ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Implementation of Parallel Prefix Adders Using Reversible Logic Gates

Lakkakula Karthik
ES&VLSID (M.E.)
Department of ECE
MVSR Engineering College, Nadergul

E.V.Nagalakshmi
Assistant Professor
Department of ECE
MVSR Engineering College, Nadergul

ABSTRACT:-

In Very Large Scale Integration (VLSI) designs, Parallel prefix adders (PPA) have the better delay performance. This paper presents four types of PPA's Reversible Kogge Stone Adder (RKSA), Reversible Spanning Tree Adder (RSTA), Reversible Brent Kung Adder (RBKA) and Reversible Sparse Kogge Stone Adder (RSKA). Additionally Ripple Carry Adder (RCA) is also investigated. These adders are implemented using Verilog Hardware Description Language (HDL) in Cadence 180nm technology, nc-sim for simulation, Simvision to view the waveforms and Encounter tool is used for synthesis. The parallel prefix-adder's Delay, Power and Area are Optimized and compared successively.

KEYWORDS:

Parallel Prefix Adders (PPA), Reversible Brent Kung Adder (RBKA), Reversible Kogge Stone Adder (RKSA), Reversible Spanning Tree Adder (RSPA), Reversible Sparse Kogge Stone Adder.

1. INTRODUCTION

The binary addition is the basic arithmetic operation in digital circuits and it became essential in most of the digital systems including Arithmetic and Logic Unit (ALU), Microprocessors and Digital Signal Processing (DSP). At present, the research continues on increasing the adder's delay performance. The choice of which adder architecture to use is of utmost importance, since the performance of adders may determine the whole system performance. Area and power consumption are also relevant figures of merit to be considered, especially when the design targets VLSI realization. Recently energy-efficiency has also become an important metric due to the dramatic growth of battery powered portable device marked. Parallel prefix adders have better performance. The delays of the adders are discussed. In this paper, above mentioned PPA's and RCA are implemented in Cadence 180nm technology. Finally, delay, power and area for the designed adders are presented and compared.

2. DRAWBACKS OF RIPPLE CARRY AND CARRY LOOKAHEAD ADDER

The first sum bit should wait until input carry is given; the second sum bit should wait until previous Carry is propagated and so on. Finally the output sum should wait until all previous carries are generated.

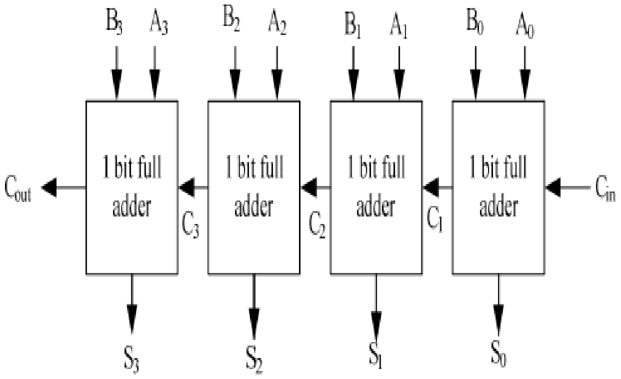


Fig1: Ripple Carry Adder

In order to reduce the delay in RCA an Carry Look Ahead Adder(CLA)is to propagate the carry in advance, we go for carry look ahead adder .Basically this adder works on two operations called propagate and generate The propagate and generate equations are given by.

$$P_i = A_i \text{ xor } B_i, \ldots\ldots.(1)$$

$$G_i = A_i .Bi, \ldots\ldots\ldots..(2)$$

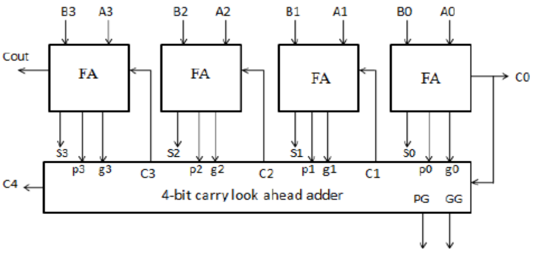


Fig.2. Carry Look Ahead Adder

For 4 bit CLA, the propagated carry equations are given as

$C_1 = G_0 + P_0 .C_0$,…………. (3)

$C_2 = G_1 + P_1\ C_1 = G_1 + P_1\ G_0 + P_1\ P_0\ C_0$………….(4)

$C_3 = G_2 + P_2\ C_2 = G_2 + P_2\ G_1 + P_2\ P_1\ G_0 + P_2\ P_1\ P_0\ C_0$,…………. (5)

$C_4 = G_3 + P_3\ C_3 = G_3 + P_3\ G_2 + P_3\ P_2\ G_1 + P_3\ P_2\ P_1\ G_0 + P_3\ P_2\ P_1\ P_0\ C_0$,………….(6).

Equations (3), (4), (5) and (6) are observed that, the carry complexity increases by increasing the adder bit width. So designing higher bit CLA becomes complexity. In this way, for the higher bit of CLA's, the carry complexity increases by increasing the width of the adder. So results in bounded fan-in rather than unbounded fan-in, when designing wide width adders. In order to compute the carries in advance without delay and complexity, there is a concept called Parallel prefix approach.

3. DIFFERENT TYPES OF PARALLEL-PREFIX ADDERS

The PPA's pre-computes generate and propagate signals are presented in [2].Using the fundamental carry operator (FCO), these computed signals are combined in [3].The fundamental carry operator is denoted by the symbol 'o'.$(g_L,p_L)o(g_R,p_R)=(g_L+p_L{\cdot}g_R\ ,\ p_L\ .\ p_R)$ (7)

For example, 4 bit CLA carry equation is given by

$C_4 =(g\ _4,p_4)o[(g_3,p_3)o[(g_4,p_4)o(g_3,p_3)]]$--(8)

For example, 4 bit PPA carry equation is given by $C_4 =[(g_4\ ,\ p_4\ )o\ (\ g_3\ ,\ p_3\ )]o\ [(\ g_4\ ,\ p_4\ )o\ (\ g_3\ ,\ p3\ )]$ (9)

Equations (8) and (9) are observed that, the carry look ahead adder takes 3 steps to generate the carry, but the bit PPA takes 2 steps to generate the carry.

Parallel-prefix structures are found to be common in high performance adders because of the delay is logarithmically proportional to the adder width [2]. PPA's basically consists of 3 stages as follows:

- Pre computation
- Prefix stage
- Final computation

The Parallel-Prefix Structure is shown in below

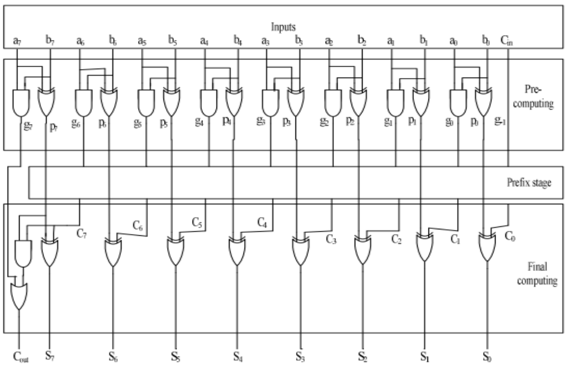


Fig3: Architecture of Parallel Prefix Adder.

A. Pre computation:

In Pre Computation Stage, the work of Propagation and Generation are computed for the given inputs using the basic equations (1) and (2).

B. Prefix stage:

In the prefix stage, group generate/propagate signals are computed at each bit using the given equations. The black cell (BC) generates the ordered pair in equation (7), the gray cell (GC) generates only left signal, following [2].

$$G_{i:k} = G_{i:j} + P_{i:j} . G_{j-1:k} \qquad (10)$$

$$P_{i:k} = P_{i:j} . P_{j-1:k} \qquad (11)$$

More practically, the equations (10) and (11) can be expressed using a symbol "o" denoted by Brent and Kung. Its function is exactly same as that of a Black Cell and Gray Cell.

C. Final computation:

In the final computation, the sum and carryout are the final output.

$$S_i = P_i . G_{i-1:-1} \tag{13}$$
$$C_{out} = G_{n:-1} \tag{14}$$

Where "-1" is the position of carry-input. The generate/propagate signals can be grouped in different fashion to get the same correct carries. Based on different ways of grouping the generate/propagate signals, different prefix architectures can be created.

4.1 Introduction to Reversible Logic Gates

Energy dissipation is one of the major issues in present day technology. Energy dissipation due to information loss in high technology circuits and systems constructed using irreversible hardware was demonstrated by R. Landauer in the year 1960. According to Landauer's principle, the loss of one bit of information lost, will dissipate kT*ln (2) joules of energy where, k is the Boltzmann's constant and k=1.38x10 -23 J/K, T is the absolute temperature in Kelvin. The primitive combinational logic circuits dissipate heat energy for every bit of information that is lost during the operation. This is because according to second law of thermodynamics, information once lost cannot be recovered by any methods. In 1973, Bennett, showed that in order to avoid kTln2 joules of energy dissipation in a circuit it must be built from reversible circuits. According to Moore's law the numbers of transistors will double every 18 months. Thus energy conservative devices are the need of the day. The amount of energy dissipated in a system bears a direct relationship to the number of bits erased during computation. Reversible circuits are those circuits that do not lose information. The most prominent application of reversible logic lies in quantum computers. A quantum computer will be viewed as a quantum network (or a family of quantum networks) composed of quantum logic gates; It has applications in various research areas such as Low Power CMOS design, quantum computing, nanotechnology and DNA computing. Quantum networks composed of quantum logic gates; each gate performing an elementary unitary operation on one, two or more two–state quantum systems called qubits. Each qubit represents an elementary unit of information; corresponding to the classical bit values 0 and 1. Any unitary operation is reversible and hence quantum networks effecting elementary arithmetic operations such as addition, multiplication and exponentiation cannot be directly deduced from their classical Boolean counterparts (classical logic gates such as AND or OR are clearly irreversible). Thus, quantum arithmetic must be built from reversible logical components. Reversible computation in a system can be performed only when the system comprises of reversible gates. A circuit/gate is said to be reversible if the input vector can be uniquely recovered from the output vector and there is a one-to-one correspondence between its input and output assignments. An N*N reversible gate can be represented as
Iv=(I1,I2,I3,I4,……………………IN)
Ov=(O1,O2,O3,…………………ON).
Where Iv and Ov represent the input and output vectors respectively. In quantum computing, by considering the need of reversible gates, a literature survey has been done and the mostly available reversible logic gates are presented in this paper.

4.2 Basic Definitions Pertaining To Reversible Logic

A. Reversible Function:

The multiple output Boolean function F(x1; x2; :::; xn) of n Boolean variables is called reversible if: (a) The number of outputs is equal to the number of inputs;
(b) Any output pattern has a unique pre-image.

B. Reversible logic gate:

Reversible Gates are circuits in which number of outputs is equal to the number of inputs and there is a one to one correspondence between the vector of inputs and outputs. It not only helps us to determine the outputs from the inputs but also helps us to uniquely recover the inputs from the outputs.

C. Ancilla inputs/ Constant inputs:

This refers to the number of inputs that are to be maintain constant at either 0 or 1 in order to synthesize the given logical function.

D. Garbage outputs:

Additional inputs or outputs can be added so as to make the number of inputs and outputs equal whenever necessary. This also refers to the number of outputs which are not used in the synthesis of a given function. In certain cases these become mandatory to achieve reversibility. Garbage is the number of outputs added to make an n-input k-output function ((n; k) function) reversible. We use the words —constant inputs to denote the present value inputs that were added to an (n; k) function to make it reversible. The following simple formula shows the relation between the number of garbage outputs and constant inputs. Input + constant input = output + garbage.

E. Quantum cost:

Quantum cost refers to the cost of the circuit in terms of the cost of a primitive gate. It is calculated knowing the number of primitive reversible logic gates (1*1 or 2*2) required to realize the circuit. The quantum cost of a circuit is the minimum number of 2*2 unitary gates to represent the circuit keeping the output unchanged. The quantum cost of a 1*1 gate is 0 and that of any 2*2 gate is the same, which is 1.

F. Flexibility:

Flexibility refers to the universality of a reversible logic gate in realizing more functions.

G. Gate Level:

This refers to the number of levels in the circuit which are required to realize the given logic functions.

H. Hardware Complexity:

This refers to the total number of logic operation in a circuit. Means the total number of AND, OR and EXOR operation in a circuit The following are the important design constraints for reversible logic circuits.

- Reversible logic gates do not allow fan-outs.
- Reversible logic circuits should have minimum quantum cost.
- The design can be optimized so as to produce minimum number of garbage outputs.
- The reversible logic circuits must use minimum number of constant inputs.
- The reversible logic circuits must use a minimum logic depth or gate levels

Goals of reversible logic:

1. Minimize the garbage outputs
2. Minimize the constant inputs
3. Minimize the total number of gates
4. Minimize the quantum cost

Several reversible logic gates have been proposed in the past years. Some of them are: Feynman gate(FG), Toffoli gate(TG), Fredkin gate(FRG), Peres gate(PG), New Gate(NG), TSG gate(TSG), MKG gate(MKG) and HNG gate(HNG).

In this section we review these reversible logic gates. Some of them are presented to allow for comparison with existing studies.

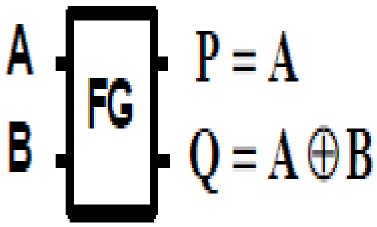


Fig 4. Feynman gate

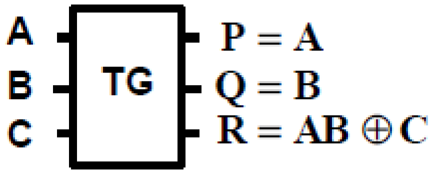


Fig 5. Toffoli gate

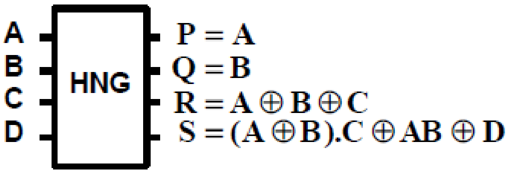


Fig.6.Reversible HNG gate

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

The reversible HNG gate can work singly as a reversible full adder. If the input vector IV = (A, B, Cin, 0), then the output vector becomes OV = (P=A, Q=Cin, R=Sum, S=Cout)

Reversible Parallel Prefix Adders are classified into
1. Reversible Kogge- Stone Adder
2. Reversible Brent-Kung Adder
3. Reversible Sparse- Kogge Stone Adder
4. Reversible Spanning Tree Adder

1.Reversible Kogge - Stone Adder

Reversible Kogge-Stone adder is a parallel prefix form carry look ahead adder. The Kogge-Stone adder [3] was developed by peter M. Kogge and Harold S. Stone which they published in 1973. In RKSA all conventional full adders are replaced with reversible HNG gate. Reversible Kogge-Stone prefix adder is a fast adder design.

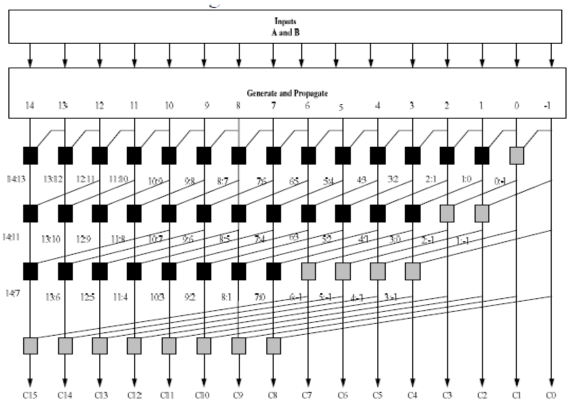


Fig 7: 16-bit Reversible Kogge Stone Adder

KS adder has best performance in VLSI implementations. Reversible Kogge-Stone adder has large area with minimum fan-out. The Reversible Kogge Stone Adder is widely known as a parallel prefix adder that performs fast logical addition. Reversible Kogge Stone adder is used for wide adders because it shows the less delay among other architectures. In each vertical stage produces Propagate and Generate bits. Generate bits are produced in the last stage and these bits are XORed with the initial propagate after the input to produce the sum bits.

2. Reversible Brent-Kung Adder

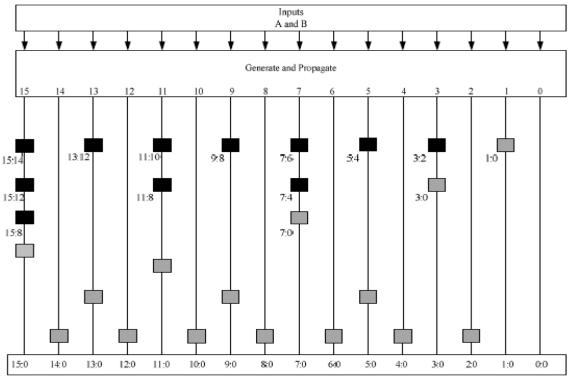


Fig 8: 16-bit Reversible Brent Kogge Adder

The ReversibleBrentKung adder is a parallel prefix adder.The BrentKung adder was developed by Brentand Kung which they published in 1982. Brent Kung adder has maximum logic depth and minimum area. The number of cells is calculated by using 2(n-1) -Log2n.The 16bit Reversible Brent Kung adder figures are shown below.

3. Reversible Spanning tree adder

RSTA is also tested. Like the RSKA, this adder also terminates with a RCA. It also uses the BC's and GC's and full adder blocks like RSKA's but the difference is the interconnection between them [7].The 16 bit RSTA is shown in the Fig 9.

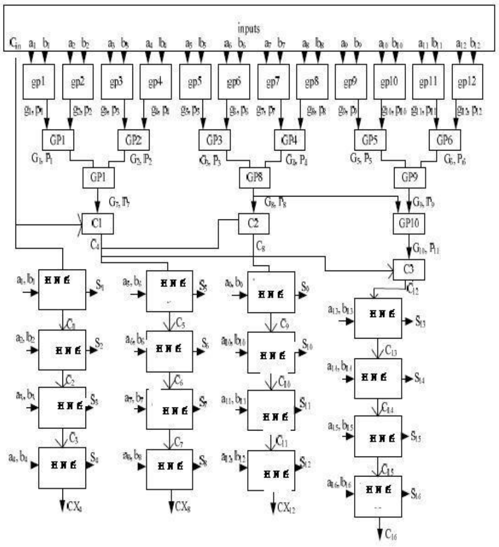


Fig 9: 16-bit Reversible Spanning tree adder

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

4. Reversible Sparse kogge stone adder

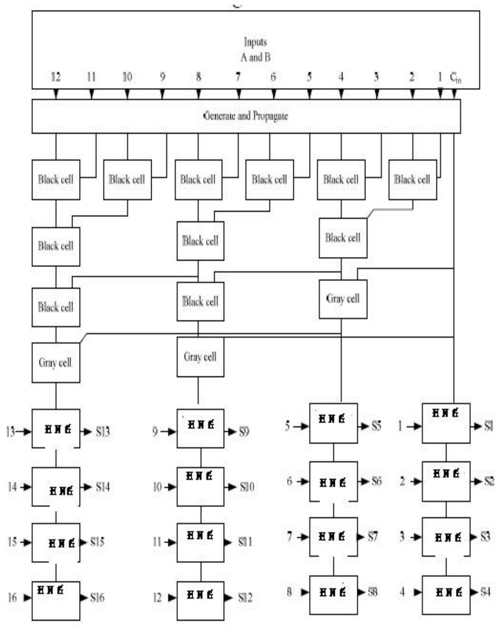


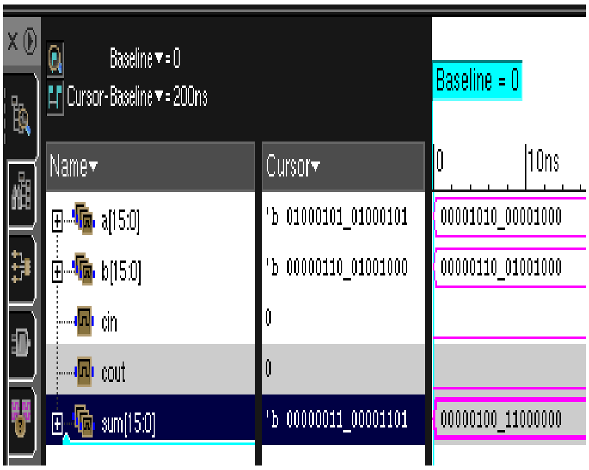
Fig10:16 bit Reversible sparse kogge stone adder

The 16 bit RSKA uses black cells and gray cells as well as HNG gates blocks too. This adder computes the carries using the BC's and GC's and terminates with 4 bit RRCA's.

Totally it uses 16 HNG gates. In this adder, first the input bits (a, b) are converted as propagate and generate (p, g). Then propagate and generate terms are given to BC's and GC's. The carries are propagated in advance using these cells. Later these are given to reversible full adder Blocks.

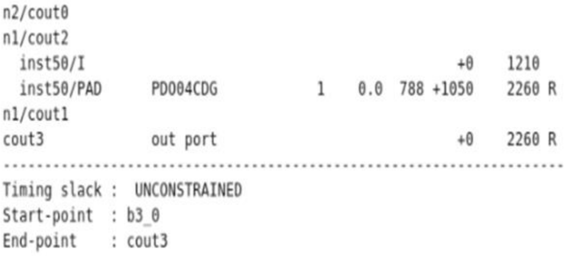
5. SIMULATION AND SYNTHESIS RESULTS

Various adders were designed using Verilog language The performance of proposed adders are analyzed and compared. In this proposed architecture, the implementation code for 16-bit Reversible Kogge-Stone, Reversible Brent-Kung adder, Reversible Spanning Tree adder, Reversible Sparse Kogge Stone Adder were developed and corresponding values of Delay, Power and Area were observed. Table1 shows the trade-off between different topologies and table2 shows the comparison of adders. The simulated outputs of 16-bit proposed adders are shown in Figure.



```
Baseline▾=0
Cursor-Baseline▾=200ns
Baseline = 0
0          10ns
Name▾        Cursor▾
a[15:0]      'b 01000101_01000101    00001010_00001000
b[15:0]      'b 00000110_01001000    00000110_01001000
cin          0
cout         0
sum[15:0]    'b 00000011_00001101    00000100_11000000
```

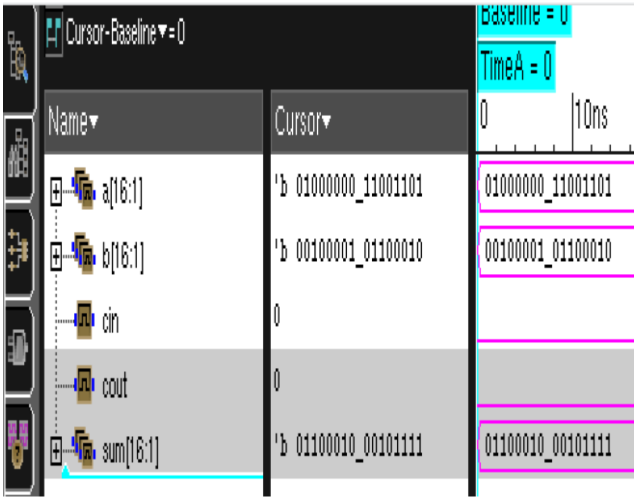
Fig11: Simulation Result of 16-bit Reversible kogge Stone Adder



```
n2/cout0
n1/cout2
  inst50/I                                +0    1210
  inst50/PAD      PD004CDG         1   0.0  788 +1050   2260 R
n1/cout1
cout3             out port                +0    2260 R
-------------------------------------------------------------------
Timing slack :  UNCONSTRAINED
Start-point  : b3_0
End-point    : cout3
```

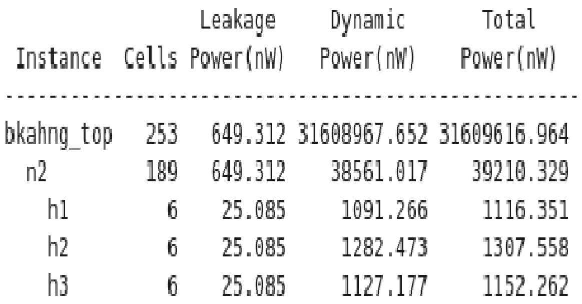
Fig12: Timing report of 16-bit RKSA (Delay)

Simulated waveform and delay of of 16 bit Reversible Kogge Stone Adder are shown in Figure 8 and 9. Delay of RKSA is 2260ps.



```
Cursor-Baseline▾=0
Baseline = 0
TimeA = 0
0          10ns
Name▾        Cursor▾
a[16:1]      'b 01000000_11001101    01000000_11001101
b[16:1]      'b 00100001_01100010    00100001_01100010
cin          0
cout         0
sum[16:1]    'b 01100010_00101111    01100010_00101111
```

Fig13: Simulation Result of 16-bit Reversible Brent Kung adder.



```
                        Leakage     Dynamic       Total
Instance  Cells Power(nW)   Power(nW)     Power(nW)
-------------------------------------------------------
bkahng_top   253   649.312 31608967.652 31609616.964
  n2         189   649.312   38561.017    39210.329
    h1         6    25.085    1091.266     1116.351
    h2         6    25.085    1282.473     1307.558
    h3         6    25.085    1127.177     1152.262
```

Fig14: Power dissipation of 16-bit Reversible Brent Kung Adder

6. DISCUSSION OF RESULTS

Now we Synthesis and Simulate all above adders for 16 bit and tabulated the results are show in Table 6.

Name of the Adder	Area	Delay(ps)	Power(nW)
Conentional full adder	361959	1802	2283288.622
Full adder using HNG	390169	1866	4112047.300
16-bit RCA	785705	4277	34898774.286
16-bit RCA using HNG	785617	4205	37095520.112
16-bit SKA	785983	2756	37063920.518
16-bit RSKA	785995	2710	37065902.603
16-bit STA	785965	2753	37063554.575
16-bit RSTA	785977	2707	37065536.660
16-bit BKA	786272	2617	31611171.994
16-bit RBKA	786175	2567	31609616.964
16-bit KSA	786686	2389	37086958.405
16-bit RKSA	786548	2260	36932884.418

Table 6. Area, Delay and Power of FA, HNG, RCA, RCA using HNG, RSKA, RSTA, RKSA & RBKA

7. CONCLUSION

Full adder is the basic logic element in all the 16-bit adders discussed in this thesis. One bit full adder is simulated and synthesized in cadence (180nm technology) and one bit full adder is realized using HNG gate (i.e. reversible full adder). From the Table 6 the area, delay and power dissipation of reversible one bit full adder is more when compared to the conventional one bit full adder.

16-bit ripple carry adder is simulated and synthesized followed by the synthesis of 16-bit reversible ripple carry adder. From the Table 6 the area, delay and power dissipation of reversible 16-bit ripple carry adder is less when compared to the conventional16-bit ripple carry adder. Thus we can conclude that as the number of bits is increasing reversible full adder is advantageous when compared to conventional full adder. Hence conventional full adders in 16-bit parallel prefix adders are replaced with reversible full adder.

16-bit brent kung adder (BKA) is simulated and synthesized followed by 16-bit reversible brent kung adder (RBKA). From the Table 6 the area, delay and power dissipation of RBKA is less when compared to BKA.

16-bit Kogge stone adder(KSA) and then 16-bit reversible kogge stone adder(RKSA) are simulated and synthesized. From the Table 6 the area, delay and power dissipation of RKSA is less when compared to KSA.

Finally from Table 6 we can conclude that the delay of 16-bit Reversible Kogge Stone Adder is less when compared to all other adders. The delay of 16-bit reversible kogge stone adder is 46.53% less than the delay of 16-bit ripple carry adder.

The power of Reversible Brent Kung Adder is less compared to all other adders. The power dissipation of 16-bit reversible brent kung adder is 16.55% less than the power dissipation of 16-bit ripple carry adder.

Thus Reversible kogge stone adder is used for high speed applications and Reversible brent kung adder is used for low power applications.

FUTURE SCOPE

If the complete architecture of Parallel Prefix Adders are implemented by using reversible logic gates then area, delay & power can be reduced further.

REFERENCES

[1] Sudheer Kumar Yezerla, B Rajendra Naik, "Design and Estimation of delay, power and area for Parallel prefix adders" 978-1-4799-2291-8/14/$31.00 ©2014 IEEE, Proceedings of 2014 RAECS UIET Panjab University Chandigarh, 06 - 08 March, 2014

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

[3] N. H. E. Weste and D. Harris, CMOS VLSI Design, 4th edition, Pearson Addison Wesley, 2011.

[4] R. P. Brent and H. T. Kung, *"A regular layout for parallel adders,"* IEEE Trans. Comput., vol. C-31, pp. 260-264, 1982

[5] D. Harris, "*A Taxonomy of Parallel Prefix Networks,"* in Proc. 37th Asilomar Conf. Signals Systems and Computers, pp. 2213–7, 2003.

[6] P. M. Kogge and H. S. Stone, "*A Parallel Algorithm for the Efficient Solution of a General Class of Recurrence Equations*," IEEE Trans. On Computers, Vol. C-22, No 8, August 1973.

[7] D. Gizopoulos, M. Psarakis, A. Paschalis, and Y. Zorian, "*Easily Testable Cellular Carry Lookahead Adders,"* Journal of Electronic Testing: Theory and Applications 19, 285-298, 2003.

[8] T. Lynch and E. E. Swartzlander, "*A Spanning Tree Carry Lookahead Adder*," IEEE Trans. on Computers, vol. 41, no. 8, pp. 931-939, Aug.1992.

[9] Beaumont-Smith, A, Cheng-Chew Lim,"*Parallel prefix adder design*", Computer Arithmetic, 2001. Proceedings. 15th IEEE Symposium, pp. 218 – 225, 2001.M. Young, the Technical Writer's Handbook. Mill Valley, CA: University Science, 1989

[10] K. Vitoroulis and A. J. Al-Khalili, "*Performance of Parallel Prefix Adders Implemented with FPGA technology,*" IEEE Northeast Workshop on Circuits and Systems, pp. 498-501, Aug. 2007. 172.

[11] S. Xing and W. W. H. Yu, "*FPGA Adders: Performance Evaluation and Optimal Design*," IEEE Design & Test of Computers, vol. 15, no. 1, pp.24-29, Jan. 1998.

[12] Madhusmita Mahapatro, Sisira Kanta Panda, Jagannath Satpathy, Meraj Saheel, *"Design of Arithmetic Circuits Using Reversible Logic Gates and Power Dissipation Calculation"*,978-1-4244-8979-4©2010 IEEE, Electronic System Design (ISED), 2010 International Symposium

[13] Raghava Garipelly, P.Madhu Kiran, A.Santhosh Kumar,*"A Review on Reversible Logic Gates and their Implementation"*, ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 3, Issue 3, March 2013

[14] Prashant .R.Yelekar, Sujata S. Chiwande, *"Introduction to Reversible Logic Gates & its Application"*, 2nd National Conference on Information and Communication Technology (NCICT) 2011 Proceedings published in International Journal of Computer Applications® (IJCA)

[15]. Manjeet Singh Sankhwar, Rajesh Khatri, *"Design of High Speed Low Power Reversible Logic Adder Using HNG Gate"*, Int. Journal of Engineering Research and Applications, ISSN : 2248-9622, Vol. 4, Issue 1(Version 2), January 2014, pp.152-159

[16] Zhijin Guan, Wenjuan Li, Weiping Ding, Yueqin Hang, *"An Arithmetic Logic Unit Design Based on Reversible Logic Gates"*, 978-1-4577-0253-2/11/$26.00 ©2011 IEEE

[17]. J.W.Bruce, M.A.Thornton, L.Shivakumaraiah, P.S.Kokate, X.Li, *"Efficient adder circuits based on a conservative reversible logic gate"*, Proceedings of the IEEE Computer Society Annual Symposium on VLSI (ISVLSI.02), 0-7695-1486-3/02 $17.00 © 2002 IEEE

[18] D.P. Vasudevan, P.K. Lala and J.P. Parkerson, *"CMOS Realization of Online Testable Reversible Logic Gates"*, Proceedings of the IEEE Computer Society Annual Symposium on VLSI, New Frontiers in VLSI Design, 0-7695-2365-X/05 $20.00 © 2005 IEEE

[19] Zhijin Guan, Wenjuan Li, Weiping Ding, Yueqin Hang,*"An Arithmetic Logic Unit Design Based on Reversible Logic Gates"*, 978-1-4577-0253-2/11/$26.00 ©2011 IEEE

[20] Lekshmi Viswanath, Ponni.M, *"Design and Analysis of 16 Bit Reversible ALU"*, IOSR Journal of Computer Engineering (IOSRJCE) ISSN : 2278-0661 Volume 1, Issue 1 (May-June 2012), PP 46-53

[21] R. Landauer, *"Irreversibility and Heat Generation in the Computational Process"*, IBM Journal of Research and Development, 5, pp. 183- 191, 1961.

[22]. C H Bennett, *"Notes on the History of Reversible Computation"*, IBM Journal of Research and Development, vol. 32, pp. 16-23, 1998.

[23]. Cadence, *"Encounter user guide"*, Version 6.2.4, March 2008