International Journal of Scientific Engineering and Research (IJSER)
ISSN (Online): 2347-3878
Index Copernicus Value (2015): 62.86 | Impact Factor (2015): 3.791

Design of Reversible Carry Skip Adder for Digital and Optical Circuits

[1]R. Ezhilarasi, [2]T. Perarasi, [3]R. Meera

[1,3]PG Student, Dept of ECE, Karpaga Vinayaga College of Engineering and Technology, Tamil Nadu, India
[2]Associate professor, Dept of ECE, Karpaga Vinayaga College of Engineering and Technology, Tamil Nadu, India

Abstract: ***In this paper, high speed carry skip adders for two different circuits were proposed. In the conventional structure of CSKA consists of a chain of ripple carry adder (RCA) block and 2:1 multiplexer. In the proposed structure, parallel prefix adder network is used to improve the speed and energy parameters. In addition to this the proposed structure called AND-OR-Invert (AOI) and OR-AND-Invert (OAI) gates are used for the skip logic and the Kogge-Stone adder is used. Kogge-stone adder is a type of parallel prefix. This design is used for digital circuits. In conventional digital circuits, a significant amount of energy is dissipated as the bits of information are lost during logical operations. Reversible logic is very important in low-power circuit design. This work proposes an optical reversible Carry-Skip Adder (CSKA) circuit. This design uses the Mach–Zehnder interferometers (MZI) due to its significant advantages such as high speed, low power, fast switching time and ease in fabrication. These CSKA designs use a minimum number of MZI switches. Finally, the comparison between these two circuits can be shown by using ModelSim software.***

Keywords: Carry skip adder (CSA), Kogge-stone adder, Mach-Zehnder interferometer (MZI), Reversible logic

1. Introduction

Addition and subtraction are basic arithmetic operations. It is mainly used in a lot of VLSI systems such as microprocessors and application specific digital signal processing architectures. Adders are one of the most widely used digital components in the digital integrated circuit and necessary part of Digital Signal Processing. Arithmetic units are the essential block of the digital system. Adders become a critical hardware unit for the efficient implementation of the arithmetic unit. In many arithmetic applications and another kind of applications, adders are not only in the arithmetic logic unit, it is also used in the processing unit. Addition operation can also be used in complicated operations such as encoding and decoding. The main tasks of adders are the addition of two numbers, and it is also used in subtraction, multiplication and address determinations. Adders are a key building block in arithmetic and logic units [13] so that increasing their speed and reducing their power/energy consumption strongly affect the performance of the processors. There are many methods of optimizing the speed and power of these units, which have been reported in [2]-[9]. An Addition is a process which involves the addition of two numbers and it will generate a sum and carry. The addition operations will result in sum value and carry value. The Half Adders (HA) and Full Adders (FA) is the basic block of all adder architectures. There are several many adder families. These all having various delays, powers and area usages. Examples include ripple carry adder (RCA), carry increment adder (CIA), carry select adder (CSLA), carry skip adder (CSKA) and parallel prefix adders (PPAs). The detailed description of these adder architectures along and their characteristics may found in [10] and [13]. The Ripple carry adder has the simplest structure with the low power consumption and smallest area but with the worst critical path delay. In the CLSA, the speed, power consumption, and area are larger than the RCA. The parallel prefix adder, it is also called carry look-ahead adders, that provide the direct parallel prefix structures to generate the carry as fast as possible. The CSKA is an efficient and high performance adder in terms of power consumption and area usage. The CSKA has much smaller critical path delay than the one in the RCA, whereas its area and power consumption are similar to those RCA. The design and performance of these adders are introduced in [12]. In conventional digital circuits, a significant amount of energy is dissipated as the bits of information are lost during logical operations.

Reversible circuits do not lose information during computation and reversible computation can be achieved by reversible gates. These circuits can generate unique output vector for each input vector, and unique input vector for each output vice versa, that is, there is a one-to-one mapping between the input and the output vectors. In the reversible circuit, there always exists a bijective mapping between inputs and outputs and vice-versa. The concept of the reversible circuit and its advantages in logic synthesis have already introduced in [14-15], where they have observed that reversible circuits are information lossless and can be operable in very low power. The rest of the work is organized as follows. Section II presents basics of carry skip adder and existing works. Proposed design for digital circuit and their architectural complexities have been discussed in section III. Proposed designs for optical circuits are presented in section IV. Section V presents the results of proposed work. Finally, section VI concludes the paper.

2. Existing Works

The existing works on carry skip adder are discussed in this section and a novel strategy to design carry-skip adders is proposed.

A. Carry Skip Adder:

A CSKA consists of full adder gates and multiplexer logic this configuration strongly affects the speed. These full adder blocks are connected by 2:1 multiplexers. Instead of the constant block size, we can use a variable block size to

improve the performance of the CSKA. The number of bits can be further increased the propagation delay of the CSKA, An adder should be considered optimal if it is not possible to add bits without increasing the delay.

B. Conventional Carry Skip Adder:

The conventional carry skip adder (Conv-CSKA) structure consist of the ripple carry adder blocks (RCA) and multiplexers. Each Ripple carry adder to produce a carry and it will be fed into the multiplexer block for skip logic. At each stage inputs of the multiplexer are the carry input of the stage and the carry output of its RCA stage. Consider this is an N-bit RCA so it contains N cascaded Full adders (FAs), which lead the worst propagation delay. This propagation delay belongs to the two inputs X and Y.

C. High-Speed Energy Efficient Carry Skip Adder: (CI-CSKA)

The above sections clearly present that delay mainly depends on the skip logic. Reducing the delay of multiplexer skip logic may reduce the propagation delay of Conv-CSKA. This structure is based on the concatenation and incrementation methods. This type is denoted by CI-CSKA. In this, the multiplexer skip logic can be replaced by And-Or-Invert (or) Or-And-Invert compound gates. These AOI/OAI compound gates consist of fewer transistors. The power consumption is less in this skip logic when compared to normal multiplexer logic. In this structure, carry propagates through the skip logics and it produces complemented output. The structure has a lower propagation delay with a smaller area. The inverting gates of AOI and OAI are available in standard libraries; this is the reason of using this logic. In this way, increasing power consumption and delay is eliminated. The first skip logic uses the AOI logic then the next skip logic should use the OAI logic. In Conv-CSKA structure, the skipping structure increases the delay in the critical path. In the Conv-CSKA, the multiplexer logic is not able to bypass the zero carry input until the zero carry input propagates from the RCA block. By using the concatenation approach this problem can be solved. The area and delay of this structure are mainly depending on the AOI/OAI compound gates. Static AOI and OAI compound gates use 6 transistors when the static 2:1 multiplexer uses 12 transistors. It's clearly shown that the number of transistors get reduced in modified skip logic so that the area and delay are reduced.

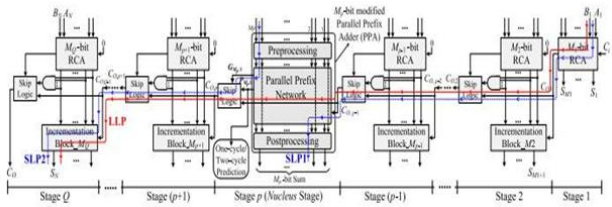

Figure 1: Proposed CI-CSKA

3. Proposed CSKA for Digital Circuits

As mentioned in above section the speed of the normal carry skip adder improved by using incrementation and concatenation methods. The proposed work is also using these methods. Reducing the skip logic delay may reduce the overall propagation delay. In this section, first, the structure carry skip adder, which considers the digital inputs for an operation is described.

In the proposed system, we use the Kogge-Stone adder. It is a parallel prefix form carry look-ahead adder. For the proposed work parallel prefix network is included in between the stages of RCA. This parallel prefix network combined with processing units is referred to as nucleus stage. The modified CSKA structure with parallel prefix network is shown in Figure 1. In this structure consist of two processing networks and parallel prefix adder network. The processing networks are used for different purposes. First one is used for pre-processing step and the second one is used for post-processing step. An input from the second RCA block is given to the parallel prefix network and the output of this network is fed to the third RCA network. This is a main modification in the proposed system. The critical path from the first stage of RCA to the last stage of RCA is termed as Long latency path (LLP). The path from the first stage of RCA to post processing network is termed as Short latency path 1 (SLP1) and the path from preprocessing unit to the last stage of RCA is termed as Short latency path 2 (SLP2).These two paths are considered for a calculation of critical path in the overall structure.

There are different types of parallel prefix structure namely, Brent-Kung adder, Han –Carlson adder, Lynch Swartzlander, and Kogge-Stone adder. In the previous case [13], Brent-Kung adder is used but it has a drawback such as fan-out problems. Here in our proposed system, we use Kogge stone adder. The Kogge-stone adder was developed by Peter M. Kogge and Harold S. Stone. This is one type of parallel prefix adder. It has lower fan-out problems compared with other parallel prefix adders (PPAs). Figure 2. shows the example of 8-bit Kogge –stone adder. In this structure, each vertical stage generates a propagate and generate outputs.

The carries are generated vertically and these output bits are XOR'd with the initial propagate bits after the input to produce the sum output. This is a more efficient adder especially the power consumption of this adder is significantly low when compared to other adders. As shown in below adder the carry input is given into first vertical stage, this first stage performs the XOR'd operation. A carry may be considered as 0 or 1.In this example, it is taken as 0. The functioning of Kogge-Stone adder (KSA) is divided into three parts. These are1) Pre-processing, 2) Carry look ahead network, 3) Post processing. In a Pre-processing step, propagate and generate signal bits are generated for the pair input bits A and B.

$P_i = A_i$ XOR B_i (1)
$G_i = A_i$ AND B_i (2)

A carry looks ahead block differentiates Kogge-Stone adder from others. This is the main difference to make this adder has more high performance. This step produce

carries for each bit and it uses group propagate and generates given by the equations,

Pi:j = Pi:k-1 AND Pk:j (3)
Gi:j = Gi:k+1 OR (Pi:k+1 AND Gk:j) (4)

Post processing is the final step of this KSA and it is also common to all adders of this carry look ahead families. Sum bits are generated by using below equation,

Si=Pi XOR Ci-1 (5)

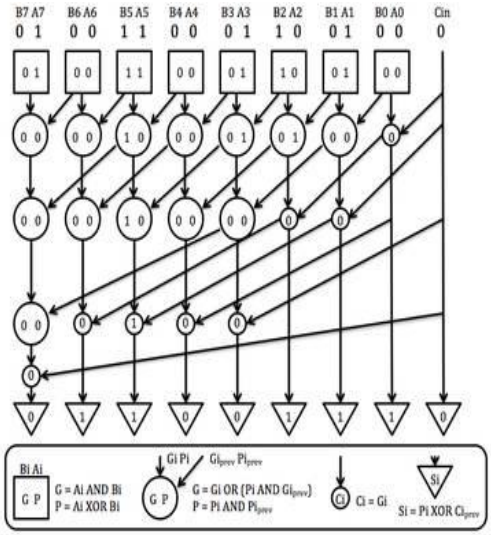

Figure 2: 8-bit Kogge-Stone adder structure

Kogge-Stone adder is widely used adder because it generates carry in O (log N) time. In more industries, this adder is used for high performance.

4. Proposed CSKA for Optical Circuits

This section describes the design and working of reversible carry skip adder based on Mach-Zehnder interferometer.

A. Basics of Reversible Logic Circuits

Among the emerging computing paradigms, reversible logic is a promising technology due to its wide applications. Reversible logic is also being investigated for its promising applications in power-efficient nanocomputing. Reversible circuits do not lose information during computation and reversible computation can be performed only when the system consists of reversible gates. These circuits can generate unique different output value from each input vector value and vice versa, that is, there is a one-to-one mapping between the input vector values and the output vectors values.

The design of reversible logic gates like NOT, k-CNOT, Toffoli, Fredkin, Peres may be possible in many ways. From the quantum technology point of view, the basic quantum gates such as NOT, CNOT, V and V+ are used to implement the reversible gates. In optical domain, Mach-Zehnder interferometer (MZI) based optical switches are used to implement optically reversible gates. An optical MZI switch can be designed using the following components: two Semiconductor Optical Amplifiers (SOA-1, SOA-2) and two couplers (C-1, C-2) as shown in Figure 3. MZI switch has two inputs ports namely, A and B and two output ports known as bar port and cross port, respectively. The optical signals coming from port B and port A at the input side are the control signal (2), and the incoming signal (1), respectively. The working principle of an MZI is explained as follows.

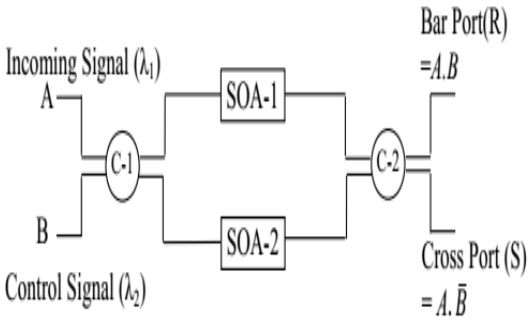

Figure 3: Semiconductor optical amplifier based MZI

When both incoming optical signal at port A and control signal at port B are high (i.e. A=1, B=1), then the light will appear at the output bar port and no light is seen at the cross port. Again, due to the absence of control signal at input port B and the presence of an incoming signal at input port A (A=1, B=0), then the light blink at the output cross port and no light appears at the output bar port. In all other cases, (i.e. A=0, B=1 and A=0, B=0), no light appears at output bar port and output cross port. The values for the absence and the presence of light are denoted by 0 and 1, respectively. From the perspective of Boolean functions, the above behavior of MZI switch can be written as R (Bar Port) =A.B and S (Cross Port) =AB'.

Beam combiner (BC) simply combines the optical beam while the beam splitter (BS) splits the beam into two optical beams. According to [11-12], the optical cost and the delay of beam combiner and beam splitter are very negligible. Hence, while calculating the optical cost of a circuit, they are assumed to be zero.

As the optical cost of BS and BC is comparatively small, the optical cost of a given circuit is the number of MZI switches required to design the realization. Optical delay is estimated as the number of stages of MZI switches multiplied by a unit Δ.

B. Structure of Proposed CSKA

In this section, we present all-optical design of n-bit CSA circuit using MZI based optical devices and interconnects. We have mapped the digital circuit of the CSA module to its equivalent optical structure. The specialty of our design is that we have used the minimum number of optical devices to implement it and also have configured the hardware design in such a manner that it requires a minimum number of clock pulses to respond. Another concern is that we have made both the designs functionally reversible so that the circuit dissipates minimum energy when they are activated. Two different architectures of CSA circuit have been presented, where the second design

is more delay efficient (nearly 50%) compared to the first one, but while achieving this efficiency, the optical cost of the second design has increased by 30%.In both the design, initially, the optimized architecture of fully optical 2-bit and 4-bit CSA module has been constructed. Then, using this small modules n-bit design is made. We have mapped the digital circuit of the CSA module to its equivalent optical structure. The specialty of our design is that we have used the minimum number of optical devices to implement it and also have configured the hardware design in such a manner that it requires a minimum number of clock pulses to respond. Another concern is that have made both the designs functionally reversible so that the circuit dissipates minimum energy when they are activated.

The entire design of the n-bit CSA circuit is divided into two phases. In the initial phase, an optimized 2-bi bit reversible CSA is designed using MZI based optical devices and in the second phase, this design is generalized for n-bit CSA where several 2-bit CSA modules are integrated.

The proposed design of 2-bit reversible CSA circuit is presented in Figure 4. It consists of 6 MZI switches, 6 beam splitters, and 4 beam combiners which incur the optical cost six. The entire CSA circuit requires two clock cycles to synchronize (as the design has two levels and is shown by dotted boxes in Figure 4, where four MZIs in the first level and two MZIs in the second level are connected in parallel). Hence it can be said that the optical delay of the entire module is 2Δ and the optical cost of this block is six as the 2-bit CSA module consists of six MZI switches. A 2-bit CSA performs addition of two 2-bit binary numbers. Let the pairs of binary numbers be A(a_1a_0) and B(b_1b_0), where the initial value of input carry bit C_0 is assumed to be zero. Apart from sum bit (Si) and carry bit (Ci) as shown in Figure 4, the CSA circuit has carry generator (Gi = ai.bi) and carry propagator (Pi = ai⊕bi), where ai and bi are the i^{th} cell input bits. The logic expression corresponding to Si and Ci is expressed as Si = Pi⊕Ci and Ci+1= Gi+PiCi. Using above equations, we deduce the following relations; S_0= $P_0⊕C_0$, and $C_1=G_0+P_0C_0$. As C_0 = 0, hence $S_0 = P_0⊕0= P_0$, $C_1 = G_0 + 0 = G_0$ and $S_1 =P_1⊕C_1$, $C_2 = G_1+P_1C_1$.

The operation principle of the proposed design is discussed here. Consider two n-bit binary numbers, A (an-1an-2...a2a1a0) and B (bn-1bn-2...b2b1b0) to be added using a n-bit reversible CSA. Two additional signals are used in this design that are Gj:i = Gj+PjGj-1+PjPj-1Gj-2+...+ PjPj-1Pj-2 ... Pi+1Gi and Pj:i = PjPj- 1Pj-2 ... Pi+1Pi, where Gj:i and Pj:i are group generate and group propagates signals from ith bit to jth bit. Now, the carry out expression at MSB position is obtained using these groups generate and groups propagate values. So, C_{j+1} = Gj:i + Pj:iCi, where the index variable j and i signify the MSB and LSB bit positions of a particular block. In the design, for each block we are skipping one intermediate carry bit Ci , where i ∈ 1,3,5,..., n-1 and this carry bits are calculated from Ci-1 carry inputs but the interesting fact is that C_{i+1th} carry bit is not obtained using Ci but using C_{i-1}. For example, the carry input to 2nd block is $C_2=G_1+P_1G_0$ and the carry output is $C_4= G_3+P_3G_2+P_3P_2C_2$ which does not contain carry bit C_3. So, it can be seen that intermediate carry bit C_3 is skipped.

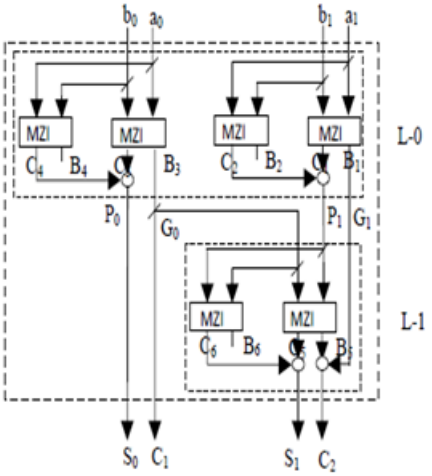

Figure 4.2 Bit carry skip adder

Here, we analyze the working principle of the blocks (total n/2 CSA blocks) in the n-bit design. First block: LSB position value of output sum is S_0 = $P_0⊕C_0$, where $P_0=a_0⊕b_0$ and input carry bit $C_0=c_0=0$. So, sum bit S_0 $=P_0⊕0=P_0$. Again, the MSB position value in output sum is S1 $=P_1⊕ C_1$, where $P_1 =a_1⊕ b_1$ and carry bit is C1 = G_0 + P_0C_0, where $C_0=c_0=0$ and $G_0 = a_0b_0$. Hence we can write $C_1 = G_0 +0=G_0$. For second block, the sum and carry bit expressions are: $S_2 = P_2⊕ C_2$, where $P_2 =a_2⊕ b_2$ $C_2 = G_1 + P_1G_0+P_2P_1C_0$, where $C_0=c_0=0$ and $G_1=a_1b_1$ or C_2 can be expressed as $C_2 = G_1 + P_1G_0$. In the initial phase, we have designed a 4-bit CSA module and later have integrated several such modules to design the n-bit CSA circuit. The basic design of 4-bit CSA module is shown in Figure 3, which is consisting of twenty MZIs and incurs a total optical cost of twenty. The entire circuit requires four clock pulses to synchronize it. So, the optical delay of the 4-bit CSA module is 4Δ.

The proposed design as shown in Figure 4, needs exact (3+ (n/4− 1) + 1) clock pulses to synchronize the entire circuit. The group generates (G_k) and group propagates (P_k) signal for all the n/4 blocks are calculated in the first clock pulse. To compute both the product expressions P_kG_{k-1} and $P_{k+1}Pk$, (where k ∈ 1,2,3, … , n/4) which are required to calculate the carry bit for all n/4 blocks needs another clock cycle and is computed in second clock pulse.

In third clock cycle, the three carry bits C_2, C_3 and C_4 in the first block (B1) are calculated simultaneously to propagate carry bit C_4 to the 2nd block (B_2). The n-bit CSA calculates four carry bits $C_{4i},C_{4i-1},C_{4i-2},C_{4i-3}$ (where i∈ 2,3…n/4) simultaneously in each block except the first one where the pre-assigned value of input carry (C0) is 0 and performs three more parallel operations in the same clock pulse. The operations are as follows: first parallel operation is the propagation of MSB carry bit C4i of block Bi to the input carry bit $C_{4(i+1)-3}$ of block Bi+1 ; in second parallel operation, the three consecutive sum bits $S_{(4i-5)}$, $S_{(4i-6)}$ and $S_{(4i-7)}$ of block B $_{(i-1)}$ are computed using three carry bits $C_{4(i-1)\ 1}$,$C_{4(i-1)-2}$,$C_{4(i-1)-3}$ which are generated in the

block B(i-1); the third parallel operation is a calculation of sum bit S_{4i-4} of block B_i using C4(i- 1) of Bi-1 and this entire operation is repeated (n/4− 1) times for n/4 blocks which require a total (n/4− 1) clock pulses. In the last single clock cycle, the three sum bits Sn-1, Sn-2 , Sn-3 of block Bi(where i = 4) are obtained.

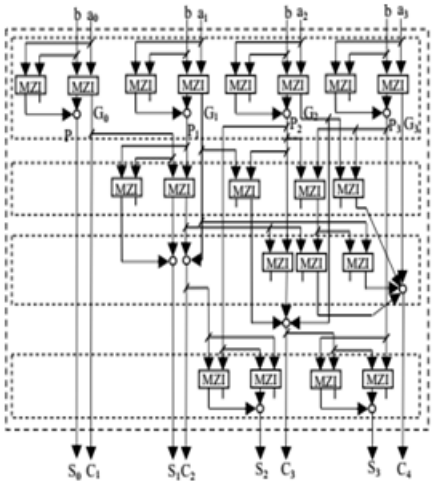

Figure 5: 4 bit carry skip adder

5. Simulation Result

In this section, the simulation outputs of the proposed carry skip adder are explained. The carry skip adder can be simulated using Modelsim 6.4a and synthesized using ISE design suite. The below Figure 6 shows the output of proposed high-speed carry skip adder. The output of this high-speed CSKA also gives same output of ordinary adders. But by using this modified high-speed carry skip produce the output with minimum delay time. This output taken for overall output of modified high speed carry skip adder. Here the values are given only for A and B. The carry may be either zero or one. The design functionality has been verified using Xilinx ISE design suite 14.5.

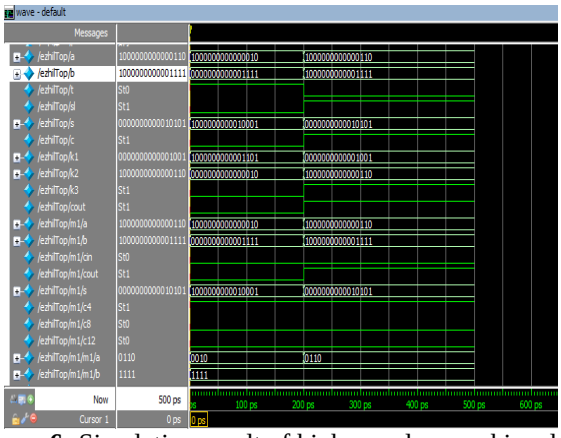

Figure 6: Simulation result of high speed carry skip adder

The optical reversible carry skip adder was implemented by coding a structural description in Verilog HDL. All the designs are synthesized with the Xilinx Synthesis Tool and Simulated using Xilinx ISE simulator. Simulation result for 4-bit carry skip adder is shown in Figure 7.

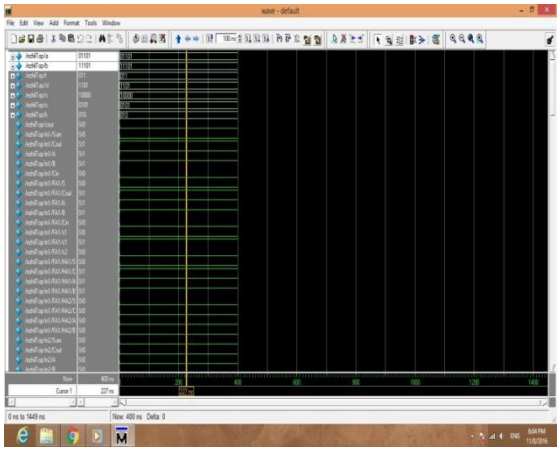

Figure 7: Simulation result MZI based carry skip adder

The below Table 1 shows the comparison table of conventional CSKA, High speed CI-CSKA and Proposed modified CSKA.

Table 1: Comparison table of existing CSKA, Proposed CI-CSKA and reversible CSKA

Parameters	Existing CSKA	Proposed CSKA using kogge-stone adder	Proposed reversible CSKA using MZI
No of bounded IO's	50	34	27
Delay (ns)	6.212	3.905	1.95
Power (mW)	46.12	42.38	38.25

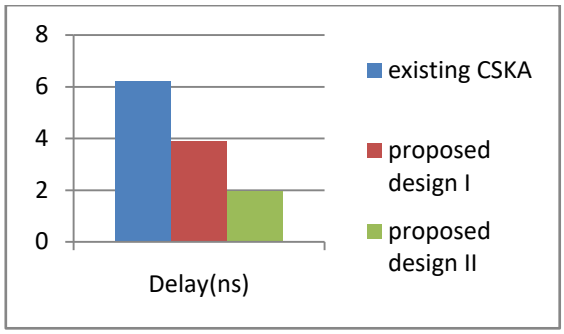

Figure 8: Delay comparison

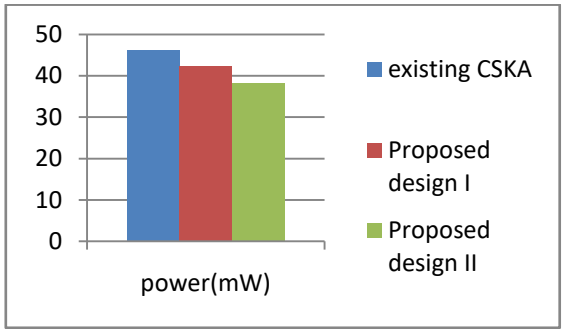

Figure 9: Power Analysis

Above Figure 8 and 9 shows the delay and power comparison of existing and two proposed CSKA. It is clearly shows that CSKA using MZI technique provides delay efficient structure. The delay can be reduce 50% when compare to the second design.

6. Conclusion

In this paper, a structure called high speed carry skip adder for digital circuit and optical reversible carry skip adder for optical circuits was proposed. In this work, various architectures of MZI based functionally reversible all-optical carry skip adder have been proposed. As far as our knowledge is concerned, the design of reversible all-optical adder is a newer one. The proposed design techniques implement all the optical functionally reversible counters with a minimum number of optical delay and minimum optical cost. Comparative analysis with existing carry skip adder circuits shows that both the designs are cost and delay efficient compared to the existing designs. But proposed optical reversible carry skip adder is more efficient than the proposed high-speed carry skip adder.

References

[1] Milad Bahadori, Mehdi Kamal and Ali AfzaliKusha, "High-speed and energy efficient carry skip adder operating under a wide range of supply voltage levels", IEEE Transaction on Very Large Scale Integration (VLSI) Systems, 2015, Vol.67, no.4, pp.324-335.

[2] R. Zlatanovici, S. Kao, and B. Nikolic, "Energy–delay optimization of 64-bit carry lookahead adders with a 240 ps 90 nm CMOS design example," IEEE J. Solid-State Circuits, vol. 44, no. 2, pp. 569–583, Feb. 2009.

[3] S. K. Mathew, M. A. Anders, B. Bloechel, T. Nguyen,R. K. Krishnamurthy, and S. Borkar, "A 4-GHz 300-mW 64-bit integer execution ALU with dual supply voltages in 90-nm CMOS," IEEE J. Solid-State Circuits, vol. 40, no. 1, pp. 44–51, Jan. 2005.

[4] V. G. Oklobdzija, B. R. Zeydel, H. Q. Dao, S. Mathew, and R. Krishnamurthy, "Comparison of high-performance VLSI adders in the energy-delay space," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 13, no. 6, pp. 754–758, Jun. 2005.

[5] B. Ramkumar and H. M. Kittur, "Low-power and area-efficient carry select adder," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 20, no. 2, pp.371–375, Feb. 2012

[6] M. Vratonjic, B. R. Zeydel, and V. G. Oklobdzija, "Low- and ultra low-power arithmetic units: Design and comparison," in Proc. IEEE Int. Conf. Comput. Design, VLSI Comput. Process. (ICCD), Oct. 2005, pp. 249–252.

[7] C. Nagendra, M. J. Irwin, and R. M. Owens, "Area-time-power tradeoffs in parallel adders," IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process., vol. 43, no. 10, pp. 689–702, Oct. 1996.

[8] Y. He and C.-H. Chang, "A power-delay efficient hybrid carrylookahead/carry-select based redundant binary to two's complement converter," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 55, no. 1, pp. 336–346, Feb. 2008.

[9] C.-H. Chang, J. Gu, and M. Zhang, "A review of 0.18 μm full adder performances for tree structured arithmetic circuits," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 13, no. 6, pp. 686–695, Jun. 2005.

[10] R. Zimmermann, "Binary adder architectures for cell-based VLSI and their synthesis," Ph.D. dissertation, Dept. Inf. Technol. Elect. Eng., Swiss Federal Inst. Technol. (ETH), Zürich, Switzerland, 1998

[11] M. Lehman and N. Burla, "Skip techniques for high-speed carry propagation in binary arithmetic units," IRE Trans. Electron. Comput, vol. EC-10, no. 4, pp. 691–698, Dec. 1961

[12] Maroju SaiKumar, Dr. P. Samundiswary "Design and Performance Analysis of Various Adders using Verilog", IJCSMC, Vol**. 2,** Issue. 9, September 2013

[13] Koren, Computer Arithmetic Algorithms, 2nd ed. Natick MA, USA: A K Peters, Ltd. , 2002.

[14] R. W. Keyes and R. Landauer, "Minimal energy dissipation in logic", IBM Journal of Research and Development, pp. 152–157, March2001

[15] A. Poustite and K. Blow, "Demonstration of an all-optical Fredkin gate", Optics Communications, 174:317–320, 2000