Implementation of Reversible Logic Based Carry Tree Adders

P. Vasavi¹, V. Madhurima²

¹Mtech. Student, Department of ECE, SV College of Engineering, Tirupati, AP, India
Email: vasavipulasetty455@gmail.com
²Assistant Professor, SV College of Engineering, Tirupati, AP, India
Email: madhurimaV16@gmail.com

Abstract—In Very Large Scale Integration (VLSI) designs, Parallel prefix adders (PPA) have the better delay performance. This paper presents three types of PPA's Reversible Kogge Stone Adder (RKSA), Reversible Spanning Tree Adder (RSTA), and Reversible Brent Kung Adder (RBKA). These adders are implemented using Verilog Hardware Description Language (HDL) in Xilinx 14.3 for simulation and synthesis results. The parallel prefix-adder’s Delay, Power and Area are optimized compared to normal carry tree adders.

Keywords: Reversible logic, Parallel Prefix Adders (PPA), Reversible Brent Kung Adder (RBKA), Reversible Kogge Stone Adder (RKSA), Reversible Spanning Tree Adder (RSPA).

I. INTRODUCTION

The first semiconductor chips held one transistor each, subsequent advances added more and more transistors and as a consequence more individual functions or systems were integrated over time. The first integrated circuits held only a few devices, perhaps as many as ten diodes, transistors, resistors and capacitors, making it possible to fabricate one or more logic gates on a single device. Now known respectively as "small-scale integration" (SSI), improvements in technique led to devices with hundreds of logic gates, known as large-scale integration (LSI) i.e., systems with at least a thousand logic gates. Current technology has moved far past this mark and today's microprocessors have many millions of gates and hundreds of millions of individual transistors.

In reversible logic there exists a one to one mapping between the inputs and the outputs vectors. In an irreversible circuit erasing a bit is equivalent to dissipation of kTln2 joules of heat energy where k is the Boltzmann’s constant and T is the absolute temperature of environment which is demonstrated by Landauer. This resulting dissipated heat also causes Noise in the remaining circuitry, which results in computing errors. Bennett showed that the dissipated energy directly correlated to the number of lost bits, and that computers can be logically reversible, maintain their simplicity and provide accurate calculations at practical speeds. Resultantly, a new paradigm in computer design arose with the goal of reducing the entropy increase and subsequent energy dissipation. Such a logical structure must possess the same number of inputs and outputs of a reversible circuit erasing a bit is equivalent to dissipation.
called as garbage output (GO). The number of garbage output for a particular reversible gate is not fixed. The two main constraints of reversible logic circuit is

1) Fan out not allowed
2) Feedbacks or loops not allowed.

**A. Basic Reversible Gates**

Reversible Gates are circuits in which number of outputs is equal to the number of inputs and there is a one to one correspondence between the vector of inputs and outputs. It not only helps us to determine the outputs from the inputs but also helps us to uniquely recover the inputs from the outputs.

1. **Feynman Gate**

The most basic reversible gate is the Feynman Gate and is shown in Figure 1. It was defined and used to compute most other reversible gates. The Feynman gate basically consists of 2 inputs A and B and 2 outputs P and Q. Output P follows the input A and output Q is the XOR of both inputs A and B.

![Figure 1 - Feynman Gate](image)

When A=0, Q=B; When A=1, Q= B. And Also with B=0, P follows A and hence Feynman gate is used as a Fan-out gate or a copying gate. Its quantum cost is 1.

2. **Toffoli Gate (TFG)**

The next important reversible gate is the Toffoli Gate shown in Figure 2. The Toffoli Gate is a 3x3 reversible gate.

![Figure 2 - Toffoli Gate](image)

It provides, a very unique case of reversibility due to Which it is used in arithmetic applications usually. Its quantum cost is 5. This gate can be used to obtain AND logic function.

3. **Peres Gate (PRG)**

Another important gate is the Peres Gate. The basic Pere gate is as shown in Figure 3. It is a very important gate as it has a low quantum cost of 4 compared to other gates in reversibility.

![Figure 3 - Peres Gate](image)

This gate alone provides logic AND operation and logic XOR operation with C input being zero. In this work Peres gate is used for implementation of reversible logic based basic blocks like PG block, Grey cell and Black cell.

4. **HNG gate**

The HNG is a 4 inputs 4 outputs (4x4) reversible gate having the mapping (A, B, C, D) to (P=A, Q=B, R= A⊕B⊕C, S = (A⊕B).C⊕AB⊕D)) where A, B, C, D are the inputs and P, Q, R, S are the outputs respectively. It has a quantum cost of six. Its logic circuit is shown in Fig 4 below.
In pre computation stage, propagates and generates are computed for the given inputs using the given equations (1) and (2).

\[ p_i = A_i \land B_i \quad (1) \]
\[ G_i = A_i B_i \quad (2) \]

Prefix Stage

In the prefix stage, group generate/propagate signals are computed at each bit using the given equations. The black cell generates the ordered pair and the grey cell generates only the left signal. The fundamental carry operator is denoted by the symbol “\( o \)”. The generated and propagated signals are combined using the fundamental operator as shown in equation (3).

\[
(g_L, p_L) o (g_R, p_R) = (g_L + p_L, g_R, p_L, g_R) \quad (3)
\]

The black cell and grey cell logic definition are shown in Figure 6.

From the black and grey cell logic definition, the generated and propagated signals are given by equations (4) and (5).

\[ G_{i,k} = G_{i,j} + P_{i,j} G_{j-1,k} \quad (4) \]
\[ P_{i,k} = P_{i,j} P_{j-1,k} \quad (5) \]

More practically, the equations (4) and (5) can be expressed using a symbol “\( o \)” denoted by Brent and Kung. Its function is exactly the same as that of a black cell and is given in (6). The “\( o \)” operation will help make the rules of building prefix structures [7].

\[
G_{i,k} : P_{i,k} = (G_{i,j}, P_{i,j}) o (G_{j-1,k}, P_{j-1,k}) \quad (6)
\]
Final Computation

In the final computation, the sum and carryout are the final output. These are given by equations (7) and (8).

The generate/propagate signals can be grouped in different fashion to get the same correct carries. Based on different ways of grouping the generate/propagate signals, different prefix architectures can be created.

\[ S_i = P_i, G_{i-1} \quad (7) \]
\[ C_{\text{out}} = G_{n:1} \quad (8) \]

IV. REVERSIBLE PARALLEL PREFIX ADDERS

The introduction of reversibility into the prefix adders was Accomplish by the usage of its basic structural units PG block, the Grey Cell and the Black Cell. In these blocks the equations for generate and propagate were implemented in structural description using reversible logic gates rather than using basic gates. Thereby the entire structure was simulated with reversible gates by making the fundamental units of the implementation as reversible. The reversible Propagate and generate block designed using single Peres gate.

The reversible implementation of the Grey cell structure is as shown in Figure 7. This design was carried out by using two Peres gate to get generate.

\[ P_{\text{ex}[1]} \]
\[ G_{k;j} \]
\[ P_{\text{ex}[0]} \]
\[ \text{PERES GATE} \]
\[ A \]
\[ B \]
\[ C \]
\[ G_{k;j} \]

**Figure 7:** Structure of reversible Grey Cell

Similarly the Black cell was also implemented using the Peres Gate along with the reversible Grey cell as shown in Figure 8. Black cell can also be designed using Toffoli gate.

\[ G_{k;j} \]
\[ P_{j;j} \]
\[ \text{PERES GATE} \]
\[ \text{PERES GATE} \]
\[ G_{k;j} \]

**Figure 8:** Structure of reversible Black Cell

Reversible Parallel Prefix Adders are classified into
1. Reversible Kogge- Stone Adder
2. Reversible Brent-Kung Adder
3. Reversible Spanning Tree Adder

1. Reversible Kogge - Stone Adder

Reversible Kogge-Stone adder is a parallel prefix form carry look ahead adder. The Kogge-Stone adder was developed by peter M. Kogge and Harold S. Stone which they published in 1973. In RKSA all conventional full adders are replaced with reversible HNG gate. Reversible Kogge-Stone prefix adder is a fast adder design.

\[ \text{Figure 9: 16-bit Reversible Kogge Stone Adder} \]

KS adder has best performance in VLSI implementations. Reversible Kogge-Stone adder has large area with minimum fan-out. The Reversible Kogge Stone Adder is widely
known as a parallel prefix adder that performs fast logical addition. Reversible Kogge Stone adder is used for wide adders because it shows the less delay among other architectures. In each vertical stage produces Propagate and Generate bits. Generate bits are produced in the last stage and these bits are XOR ed with the initial propagate after the input to produce the sum bits.

2. Reversible Brent-Kung Adder

The Reversible Brent Kung adder is a parallel prefix adder. The Brent Kung adder was developed by Brent and Kung which they published in 1982. Brent Kung adder has maximum logic depth and minimum area. The number of cells is calculated by using 2(n-1) -Log2n. The 16bit Reversible Brent Kung adder figures are shown below.

![Figure 10: 16-bit Reversible Brent Kung Adder](image)

3. Reversible Spanning tree adder

RSTA is also tested. Like the RSKA, this adder also terminates with a RCA. It also uses the BC’s and GC’s and full adder blocks like RSKA’s but the difference is the interconnection between them. The 16 bit RSTA is shown in the Figure 11.

![Figure 11: 16-bit Reversible Spanning tree adder](image)

V. SIMULATION & SYNTHESIS RESULTS

In this proposed architecture, the implementation code for 16-bit Reversible Kogge-Stone, Reversible Brent-Kung adder, Reversible Spanning Tree adder, Reversible Sparse Kogge Stone Adder was developed and corresponding values of Delay and Area were observed.

<table>
<thead>
<tr>
<th>Name of the Adder</th>
<th>No. of LUT’s</th>
<th>No. of Bonded IOB’s</th>
<th>Delay(ns)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brent Kung Adder</td>
<td>31</td>
<td>50</td>
<td>3.540</td>
</tr>
<tr>
<td>Reversible Brent Kung Adder</td>
<td>35</td>
<td>50</td>
<td>3.588</td>
</tr>
<tr>
<td>Kogge Stone Adder</td>
<td>42</td>
<td>50</td>
<td>3.673</td>
</tr>
<tr>
<td>Reversible Kogge stone Adder</td>
<td>77</td>
<td>50</td>
<td>2.595</td>
</tr>
<tr>
<td>Spanning Tree Adder</td>
<td>28</td>
<td>50</td>
<td>3.250</td>
</tr>
<tr>
<td>Reversible Spanning tree Adder</td>
<td>30</td>
<td>50</td>
<td>3.242</td>
</tr>
</tbody>
</table>

Table 1. Comparison of parallel prefix and reversible parallel prefix adders.
VI. CONCLUSION
A modest approach is suggested in this paper to reduce the power and delay of Parallel Prefix Adders. The Parallel Prefix Adders are better than the serial adders in terms of delay and at the same time there is a trade-off with the area occupied. The results obtained for carry chain adders at higher bit widths have higher performance when compared to serial adders. Because the adder is often the critical element which determines to a large part the cycle time and power dissipation for many digital signal processing and cryptographically implementations. Thus a design of prefix adder amalgamation reversible logic gates using 16 bit kogge stone adder, 16 bit Brent kung adder and 16bit spanning tree adder.

REFERENCES
[4]. Sudheer Kumar Yezerla and B RajendraNaik, “Design and Estimation of delay, power and area for Parallel prefix adders” Proceedings of 2014 RAECs UIET Punjab University Chandigarh, 06 - 08 March, IEEE.
[7]. Toffoli, T. (1980), Reversible computing (pp. 632-644). Springer Berlin Heidelberg