Journal of Advances in Computer Research
Quarterly ISSN: 2008-6148
Sari Branch, Islamic Azad University, Sari, I.R.Iran
(Vol. 11, No. 2, May 2020), Pages: 93-104
www.jacr.iausari.ac.ir

Modular Parallel-Prefix Adder and Multiplier design based on Reversible Computing

Ailin Asadpoor[1], Amir Sabbagh Molahosseini * [1], Azadeh Alsadat Emrani Zarandi[2]
1) Department of Computer Engineering, Kerman Branch, Islamic Azad University, Kerman, Iran
2) Department of Computer Engineering, Shahid Bahonar University of Kerman, Kerman, Iran
ailinasadpoor@iauk.ac.ir; sabbagh@iauk.ac.ir; a.emrani@uk.ac.ir

Received: 2020/10/28; Accepted: 2021/01/09

Abstract

Power consumption in today's modern high-performance computing systems is one of the most important design issues. Reversible computations have attracted a lot of attention compared to classical calculations due to their ability to reduce energy loss and power consumption of circuits. The reversible logic has applications in various technologies such as quantum circuits, low power circuit design, nanotechnology, optical information processing, DNA and bioinformatics calculations. Adder and multiplier are the main parts of any computing systems and therefore play an important role in the performance of reversible computations. Features of a reversible ripple-carry adder are high quantum depth, low quantum cost, low garbage outputs, and low constant input bits. In this paper, a new reversible kogge-Stone parallel-prefix adder and multiplier is proposed for modulo 2n±1. The analysis shows that the reversible-logic parallel-prefix adder and multiplier are faster and have the lowest depth compared to the reversible-logic-based ripple-carry adder and multiplier.

Keywords: Reversible Logic, Modular adder, Modular multiplier, Parallel-Prefix Adder

1. Introduction

Energy loss can be attributed to different sources; however, some energy remains in the system due to the irreversibility of classical computing. In circuits designed by irreversible logic, the number of circuit inputs is higher than outputs. Therefore, throughout processing, some bits containing information are lost and their electrical energy is released as heat. Reversible gates have been studied by R. Landauer since 1960 and the main purpose in designing reversible gates was very low heat production (factually they are heat-free) [1]. The heat generated in irreversible logical calculation for every bit of information is $KT´\ \ln 2$, where k is the Boltzmann constant and T is the temperature. At room temperature the amount of heat consumed is low, about $(2.9´\ 10^{-21}$ joules$)$ [2]. This value, on a small scale, seems insignificant but it cannot be ignored, because, according to Moore's Law," the number of transistors and elements in digital circuits increases nearly 100% every 18 months" in near future; consequently, the loss of electrical energy and the resulting heat generation will be the major challenges in designing large-scale integrated circuits[3,4]. In 1993, Bennet showed that by reversible calculation, the energy consumed by $KT´\ \ln 2$ due to loss of information is eliminated in reversible process. In reversible circuits that have reverse circuit, the direction of calculation can be considered as inverse. This means that with output on the reverse circuit, the input of the circuit can be reached or stopped and returned to at any point in calculation history [5]. According to Moore's Law, by halving the size of the transistors over a period of 18 months now the digital world seeks a suitable solution for designing nano scale circuits [6], and even smaller ones, to meet the incoming needs of processing systems [3]. Therefore, it is expected that all technologies in the future use reversible logic and quantum processing to reduce energy loss as well as the size of computational elements. In reversible design, there are three criteria to measure the efficiency and complexity of a circuit: the quantum cost, the depth and the number of garbage output [7]. On the other hand, in a computer arithmetic unit, two addition and multiplication operations are considered to be the main operations. To make quantum computers, it is necessary to design reversible addition and multiplication units. Therefore, this paper investigates the design and evaluation of modular Parallel prefix adder and multiplier arithmetic circuits by reversible circuits. The adder circuit is one of the most important digital circuits offered to reduce energy loss in a reversible form. Using adders, more complex circuits can be designed including subtractor, multiplier and divider. Adders have different types such as Ripple Carry Adder(RCA) [8], Carry Skip Adder (CSKA) [9], Carry Look ahead Adder (CLA) [10],Carry Select Adder (CSLA) [8],Carry Save Adder (CSA)[11], Parallel Prefix Adder (PPA) [12], [13]. This paper describes three efficient types of adders, namely ripple carry adder, carry save adder, and parallel prefix adder which is the fastest adder type. The criteria to evaluate reversible circuits are quantum cost, quantum depth, and the number of constant inputs and garbage outputs. By comparing these designs based on circuit complexity, the most appropriate type of adders can be determined. This paper is organized as follows: Section 2 briefly introduces reversible gates and their properties and gives an overview of modular computing. Section 3 presents a new design of parallel prefix adder circuit building blocks for modulo 2^n+1. Similarly, a new design of the parallel prefix multiplier circuit in $2^n \pm 1$ modulo is suggested using reversible logic and a new method of modular calculations in the reversible logic is presented. In Section 4, the reversible Kogge-Stone parallel prefix multiplier proposed in Section 3 has been improved to reduce the cost. Finally, conclusion has been presented in Section 5.

2. Reversible gates and modular computing

In this section, first the features of reversible gates such as quantum cost and depth are presented, then RCA and CSA for moduli $2^n - 1$and 2^n+1will be described.

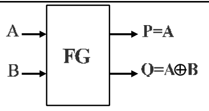
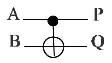
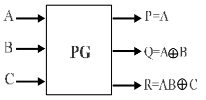
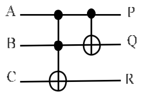
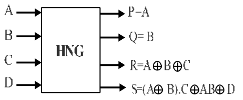
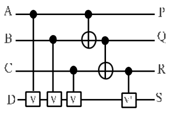
2.1. Reversible gates

In a system, computation is reversible if reversible gates are used. A reversible gate creates a one-to-one mapping between inputs and outputs [14]. Information is lost when inputs cannot be retrieved from outputs. For this purpose, the number of inputs and outputs of reversible circuits is the same [15]. In reversible logic, feedback from output to input is not possible and there is no Fan out [16]. Therefore, all reversible gates are also reversible [17]. In these circuits, if the set of inputs is shown with I and the set of gate outputs for these inputs is shown with O [18]:

$I=(I_i,I_{i+1},...,I_k)$ & $O=(O_i,O_{i+1},...,O_k)$

There is a relationship between each input and output $(I_u \rightarrow O_i)$. When a reversible gate is introduced as $n \times n$, it means that this gate has n inputs and n outputs that some outputs cannot be used. Garbage outputs include logic functions that are not used either as acceptable output to the circuit or as inputs to other classes of circuit for designers. In optimized designs, one of the purposes is to reduce these outputs [17].

Table 1. Reversible gate; quantum cost (QC), depth(D) [19]

Reversible gates	Quantum implementation	QC	D
		1	1
		4	2
 A B C D HNG P=A Q=B R=A⊕B⊕C S=(A⊕B).C⊕AB⊕D		6	4

2.2. Adder circuits for moduli $2^n \pm 1$

Computations in modulo 2^n-1 and 2^n+1 are used in various applications, such as residue number system (RNS) and encryption [20]. Fast and efficient modular adders and multipliers are prerequisites for integrated circuits with high performance [21].

A is a n-bit binary number $A=a_{n-1}a_{n-2}...a_0$ which are given below [22]:

$$A=\sum_{i=0}^{n-1} 2^i a_i$$

The reduction of A to modulo M (A mod M) can be done by a division (with residue as a result) or by repeated subtraction of the modulo until $A<M$ [23]. For 2^n-1 and 2^n+1 modules, the modular reduction of an A number with a maximum of $2n$ bits can be done simply by addition or subtraction. The addition to modulo 2^n-1 or the addition to complement one can be easily obtained by (1) [24].

$$(A+B)\bmod(2^n-1)=\begin{cases} A+B-(2^n-1)=(A+B+1)\bmod 2^n & \text{if} \quad A+B\geq 2^n \\ A+B & \text{otherwise} \end{cases} \tag{1}$$

For the addition to the modulo 2^n+1, a reduced number system is often used. So that the number A is represented by $A'=A-1$ and the value of zero is not used or examined separately. The usual addition to this numbering system is as follows [5]:

$$\begin{aligned} &A+B=S \\ &(A'+1)+(B'+1)=(S'+1) \\ &A'+B'+1=S' \end{aligned} \tag{2}$$

The modular addition of 2^n+1 can be formulated as follows:

$$(A'+B'+1)\bmod(2^n+1)=\begin{cases}A'+B'+1-(2^n+1)=(A'+B'+1)\bmod 2^n & \text{if}\quad A'+B'\geq 2^n\\ A'+B'+1 & \text{otherwise}\end{cases}\qquad(3)$$

If $A'+B'<2^n$ hence, if $C_{out}=0$, the sum of $A'+B'$ increases by one unit. Therefore, the modular addition of 2^n+1 can be achieved by CSA-EAC with $C_{in}=\overline{C}_{out}$ [24].

$$(A'+B'+1)\bmod(2^n+1)=(A'+B'+\overline{C}_{out})\bmod 2^n \qquad (4)$$

The CSA-CEAC adder, which uses n FAs to calculate the sum, is more cost-effective than other adders. In contrast, the RCA-CEAC adder is composed of n FAs and n HAs with the lowest speed. The fastest adder is parallel prefix, which carry out using a network, independent of other computations [25].

3. New reversible design of modular parallel prefix adder and multiplier circuits

Logic and optimization performance parallel prefix adders are used for high-speed adders. A detailed study of various parallel prefix adders is presented in [25]. In this section, a method for designing reversible parallel prefix adders has been proposed. Then a new design is proposed for the reversible modular Kogge-Stone parallel prefix adder. In a similar way, a reversible modular Kogge-Stone parallel prefix multiplier has been offered for two different modules using a single hardware.

3.1. Parallel prefix adder

If A and B are two n-bit sequences of $A_{n-1}\ldots A_1A_0$ and $B_{n-1}\ldots B_1B_0$, the binary sum is defined by (5) [19].

$$S_i=a_i\oplus b_i\oplus c_{i-1}\quad,\qquad c_i=a_ib_i+a_ic_{i-1}+b_ic_{i-1}\qquad(5)$$

In this way the Prefix addition is done in three steps [19].

Step 1: Pre-processing

Pre-processing involves generating and producing signals. According to parallel prefix calculations, the g_i (generation) and p_i (propagation) signals are defined by (6).

$$g_i=a_ib_i\quad,\quad p_i=a_i\oplus b_i\qquad(6)$$

Step 2: Calculate Prefix

The structure of the PPA is dependent on the concept of the carrier signal generation and propagation signals. The set of generating and propagating signals is defined by (7).

$$G_{[i:k]}=\begin{cases}g_i\ , & \text{if}\ \ i=k\\ G_{[i:j]}+P_{[i:j]}.G_{[j-1:k]}, & \text{otherwise}\end{cases}$$

$$P_{[i:k]} = \begin{cases} p_i \,, & \text{if } i = k \\ p_{[i:j]}.P_{[j-1:k]}, & \text{otherwise} \end{cases} \tag{7}$$

To simplify the representation of G and P, a multiplication operator is shown as * to represent the generation and propagation group and is defined by (8).

$$(G,P)_{[i:k]} = (G,P)_{[i:j]} * (G,P)_{[j-1:k]} \tag{8}$$

Step 3: Post- processing

The step after processing is to form sum and carry bits for each bit of operand. The equations ci and Si are defined as follows.

$$c_i = G_{[i:0]} S_i = p_i \oplus c_{i-1} \tag{9}$$

3.2. The Kogge-Stone parallel prefix adder for moduli $2^n \pm 1$

Parallel prefix reversible adder circuits have high quantum cost but lower quantum depth compared to reversible RCA adder. Parallel prefix adders have been widely used in high performance adders due to logarithmic delay and efficient implementation in VLSI and have attracted much attention in specific application architectures and general purposes. As shown in Figure 1, the overall structure of all parallel-prefixes is the same in the pre-processing and post-processing steps, and the only difference is in the number of black and gray cells in the middle step, which causes differences in depth and cost of parallel-prefixes.

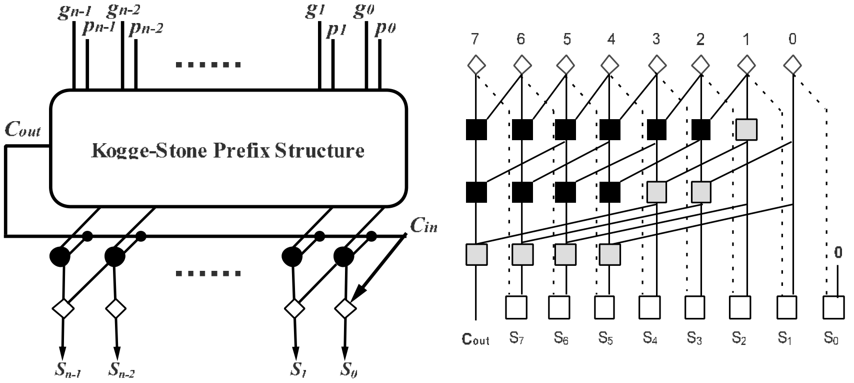


Figure 1.left side: the parallel prefix adder structure for modulo $2^n - 1$ and right side: the Kogge-Stone parallel prefix structure.[26]

For the Kogge-Stone parallel-prefix adder for modulo $2^n - 1$, the same method as the parallel-prefix Brent-Kung adder in [26], can be used by adding a row of black cells at the end of the adder to prepare the carry out in the middle step. To design a Kogge-Stone parallel prefix adder for modulo $2^n + 1$given the Diminished-One (D-O) design [27], just we have to compute the inversion of the $c_{out}$of Kogge-Stone parallel prefix structure to modulo $2^n - 1$ in Figure 1(left side). Because the modular addition of $2^n + 1$ can be achieved by end around carry parallel prefix adder with $C_{in} = \overline{C}_{out}$.

3.3. The reversible Kogge-Stone parallel prefix adder for moduli $2^n \pm 1$

The design of the gray and black cells of the Kogge-Stone parallel prefix adder to modulo $2^n - 1$is similar to the Brent-Kung parallel prefix adder to modulo $2^n - 1$ [26]. In this paper, using the D-O method for the first time to design the Kogge-Stone parallel prefix to modulo $2^n + 1$, the reversible FG (Feynman Gate) is used (Figure 2).

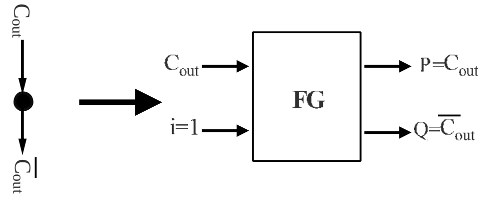


Figure 2. NOT cell in Kogge-Stone parallel prefix adder modulo $2^n + 1$

According to Figure 2, to produce $\bar{c}_{out}$ at the output, it is sufficient that the input of the FG control is a constant value of 1. Note that, the quantum depth of the proposed has been computed using RcViewer tool [28]. The calculation of Kogge-Stone parallel prefix parameters to modulo $2^n + 1$ is as follows.

- Quantum cost: $4N + (\text{no.of blackcells}´\ 9) + (\text{no.of graycells}´\ 4) + N + 4N + 1$
- Quantum depth:
$(4 + (\text{no.of blackcells in critical path}´\ 5) + (\text{no.of graycells in critical path}´\ 4) + 1 + 4 + 1)D$
- Number of constant input: $N + (\text{no.of blackcells}´\ 2) + (\text{no.of graycells}´\ 0) + 0 + 0 + 1$
- Number of garbage output: $N + (\text{no.of blackcells}´\ 2) + (\text{no.of graycells}´\ 1) + N + 2N + 1$

3.4. The reversible Kogge-Stone parallel prefix multiplier for moduli $2^n \pm 1$ with a single hardware

The modular multiplication is done by the modular addition of partial products (PP_i's). To calculate the modular multiplication, PP_i must first be generated. The calculation method of PP_is for moduli $2^n - 1$ and $2^n + 1$ is shown in Tables 2 and 3, respectively.

Table 2. Calculate PP_is to modulo $2^n - 1$ [29]

	2^{n-1}	2^{n-2}	…	2^1	2^0
PP_0=	$P_{n-1,0} \vee P_{n,n-1} \vee P_{n-1,n}$	$P_{n-2,0}$	…	$P_{1,0}$	$P_{0,0} \vee P_{n,n}$
PP_1=	$P_{n-2,1}$	$P_{n-3,1}$	…	$P_{0,1}$	$P_{n-1,1} \vee P_{n,0} \vee P_{0,n}$
PP_2=	$P_{n-3,2}$	…	…	$P_{n-1,2} \vee P_{n,1} \vee P_{1,n}$	$P_{n-2,2}$
⋮	…	…	…	…	…
PP_{n-2}=	$P_{1,n-2}$	$P_{0,n-2}$	…	$P_{3,n-2}$	$P_{2,n-2}$
PP_{n-1}=	$P_{0,n-1}$	$P_{n-1,2} \vee P_{n,n-2} \vee P_{n-2,n}$	…	$P_{2,n-1}$	$P_{1,n-1}$
PP_n=	0	0	…	0	$P_{n,n-1} \vee P_{n-1,n}$

The PP_is calculation for modulo $2^n + 1$ is similar to PP_is calculation for modulo $2^n - 1$ except that it uses $å_{i=1}^{n-1} i$, NOT gate number.

Table 3. Calculate PP_is modulo $2^n + 1$ [29]

	2^{n-1}	2^{n-2}	…	2^1	2^0
$PP_0=$	$P_{n-1,0} \vee P_{n,n-1} \vee P_{n-1,n}$	$P_{n-2,0}$	…	$P_{1,0}$	$\overline{P_{0,0} \vee P_{n,n}}$
$PP_1=$	$P_{n-2,1}$	$P_{n-3,1}$	…	$P_{0,1}$	$\overline{P_{n-1,1} \vee P_{n,0} \vee P_{0,n}}$
$PP_2=$	$P_{n-3,2}$	…	…	$\overline{P_{n-1,2} \vee P_{n,1} \vee P_{1,n}}$	$\overline{P_{n-2,2}}$
⋮	…	…	…	…	…
$PP_{n-2}=$	$P_{1,n-2}$	$P_{0,n-2}$	…	$\overline{P_{3,n-2}}$	$\overline{P_{2,n-2}}$
$PP_{n-1}=$	$P_{0,n-1}$	$\overline{P_{n-1,2} \vee P_{n,n-2} \vee P_{n-2,n}}$	…	$\overline{P_{2,n-1}}$	$\overline{P_{1,n-1}}$
$PP_n=$	0	0	…	1	$\overline{P_{n,n-1} \vee P_{n-1,n}}$

As shown in Tables 3 and 4, to calculate the reversible PP_is, we implement the AND gate by PG and the OR gate by FrG (Fredkin Gate).So in the generation step of PP_is to calculate the cost, the number of PGs must be multiplied by the cost of PG and the number of FrGs multiplied by the cost of FrG, plus the number of FGs to obtain the final cost at the generation step of PP_is. Similarly, all calculations are done in parallel because the inputs are independent of each other. The calculation of the parameters for the generation step of PP_i is as follows:

- Quantum cost: $4 \times [(n-2)+5+(n-1)(n+2)+2]+5 \times [2(n-1)+4]+\sum_{i=1}^{n-1} i$
- Quantum depth: $10D$ (This value obtained with the RcViewer simulation).
- Number of constant input: $(n-2)+5+(n-1)(n+2)+2+2(n-1)+4+\sum_{i=1}^{n-1} i$
- Number of garbage output: $2 \times ((n-2)+5+(n-1)(n+2)+2+2(n-1)+4)+\sum_{i=1}^{n-1} i$

To calculate the modular multiplication,PP_is which are produced in Tables 3 and 4, are added together using CSA-EAC or CSA-CEAC in one hardware. In order to control the operation, we use the T signal, which determines the moduli of addition (for modulo $2^n - 1$, and T=0 for modulo $2^n +1$, T=1) [29].

$$A=\sum_{i=0}^{n-1} 2^i a_i \qquad ,B=\sum_{j=0}^{n-1} 2^j b_j \tag{10}$$

$$\left|A \times B\right|_{2^n-1} = \left|\sum_{i=0}^{n-1} 2^i a_i \times B\right|_{2^n-1} = \left|\sum_{i=0}^{n-1} 2^i a_i \times (b_{n-i-1} \ldots b_0 b_{n-1} \ldots b_{n-i})\right|_{2^n-1} = \left|\sum_{i=0}^{n-1} PP_i\right|_{2^n-1} \tag{11}$$

$$\left|A \times B\right|_{2^n+1} = \left|\sum_{i=0}^{n} 2^i a_i \times \sum_{j=0}^{n} 2^j b_j\right|_{2^n+1} = \left|\sum_{i=0}^{n} (\sum_{j=0}^{n} PP_{i,j} 2^{i+j})\right|_{2^n+1} \tag{12}$$

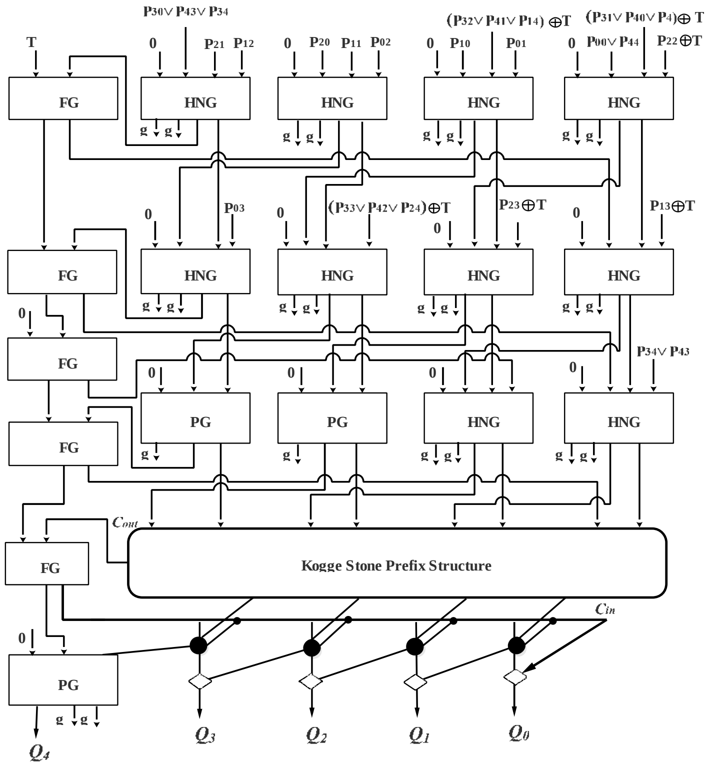


Figure 3.The sum of PP_is using CSA-EAC and CSA-CEAC for moduli $2^n - 1$ and $2^n + 1$ respectively for n = 4

The calculation of the parameters for Figure 3 is as follows (the quantum depth of the proposed design has been computed using RcViewer tool [28]):

- Quantum cost: 6´ [(n - 2)´ n + 2] + 4´ [n - 2] + 4n + (no.of black cells´ 9)´ (no.of gray cells´ 4) + n + 4n + (n + 1) + 4
- Quantum depth: ((5 + ((n - 2)´ 3) + 1 + 4 + (no.of black cells in criticl path´ 5)´ (no.of gray cells in criticl path´ 4) + 1 + 4 + 3)D
- Number of constant input: (n - 2)´ n + 2 + (n - 2) + n + (no.of black cells´ 2) + (no.of gray cells´ 0) + 0 + 0 + 1
- Number of garbage output: 2´ ((n - 2)´ n + 2) + (n - 2) + n + (no.of black cells´ 2) + (no.of gray cells´ 1) + n + 2n + 2

In reversible modular multiplication, a combination of reversible gates has been used in an innovative way. As shown for faster applications where speed is more important than cost, the Kogge-Stone parallel prefix method is justified; of course, the disadvantage of using this method is high cost. Similarly, the RCA-EAC method can be used for cases where cost is a more important parameter than speed.

4. The design proposed to improve cost in reversible parallel prefix multiplier circuits

To solve the parallel prefix cost problem, hybrid circuit model is used. These hybrid prefix circuits are not only structures for constructing fast parallel prefix circuits at optimum depth, but are also themselves fast problem-size-independent prefix circuits[30]. When the size of the problem is larger than the width of the circuit, hybrid parallel prefix circuits provided may be much faster than any other parallel prefix circuit of the same width, especially when the size of the problem is larger than or equal to twice the width of the circuit. Also these circuits have the least depth and are the fastest of all parallel prefix circuits with the same width and fan-out. In this type of hybrid circuits, data is entered in parallel (8 bits) and each time a number of parallel prefix nodes are calculated and sent to the next level. For this reason, after calculating the first level and going to the next level, eight bits are entered and repeated to perform all levels of operation; therefore, the speed in parallel prefix parallelization increases significantly. The interesting thing is that after calculating every eight bits, the carry bit is sent to the next batch. This way, the first 9 bits are imported. The first bit is used to calculate the input carry digit, eight bits are used later. We suggest that in this model of the hybrid parallel-prefix circuit will be further improved if we implement the Kogge-Stone parallel-prefix method, because the Kogge-Stone prefix method has a lower depth than the other prefix methods, so it will certainly be faster. The only defect of reversible modular Kogge-Stone parallel prefix circuit is the high cost of the circuit, which the cost parameter will be lowered significantly with this hybrid model (Table 4).

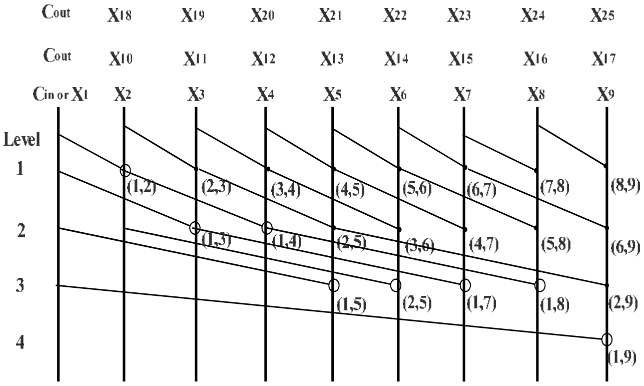


Figure 4. Kogge-Stone parallel prefix structure using hybrid model

For example, according to figure 4, in a Kogge-Stone parallel prefix circuit with 24 input nodes, the circuit can be modified to use a Kogge-Stone parallel prefix circuit with 9 input nodes.

In this method, the first input line is used as the input carry digit bit, and the eight carry digit for next steps. That is, the line c_{in} is set as x_1 and in the next steps, this line is used only as the c_{out} input of the previous step. In the second step c_{in} will be $x_{(1,9)}$, and in the third step c_{in} will be $x_{(1,17)}$. In reversible modular Kogge-Stone parallel prefix multipliers (Figure 3), the appropriate depth is obtained but the cost values, are large numbers, therefore it is suggested that smaller hardware be used in this section to lower costs.

Table 4. Comparison of Kogge-Stone Parallel Prefix Adder and Multiplier Performance Based on Reversible Logic

Adders and Multipliers		Circuit Parameters															
		Quantum Cost				Quantum Depth (Δ)				Constant Inputs				Garbage Outputs			
		8	16	32	64	8	16	32	64	8	16	32	64	8	16	32	64
Regular Adders [26]	Kogge-Stone	158	446	1166	2894	19	24	29	34	28	84	228	580	43	115	291	707
	RCA	48	96	192	384	27	51	99	195	8	16	32	64	16	32	64	128
Modular Proposed Adders	Kogge-Stone with EAC	190	510	1294	3150	23	28	33	38	28	84	228	580	59	147	355	835
	RCA with EAC[26]	80	160	320	640	51	99	195	387	16	32	64	128	25	49	97	193
	Kogge-Stone with CEAC	191	511	1295	3151	24	29	34	39	29	85	229	581	60	148	356	836
Modular Proposed Multipliers	Kogge-Stone with EAC & CEAC	906	3398	13095	46862	61	90	143	241	215	771	2875	10987	398	1442	5122	19570
	Improve Kogge-Stone w. EAC & CEAC	906	906	906	906	61	180	572	1928	215	215	215	215	398	398	398	398

5. Conclusion

This paper presents the reversible design of the modular adder and multiplier, which are essential elements in computation. The reversible parallel-prefix adder moduli $2^n \pm 1$ presented in this paper has designed in a manner to have the minimum overhead than reversible regular parallel-prefix adders. Besides, quantum cost, quantum depth, constant inputs and garbage outputs of the proposed reversible modular parallel-prefix adder are calculated for various bit-width including N = 8,16,32, and 64 bits and compared with the reversible regular adder. The main feature of the proposed reversible multiplier is that it uses the same primitive hardware of the reversible multiplier at N = 8, for more Ns. Therefore, the cost values, constant inputs and garbage outputs remain the same and the depth value changes slightly.

References

[1] R. Landauer, “Irreversibility and heat generation in the computing process”,IBM J.Research and Development, 1961, pp. 183- 191.

[2] R. Singh and M.K. Pandey, “Design and optimization of sequential counters using a novel reversible gate”, IEEE International Conference on Computing, Communication and Automation (ICCCA), 2016.

[3] T.M. Conte, E.P. DeBenedictis, P.A. Gargini, and E. Track, “Rebooting Computing: The Road Ahead,” Computer, Vol. 50, No. 1, 2017, pp. 20-29.

[4] Sh. Thakral, D. Bansal, “A Novel Reversible DSG Gate and Its Quantum Implementation”, Springer, 2019.

[5] H. P. Sinha, N. Syal, “Design of Fault Tolerant Reversible Multiplier”, International Journal of Soft Computing and Engineering (IJSCE), Vol. 1, 2012.

[6] H.A. Mousavi, P. Keshavarzian, A.S. Molahosseini, “A novel fast and small XOR-base full adder in quantum-dot cellular automata,”Applied Nanoscience,2020.

[7] M. Noorallahzadeh and M. Mosleh, “Efficient designs of reversible latches with low quantum cost”, IET Circuits, Devices & Systems,Vol. 13, 2019, p. 806 – 815.

[8] T.-Y. Chang and M.-J. Hsiao, “Carry-select adder using single ripple-carry adder”, IET, Vol. 34, 1998, pp. 2101 – 2103.

[9] K. Chirca ,M. Schulte ,J. Glossner ,Horan Wang ,B. Mamidi ,P. Balzola and S. Vassiliadis, “A static low-power, high-performance 32-bit carry skip adder”, IEEE, 2004.

[10] R.W. Doran, “Variants of an improved carry look-ahead adder”, IEEE Transactions on Computers, Vol. 37, 1988, pp. 1110 – 1113.

[11] T. Kim , W. Jao and S. Tjiang, “Circuit optimization using carry-save-adder cells”, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 17, 1998, pp. 974 – 984.

[12] A. Beaumont-Smith and C.-C. Lim, “Parallel prefix adder design”, Proceedings 15th IEEE Symposium on Computer Arithmetic, 2002.

[13] B. Koyada, N. Meghana, Md. O. Jaleel and P. R. Jeripotula, “A comparative study on adders”, IEEE WiSPNET 2017.

[14] A. Zulehner and R. Wille, “Taking one-to-one mappings for granted: Advanced logic design of encoder circuits”, Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017.

[15] A. Niknafas, “Optimized Reversible Programmable Logic Array (PLA)”, Journal of Advances in Computer Research (JACR), Vol 4, No.1, pp. 81-88, 2013.

[16] D. Maslov, Y. Nam and J. Kim, “An Outlook for Quantum Computing,”Proceedings of the IEEE, vol. 107, no. 1, pp. 5-10, 2019.

[17] S.M.R. Taha, “Reversible Logic Synthesis Methodologies with Application to Quantum Computing”, Springer, 2016.

[18] O. Coudert and J. Christophe Madre, “A unified framework for the formal verification of sequential circuits”,springer, pp 39-50, 2003.

[19] A.A.E. Zarandi, A.S. Molahosseini, M. Hosseinzadeh, S. Sorouri, S.F. Antão and L. Sousa, “Reverse Converter Design via Parallel-Prefix Adders: Novel Components, Methodology and Implementations,” IEEE Transactions on Very Large Scale Integration (VLSI) systems, Vol. 2, No. 374-378, 2015, p. 23.

[20] A.S.Molahosseini, L.Sousa and C.H. Chang (Eds.), “Embedded Systems Design with Special Arithmetic and Number Systems”, Springer, 2017.

[21] Sh. Taghipour Eivazi, “Low Complexity Converter for the Moduli Set $\{2^{n+1},2^{n-1},2^n\}$ in Two-Part Residue Number System”, Journal of Advances in Computer Research (JACR), Vol.10, No.3, pp. 31-39, 2019.

[22] Sh. Taghipour Eivazi, “Efficient Reverse Converter for Three Moduli Set $\{2^n\text{-}1,2^{n+1}\text{-}1,2^n\}$ In Multi Part RNS”, Journal of Advances in Computer Research (JACR), Vol.8, No.4, pp. 87-94, 2017.

[23] J. Beuchat, “Some modular adders and multipliers for field programmable gate arrays,”In Proc. ofInternational Parallel and Distributed Processing Symposium, Nice, France, 2003, pp. 8.

[24] R. Zimmermann, "Efficient VLSI implementation of modulo 2n±1 addition and multiplications", Proceedings IEEE Symposium on Computer Arithmetic (ARITH), 1999.

[25] C. Vudadha, and et. al., “Design and Analysis of Reversible Ripple, Prefix and Prefix-Ripple Hybrid Adders,” In Proc. of IEEE Computer Society Annual Symposium on VLSI (ISVLSI), 2012.

[26] A.S. Molahosseini, A.Asadpoor, A.A.E. Zarandi and L.Sousa, "Towards Efficient Modular Adders based on Reversible Circuits", IEEE International Symposium on Circuits and Systems (ISCAS),2018.

[27] E. Vassalos , D. Bakalis , H.T. Vergos, "Reverse converters for RNSs with diminished-one encoded channels", IEEE Eurocon ,2013.

[28] D. Maslov, W. Dueck,and N. Scott, "Reversible logic synthesis benchmarkpage", http://www.cs.uvic.ca/~dmaslov/, 2018.

[29] S. Menon ,C.H. Chang, "Reconfigurable Modulo 2n±1 Multipliers", IEEE Asia Pacific Conference on Circuits and Systems, 2006.

[30] Y.Ch. Lin, L.L. Hung,"Fast problem-size-independent parallel prefix circuits", Vol. 69, 2009, pp. 382-388.