Area efficient Full Subtractor design using CMOS Technology

1Monikashree T.S, 2Divya A, 3Kesthara V, 4Nithya Shree S
1Assistant Professor, Department of ITE, Dr.AIT Bangalore, Karnataka, India
2,3,4 Assistant Professor, Department of ECE, Dr.AIT Bangalore, Karnataka, India
Email: 1monikashree.ts@gmail.com

Abstract — Full Subtractor is a combinational digital circuit that performs 1-bit subtraction with borrow-in. The main objective of this project is to design a 1-bit Full Subtractor by using CMOS180nm technology with reduced number of transistors, and hence it efficiency in area, speed and power consumption. Two types of simulation or test bench will be performed in order to ensure that the implementation is fully functional. First, a schematic simulation will be performed by means of the “Cadence Schematic Editor and Analog Environment” software. Second, the 1-bit subtrator layout model will be emulated by the “Cadence Virtuoso Editor”. Performing the simulation will mainly consist in evaluating the quality of the output signals in terms of voltage levels, to assess the performance of the circuit in terms of speed, area and power usage.

Keywords: Cadence, 1-bit Half Subtractor, 1-bit full subtractor, logic gate, Virtuoso,

I. INTRODUCTION

Arithmetic circuits are important part of Digital circuits. In the digital, circuits subtractor is one of the most critical components used in the processor of portable devices [3]. Hence the area and power efficient design of 1-bit Subtractor is necessary for design of small size portable devices. There are various possible logic styles that can give better performance as compared to the basic CMOS logic style. The performance estimation of 1-bit full Subtractor is based on area, delay and power consumption. The purpose of this work is [5],[6].

1. To perform the design, full custom implementation and simulation of a 1-bit subtractor at the transistor level by means of CMOS180nm technology. Subtractor at the transistor level by means of CMOS 180nm technology.
2. To verify if the circuit can perform with all the possible combinations of the inputs alongside the logic function which it is designed for.
3. To evaluate the quality of the output signals in terms of voltage levels.
4. To assess the performance of the circuit in terms of speed, area

In the recent, various approaches of CMOS 1- Bit full Subtractor design by using various different logic styles has presented and unified into an integrated design methodology. The Conventional 1 bit full subtractor diagram is shown in fig1 and its truth table in Table 1. The number of logic gates required are more, it leads in increasing number of transistors, hence the area, delay will be large. The need in optimising 1 bit full subtractor using cadence is that, the CMOS gpdk180nm package incorporates the “Cadence Schematic Editor and Analog Environment” software used to create a schematic diagram and a simulation of our implementation. Moreover, it contains the Cadence Virtuoso Editor which will allows us to design the layout of the 1-bit subtractor, as well as to perform the assessment of several performance parameters for the circuit. In addition, transient analysis will be performed. The proposed specifications offered a wide selection varying between two types of implementations, each with an analogous kind of difficulty. We chose the design of a 1-bit subtractor in order to minimize while trying to maintain the area as much as possible.

II. SUBTRACTOR

A subtractor performs subtraction which is one of the four basic binary operations. In many computers and other kinds of device processors, subtractors are used not only for the arithmetic calculations, but are also frequently used in other parts of the processor. The subtractors can be constructed to operate on binary numbers and represents many binary numbers.

Depending upon the application of the device or upon the purpose of the application to be performed, the inputs to the circuit device may vary from two to three. We could possibly use a Half-Subtractor if we have two inputs while for three inputs, a Full-Subtractor can be used.

2.1 1 Bit Half Subtractor

A conventional Half-subtractor circuit is a combinational circuit that can be used to subtract one binary digit from another to produce a Difference output and a Borrow output. Functionally, the half subtractor consists of single 2 inputs XOR Gate, INVERTOR Gate and single 2 inputs AND gate.

The Borrow output here specifies whether a ‘1’ has been borrowed to perform the subtraction. The Half-Subtractor at the gate-level and truth table are shown in Fig 1 and table 1.
The truth table and boolean expression for the two output variables are given by the equations:

\[ \text{Diff} = A'B + AB' \]

\[ \text{Bout} = AB \]

### 2.2 1 Bit Full Subtractor

A 1-bit full subtractor is a combinational circuit that performs a subtraction between two binary bits and 1 may have been borrowed by a lower significant stage. This circuit has three inputs and two outputs. Let the three inputs are A, B and Bin and Borrow and Difference are two outputs of the 1-bit Subtractor and denoted by Bout and Diff respectively. Logic diagram of 1-bit full subtractor has been shown in fig.2.

![Fig. 2 Logic Diagram of conventional 1 bit Full Subtractor](image)

In subtraction process on two bits, a minuend and a subtrahend, and also takes into consideration whether a ‘1’ has been borrowed by the previous adjacent lower minuend bit or not. As a result, there are three bits to be handled at the input of a Full-Subtractor, namely the two bits to be subtracted and a borrow bit designated as Bin. There are two outputs, namely the Difference output D and the Borrow output Bout. The Borrow output bit tells whether the minuend bit needs to borrow a ‘1’ from the next possible higher minuend bit.

The truth table and boolean expression for the two output variables are given by the equations:

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>Bin</th>
<th>Bout</th>
<th>Diff</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0=0, No borrow</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0-1=-1, borrow 2, so: 2-1 = 1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1-0=1, No borrow</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1-0=1, No borrow</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0-1=-1, borrow 2, so: 2-1 = 1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1-0=0, No borrow</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1-1=-1, borrow 2, so: 2-1 = 1</td>
</tr>
</tbody>
</table>

Table 2: Truth Table of 1 Bit Full Subtractor

\[ \text{Diff} = A \oplus B \oplus \text{Bin} \]

\[ \text{Bout}= AB \overline{B} \text{in} + \overline{A} \overline{B} \text{in} \]

III. PROPOSED 1 BIT FULL SUBTRACTOR DESIGN

Here the primary goal is to reduce the area and the transistor count in order to reduce area, power, delay parameters. The proposed full subtractor implemented by cascading two 1-bit half subtractor. In order to build reduced transistor 1-bit subtractor, the idea has to begin with the basic gates. Each individual logic gates used in the circuit is constructed by minimum no transistors so as to reduce the circuit area, the performance criteria’s of each gates are individually investigated, analyzed and their interaction to develop main design.

3.1 Logic Gates Design

The implementation consists of four types of logic components such as such as inverters, 2-input AND gates, 2-input OR gates and 2-input XOR gate.

3.1.1 Inverter

An inverter gate usually output signals either representing a “true” or “false”. In this case, the inverter’s output OUT is true when the input IN is “false”, OUT=IN, where IN is the inverted input and OUT the output. Moreover, the truth table below shows the function of the inverter logic gate:

<table>
<thead>
<tr>
<th>Input IN</th>
<th>Output OUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 3: Truth Table of an Inverter

3.1.2 2 Inputs AND Gate

A 2-input AND gate usually output signals as “true” when inputs A and B are both “true”. Therefore, we can write the 2-input AND logic equation which is simply provided by the following equation below:

\[ \text{Vout} = A \times B \]
### 3.1.3 2 Inputs OR Gate

A 2-input OR gate usually output signals as “true” when any one inputs A, B or both are “true”. Therefore, we can write the 2-input AND logic equation which is simply provided by the following equation below:

\[ V_{out} = A + B \]

<table>
<thead>
<tr>
<th>Input A</th>
<th>Input B</th>
<th>Output vout</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

#### Table 5: Truth Table of 2 inputs OR Gate

### 3.1.4 2 Inputs XOR Gate

A 2-input XOR gate usually output signals as “true” when any one inputs A or B are “true”, else false. Therefore, we can write the 2-input AND logic equation which is simply provided by the following equation below:

\[ V_{out} = \overline{A} \cdot B + A \cdot \overline{B} \]

<table>
<thead>
<tr>
<th>Input A</th>
<th>Input B</th>
<th>Output vout</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

#### Table 6: Truth Table of 2 inputs XOR Gate

The schematic of all the Logic gates implemented using CMOS 180nm technology as shown in fig. below:

![Schematics](image)

Fig 3: Schematics of (a) Inverter, (b) 2 Inputs AND Gate, (c) 2 Inputs OR Gate, (d) 2 Inputs XOR Gate

### 3.2 Subtractor Design

In this section we presented the transistor level implementation of both 1 bit Half Subtractor and 1 bit Full Subtractor. The pmos and nmos transistors used to construct the subtractor are has their own specification, and those are not altered.

#### 3.2.1 1 Bit Half Subtractor

The Schematic of 1 Bit Half subtractor with one XOR, Inverter, AND are implemented at transistor level using Cadence Tool is shown in below fig.

![Schematic](image)

Fig 4: Schematic of transistor level 1 bit Half Subtractor

#### 3.2.2 1 Bit Full Subtractor

Here we presented both conventional and proposed full subtractor to show the reduced transistor count. Both are implemented in same cadence tool.

![Schematic](image)

Fig 5: Schematic of transistor level 1 bit Conventional Full Subtractor

![Schematic](image)

Fig 6: Schematic of transistor level proposed 1 Bit Full Subtractor
IV. LAYOUTS AND SIMULATION RESULTS

4.1 Layouts

The layout of all Basic logic gates are designed using CMOS 180nm technology as shown in fig below. These layouts help as a reference model to construct a whole full subtractor layout.

The implemented layouts of all basic gates used to construct subtractor in cadence layout window are:

![Layouts of Basic Logic Gates](image)

Fig 7: Layouts of (a) Inverter, (b) 2 inputs AND Gate, (c) 2 inputs OR Gates, (d) 2 inputs XOR Gate

The implemented layouts of 1-bit half subtractor and 1 bit full subtractor in cadence layout window are:

![Layout of 1 bit half subtractor](image)

Fig 8: Layout of 1 bit half subtractor

![Layout of proposed 1 bit full subtractor](image)

Fig 9: Layout of proposed 1 bit full subtractor

4.1 Simulation Results

The implementation of the 1-bit subtractor circuit will be performed progressively by implementing and creating instances of the components independently. Subsequently, we will merge all the components together to create the 1-bit subtractor. At last, we will obtain the 4-bit subtractor by cascading four instances of the 1-bit subtractor created.

This section will be broken into two main parts. First, we will perform the step by step implementation and simulation of the 1-bit subtractor by means of the “Cadence Schematic Editor and Analog Environment”. Secondly, we will achieve the same implementation by designing gradually the layout of the 4-bit subtractor by means of the “Cadence Virtuoso Editor”. Finally, we will record the transient and experimental results for both simulations.

The simulated results are:

![Transient responses of Basic Logic Gates](image)

Fig 10: Transient responses of (a) Inverter, (b) 2 inputs AND Gate, (c) 2 inputs OR Gates, (d) 2 inputs XOR Gate

The resulted output wave forms of the 1 bit half subtractor:
subtractor and 1 bit full subtractor is shown below:

![Image](image1)

**Fig 11: Transient responses of 1 bit half subtractor**

![Image](image2)

**Fig 12: Transient responses of proposed 1 bit full subtractor**

**V. CONCLUSION**

Overall, in this experiment different skills and techniques were gained and applied in a fundamental part. Indeed, one can able to design, implement and successfully analyze the characteristics of a 1-bit subtractor circuit. The completion of this main task was entirely satisfactory since the theoretical expectations matched our experimental results. The performance of the 4-bit subtractor was assessed in terms of area, speed, also evaluated the quality of the output signals by comparing the timing measurements of each circuit by means of the propagation delays.

The total delay and power utilized in this project is presented in the below table:

<table>
<thead>
<tr>
<th>Logic Gate</th>
<th>Number of Transistors</th>
<th>Delay</th>
<th>Power Consumed (micro watt)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inverter</td>
<td>2</td>
<td>138ps</td>
<td>4.779</td>
</tr>
<tr>
<td>2 input AND</td>
<td>6</td>
<td>21.42ps</td>
<td>5.703</td>
</tr>
<tr>
<td>2 input OR</td>
<td>6</td>
<td>20.1ps</td>
<td>3.08</td>
</tr>
<tr>
<td>2 input XOR</td>
<td>6</td>
<td>20.4ns</td>
<td>6.392</td>
</tr>
<tr>
<td>1 bit half subtractor</td>
<td>14</td>
<td>124ps</td>
<td>17.91</td>
</tr>
<tr>
<td>1 bit Full Subtractor</td>
<td>34</td>
<td>250ps</td>
<td>47.05</td>
</tr>
<tr>
<td>1 bit Full Subtractor (conventional)</td>
<td>40</td>
<td>235ps</td>
<td>47.99</td>
</tr>
</tbody>
</table>

**Table 7: Summary of Delay and Power**

**REFERENCES**


