Adder Designs using Reversible Logic Gates

P. K. LALA*, J.P. PARKERSON**, P. CHAKRABORTY**
* Electrical Engineering Department, Texas A&M University-Texarkana, Texas 75503, USA
** Computer Science and Computer Engineering Department, University of Arkansas, Fayetteville, Arkansas, 72701, USA
plala@tamut.edu

Abstract: A new reversible logic gate was proposed in Ref. [1]. This gate can be used to implement any classical Boolean logic function. This paper shows the application of the reversible gate in implementing ripple carry, carry skip and carry look-ahead adders. These adders are more efficient than adders implemented using Fredkin Gates

Key-Words: reversible logic, Fredkin gate, garbage output, adder design

1. Introduction

In modern VLSI systems power dissipation is very high due to rapid switching of internal signals. It has been shown that for every bit of information lost in logic computations that are not reversible, *kT*log2* joules of heat energy is generated, where *k* is Boltzmann's constant and *T* the absolute temperature at which computation is performed [2]. In fact zero power dissipation in logic circuits is possible only if a circuit is composed of reversible logic gates [3].

Reversible gates have applications in nuclear magnetic resonance (NMR) quantum computation, quantum dot cellular automata (QCA), and optical computing [A, B, C].

A gate is considered to be reversible only if for each distinct input there is a distinct output assignment. Thus inputs to reversible gates can be uniquely determined from its outputs. A reversible logic gate must have the same number of inputs and outputs [3]. In an *n*-output reversible gate the output vectors are permutation of the numbers 0 to 2^n-1. A reversible gate is *balanced*, i.e. the outputs are 1s for exactly half of the inputs. A circuit without constants on its inputs and composed of reversible gates realizes only balanced functions. It can realize non-balanced functions only with garbage outputs. Some of the major problems with reversible logic synthesis are fan outs cannot be used, and also feedback from gate outputs to inputs are not permitted [3]. In this paper the implementation of three types of adder circuits using the reversible gate R are presented [1].

2. Reversible gates

Several reversible gates have been proposed over the years, e.g., the Toffoli gate, the Fredkin gate etc. [34]. A 3-input and 3-output reversible logic gate was proposed in [1]. It has inputs *a, b, c* and outputs *x, y and z* as shown in Fig 1. The truth table of the gate is shown in the Table 1. It can be verified from the truth table that the input pattern corresponding to a particular output pattern can be uniquely determined. The gate can be used to invert a signal and also to duplicate a signal. The signal duplication function can be obtained by setting input *b* to 0 as shown in Fig 2a. The EX-OR function is available at the output *x* of the gate. The AND function is obtained by connecting the input *c* to 0, the output is obtained at the terminal *z* as shown in Fig

2b. The implementation of a NAND gate is shown in Fig 2c. An OR gate is realized by connecting two new reversible gates as shown in the Fig 2d shown in the Fig 2d.

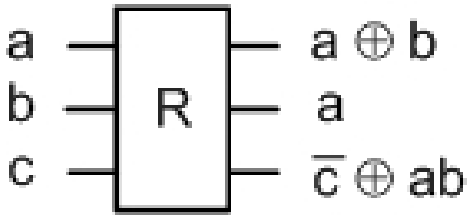
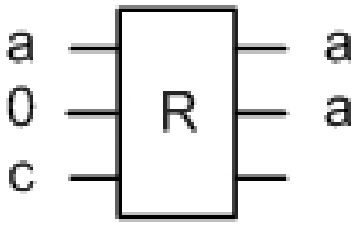


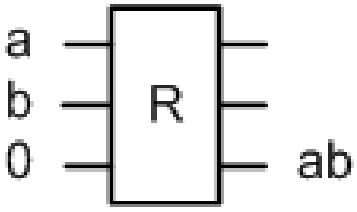
Fig. 1 Reversible gate R

A	*b*	*c*		*X*	*y*	*Z*
0	0	0		0	0	1
0	0	1		0	0	0
0	1	0		1	0	1
0	1	1		1	0	0
1	0	0		1	1	1
1	0	1		1	1	0
1	1	0		0	1	0
1	1	1		0	1	1

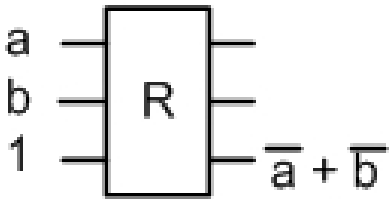
Table 1 Truth Table of the Reversible gate R



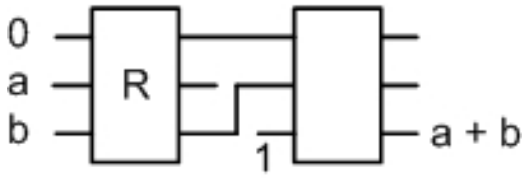
a) Signal copying



b) AND function



c) NAND function



d) OR function

Fig. 2 Different functions using the R gate

F
i

3. Adder circuits

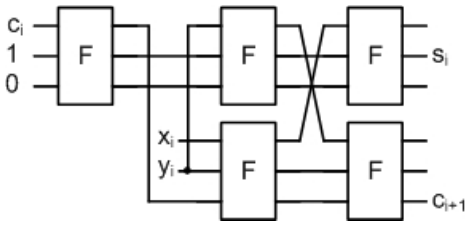
Several types of adders are used in computing systems [D, E]. A ripple carry adder has the simplest structure. In a ripple carry adder, full adders connected in series generate the sum and the carry outputs based on the addend bits and the carry input. The disadvantage of a ripple carry adder is that the carry has to propagate through all stages. Carry look-ahead adders are the fastest of all adders. They achieve speed through parallel carry computations, but employ a large number of gates. The carry skip adder presents a hardware and performance compromise between a ripple carry adder and a carry look-ahead adder. This paper shows the implementations of ripple carry adders, carry skip adders, and carry look-ahead adders using the reversible logic gate R of Fig.1. These adders are compared with similar types of adders using the Fredkin gate in [4].

3.1 Ripple carry adders

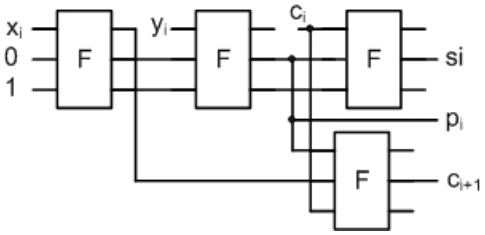
The basic building block of a ripple carry adder is a full adder block. A full adder computes the sum bit S_i and the carry output c_{i+1} based on addend inputs a and b and carry input c. The output expressions for a ripple carry adder are

(1) S_i = a xor b xor c;
(2) C_{i+1} = ab + bc + ca;
(i = 0,1,2,….)

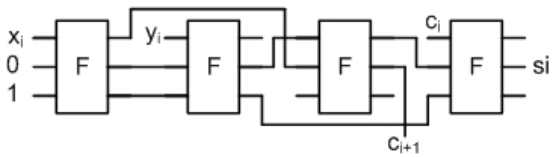
Fig 3a shows the ripple carry adder circuit implemented using Fredkin gates [3]. This circuit is improved by removing the fan out [4] as shown in Fig 3b. The number of garbage outputs in Fig 3b is 3. Sum output Si is obtained at the output of the fourth gate (from the left) and Carry output C_{i+1} is obtained at the third gate output. Fig 3c shows the ripple carry adder implementation using the reversible logic gate of Fig.1. It uses four gates and the number of garbage outputs is 5. The sum output S_i is obtained at the second gate output itself. The carry output is obtained at the fourth gate. Although the implementation of the adder using Fredkin Gate has fewer garbage outputs, it uses feedback signal from fourth to the third gate. The adder implemented with proposed gate does not use any feedback.



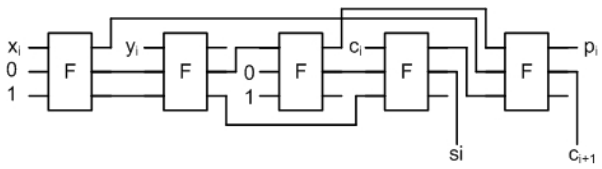
(a) Full Adder with fan out [4]



(d) Full Adder with propagate [4]

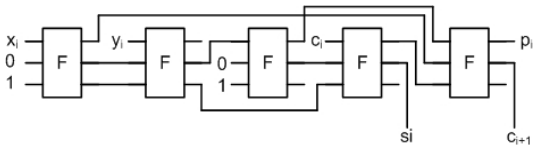


(b) Full Adder without fan out [5]

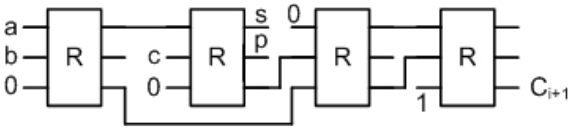


(e) Full Adder with propagate [5]

Fig.3 Full Adders



(c) Full adder without propagate using gate



(f) Full adder with propagate using gate

Fig.3 (continued) Full Adders

3.2 Carry skip adders

As stated earlier, carry skip adders reduce the carry computation delay. A full adder with a propagate signal is used as a building block in carry skip adders. In a full adder, if either input is a '1', the carry input is propagated to the carry output. Therefore, the carry input ci to the i^{th} full adder will propagate to its carry output C_{i+1} when $p = a \oplus b = 1$; such an adder is called a carry skip compatible adder [4]. As stated earlier, the fan out in Fig 3.d was eliminated in the design proposed in Ref [5] shown in Fig 3e. The design in Fig 3e used 5 Fredkin gates. The carry out and the propagate signals are obtained at the fifth gate output. The number of garbage outputs of this design is 4. The carry skip compatible full adder implemented using the reversible gate of Fig.1e is shown in Fig 3f. Notice that this adder circuit is identical to the adder circuit implemented without carry-propagate signal except that the garbage output g_2 in Fig 2c is used as propagate signal in Fig 2f. The number of garbage output used is as same as that used in Fig 2e. However, the number of gates used is only 4. Both the sum output S_i and the propagate output are obtained at the second gate output.

A 4-bit carry skip adder can be constructed by using four carry-propagate compatible full adders as shown in Fig 4a. The carry skip adder consists of four carry-propagate compatible full adders. The propagate signals p_0, p_1, p_2 and p_3 generated by each adder are ANDed. The resulting output is ANDed with carry input c_i. The corresponding output is ORed with carry output of the fourth full adder to get the carry output as shown in Fig 4a. The same implementation can also be implemented using the reversible logic gate as shown in Fig 4b. Four NFA blocks used in the design are the carry skip compatible full adder blocks designed using reversible gate shown in Fig 3f. The propagate signals are ANDed and the carry out is generated by ORing this output with the C3, carry output of the fourth NFA (New Full Adder).

3.3 Carry look-ahead adders

Carry look-ahead adders (CLA) are the fastest of all adders and achieve speed through parallel carry computations. For each bit in a binary sequence to be added, the CLA logic determines whether that bit pair will generate a carry or propagate a carry. This allows the circuit to "pre-process" the two numbers being added to determine the carry ahead of time. Then, when the actual addition is performed, there is no delay from waiting for the ripple carry effect. The adder is based on the fact that a carry signal will be generated in two cases:

1. When both bits Ai and Bi are 1, or
2. When one of the two bits is 1 and the carry-in is 1.

Thus, we can write:

(3) $C_{OUT} = C_{i+1} = A_i \cdot B_i + (A_i \oplus B_i) \cdot C_i$.

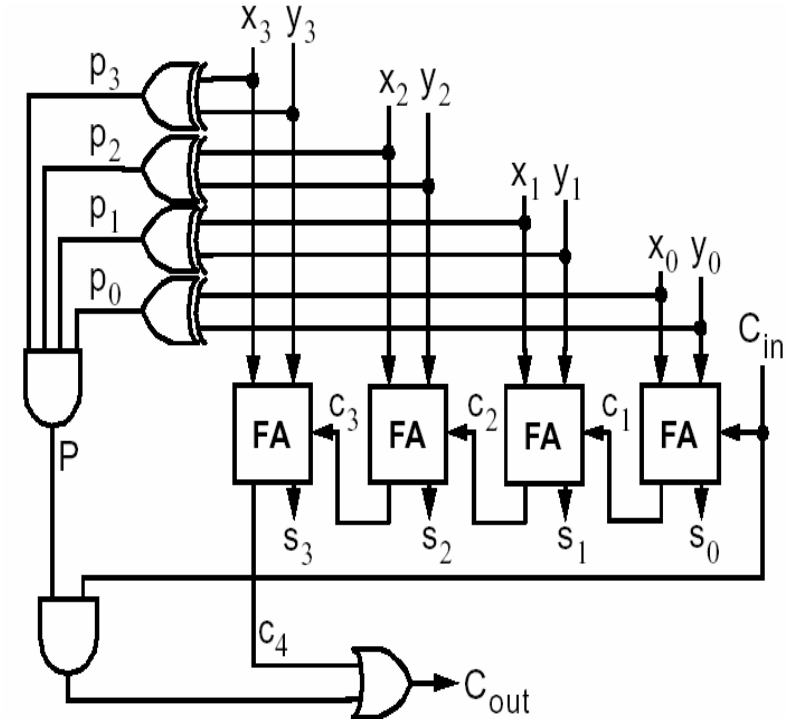


Figure 4 a 4-bit Carry skip Adder from Ref [5]

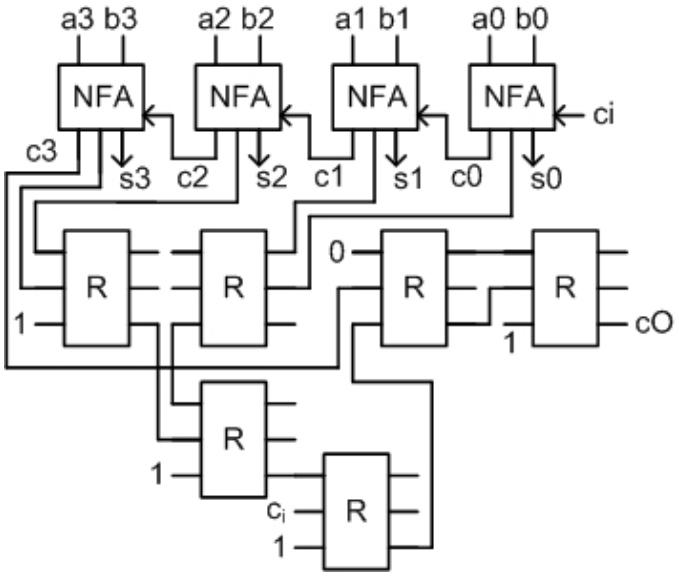


Figure 4b: 4-bit Carry skip Adder using proposed gate

The above expression can also be represented as:

(4) $C_{i+1} = G_i + P_i \cdot C_i$

Where, $G_i = A_i \cdot B_i$ and $Pi = A_i \oplus B_i$
Applying this to a 4-bit adder, we have:

(5) $C1 = G0 + P0\ C0$

(6) $C2 = G1 + P1\ C1$

$= G1 + P1\ G0 + P1\ P0\ C0$

(7) $C3 = G2 + P2\ C2$

$= G2 + P2\ G1 + P2\ P1\ G0$

$+ P2\ P1\ P0\ C0$

(8) $C4 = G3 + P3\ C3 = G3 + P3\ G2$

$+ P3\ P2\ G1 + P3\ P2\ P1\ G0$

$+ P3\ P2\ P1\ P0\ C0$

The Sum signal can be calculated as follows:

(9) $S_i = A_i \oplus B_i \oplus C_i = P_i \oplus C_i$

The CLA can be broken up into two modules:

1. Partial Full Adder (PFA): This generates Gi, Pi, and Si.
2. Carry Look-Ahead Logic: The CLA generates the carry-out bits as shown in Fig. 6

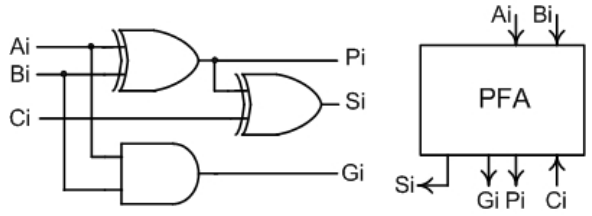


Figure 5: Partial Full Adder (PFA)

Thus, a 4-bit adder can then be built by using four PFAs and the carry look-ahead logic block. The complete architecture is as follows:

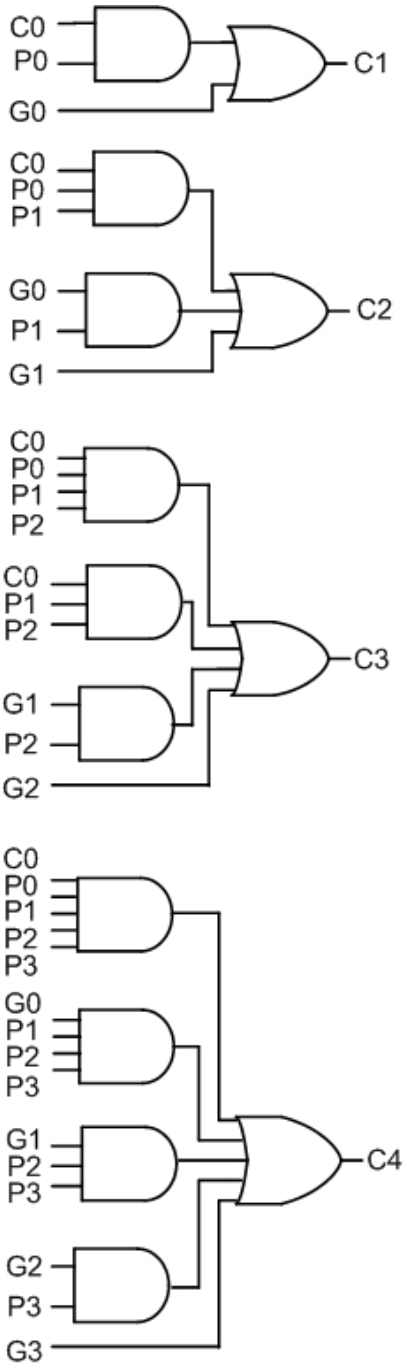


Figure 6: Carry Look-Ahead Logic

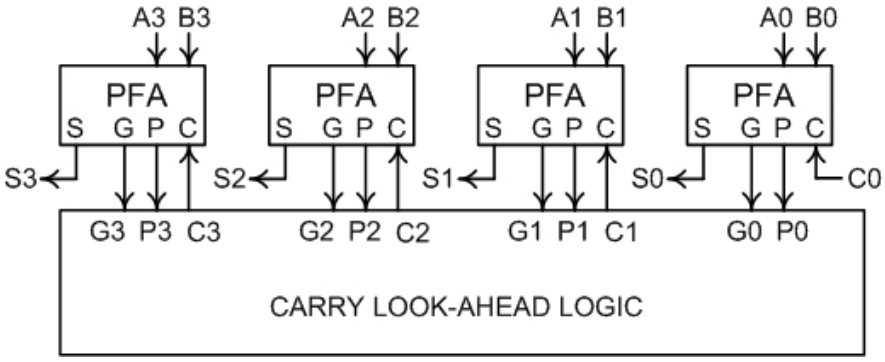


Figure 7: Block diagram of a 4-bit CLA

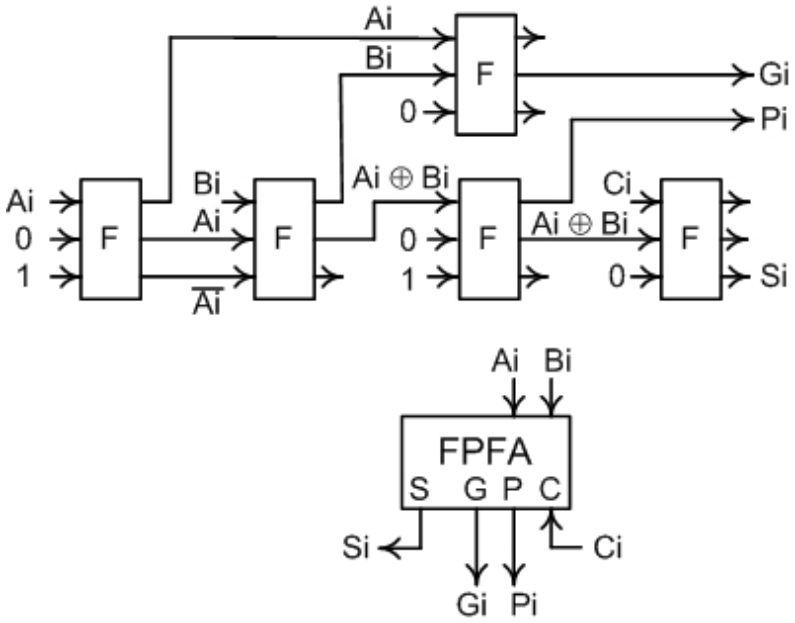


Figure 8: Partial Full Adder using Fredkin Gates (FPFA)

3.4 CLA using Fredkin Gates

As mentioned earlier, the Fredkin gate is a reversible 3-bit gate that swaps the last two bits if the first bit is 1. A 4-bit CLA can be implemented by using Fredkin gates. As in the case of an ordinary 4-bit CLA, the Fredkin CLA (FCLA) can be broken into 2 modules:

1. Partial Full Adder using Fredkin Fates (FPFA)
2 Carry Look-Ahead Logic using Fredkin Gates

3.4.1 PFA using Fredkin Gates

As evident from Figure 9, a partial full adder requires five Fredkin gates. This circuit has a critical path gate delay of 4. The gate delay for generating Pi is 3, for generating Gi is 2, and for generating Si is 4.

In the implementation of the carry look-ahead logic using Fredkin gates, 35 Fredkin

gates are used. The total number of garbage outputs produced is 55. The critical path delay (for generating C_4) is 11.

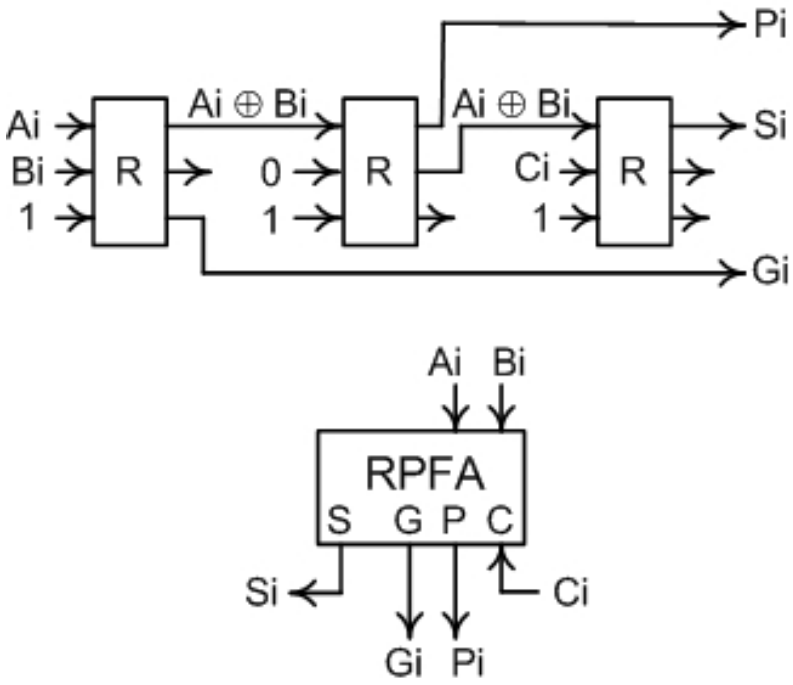


Figure 9: Partial Full Adder using R Gates (RPFA)

3.4.2 CLA using R Gates

As mentioned earlier, gate R is a new 3-input and 3-output reversible logic gate that was proposed in [3]. Gate R is shown in the Figure 1. A 4-bit CLA can be implemented by using R gates. As in the case of an ordinary 4-bit CLA (Figure 7), the R gate CLA (RCLA) can be broken into 2 modules:

1. Partial Full Adder using R Gates
2. Carry Look-Ahead Logic using R Gates

3.4.3 PFA using R Gates

As evident from Figure 9, a partial full adder requires only three R gates. This circuit has a critical path gate delay of 3. The gate delay for generating Pi is 2, for generating Gi is 1, and for generating Si is 3.

The comparison of the designs implemented using Fredkin Gate and the proposed gate is shown in Table 2. From the table it is evident that the adder circuits realized using the R gate in Fig.1 are more efficient.

4. Conclusion

The implementations of three types of adders, ripple carry, carry skip, and carry look-ahead, using a new reversible gate R is shown. Results show that the new gate results in more efficient adders in terms of gate count than those using the Fredkin gate.

	Fredkin Gate						Reversible Gate [Ref.1]					
	No of Gates	Garbage Output	Feedback	Gate Stage where output is obtained			No of Gates	Garbage Output	Feedback	Gate Stage where output is obtained		
				S_i	C_{out}	p				S_i	C_{out}	P
Full adder (with propagate)	5	4	Y	4^{th}	5^{th}	5^{th}	4	4	N	2^{nd}	4^{th}	2^{nd}
Full adder (without propagate)	4	3	N	3^{rd}	4^{th}	-	4	4	N	2^{nd}	4^{th}	-

Table 2. Comparisons of FA designs

References

[1] D.P.Vasudevan, P.K.Lala, J.Di and J.P.Parkerson, " Reversible-logic design with online testability", *IEEE Trans. on Instrumentation and Measurement*, vol.55., no.2, pp.406-414, April 2006.

[2] R. Landauer, "Irreversibility and Heat Generation in the Computing Process", *IBM J. Research and Development*, vol. 3, pp. 183-191,July 1961.

[3] C. H. Bennett, "Logical Reversibility of Computation", *IBM J. Research and Development*, pp. 525-532, November 1973.

[4] M Perkowski and P Kerntopf, "Reversible Logic", invited tutorial, Proc, EURO-MICRO,Warsaw, Poland Sept 2001.

[5] J.W. Bruce, M.A. Thornton, L. Shivakumaraiah, P.S. Kokate, and X. Li, "Efficient Adder Circuits Based on a Conservative Reversible Logic Gate," *Proc. of the IEEE Computer Society Annual Symposium on VLSI*, pp. 83-88, Pittsburgh, Pennsylvania, April 2002.

[6] J. Tamuliene, A. Tamulis, and V. Tamulis, "One Possible Way of Information Writing to Quantum Computer Device," 4th WSEAS Int. Conf. on Nanoelectronics and Nanotechnology (ICONN 2004), Howard Beach Resort Kenting, Taiwan, Jan. 14-16, 2004, pp 1611-1616.

[7] M. Janez, L. Bajec, P. Pecar, A. Jazbec, N. Zimic, and M. Mraz, "

Automatic Design of Optimal Logic Circuits Based on Ternary Quantum-dot Cellular Automata," WSEAS Trans. on Circuits and Systems, Issue 9, Vol. 7, Sept. 2008, pp 919-928.

[8] T. Rahman, M. Ahmed, and E. Saad, "All-Optical Arithmetic Unit Based on the Hardlimiters," Proc. Of the 6th Int. Conf. on Electronics, Hardware, Wireless and Optical Communications, Corfu Island, Greece, Feb. 16-19, 2007.

[9] B. Chean and R. Siferd, "High-Speed Hardware Effieient FIR Compensation Filter for Delta-Sigma Modulator Analog-to-Digital in 0.13 um CMOS Technology," 4th WSEAS Int. Conference on Electronics, Control, and Signal Processing, Miami, Florida, USA, Nov. 17-19, 2005, pp 1-5.

[10] P. Balasubramanian and N. Mastorakis, "High Speed Gate Level Synchronous Full Adder Designs," WSEAS Trans. on Circuits and Systems, Issue 2, Volume 8, Feb. 2009, pp 290-300.