Reversible Logic based Kogge-Stone Adder

Ashwini B N[1], Dr. U B Mahadevaswamy [2], Abhilash G[3]

Assistant Professor, Department of TCE, GSSSIETW, Mysuru, INDIA [1]
Associate Professor, Department of ECE, SJCE, Mysuru, INDIA[2]
Assistant Professor, Department of ECE, ATMECE, Mysuru, INDIA [3]

***Abstract*— Parallel-prefix adders (also known as carry tree adders) are known to have the best performance in VLSI Design. In this paper we present a faster adder for high speed computation with reduced power consumption. Kogge Stone Adder is a parallel prefix form of carry look Ahead adder & is widely used adder in the industries of the present day. Since the adder generates the carry signal in log2N time, it is considered to be the fastest adder design possible. For the reduction of power dissipation reversible logic gates are used and hence this paper Kogge Stone Adder is implemented with reversible logic. All the structures discussed in paper have been designed in structural description using Verilog HDL and simulated in Xilinx ISE 14.7.**

***Keywords*—Parallel Prefix Adder, Kogge Stone Adder, Reversible logic, Peres gate.**

I. INTRODUCTION

All the digital circuit designs including digital signal processors (DSP) and microprocessor units require binary addition. Because of high demand for speed and low power multiplier research continues to be focused on improving performance of the adder. Parallel-prefix adders are known for their good performance in VLSI implementations. Reconfigurable logic such as Field Programmable Gate Arrays (FPGAs) has popular in recent years because it offers improved performance in terms of speed and power over DSP-and microprocessor-based solutions for many practical designs including mobile DSP and telecommunications applications and a significant reduction in development time and cost over Application Specific Integrated Circuit (ASIC) designs.

Reversible logic is one of the promising fields for future low power design technologies. Since all DSP processors and other hand held devices requires minimization of power dissipation of multipliers with high speed in turn gives priority to improve Adder block for multiplier.

The need and possibility for reversibility was shown by Landauer et. al. [9]. Accordingly the need for reversibility was the reductions of energy dissipation which was shown to be kT.ln2 , where k is Boltzmann's constant (1.3807 X 10-23JK-1) and T the temperature. Applying reversibility one can reduce the amount of energy due to an irreversible operation. A gate M with N inputs is said to be reversible, if every output state y_i of the circuit has a unique input state x_i such that

$$M(x_i) = y_i \quad i,j = 1 \text{ to } N \qquad (1)$$

The basic reversible gate is the Inverter. It has only a single input and single output. But it satisfies the above defined function of possessing a unique output which is mapped to a unique input. The unique mapping of input and output is required to allow the identification of the input from the obtained output. As the above mentioned definition and the logic of direct mapping of individual states of the output to which a specific input, a number of reversible gates have been identified.

A. 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.

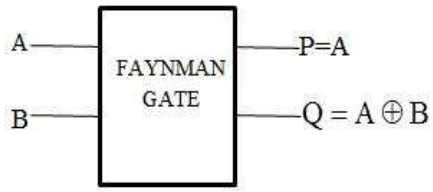

Figure 1 - Feynman Gate

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.

B. Toffoli Gate (TFG)

The next important reversible gate is the Toffoli Gate [7] shown in Figure 2. The Toffoli Gate is a 3x3 reversible gate.

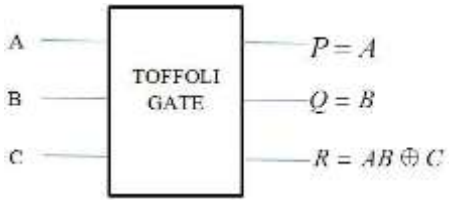

Figure 2 - Toffoli Gate

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.

C. Peres Gate (PRG)

Another important gate is the Peres Gate [8]. 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.

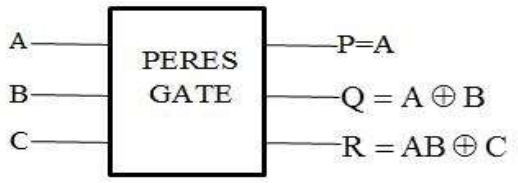

Figure 3 - Peres Gate

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.

The main objective of this paper is the discussion of the work carried out in the development of a Reversible logic based Prefix adder popularly known as the Kogge-Stone adder (KSA) [2]. Therefore primarily the actual structure of the KSA is discussed followed by the reversible structure realized for the KSA.

The remaining paper consists of three sections. In Section II the work related to Kogge stone adder has been shown. In section III the reversible logic based KSA structure used in this paper has been discussed. Section IV summarizes the work carried out followed by the conclusion drawn from the work carried out.

II. KOGGE-STONE ADDER

For wide adders (roughly, N > 16 bits), the delay of carry-lookahead (or carry-skip or carry-select) adders becomes dominated by the delay of passing the carry through the lookahead stages. This delay can be reduced by looking ahead across the lookahead blocks [3]. Three fundamental trees are the Brent-Kung, Sklansky, and Kogge-Stone architectures. The Kogge-Stone tree [2] (Figure 4) achieves both log2 N stages and fan out of 2 at each stage.

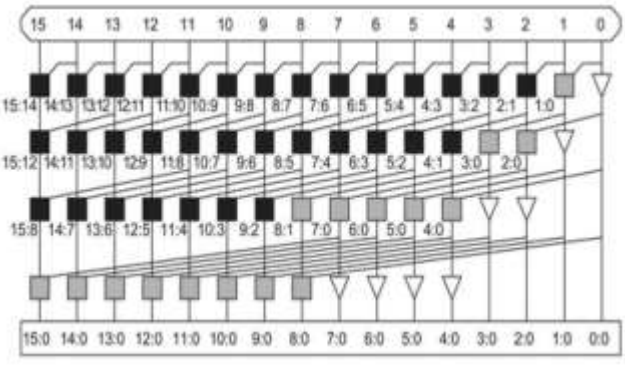

Figure 4 - PG network of 16-bit KSA structure (Courtesy [3])

This comes at the cost of many long wires that must be routed between stages. The Kogge-Stone tree is widely used in high-performance 32-bit and 64-bit adders.

The Kogge-Stone Adder (KSA) was shown to possess the minimum fan-out for the maximum wire tracks as shown in the taxonomy of PPA's as shown in Figure 5 . Hence the KSA has been chosen as the most desirable adder structure for parallel addition operations in this paper.

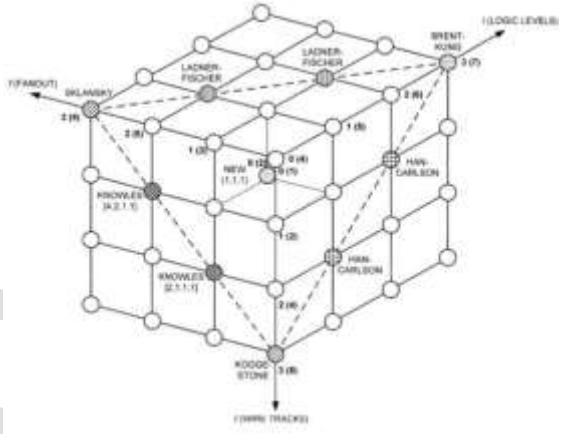

Figure 5 - Prefix Adder Taxonomy (Courtesy [3])

Based on this choice the KSA has been designed for different bit widths. For the implementation of the KSA the main basic blocks are again the Generate and Propagate structures. Essentially the KSA uses three different stages for the generation of the sum output. It initially accepts the addends. From the addends, it uses the Parallel-Prefix structure to generate the carry. From the carry it generates the sum output for the entire adder.

The entire KSA is a process divided into the following three stages:

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

During Pre-processing stage, the generate and propagate signals corresponding to each pair of bits from the inputs A and B are obtained. Mathematically, this may be defined as

$$Pi = Ai \oplus Bi \qquad \text{-- (2)}$$

$$Gi = Ai \bullet Bi \qquad \text{-- (3)}$$

Where P_i and G_i are the propagate and generate for i^{th} input and Ai and Bi are the inputs for the i^{th} stage.

During the next stage, carries corresponding to each bit is computed using the carry generation network. This network consists of two fundamental elements called as Grey cell and Black cell. The grey cell computes the carry output of the current stage based on all previous carried directly, while the

black cell computes both generate and propagate based on previous inputs. The generation operation is carried out in parallel and therefore accounts for the high processing speed of this adder. The functioning of this adder may be explained by the following set of equations:

$$Pi:j = P\,i{:}k+1 \quad \bullet Pk{:}j \qquad \text{-- (4)}$$

$$Gi:j = G\,i{:}k+1 \quad | (Pi:k+1 \bullet Gk:j) \qquad \text{-- (5)}$$

The Generate $G_{i:j}$ is generated by the grey cell. While the Propagate $P_{i:j}$ is generated by the black cell along with the generate block for the same stage. The structure of the propagate and generate, black cell and the grey cell are as shown in Figure 6, Figure 7 and Figure 8 respectively. After obtaining final generate signal, sum is obtained by performing XOR operation between propagate signal from PG block and generate signal from each grey cell..

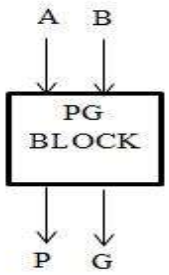

Figure 6 - Propagate and Generate block of KSA

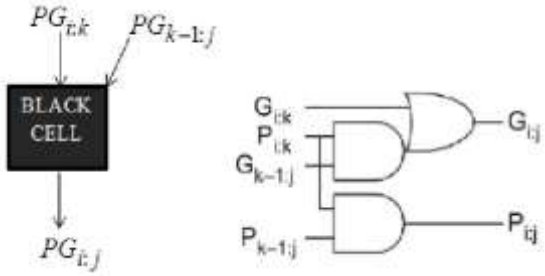

Figure 7 - Structure of Black Cell for the implementation of KSA

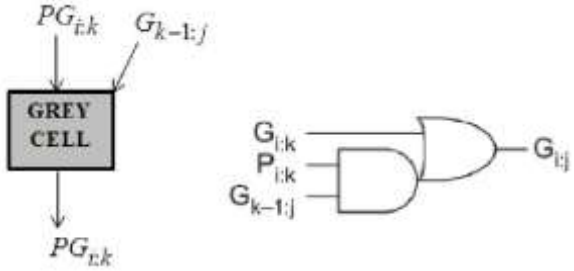

Figure 8 - Structure of Grey Cell for the implementation of KSA

III. REVERSIBILITY FOR KSA

The introduction of reversibility into the KSA was accomplished 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. Figure 9 shows the reversible Propagate and generate block designed using single Peres gate.

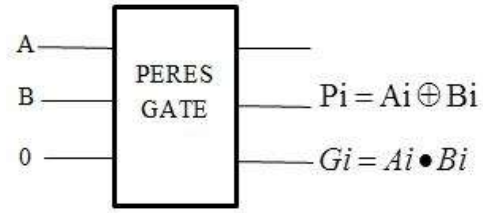

Figure 9 - Structure of reversible Propagate and Generate Block

The reversible implementation of the Grey cell structure is as shown in Figure 10. This design was carried out by using two Peres gate to get generate.

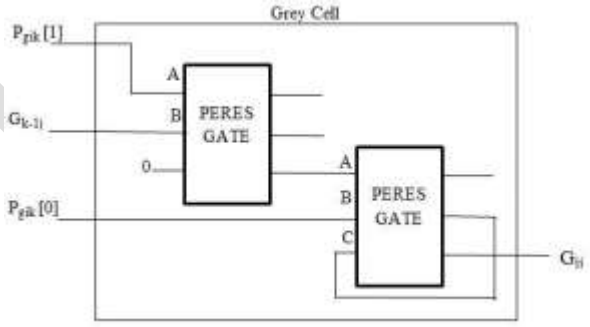

Figure 10 - Structure of reversible Grey Cell

Similarly the Black cell was also implemented using the Peres Gate along with the reversible Grey cell as shown in Figure 11. Black cell can also be designed using Toffoli gate.

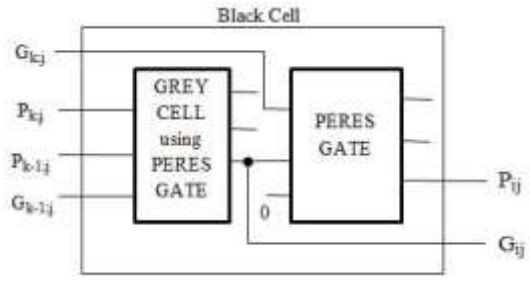

Figure 11 - Structure of reversible Black Cell

Using these basic reversible structures, propagate and final generate signals are obtained. To implement final sum of reversible logic based KSA, Peres gate is used to perform XOR operation between generate signal from grey cell and propagate signal from PG cell. Carry out is obtained using last

stage black cell. TSG gate [6] can also be used as it is having more logic function, but it leads to more garbage outputs.

IV. SIMULATION RESULTS

The conventional KSA was implemented by coding a structural description in Verilog HDL. The description was begun by a basic gate implementation of PG block, Grey and Black cells. Using these basic structures the configurations of KSA have been simulated as shown in Figure 4 which is a PG network of KSA structure. To verify the working of the adder different values are taken as inputs and obtained corresponding sum and carry. Simulation result for 16-bit conventional KSA is as shown in Figure 12.

Figure 12 - 16-bit conventional KSA (without reversibility)

For a reversible implementation, all the basic gates: PG block, Grey and Black cells have been simulated with reversible gates as discussed in section III. Simulation results of reversible logic based 16-bit and 32-bit KSA is shown in Figure 13.

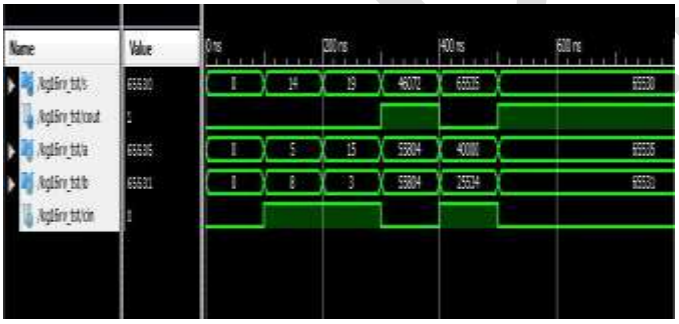

Figure 13 - (a) Reversible logic based16-bit KSA

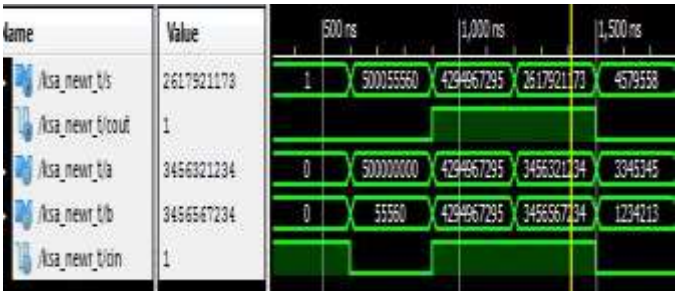

Figure 13 (b) - Reversible logic based 32-bit KSA

Figure 13 (a-b) - Simulation results of Reversible logic based KSA

Table 1 Comparison of 16-bit KSA Implementations.

Design Configuration	Number of Slice LUTs	Number of bonded IOBs	Delay	% of delay reduced	
				w.r.t KSA[4]	w.r.t KSA[1]
KSA [4]	71	50	6.70ns	-	
KSA [1]	-	-	6.286ns	-	
KSA without reversible Logic(Proposed Design)	38	50	4.956ns	21	26
KSA with reversible Logic(Proposed Design)	87	50	4.488ns	33	28

The comparison of the KSA with a reversible logic based implementation is providing reduced delay and increased area. Table 1 shows the obtained results of the16-bit KSA implementations. The area is increased by 22% in reversible based KSA compared to the structure without reversibility and ref [4]

Table 2 Comparison of 32-bit KSA Implementations.

Design Configuration	Number of Slice LUTs	Number of bonded IOBs	Delay	% of delay reduced w.r.t KSA[1]
KSA [1]	-	-	7.438ns	-
KSA without reversible Logic(Proposed Design)	141	98	7.165ns	3
KSA with reversible Logic(Proposed Design)	219	98	5.737ns	22

Comparison of 32bit reversible logic based KSA and conventional KSA is shown in Table 2. The delay reduction is obtained for 32bit KSA. The area is increased by 55% in reversible based KSA compared to the structure without reversibility (proposed design).

V. CONCLUSION

From the study of analysis done on area and delay, we have concluded that the delay efficiency is improved by 33 % in proposed design for KSA, when compared to [4] and it is improved by 28 % when compared with [1]. Performance of KSA structure is improved in terms of delay with the increase of area.

DECLARATION

This work is being carried out as an academic pursuit. It is not being published with any secondary interest like financial gain. We therefore declare that there is no conflict of interest regarding the publication of this paper.

REFERENCES

[1]. David H.K.Hoe, Chris Martinez and Sri Jyothsna Vundavalli, ", Design and Characterization of Parallel Prefix Adders using FPGAs", 2011 IEEE 43rd Southeastern Symposium in pp. 168-172, 2011.

[2]. P. M. Kogge and H. S. Stone, "A Parallel Algorithm for the Efficient Solution of a General Class of Recurrence Equations," IEEE Trans. On Computers, Vol. C-22, No 8, August 1973
[3]. N. H. E. Weste and D. Harris, CMOS VLSI Design, 4th edition, Pearson–Addison-Wesley, 2011.
[4]. Sudheer Kumar Yezerla and B Rajendra Naik, "Design and Estimation of delay, power and area for Parallel prefix adders" Proceedings of 2014 RAECS UIET Panjab University Chandigarh, 06 - 08 March, IEEE.
[5]. Fredkin, E., & Toffoli, T. (2002),Conservative logic,(pp. 47-81). Springer London.
[6]. Thapliyal, Himanshu, and M. B. Srinivas. "A new reversible TSG gate and its application for designing efficient adder circuits." arXiv preprint cs/0603091(2006).
[7]. Toffoli, T. (1980),Reversible computing(pp. 632-644). Springer Berlin Heidelberg
[8]. Peres, A. (1985). Reversible logic and quantum computers. Physical review A, 32(6), 3266.
[9]. Landauer, Rold. "Irreversibility and heat generation in the computing process." IBM Journal of Research and Development 44.1 (2000): 261-269.