DESIGN AND IMPLEMENTATION OF LOW POWER ERROR TOLERANT ADDER

LINTU K BABU
ELECTRONICS AND COMMUNICATION
MANGALAM COLLEGE OF ENGINEERING
ETTUMANOOR
lintukbabu@gmail.com

HIMA SARA JACOB
ELECTRONICS AND COMMUNICATION
MANGALAM COLLEGE OF ENGINEERING
ETTUMANOOR
Hima.jacob@mangalam.in

***Abstract*— This Mini Project presents a number of low power error tolerant adder designs. In modern VLSI technology, the occurrence of all kinds of errors has become inevitable. By adopting an emerging concept in VLSI design and test, error tolerance (ET), a novel error-tolerant adder (ETA) is proposed The ETA is able to ease the strict restriction on accuracy and at the same time achieve tremendous improvements in both the power consumption and speed performance. When compared to its conventional counterparts, the proposed ETA is able to attain improvement in the Power. One important potential application of the proposed ETA is in digital signal processing systems that can tolerate certain amount of errors. In this method the power dissipation due to carry propagation is greatly reduced unlike other conventional adders. In the conventional adder circuit, the delay is mainly attributed to the carry propagation chain along the critical path, from LSB to MSB. Also glitches in the carry propagation chain dissipate a significant proportion of dynamic power dissipation. Therefore, if the carry propagation can be eliminated or curtailed, a great improvement in speed performance and power consumption can be achieved. These adders were implemented in Verilog HDL.Delay and power analyses were carried out using Xilinx ISE 13.2. The results obtained are presented and compared. Conventional ripple carry adder and error tolerant adder were compared, and find out that the error tolerant adder be the minimum delay, area and power.**

Keywords—ETA,adder,xilinx.

I. INTRODUCTION

Adder is one among the fundamental components of many digital and non-digital systems and hence, their power dissipation and speed are of prime concern. In portable analog applications where power consumption is the most important parameter, one should reduce power dissipation to the possible limit. In analog computations, generation of ―good enough‖ results is more important than totally accurate results .Hence, by adopting error tolerance concept in design and test; it is possible to generate good enough results .To deal with high speed and low power circuits for analog computations.

CONVENTIONAL ADDER

Ripple-Carry Adder (RCA):

The n-bit adder is built from n-one-bit full adders is known as a ripple carry adder, because of the way the carryis computed. Each full adder inputs a Cin, which is the Cout of the previous adder. This kind of adder is a ripple carry adder, since each carry bit ―ripples‖ to the next full adder. Block diagram of Ripple Carry Adder is as in Fig. 1.**Ripple-Carry Adder (RCA):** The n-bit adder is built from n-one-bit full adders is known as a ripple carry adder, because of the way the carry is computed. Each full adder inputs a Cin, which is the Cout of the previous adder. This kind of adder is a ripple carry adder, since each carry bit ―ripples‖ to the next full adder. Block diagram of Ripple Carry Adder is as in Fig. 1.".

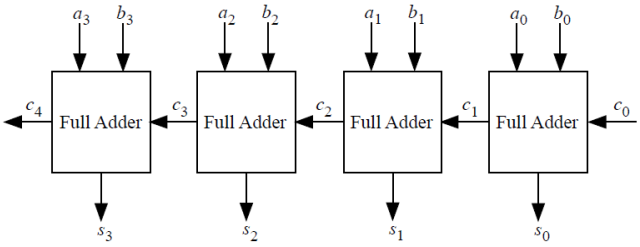


fig .1 4-bit Ripple Carry Adder

The ripple carry adder is simple, which allows for fast design time; however, the ripple carry adder is relatively slow, since each full adder must wait for the carry bit to be calculated from the previous full adder. Adder requires three levels of logic. In a 32-bit (ripple carry) adder, there are 32 full adders, so the critical path (worst case) delay is 31 * 2(for carry propagation) +3(for sum) = 65 gate delays.

II. ERROR TOLERANT ADDER

Before detailing the ETA, the definitions of some commonly used terminologies shown in this study are given as follows:

- Overall error (OE) OE = Rc-RE, where RE, is the result obtained by the adder and Rc denotes the correct result (all the results are represented as decimal numbers)
- Accuracy (ACC): In the scenario of the error tolerant design, the accuracy of an adder is used to indicate how ―correct‖ the output of an adder is for a

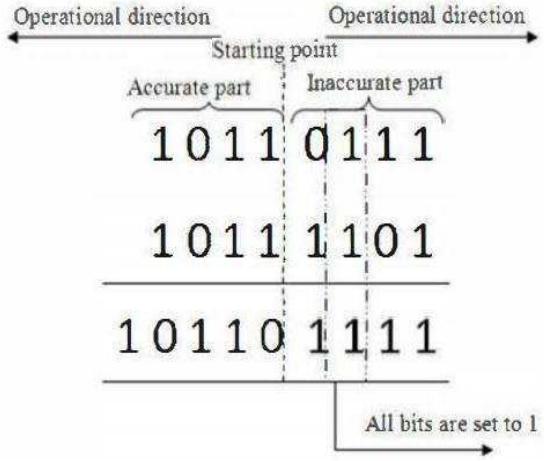
particular input. It is defined as: ACC = (1-(OE/Rc))/100%.Its value ranges from 0-100%.

- Minimum Acceptable Accuracy (MAA): Although some errors are allowed to exist at the output of an ETA, the accuracy of an acceptable output should be ―high enough‖ (higher than a threshold value) to meet the requirement of the whole system. Minimum acceptable accuracy is just that threshold value. The result obtained whose accuracy is higher than the minimum acceptable accuracy is called acceptable result.
- Acceptance Probability (AP): Acceptance probabilitis the probability that the accuracy of adder is higher than the minimum acceptable accuracy

III Proposed addition arithmetic:

In a conventional adder circuit, the delay is mainly attributed to the carry propagation chain along the the critical path, from the Least Significant Bit (LSB) to the Most Significant Bit (MSB). Meanwhile, a significant proportion of the power consumption of an adder is due to the glitches that are caused by the carry propagation. Therefore, if the carry propagation can be eliminated or curtailed, a great improvement in speed performance and power consumption can be achieved. In this study, we propose for the first time, an innovative and novel addition arithmetic that can attain great saving in speed and power consumption.

This new addition arithmetic can be illustrated via an example shown below. Here, we discuss about the addition arithmetic proposed in (Zhu *et al*., 2010), where the input operand is split into two parts: with higher order bits grouped into accurate part and remaining lower order bits into inaccurate.



. FIG 1 :proposed addition arithmetic [1]

The length of each part need not necessary be equal. The addition process starts from the demarcation line toward the two opposite directions simultaneously. In the example of Figure 1, the two 8-bit input operands, A= "10110111" (183) and B= "10111101" (189), are divided equally into 4 bits each for the accurate and inaccurate parts. The addition of the higher order bits (accurate part) of the input operands is performed from right to left (LSB to MSB) starting from the demarcation line with normal addition method applied. This is to preserve its correctness since the higher order bits play a *more important* role than the lower order bits. The lower order bits of input operands (inaccurate part) are added using error tolerant addition mechanism. No carry signal will be generated or taken in at any bit position to eliminate the carry propagation path.To minimize the overall error due to the elimination of the carry chain, a special strategy is adapted (Zhu et al., 2010), and can be described as follows:

- Check every bit position from left to right (MSB-LSB)starting from right of demarcation line;
- If both input bits are "0" or different, normal one-bit addition is performed and the operation proceeds to next bit position.
- The checking process is stopped when both input bits are encountered as high i.e., 1, and from this bit on wards, all sum bits to the right (LSB) are set to "1."

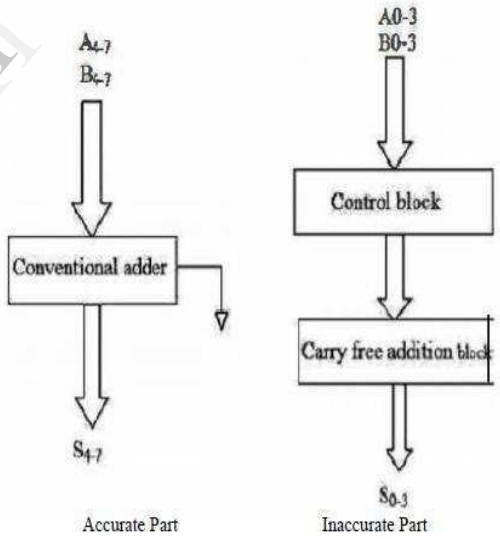


FIG 2:Block diagram of Error tolerant adder [1]

The block diagram of the Error Tolerant adder that adapts to our proposed addition arithmetic is shown in Fig. 2. This most straight forward structure consists of two parts: an accurate part and an inaccurate part.

Design of the accurate part

In our proposed 32 bit ETA, the inaccurate part has 20 bits as opposed to the 12 bits used in the accurate part. The overall delay is determined by the inaccurate part, and so the accurate part need not be a fast adder. The ripple-carry adder, which is the most power-saving conventional adder, The Ripple Carry Adder, being the simplest one, uses the least hardware circuitry when compared to all other traditional adder circuits

in use is show in fig 3.The delay of the ripple Carry adder increases linearly with the number of bits with a worst case delay of O(n).This worst case delay makes it slow when large bit sizes are used.

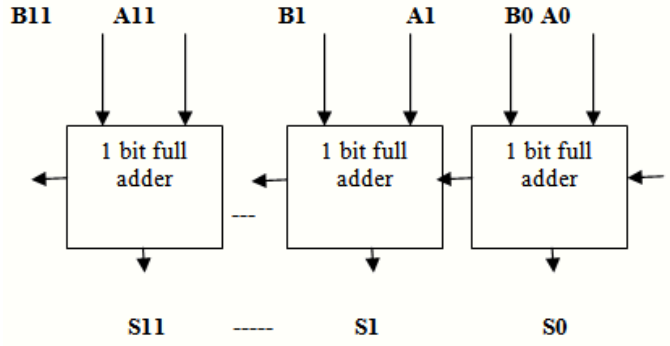


FIG 3:Block diagram of ripple carry adder [1]

Design of the inaccurate part

The inaccurate part is the most critical section in the proposed ETA as it determines the accuracy, speed performance and power consumption of the adder. The inaccurate part consists of two blocks: the carry free addition block and the control block. The carry-free addition block is made up of 20 modified XOR gates and each of which is used to generate a sum bit. The block diagram of the carry-free addition block and the schematic implementation of the modified XOR gate are presented in Fig.4.

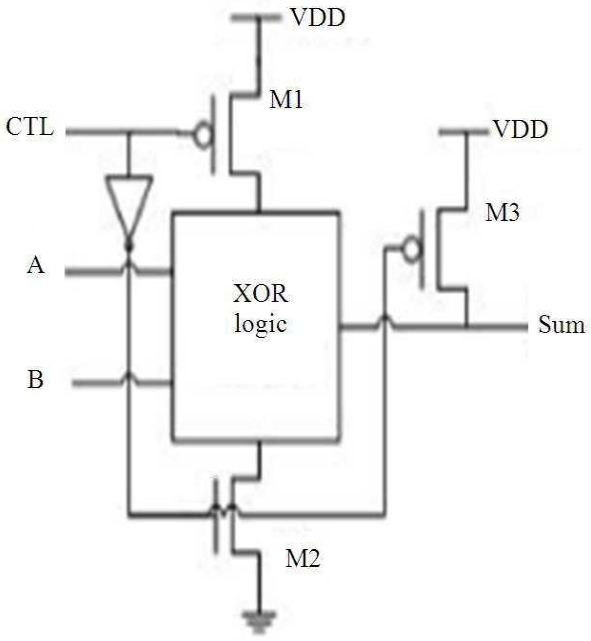


FIG 4: Block diagram of Modified XOR Gate [1]

In the modified XOR gate, three extra transistors, M1, M2 and M3, are added to a conventional XOR gate. CTL is the control signal coming from the control block of Fig. 7 and is used to set the operational mode of the circuit. When CTL = 0, M1 and M2 are turned on, while M3 is turned off, leaving the circuit to operate in the normal XOR mode. When CTL = 1 M1 and M2 are both turned off, while M3 is turned on, connecting the output node to VDD and hence setting the sum output to "1." The function of the control block is to detect the first bit position when both input bits are "1," and to set the control signal on this position as well as those on its right to high. It is made up of Control Signal Generating Cells (CSGCs) and each cell generates a control signal for the modified XOR gate at the corresponding bit position in the carry-free addition block. Instead of a long chain of cascaded GSGCs, the control block is arranged into five equal-sized groups, with additional connections between every two neighboring groups. Two types of CSGC, labeled as type I and II in below Fig. are designed and the schematic implementations of these two types of CSGC are provided in Fig.5 .

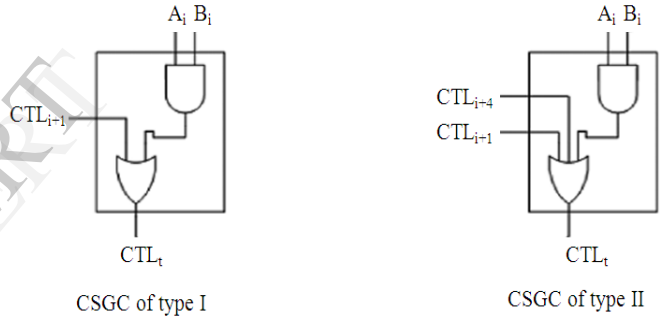


FIG 5: Diagram of control signal generating cells [1]

Made up of control signal generating cells (CSGCs) and each cell generates a control signal. Two types o f CSGC, labeled as type I and II.

CARRY FREE ADDITION BLOCK

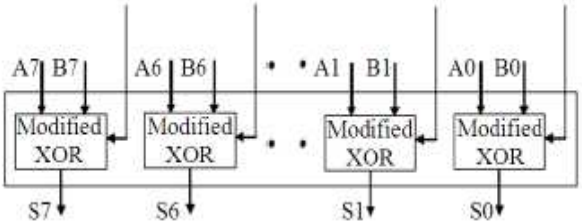


FIG 6:Block diagram of carry free addition block [1]

The modified xor gates are then combined together to obtain the carry free addition block as shown the fig.6.

IV. SIMULATION RESULT AND ANALYSIS

we compare the different error tolerant adders and conventional for speed, power, and area. The adders schematic diagrams are depicted in Fig.7 to and results of evaluation are tabulated in Table I. To demonstrate the advantages of the proposed ETA, we simulated the ETA along with conventional adders. Different error tolerant adder designs were compared with different bit delays. These adders were implemented in Verilog HDL. Delay, Power, Area analyses were carried out using Xilinx ISE 13.2. The results obtained are presented and compared. These adders were implemented in VHDL.Delay analyses were carried out using Xilinx ISE 13.2. To evaluate the efficiency of the proposed architecture,

4-BIT ERROR TOLERANT ADDER

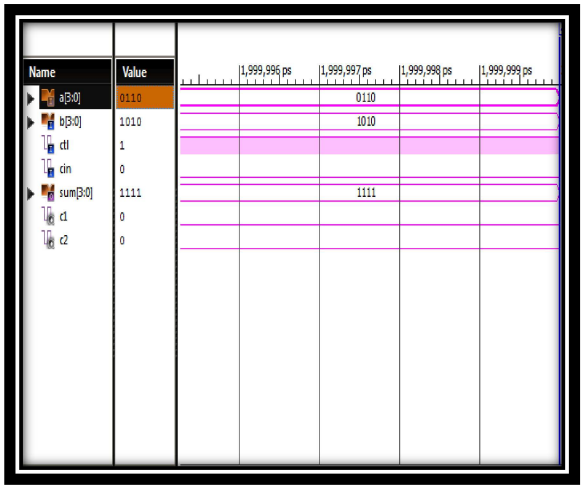


FIG 7: 4-BIT ERROR TOLERANT ADDER

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

Selected Device : 3s200ft256-4

 Number of Slices:                        3  out of   1920     0%
 Number of 4 input LUTs:                  5  out of   3840     0%
 Number of IOs:                          14
 Number of bonded IOBs:                  14  out of    173     8%
```

FIG 8: 4-BIT ERROR TOLERANT ADDER (DUS)

```
Delay:               10.724ns (Levels of Logic = 4)
  Source:            ctl (PAD)
  Destination:       sum<0> (PAD)

  Data Path: ctl to sum<0>
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     IBUF:I->O             1   0.821   1.140  ctl_IBUF (ctl_IBUF)
     LUT3:I0->O            2   0.551   1.216  i1/m1/cout1 (sum_1_OBUF)
     LUT3:I0->O            1   0.551   0.801  i2/m1/cout1 (sum_0_OBUF)
     OBUF:I->O                 5.644          sum_0_OBUF (sum<0>)
    ----------------------------------------
    Total                     10.724ns (7.567ns logic, 3.157ns route)
                                       (70.6% logic, 29.4% route)
```

FIG 9: 4-BIT ERROR TOLERANT ADDER (DELAY)

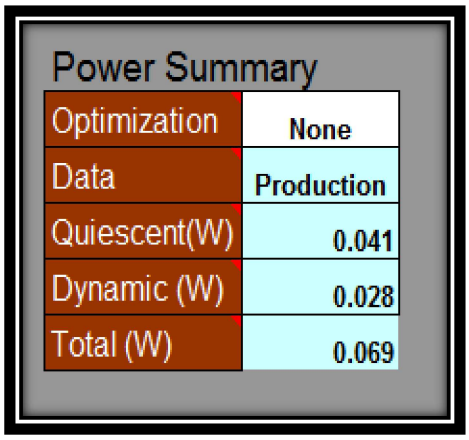


FIG 10: 4-BIT ERROR TOLERANT ADDER (POWER)

8-BIT ERROR TOLERANT ADDER

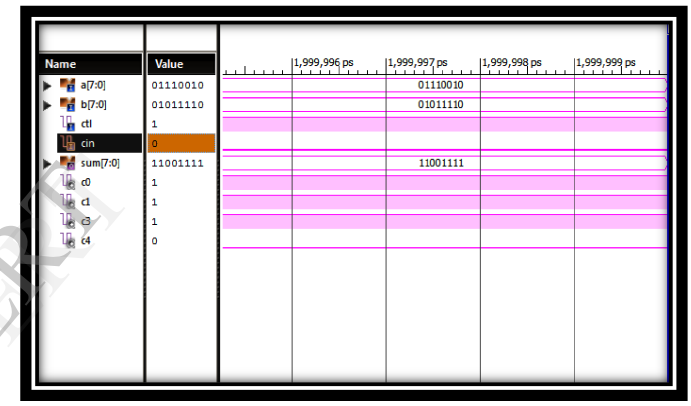


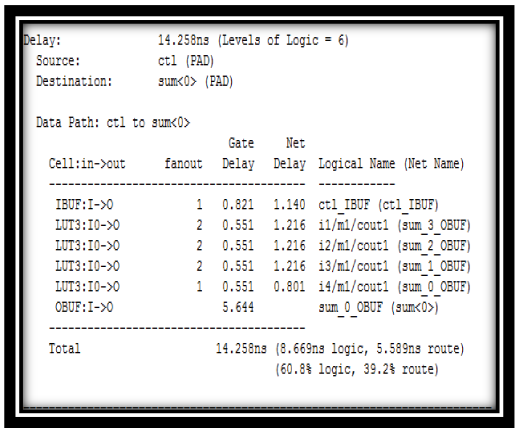
FIG 11: 8-BIT ERROR TOLERANT ADDER

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

Selected Device : 3s200ft256-4

 Number of Slices:                        6  out of   1920     0%
 Number of 4 input LUTs:                 11  out of   3840     0%
 Number of IOs:                          26
 Number of bonded IOBs:                  26  out of    173    15%
```

FIG 12: 8-BIT ERROR TOLERANT ADDER (DUS)



```
Delay:              14.258ns (Levels of Logic = 6)
  Source:           ctl (PAD)
  Destination:      sum<0> (PAD)

Data Path: ctl to sum<0>
                                   Gate     Net
  Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
  ----------------------------------------  ------------
   IBUF:I->O             1   0.821   1.140  ctl_IBUF (ctl_IBUF)
   LUT3:I0->O            2   0.551   1.216  i1/m1/cout1 (sum_3_OBUF)
   LUT3:I0->O            2   0.551   1.216  i2/m1/cout1 (sum_2_OBUF)
   LUT3:I0->O            2   0.551   1.216  i3/m1/cout1 (sum_1_OBUF)
   LUT3:I0->O            1   0.551   0.801  i4/m1/cout1 (sum_0_OBUF)
   OBUF:I->O                 5.644          sum_0_OBUF (sum<0>)
  ----------------------------------------
  Total                     14.258ns (8.669ns logic, 5.589ns route)
                                     (60.8% logic, 39.2% route)
```

FIG 13:_8-BIT ERROR TOLERANT ADDER (DELAY)

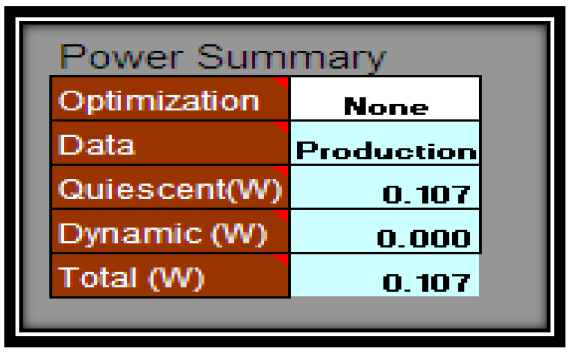


FIG 14:_8-BIT ERROR TOLERANT ADDER (POWER)

16-BIT ERROR TOLERANT ADDER

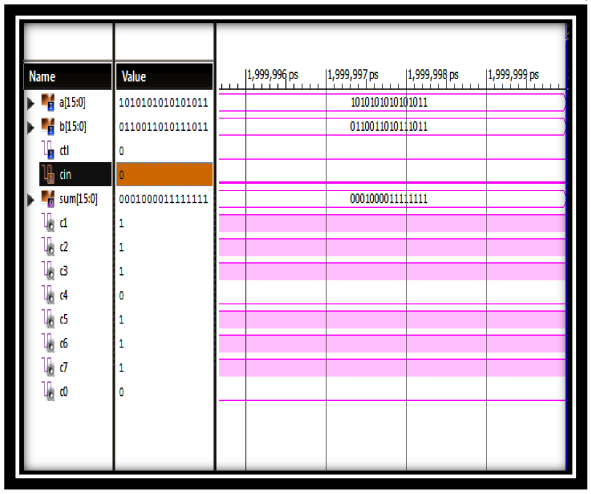
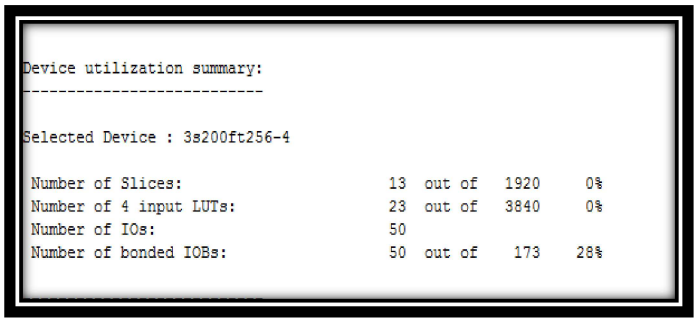


FIG 15:_16-BIT ERROR TOLERANT ADDER

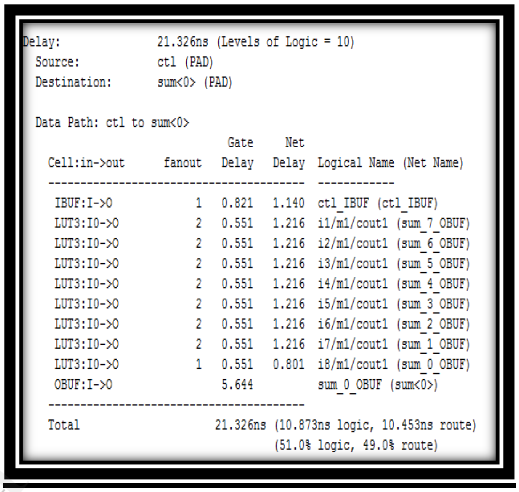


```
Device utilization summary:
---------------------------

Selected Device : 3s200ft256-4

 Number of Slices:                       13  out of   1920     0%
 Number of 4 input LUTs:                 23  out of   3840     0%
 Number of IOs:                          50
 Number of bonded IOBs:                  50  out of    173    28%
```

FIG 16:_16-BIT ERROR TOLERANT ADDER (DUS)



```
Delay:              21.326ns (Levels of Logic = 10)
  Source:           ctl (PAD)
  Destination:      sum<0> (PAD)

Data Path: ctl to sum<0>
                                   Gate     Net
  Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
  ----------------------------------------  ------------
   IBUF:I->O             1   0.821   1.140  ctl_IBUF (ctl_IBUF)
   LUT3:I0->O            2   0.551   1.216  i1/m1/cout1 (sum_7_OBUF)
   LUT3:I0->O            2   0.551   1.216  i2/m1/cout1 (sum_6_OBUF)
   LUT3:I0->O            2   0.551   1.216  i3/m1/cout1 (sum_5_OBUF)
   LUT3:I0->O            2   0.551   1.216  i4/m1/cout1 (sum_4_OBUF)
   LUT3:I0->O            2   0.551   1.216  i5/m1/cout1 (sum_3_OBUF)
   LUT3:I0->O            2   0.551   1.216  i6/m1/cout1 (sum_2_OBUF)
   LUT3:I0->O            2   0.551   1.216  i7/m1/cout1 (sum_1_OBUF)
   LUT3:I0->O            1   0.551   0.801  i8/m1/cout1 (sum_0_OBUF)
   OBUF:I->O                 5.644          sum_0_OBUF (sum<0>)
  ----------------------------------------
  Total                     21.326ns (10.873ns logic, 10.453ns route)
                                     (51.0% logic, 49.0% route)
```

FIG 17:_16-BIT ERROR TOLERANT ADDER (DELAY)

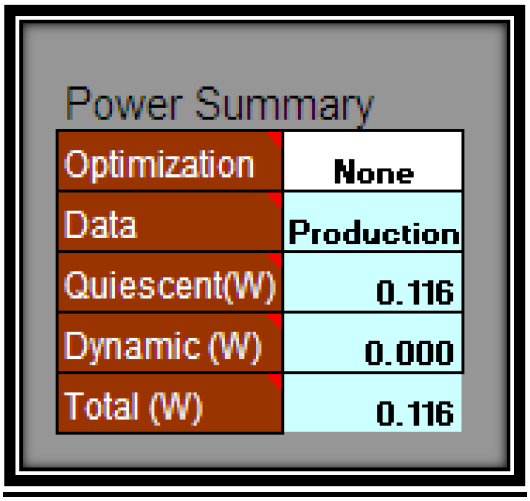
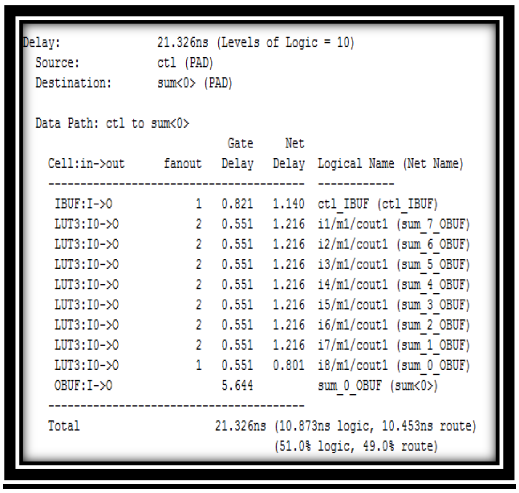


FIG 18:_16-BIT ERROR TOLERANT ADDER (POWER)



```
Delay:              21.326ns (Levels of Logic = 10)
  Source:           ctl (PAD)
  Destination:      sum<0> (PAD)

  Data Path: ctl to sum<0>
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     IBUF:I->O             1   0.821   1.140  ctl_IBUF (ctl_IBUF)
     LUT3:I0->O            2   0.551   1.216  i1/m1/cout1 (sum_7_OBUF)
     LUT3:I0->O            2   0.551   1.216  i2/m1/cout1 (sum_6_OBUF)
     LUT3:I0->O            2   0.551   1.216  i3/m1/cout1 (sum_5_OBUF)
     LUT3:I0->O            2   0.551   1.216  i4/m1/cout1 (sum_4_OBUF)
     LUT3:I0->O            2   0.551   1.216  i5/m1/cout1 (sum_3_OBUF)
     LUT3:I0->O            2   0.551   1.216  i6/m1/cout1 (sum_2_OBUF)
     LUT3:I0->O            2   0.551   1.216  i7/m1/cout1 (sum_1_OBUF)
     LUT3:I0->O            1   0.551   0.801  i8/m1/cout1 (sum_0_OBUF)
     OBUF:I->O                 5.644          sum_0_OBUF (sum<0>)
    ----------------------------------------
    Total                     21.326ns (10.873ns logic, 10.453ns route)
                                       (51.0% logic, 49.0% route)
```

FIG 17:_16-BIT ERROR TOLERANT ADDER (DELAY)

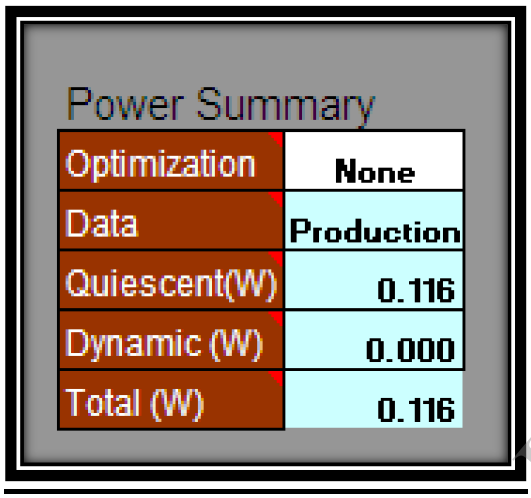


FIG 18:_16-BIT ERROR TOLERANT ADDER (POWER)

32-BIT ERROR TOLERANT ADDER

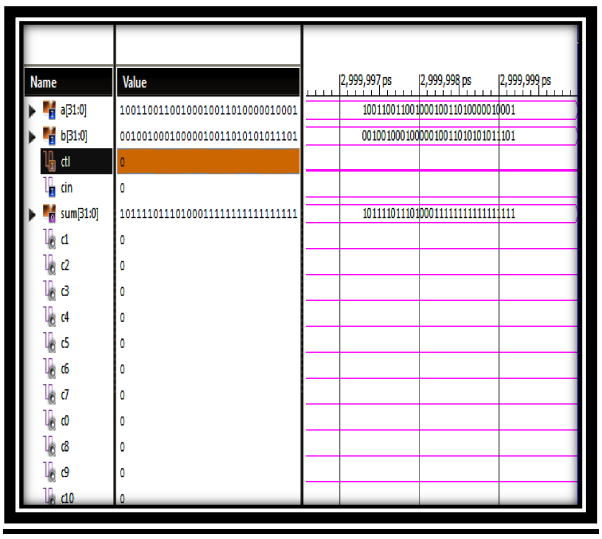
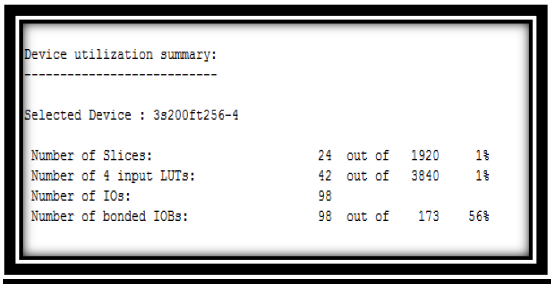


FIG 19:_32-BIT ERROR TOLERANT ADDER

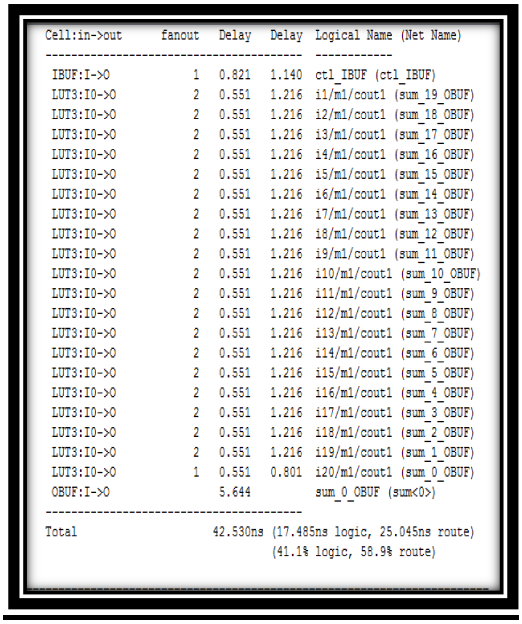


```
Device utilization summary:
---------------------------

Selected Device : 3s200ft256-4

 Number of Slices:                       24  out of   1920     1%
 Number of 4 input LUTs:                 42  out of   3840     1%
 Number of IOs:                          98
 Number of bonded IOBs:                  98  out of    173    56%
```

FIG 20: 32-BIT ERROR TOLERANT ADDER (DUS)



```
Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
----------------------------------------  ------------
 IBUF:I->O             1   0.821   1.140  ctl_IBUF (ctl_IBUF)
 LUT3:I0->O            2   0.551   1.216  i1/m1/cout1 (sum_19_OBUF)
 LUT3:I0->O            2   0.551   1.216  i2/m1/cout1 (sum_18_OBUF)
 LUT3:I0->O            2   0.551   1.216  i3/m1/cout1 (sum_17_OBUF)
 LUT3:I0->O            2   0.551   1.216  i4/m1/cout1 (sum_16_OBUF)
 LUT3:I0->O            2   0.551   1.216  i5/m1/cout1 (sum_15_OBUF)
 LUT3:I0->O            2   0.551   1.216  i6/m1/cout1 (sum_14_OBUF)
 LUT3:I0->O            2   0.551   1.216  i7/m1/cout1 (sum_13_OBUF)
 LUT3:I0->O            2   0.551   1.216  i8/m1/cout1 (sum_12_OBUF)
 LUT3:I0->O            2   0.551   1.216  i9/m1/cout1 (sum_11_OBUF)
 LUT3:I0->O            2   0.551   1.216  i10/m1/cout1 (sum_10_OBUF)
 LUT3:I0->O            2   0.551   1.216  i11/m1/cout1 (sum_9_OBUF)
 LUT3:I0->O            2   0.551   1.216  i12/m1/cout1 (sum_8_OBUF)
 LUT3:I0->O            2   0.551   1.216  i13/m1/cout1 (sum_7_OBUF)
 LUT3:I0->O            2   0.551   1.216  i14/m1/cout1 (sum_6_OBUF)
 LUT3:I0->O            2   0.551   1.216  i15/m1/cout1 (sum_5_OBUF)
 LUT3:I0->O            2   0.551   1.216  i16/m1/cout1 (sum_4_OBUF)
 LUT3:I0->O            2   0.551   1.216  i17/m1/cout1 (sum_3_OBUF)
 LUT3:I0->O            2   0.551   1.216  i18/m1/cout1 (sum_2_OBUF)
 LUT3:I0->O            2   0.551   1.216  i19/m1/cout1 (sum_1_OBUF)
 LUT3:I0->O            1   0.551   0.801  i20/m1/cout1 (sum_0_OBUF)
 OBUF:I->O                 5.644          sum_0_OBUF (sum<0>)
----------------------------------------
Total                     42.530ns (17.485ns logic, 25.045ns route)
                                   (41.1% logic, 58.9% route)
```

FIG 21:_32-BIT ERROR TOLERANT ADDER (DELAY)

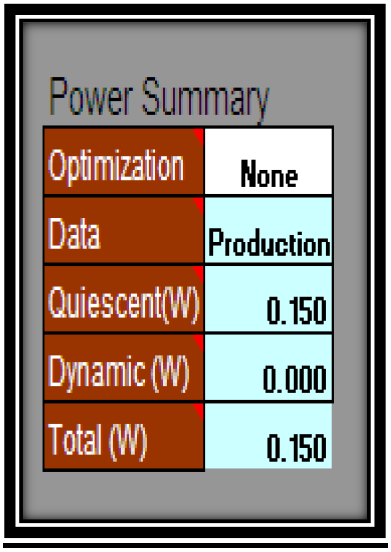


FIG 22: 32-BIT ERROR TOLERANT ADDER (POWER)

4-BIT CONVENTIONAL ADDER

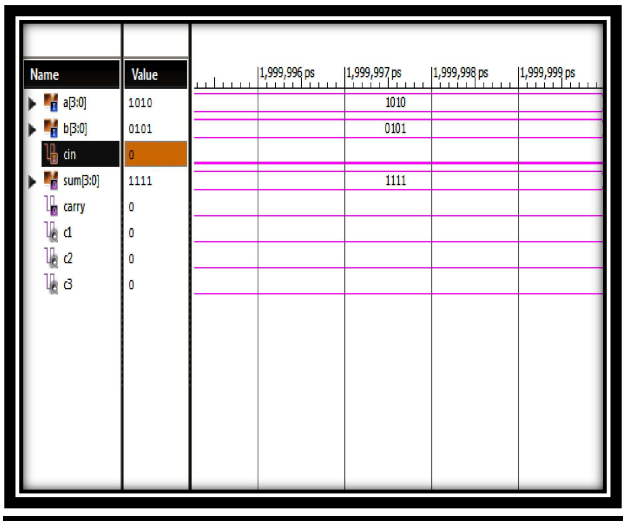


FIG 23: 4-BIT CONVENTIONAL ADDER

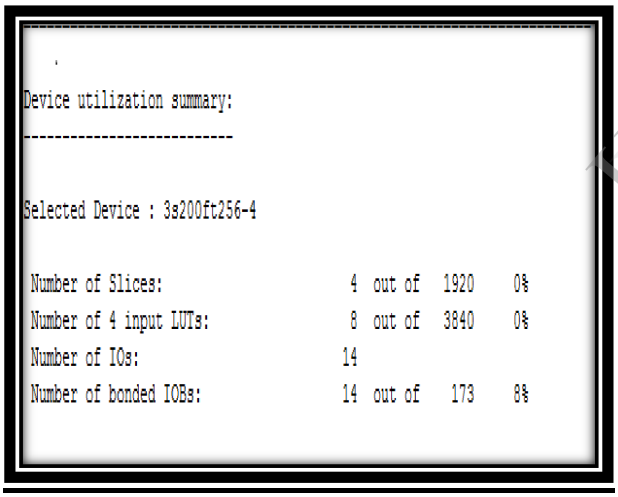


FIG 24: 4-BIT CONVENTIONAL ADDER (DUS)

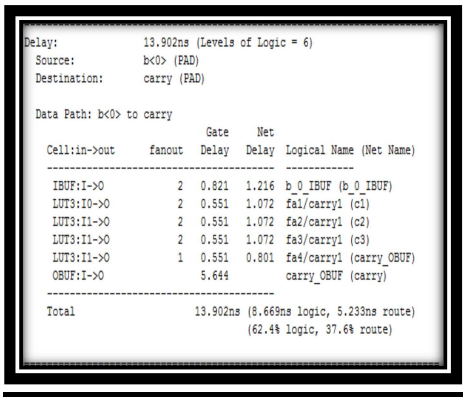


FIG 25: 4-BIT CONVENTIONAL ADDER (DELAY)

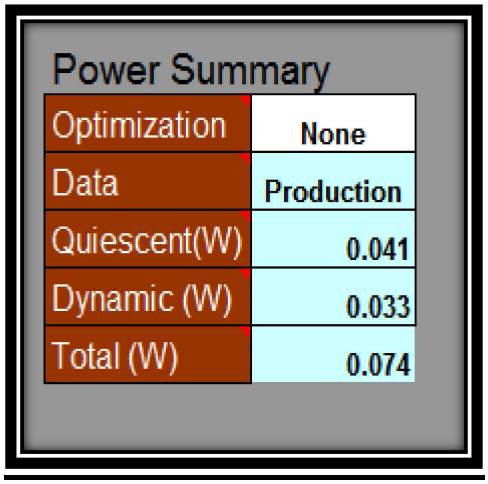


FIG 26: 4-BIT CONVENTIONAL ADDER (POWER)

8-BIT CONVENTIONAL ADDER

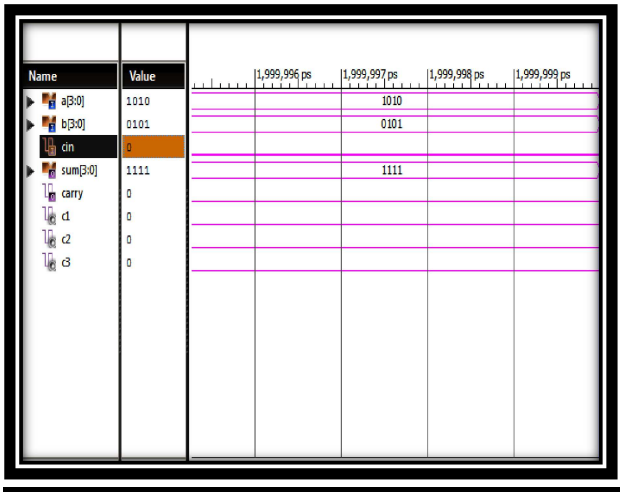
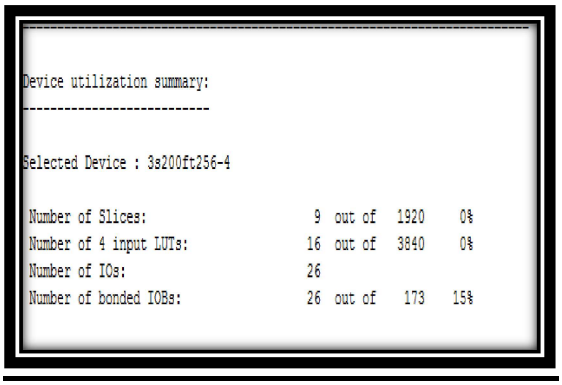


FIG 27: 8-BIT CONVENTIONAL ADDER

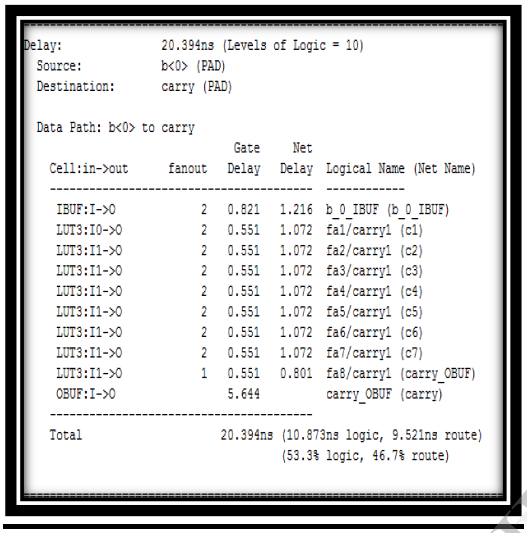


```
Device utilization summary:
---------------------------

Selected Device : 3s200ft256-4

 Number of Slices:                      9  out of   1920     0%
 Number of 4 input LUTs:               16  out of   3840     0%
 Number of IOs:                        26
 Number of bonded IOBs:                26  out of    173    15%
```

FIG 28:_8-BIT CONVENTIONAL ADDER (DUS)



```
Delay:               20.394ns (Levels of Logic = 10)
  Source:            b<0> (PAD)
  Destination:       carry (PAD)

  Data Path: b<0> to carry
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     IBUF:I->O             2   0.821   1.216  b_0_IBUF (b_0_IBUF)
     LUT3:I0->O            2   0.551   1.072  fa1/carry1 (c1)
     LUT3:I1->O            2   0.551   1.072  fa2/carry1 (c2)
     LUT3:I1->O            2   0.551   1.072  fa3/carry1 (c3)
     LUT3:I1->O            2   0.551   1.072  fa4/carry1 (c4)
     LUT3:I1->O            2   0.551   1.072  fa5/carry1 (c5)
     LUT3:I1->O            2   0.551   1.072  fa6/carry1 (c6)
     LUT3:I1->O            2   0.551   1.072  fa7/carry1 (c7)
     LUT3:I1->O            1   0.551   0.801  fa8/carry1 (carry_OBUF)
     OBUF:I->O                 5.644          carry_OBUF (carry)
    ----------------------------------------
    Total                     20.394ns (10.873ns logic, 9.521ns route)
                                       (53.3% logic, 46.7% route)
```

FIG 29:_8-BIT CONVENTIONAL ADDER (POWER)

16-BIT CONVENTIONAL ADDER

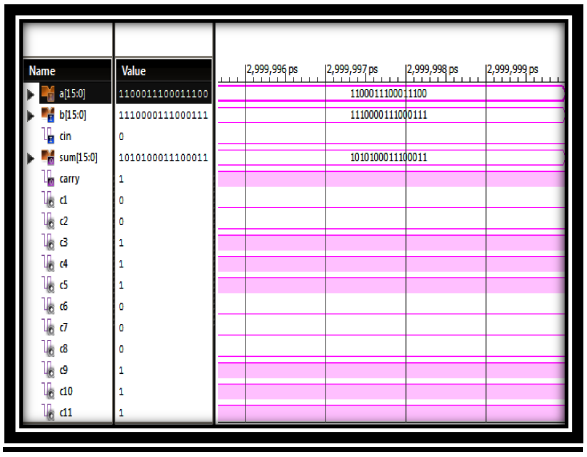


FIG 30:_16-BIT CONVENTIONAL ADDER

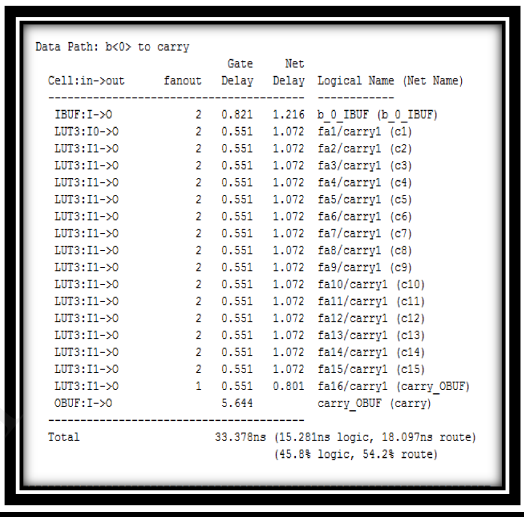

```
Device utilization summary:
---------------------------

Selected Device : 3s200ft256-4

 Number of Slices:                     18  out of   1920     0%
 Number of 4 input LUTs:               32  out of   3840     0%
 Number of IOs:                        50
 Number of bonded IOBs:                50  out of    173    28%

---------------------------
```

FIG 31:_16-BIT CONVENTIONAL ADDER (DUS)



```
Data Path: b<0> to carry
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     IBUF:I->O             2   0.821   1.216  b_0_IBUF (b_0_IBUF)
     LUT3:I0->O            2   0.551   1.072  fa1/carry1 (c1)
     LUT3:I1->O            2   0.551   1.072  fa2/carry1 (c2)
     LUT3:I1->O            2   0.551   1.072  fa3/carry1 (c3)
     LUT3:I1->O            2   0.551   1.072  fa4/carry1 (c4)
     LUT3:I1->O            2   0.551   1.072  fa5/carry1 (c5)
     LUT3:I1->O            2   0.551   1.072  fa6/carry1 (c6)
     LUT3:I1->O            2   0.551   1.072  fa7/carry1 (c7)
     LUT3:I1->O            2   0.551   1.072  fa8/carry1 (c8)
     LUT3:I1->O            2   0.551   1.072  fa9/carry1 (c9)
     LUT3:I1->O            2   0.551   1.072  fa10/carry1 (c10)
     LUT3:I1->O            2   0.551   1.072  fa11/carry1 (c11)
     LUT3:I1->O            2   0.551   1.072  fa12/carry1 (c12)
     LUT3:I1->O            2   0.551   1.072  fa13/carry1 (c13)
     LUT3:I1->O            2   0.551   1.072  fa14/carry1 (c14)
     LUT3:I1->O            2   0.551   1.072  fa15/carry1 (c15)
     LUT3:I1->O            1   0.551   0.801  fa16/carry1 (carry_OBUF)
     OBUF:I->O                 5.644          carry_OBUF (carry)
    ----------------------------------------
    Total                     33.378ns (15.281ns logic, 18.097ns route)
                                       (45.8% logic, 54.2% route)
```

FIG 32:_16-BIT CONVENTIONAL ADDER (DELAY)

32-BIT CONVENTIONAL ADDER

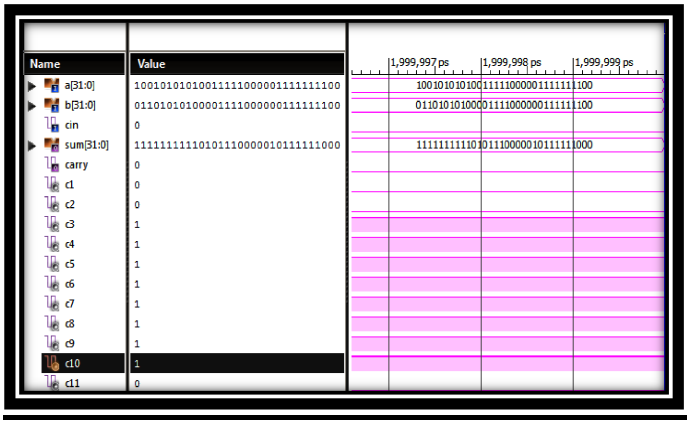


FIG 32:_32-BIT CONVENTIONAL ADDER

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

Selected Device : 3s200ft256-4

 Number of Slices:                       37  out of   1920    1%
 Number of 4 input LUTs:                 64  out of   3840    1%
 Number of IOs:                          98
 Number of bonded IOBs:                  98  out of    173   56%

---------------------------
```

FIG 33: 32-BIT CONVENTIONAL ADDER (DUS)

```
  Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
  ----------------------------------------  ------------
   IBUF:I->O             2   0.821   1.216  b_0_IBUF (b_0_IBUF)
   LUT3:I0->O            2   0.551   1.072  fa1/carry1 (c1)
   LUT3:I1->O            2   0.551   1.072  fa2/carry1 (c2)
   LUT3:I1->O            2   0.551   1.072  fa3/carry1 (c3)
   LUT3:I1->O            2   0.551   1.072  fa4/carry1 (c4)
   LUT3:I1->O            2   0.551   1.072  fa5/carry1 (c5)
   LUT3:I1->O            2   0.551   1.072  fa6/carry1 (c6)
   LUT3:I1->O            2   0.551   1.072  fa7/carry1 (c7)
   LUT3:I1->O            2   0.551   1.072  fa8/carry1 (c8)
   LUT3:I1->O            2   0.551   1.072  fa9/carry1 (c9)
   LUT3:I1->O            2   0.551   1.072  fa10/carry1 (c10)
   LUT3:I1->O            2   0.551   1.072  fa11/carry1 (c11)
   LUT3:I1->O            2   0.551   1.072  fa12/carry1 (c12)
   LUT3:I1->O            2   0.551   1.072  fa13/carry1 (c13)
   LUT3:I1->O            2   0.551   1.072  fa14/carry1 (c14)
   LUT3:I1->O            2   0.551   1.072  fa15/carry1 (c15)
   LUT3:I1->O            2   0.551   1.072  fa16/carry1 (c16)
   LUT3:I1->O            2   0.551   1.072  fa17/carry1 (c17)
   LUT3:I1->O            2   0.551   1.072  fa18/carry1 (c18)
   LUT3:I1->O            2   0.551   1.072  fa19/carry1 (c19)
   LUT3:I1->O            2   0.551   1.072  fa20/carry1 (c191)
   LUT3:I1->O            2   0.551   1.072  fa21/carry1 (c20)
   LUT3:I1->O            2   0.551   1.072  fa22/carry1 (c21)
   LUT3:I1->O            2   0.551   1.072  fa23/carry1 (c22)
   LUT3:I1->O            2   0.551   1.072  fa24/carry1 (c23)
```

```
   LUT3:I1->O            2   0.551   1.072  fa25/carry1 (c24)
   LUT3:I1->O            2   0.551   1.072  fa26/carry1 (c25)
   LUT3:I1->O            2   0.551   1.072  fa27/carry1 (c26)
   LUT3:I1->O            2   0.551   1.072  fa28/carry1 (c27)
   LUT3:I1->O            2   0.551   1.072  fa29/carry1 (c28)
   LUT3:I1->O            2   0.551   1.072  fa30/carry1 (c29)
   LUT3:I1->O            2   0.551   1.072  fa31/carry1 (c30)
   LUT3:I1->O            1   0.551   0.801  fa32/carry1 (carry_OBUF)
   OBUF:I->O                 5.644          carry_OBUF (carry)
  ----------------------------------------
  Total                     59.346ns (24.097ns logic, 35.249ns route)
                                     (40.6% logic, 59.4% route)
```

FIG 34: 32-BIT CONVENTIONAL (DELAY)

COMPARISON TABLE 4 TO 32 BIT ETA

TABLE1

ETA	4-BIT	8-BIT	16-BIT	32-BIT
POWER(W)	0 .069	0.107	0 .116	0.150
DELAY(ns)	10.724	14.258	21.326	42.530
AREA NO OF CELLS	36	69	136	262

COMPARISON OF ETA WITH CONVENTIONAL ADDER

TABLE2

PARAMETERS	ETA	CONVENTIONAL ADDER(RCA)
POWER(W)	0.069	0.074
DELAY(ns)	10.724	13.902

IV. CONCLUSION

The proposed Error Tolerant Adder trades a certain amount of accuracy for significant power saving and performance improvement. Extensive comparisons with conventional Adders i.e. Ripple Carry Adder is shown in the table.2 indicate that the proposed ETA out-performed the conventional Adders Applications Power Performance. conventional ripple carry adder and error tolerant adder were compared, and find out that the error tolerant adder be the minimum delay, area and power.

REFERENCES

[1] "Design of low- power high-speed error Tolerant adder" Journal of computer science 7 (12): 1839-1845, 20111ssn 1549- 3636 © 2011 science publications Corresponding author:1k.n.vijeyakumar.

[2] "Design of low-power high-speed truncation error- tolerant adder and its application in digital signal processing" ning zhu, wang ling goh, weija zhang, kiat seng yeo,and zhi hui kong ieee transactions on very large scale integration (vlsi) systems, vol. 18, no. 8, august 2010.

[3] "Design and error-tolerance in the presence of massive numbers of defects," M.A. Breuer, s. K. Gupta, and t. M Mak, ieee des. Test Computer., vol. 24, no. 3, pp. 216-227, may-jun. 2004.

[4] "A novel L. Sterpone, M. SonzaReorda and M. Violante, ―Evaluating Different Solutions to Design Fault Tolerant Sytems with SRAM based FPGAs,‖ Journal of Electronic Testing: Theory and Applications, vol. 23, pp. 47-54, 2007.

[5] K. Kyriakoulakos and D. Pnevmatikatos, ―A Novel SRAM-Based FPGA Architecture for Efficient TMR Fault Tolerance Support,‖ International Conference on Field

[6] Breuer, M.A., S.K. Gupta and T.M. Mak, 2004. Defect and error tolerance in the presence of massive numbers of defects. IEEE Des. Test Comp., 21216227.DOI:10.1109/MDT.2004.8

[7] Breuer, M.A. and H.H. Zhu, 2006. Error tolerance and multi-media. Proceedings of the International Conference Intelligent Information Hiding and Multimedia Signal Process, (IIHMSP‘ 06), IEEE Xplore Press, Pasadena, USA., pp: 521-524. DOI: 10.1109/IIHMSP.2006.265055

[8] Cheemalavagu, S., P. Korkmaz and K.V. Palem, 2004. Ultra low energy computing via probabilistic algorithms and devices: CMOS device primitives and the energy-probability relationship. Proceedings of the International Conference Solid StaDevices and Materials,(SSDM‘ 04), Tokyo,Japan,pp:402-403.