ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijareeie.com

Vol. 6, Issue 11, November 2017

Low Power DADDA Multiplier Design using Adaptive Hold Logic for Canny Edge Detection

S.Mamatha

PG Scholar, Dept. of ECE, SWEC, Hyderabad, TS, India

ABSTRACT: Real time video and image processing is used in wide variety of applications such as edge detection and image enhancement from video surveillance systems. These operations typically required very high computation power and area. The paper presents design and hardware implementation of real time multiplier for canny edge detection circuits on ASIC(Application Specific integrated circuit)& SOC(System on chip) using adaptive hold logic techniques with modified dadda concepts using partial product suppression techniques method. Through extensive experimental evaluation we applied a new partial product technique method is used on different multiplier architectures. The experimental results shows that compared with the existing designs, the new partial product method delivers the reduction of area up to 13%,power consumption reduced 10% and 12% increased in critical delay. The results show that the proposed multiplier excels them in terms power and area.

KEYWORDS: Adaptive Hold logic Modified DADDA Multiplier, Razor Flip Flops, Encoder Decoder.

I. INTRODUCTION

The real time image processing technique is used in video-surveillance, efficient human computer interaction ,traffic monitoring and medical image surgery ,this image guided surgery is used in neurosurgery and sinus surgery which is a great achievement because of the risks involved in such disturbances or any interferences. The digital image processing technique is used to process the normal image by quantization and sampling.

For edge detection methods, the Canny edge detector, first or second derivative operator methods are popular choices. As for corner feature detection, the SUSAN corner detector [1] and Harris corner detectors [2] are effective tools. These algorithms are local operators that involve operations of nearly all pixels in an image. Therefore, the running time of these kinds of operators are directly proportional to the size of the image and the number of the neighbouring pixels required to be examined. This property prohibits the use of these methods on a small microcontroller because of the limited processing power and memory size available. Therefore, unless special hardware chips (for example, Digital Signal Processor (DSP) and Single Instruction Multiple Destination (SIMD) processor) are used, it is difficult to achieve a high throughput computer vision system at low cost. Many people are interested in finding hardware solutions for computer vision algorithms. For example, it is possible to use FPGAs for building feature detectors to enable them to be more efficient in processing and power consumption. In the paper [3] the whole Harris detector is implemented using purely FPGAs. The speed is fast but the function may not be flexible enough to cater for different applications. That means you need to redesign the whole system for a specific task and specification. To make the system more flexible, some projects have demonstrated that the computationally intensive computer vision algorithms can be running at high frame rate on heterogeneous systems with an ordinary microcontroller and an FPGA. There are other examples such as [4], Benedetti and Perona designed and implemented a real-time 2D feature detection algorithm on FPGA. In the many projects the feature detection system is based on the work of feature detectors such as that in [5]-[6] and the KLT tracker [7].

However all these depend on image gradient information in both horizontal (x) and vertical (y) directions. To compute these image gradients, the whole image is convolved with a kernel and it is therefore very time consuming. There are

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijareeie.com

Vol. 6, Issue 11, November 2017

also recent projects using FPGAs for computer vision, examples are [8], [9] and [10]. However, they are using high speed FPGAs for the vision tasks, the performances are good but the costs are relatively higher. Leaser, Miller and Yu [11] designed a smart camera setup based on FPGAs and demonstrated two very different applications of this setup: medical image processing and fluid dynamics computation. Part of the computation is shared by the FPGA and the result is sent to a computer for further processing. The first application requires comparing eight 11 ×11 templates to be operated over the whole image. While cross-correlation of two areas (40×40 and 32×32) over two images of size 1008×1016 is needed in the second application. In both applications, they have speedups of over 20 times compared to a pure software implementation. There is a also hybrid approach such as the one by Tippetts *et al*. [12], it implemented a Harris Feature detector and priority queue using an FPGA.Georgios zervakis[13] implemented a design efficient approximate multiplication circuits through partial product and done application of canny edge detection using DADDA[4:2] multiplier.

High speed multiplication is a primary requirement of high performance digital systems. In recent trends the column compression multipliers are popular for high speed computations due to their higher speeds [14-15]. The first column compression multiplier was introduced by Wallace in [16]. He reduced the partial product of *N* rows by grouping into sets of three row set and two row set using (3,2) counters and (2,2) counters respectively.

Dadda altered the approach of Wallace by starting with the exact placement of the (3,2) counters and (2,2) counters in the maximum critical path delay of the multiplier and the hardware required for Dadda multiplier is lesser than the Wallace multiplier. The column compression multipliers have total delays that are proportional to the logarithm of the operand word lengths which is unlike the array multipliers which have speeds proportional to the word length [17-18]. The total delay of the multiplier can be split up into three parts: due to the Partial Product Generation (PPG), the Partial Product Summation.

Tree (PPST), and finally due to the Final Adder [19]. Of these the dominant components of the multiplier delay are due to the PPST and the final adder. The relative delay due to the PPG is small. Therefore significant improvement in the speed of the multiplier can be achieved by reducing the delay in the PPST and the final adder stage of the multiplier.

In [20] proposed a new efficient approximate multiplier with partial product perforation for canny edge detection. This method provides better efficiency in terms of PSNR.But fails Hardware efficiency like area and power when it implement on ASIC TSMC 65nm standard cells library due to difficulty of partial product perforation. To improve the hardware parameters in this paper propose a new design of modified dadda multiplier with PPST and using AHL circuit.

The AHL circuit can determine which input pattern to need one or two cycles and then the circuit decides to make suitable the judging block to reduce timing waste and also low power consumption occurring in traditional circuits that use the critical path cycle as an execution cycle period. The experimental results show that 16x16 bit modified dadda multiplier with AHL can achieve the power and delay is significantly reduced as compared to dadda multiplier without AHL.

This paper is structured as follows: section II will describe the preliminaries for the design of multiplier structures Section III describe proposed modified dadda multiplier follows as section IV describe about the application canny edge detection function section v describe about results and discussion and finally in section VI we concluded about conclusion.

II.PRELIMINARIES FOR THE DESIGN OF MULTIPLIER STRUCTURES

The multiplication process begins with the generation of all partial products in parallel using an array of AND gates. The next major steps in the design process are partitioning of the partial products and their reduction process. Each of these steps are elaborated in the following subsections.

PARTITIONING THE PARTIAL PRODUCTS

Consider two n-bit operands an-1an-2…a2a1a0 and bn-1bn-2…b2b1b0 for n by n Baugh-Wooley multiplier, the partial products of two n-bit numbers are aibj where i,j go from 0,1,..n-1. The partial products form a matrix of n rows and 2n-1 columns as show in Fig. 1(a). To each partial product we assign a number as shown in Fig. 1 (a), e.g. a0b0 is given an

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijareeie.com

Vol. 6, Issue 11, November 2017

index 0, a1b0 the index 1 and so on. For convenience we rearrange the partial products as shown in Fig 1(b). The longest column in the middle of the partial products contributes to the maximum delay in the PPST.

MULTIPLICATION

Compressors for multiplication are investigated. A fast (exact) multiplier is usually composed of three parts (or modules) • Partial product generation. • A Carry Save Adder (CSA) tree to reduce the partial products' matrix to an addition of only two operands • A Carry Propagation Adder (CPA) for the final computation of the binary result. In the design of a multiplier, the second module plays a pivotal role in terms of delay, power consumption and circuit complexity. Compressors have been widely used to speed up the CSA tree and decrease its power dissipation, so to achieve fast and low-power operation. The use of approximate compressors in the CSA tree of a multiplier results in an approximate multiplier. A 8×8 unsigned Dadda tree multiplier is considered to assess the impact of using the compressors in approximate multipliers. multiplier uses in the first part AND gates to generate all partial products. In the second part, the approximate compressors proposed in the previous section are utilized in the CSA tree to reduce the partial products. The last part is an exact CPA to compute the final binary result. Figure 1(a) shows the reduction circuitry of an exact multiplier for n=8. In this figure, the reduction part uses half-adders, full-adders and 4-2 compressors; each partial product bit is represented by a dot. In the first stage, 2 half adders, 2 full-adders and 8 compressors are utilized to reduce the partial products into at most four rows. In the second or final stage, 1 half-adder, 1 full-adder and 10 compressors are used to compute the two final rows of partial products. Therefore, two stages of reduction and 3 half-adders, 3 full adders and 18 compressors are needed in the reduction circuitry of an 8×8Dadda multiplier.

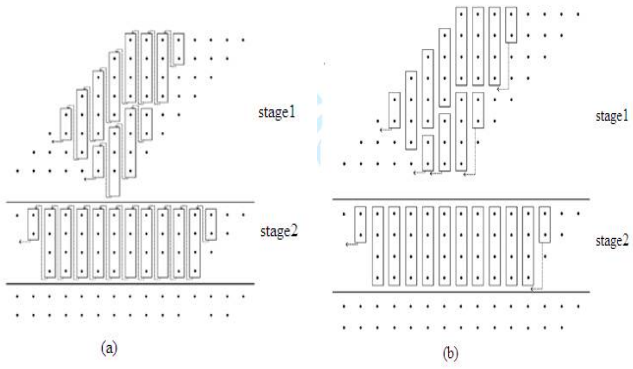


Figure 1 Reduction circuitry of an *8×8*Dadda multiplier, (a) using Design1 compressors, (b) using Design 2 compressors

• In the first case (Multiplier 1), Design 1 is used for all 4-2compressors in Figure 1(a).
• In the second case (Multiplier 2), Design 2 is used for the 4-2 compressors. Since Design 2 does not have *cin* and *cout*, the reduction circuitry of this multiplier requires a lower number of compressors Figure 1(b). Multiplier 2 uses 6 half-adders, 1 full-adder and 17 compressors.
• In the third case (Multiplier 3), Design 1 is used for the compressors in the*n-1* least significant columns. The other *n* most significant columns in the reduction circuitry use exact 4-2 compressors.

In the fourth case (Multiplier 4), Design 2 and exact 4-2compressors are used in the*n-1* least significant columnsand the*n* most significant columns in the reduction circuitry respectively.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijareeie.com

Vol. 6, Issue 11, November 2017

DESIGN STRUCTURE OF 4:2 COMPRESSOR

The 4-2 compressor which has 4 inputs (x1, x2, x3 and x4) and 2 outputs (Sum & Carry) along with a Carry-in (Cin) and a Carry-out (Cout) as shown in Fig 2. The input Cin is the output from the neighboring lower significant compressor.

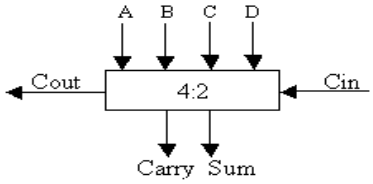


Figure.2. 4:2 Compressor

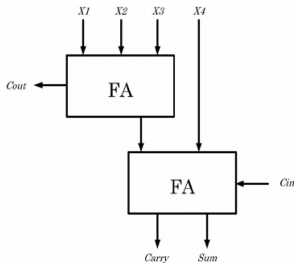


Figure.3. 4:2 Compressor using full adder.

The Cout is the output to the next significant stage compressor. It consists of two 3-2 compressors (full adders) in series and involves a critical path of 4 XOR delays. An alternative implementation is shown in Fig.3. This implementation is better and involves a critical path delay of three XOR's, hence reducing the critical path by 1 XOR delay.

III. PROPOSED MODIFIED DADDA MULTIPLIER

The proposed scheme is aging-aware reliable multiplier design as shown in Fig. It introduces the overall architecture and the functions of each component and also describes how to design AHL that adjusts the circuit when significant aging occurs.
Proposed aging-aware multiplier architecture, which includes two m-bit inputs (m is a positive number), one 2mbit output, one dadda multiplier, 2m 1-bit Razor flip-flops, and an AHL circuit encoder, decoder and mux.

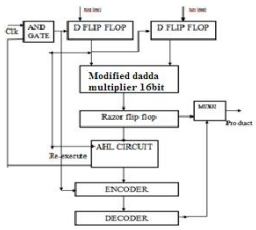


Figure.4 Modified Dadda Multiplier Using AHL Circuit

Consequently, the two maturing mindful multipliers can be executed utilizing comparable engineering, and the distinction between the multipliers lies in the info signs of the AHL. As indicated by the multiplier, the info flag of the AHL in the engineering with the mr multiplier is the md multiplicand, though of the dadda is the multiplicator. Razor flip-failures can be utilized to identify in the

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijareeie.com

Vol. 6, Issue 11, November 2017

case of timing infringement happen before the following info design arrives. A 1-bit Razor flip-tumble contains a primary flip slump, shadow hook, XOR door, and mux.

The primary flip-slump gets the execution result for the blend circuit utilizing an ordinary clock flag, and the shadow hook gets the execution result utilizing a deferred clock flag, which is slower than the typical clock flag. On the off chance that blunders happen, the Razor flip-flounder will set the mistake flag to 1 to inform the framework to reexecute the operation and advise the AHL circuit that a blunder has happened. If not, the operation is re-executed with two cycles. Despite the fact that the re-execution may appear to be exorbitant, the general cost is low the fact that the re-execution recurrence is low.

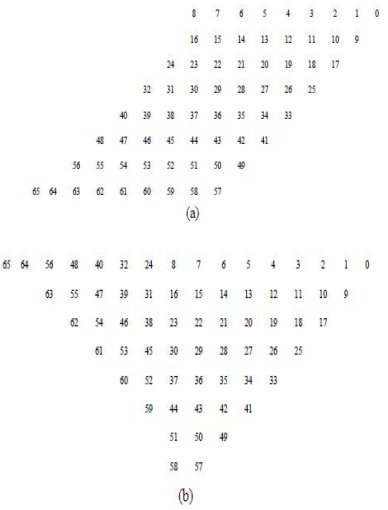
More points of interest for the Razor flip-flop can be found. The AHL circuit is the key part in the maturing product variable-dormancy multiplier. demonstrates the subtle elements of the AHL circuit. The AHL circuit contains a maturing pointer, two judging squares, one mux, and one D flip-slump. The maturing marker demonstrates whether the circuit has endured critical execution corruption because of the maturing impact. The maturing pointer is actualized in a basic counter that tallies the quantity of mistakes over a specific measure of operations and is reset to zero toward the finish of those operations. On the off chance that the cycle period is too short, the dadda multiplier can't finish these operations effectively, causing timing infringement. This planning infringement will be gotten by the Razor flip-flops.

MODIFIED 16 BIT DADDA MULTIPLIER

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).

THE DADDA BASED REDUCTION

The partial products of each part are reduced to two rows by the using (3,2) and (2,2) counters based on the regular Dadda reduction algorithm as shown in Fig. 2 and Fig. 3. The grouping of 3-bits and 2-bits indicates (3,2) and (2,2) counters respectively and the different colors classify the difference between each column, where *s* and *c* denote *partial sum* and *partial carry* respectively. E.g. the bit positions of *6* and *13* in part0 are added using a (2,2) counter to generate sum *s0* and *c0*. The *c0* is carried to the next column where it is to be added up with the sum *s1* of a (3,2)



ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijareeie.com

Vol. 6, Issue 11, November 2017

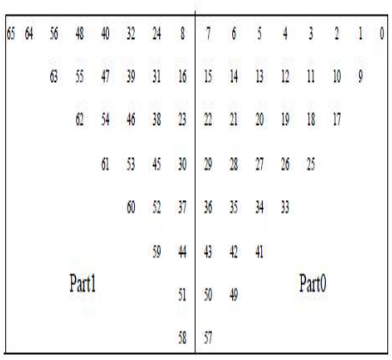


Figure.5 Partitioning the partial products:

(a) Partial product array diagram for 8*8 multiplier, (b) An Alternative Representation, (c) Partitioned structure of multiplier showing part0 and part1.

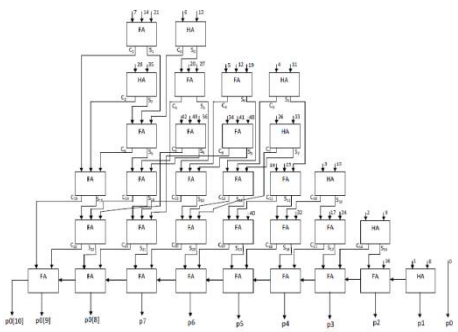
counter adding 7, 14 and 21. The carry c1 of (3,2) counter is added to the next column. The final two rows of each part are summed using a Carry Look-ahead Adder (CLA) to form the partial final products of a height of one bit column which indicated at the bottom of Fig. 3.1.

The two parallel structures for a and b based on the Dadda approach are shown in Fig. 3.3 where HA, FA, p0, p1 and p denote Half Adder ((2,2)counter)), Full Adder ((3,2)counter) , partial final product from part0, partial final product from part1 and final product respectively. The numerals residing on the HA and FA indicates the position of partial products. The output of part0 and part1 are computed independently in parallel and those values are added using a high speed hybrid final adder to get the final product.

However, before we proceed to carry out the final addition with the proposed hybrid adder, we first carry out the final addition with the CLA for both the unpartitioned Dadda multiplier and the partitioned Dadda multiplier. This enables us to evaluate and analyze the effect of partitioning the PPST into two parts.

It can be seen that for the 8-bit multiplier, there is no improvement in the speed, area and power. But with the increase in the word size, the improvement in the speed, area and power of the partitioned multipliers increases. There is a maximum of 10.5% improvement in delay for the 64-bit multiplier with only a slight increase in the area and power of 1% and 1.8% respectively.

Having clearly demonstrated the reduction in the delay of the Dadda multipliers due to the partitioning of the partial products we now proceed to further enhance the speed of the proposed multiplier. The further improvement in the performance can be achieved by replacing the CLA with the proposed hybrid final adder structure which is elaborated in the next section.Once each part of the partial products has been reduced to a height of one bit column, we get the final partial products as follows



(a)

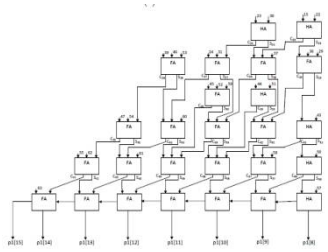
ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijareeie.com

Vol. 6, Issue 11, November 2017



(b)

Figure.6 The Dadda based implementation: (a) Implementation of part1, (b) Implementation of part2

The p0[10:8] and p1[10:8] are added using 3-bit CLA which finds p[10:8]. To obtain the remaining p[15:11], the p1[15:11] are assigned to the input of 5-bit MBEC, which produce the two partial results p1[15:11] with Cin of '0' and the 5-bit BEC output with the Cin of '1'. Depending on the Cout of CLA(c[10]), the mux provides the final p[15:11] without having to ripple the carry through p1[15:11].

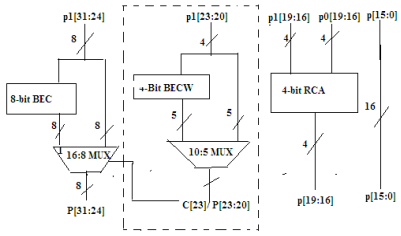


Figure.7 Modified dadda 16*16 multiplier

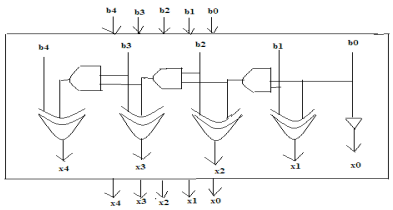


Figure.8 Binary to excess 1 code Converter (Without carry)

MBEC STRUCTURE AND ITS OPERATION

The 8-bit multiplier uses a single 5-bit MBEC in the final adder. But the large bit sized multipliers requires multiple MBEC and each of them requires the selection input from the carry output of the preceding MBEC. Therefore to generate the carry output from the MBEC, an additional block is developed which is called MBECWC (MBEC With Carry). The detailed structures of the 5-bit BEC without carry (BEC) and with carry (BECWC) are shown Fig. 3.4 and Fig. 3.5. The BEC gets *n* inputs and generates *n* output; the BECWC gets *n* input and generates *n+1* output to give the carry output as the selection input of the next stage *mux* used in the final adder design of 16, 32 and 64-bit multipliers.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijareeie.com

Vol. 6, Issue 11, November 2017

TABLE I
FUNCTION TABLE OF 5-BIT BEC & BECWC

Input	BEC without carry	BEC with carry	
b[4:0]	x[4:0]	cy	x[4:0]
00000	00001	0	00001
00001	00010	0	00010
00010	00011	0	00011
00011	00100	0	00100
00100	00101	0	00101
⋮	⋮	⋮	⋮
11011	11100	0	11100
11100	11101	0	11101
11101	11110	0	11110
11110	11111	0	11111
11111	00000	1	00000

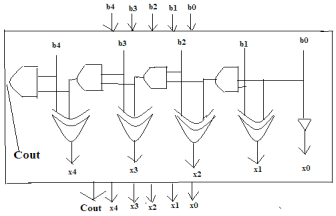


Figure.9 Binary to excess 1 code Converter
(With carry)

ADAPTIVE HOLD LOGIC ARCHITECTURE

A novel architecture of an Adaptive Hold Logic (AHL) circuit is proposed which will reduce the aging effects. The Adaptive Hold Logic (AHL) circuit can decide whether the input patterns require one or two cycles and can adjust the judging criteria to ensure that there is minimum performance degradation after considerable aging occurs.The Adaptive Hold Logic (AHL) circuit is as shown in fig. Assume the AHL circuit has a m bit input. The Adaptive Hold Logic (AHL) circuit consists of the following blocks.

A.Judging Blocks
B. Aging Indicator

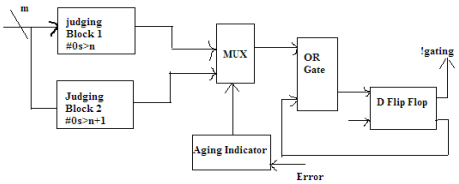


Figure.10. Adaptive Hold Logic (AHL) circuit

JUDGING BLOCKS

There are two judging obstructs in Adaptive Hold Logic (AHL) circuits. The first judging square will create a yield as 1 if the quantity of zeros in the info grouping is bigger than n. On the off chance that the quantity of zeros in the information arrangement is bigger than n+1 then the yield of the second judging square is 1. The estimation of n is characterized by the client. The operation of Adaptive Hold Logic (AHL) circuit are as per the following at the point when an info arrangement is given, both the judging pieces will choose whether the succession requires one cycle or two cycle to finish their operation and pass the two outcomes to the multiplexer.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijareeie.com

Vol. 6, Issue 11, November 2017

The !(gating) flag will move toward becoming 1 and the information flip-lemon will lock new information to perform operation in the following cycle. In the event that the yield of the multiplexer is 0 which implies the information grouping requires two cycles to finish it's operation. The OR door will yield 0 to the D flip-flop. In this manner, to incapacitated the clock flag of the information flip-flounders in the following cycle the !(gating) flag will move toward becoming 0. Just a single cycle of the information flip flounder will be handicapped in light of the fact that the D flip-flop will hook 1 in the following cycle again, when the yield of the multiplexer is 0, which implies the info design requires two cycles to finish, the OR door will yield 0 to the D flip-tumble.

AGING INDICATOR

The Aging Indicator demonstrates that whether the circuit has endured huge execution debasement because of maturing impacts. The maturing impact isn't noteworthy in the first place, so the maturing pointer produces yield as 0. The Aging Indicator is executed in a counter that checks the quantity of mistakes over a specific measure of operations. These operations might be duplication or expansion. It resets to zero toward the finish of those operations.

RAZOR FLIP-FLOP

The key idea of Razor is to purposely operate the circuit at sub-critical voltage and tune the operating voltage by monitoring the error rate. This eliminates the need for conservative voltage margins. The trade-off would be between the power penalty incurred from error correction against the additional power savings obtained from operating at a lower supply voltage.

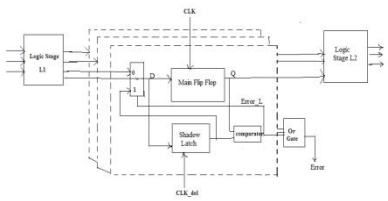


Figure.11. Main Razor flip flops.

correcting these errors is minimal, while the power savings from the reduced operating voltage can be substantial. The key idea of the Razor was to purposely operate the circuit at the sub-critical voltage and reduce the operating voltage by analyzing the error rate. This eliminates the need for conservative voltage margins.

The trade-off would be between the powers penalties incurred from error correction against the additional power attained from working at a lower supply voltage.A 1-bit Razor flip-slump contains a fundamental flip-flounder, shadow hook, XOR door, and mux. The primary flip-flounder gets the execution result for the mix circuit utilizing a typical clock flag, and the shadow hook gets the execution result utilizing a postponed clock flag, which is slower than the ordinary clock flag. the Razor flip-tumble will set the mistake flag to 1 to advise the framework to re execute the operation and inform the AHL circuit that a blunder has happened. We utilize Razor flip-flop to identify whether an operation that is thought to be a one-cycle example can truly complete in a cycle. If not, the operation is re executed with two cycles. In spite of the fact that the re execution may appear to be exorbitant, the general cost is low on the grounds that the re-execution recurrence is low. More subtle elements for the Razor flip-flounder can be found. Razor depends a mix of building and circuit-level strategies for effective mistake checking.

ENCODER ,DECODER & D-FLIP FLOP

ENCODER AND DECODER

The purpose of encoder is standardization, speed, secrecy, security, size. Encoders are combinational logic circuits and they are reverse of decoders. They accept one or more inputs and generate a multibit output code. Decoder is performance quit opposite to encoder operation.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijareeie.com

Vol. 6, Issue 11, November 2017

D-FLIPFLOP

Data flip-flop tracks the input, making transitions with match those of the input D. The D stands for "data"; this flip-flop stores the value that is on the data line. It can be thought of as a basic memory cell. A Data flip-flop can be made a set/reset flip-flop by tying the set to the reset through an inverter.

IV. CANNY EDGE DETECTION FUNCTION

Each heading of spreads is related with a photograph, by then two new pictures are made. One picture displays the vertical reaction and trade demonstrates the even reaction. Two pictures joined into a solitary picture. The clarification behind existing is to pick the proximity and scope of edges in a photo. This two picture blend is cleared up that the square of made shroud pixel survey upbeat event every special as empower are summed.

The Process of Canny edge detection algorithm can be broken down to 5 different steps:

Apply Gaussian filter to smooth the image in order to remove the noise

Find the intensity gradients of the image

Apply non-maximum suppression to get rid of spurious response to edge detection

Apply double threshold to determine potential edges

Track edge by hysteresis: Finalize the detection of edges by suppressing all the other edges that are weak and not connected to strong edges.

GAUSSIAN FILTER

Edge detection results are easily affected by image noise, it is essential to filter out the noise to prevent false detection caused by noise.

$$G_{xy} = \frac{1}{2\pi\sigma^2} \exp\left[- \frac{(x-(z+1))^2 + (y-(z+1))^2}{2sigma^2} \right] ; 1 \le x,y \le (2z+1)$$

FINDING THE INTENSITY GRADIENT OF THE IMAGE

An edge in an image may point in a variety of directions, so the Canny algorithm uses four filters to detect horizontal, vertical and diagonal edges in the blurred image. The edge detection operator (such as Roberts, Prewitt, or Sobel) returns a value for the first derivative in the horizontal direction (G_x) and the vertical direction (G_y). From this the edge gradient and direction can be determined

$$\mathbf{G} = \sqrt{\mathbf{G}_x^2 + \mathbf{G}_y^2}$$
$$\mathbf{\Theta} = \operatorname{atan2}(\mathbf{G}_y, \mathbf{G}_x),$$

NON-MAXIMUM SUPPRESSION

Non-maximum suppression is an edge thinning technique.Non-Maximum suppression is applied to "thin" the edge. After applying gradient calculation, the edge extracted from the gradient value is still quite blurred.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijareeie.com

Vol. 6, Issue 11, November 2017

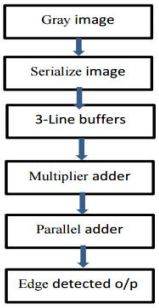
DOUBLE THRESHOLD

After application of non-maximum suppression, remaining edge pixels provide a more accurate representation of real edges in an image. However, some edge pixels remain that are caused by noise and color variation. In order to account for these spurious responses, it is essential to filter out edge pixels with a weak gradient value and preserve edge pixels with a high gradient value.

EDGE TRACKING BY HYSTERESIS

The strong edge pixels should certainly be involved in the final edge image, as they are extracted from the true edges in the image. However, there will be some debate on the weak edge pixels, as these pixels can either be extracted from the true edge, or the noise/color variations.

The outcome which is returned by this farthest point is the last picture in which Edge disclosure is a photo taking care of procedure for finding the cut-off points of things inside pictures. It works by distinguishing discontinuities. Edge area is used for picture division and data extraction in areas, for instance, picture taking care of, PC vision, and machine vision.



.

Image gradient are to extract the information from the images ,the gradient are used to detect the edge.There are white and gray pixel where use to convert the white image in to gray scale image.The white image has the large gradient values the gray scale image is has the less or small gradient values.The thresholding is used to eliminate the noise from image while retaining the important edges.This edge detection using 2d filter horizontal and vertical convolution.The RGB data is converted in to gray scale and this is multiplied with 0 degrees and 90 degrees. For the convolution using 3line buffer are nothing but the shift registers,the multiplier are coefficients of matrix and parallel adder .The parallel adder adds the results for total gradient.The image is reduced the size when it is converted to gray scale image the gray image is converted to edge detection image.Less utilization of hardware is advantage.

V.RESULTS AND DISCUSSION

LOGICAL SIMULATION RESULTS

This logical simulation result shown into the Simulink block in Matlab 2014b.The Matlab 2014b is a user friendly software. Here the figure shows the canny edge detection implementation.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijareeie.com

Vol. 6, Issue 11, November 2017

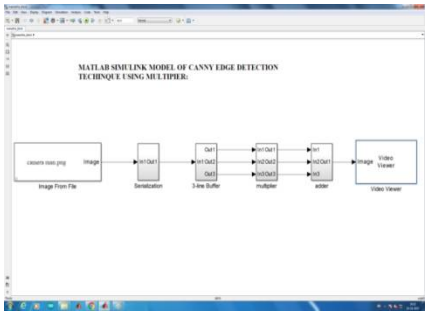


Figure.12. Matlab simulation result

(a)Original Grayimage (b) edge detected ouput

Figure.13 Simulation result for canny edge detection

The canny edge detector plays an important role in pre-processing image for many virtual reality systems. The multiplier less implementation of the canny edge detector is proposed which enables the system to rebuild at a lower cost. Canny edge detector involves a number of local operators such as image smoothing and gradient computation many methods are present but canny edge detector is so useful in image edge detections.

XILINX SYNTHESIS RESULT

MODIFIED DADDA MULTILPLIER OF 16BIT

In this project for hardware implementation verilog code was written for the multiplier architecture block the multiplier architecture is implemented in cadence tool. Many software consists in cadence tool. For this project tools were utilized was (NC- NATIVE CODE for Simulation), (RC- RTL complier for synthesis) and encounter digital implementation tool for soc (Physical layout of standard cells). Finally we obtain a chip layout of Multiplier architecture, Area and power of the chip layout. From there GDSII file was generated for fabrication of the chip.

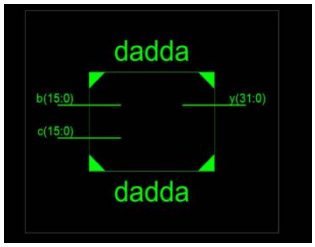


Figure.14. 16 bit dadda multiplier sub block

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijareeie.com

Vol. 6, Issue 11, November 2017

Figure.15.Rtl schematic of dadda 16bit multiplier

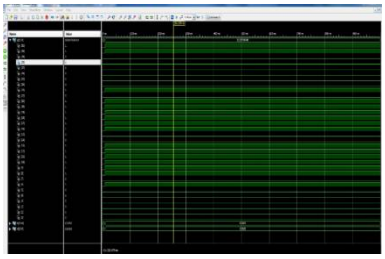


Figure.16. 16 bit dada multiplier using Xilinx

PROPOSED ARCHITECTURE FOR CANNY EDGE DETECTION SYNTHESIS OUTPUTS

In extension, we have designed a architecture by Adaptive hold logic 16bit which is the fastest and more efficient than dadda 16bit. Here we used 16bits for dadda16bit ,razor flip flop,AHL for error free data. So, by combining this we are designing an efficient AHL(adaptive hold logic).

The processor output for simulation is designed below:

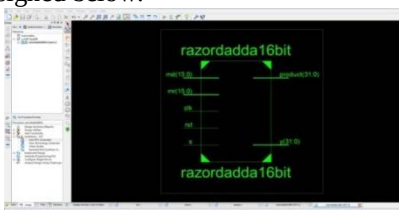


Figure.17. sub system of proposed architecture

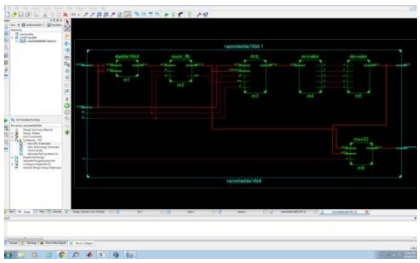


Figure.18 RTL view of proposed architecture

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijareeie.com

Vol. 6, Issue 11, November 2017

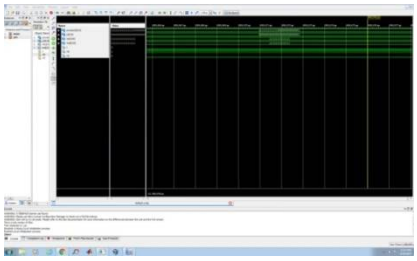


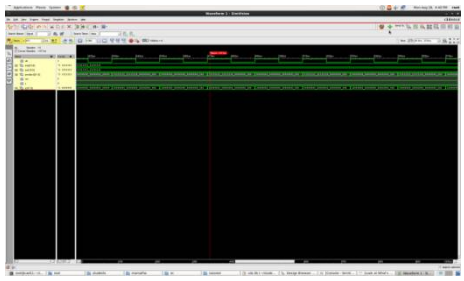
Figure.19 Output waveform of proposed architecture in Xilinx.

NC OF MODIFIED DADDA 16*16 RESULT

NC Verilog gives the propel innovation that is mostly utilized for Simulation reason, and general efficiency. It is basically used to check whether the processor performs adjust or not. In proposed paper, we composed design of various 16Bit multiplier.

The outline beneath demonstrates NC of proposed paper.

NC OF modified dadda 16*16 RESULT



RC (RTL) RESULTS (CADENCE): SCHEMATIC RESULTS

Resulting to finishing the Native code amusement, we have to check the schematic. So for that we ought to use the RTL compiler.

RC OF MODIDIED DADDA 16*16 MULTIPLIER RESULT

The Cadence Encounter gathering of things gives a fused response for a RTL-to-GDSII setup stream. The Cadence Encounter gathering of things gives a grouping of modernized responses for nanometre design. The above figure exhibits the RTL limit of the Native code generation.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijareeie.com

Vol. 6, Issue 11, November 2017

RC RESULTS

MODIDIED DADDA 16*16 MULTIPLIER RC (SYSTEM ON CHIP) IN 180NM TECHNOLOGY

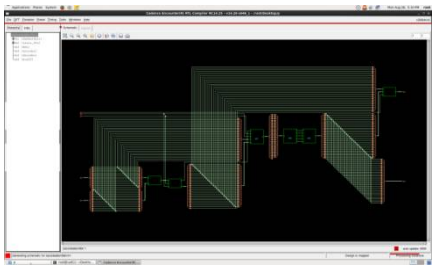


Figure.20.Main block of extension circuit

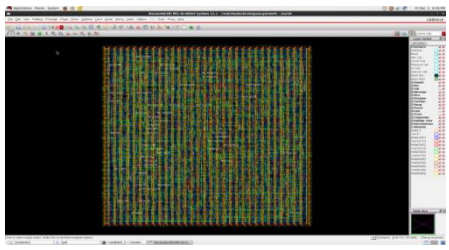
PD RESULTS

MODIFIED DADDA 16*16 PD RESULTS

MODIDIED DADDA 16*16 MULTIPLIER PD (SYSTEM ON CHIP) IN 180NM TECHNOLOGY

SOC Encounter for backend design (floor planning, place and course, power and clock distribution).Now you should have the ability to run the Cadence gadgets. We should not run Cadence from root registry, it should be continue running from the home index. Since root registry makes various extra records that will discard your root inventory. Instead of make a list (e.g. mood, you should have this starting at now made) and another registry for the arrangement (e.g. instructional exercise) now associate the library records to list

MODIFIED DADDA 16*16 MULTIPLIER PD(Physical design) OUTPUT MULTIPLIER



COMPARISON RESULTS

Sl no.	Multipliers	Area (sq microns)	Power (milli watts)	Delay (ns)
1.	AHL with 16bit column multiplier[15]	4057.9	8.6	25.2
2.	Modified dadda multiplier [proposed-1]	2550.7	8.8	23.3
3.	AHL with 16bit Modified dadda multiplier[proposed -2]	3575.2	7.8	28.2

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

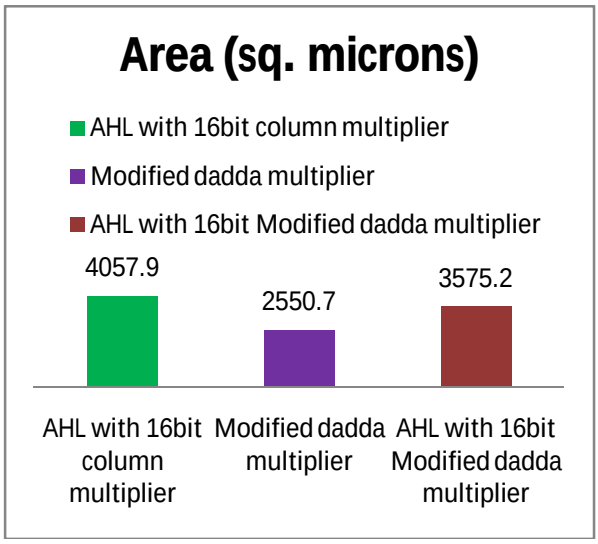
International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

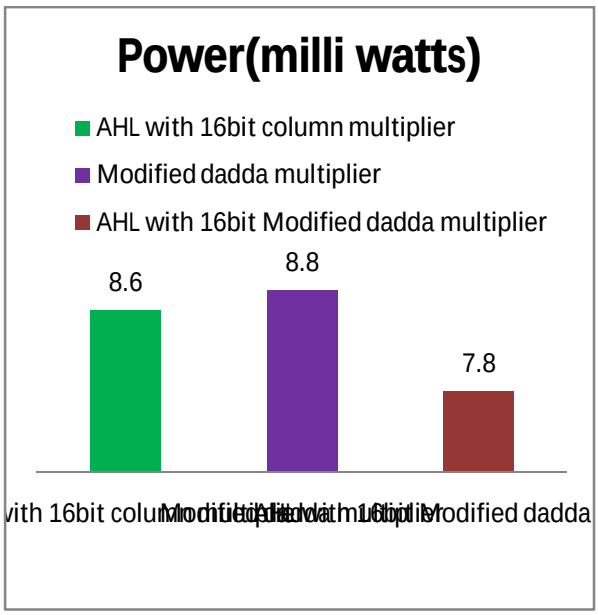
(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijareeie.com

Vol. 6, Issue 11, November 2017

From above tables, there is a decrease in area and power. it consumes less area and low Power. And the Proposed & extension is done in 180nm technology. The Implementation is done in SOC when compare
AHL with 16bit column multiplier got a better results with AHL with 16bit modified dadda multiplier. Reduction of area up to 13% ,power consumption reduction up to 10% and 12% increased in critical delay. The results shows that the proposed multiplier excel them in terms power and area.





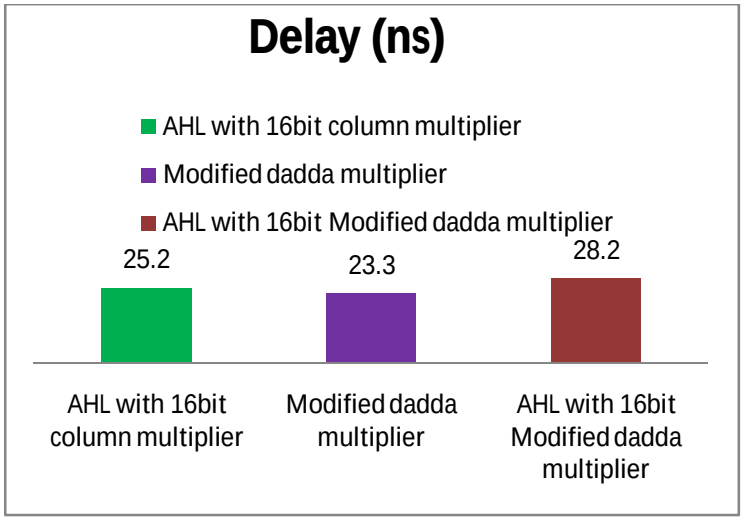
ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijareeie.com

Vol. 6, Issue 11, November 2017



VI.CONCLUSION

In modified dadda multiplier design with the AHL the multiplier is able to adjust the AHL to mitigate performance degradation due to increased delay. This proposed work is done using cadence 180nm Technology. When comparing with the previous proposed background they have implemented in an synopsis TSMC 65nm standard Library cells. Modified dadda multiplier is more efficient as compared to column and row Multiplier. Using PPST Technique dadda multiplier is Implemented so the area and power is less compared to column and row Multiplier. In Future dadda multiplier 32*32 and 64*64 Multiplier can be implement in cadence tool to get a better performance of area and power.Multilier can be used in application parts like (RS-Encoder) design in perspective of this multiplier.

REFERENCES

[1] S. M. Smith and J. M. Brady, "Susana new approach to low level image processing," *International Journal of Computer Vision*, vol. 23, no. 1, pp. 45–78, 1997.
[2] C. Harris and M. Stephens, "A combined corner and edge detection," in *Proc. the Fourth Alvey Vision Conference*, 1988, pp.147–151.
[3] T. L. Chao and K. H. Wong, "An efficient fpga implementation of the Harris corner feature detector," in *Proc. 2015 14th IAPR International Conference on Machine Vision Applications (MVA)*, 2015, pp. 89–93.
[4] A. Benedetti and P. Perona, "Real-time 2-d feature detection on a reconfigurable computer," in *Proc. the IEEE Computer SocietyConference on Computer Vision and Pattern Recognition*, 1998, p.
586.
[5] M. Krystian and C. Schmid, "An affine invariant interest point detector," *Computer Vision—ECCV 2002*, Springer Berlin Heidelberg, pp. 128-142, 2002.
[6] J. Shi and C. Tomasi, "Good features to track," in *Proc. 1994 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1994, pp. 593–600.
[7] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," pp. 674–679, 1981.
[8] C. Tomasi and T. Kanade, "Detection and tracking of point features," *International Journal of Computer Vision, Tech. Rep.*, 1991.
[9] D. Honegger, H. Oleynikova, and M. Pollefeys, "Real-time and low latency embedded computer vision hardware based on a combination of fpga and mobile cpu," in *Proc. 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 4930–4935.
[10] Q. Xu, S. Varadarajan, C. Chakrabarti, and L. J. Karam, "A distributed canny edge detector: algorithm and fpga implementation," *IEEE Transactions on Image Processing*, vol. 23, no. 7, pp. 2944–2960, 2014.
[11] P. R. Possa, S. A. Mahmoudi, N. Harb, C. Valderrama, and P.
Manneback, "A multi-resolution fpga-based architecture for real-time edge and corner detection," *IEEE Transactions on Computers*, vol. 63, no. 10, pp. 2376–2388, 2014.
[12] M. Leeser, S. Miller, and H. Yu, "Smart camera based on reconfigurable hardware enables diverse real-time applications," in *Proc. the 12th Annual IEEE Symposium on Field-Programmable International Journal of Computer Theory and Engineering, Vol. 9, No. 3, June 2017* 177 *Custom Computing Machines*, Washington, DC, USA, 2004, pp. 147–155.
[13] B. Tippetts, S. Fowers, K. Lillywhite, D.-J. Lee, and J. Archibald, "Fpga implementation of a feature detection and tracking algorithm for realtime applications," in *Proc. the 3rd International Conferenceon Advances in Visual Computing*, Volume Part I, , 2007, pp. 682–691.
[14] "Aging-aware reliable multiplier design with AHL," *IEEE Trans. Very Large Scale Integr.(VLSI).*

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijareeie.com

Vol. 6, Issue 11, November 2017

[15] K.-C. Wu and D. Marculescu, "Aging-aware timing analysis and optimization considering path sensitization," in *Proc. DATE*, 2011, pp. 1-6.
[16] Y.-S. Su, D.-C. Wang, S.-C. Chang and M. Marek-Sadowska, "Performance optimization using variable-latency design style," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 19, no. 10, pp.1874-1883, Oct.2011.
[17] K. Du, P. Varman, and K. Mohanram, "High performance reliable variable latency carry select addition," in *Proc. DATE*, 2012, pp. 1257-1262.
[18] D. Baneres, J. Cortadella and M. Kishinevsky, "Variable –latency design by function speculation," in *Proc. DATE*, 2009, pp. 1704-1709.
[19] Y. Chen et al., "Variable-latency adder (VL-Adder) designs for low power and NBTI tolerance,"
IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 18, no.11, pp.16211624, Nov.2010.
[20] Georgios Zervakis:Design Efficient Approximate multiplication circuits through partial product perforation"

BIOGRAPHY

Ms. S.MAMATHA has completed B.Tech in ECE Department from LIET, Himayathsagar, Hyderabad. Presently she is pursuing her Masters in VLSI System Design in Sridevi Women‟s Engineering College, Vattina-gulapally, Gandipet, Hyderabad, India.