ISSN (Print) : 2320 – 9798
ISSN (Online): 2320 – 9801

International Journal of Innovative Research in Computer and Communication Engineering
Vol. 1, Issue 3,May 2013

Computational Reduction Logic for Adders

[1]M. Siva Kumar, [2]Dr.S.K.Srivatsa

[1]Associate Professor , Dept of ECE, K L University, Andhra Pradesh, India

[2]Sr.Professor, Bharath University, Chennai, India

ABSTRACT: As the technology is increasing, the designer is trying to increase the density of chips, decrease the power consumption and power dissipation, decrease the area and increase the computational and storage logics on single chip by maintaining lower complexity. But there is no single approach till now having following ideal outputs. When complexity of computational logic increases, the designing, testing, debugging becomes even more complex. So this paper gives a different approach of any n-bit adder using less computations, Look up tables, Slices and gates, which is verified and simulated in Xilinx ISE DESIGN SUITE 14.2 (Verilog Hardware Descriptive Language) and Cadence .In general we use carry look ahead adder for adding the two numbers along with carry . But this is not applicable because when we consider a large number of bits, complexity increases. So this paper gives an alternative for decreasing complexity and the memory.

KEYWORD: Look Up tables, Slices, Carry Look Ahead adder.

I.INTRODUCTION

This paper is aimed to give a description about a different logic for n-bit adding. The traditional method of using carry look ahead adder is considered risky since it is only suitable for 4 bits addition. But as number of bits increases the delay and complexity increases. The same happens in case of carry skip adder, carry save adder, ripple carry adder. By this procedure we decrease the number of steps required to add two signals.

II.OVERVIEW OF DIFFERENT ADDERS

At first a general adding procedure is used for the signals either with or without carry. But this increases the delay as number of bits increases. Thus to avoid this problem there evolved many types of adders which are described briefly.

A. Ripple Carry Adder

This is a logical circuit having many full adders connected to add n-bit numbers. The first block can be a half adder since the carry in (Cin) is taken Zero or else we can consider it to be a Full Adder having Cin=0.And the Output of 1st adder Cout acts as Cin for second adder. But the limitation is that each full adder has to wait for the Cout of previous full adder for execution. Thus the delay is more i.e execution through RCA is relatively slow. A n-bit adder have a delay in order of 2^(n-1)-1.

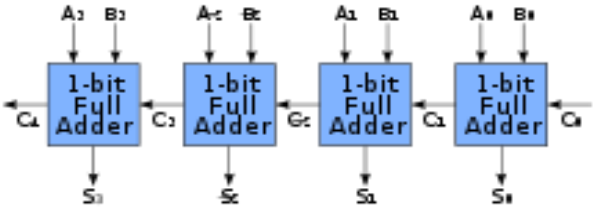


Fig 1. Ripple carry adder

B. Carry Look Ahead Adder

In order to overcome the limitations, this was followed where for every bit, a Propagator bit and Generator bit (P& G) is created. Generally P is Sum output of Half adder and G is the carry output of the same half adder. The equations for P,G, Sum and carry for every bit is

Pi=XOR(Ai,Bi)
Gi=OR(Ai,Bi)
Si=XOR(Pi,Ci)
Ci+1=Gi+OR(Ci,Pi)
And the delay caused by it is (n/2+3)D where D is the delay of each gate.

ISSN (Print) : 2320 – 9798
ISSN (Online): 2320 – 9801

International Journal of Innovative Research in Computer and Communication Engineering
Vol. 1, Issue 3,May 2013

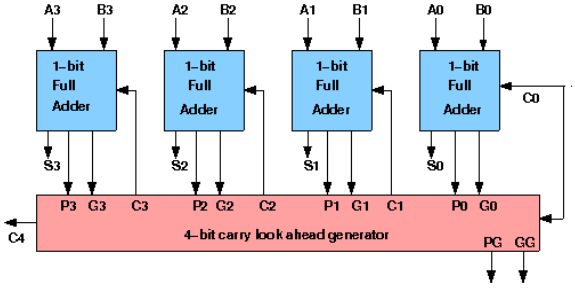


Fig 2.Carry look ahead adder

C. Carry skip adder

Delay is minimized here by skipping the generation of carry in certain bits. i.e when group of 4 bits are totally different, then the cin =cout, and thus the carry generation is skipped here. CSA takes advantage of both the generation and propagation of carry signal .It shortens the critical path by computing the group propagate signal for each carry change.

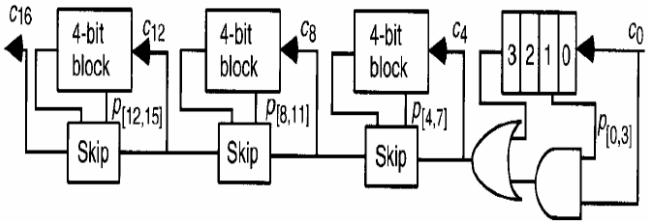


Fig3.Carry skip adder

D. Carry Select Adder

In this adder, the delay is reduced efficiently without waiting for the carry to generate. This is an extended logic for CSA where 0 and 1 possibilities are analyzed in earlier stage. When carry out is generated, either the result of 0 or 1 is selected by multiplexer, by which minimum delay is possible.

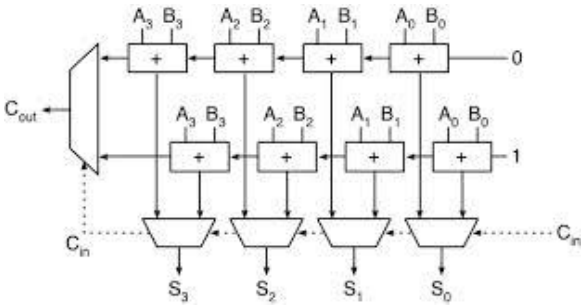


Fig4.Carry select adder

III.PROPOSED LOGIC

The main strategy of adders is that the minimum memory consumption, which is given a priority since there is a need for low density of chips. So in this approach, other than computing the carry separately, it is done using sum output. For a example, for an 4 bit numbers, sum is computed using xor gate, and carry is calculated using negation for xor gate, by using a condition that if any two bits in two numbers are zero then the same bit of carry is assigned as zero. And a normal addition is done.

The assignment of the 0 for the bit having 0 as bit value for both numbers is

```
for (i = 0; i<=3;i=i+1)
begin
if(a[i]==0 && b[i]==0)
        s2[i]=1'b0;
        else      s2[i] = s2[i];
end
```

ISSN (Print) : 2320 – 9798
ISSN (Online): 2320 – 9801

International Journal of Innovative Research in Computer and Communication Engineering
Vol. 1, Issue 3,May 2013

IV.ADVANTAGES

When compared with normal carry look ahead adder, the following are greatly reduced.
a)No.of slices
b)No.of LUT's
LUT's are nothing but Look Up Table where we store the predefined output for each and every input. So that every time the output need not be computed for the input. LUT's are kind of logics that are used in SRAM based FPGA's.
An Example for its working is, Consider a 2 input NAND gate where 1,0,0,0 are stored respectively in address 00,01,10,11.Now input becomes address lines. So when input is 0 and 0,the output comes as 1.It is used to reduce the computations.

V.RESULTS

Program is simulated in Xilinx and following parameters are observed.

A.RTL Schematic

The Rtl schematic of following logic will be similar to

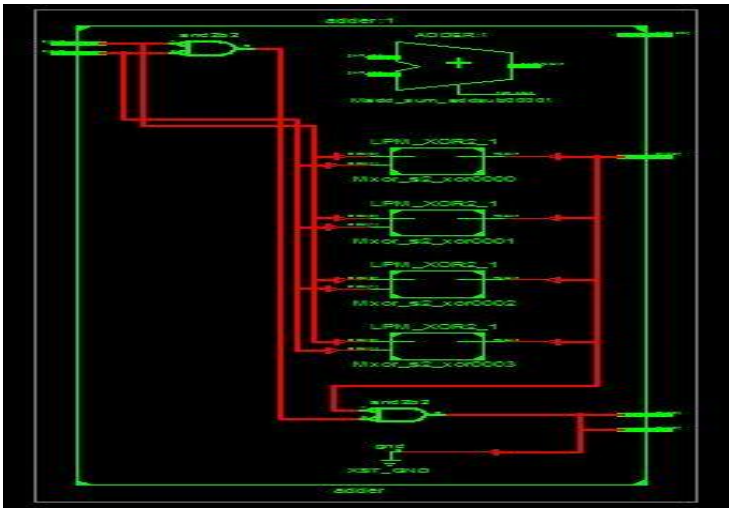


Fig5.RTL schematic of proposed logic

Where Rtl schematic means logic is represented with the help of gates.

B. Technology Schematic

Technology schematic of the above logic is similar to

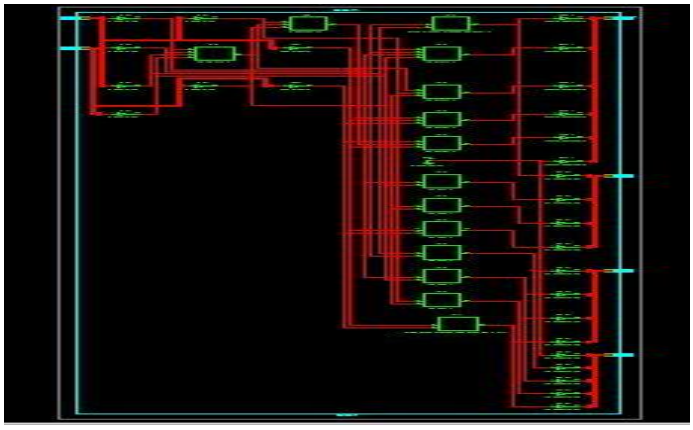


Fig6.Techonoly schematic of proposed logic

Where technology schematic means the logic is represented using buffers, LUT's, CLB's.

C. Computational logics involved

When simulated in Xilinx, the following parameters like No .of slices, No. of LUT's, No. of bonded IOB's

ISSN (Print) : 2320 – 9798
ISSN (Online): 2320 – 9801

International Journal of Innovative Research in Computer and Communication Engineering
Vol. 1, Issue 3,May 2013

```
Logic Utilization:
  Number of 4 input LUTs:                14 out of   9,312    1%
Logic Distribution:
  Number of occupied Slices:              8 out of   4,656    1%
    Number of Slices containing only related logic:       8 out of       8 100%
    Number of Slices containing unrelated logic:          0 out of       8   0%
      *See NOTES below for an explanation of the effects of unrelated logic.
  Total Number of 4 input LUTs:          14 out of   9,312    1%
  Number of bonded IOBs:                 27 out of     190   14%

Average Fanout of Non-Clock Nets:              2.41

Peak Memory Usage:  251 MB
Total REAL time to MAP completion:  4 secs
Total CPU time to MAP completion:   1 secs
```

Where it is clearly given that
No. of slices containing only related logic is 100% and No. of slices containing unrelated logic is 0%.
Even utilization of 4 input LUT's is 1% and bonded IOB's is 14%, No.of slices is 1%.
Fan out problem because of Non-Clock Nets is 2.41 which is low when compared to previous ones.

D. Power Analysis

Power Analysis is done in Cadence tool, which gave a result having very low power consumption for 4 bit adders

Instance	Cells	Leakage Power(nW)	Dynamic Power(nW)	Total Power(nW)
adder	56	405.405	5142.481	5547.887
add_24_14	32	191.822	2363.143	2554.965
mux_s2_17_11	1	13.605	164.608	178.214
mux_s14	1	13.605	184.858	198.464
mux_s15	1	13.605	164.608	178.214
mux_s16	1	13.605	174.407	188.013

E.Noise

Noise produced while executing this logic is observed as

Device Parameters	
Maximum ground bounce	600.0 mV
Capacitance per output driver	15.0 pF
Board Parameters	
Board Thickness	62.0 mils
Finished via diameter	12.0 mils
Pad to via breakout length	33.0 mils
Breakout width	12.0 mils
Other PCB parasitic inductance	0.0 nH
Socket inductance	0.0 nH

F. Area

Total area required to implemented this logic is even calculated using Cadence Tool

ISSN (Print) : 2320 – 9798
ISSN (Online): 2320 – 9801

International Journal of Innovative Research in Computer and Communication Engineering
Vol. 1, Issue 3,May 2013

Instance	Cells	Cell Area	Net Area	Wireload
adder	56	303	0	<none> (D)
add_24_14	32	171	0	<none> (D)
mux_s16	1	12	0	<none> (D)
mux_s15	1	12	0	<none> (D)
mux_s14	1	12	0	<none> (D)
mux_s2_17_11	1	12	0	<none> (D)

(D) = wireload is default in technology library

G. Gates Required

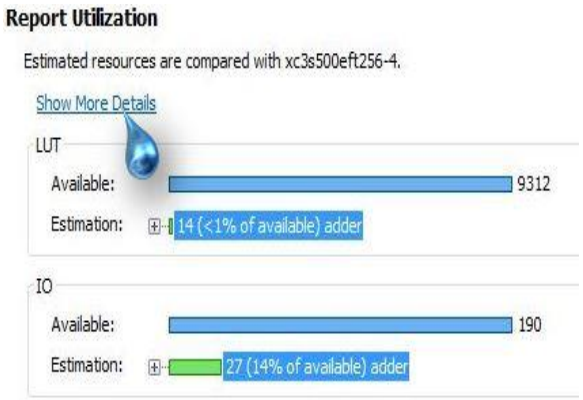
No. of gates required to implement this logic is found using Cadence Tools

Type	Instances	Area	Area %
inverter	19	0.000	0.0
logic	37	0.000	0.0
total	56	0.000	0.0

H. Comparison with CLA

The summary of both the adders are given separately, where it is observed that utilization of IOB's is more in the proposed adder rather than in CLA. But the amount of memory stored in form LUT's are less in the proposed adder when compared with CLA.

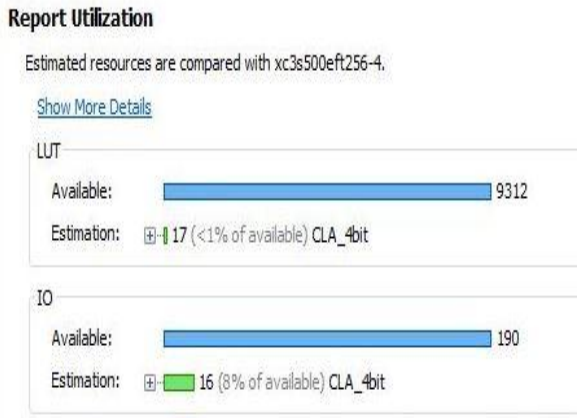
Result of proposed Adder is



And report utilization for a CLA is

ISSN (Print) : 2320 – 9798
ISSN (Online): 2320 – 9801

International Journal of Innovative Research in Computer and Communication Engineering
Vol. 1, Issue 3,May 2013



VI.CONCLUSION

By comparing with the traditional adders,its better to use the above logic for adders for any n-bit numbers,because of less complexity,and fewer no.of gates,more utilization factor,less power loss when compared with CLA.

REFERENCES

1. VasuDevan.D.P, Parkerson “Two Rail Encoding Design based Adder” ,PP325-330 2004
2. C.K.Tang “Group Look a Head binary adder”, pp563-573,1970
3. Neil H.E.Weste, David Money Harris “CMOS VLSI Design
4. Sharat.C.Prasad, Kaushik Roy “Low Power CMOS VLSI Design”
5. Gurjar.P “Adders for high speed ALU” Indian conference pp1-6 2011
6. FarzhanFalla “ High Performance adder of New class” pp40-45 2011
7. JafferSaniee “Constant addition adder” pp1-6 ,2001
8. V.Vamsimohan Krishna “Kogge stone carry select adder” vol2, 2013
9. S.N.Singh “Reduced Area Carry save adder”, vol2 issue 5, 2013
10. Ravichandran “high speed adder” pp 99-104,2013