The Sum is implemented by transferring these levels through 2T multiplexer which is controlled by input A. For A=0, it passes $B \oplus C_{in}$ else, it passes $B \odot C_{in}$. $C_{out}$ is implemented by using another 2T multiplexer which is controlled by output of XOR gate. Thus, Sum and Carry output of the proposed adder cell can express as:

When $A = 0$,  
\[ \text{Sum} = B \oplus C_{in} \]

When $A = 1$,  
\[ \text{Sum} = B \odot C_{in} \quad (3.7) \]

When $B \oplus C_{in} = 0$,  
\[ C_{out} = A; \]

When $B \oplus C_{in} = 1$,  
\[ C_{out} = C_{in} \quad (3.8) \]

Briefly, the above equations and discussions for the low transistor count full adder design techniques give the choice to designer to select the adder cell depending upon the application and specifications of the system so that overall performance of the system improves with no throughput penalty.