Abstract— In this paper, we propose Kogge-Stone and Brent-Kung parallel prefix adders based on degenerate pass transistor logic (PTL). Threshold loss problem are the main drawback in most pass transistor logic family. This threshold loss problem can be minimized by using the complementary control signals. These complementary control signals are obtained by 5-Transistor XOR-XNOR module. By using these complementary outputs we designed parallel prefix adders based on 10-Transistor full adder. Parallel prefix adders are used to speed up the binary addition and these adders are more flexible to perform addition of higher order bits in complex circuits. The transistor level implementation of parallel prefix adders based on degenerate PTL gives better performance compared to CPL and DPL pass transistor logic.

Keywords— Power Dissipation, degenerate, complexity, Threshold loss.

I. INTRODUCTION

Low power very large scale integration (VLSI) circuits are important for designing of high performance and portable devices. The high speed, small area and low cost are the main considerations of VLSI circuits. The importance of low power is increasing day by day because of changing trend, packaging and cooling cost, portable systems and reliability. In nanometre technology power has become the most important issue because of increasing transistor count, higher speed of operation and greater device leakage currents. Packaging and cooling cost is dependent on the power dissipation of the chip. Now a days there is a more demand on hand held or battery devices like cell phones, laptops and palmtops etc.. As these devices are battery operated, battery life is primary concern. The reliability is mainly depends on the failure rates of the chip. These failure rates are due to changes in the temperature.

Every 10°C raise in temperature roughly double the failure rates. Reliability point of view low power design methodology is extremely important. The arithmetic circuits [2] are designed at transistor level based on different logic design styles. Those are conventional CMOS [3] logic; pass transistor logic, adiabatic logic and gate diffusion input. The details of pass transistor logic families are discussed in section II, and implementation of prefix adders are described in section III. The performance and results are discussed in section IV.

II. PASS TRANSISTOR LOGIC

Basically transistor is used as a switch. To realize the Boolean functions by using transistor as a switch is typically known as pass transistor logic. The advantages of pass transistor [10] based realization are ratio less, lower area due to smaller number of transistors and lesser power dissipation because of no static power and short circuit power dissipation. Some of the disadvantages of pass transistor logic are higher delay and multi threshold voltage drop. Pass transistor logic family [11] is used to overcome the limitations of pass transistor circuits. Some of the techniques are

1. Insertion of buffers to avoid long delay of a chain of pass transistor logic.
2. Use of swing restoration circuit to overcome multi-threshold voltage drop ($V_{out} = V_{dd}-V_{th}$).
3. Use of dual rail logic to generate complementary control signals.

Complementary Pass Transistor Logic (CPL)

The complementary means it has both outputs i.e. f and f'. In this logic two networks are used to realizing both f and f'. At the output side we use two inverting buffers and two weak pMOS transistors. These inverting buffers are used to perform restoration logic levels at the output. The pMOS latches are used to perform swing restoration. The basic AND/NAND, OR/NOR and XOR/XNOR [8] gates implemented in CPL are shown in Figure 1.

![Figure 1. The basic AND/NAND, OR/NOR and XOR/XNOR gates implemented in CPL.](image-url)

Double Pass Transistor Logic (DPL)

In double pass transistor [12] logic we are using both nMOS and pMOS transistors are used to design a circuit. Adding of pMOS transistor in parallel with nMOS will produce full swing at the output. The DPL configuration is different from static CMOS logic. The DPL is the modified version of CPL. In this logic extra transistor is not required for swing restoration. The basic AND/NAND, OR/NOR and XOR/...
XNOR [9] gates implemented in DPL are shown in Figure 2.

![XNOR Gates](image1)

Figure 2. The basic AND/NAND, OR/NOR and XOR/XNOR gates implemented in DPL.

**Degenerate Pass Transistor Logic**

Most of full adder [4] designs are based on three module implementation i.e. XOR, sum, carry modules. The 4-Transistor XOR (XNOR) module [5] is used to design a pass transistor logic based full adder [13]. But this type of design raises severe threshold voltage loss problems. The signal degradation in the output causes signal degradation in the further modules. Finally the signal degradation in the circuit causes a multi-threshold loss problem at the output stage. These multi-threshold loss problems can be reduced by using complementary outputs. The complementary outputs are obtained by using 5-Transistor XOR-XNOR design. The design of degenerate 5-T XOR-XNOR module is shown in Figure 3.

![XOR-XNOR Designs](image2)

Figure 3. XOR-XNOR designs using 5 transistors.

With the availability of complementary control signals the selection of sum and carry modules [1] are more flexible to avoid the multi-threshold loss problems, the signal degradation in the full adder output is minimized and it is suitable for low V_{DD} applications. The design of degenerate 10-T PTL full adder design is shown in Figure 4.

![10-T Full Adder Design](image3)

Figure 4. 10-T full adder design using degenerate PTL.

**III. PROPOSED ADDERS**

**Parallel prefix adders**

The parallel prefix adders [11] are more flexible and are used to speed up the binary additions. Parallel prefix adders [14] are obtained from Carry Look Ahead (CLA) structure. We use tree structure form to increase the speed [13] of arithmetic operation. Parallel prefix adders are fastest adders [4] and these are used for high performance arithmetic circuits in industries. The construction of parallel prefix adder [4] involves three stages

1. Pre-processing stage
2. Carry generation network
3. Post processing

**Pre-processing stage**

In this stage we compute, generate and propagate signals to each pair of inputs A and B. These signals are given by the logic equations 1&2:

\[
P_i = A_i \oplus B_i \quad \text{(1)}
\]

\[
G_i = A_i \land B_i \quad \text{(2)}
\]

**Carry generation network**

In this stage we compute carries corresponding to each bit. Execution of these operations is carried out in parallel [4]. After the computation of carries in parallel they are segmented into smaller pieces. It uses carry propagate and generate as intermediate signals which are given by the logic equations 3&4:

\[
CP_{i:j} = P_{i:k+1} \land P_{k:j} \quad \text{(3)}
\]

\[
CG_{i:j} = G_{i:k+1} \lor (P_{i:k+1} \land G_{k:j}) \quad \text{(4)}
\]

**Post processing**

This is the final step to compute the summation of input bits. It is common for all adders and the sum bits are computed by logic equation 5&6:

\[
C_{i-1} = (P_i \land C_{in}) \lor G_i \quad \text{(5)}
\]

\[
G_i = \text{common divisor of } (P_i, G_i) \quad \text{(6)}
\]
\[ S_i = p_i \oplus c_{i-1} \] ...................... (5)

Types of prefix adders are classified into
1. Ladner-Fischer prefix adder
2. Beaumont-Smith prefix adder

**Ladner-Fischer prefix adder**

Ladner-Fischer adder is a parallel prefix form carry look-ahead adder. A parallel prefix adder can be represented as a parallel prefix graph consisting of carry operator nodes. The time required to generate carry signals in this prefix adder is \( o(\log n) \). It is a fastest adder design and common design for high performance adders in industry. The Ladner-Fischer [14] adder concept was developed by R. Ladner and M. Fischer, which they published in 1980. The better performances of Ladner-Fischer adder are minimum logic depth and bounded fan-out. But it has large area. The block diagram of 4-bit Ladner-Fischer adder is shown in Figure 6.

![Figure 6. 4-bit Ladner-Fischer adder.](image)

Transistor level implementation of SUM block and carry operator are shown in Figure 7&8.

![Figure 7. Transistor level implementation of carry operator.](image)

![Figure 8. Transistor level implementation of SUM block.](image)

**Beaumont-Smith prefix adder**

The Beaumont-Smith adder is a parallel prefix adder. Parallel prefix adders are special class of adders that are based on the use of generate and propagate signals. Simpler Beaumont-Smith adders was been proposed to solve the disadvantages of Ladner-Fischer adders. The cost and wiring complexity is greatly reduced. But the logic depth of Beaumont-Smith adders [15] increases to \( 2\log (2n-1) \), so the speed is lower. The block diagram of 4-bit Beaumont-Smith adder is shown in Figure 9.

![Figure 9. 4-bit Beaumont-Smith adder.](image)

**IV. SIMULATION RESULTS AND COMPARISONS**

The various adders are simulated using SPICE code in Mentor Graphics tool. Adders are designed by using Design Architect in Mentor Graphics tool. The performance of proposed prefix adders in degenerate pass transistor logic are analyzed and compared. The prefix adder architecture is based on the implementation of propagate, generate and carry operator. Simulation results of Ladner-Fischer adder and Beaumont-Smith adder are compared in terms of DC power dissipation, memory size, number of components and number of steps computed.

**Table 1. Comparison between Ladner-Fischer and Beaumont-Smith adders.**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Ladner-Fischer</th>
<th>Beaumont-Smith</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dc power dissipation(mw)</td>
<td>( 3.2356 \times 10^{-03} )</td>
<td>( 2.6422 \times 10^{-03} )</td>
</tr>
<tr>
<td>Memory size(bytes)</td>
<td>43524096</td>
<td>43511808</td>
</tr>
<tr>
<td>No. components</td>
<td>74</td>
<td>68</td>
</tr>
<tr>
<td>No. Of steps computed</td>
<td>1005</td>
<td>1009</td>
</tr>
</tbody>
</table>

**Table 2. Simulation results of degenerate full adder**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Degenerate FA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dc power dissipation(mw)</td>
<td>0.0</td>
</tr>
<tr>
<td>Memory size(bytes)</td>
<td>43294720</td>
</tr>
<tr>
<td>No. components</td>
<td>13</td>
</tr>
<tr>
<td>No. Of steps computed</td>
<td>334</td>
</tr>
</tbody>
</table>
V. CONCLUSION

New approach of Degenerate pass transistor logic is used in the design of prefix adders to reduce the Threshold loss problem, power dissipation and signal degradation at the output. The transistor level implementation of prefix adders is simple, lesser power dissipation and high speed compared to other pass transistor logics. The tradeoffs of prefix adders are in terms of power dissipation and area. To design Beaumont-Smith adder it requires less number of carry operators. So reduction in carry operator reduces tradeoffs of Beaumont-Smith adder compared to Ladner-Fischer adder. The simulated result shows that the proposed prefix adder greatly reduces in power dissipation as well as area.

ACKNOWLEDGEMENT

S. Adilakshmi would like to thank Ms. M. Bharathi, Assistant professor ECE Department who had been guiding throughout the project and supporting me in giving technical ideas about the paper and motivating me to complete the work efficiently and successfully.

REFERENCES


AUTHORS PROFILE


Ms. M. Bharathi, M.Tech., is currently working as an Assistant Professor in ECE department of Sree Vidyanikethan Engineering College, Tirupati. She has completed M.Tech in VLSI Design, in Satyabhama University. Her research areas are Digital System Design, VLSI Signal Processing.