International Journal of Research in Advent Technology, Vol.3, No.1, January 2015
E-ISSN: 2321-9637

VLSI Implementation of Modified Design of 32 Bit Integer Unsigned Multiplier Using CLSA and CLAA

Sumit Dixit1 , Ghanshyam2
(MTech Scholar Department of Electronics and communication , Suresh Gyan Vihar University, Jaipur, Raj.,India.) , M.Tech,.Department of Electronics and Communication , MNIT, Jaipur, Raj.,India.)2
Email.- sdsumit09@gmail.com1, ghanu.4us@gmail.com2

Abstract— Speed and chip size optimization of digital arithmetic circuit has become important issue in VLSI design. Researchers have done lots of research in this area of optimization by using various algorithm proposed till date, so keeping all these research in concern in this dissertation we are comparing the VLSI design of 32-bit unsigned integer multiplier based on CSLA (carry select adder) and VLSI design of CLAA (carry look-ahead adder) based 32-bit unsigned integer multiplier. We are using them to multiply two 32-bit unsigned integers to get a product that will be of 64-bit. The CLAA based multiplier uses the delay time of 75.081ns for multiplication operation where as multiplier based on CSLA also uses approx 49.343 delay for same task to do but the area needed for CLAA based multiplier is 33% more then the CSLA based multiplier .

Keywords—CLAA, CSLA, Delay, Area, Array Multiplier,VHD Modeling & Simulation.

1. INTRODUCTION

Digital computer arithmetic is an aspect of logical design with the idea of developing suitable algorithms in order to achieve an efficient consumption of the existing hardware. In this, we will deal with the operations like addition and multiplication. As we know that hardware can only perform basic Boolean operation, complex operations are based on the series of these primitive operations. In VLSI, the efficiency and performance of the design is determined by speed, power and chip area.

Multiplications and additions are most widely and more often used arithmetic computations performed in all digital signal processing applications. Fundamental operation for any digital multiplication is addition. The designs area efficiency, delay time and accuracy are greatly influenced by performance of the used adder. So to improve the efficiency of our design, our main focus will be on the working of adder.

In this project, we are comparing the change in area and delay time in a multiplier by the use of different adders i.e. CLAA and CSLA. Here we are dealing with the comparison in bit range of 32 * 32 as input to the multiplier that will give us 64 bit output.

Hence, our interest must be on design of a better architecture of basic building block i.e. adder. So be need DSP style system for both area efficient and less delay. Our basic block must dominate in DSP application, VLSI architecture, computer application and where ever reduced delay is needed.

2. CARRY LOOK-AHEAD ADDER

Carry look ahead adder is an adder that can produce carries much faster due to use of parallel generation of the carry bits by using additional circuitry. It improves speed by reducing the amount of time required to determine carry bits. This technique uses calculation of carry signals in advance, based on the input signals. It results in reduced carry propagation time. For example, ripple adders are slower but uses the least energy.

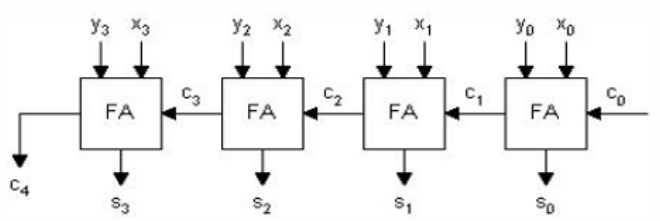


Figure: 1(a) Ripple Carry adder

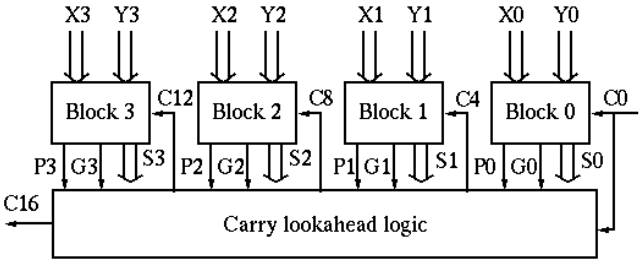


Figure: 1(b) Carry Look ahead adder

Let Gi is the carry generate function and Pi be the carry propagate function, Then we can rewrite the carry function as follows:

$Gi = Ai \cdot Bi$ …… (1)
$Pi = (Ai \text{ xor } Bi)$ …… (2)
$Si = Pi \text{ xor } Ci$ …… (3)
$Ci+1 = Gi + Pi.Ci$…… (4)

Thus, for 4-bit adder, we can compute the carry for all the stages as shown below:

$C_1 = G_0 + P_0.C_0$ (5)

$C_2 = G_1 + P_1.C_1 = G_1 + P_1.G_0 + P_1.P_0.C_0$ (6)

$C_3 = G_2 + P_2.C_2 = G_2 + P_2.G_1 + P_2.P_1.G_0 + P_2.P_1.P_0.C_0$ (7)

$C_4 = G_3 + P_3.C_3 = G_3 + P_3.G_2 + P_3.P_2.G_1 + P_3.P_2.P_1.G_0 + P_3.P_2.P_1.P_0.C_0$ (8)

In general, we can write:

The sum function:

$SUM_i = A_i$ xor B_i xor $C_i = P_i$ xor C_i (9)

The carry function:

$C_{i+1} = G_i + P_i.C_i$ (10)

3. CARRY SELECT ADDER

In CSLA, we compute alternative results in parallel and subsequently selecting the correct result with single or multiple stages of hierarchical techniques. In this added, for both alternatives we carry calculation of sum and carry bits. When Cin is delivered, the correct computation is chosen with the help of a multiplexer to get the desired output. So by this technique we can save the time which is used for computation and hence speed is improved.

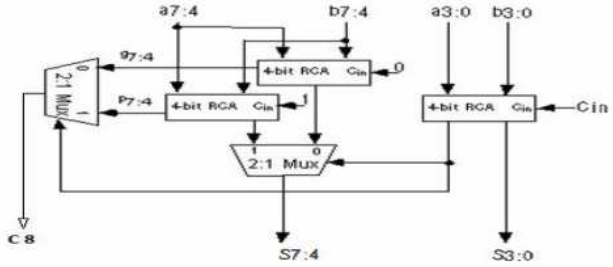


Figure: 2 Carry select adder

In general, we can write algorithm as:

If *Cin = 1*, then sum and carry out are given by

Sum(i) = a(i) xor b(i) xor '1' (11)

Carry (i+1) = (a(i) and b(i)) or (b(i) or a(i)) (12)

If *Cin =0*, then sum and carry out are given by,

Sum(i) = a(i)xor b(i) (13)

Carry(i+1) = (a(i)and b(i)) (14)

The sum function:

$S_i = C_iS_i^0 + C_iS_i^1$ (15)

The carry function:

$C_{i+1} = C_iC_{i+1}^0 + C_iC_{i+1}^1$ (16)

4. MULTIPLIER FOR UNSIGNED DATA & MULTIPLICATION ALGORITHM

Multiplication is done by addition of partial products that is generated by each digit in the multiplier as shown in the figure 3. The multiplication of two n-bit binary integers results in a product of up to 2n bits.

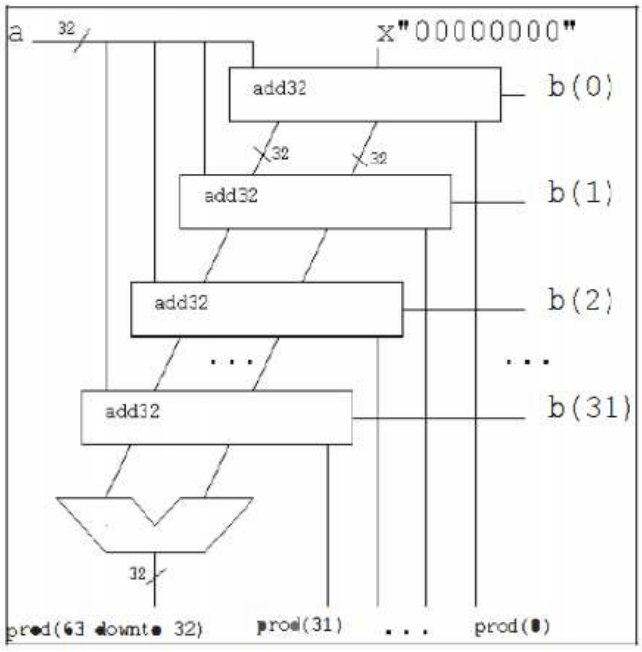


Figure: 3 Partial scheme of multiplier

ALGORITHM

Let the multiplicand register size and product register size is 32 bits and 64 bits respectively. The multiplier is stored in the least significant half of the product register and rest bits are cleared.

Repeat the following steps for 32 times:

1. If the least significant bit of the product register is "1" then add the multiplicand to the most significant half of the product register.
2. Shift the content of the product register one bit to the right (ignore the shifted-out bit.)
3. Shift-in the carry bit into the most significant bit of the product register. Figure 4.Shows a block diagram for such a multiplier.

Simulation Result

1. *32 bit Multiplier with CLA*

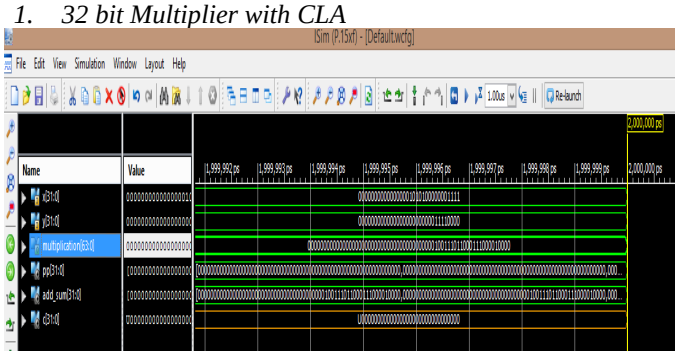


Figure: 4 Simulation Result of Multiplier with CLA

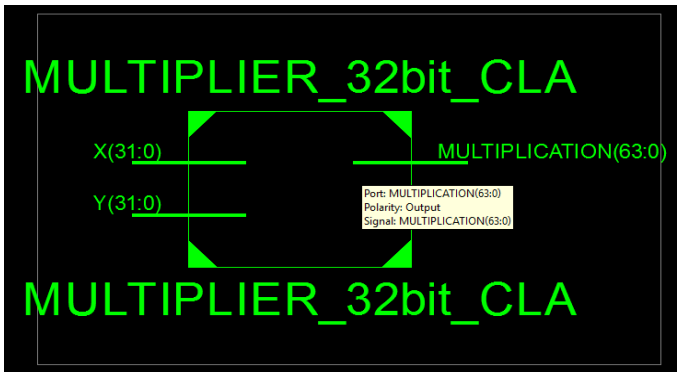


Figure: 5 RTL View of Multiplier with CLA

```
Timing Detail:
--------------
All values displayed in nanoseconds (ns)

=========================================================
========
Timing constraint: Default path analysis
  Total number of paths / destination ports:
14714169984932217000000000000000000000 / 63
-------------------------------------------------------------
--------
Delay:               75.081ns (Levels of Logic = 114)
  Source:            Y<2> (PAD)
  Destination:       MULTIPLICATION<62> (PAD)
```

Figure: 6 Timing Analysis of Multiplier with CLA

```
Device utilization summary:
---------------------------

Selected Device : 5vlx30ff324-2


Slice Logic Utilization:
 Number of Slice LUTs:                 2652  out of  19200    13%
    Number used as Logic:              2652  out of  19200    13%

Slice Logic Distribution:
 Number of LUT Flip Flop pairs used:   2652
   Number with an unused Flip Flop:    2652  out of   2652   100%
   Number with an unused LUT:             0  out of   2652     0%
   Number of fully used LUT-FF pairs:     0  out of   2652     0%
   Number of unique control sets:         0

IO Utilization:
 Number of IOs:                         128
 Number of bonded IOBs:                 127  out of    220    57%

Specific Feature Utilization:
```

Figure: 7 Area Analysis of Multiplier with CLA

2. 32 bit Multiplier with CSLA

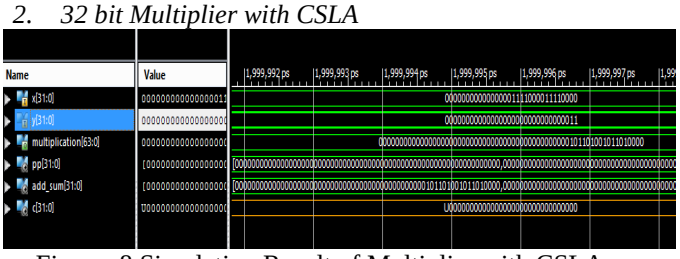
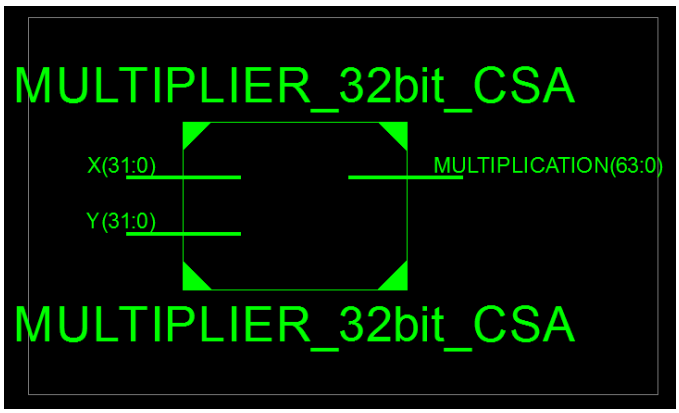


Figure: 8 Simulation Result of Multiplier with CSLA



_____ Figure: 9 RTL View of Multiplier with CSLA

```
Timing constraint: Default path analysis
  Total number of paths / destination ports:
13263098304639542000000000 / 63
-------------------------------------------------------------
--------
Delay:               49.343ns (Levels of Logic = 62)
  Source:            Y<1> (PAD)
  Destination:       MULTIPLICATION<60> (PAD)
```

Figure: 10 Timing Analysis of Multiplier with CSLA

```
Device utilization summary:
---------------------------

Selected Device : 5vlx30ff324-2


Slice Logic Utilization:
 Number of Slice LUTs:                 1457  out of  19200     7%
    Number used as Logic:              1457  out of  19200     7%

Slice Logic Distribution:
 Number of LUT Flip Flop pairs used:   1457
   Number with an unused Flip Flop:    1457  out of   1457   100%
   Number with an unused LUT:             0  out of   1457     0%
   Number of fully used LUT-FF pairs:     0  out of   1457     0%
   Number of unique control sets:         0

IO Utilization:
 Number of IOs:                         128
 Number of bonded IOBs:                 127  out of    220    57%

Specific Feature Utilization:
```

Figure: 11 Area Analysis of Multiplier with CSLA

5. CONCLUSION

Previous work based on 32 bit operand & designed on ALTERA EDA tool, our proposed work based on 32 bit operand & designed on XILINX EDA tool and during the implementation our proposed design's optimized result 75.081ns with CLA and 49.343ns with CSLA. The overall delay is reduced by 24ns i.e. 24% with CLA and 49ns i.e. 48% with CSLA. We have also worked in the field of area. In proposed design the area is in terms of LUT which is different from previous existing Quartus's logic cells, in this proposed design the occupancy of LUT is 2652 and 1457 respectively Multiplier with CLA and CSLA.

REFERENCES

1. 978-1-4673-5301-4/13/2013 IEEE Design and Implementation of 32 Bit Unsigned Multiplier Using

CLAA and CSLA V.Vijayalakshmil, R.Seshadd, Dr.S.Ramakrishnan3

2. Emerging Trends in Electrical, Electronics & Instrumentation Engineering: An international Journal (EEIEJ), Vol. 1, No. 1, February 2014 57 IMPLEMENTATION OF UNSIGNED MULTIPLIER USING MODIFIED CSLA Sooraj.N.P.
3. INTERNATIONAL JOURNAL OF PROFESSIONAL ENGINEERING STUDIES Volume II/Issue 3/JUNE 2014 IJPRES DESIGN AND PERFORMANCE ANALYSIS OF CSLA AND CLAA FOR 32-BIT UNSIGNED MULTIPLIER USING VERILOGHDL S. Mounika, Mr. B. Amarnath
4. IPASJ International Journal of Electronics & Communication (IIJEC) Volume 2, Issue 7, July 2014 VLSI IMPLEMENTATION OF 32 BIT UNSIGNED MULTIPLIER USING AN EFFICIENT CLAA AND CSLASriRamya P1, SuhaliAfroz MD2
5. International Journal of Electrical and Electronics Engineering & Telecommunication (IJEEET) Volume 3, Issue 3, July 2014 DESIGN AND IMPLEMENTATION OF 32 BIT UNSIGNED MULTIPLIER USING CLAA, CSLA, ETA