International Journal of Research
Available at https://edupediapublications.org/journals
p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 03 Issue 13
September 2016

High Accuracy Fixed Width Booth Multiplier Base on Multi Level Conditional Probability

*K. Rajashekar &**G. Santhosha

*M.Tech Dept of ECE, Vaagdevi Engineering College

**Asst.Prof Dept of Ece, Vaagdevi Engineering College

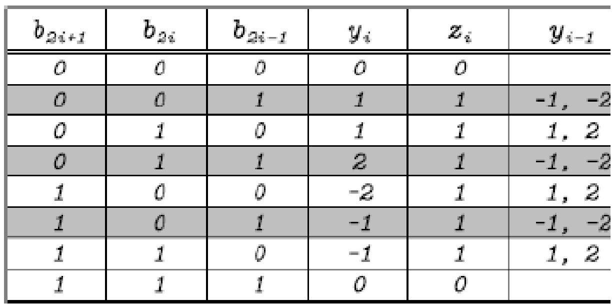
Abstract:

This brief proposes an accuracy-adjustment fixed-width Booth multiplier that compensates the truncation error using a multilevel conditional probability (MLCP) estimator and derives a closed form for various bit widths L and column information w. Compared with the exhaustive simulations strategy, the proposed MLCP estimator substantially reduces simulation time and easily adjusts accuracy based on mathematical derivations. Unlike previous conditional-probability methods, the proposed MLCP uses entire nonzero code, namely MLCP, to estimate the truncation error and achieve higher accuracy levels. Furthermore, the simple and small MLCP compensated circuit is proposed in this brief. The results of this brief show that the proposed MLCP Booth multipliers achieve low-cost high-accuracy performance. Hough transform is widely used for detecting straight lines in an image, but it involves huge computations. For embedded application, field-programmable gate arrays are one of the most used hardware accelerators to achieve real-time implementation of Hough transform. In this paper, we present a resource-efficient architecture and implementation of Hough transform on an FPGA. The incrementing property of Hough transform is described and used to reduce the resource requirement. In order to facilitate parallelism, we divide the image into blocks and apply the incrementing property to pixels within a block and between blocks. Moreover, the locality of Hough transform is analyzed to reduce the memory access.

IndexTerms: *Fixed-width Booth multiplier, multilevelconditional probability (MLCP), truncation error.*

INTRODUCTION

Fixed-width multipliers are widely used in digital signal processing (DSP) applications, such as fast Fourier transform and discrete cosine transform. To generate an output with the same width as the input, fixed-width multipliers truncate the half least significant bits (LSBs) in DSP applications. Thus, truncation errors can occur in fixed-width multiplier designs. The fixed-width multiplier with highest accuracy is called a post truncated (P-T) multiplier, which truncates half of the LSBs results after calculating all products. However, a P-T multiplier requires a large circuit area to calculate truncation part products. By contrast, a direct-truncated (D-T) multiplier truncates half of the LSBs products directly to conserve circuit area, but produces a large truncation error.



b_{2i+1}	b_{2i}	b_{2i-1}	y_i	z_i	y_{i-1}
0	0	0	0	0	
0	0	1	1	1	-1, -2
0	1	0	1	1	1, 2
0	1	1	2	1	-1, -2
1	0	0	-2	1	1, 2
1	0	1	-1	1	-1, -2
1	1	0	-1	1	1, 2
1	1	1	0	0	

Table 1 Mapped Table Of A Modified Booth Encoder

This brief proposes an accuracy-adjustment fixed-width Booth multiplier that uses the multilevel conditional probability (MLCP) method to implement the compensated circuit. The MLCP method produces a closed form with various bit widths L and column information w; thus, the compensated circuit can be established quickly, and the accuracy can be adjusted by changing w. In contrast to the conditional-probability method for ACPE, which uses single nonzero code to estimate truncation errors, the proposed MLCP generates estimates by employing all nonzero code, which demonstrates high levels of intercorrelation. Although MLCP method has higher complexity to estimate truncation errors when compared with ACPE one, the accuracy of MLCP method is higher than that of ACPE method. Furthermore, simple and small compensated circuits are proposed from a single compensated closed form. According to the tradeoff between accuracy and circuit area, the MLCP method provides a balance between accuracy and circuit area. The implementation results of this brief show that the proposed MLCP Booth multiplier achieves low-cost high-accuracy performance. Hough Transform (HT) is a well-known technique for efficient shape

p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 03 Issue 13
September 2016

recognition. High computational complexity and excessive memory requirement are the major obstacles for monolithic integration of HT(3). Memory requirement problem may be simplified by current level of memory integration technique. In this paper we restrict ourselves to speed up the computational time of transformation part of the HT i.e., the computation of vote address in the parameter space. Different architectures and algorithms have been proposed to speed up the computational time for HT. Most of the Hough – based methods encounter the evaluation problem of implicit trigonometric and transcendental functions. This makes the monolithic implementation of the entire algorithm rather difficult. The pixels with the pixel value "1" are called feature points. A line is the one that passes through many features points. Imagining we draw lines with various angles passing through a feature point, each line can be represented as a point in the – space, where is the perpendicular distance of the line to the origin and is the angle between a normal to the line and the positive -axis. Each time when a line is drawn for a feature point, it will produce a (ρ,θ) value which can be considered as a "vote" for the specific (ρ,θ) value. After processing all of the feature points, the (ρ,θ) value that has the largest accumulated votes will correspond to the line that passes through the largest number of feature points. In the implementation, the votes for a specific (ρ,θ) value can be stored in a memory addressed by the specific (ρ,θ) value. The Hough transform is robust and performs well even in the presence of noise or missing data, but also involves huge computations and excessive memory requirements. Through Hough transform, the ρ for a line with an angle θ , passing through a feature point at the image coordinate (x,y) can be calculated by

$$\rho_\theta(x, y) = x\cos\theta + y\sin\theta.$$

Practical implementations of Hough transform generally involve a voting procedure over the discrete parameter space.

Algorithm 1. Voting Process of Hough Transform

```
Initialize Votes as zeros
for all feature points $(x, y)$
    for $0^\circ \le \theta < 180^\circ$, using $180^\circ/K$ as the step-size
        $\rho = x\cos\theta + y\sin\theta$
        $\rho' = \mathrm{round}(\rho)$
        $\mathrm{Votes}(\rho', \theta) = \mathrm{Votes}(\rho', \theta) + 1$
    end for
```

After the voting process, the (ρ,θ) with local-maximum values of votes are considered as candidate lines. In this paper, we only focus on the voting process of the Hough transform. Given a CIF (352 288) video with 30 frames/second (fps) and 10% feature points, it needs 109M multiplications per second to compute the values of 180 angles. For embedded applications, it requires hardware accelerators to achieve real-time Hough transform. Compared with application-specific integrated circuits (ASICs), field-programmable gate arrays (FPGAs) usually target smaller markets and require much less development time. In FPGA, high throughput is often achieved by exploiting the parallelism of the design rather than by operating the chip at a very high clock frequency. In addition, a better architecture should have more efficient utilization of the function blocks in the FPGA. In this paper, we propose an architecture and the implementation of Hough transform on an FPGA by exploiting both angle-level and pixel-level parallelism. The goal is to achieve the highest throughput with the minimum hardware resource

OBJECTIVES

The objective of the proposed work is to introduceFGPA architecture for the Hough transform and Parallelism between the blocks of processing. The incrementing property of Hough transform is described and used toreduce the resource requirement

NEED FOR THE STUDY

Modified Booth encoding is commonly used in multiplier designs to reduce the number of partial products. The 2L-bit product P can be expressed in two's complement representation as follows:

$$A = -a_{L-1}2^{L-1} + \sum_{i=0}^{L-2} a_i \cdot 2^i$$
$$B = -b_{L-1}2^{L-1} + \sum_{i=0}^{L-2} b_i \cdot 2^i$$
$$P = A \times B.$$

Run length encoding (RLE) is a simple technique to compress digital data by representing successive runs of the same value in the data as the value followed by the count, rather than the original run of values.

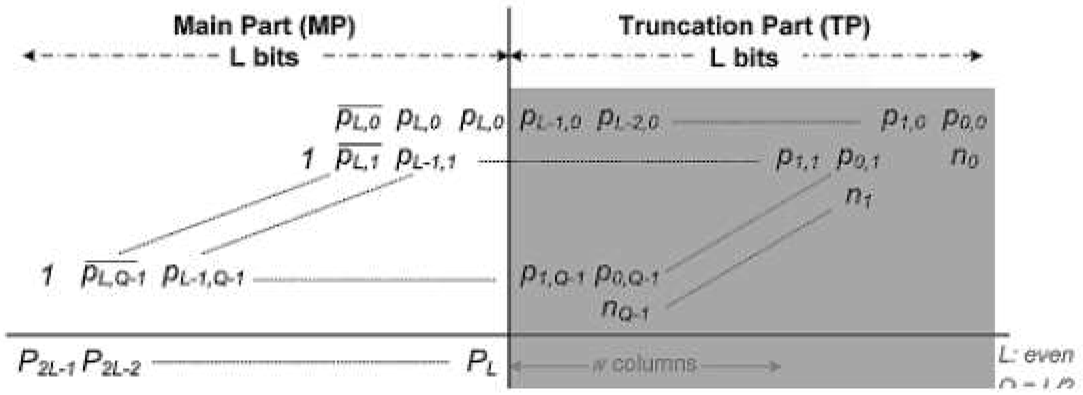

Main Part (MP)
L bits
Truncation Part (TP)
L bits
$\overline{p_{L,0}}$ $p_{L,0}$ $p_{L,0}$ $p_{L-1,0}$ $p_{L-2,0}$ $p_{1,0}$ $p_{0,0}$
1 $\overline{p_{L,1}}$ $p_{L-1,1}$ $p_{1,1}$ $p_{0,1}$
n_0
n_1
1 $\overline{p_{L,Q-1}}$ $p_{L-1,Q-1}$ $p_{1,Q-1}$ $p_{0,Q-1}$
n_{Q-1}
P_{2L-1} P_{2L-2} P_L
w columns
L: even
Q = L/2

Fig 1 Bit Encoding

The goal is to reduce the amount of data needed to be stored or transmitted. The need to use run length encoding often arises in various applications in DSP, especially image processing and compression.

Example of RLE:

Original

1	1	1	0	0	1	1	1

RLE Representation

1	3	0	2	1	3

METHODOLOGY

Booth Multiplier

To achieve a balanced design between accuracy (P-T) and area cost (D-T), several researchers have presented various error-compensated circuits to alleviate the truncation errors in Baugh–Wooley (BW) multipliers and Booth multipliers. Because a few products are truncated after Booth encoding, the multipliers have a smaller truncation error than that of BW multipliers. Therefore, many previous works have focused on the compensated circuit in Booth multipliers. Their compensated circuit consumes a large circuit area because of the complex curve fitting required for statistical analysis. use more product information to improve accuracy, but their exhaustive simulation required a considerable amount of established time. To reduce the established time for compensated circuits, Li et al. present probability estimator (PEB) that substantially reduces calculation time. An adaptive conditional-probability estimator (ACPE) is presented to improve the accuracy using conditional probability to further induce the column information w for adjusting the accuracy when applied to types of DSP systems. Therefore, two types of compensated circuits for various w are introduced, and the generalized form of PEB is presented. In sum, the established time for compensated circuits and adjustment are critical to fixed width Booth multipliers

Hough transform

The Hough transform offers several advantages. First, each image point is treated independently and therefore parallel processing is possible and the method is suitable for real-time applications. Second, the method is robust to the presence of noise, since noisy image points are very unlikely to contribute to a peak in the parameter space. Third, since each image point contributes independently to the parameter space points, the algorithm is able to work even if the shape is occluded. A final advantage is that Hough transform is able to detect different instances of the desired shape at the same time, depending on the number of peaks which are considered in the parameter space. The main drawbacks of the Hough transform are its large storage and computational requirements.

Standard Hough Transform

The Straight Line Hough transform (SLHT) was the first, and probably the most, used of the parameter-based transformations. The SLHT idea has been introduced in the previous section starting from Equation (1). However, a problem arises when lines have large slopes, i.e. . Duda and Hart [3] solved the problem of an unbounded parameter space suggesting that straight lines might be more usefully parameterized by the use of polar coordinates, i.e. the length r and the orientation θ of the normal vector to the line from the image origin. Thus, a straight line is represented by the following equation:

$$r = x\cos\theta + y\sin\theta$$

In the conventional implementation, The Hough transform essentially consists of three stages:

1. **Characteristic point detection**. Not all the image points are mapped to the parameter space. An information reduction is performed in the image such that it preserves the shapes which are wanted to be detected. Thus, some pixels of the image are selected according to certain local properties (e.g. gradientmagnitude and gradient orientation). Usually this preprocessing step is a local edge detector.

2. **Transform mapping**. Each characteristic point of the image space is mapped to the parameter space. This parameter space is represented by a two-dimensional accumulator array (n-dimensional for detection of higher-order shapes). A voting rule usually underlies the transform mapping. This voting rule determines how the transform mapping affects the contents of the accumulator array. The simplest voting rule is to increment the parameter points (θ, r) mapped from an image point (x, y).

3. **Peak detection**. To extract the corresponding parameter values of the detected shape from the

accumulator array. The simplest method consists in performing a global thresholding on the accumulator array. Since the presence of noise and distortion may result in true peaks split between several accumulator cells and, thus, not detected, different kinds of clustering procedures are also applied.

THEORIES AND ANALYSIS

Each image point (x, y) is mapped using the Hough transform to the curve Equation (2) in the

r plane. Both r and θ axes have to be quantized and hence a two-dimensional accumulator array must be constructed in the r plane. Equation (2) is applied to each point in the image and the contents of all the cells in the transform plane that the corresponding curve passes through are incremented. This equation represents the line in the r plane that has a distance $r$to the origin and the normal to which makes an angle θ with the x-axis (Fig 1). Therefore, all points in the r plane located on the line r x cos$y$sin are mapped to curves (1) in the r plane that all pass through the point

0 0 *r* , . The Hough transform method described above is known in literature as the Standard Hough Transform (SHT) [1]. Using of direct implementation of the SHT in PLD leads to slow and large multipliers and look-up table utilization. To solve this problem, we utilize the incremental Hough Transform (IHT) [4], which eliminates necessity of trigonometric operations and is based on the following ideas.

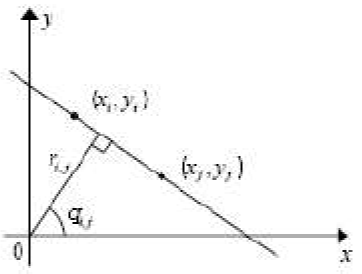


Fig 2 Relationship between (x, y) and (r, θ)

Incremental Hough Transform

The equation $r_n = x\cos\theta_n + y\sin\theta_n$ can be written as:

$$r_n = x\cos(n\varepsilon) + y\sin(n\varepsilon) \quad (3)$$

where $\varepsilon = \pi/K$ and n and K are the index and the number of the divisions of the θ-axis of the parameter space, respectively. From (3) follows:

$$\begin{aligned} r_{n+1} &= x\cos[(n+1)\varepsilon] + y\sin[(n+1)\varepsilon] \\ r_{n+1} &= x[\cos(n\varepsilon)\cos\varepsilon - \sin(n\varepsilon)\sin\varepsilon] + y[\sin(n\varepsilon)\cos\varepsilon - \cos(n\varepsilon)\sin\varepsilon] \end{aligned} \quad (4)$$

For small values of ε we can assume that $\cos\varepsilon = 1$ and $\sin\varepsilon = \varepsilon$. Then:

$$r_{n+1} = x\cos(n\varepsilon) + y\sin(n\varepsilon) + \varepsilon[y\cos(n\varepsilon) - x\sin(n\varepsilon)] \quad (5)$$

If $r'_n = x\cos(n\varepsilon) - y\sin(n\varepsilon)$, we can write:

$$\begin{aligned} r_{n+1} &= r_n + \varepsilon r'_n \\ r'_{n+1} &= r'_n - \varepsilon r_n \end{aligned} \quad (6)$$

Note that

$$\begin{aligned} r'_n &= y\cos(n\varepsilon) - x\sin(n\varepsilon) \\ r'_n &= x\cos[(\pi/2) + n\varepsilon] + y\sin[(\pi/2) + n\varepsilon] = r_{(K/2)+n} \end{aligned} \quad (7)$$

Thus, the IHT2 is defined by the equations:

$$\begin{aligned} r_{n+1} &= r_n + \varepsilon r_{(K/2)+n} \\ r_{(K/2)+n+1} &= r_{(K/2)+n} - \varepsilon r_n \end{aligned} \quad \text{for } 0 \le n < K/2 \quad (8)$$

with $r_0 = x$ and $r_{K/2} = y$ [4].

IMPLEMENTATION OF HOUGH CORDIC PROCESSOR

We implement the Hough transform using the Cordic Algorithm which allows trigonometric angles to be calculated primarily by shifting and adding. This method is very effective because if avoids the multiplication term, so we don't require the multiplier to be used. In 1957 Jack E. Volder [2] described the Coordinate Rotation Digital Computer or CORDIC for calculation of trigonometric functions, multiplication, division and conversion between binary and mixed radix number system. The CORDIC- algorithm provides an iterative method of performing vector rotations by arbitrary angles using only shift and adds. Volder's algorithm is derived from the general equation for vector rotation. If a vector V with components (xi, yi) is to be rotated through an angle (a new vector V' with components (xi',yi') is formed by equations:

$$x' = \cos(\phi)(x - y * \tan(\phi))$$

$$x' = \cos(\phi)(y + x * \tan(\phi))$$

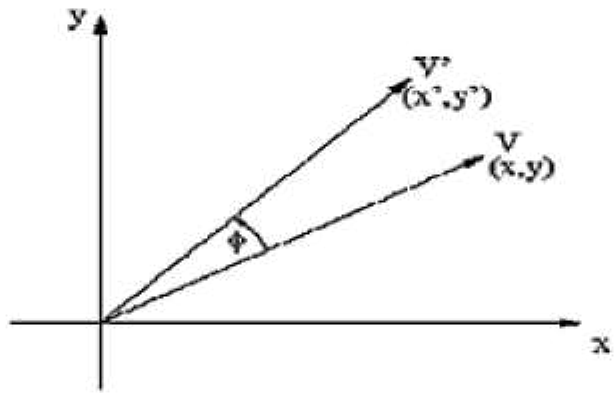


Fig 3 Rotation of a vector V by the angle

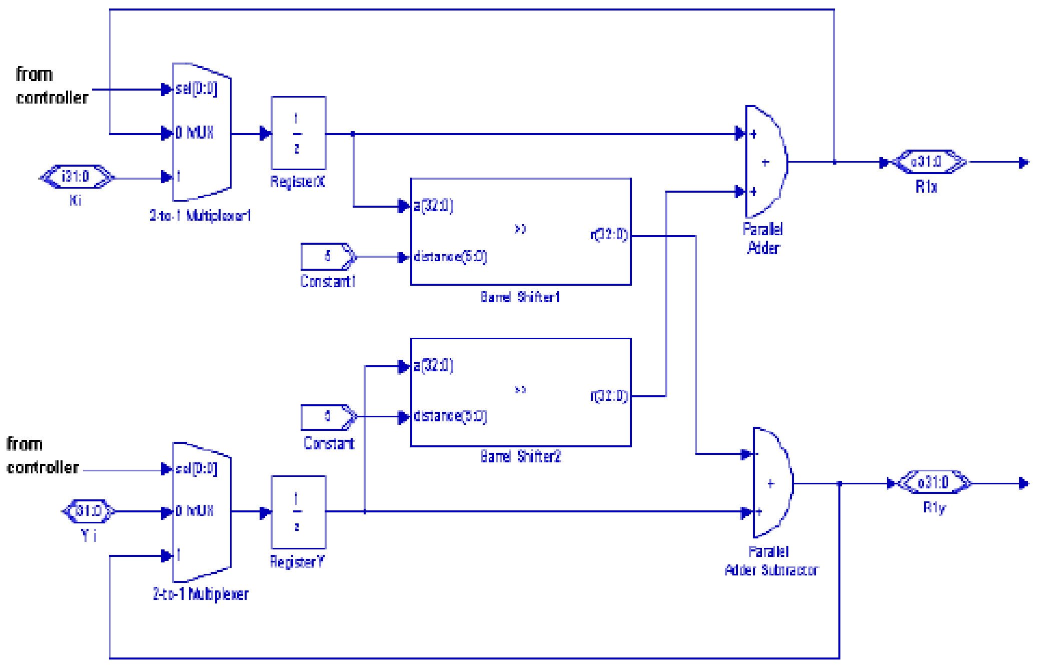


Fig 4 Hough CORDIC Block Diagram

p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 03 Issue 13
September 2016

The above schematic shows the bit parallel iterative structure of the Hough-Cordic algorithm. Other architecture that is viable is the bit parallel unrolled HT algorithm and the bit serial unrolled HT algorithm. In our design, we first model the components in VHDL and verify that these components work by simulating with test benches. An adder/subtractor (A/S), depending on a selection input, performs an addition or a subtraction. This input indicates whether an operand is negative. The basic cell of A/S is decomposed by two functions with 4 bits input each. One of them is for calculating the output and another to transmit the carry. According to this an N−bit A/S can fit in (2N+1)/2 CLBs (configurable logic block). The additional half CLB is required for introducing the least significant bit (LSB) one in case of the substraction. The critical path here is indicated by the ripple carry propagation and the routing delay of the A/S wire. This net has a fan−out of 2N in this case. It decreases the performance of the circuit and it is the main disadvantage of conventional CORDIC implementations. As the solution to this, redundant arithmetic could be used to increase the speed of the CORDIC. Implementation avoids the carry propagation from the LSB to the most significant bit (MSB), due to its carry−free property.

SURVEYS

Because the general Hough transform is very computationally intensive, other line-detection schemes have been proposed, such as gradient-based Hough transform and kernel-based transform.

Gradient-based Hough transform and kernel-based transform. These schemes require fewer computations than Hough transform, but they still require a high-end CPU, which is often unavailable in practical applications, or special hardware devices to achieve real-time performance. The gradient-based Hough transform has been implemented in special hardware, but kernel-based Hough transform, which adopts a link list data structure, is difficult to implement efficiently in hardware.

Special Hardware papers

There has been some research to implement the Hough transform on special hardware, such as graphic processors, scan line array processors, and pyramid multiprocessors. However, these devices are unsuitable for low-cost embedded systems.

CORDIC based Hough transform

This paper focuses on the implementation of the general Hough transform using FPGA. One straightforward method to implement Hough transform is using multipliers. However, **multipliers are less available on low-end FPGAs**. Hence, some researchers implement Hough transform using a coordinate rotation digital computer (CORDIC) or a simplified CORDIC algorithm such as the multisector algorithm. CORDIC is an arithmetic technique developed by Volder to solve trigonometric problems by rotating a vector in small angles until the desired angle is achieved.

The CORDIC algorithms for FPGA are surveyed. CORDIC could implement the Hough transform using only shifters and adders rather than multipliers. The major disadvantage is that it requires multiple iterationsto obtain one inthe parameter space. Hence, pipelined CORDIC implementations are proposed to improve the throughput; however, the required resources are also increased. Another drawback of CORDIC is that the result produced is not the correct value but the correct value with a constant "gain." The gain can be eliminated by applying an inverse gain to the initialization vector. However, this also requires additional

SIMULATION RESULT

TOP MODULE

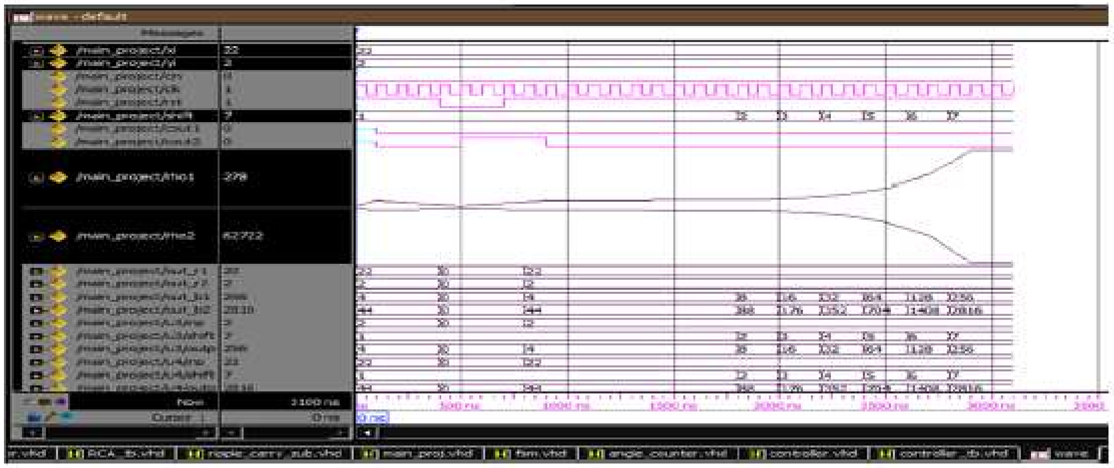


Fig 4 Top Module Output

CONCLUSIONS AND FUTURE WORKS

This brief presents a closed MLCP formula that includes column information w to adjust accuracy depending on system requirements. This formula is derived without performing time-consuming and exhaustive simulations, and can be applied to lengthy Booth multipliers to achieve high-accuracy performance. Therefore, the proposed MLCP compensated circuit can be used to develop a high-accuracy, low-cost, and flexible fixed-width Booth multiplier. In this paper, we propose a resource efficient architecture for calculating Hough transform. The incrementing property for both inter-block an intra-block incrementing are exploited to reduce the resource requirement. We use two accumulators to facilitate the inter-block incrementing, and zero-blocks are skipped by introducing a run-length coding scheme and a steptable. The intra-block incrementing could efficiently reduce the resource requirement. Instead of computing the of every pixel in a block, vote-offset is more efficient to determine the corresponding votes. We observe that pixels which are in the same block may generate identical votes in the parameter space. The locality of a block is analyzed and the votes corresponding to an identical are consolidated in order to reduce the memory access and fully utilize the FPGA memory bandwidth. The result shows that the proposed PE could achieve the best throughput with the same amount of resources compared to previously reported architectures.

REFERENCES

1. Zhong-Ho Chen, Alvin W.Y. Su and Ming-Ting Sun," Resource Efficient FPGA Architecture and Implementation of Hough Transform" in Proc. 8th Aug 2012.

2. Duda R.O and P. E. Hart, "Use of the Hough transformation to detect lines and curves in pictures," Commun. ACM, vol. 15, pp. 11–15, 1972.

3. Fernandes.L.F and M. M. Oliveira, "Real-time line detection through an improved Hough transform voting scheme," Pattern Recognit., vol. 41, no. 1, pp. 299–314, 2008.

4. Lin.L and V. K. Jain, "Parallel architectures for computing the Hough transform and CT image reconstruction," in Proc. Int. Conf. Applic. Specific Array Processors, 1994, pp. 152–163.

5. O'Gorman and M. B. Clowes, "Finding picture edges through collinearity of feature

points," IEEE Trans. Comput., vol. C-100, pp. 449–456, 1976.

6. K. K. Parhi, VLSI Digital Signal Processing Systems: Design and Implementation. New York, NY, USA: Wiley, 1999.

7. S. N. Tang, J. W. Tsai, and T. Y. Chang, "A 2.4-Gs/s FFT processor for OFDM-based WPAN applications," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 57, no. 6, pp. 451–455, Jun. 2010.

8. S. C. Hsia and S. H. Wang, "Shift-register-based data transposition for cost-effective discrete cosine transform," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 15, no. 6, pp. 725–728, Jun. 2007.

9. Y. H. Chen, T. Y. Chang, and C. Y. Li, "High throughput DA-based DCTwith high accuracy error-compensated adder tree," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 19, no. 4, pp. 709–714, Apr. 2011.

10. L. D. Van and C. C. Yang, "Generalized low-error area-efficient fixedwidth multipliers," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 52, no. 8, pp. 1608–1619, Aug. 2005.

AUTHOR 1:-

* K.RAJASHEKAR completed her B-tech in KAMALA INSTITUTE OF TECHNOLOGY & SCIENCE in 2014 and completed M-Tech in VAAGDEVI ENGINEERING COLLEGE

AUTHOR 2:-

G.SANTHOSHA is working as Assistant. professor in Dept of EEE VAAGDEVI ENGINEERING COLLEGE