

A VLSI Array Architecture for Hough Transform

K. Maharatna*

Systems Design Dept.

Institute for Semiconductor Physics (IHP)

Technology Park 25, D-15236, Frankfurt (Oder), Germany

email: maharatna@ihp-ffo.de

Swapna Banerjee

Dept. of E & ECE

Indian Institute of Technology

Kharagpur – 721302 (INDIA)

email: swapna@ece.iitkgp.ernet.in

(* Author for correspondence)

Abstract:

In this article, an asynchronous array architecture for straight line Hough Transform (HT) is proposed using a scaling free modified CORDIC (Co-Ordinate Rotation Digital Computer) unit as a basic Processing Element (PE). It exhibits four-fold angle parallelism by dividing the Hough space into four subspaces to reduce the computation burden to 25% of the conventional requirements. A distributed accumulator arrangement scheme is adopted to ensure conflict free voting operation. The architecture is then extended to compute circular and elliptic HT given their centers and orientations. Compared to some other existing architectures, this one exhibits higher computation speed.

Keywords: Hough transform, CORDIC, Low power, Image processing, Multiplierless array architecture.

1. Introduction:

Hough Transform (HT) is a well-known technique for efficient shape recognition^(1, 2). 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⁽⁴⁾. 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^(4, 5, 6, 7, 8, 9). 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. To overcome this problem, CORDIC based architectures^(3, 10). Are used to generate the vote address in parameter space.

The motivation of this work is to construct the HT architectures suitable for VLSI implementation, which can exhibit high throughput rate at reduced computational complexity. For this purpose CORDIC based asynchronous array architectures have been proposed. The total PE and angle scan range requirements are reduced by adopting an angle parallelization scheme. To overcome the scaling problem inherent to the conventional CORDIC unit, a scaling free modified CORDIC unit⁽¹¹⁾ which can be implemented using crosscoupled bus connections and adders. A high throughput asynchronous array architecture for straight line HT is proposed. Then the proposed architecture has been extended and modified to compute circular and elliptic HT. While computing circular and elliptic HT, we focus only on the estimation of the radius (for

circle), semi major and semi minor radii (for ellipse) as these parameter estimation requires exhaustive arithmetic operations like multiplication, square root evaluation, division, addition / subtraction and squaring⁽¹²⁾. To reduce the computation and hardware requirements for the estimation of these parameters, the problems are reformulated in terms of the CORDIC rotation.

The paper has been structured as follows, in Section 2, a brief description of the scaling free modified CORDIC unit is provided. The design of the CORDIC unit is carried out using Transmission Gate Logic (TGL), which shows 62 mW power consumption for 1.6 μm sea of gates technology, that has been described in this Section. In Section 3, theoretical formulation of the straight line HT using an angle parallelization scheme and the corresponding architecture are described. Comparison of this architecture with some other existing architectures is done in Section 4. In Section 5, theoretical formulation for circular and elliptic HT and the corresponding architectures are described. Conclusions are drawn in Section 6.

2. The CORDIC unit:

2.1 Brief description of modified CORDIC unit:

The CORDIC algorithm, first proposed by Volder⁽¹³⁾ and unified by Walther⁽¹⁴⁾, is an iterative procedure to compute magnitude and phase or the rotation of a vector in circular, linear and hyperbolic co-ordinate systems, described by the parameter m shown in Table 1.

An initial vector $[x \ y]^T$ undergoing a rotation through an angle ψ , will generate the final vector $[x' \ y']^T$ according to the following relation,

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \psi & \sin \psi \\ -\sin \psi & \cos \psi \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (1)$$

The total rotation ψ can be expressed in the steps of smaller angles α_i s, such that

$$\psi = \sum_{i=1}^M \alpha_i \quad (2)$$

where M is an integer.

Equation (1) can be computed by cascading a number of elementary rotational stages as follows:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \prod_{i=1}^M \begin{bmatrix} \cos \alpha_i & \sin \alpha_i \\ -\sin \alpha_i & \cos \alpha_i \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (3)$$

If the elementary angles α_i are small enough such that $\sin \alpha_i \cong \alpha_i = 2^{-i}$ and $\cos \alpha_i = 1 - 2^{-(2i+1)}$, equation 3 may be written as⁽¹¹⁾

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \prod_{i=1}^M \begin{bmatrix} 1 - 2^{-(2i+1)} & 2^{-i} \\ -2^{-i} & 1 - 2^{-(2i+1)} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (4)$$

The largest term that we are neglecting in the process of such approximation is

$$\alpha_i^3/3! = 2^{-3i}/6 = 2^{-(3i+2.585)}$$

If the machine in which the operations are supposed to be implemented has got an accuracy of b -bits, then multiplying any quantity with $\alpha_i^3/3!$ will have no effect if $(3i+2.585)$ equals or exceeds b , that is,

$$3i+2.585 \geq b \text{ or } i \geq 1/3 (b-2.585)$$

Since i can adopt only integer values, the above condition essentially becomes

$$i \geq \lceil 1/3 (b-2.585) \rceil$$

$\lceil \chi \rceil$ is the smallest integer greater than χ and is called the ceiling function of χ . The upper limit of i is $(b-1)$ since the next higher value of i implies a right shift by b -bit position which yields a zero result. Thus, the range of i is $\lceil 1/3 (b-2.585) \rceil \leq i \leq (b-1)$. For

a 16-bit machine, $i \in \{4, 5, \dots, 15\}$. The block diagram of the elementary CORDIC rotor stage i. e., one section corresponding to α_i , using this principle is shown in Figure 1. The detailed description of this modified CORDIC is given in the reference⁽¹¹⁾.

2.2 Design of the low power CORDIC processor:

A 16-bit CORDIC processor for $\psi = 3.583^\circ$ is designed using the TGL methodology on the sea of gates semicustom design environment. The sea of gates image used here is provided by the OCEAN software (developed in the Delft Technical University, Netherlands). It consists of symmetrically placed fishbone structure constructed by following C3DM (Philips) 1.6 μm double layer CMOS technology. The dimensions of minimum size transistor are 1.6 $\mu\text{m} \times 23.2 \mu\text{m}$ (NMOS), 1.6 $\mu\text{m} \times 29.6 \mu\text{m}$ (PMOS) having transistor pitch = 8 μm , metal layer width = 2.4 μm (for both metal 1 and metal 2) and the threshold voltage of the devices are 0.7 V (NMOS) and -1.1 V (PMOS)⁽¹⁵⁾.

A performance comparison of the TGL design style with the conventional CMOS, NMOS pass transistor and Domino CMOS logic style is carried out using an XOR structure. The simulated results are shown in Table 2, which reveals that the TGL style exhibits somewhat better power and delay performance than the CMOS style. The NMOS pass transistor style shows less power consumption than the TGL but they are not suitable for sea of gates design style as they leads to an wastage of prefabricated PMOS transistors. The critical sizing of the swing restoration buffer required for NMOS pass transistor logic is also difficult to carry out in the sea of gates environment. However, from the layout point of view, implementation of TGL on sea of gates minimizes the wastage of prefabricated PMOS transistors. Unlike NMOS logic the swing restoration

buffer is not required in TGL and the body effect can be made symmetrical for long TGL chain⁽¹⁶⁾. Since the direct powerline access is not required in TGL style, the static power dissipation due to leakage current is expected to be low. Implementation of the logic circuits using TGL requires less number of transistors than the conventional CMOS design style and thus the area consumption in the former case is lower. Considering these features, the TGL style is selected for our purpose.

The performance of the circuit is analyzed by the Switch Level timing Simulator (SLS) provided with the OCEAN package. The extracted netlist from the layout contains nodal, parasitic and routing capacitance. The design is characterized by its delay, dynamic power consumption, Power-Delay Product (PDP) and Energy-Delay Product (EDP). The dynamic power calculation of the circuit is carried out by conventional dynamic power dissipation formula⁽¹⁶⁾

$$P = \sum_{i=1}^n \beta_i C_{Li} V_{DD}^2 f$$

where P is the power consumption, n is the number of internal nodes, β_i is the switching probability of the i th node, C_{Li} is the i th load capacitance, f is the operation frequency and V_{DD} is the supply voltage. The switching probability is considered as 1 in order to include the glitching effect which may exhibit the upper limit of worst case power consumption.

The design of the CORDIC processor is carried out by using two levels of metalization. For some critical routing portions the prefabricated polysilicon gates of the fishbone structure are used. The individual cell isolation is done by connecting the polysilicon gates to the power rails. All the designs of the datapath elements have been carefully optimized.

The simulated circuit extracted from the layout shows that the worst case delay of the CORDIC processor is 22.72 nsec. At 5 V supply with 44 MHz operation frequency, the dynamic power consumption, PDP and EDP of the CORDIC are 62 mW, 1.408 nJ and 3.2×10^{-17} Jsec. respectively. With proper threshold voltage and device scaling, the supply voltage can be lowered further to achieve quadratic improvement in power performance⁽¹⁶⁾.

3. The straight line HT:

3.1 The mathematical formulation:

The Duda – Hart parameterization for detecting straight lines in an edge image is defined as⁽¹⁷⁾

$$x \cos \theta + y \sin \theta = \rho \quad (5)$$

where ρ is the normal distance of the straight line from the origin of the co-ordinate system and θ is the angle between the normal and x -axis as shown in Figure 2. The values of θ and ρ are restricted in the intervals $[0, \pi]$ and $[-R, R]$ respectively. In computing the transform, the $\rho - \theta$ space (often called the parameter space or the Hough space) is quantized in steps of $[\theta_i, \rho_j]$, where i, j are two integers. The quantized parameter space is represented by a 2-D accumulator array. The image space points lying on the line defined by equation (5) with the parameters (θ_i, ρ_j) will vote to the (θ_i, ρ_j) th accumulator cell and generate a histogram. Extraction of the straight line can be done by considering the accumulator counts above a predefined threshold value.

Equation (5) can be implemented using CORDIC which is evident from equation (1). From equation (1), one gets,

$$x' = x \cos \theta + y \sin \theta \quad (6)$$

$$y' = -x \sin \theta + y \cos \theta \quad (7)$$

Equation (6) and (7) show that the CORDIC provides two concurrent outputs with their arguments lying $\pi/2$ angle apart.

Now replacing $(45^\circ + \theta)$ in place of θ in equations (6) and (7), we have another two equations as follows:

$$\sqrt{2}x'' = [(x \cos \theta + y \sin \theta) + (-x \sin \theta + y \cos \theta)] \quad (8)$$

$$\sqrt{2}y'' = [(-x \sin \theta + y \cos \theta) - (x \cos \theta + y \sin \theta)] \quad (9)$$

These equations imply that a scan range of $\theta \in [0, \pi]$ can be divided into four independent subspaces A ($\theta \in [0^\circ, 45^\circ]$), B ($\theta \in [45^\circ, 90^\circ]$), C ($\theta \in [90^\circ, 135^\circ]$) and D ($\theta \in [135^\circ, 180^\circ]$). Thus, parallelly computing equations (6), (7), (8) and (9) with $\theta \in [0^\circ, 45^\circ]$ covers the whole scan range of θ . This result can be utilized for parallel computation of straight line HT.

Defining ρ_A , ρ_B , ρ_C and ρ_D as the sets of ρ values in the subspaces A , B , C and D respectively, four equations can be formulated corresponding to the four subspaces as shown below,

$$\rho_A = x \cos \theta + y \sin \theta \quad (10)$$

$$\sqrt{2}\rho_B = [(x \cos \theta + y \sin \theta) + (-x \sin \theta + y \cos \theta)] \quad (11)$$

$$\rho_C = -x \sin \theta + y \cos \theta \quad (12)$$

$$\sqrt{2}\rho_D = [(-x \sin \theta + y \cos \theta) - (x \cos \theta + y \sin \theta)] \quad (13)$$

In equations (11) and (13) the term $\sqrt{2}$ is a constant and can be taken care by look up table approach or by the addressing logic. Alternatively, $\sqrt{2}\rho_B$ and $\sqrt{2}\rho_D$ can be considered as modified parameters instead of ρ_B and ρ_D . Finally, ρ_B and ρ_D can be

computed from their modified values after thresholding. Thus, defining $\rho_B' (= \sqrt{2}\rho_B)$ and $\rho_D' (= \sqrt{2}\rho_D)$ as the modified parameters in the subspaces B and D respectively, one can rewrite equations (11) and (13) in terms of ρ_A and ρ_C as follows,

$$\rho_B' = \rho_A + \rho_C \quad (14)$$

$$\rho_D' = \rho_C - \rho_A \quad (15)$$

Using CORDIC, equations (10) and (13) can be computed concurrently and from this, equations (14) and (15) can also be computed.

3.2 Array architecture for straight line HT:

The array architecture for straight line HT has been constructed by suitable mapping of equations (10), (12), (14) and (15). The entire θ scan range $[0, \pi/4]$ is quantized into N equal angular segments each having a value θ_0 such that,

$$N\theta_0 = \pi/4 \pm \delta \quad \text{where } \delta = 0, \text{ if } \pi/4 \text{ is an integer multiple of } \theta_0$$

$$\delta \neq 0, \text{ if } \pi/4 \text{ is not an integer multiple of } \theta_0$$

The basic PE is shown in Figure 3 which is designated as H_S . It consists of one CORDIC rotor unit, two adders and four independent accumulator banks: A_A , A_B , A_C and A_D for the storage of ρ_A , ρ_B' , ρ_C and ρ_D' values respectively. The CORDIC rotor parallelly generates the addresses of ρ_A and ρ_C by computing equations (13) and (15). These two ρ values are then utilized for parallel address computation of ρ_B' and ρ_D' using the adders.

N number of such PE (H_S) are cascaded to realize the transform. The distributed accumulator arrangement with each PE ensures conflict free voting operation. The data transfer between the adjacent PE is done asynchronously. This will suppress the data skewing and the computation becomes data driven. However, a suitable handshaking protocol has to be adopted. Since the PEs are pipelined, in the steady state, parallel HT

computation at different $\theta (= j\theta_0, j \in \{1, 2, \dots, N\})$ can be done for N feature points. The peak detection can be carried out by checking the accumulator counts parallelly for all H_s . The total architecture is shown in Figure 4. The whole operation is summarized in the following pseudocode,

Let $p \in \{1, 2, \dots, N\}$ be the index of the PE and $q \in \{1, 2, \dots, M\}$ be the index of the accumulator array for each PE. θ_0 is the rotation introduced by a single processor and $N\theta_0 = \pi/4 \pm \delta$. ρ_{pqA} denotes the value of ρ corresponding to the q th accumulator cell in subspace A for angle $p\theta_0$ and so on.

1. *$\forall p$ th PE, initialize the accumulator cell counts to zero.*
2. *For each edge pixel (x, y) with grey level equal to one,*

$\forall p$ th PE, do in parallel

(a) compute in parallel

$$\rho_{pqA} = x_p = x_{(p-1)} \cos \theta_0 + y_{(p-1)} \sin \theta_0 = x \cos (p\theta_0) + y \sin (p\theta_0)$$

$$\rho_{pqC} = y_p = -x_{(p-1)} \sin \theta_0 + y_{(p-1)} \cos \theta_0 = -x \sin (p\theta_0) + y \cos (p\theta_0)$$

(b) compute in parallel

$$\rho_{pqB}' = \rho_{pqA} + \rho_{pqC}$$

$$\rho_{pqD}' = -\rho_{pqA} + \rho_{pqC}$$

(c) update q th Hough array in parallel for all the subspaces.

(d) Check the busy bit of $(p+1)$ th PE.

if busy bit is high

enter in wait state.

if busy bit is low

transfer x_p, y_p to $(p+1)$ th PE in logic low and high state respectively.

(e) assert busy bit of p th and $(p+1)$ th PE in logic low and high state respectively.

(f) get new input.

(g) assert busy bit of p th PE in logic high state.

3. Look for peaks in the accumulator array $\forall p$.

3.3 Performance of the architecture:

To evaluate the performance of the proposed architecture and to compare it with the other proposed methods we assume that in the proposed one θ space is quantized in step of θ_0 , where $N\theta_0 = \pi/4 \pm \delta$, n be the number of edge pixels to be processed and m be the number of accumulators per subspace for full set of ρ for each θ_0 .

3.3.1 Computational complexity:

The total number of operations required for ρ computation using the conventional method is $2n\pi/\theta_0$ trigonometric multiplication + $n\pi/\theta_0$ additions whereas, in the proposed method, the total arithmetic operations required is $6n\pi/4\theta_0 (=1.5 n\pi/\theta_0)$ additions which is much less than the conventional method as the θ scan range is restricted between $[0, \pi/4 \pm \delta]$. The total accumulator cell requirement in the proposed method is equal $m\pi/\theta_0$, which is same as the conventional one.

3.3.2 Area – Time complexity (AT):

Considering the area of one adder be $O(a)$ and the area of one accumulator cell be $O(a_c)$, the area of one PE is $O(6a+4ma_c)$. Thus, the area consumed by the proposed architecture is

$$A = O[(6a+4ma_c) (\pi/4\theta_0)] = O[N (6a+4ma_c)]$$

The latency of the proposed architecture is $O(\pi/4\theta_0)$ and the time required to compute the rest $(n-1)$ feature points is $O(n-1)$, where the time required for one PE is taken as $O(1)$. Thus, the total computation time becomes,

$$T = O[(\pi/4\theta_0) + (n-1)] = O[N + (n-1)]$$

If the time required for an adder is T_a , the total computation time T can be represented as

$$T = O[2\{N + (n-1)\}T_a]$$

So the AT of the proposed one is equal to $O[2N(6a+4ma_c)\{N + (n-1)\}T_a]$.

4. Comparison with other architectures:

In this section the proposed architecture is compared with some of the existing architectures based on the nature of PE, angle scan range, time requirement for histogram generation and extra hardware requirements. The comparison is carried out by considering the number of θ_0 values in the range $[0, \pi/4+\delta]$ to be N , $O(T_s)$ and $O(T_a)$ be the time required for one shift and one addition operation respectively, n is the number of feature points and M be the required number of iterations for conventional CORDIC unit. The results are shown in Table 3. All the referenced architectures except the architecture in the reference⁽³⁾ requires larger θ scan range than the proposed architecture implying higher computational requirement than the proposed one. Though the effective scan range for the architecture in reference⁽³⁾ is approximately same to that of our architecture, the total time requirement of the proposed one is less than that of the architecture of the reference⁽³⁾ as is evident from the Table 3. Thus, the proposed architecture enjoys superiority in speed and computational requirement than others. Quantitative measurements in Table 3 are done by considering $\theta_0 = 2^{-4} = 0.625$ radians = 3.579545° ,

$N = 13$ and $\delta = 1.534085^\circ$ and $T_a = 7.1$ nsec (in $1.6 \mu\text{m}$ sea of gates technology). Under these considerations, a full set of ρ value generation for one feature point takes 295.36 nsec, which seem to be considerably low.

Since this architecture utilizes CORDIC, unlike multiplier based designs, the precomputations of 'cos' and 'sin' values are not required which in its way eliminates the requirement of RAM. This makes the architecture more time effective compared to the multiplier based designs, as in the later case, the RAM access time become a deterministic constraint for ρ computation as is evident in the reference⁽⁴⁾.

In the proposed architecture, the CORDIC units require only adder-subtractor and the architecture can simultaneously compute ρ for N angles in the θ scan range of $[0, \pi/4 + \delta]$. Being composed of the scaling free CORDIC (discussed in Section 2), the architecture is more hardware efficient compared to the other CORDIC based implementations and does not require the extra conversion unit like the architecture of reference⁽¹⁰⁾.

The distributed accumulator cell arrangement with each PE ensures conflict free voting operation. This facilitates a parallel approach for peak detection by simultaneously checking the count of the accumulators for all θ_0 , i. e. for all PE.

The proposed one is modular and shows better regularity than other architectures which is suitable for VLSI implementation. Being asynchronous and pipelined, it is advantageous from low power and fault tolerant application point of view. Since the computation is data driven, the PE synchronization problem (typical to the systolic arrays when the array size becomes large) does not occur. This, in turn, suppresses the data skewing and subsequent glitches which leads to power saving.

In light of the above results and discussion, it can be conjectured that this architecture can be considered as a potential candidate for low power high performance real time straight line HT using VLSI.

5. Circular and elliptic HT:

One common method applied for extraction of elliptic pattern from a given image data is the tristage⁽¹²⁾ approach. In such an approach, the computation is carried out in three hierarchical stages namely, detection of the center, detection of orientation and the major and minor radii estimation. This method can be applied for detecting circular pattern as well where instead of three hierarchical stages only two hierarchical stages are required *viz.*, the estimation of the center and the radius of the circle. In both the cases, the pattern detection procedure is computation intensive and one may require parallel processing array architectures corresponding to the different stages of the hierarchy where each array architecture can be considered as a subunit of the whole system. Though in the hierarchical approach for detecting circle and ellipse all the stages are computation intensive, the maximum computation involves at the final stage of the hierarchy i.e., for estimating radius of the circle and the major and minor radii of the ellipse. These stages demand diversified mathematical operations like squaring, division, addition, square root evaluation and multiplication. From this point of view, in this section, we have concentrated on developing parallel processing array architectures corresponding to this stage of the hierarchy (which can be considered as a subunit of the entire system for circular or elliptic Hough transform respectively) only. Our principal aim is to reduce the computational requirements for detecting the radius of the circle and semi-major and semi-minor radii of the ellipse using their parametric representation.

Subsequently, CORDIC based array architectures are proposed for them. Analyses made here are based on two considerations that are,

- The origin of the curves is already known.
- The orientation angle of the ellipse is known.

5.1 Circular HT:

The equation of a circle can be stated as,

$$x^2 + y^2 = r^2 \quad (16)$$

where, (x, y) is a point lying on the circle and ' r ' is the radius. In parametric form the length of the radius is given by,

$$x \cos \theta + y \sin \theta = r \quad (17)$$

where θ is the angle made by the radius vector with the positive x -axis as shown in Figure 5. Equation (17) is exactly similar to equation (5) and thus the same architecture for straight line HT can be extended for circular HT. All the points lying on the same circle will give same radius value for different θ . Considering the co-ordinate system where the origin is coincident with the center of the circle, the θ scan range will be of $[0, 2\pi]$. This range can be divided into eight subspaces (a, b, c, d, e, f, g, h) and the θ scan range can be restricted to $[0, \pi/4 \pm \delta]$. The values of r in different subspaces can be calculated according to the following equations,

$$r_a = x \cos \theta + y \sin \theta \quad (\theta \in [0, 45^\circ \pm \delta]) \quad (18)$$

$$r_c = -x \sin \theta + y \cos \theta \quad (\theta \in [90^\circ, 135^\circ \pm \delta]) \quad (19)$$

$$\sqrt{2}r_b = r'_b = r_a + r_c \quad (\theta \in [45^\circ, 90^\circ \pm \delta]) \quad (20)$$

$$\sqrt{2}r_d = r'_d = r_c - r_a \quad (\theta \in [135^\circ, 180^\circ \pm \delta]) \quad (21)$$

$$r_e = -r_a \quad (\theta \in [180^\circ, 225^\circ \pm \delta]) \quad (22)$$

$$r_f = -r_b' \quad (\theta \in [225^\circ, 270^\circ \pm \delta]) \quad (23)$$

$$r_g = -r_c \quad (\theta \in [270^\circ, 315^\circ \pm \delta]) \quad (24)$$

$$r_h = -r_d' \quad (\theta \in [315^\circ, 360^\circ \pm \delta]) \quad (25)$$

Where, the suffix of r defines their values in appropriate subspaces and r_b' and r_d' are considered as modified parameters in the respective subspaces. It can be observed that only (18) and (19) are needed to be computed which can be readily done using CORDIC. Equations (20) and (21) can be derived from (18) and (19) by simple addition and subtraction. The other four equations can be directly computed by only changing the signs of the equations (18), (19) and (21). Thus, for detecting the radius of circle, the architecture for straight line HT can be used with extra four accumulator arrays for each PE since r -values for eight subspaces are to be stored. Finally, checking the votes of the same indexed accumulator cells for different PE (i. e. for different θ), the radius of the circle can be found out. If the circle has its center at (x_0, y_0) , then in this formulation, x and y have to be replaced by $X = (x-x_0)$ and $Y = (y-y_0)$. The basic PE (designate as H_C) and the architecture for the circular HT are shown in Figure 6 (a) and (b) respectively.

5.2 Elliptic HT:

The parametric equation of a point (x, y) lying on an ellipse with semi-major and semi-minor radii ' a ' and ' b ' respectively, is given by

$$x = a \cos \theta \quad (26)$$

$$y = b \sin \theta \quad (27)$$

where θ is the angle made by the radius vector (from origin to the (x, y) point) with the positive x -axis.

Now, defining $1/a = a'$ and $1/b = b'$, equation (26) and (27) can be written as

$$a' = (1/x) \cos \theta \quad (28)$$

$$b' = (1/y) \sin \theta \quad (29)$$

The quantities a' and b' can be considered as modified parameters instead of a, b and can be quantized accordingly. Following the same line of mathematical formulation of circular HT, here also the total θ scan range can be restricted to $[0, \pi/4 \pm \delta]$ and the whole Hough space of $[0, 2\pi]$ can be divided into eight subspaces (a, b, c, d, e, f, g, h). The modified parameter values in these subspaces can be computed according to the following equations,

$$a'_a = (1/x) \cos \theta \quad \text{and} \quad b'_a = (1/y) \sin \theta \quad (\theta \in [0, 45^\circ \pm \delta]) \quad (30)$$

$$a'_c = -(1/x) \sin \theta \quad \text{and} \quad b'_c = (1/y) \cos \theta \quad (\theta \in [90^\circ, 135^\circ \pm \delta]) \quad (31)$$

$$\sqrt{2}a'_b = a''_b = a'_a + a'_c \quad \text{and} \quad \sqrt{2}b'_b = b''_b = b'_a + b'_c \quad (\theta \in [45^\circ, 90^\circ \pm \delta]) \quad (32)$$

$$\sqrt{2}a'_d = a''_d = a'_c - a'_a \quad \text{and} \quad \sqrt{2}b'_d = b''_d = b'_c - b'_a \quad (\theta \in [135^\circ, 180^\circ \pm \delta]) \quad (33)$$

$$a'_e = -a'_a \quad \text{and} \quad b'_e = -b'_a \quad (\theta \in [180^\circ, 225^\circ \pm \delta]) \quad (34)$$

$$a'_f = -a''_b \quad \text{and} \quad b'_f = -b''_b \quad (\theta \in [225^\circ, 270^\circ \pm \delta]) \quad (35)$$

$$a'_g = -a'_c \quad \text{and} \quad b'_g = -b'_c \quad (\theta \in [270^\circ, 315^\circ \pm \delta]) \quad (36)$$

$$a'_h = -a''_d \quad \text{and} \quad b'_h = -b''_d \quad (\theta \in [315^\circ, 360^\circ \pm \delta]) \quad (37)$$

The suffixes of a' and b' define their values in appropriate subspaces. Thus, as in the case of circular HT, only two equations (30) and (31) are to be computed to get the addresses of the appropriate accumulator cells. Accumulator addresses governed by equations (32) and (33) can be generated by simple addition and subtraction of equations (30) and (31).

The other four addresses can be computed by changing the sign of the addresses given by equations (30) and (33). Finally, the votes of the same indexed accumulator cells for different PE will determine the shape of the ellipse and the conversion from a', b' to a, b can be carried out using a look-up table. However, the nature of equations (32) and (33) suggests that each PE requires two CORDIC units operating parallelly. Each PE also requires eight 2-D accumulator arrays of which each one is dedicated for a particular subspace. The basic PE designated as H_e and the architecture are shown in Figure 7 (a) and (b) respectively.

If the center of the ellipse lies at (x_0, y_0) point, then in the above formulation the x and y values have to be replaced by $X = (x - x_0)$ and $Y = (y - y_0)$ respectively.

5.3 Discussions on elliptic and circular HT architecture:

Compared to the conventional method, the proposed formulations require less number of arithmetic operations to detect the radius of the circle and semi-major and semi-minor radii of the ellipse. In evaluating these parameters conventional method requires multiplication, squaring, subtraction, division and square root evaluation⁽¹²⁾. In our formulation, only the CORDIC rotation is required which in turn requires only additions and cross-coupled bus connections. Thus, a large area and resource saving is possible. In the proposed architectures concentric circles and ellipses can be found out directly by checking the votes of the accumulator cells with different indices in their respective cases.

6. Conclusions:

In this paper, a modified scaling free CORDIC based asynchronous array architecture for straight line HT is proposed which eliminates the requirement of

precomputations and RAM, making this one hardware and time efficient compared to the multiplier based architectures. Using an angle parallelization scheme the computation burden is reduced to approximately 25 %. Moreover, this one enjoys superiority in processing speed compared to some other architectures.

The architectures proposed in this paper for computing circular and elliptic HT with known centers and orientations require less number of arithmetic operations compared to the conventional formulations. In our formulation, the computation in eight subspaces can be carried out parallelly which results into saving of hardware resources and speeds up the computation time. For computation of circular and elliptic Hough transform utilizing the hierarchical method, these architectures can be considered as the subunits of the respective systems. On the other hand, one may compute the less computation intensive stages of the hierarchy *viz.*, centers (for circle and ellipse) and the orientation (for ellipse) using software and then can utilize these array architectures for fast estimation of the radius (for circle) and major and minor radii (for ellipse).

All the proposed architectures require same number of accumulator cells as that of the conventional formulations. The distributed accumulator arrangement ensures conflict free voting operation and facilitates parallel peak detection. Concentric circles and ellipses can be found out directly by checking the votes of different indexed accumulator cells. The modularity and regularity of the proposed architectures makes them attractive for VLSI monolithic integration. Being asynchronous and data driven, these architectures may be advantageous for low power and fault tolerant applications. However, the elliptic HT architecture suffers from the requirement of inverse of the pixel co-ordinates as inputs. This can be solved by using two conventional CORDIC units operating in

vectorization mode. Though, this problem is not present in straight line and circular HT architectures.

The basic CORDIC unit has been designed using TGL on 1.6 μm sea of gates semicustom environment which exhibits 62 mW power consumption at 5 V supply and 44 MHz operation frequency. With device scaling, this CORDIC unit is expected to operate at lower supply voltage, which implies that a quadratic advantage in power consumption can be achieved.

Considering all these points, it can be conjectured that the proposed architectures can be considered as good candidates for low power high performance real time HT computation.

References

1. P. V. C. Hough, Method and means of for recognizing complex patterns, U. S. Patent 3069654 (1962).
2. K. Y. Huang, K. S. Fu, T. H. Sheen and S. W. Cheng, Image processing of seismograms: (A) Hough transformation for the detection of seismic patterns; (B) thinning process in the seismogram, *Pattern Recognition* **18**, 429 – 440 (1985).
3. D. Timmerman, H. Hahn and B. J. Hosticka, Hough transform using CORDIC method, *Electronics Letters* **25**, 205 0 206 (1989).
4. K. Hanahara, T. Maruyama and T. Uchiyama, A real time processor for the Hough transform, *IEEE Trans. PAMI* **10**, 121 – 125 (1987).
5. H. Y. H. Chuang and C. C. Li, A systolic array processor for straight line detection by modified Hough transform, *IEEE Workshop, Comput. Arch. Pattern Analysis Database Mgmt.*, pp. 300 – 303 (1985).
6. H. A. H. Ibrahim, J. R. Kender and D. E. Shaw, The analysis and performance of two middle-level vision tasks on a fine grained SIMD tree machine, *Conf. Comput. Vision Pattern Recognition*, 248 – 256 (1985).
7. H. F. Li, D. Pao and R. Jayakumar, Improvements and systolic implementation of the Hough transformation for straight line detection, *Pattern Recognition* **22**, 697 – 706 (1989).
8. F. M. Rhodes et al., A monolithic Hough transform processor based on restructurable VLSI, *IEEE Trans. PAMI* **10**, 106 – 110 (1988).
9. T. M. Silberberg, The Hough transform on the geometric arithmetic parallel processor, *IEEE Workshop, Comput. Arch. Pattern Analysis Database Mgmt.*, pp. 387 – 393 (1985).
10. J. D. Bruguera, N. Guil, T. Lang, J. Villalba and E. L. Zapata, CORDIC based parallel / pipelined architecture for the Hough transform, *VLSIVideo* **12**, pp. 207 – 221 (1996).
11. A. S. Dhar and Swapna Banerjee, An array architecture for fast computation of discrete Hartley transform, *IEEE Trans. Circuits Syst.* **38**, 1095 – 1098 (1991).

12. H. K. Muammar and M. Nixon, Tristage Hough transform for multiple ellipse extraction, *IEE Proc. – E* **138**, 27 – 35 (1991).
13. J. E. Volder, The CORDIC trigonometric computing technique, *IRE Trans. Electronic Computers* **EC-8**, 330 – 334 (1959).
14. J. S. Walther, A unified algorithm for elementary functions, *AFIPS Conf. Proc.* **38**, 379 – 385 (1971).
15. P. Groeneveld and P. Stravers, *OCEAN: The sea-of-gates design system user's manual* (1993).
16. A. Bellaouar and M. I. Elmasry, Low-Power Digital VLSI Design, Circuits and Systems, *Kluwer Academic Publishers*, 1995.
17. R. O. Duda and P. E. Hart, Use of the Hough transformation to detect lines and curves in pictures, *Communs. ACM* **15**, 11 – 15 (1975).

Table 1

	$m = 1$	$m = 0$	$m = -1$
Rotation $z \rightarrow 0$	$x' = x \cos z + y \sin z$ $y' = -x \sin z + y \cos z$	$x' = x$ $y' = y - zx$	$x' = x \cos hz - y \sin hz$ $y' = -x \sin hz + y \cos hz$
Vectoring $y \rightarrow 0$	$x' = \sqrt{(x^2 + y^2)}$ $z' = z - \tan^{-1}(y/x)$	$x' = x$ $z' = z - (y/x)$	$X' = \sqrt{(x^2 - y^2)}$ $Z' = z - \tanh^{-1}(y/x)$

Table 2

Logic family	Average output capacitance (fF)	Average Delay (nsec.)	Power dissipation (mW)	Power Delay Product (pJ)	Energy Delay product (10^{-21} Jsec.)
Static CMOS	304.106	1.256	1.5329	1.9253	2.4181
Domino CMOS	192.969	1.35	2.1867	2.9522	3.9854
NMOS pass logic	42.1623	0.153	0.052	0.007956	0.001217
TGL	138.609	0.256	0.1732	0.04433	0.01134

Table 3

Architecture	Nature of PE	Scan range of θ	Time required to generate histogram	Extra requirements
Rhodes et al. ⁽⁸⁾	Multipliers, architecture is WSI	$[0, \pi]$	20 msec. (image size 256×256 , 1/10 of the image are edge pixels)	Precomputed values of $\sin\theta$, $\cos\theta$ and RAM
Hanahara et al. ⁽⁴⁾	Array multipliers and off chip components	$[0, \pi]$	256 msec. For 1024 feature points.	Precomputed values of $\sin\theta$, $\cos\theta$ and RAM
Timmerman et al. ⁽³⁾	Radix-2 conventional CORDIC unit.	Effective scan range is $[0, \pi/4]$	$O[2MNn (T_s + T_a)]$	Scaling factor compensation.
Bruguera et al. ⁽¹⁰⁾	Mixed radix pipelined CORDIC	$[0, \pi/2]$	$O[52T_a + 4(n-1) + T_{\text{conv}}]$	Scaling factor compensation, extra conversion unit and RAM.
Proposed	Scaling free CORDIC. The architecture is asynchronous.	$[0, \pi/4 \pm \delta]$	$O[2\{N+(n-1)\} T_a]$ 149.179 μsec for 256×256 image and 23.569 μsec for 1024 points.	Scaling of ρ by the constant factor $\sqrt{2}$ in B and D subspaces.

Table Captions

Table 1. The CORDIC arithmetic function.

Table 2. Comparison of different logic families using the XOR structure.

Table 3. Comparison of different architectures for straight line Hough transform.

Figure Captions

Figure 1. The elementary CORDIC arithmetic unit.

Figure 2. Normal description of the straight line.

Figure 3. The basic PE for straight line Hough transform.

Figure 4. The array architecture for straight line Hough transform.

Figure 5. The parametric representation of a circle.

Figure 6 (a). The basic PE for circular Hough transform.

Figure 6 (b). The array architecture for circular Hough transform.

Figure 7 (a). The basic PE for elliptic Hough transform.

Figure 7 (b). The array architecture for elliptic Hough transform.

Authors' biography

Koushik Maharatna was born in Calcutta, India in the year 1972. He received his Bachelors degree in Physics in the year 1993 from the University of Calcutta. In 1995 he received Masters degree in Electronics Science from the same University. In 1997 he joined the Ph. D. program under the joint collaboration of Jadavpur University, Calcutta and Indian Institute of Technology, Kharagpur and completed the doctoral work in the year 2000. Currently he is a Post Doctoral fellow in the Institute for Semiconductor Physics, Frankfurt (Oder), Germany. His research interests include digital signal processing, VLSI array architectures and low power circuit realization.

Swapna Banerjee received her B.E. and M.E. degree in Electronics and Telecommunication Engineering from Jadavpur University, India in 1971 and 1974 respectively. In 1981 she received her Ph. D. degree from the Indian Institute of Technology, Kharagpur. She did her Post Doctorate from the Tokyo University, Japan. Since 1981 she has been with the Dept. of Electronics and Electrical Communication Engineering at Indian Institute of Technology, Kharagpur. At present she is Professor. Her research interests include device modeling, array architecture of signal processing for biomedical applications and knowledge base systems.

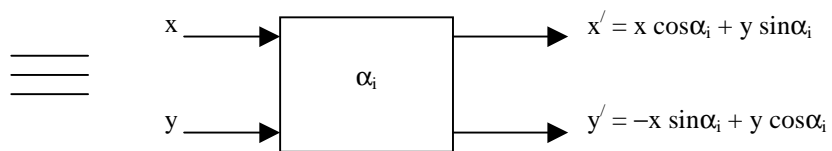
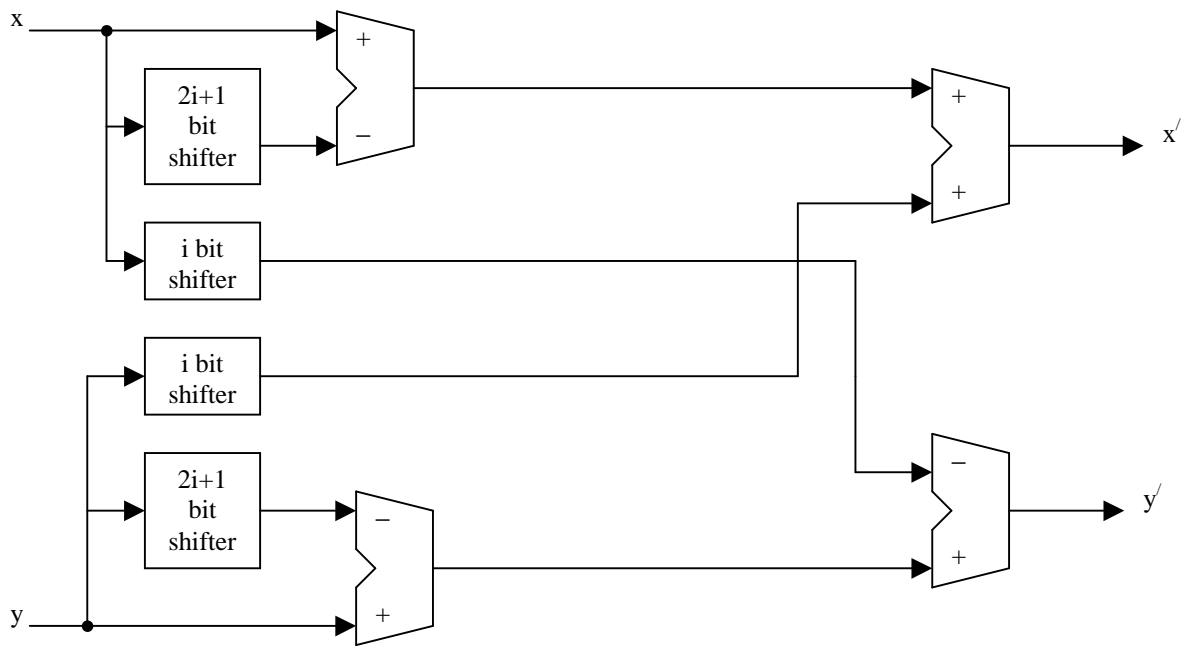


Figure 1

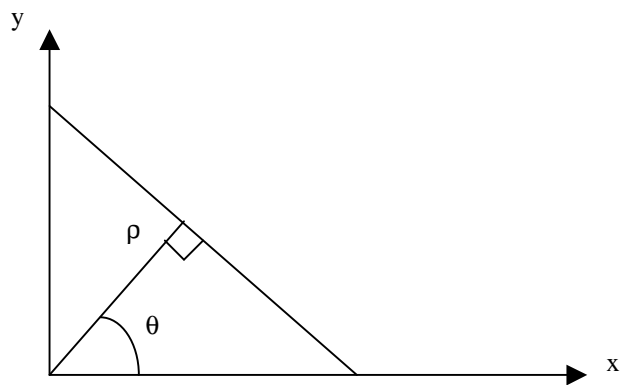


Figure 2

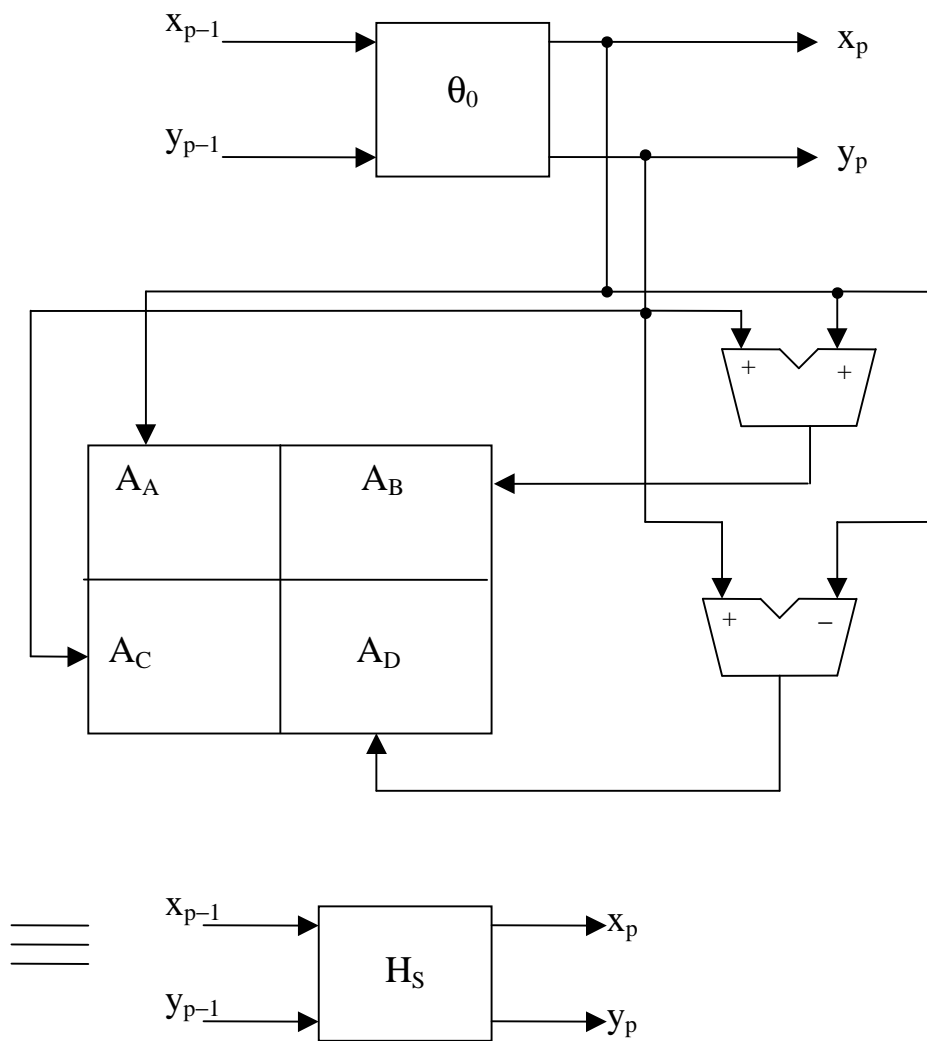


Figure 3

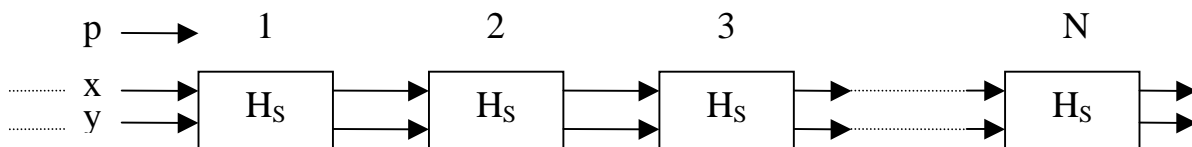


Figure 4

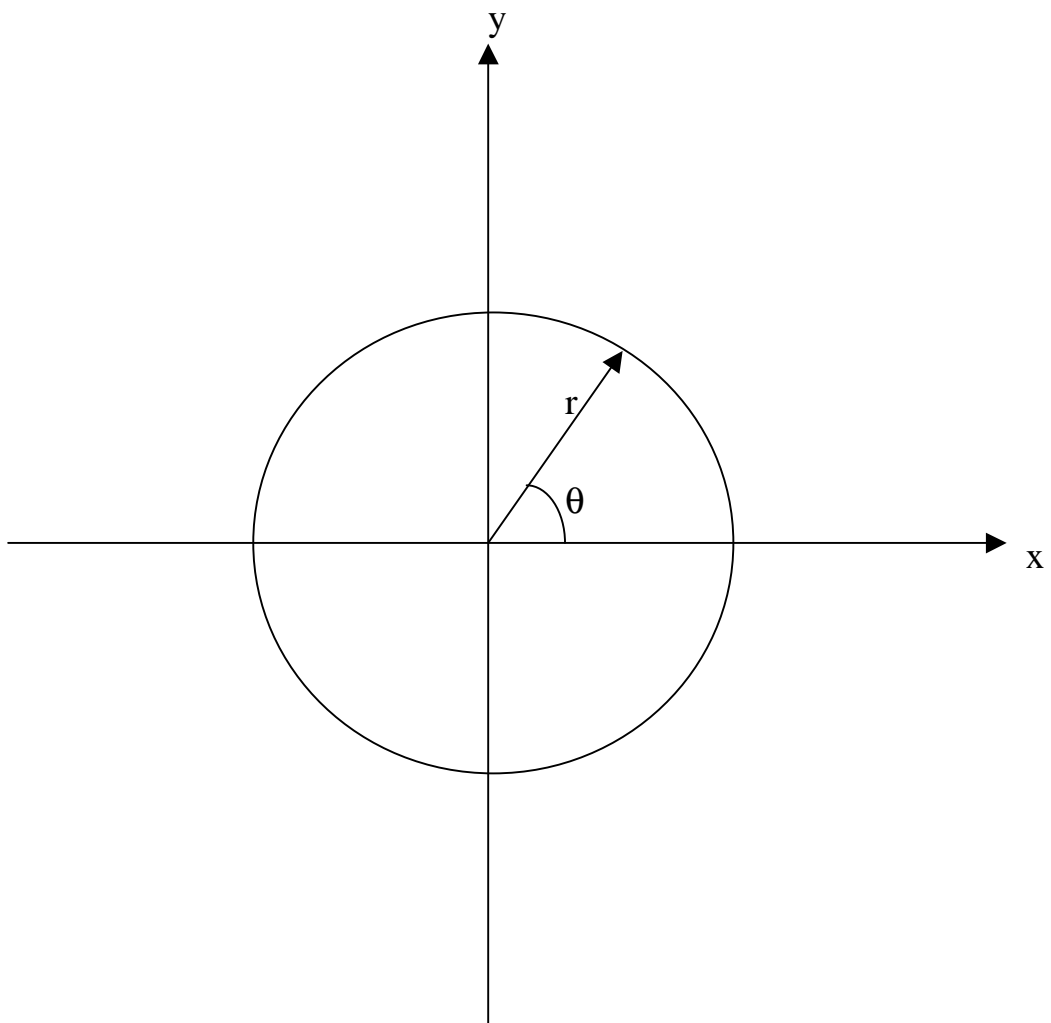


Figure 5

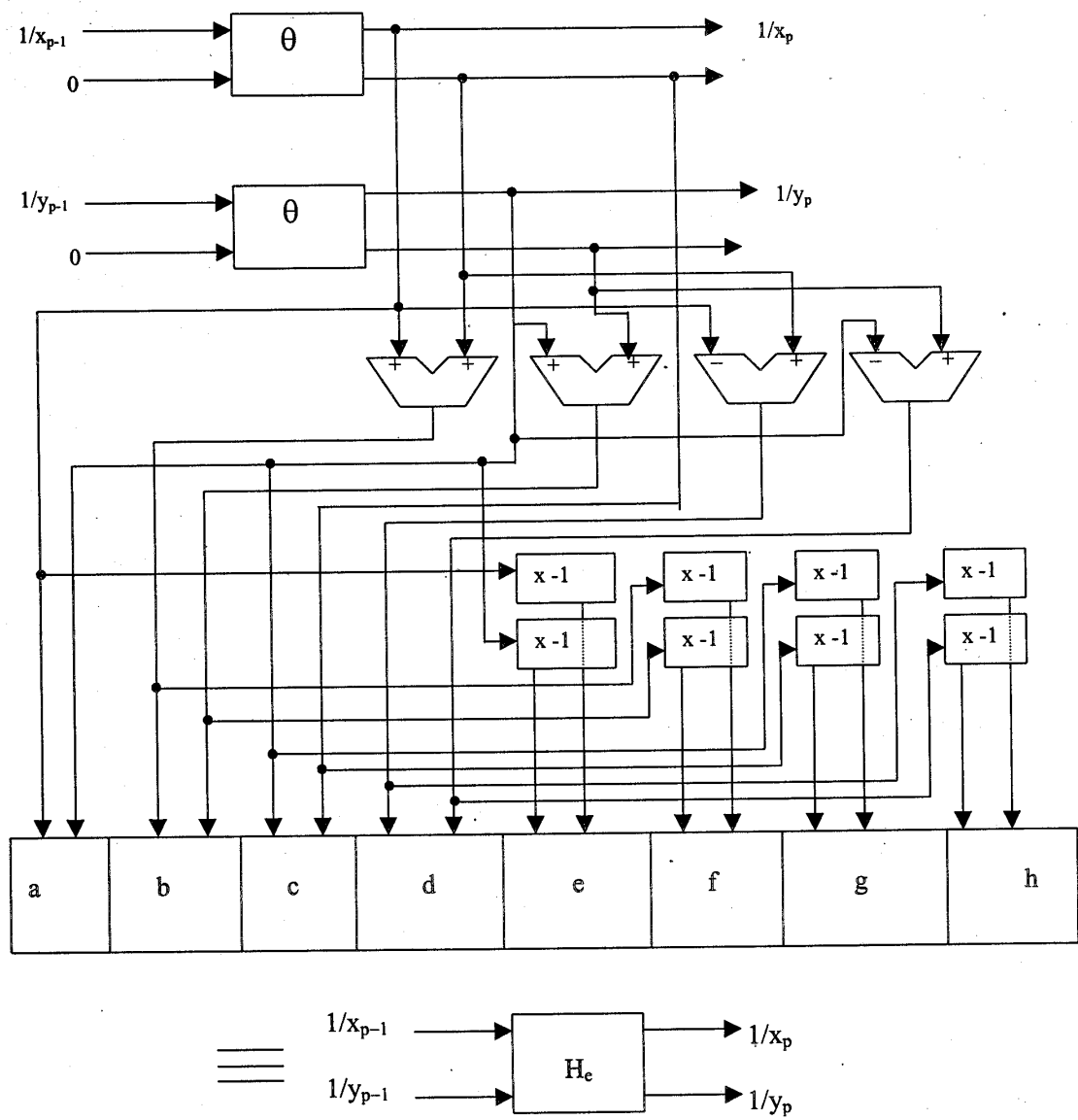


Figure 7 (a)

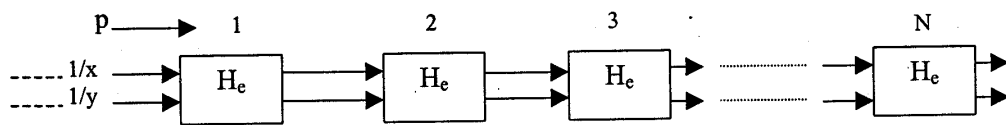


Figure 7 (b)