

VLSI Implementation and Design of Carry Skip Adder Using Reversible Gates for Arithmetic Applications

Mrs.R.J.Vijaya Saraswathi¹, Dhana Shree.B², Jenifa Jafinth Shalom.S³, Deepika.S⁴

¹Asst Professor, Dept. of EIE, Panimalar engineering college, Chennai, Tamil Nadu, India.

²Student, Dept. of EIE, Panimalar engineering college, Chennai, Tamil Nadu, India.

³Student, Dept. of EIE, Panimalar engineering college, Chennai, Tamil Nadu, India.

⁴Student, Dept. of EIE, Panimalar engineering college, Chennai, Tamil Nadu, India.

ABSTRACT

ABSTRACT-In this paper, we present a Carry Skip Adder (CSKA) structure using reversible logic that has a higher speed yet lower energy consumption compared with the existing system. The existing system makes use of AND-OR Invert (AOI) and OR-AND Invert (OAI) compound gates for the skip logic with both fixed stage size and variable stage size. In the proposed system, low complexity Carry Skip Adder is designed by using reversible gates. Reversible gate having the properties of moderate delay, low complexity and low quantum cost is chosen especially as New Toffoli gate or Peres gate. This gate acts as universal reversible three bit gate that swaps last two bits if first bit is high. If it is set, it inverts or remains same during the operations. Each and every block in skip adder is modified as proposed gate. By this process, full adder is designed by New Toffoli gate or Peres gate and then it is extended to RCA and incrementation blocks. The proposed structures are assessed by comparing their propagation delay, power consumption and area parameters with those of other adders. Simulation on the proposed CSKA reveal reduction in the power consumption and propagation delay but the area remains almost same as the existing system. Compared with the latest works in this field, the proposed system is having a reasonably high speed.

Keyword: - Carry Skip Adder (CSKA), New Toffoli gate or Peres gate, AND-OR Invert (AOI), OR-AND Invert (OAI).

1. INTRODUCTION

The binary adder is the critical element in most digital circuit designs including digital signal processors (DSP) and microprocessor data path units. As such, extensive research continues to be focused on improving the power delay performance of the adder. In VLSI implementations, parallel-prefix adders are known to have the best performance. Binary adders are one of the most essential logic elements within a digital system. In addition, binary adders are also helpful in units other than Arithmetic Logic Units (ALU), such as multipliers, dividers and memory addressing. Therefore, binary addition is essential that any improvement in binary addition can result in a performance boost for any computing system and, hence, help improve the performance of the entire system. Parallel-prefix adders (also known as carry-tree adders) are known to have the best performance in VLSI designs.

A **carry-skip adder** (also known as a **carry-bypass adder**) is an **adder** implementation that improves on the delay of a ripple-carry adder with little effort compared to other adders. The improvement of the worst-case delay is achieved by using several **carry-skip adders** to form a block-carry-skip adder. A fast carry look-ahead logic using group generate and group propagate functions is used to speed up the performance of multiple stages of ripple carry adders. This greatly reduces the latency of the adder through its critical path, since the carry bit for each block can now "skip" over blocks with a group propagate signal set to logic 1 (as opposed to a long ripple-carry chain, which would require the carry to ripple through each bit in the adder). The number of inputs of the AND-gate is equal to

the width of the adder. For a large width, this becomes impractical and leads to additional delays, because the AND-gate has to be built as a tree. A good width is achieved, when the sum-logic has the same depth like the n -input AND-gate and the multiplexer. In the carry skip adder to speed-up operation, carry propagation is skipped to position i without waiting for rippling. A carry-skip adder reduces the carry-propagation time by skipping over groups of consecutive adder stages. The carry-skip adder is usually comparable in speed to the carry look-ahead technique, but it requires less chip area and consumes less power.

2. EXISTING SYSTEM

In this section, first, the structure of a generic variable latency adder, which may be used with the voltage scaling relying on adaptive clock stretching, is described.

2.1 A Variable Latency Adders Relying on Adaptive Clock Stretching:

The basic idea behind variable latency adders is that the critical paths of the adders are activated rarely [1]. Hence, the supply voltage may be scaled down without decreasing the clock frequency. If the critical paths are not activated, one clock period is enough for completing the operation. In the cases, where the critical paths are activated, the structure allows two clock periods for finishing the operation. Hence, in this structure, the slack between the longest off-critical paths and the longest critical paths determines the maximum amount of the supply voltage scaling. Therefore, in the variable latency adders, for determining the critical paths activation, a predictor block, which works based on the inputs pattern, is required [2]. The concepts of the variable latency adders, adaptive clock stretching, and also supply voltage scaling in an N -bit RCA adder may be explained using Fig. 1. The predictor block consists of some XOR and AND gates that determines the product of the propagate signals of considered bit positions. Since the block has some area and power overheads, only few middle bits are used to predict the activation of the critical paths at price of prediction accuracy decrease [3], [1]. In Fig. 1, the input bits $(j + 1)$ th– $(j + m)$ th have been exploited to predict the propagation of the carry output of the j th stage (FA) to the carry output of $(j + m)$ th stage. For this configuration, the carry propagation path from the first stage to the N th stage is the longest critical path (which is denoted by Long Latency Path (LLP), while the carry propagation path from first stage to the $(j+m)$ th stage and the carry propagation path from $(j + 1)$ th stage to the N th stage (which are denoted by Short Latency Path (SLP1) and SLP2, respectively) are the longest off-critical paths. It should be noted the paths that the predictor shows are (are not) active for a given set of inputs are considered as critical (off-critical) paths. Having the bits in the middle decreases the maximum of the off-critical paths [1]. The range of voltage scaling is determined by the slack time, which is defined by the delay difference between LLP and max (SLP1, SLP2).

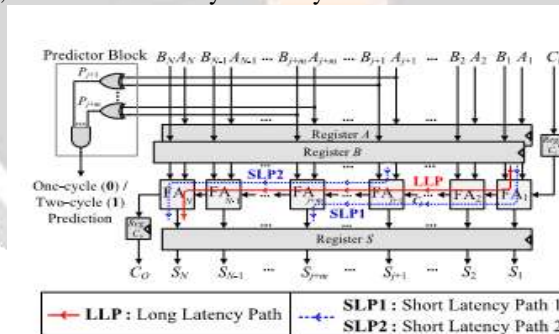


Fig -1. Generic Structure of Variable Latency Adders based on RCA

2.2. Existing Hybrid Variable Latency CSKA Structure:

The basic idea behind using VSS CSKA structures was based on almost balancing the delays of paths such that the delay of the critical path is minimized compared with that of the FSS structure [4]. This deprives us from having the opportunity of using the slack time for the supply voltage scaling. To provide the variable latency feature for the VSS CSKA structure, we replace some of the middle stages in our existing structure with a PPA modified in this paper. The existing hybrid variable latency CSKA structure is shown in Fig. 2 where an M_p -bit modified PPA is used for the p th stage (nucleus stage). Since the nucleus stage, which has the largest size (and delay) among the stages, is present in both SLP1 and SLP2, replacing it by the PPA reduces the delay of the longest off-critical paths.

Thus, the use of the fast PPA helps increasing the available slack time in the variable latency structure. It should be mentioned that since the input bits of the PPA block are used in the predictor block, this block becomes parts of both SLP1 and SLP2. In the existing hybrid structure, the prefix network of the Brent–Kung adder [5] is used for constructing the nucleus stage (Fig.3).

One the advantages of the this adder compared with other prefix adders is that in this structure, using forward paths, the longest carry is calculated sooner compared with the intermediate carries, which are computed by backward paths. In addition, the fan-out of adder is less than other parallel adders, while the length of its wiring is smaller [6]. Finally, it has a simple and regular layout. The internal structure of the stage p , including the modified PPA and skip logic, is shown in Fig. 3. Note that, for this figure, the size of the PPA is assumed to be 8 (i.e., $M_p = 8$).

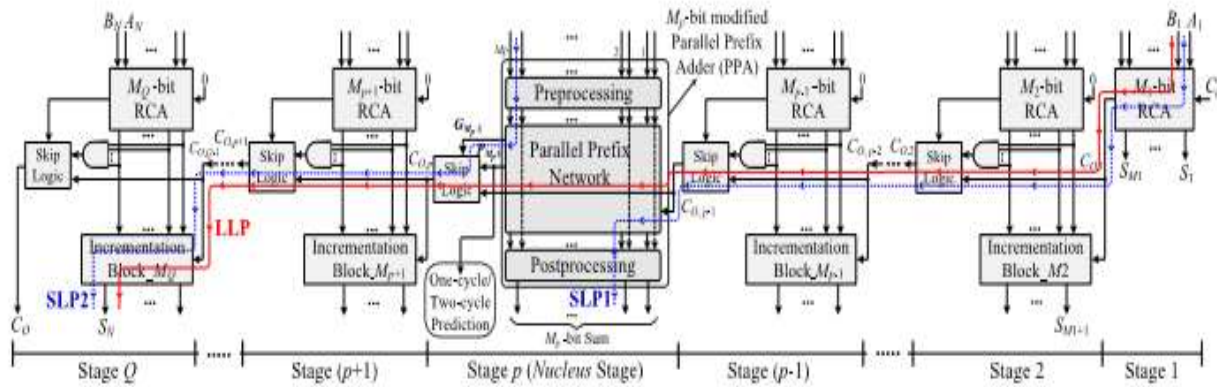


Fig 2. Structure of existing hybrid variable latency CSKA

As shown in the figure, in the preprocessing level, the propagate signals (P_i) and generate signals (G_i) for the inputs are calculated. In the next level, using Brent–Kung parallel prefix network, the longest carry (i.e., $G8:1$) of the prefix network along with $P8:1$, which is the product of the all propagate signals of the inputs, are calculated sooner than other intermediate signals in this network. The signal $P8:1$ is used in the skip logic to determine if the carry output of the previous stage (i.e., $C0, p-1$) should be skipped or not.

In addition, this signal is exploited as the predictor signal in the variable latency adder. It should be mentioned that all of these operations are performed in parallel with other stages. In the case, where $P8:1$ is one, $C0, p-1$ should skip this stage predicting that some critical paths are activated. On the other hand, when $P8:1$ is zero, $C0, p$ is equal to the $G8:1$.

In addition, no critical path will be activated in this case. After the parallel prefix network, the intermediate carries, which are functions of $C0, p-1$ and intermediate signals, are computed (Fig. 3). Finally, in the post processing level, the output sums of this stage are calculated. It should be noted that this implementation is based on the similar ideas of the concatenation and incrementation concepts used in the CI-CSKA. It should be noted that the end part of the SLP1 path from $C0, p-1$ to final summation results of the PPA block and the beginning part of the SLP2 paths from inputs of this block to $C0, p$ belong to the PPA block (Fig. 3). Since the PPA structure is more efficient when its size is equal to an integer power of two, we can select a larger size for the nucleus stage accordingly [6]. The larger size (number of bits), compared with that of the nucleus stage in the original CI-CSKA structure, leads to the decrease in the number of stages as well smaller delays for SLP1 and SLP2. Thus, the slack time increases further.

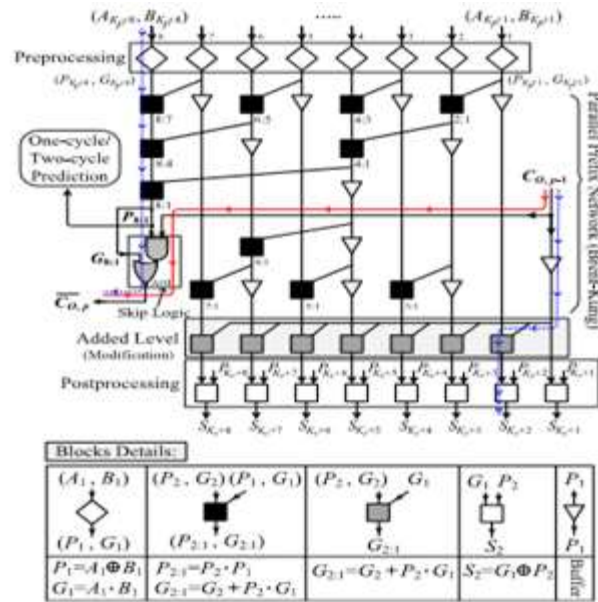


Fig 3. Internal Structure of pth stage of existing hybrid variable latency CSA

3. PROPOSED SYSTEM:

One or more operation can implement in a single unit called Reversible Gate. A reversible logic gate is an n-input and n-output device with one-to-one mapping. These gates are helps to determine the outputs from the inputs and also the inputs can be uniquely recovered from the outputs. By using these gates, the power dissipation can be lowered. Different reversible gates are Feymann gate, Peres gate, HNG gate, Toffoli gate, etc.

3.1 New Toffoli gate or Peres gate

The figure below shows a 3*3 Peres gate. The input vector is I (A, B, C) and the output vector is O (P, Q, and R). The output is defined by $P = A$, $Q = A \oplus B$ and $R = AB \oplus C$. Quantum cost of a Peres gate is 4[7].

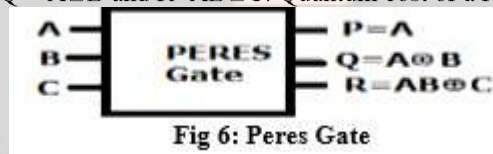


Fig 6: Peres Gate

Fig 1. 3*3 Peres gate

A	B	C	P	Q	R
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	1	0
1	0	1	1	1	1
1	1	0	1	0	1
1	1	1	1	0	0

Truth table of Peres gate

Peres [7] introduced a new gate known as Peres gate. Peres gate is also a 3*3 gate but it is not a universal gate like the Fredkin and Toffoli gate. Even though this gate is not universal gate it is widely used in much application because it has less quantum cost with respect to the universal gate. The quantum cost of the Peres gate is 4. Peres Gate (PG) is a reversible logic gates. Reversible logic gates are attracting a lot of attention due to their zero power dissipation under ideal conditions. Reversible logic circuits are useful for constructing quantum computers.

In low power VLSI design and power optimization reversible logic method is being used more frequently. It is also the fundamental need for the emerging field of the quantum computing, Digital signal processing and communications. The use of reversible logic in VLSI design has many advantages because it reduces number of gates and garbage outputs. When we use a logically irreversible gate we dissipate energy into the environment. The loss of energy equals to the information loss. One bit information lost dissipates $KT \ln 2$ of energy. By reversible logic we can reduce power dissipation. In reversible logic number of input equals to the number of outputs. i.e. $k*k$. All the outputs are not used as input to other gate is called garbage output. Especially, Peres gate is used because of its lowest quantum cost. We have used a Full adder having zero garbage outputs and zero constant inputs. The proposed full adder will be of great help in reducing the garbage outputs and constant input parameters.

Full adder is the fundamental building block in many computational units. The full adder circuit's output is given by the following equations:

$$\text{Sum} = A \oplus B \oplus C_{in}$$

$$C_{out} = (A \oplus B) C_{in} \oplus AB$$

A full-adder using two Peres gates is as shown in fig 2. The quantum realization of this shows that its quantum cost is 8 two Peres gates are used.

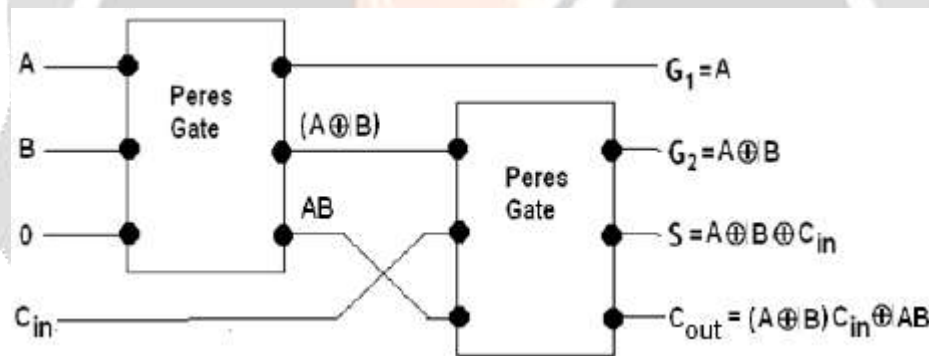


Fig.2. Full adder using two Peres gates

In the proposed system, the logic gates in the Ripple carry adder and incrementation block are replaced by Peres gate or new Toffoli gate.

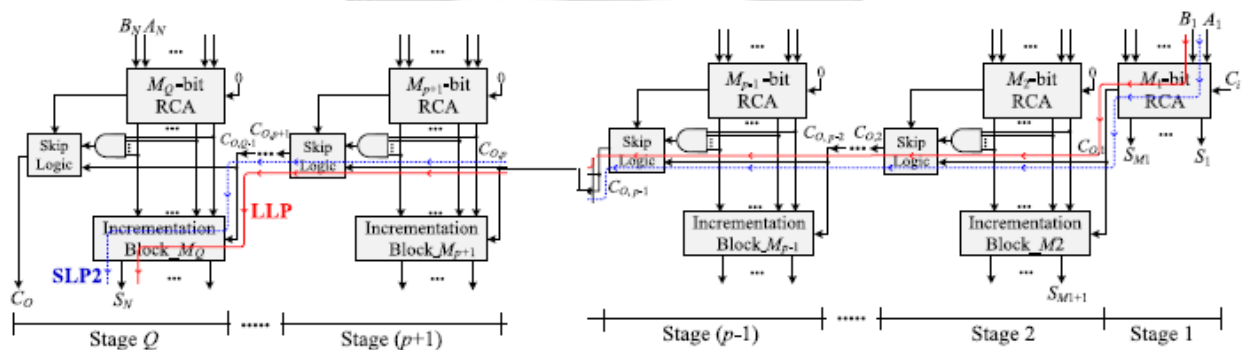


Fig.3. Structure of proposed system CSKA

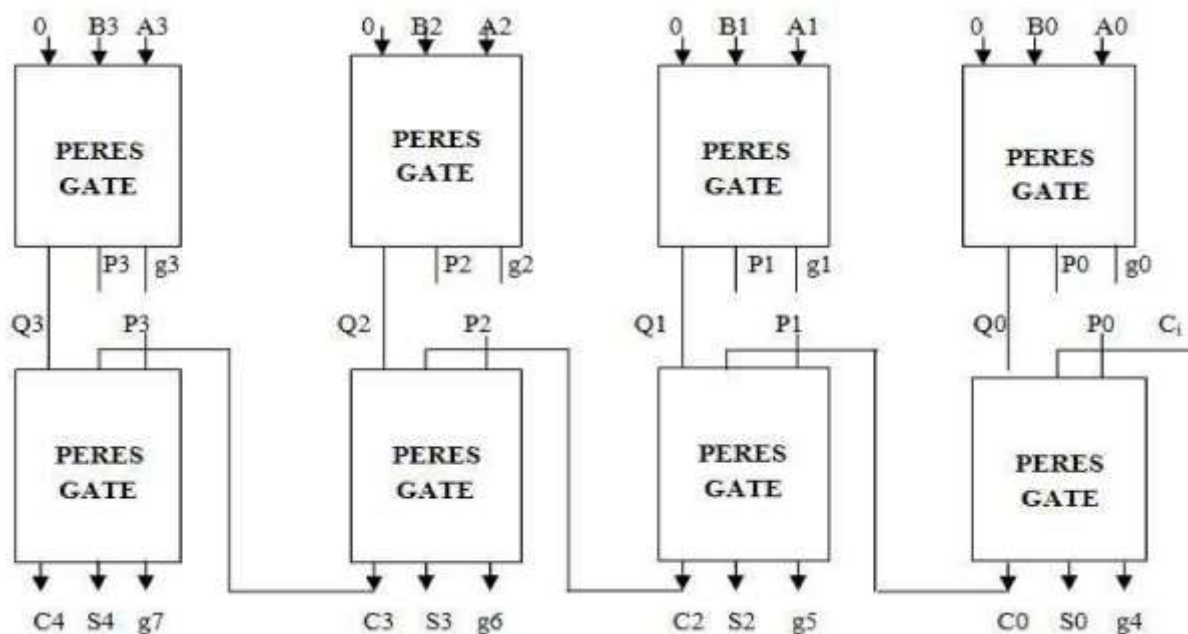
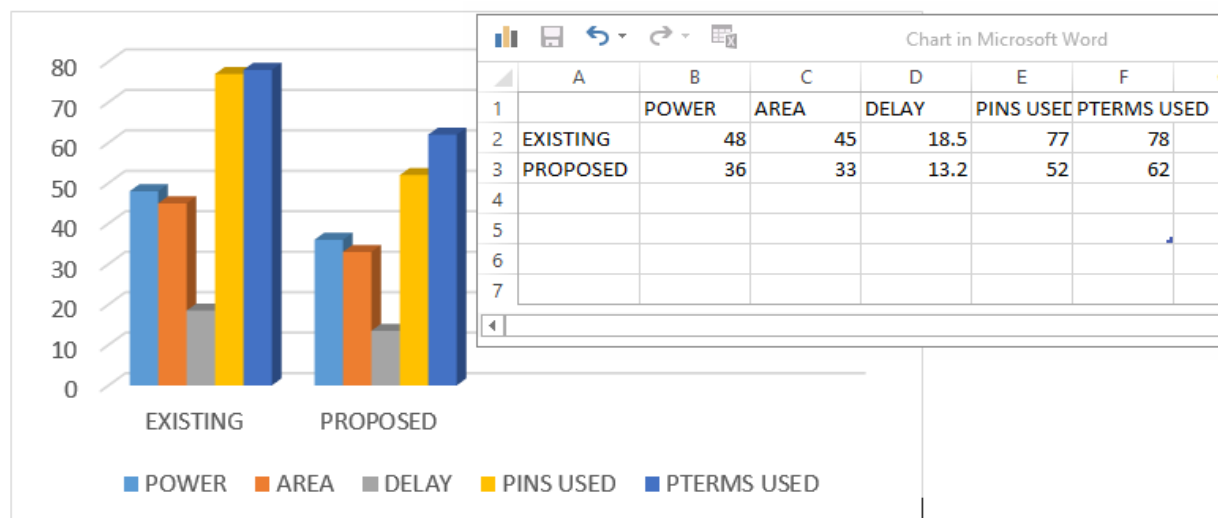


Fig.4. Ripple carry adder using Peres gates

The above shown fig.4 illustrates the use of Peres gate in Ripple carry adder.

4. RESULT

The bar graph shown below is the comparison of the existing and proposed system in terms of power, area, delay, pins used and pterms used (in-outs). In VLSI, the main aim is to reduce the power, area and delay of the system used. The below graph shows that the power, area and delay has been reduced when compared to the existing system.



5. CONCLUSION

In this paper, a New Toffoli gate or Peres gate has been proposed for the replacement of logic gates in RCA and incrementation block. As far as verifying the functionality, this proposed gate has been proven by some physical relations such as power, area and delay well as computer simulations. These designs have been verified in

ModelSim simulator, Xilinx ISE simulator and CPLD development kit. This design is very useful for future ultra-low power digital circuits and quantum computers.

6. REFERENCES

- [1]. S. Ghosh, D. Mohapatra, G. Karakonstantis, and K. Roy, "Voltage scalable high-speed robust hybrid arithmetic units using adaptive clocking," *IEEE Trans. Very Large Scale Integer. (VLSI) Syst.*, vol. 18, no. 9, pp. 1301–1309, Sep. 2010
- [2]. H. Suzuki, W. Jeong, and K. Roy, "Low-power carry-select adder using adaptive supply voltage based on input vector patterns," in *Proc. Int. Symp. Low Power Electron. Design (ISLPED)*, Aug. 2004, pp. 313–318
- [3]. S. Ghosh and K. Roy, "Exploring high-speed low-power hybrid arithmetic units at scaled supply and adaptive clock-stretching," in *Proc. Asia South Pacific Design Autom. Conf. (ASPDAC)*, Mar. 2008, pp. 635–640
- [4]. A. Guyot, B. Hochet, and J.-M. Muller, "A way to build efficient carry skip adders," *IEEE Trans. Comput.*, vol. C-36, no. 10, pp. 1144–1152, Oct. 1987.
- [5]. R. P. Brent and H. T. Kung, "A regular layout for parallel adders," *IEEE Trans. Comput.*, vol. C-31, no. 3, pp. 260–264, Mar. 1982.
- [6]. D. Harris, "A taxonomy of parallel prefix networks," in *Proc. IEEE Conf. Rec. 37th Asilomar Conf. Signals, Syst., Comput.*, vol. 2, Nov. 2003, pp. 2213–2217.
- [7]. A. Peres, *Reversible Logic and Quantum Computers*, *Physical Review A*, vol. 32, pp. 3266–3276, 1985.

