iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 5 **Issue:** VIII **Month of publication:** August 2017

DOI: http://doi.org/10.22214/ijraset.2017.8207

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor:6.887
Volume 5 Issue VIII, August 2017- Available at www.ijraset.com

Reliable Low-Power DADDA Multiplier Design Using Fixed-Width Replica Redundancy Block

K Venkatesh[1], V Siva Reddy[2]
[1]*Studying M.Tech. in VLSI & ES specialization,* [2]*Assistant Professor, Electronics and Communication Engineering department*
Velagapudi Ramakrishna Siddhartha Engineering College, Jawaharlal Nehru Technological University -Kakinada, Kanuru, Vijayawada-7, Andhra Pradesh

Abstract: ***The low power multiplier in this paper is consisting algorithmic noise tolerant (ANT) architecture with the fixed-width multiplier to build the reduced precision replica redundancy block (RPR).Using the ANT architecture meets the demand of high precision results with low power consumption with effective area efficiency. In this design we use fixed-width RPR with error compensation circuit through analyzing of probability and statistics. Here we use input correction vector and minor input correction vector partial products to lower the hardware complexity of error compensation circuit and the truncation errors. Using the fixed width RPR can lowered the area and save the power compared with full width RPR .In this paper we implement DADDA multiplier due to its faster multiplication mechanism and fewer gate requirement compared to other multipliers using same technique.***

Keywords: Algorithmic noise tolerant (ANT), Voltage over scaling (VOS), Reduced replica redundancy block (RPR), DADDA Multiplier.

I. INTRODUCTION

Low power consumption and smaller area are some of the most important criteria in the DSP systems and high performance systems . The low power technique is the voltage over scaling (VOS), was proposed in lower supply voltage beyond critical supply voltage without sacrificing the throughput.

A novel algorithmic noise tolerant (ANT) technique combined VOS main block with reduced precision replica (RPR), which combats soft errors effectively while achieving significant energy saving. VOS increase the delay in all paths of a system and may limit high performance required by today complex applications ANT is the combined VOS block and RPR block, the error is occur. It is a very fast manner but hardware complexity is too difficult.

The Rapid growth of portable and wireless computing systems in recent years drives the need for ultralow power systems. To lower the power dissipation, supply voltage scaling is widely used as an effective low-power technique since the power consumption in CMOS circuits is proportional to the square of supply voltage. However, in deep-sub micrometer process technologies, noise interference problems have raised difficulty to design the reliable and efficient microelectronics systems; hence, the design techniques to enhance noise tolerance have been widely developed. An aggressive low-power technique, referred to as voltage over scaling (VOS), was proposed in to lower supply voltage beyond critical supply voltage without sacrificing the throughput. However, VOS leads to severe degradation in signal-to-noise ratio (SNR). A novel algorithmic noise tolerant (ANT) technique combined VOS main block with reduced-precision replica (RPR), which combats soft errors effectively while achieving significant energy saving. Some ANT deformation designs are presented in and the ANT design concept is further extended to system level in. However, the RPR designs in the ANT designs of are designed in a customized manner, which are not easily adopted and repeated. The RPR designs in the ANT designs of and can operate in a very fast manner, but their hardware complexity is too complex. As a result, the RPR design in the ANT design of is still the most popular design because of its simplicity. However, adopting with RPR in should still pay extra area overhead and power consumption. In this paper, we further proposed an easy way using the fixed-width RPR to replace the full-width RPR block in. Using the fixed-width RPR, the computation error can be corrected with lower power consumption and lower area overhead. We take use of probability, statistics, and partial product weight analysis to find the approximate compensation vector for a more precise RPR design. In order not to increase the critical path delay, we restrict the compensation circuit in RPR must not be located in the critical path. As a result, we can realize the ANT design with smaller circuit area, lower power consumption, and lower critical supply voltage.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor:6.887
Volume 5 Issue VIII, August 2017- Available at www.ijraset.com

II. MULTIPLIER

Multipliers are key components of many high performance systems such as FIR filters, microprocessors, digital signal processors, etc. A system's performance is generally determined by the performance of the multiplier because the multiplier is generally the slowest element in the system.

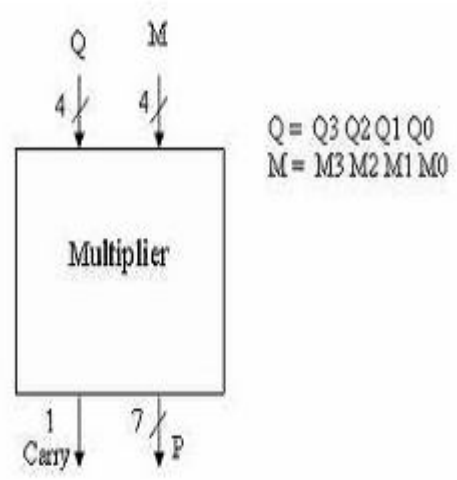


Figure 1. Multiplier

A. Fixed Width Multiplier

The fixed-width multipliers have been widely y used in the design of digital signal processor(DSP) due to their smaller area and lower power dissipation. In order to reduce the chip area of channel detector for cognitive radio, many fixed width Booth multipliers have been used. However, they reduce the detection accuracy because of truncated partial products. This method can reduce the truncated error by using variable compensation value. The third category is hybrid error compensation, which uses both constant and adaptive QEC techniques together to reduce the truncated error. In order to overcome the disadvantages of has presented a method of dividing the truncated partial products into the major truncated section and the minor truncated section.

B. Algorithmic Noise Tolerance

The motivation is to reduce power of the traditional methods for noise tolerance. ANT can be mainly divided into Prediction based ANT and Reduced Precision redundancy based ANT. Using ANT technique to improve the performance of DSP algorithms in presence of bit error rates. Therefore ANT can produce more effective signals. ANT to compensate for degradation in the system output due to errors from soft computations.

C. Reduced Precision Redundancy

The MDSP block is subject to VOS, which results in soft errors in its output . When a soft error in MDSP is detected using an error control (EC) block, the RPR output is used as an output . Next, we describe the error characteristics of a system under VOS and then present the proposed error control algorithm.

D. Soft Error Characteristics

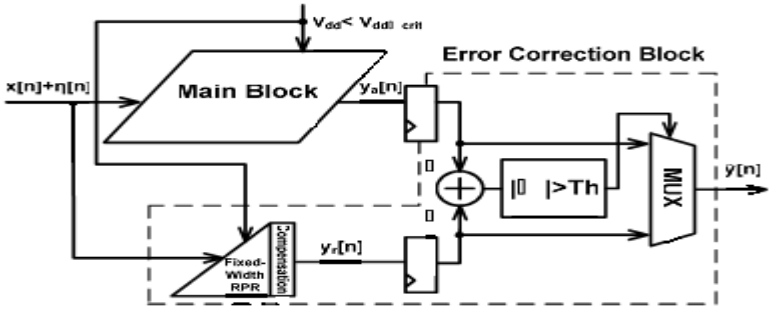
Voltage over scaling introduces input dependent soft errors whenever a path with delay greater than the sample period is excited. Since the arithmetic units employed in DSP systems are based on least significant bit (LSB) first computation, soft errors appear first in the most significant bits (MSBs), resulting in errors of large magnitude. These errors severely degrade the performance but are desirable because they are easy to detect. This fraction depends upon the delay distribution of a system, which in turn depends on the architecture. The path delay distribution for all possible input combinations of an 8× 8 Baugh Wooley multiplier.

E. ANT Multiplier design Using Fixed-Width RPR

We further proposed the fixed-width RPR to replace the full-width RPR block in the ANT design, as shown in Fig. 2, which can not only provide higher computation precision, lower power consumption, and lower area overhead in RPR, but also perform with higher SNR, more area efficient, lower operating supply voltage, and lower power consumption in realizing the ANT architecture. We demonstrate our fixed-width RPR-based ANT design in an ANT multiplier. The fixed-width designs are usually applied in DSP applications to avoid infinite growth of bit width. Cutting off n-bit least significant bit (LSB) output is a popular solution to construct a fixed-width DSP with n-bit input and n-bit output. The hardware complexity and power consumption of a fixed-width DSP is usually about half of the full-length one. However, truncation of LSB part results in rounding error, which needs to be compensated precisely. Many literatures have been presented to reduce the truncation error with constant correction value or with variable correction value. The circuit complexity to compensate with constant corrected value can be simpler than that of variable correction value; however, the variable correction approaches are usually more precise.

Their compensation method is to compensate the truncation error between the full-length multiplier and the fixed-width multiplier. However, in the fixed-width RPR of an ANT multiplier, the compensation error we need to correct is the overall truncation error of MDSP block. Unlike, our compensation method is to compensate the truncation error between the full-length MDSP multiplier and the fixed-width RPR multiplier. In nowadays, there are many fixed-width multiplier designs applied to the full-width multipliers. However, there is still no fixed-width RPR design applied to the ANT multiplier designs.

To achieve more precise error compensation, we compensate the truncation error with variable correction value. We construct the error compensation circuit mainly using the partial product terms with the largest weight in the least significant segment. The error compensation algorithm makes use of probability, statistics, and linear regression analysis to find the approximate compensation value. To save hardware complexity, the compensation vector in the partial product terms with the largest weight in the least significant segment is directly inject into the fixed-width RPR, which does not need extra compensation logic gates. To further lower the compensation error, we also consider the impact of truncated products with the second most significant bits on the error compensation. We propose an error compensation circuit using a simple minor input correction vector to compensation the error remained. In order not to increase the critical path delay, we locate the compensation circuit in the noncritical path of the fixed-width RPR. As compared with the full-width RPR design the proposed fixed-width RPR multiplier not only performs with higher SNR but also with lower circuitry area and lower power consumption.



Proposed ANT architecture with fixed-width RPR

F. Proposed Precise Error Compensation Vector for Fixed-Width RPR Design

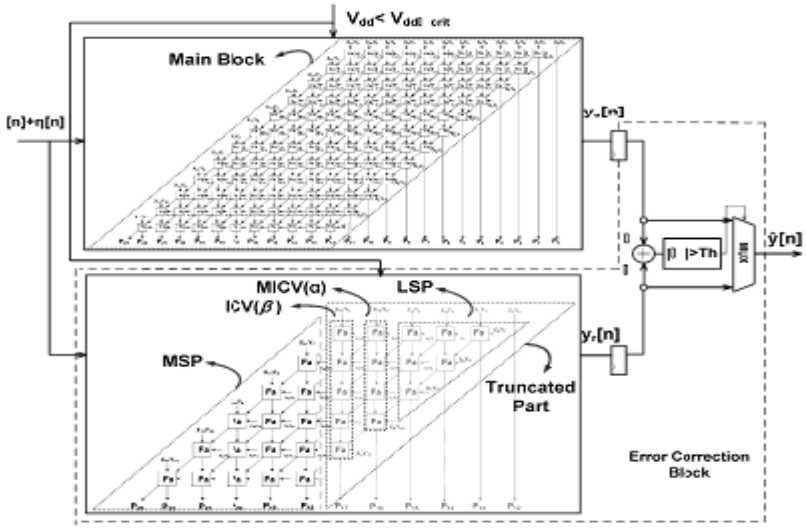
In the ANT design, the function of RPR is to correct the errors occurring in the output of MDSP and maintain the SNR of whole system while lowering supply voltage. In the case of using fixed-width RPR to realize ANT architecture, we not only lower circuit area and power consumption, but also accelerate the computation speed as compared with the conventional full-length RPR. However, we need to compensate huge truncation error due to cutting off many hardware elements in the LSB part of MDSP.

In the MDSP ofn-bit ANT Baugh–Wooley array multiplier, its two unsigned n-bit inputs of XandYcan be expressed as

$$X = \sum_{i=0}^{n-1} x_i \cdot 2^i, \qquad Y = \sum_{j=0}^{n-1} y_j \cdot 2^j.$$

The multiplication result P is the summation of partial products of xiyj, which is expressed as

$$P = \sum_{k=0}^{2n-1} p_k \cdot 2^k = \sum_{j=0}^{n-1}\sum_{i=0}^{n-1} x_i y_j \cdot 2^{i+j}.$$



12×12 bit ANT multiplier is implemented with the six-bit fixed width replica redundancy block.

The (n/2)-bit unsigned full-width Baugh–Wooley partial product array can be divided into four subsets, which are most significant part (MSP), input correction vector [ICV(β)], minor ICV [MICV(α)], and LSP, as shown in Fig. 3. In the fixed width RPR, only MSP part is kept and the other parts are removed. Therefore, the other three parts of ICV(β), MICV(α), and LSP are called as truncated part. The truncated ICV (β) and MICV(α)are the most important parts because of their highest weighting. Therefore, they can be applied to construct the truncation error compensation algorithm. To evaluate the accuracy of a fixed-width RPR, we can exploit the difference between the (n/2)-bit fixed-width RPR output and the 2n-bit full-length MDSP output, which is expressed as

$$\varepsilon = P - P_t$$

Where P is the output of the complete multiplier in MDSP and Pt is the output of the fixed-width multiplier in RPR. Pt can be expressed as

$$\begin{aligned} P_t &= \sum_{j=\frac{n}{2}+1}^{n-1} y_j 2^j \sum_{i=\frac{3n}{2}-j}^{n-1} x_i 2^i \\ &\quad + f\left(x_{n-1}y_{\frac{n}{2}}, x_{n-2}y_{\frac{n}{2}+1}, x_{n-3}y_{\frac{n}{2}+2}, \ldots, x_{\frac{n}{2}}y_{\frac{n}{2}+2}\right) \\ &\quad + f\left(x_{n-2}y_{\frac{n}{2}}, x_{n-3}y_{\frac{n}{2}+1}, x_{n-4}y_{\frac{n}{2}+2}, \ldots, x_{\frac{n}{2}}y_{n-2}\right) \\ &= \sum_{j=\frac{n}{2}+1}^{n-1} y_j 2^j \sum_{i=\frac{3n}{2}-j}^{n-1} x_i 2^i + f\,(\mathrm{ICV}) + f\,(\mathrm{MICV}) \\ &= \sum_{j=\frac{n}{2}+1}^{n-1} y_j 2^j \sum_{i=\frac{3n}{2}-j}^{n-1} x_i 2^i + f\,(\mathrm{EC}) \end{aligned}$$

III. DADDA MULTPLIER

The Dadda multiplier is a hardware multiplier design invented by computer scientist Luigi Dadda in 1965. It is similar to the Wallace multiplier, but it is slightly faster (for all operand sizes) and requires fewer gates (for all but the smallest operand sizes).
In fact, Dadda and Wallace multipliers have the same 3 steps for two bit strings w_1 and w_2 of lengths l1 and $l_2$respectively:

Multiply (logical AND) each bit w_1 of, by each bit w_2 of, yielding l_1*l_2 results, grouped by weight in columns
Reduce the number of partial products by stages of full and half adders until we are left with at most two bits of each weight.
Add the final result with a conventional adder.

As with the Wallace multiplier, the multiplication products of the first step carry different weights reflecting the magnitude of the original bit values in the multiplication. For example, the product of bits a_nb_m has weight n+m.
Unlike Wallace multipliers that reduce as much as possible on each layer, Dadda multipliers attempt to minimize the number of gates used, as well as input/output delay. Because of this, Dadda multipliers have a less expensive reduction phase, but the final numbers may be a few bits longer, thus requiring slightly bigger adders.
To achieve a more optimal final product, the structure of the reduction process is governed by slightly more complex rules than in Wallace multipliers.

The progression of the reduction is controlled by a maximum-height sequence d_j, defined by:

$d_1 = 2$ and d_{j+1}=floor (1.5 *d_j}

This yields a sequence like so:

$d_1 = 2$, $d_2 = 3$, $d_3 = 4$, $d_4 = 6$, $d_5 = 9$, $d_6 = 13$

The initial value of j is chosen as the largest value such that d_j < max (n_1, n2), where n_1 and n_2 are the number of bits in the input multiplicand and multiplier. The larger of the two bit lengths will be the maximum height of each column of weights after the first stage of multiplication. For each stage j of the reduction, the goal of the algorithm is the reduce the height of each column so that it is less than or equal to the value ofd_j.

For each stage from j..1, reduce each column starting at the lowest-weight column, c0 according to these rules:

If height(c_i) ≤ d_j the column does not require reduction, move to column c_{i+1}
If height (c_i) ≤ d_{j+1} add the top two elements in a half-adder, placing the result at the bottom of the column and the carry at the top of column c_{i+1}, then move to column c_{i+1}.
Else, add the top three elements in a full-adder, placing the result at the bottom of the column and the carry at the top of column c_{i+1}, restart c_i at step 1.

Algorithm:-
The example in the image on the right illustrates the reduction of an 8x8 multiplier, explained here.
The initial state j=4 is chosen as $d_4 = 6$, the largest value less than 8.

Stage j=4, $d_4 = 6$
Height(c_0…..c_5) are all less than or equal to six bits in height, so no changes are made
Height (c_6) =d_4 + 1 = 7, so a half-adder is applied, reducing it to six bits and adding its carry bit to c_7 .
Height(c_7)= 9 including the carry bit from c_6 , so we apply a full-adder and a half-adder to reduce it to six bits
Height(c_8) = 9 including two carry bits from c_7, so we again apply a full-adder and a half-adder to reduce it to six bits
Height(c_9) = 8 including two carry bits from c_8, so we apply a single full-adder and reduce it to six bits
Height(c_{10}…..c_{14}) are all less than or equal to six bits in height including carry bits, so no changes are made

Stage j=3, $d_3 = 4$
Height(c_0…..c_3) are all less than or equal to four bits in height, so no changes are made
Height(c_4)=d_3 + 1 = 5, so a half-adder is applied, reducing it to four bits and adding its carry bit to c_5

Height(c_5)= 7 including the carry bit from c_4, so we apply a full-adder and a half-adder to reduce it to four bits
Height(c_6…..c_{10}) = 8 including previous carry bits, so we apply two full-adders to reduce them to four bits
Height(c_{11}) = 6 including previous carry bits, so we apply a full-adder to reduce it to four bits
Height(c_{12}…..c_{14}) are all less than or equal to four bits in height including carry bits, so no changes are made Stage

Stage j=2, $d_2 = 3$
Height(c_0…..c_2) are all less than or equal to three bits in height, so no changes are made
Height(c_3)=$d_2 + 1 = 4$, so a half-adder is applied, reducing it to three bits and adding its carry bit to c_4
Height(c_4…..c_{12}) = 5 including previous carry bits, so we apply one full-adder to reduce them to three bits
Height(c_{13}…..c_{14}) are all less than or equal to three bits in height including carry bits, so no changes are made

Stage j=1, $d_1 = 2$

Height(c_0…..c_1) are all less than or equal to two bits in height, so no changes are made
Height(c_2)=$d_1 + 1 = 3$, so a half-adder is applied, reducing it to two bits and adding its carry bit to c_3
Height(c_3…..c_{13}) = 4 including previous carry bits, so we apply one full-adder to reduce them to two bits
Height(c_{14}) = 2 including the carry bit from c_{13}, so no changes are made

Addition
The output of the last stage leaves 14 columns of height two or less which can be passed into a standard adder.

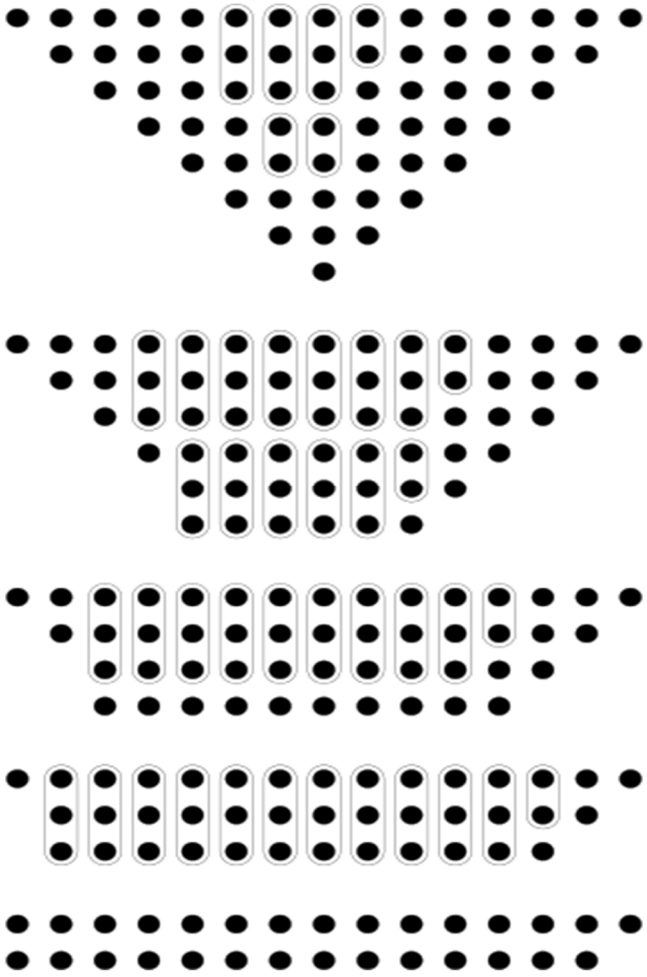


Fig: Reduction on 8x8 Dadda Multiplier

IV. EVALUATION AND COMPARISONS

A. *12x12 ANT with Normal Multiplier Results*

1) Area:

Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slices	221	4656	4%
Number of Slice Flip Flops	36	9312	0%
Number of 4 input LUTs	389	9312	4%
Number of bonded IOBs	50	232	21%
Number of GCLKs	1	24	4%

2) Synthesis Report of Normal Multiplier:

a) Timing Summary:

i) Speed Grade: 5

ii) Minimum Period: No path found

iii) Minimum Input Arrival Time Before Clock: 25.639ns

iv) Maximum Output Required Time After Clock: 10.918ns

v) Maximum Combinational Path Delay: No path found

B. *Extension with DADDA Results:*

1) Area:

Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slices	221	4656	4%
Number of Slice Flip Flops	61	9312	0%
Number of 4 input LUTs	383	9312	4%
Number of bonded IOBs	38	232	16%
Number of GCLKs	1	24	4%

2) Synthesis Report of DADDA Multiplier:

a) Timing Summary:

i) *Speed Grade:* 5

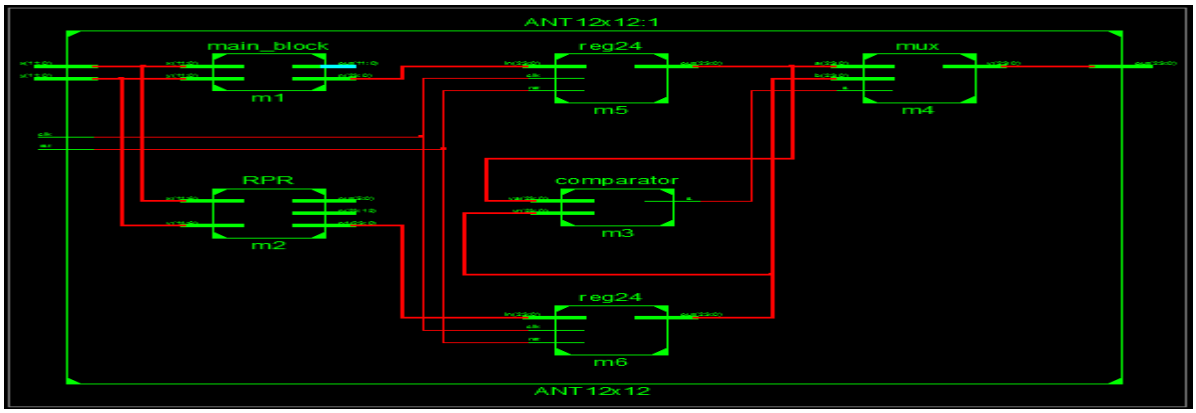
ii) *Minimum Period:* No path found

iii) *Minimum Input Arrival Time Before Clock:* 25.639ns

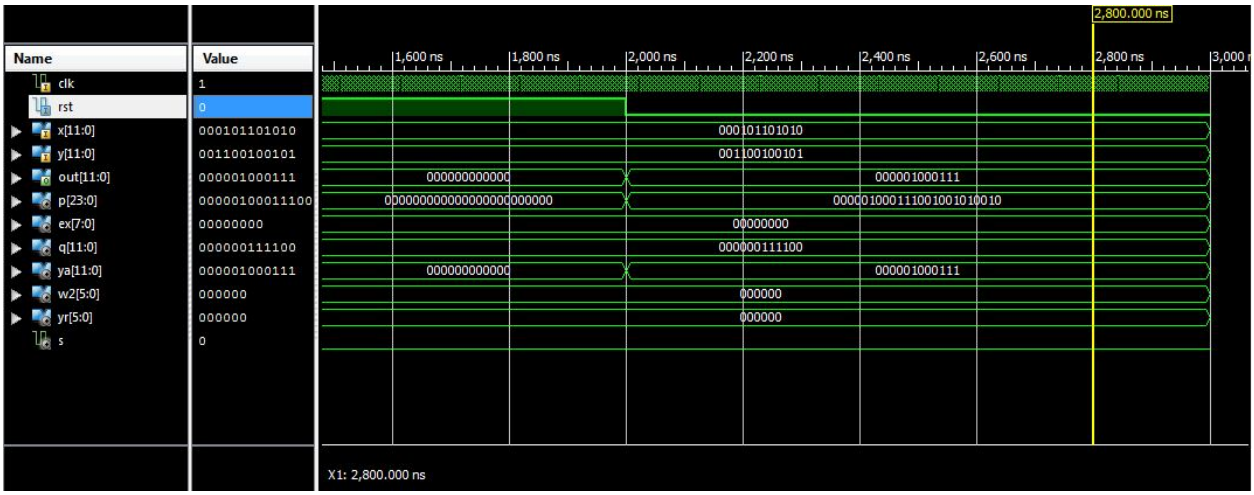
iv) *Maximum Output Required Time After Clock:* 10.918ns

v) *Maximum Combinational Path Delay:* No path found

C. *RTL SCH*

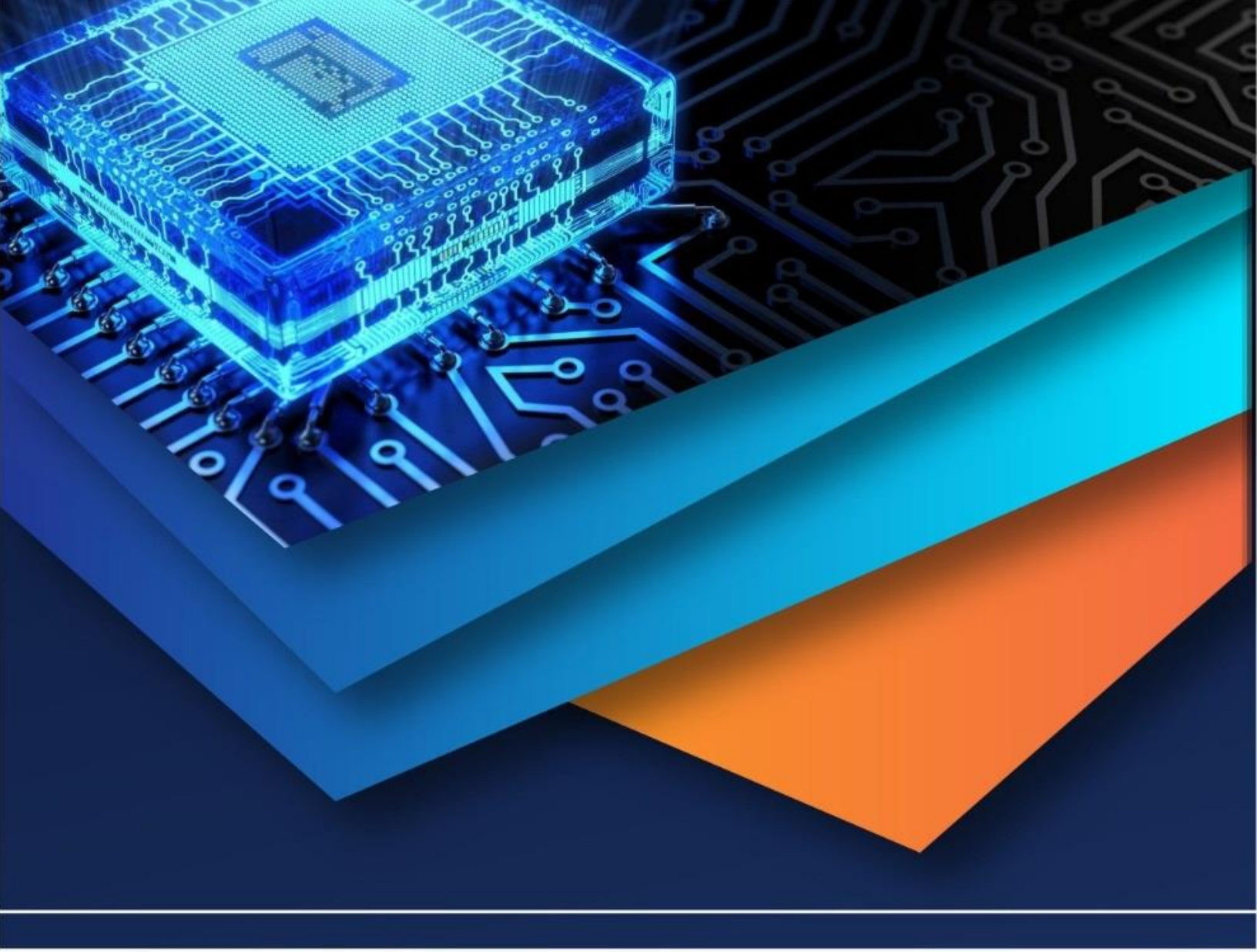


1) Simulation Results:



REFERENCES

[1] Reliable Low-Power Multiplier Design Using Fixed-Width Replica Redundancy Block IEEE Trans., March 2016.
[2] B. Shim, S. Sridhara, and N. R. Shanbhag, “Reliable low-power digital signal processing via reduced precision redundancy,”IEEE Trans.May 2004.
[3] B. Shim and N. R. Shanbhag, “Energy-efficient soft-error tolerant digital Signal processing,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., Apr. 2006.
[4] V. Gupta, D. Mohapatra, A. Raghunathan, and K. Roy, “Low-power digital signal processing using approximate adders,” IEEE Trans. Comput. Jan. 2013.
[5] Y. Liu, T. Zhang, and K. K. Parhi, “Computation error analysis in Digital signal processing systems with over scaled supply voltage,”IEEE Apr. 2010.
[6] “Low power digital signal processing Scheme via stochastic logic protection,” Syst., May 2012.
[7] S. S. Kidambi, F. El-Guibaly, and A. Antoniou, “Area-efficient multipliers for digital signal processing applications, ”IEEE Trans. Circuits Feb. 1996.
[8] J. M. Jou, S. R. Kuang, and R. D. Chen, “Design of low-error fixed-width multipliers for DSP applications, ”IEEE Trans. Circuits Syst Jun. 1999.
[9] S. J. Jou and H. H. Wang, “Fixed-width multiplier for DSP application,” in Proc. IEEE Int. Symp. Comput. Des., Sep. 2000.
[10] F. Curticapean and J. Niittylahti, “A hardware efficient direct digital frequency synthesizer,” in Proc. 8th IEEE Int. Conf. Electron., Circuits,Sep. 2001 .



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY