Design of Pipelined Adaptive DFE Architecture For High Speed Channel Equalization

A. Mandal, *Student Member, IEEE*, R. Mishra, *Member, IAENG*

***Abstract*— An adaptive equalizer is a vital and essential component for high-speed reliable data transmission through wired as well as wireless communication channels to minimize Inter Symbol Interference (ISI). The non-linear Adaptive Decision Feedback Equalizer (ADFE) finds enormous use in applications where the channel distortion is severe for a linear equalizer to handle. High hardware cost of the system makes an urge to redesign it in order to minimize overall circuit complexity and cost. In this paper, realization of COordinate Rotation Digital Computer (CORDIC) based pipelined ADFE architecture using reformulated least-mean-square (LMS) algorithm has been presented. The realization results area efficient, better speed and throughput with good convergence. The efficacy of the proposed equalizer is corroborated with MATLAB simulations for mitigation of severe channel distortion arise due to ISI in high speed communication systems.**

***Index Terms*— Inter Symbol Interference (ISI), Adaptive Decision Feedback Equalizer (ADFE), CORDIC, LMS algorithm, Feed Forward Filter (FFF), Feedback Filter (FBF), Pipelined Architecture.**

I. Introduction

When channel frequency response deviates from the ideal of flat magnitude and linear phase response, both tails of the transmitted pulse will interfere with the neighboring pulses creating an *Inter Symbol Interference* (ISI) and it may lead to erroneous decision that increases the probability of error. To overcome this problem an equalizer at the receiver with a transfer function which is essentially an inverse of the channel transfer function is required [1]. So within the channel pass-band, ADFE ideally forms inverse transfer function to eliminate ISI .The performance of non-linear ADFE is better on a channel with spectral nulls in their frequency response characteristics than the linear equalizer which introduces significant amount of additive noise present in the received signal by introducing a large gain [2].

Since last few decades, a tremendous inclination among the researchers toward the implementation of CORDIC algorithm [3, 4] in various applications of digital signal processing arena had been witnessed. The modularity, compatibility, pipelinability, numerical stability and efficiency in generation of trigonometric and hyperbolic function have made the CORDIC algorithm suitable for implementation of fast Fourier transform (FFT), discrete Fourier transforms (DFT), adaptive lattice filter and many more applications [5, 6]. In this paper, an adaptive transversal filter has been designed with a pipelined CORDIC unit as a core processing element in the equalizer for filtering and filters weight updating. Unlike the conventional LMS algorithm, the rotational angles rather than the tap weights are updated directly. The most popular and highly robust LMS algorithm has been efficiently mapped to obtain trigonometric form of LMS algorithm which facilitates in incorporating CORDIC processing element to participate in filtering as well as weight updating operations of the Feed Forward Filter (FFF) of the ADFE. Due to trivial bit length, decision feedback loop (DFL) has been simplified with general purpose multiplier to reduce complexity of the Feedback Filter (FBF). The proposed architecture uses CORDIC blocks instead of multipliers in the FFF and hence is more efficient in terms of internal numerical errors and power consumption.

Manuscript submitted April 1, 2015, revised September 3, 2015.

A. Mandal and R. Mishra are with the School of Information and Communication Technology, Gautam Buddha University, Greater Noida, U.P.- 201308, India (e-mail: amritkar2k@gmail.com; rmishra@gbu.ac.in).

The ADFE is widely used equalization technique for radar, antenna beam forming, digital communication systems, wireless video telephony, magnetic storage and many more applications [7, 8]. Wide variants of ADFE are available in the literature [9-12]. However, pipeline design of ADFE is known to be a difficult task for high speed applications. It is well known fact that the clock rate is limited by DFL. A lot of researches have been done to reduce hardware overhead and at the same time, to obtain better convergence. In this paper, we have used pipeline design technique throughout the architecture to reduce hardware complexity and to obtain high throughput for high speed applications. The modular design of multiplier has been incorporated to avoid place and route problem during physical layout. The combination of modified booth encoding (MBE) techniques [13, 14] along with partial product reduction [15-17] have facilitated in reduction of overall latency of the design to a single adder which has been implemented with carry look-ahead adder (CLA). This kind of design not only facilitates in easy implementation on FPGA device, but also reduces complexity significantly as compared to multiplier and accumulator (MAC) based design. The CORDIC processing element has been used by optimizing the quantization error [18] and to facilitate easy implementation of the equalizer for digital signal processing application. The convergence behaviors under various step sizes and its ability to generate inverse transfer function in severe channel mismatched situation have been adequately analyzed.

The rest of the paper is organized as follows. In the next section, we review the simple LMS algorithm and subsequently mapped into trigonometric form. In section III, CORDIC algorithm has been revisited and corresponding pipelined architecture is implemented. Section IV deals with the design of pipelined Adaptive Decision Feedback Equalizer along with explanation of associated components used with the design. Performance analysis of proposed design and discussion part of the paper is accommodated section V and finally, concluding remarks are presented in section VI.

II. Review of LMS Algorithm for CORDIC Based Application

The LMS algorithm is one of the simplest as well as robust adaptive algorithms available in literature. We would like to reformulate LMS in trigonometric form for our design. Let the FIR filter is linear discrete time filter $x(n)$ and $d(n)$ are input and desired output sequence of the FIR filter (**Fig. 1**).

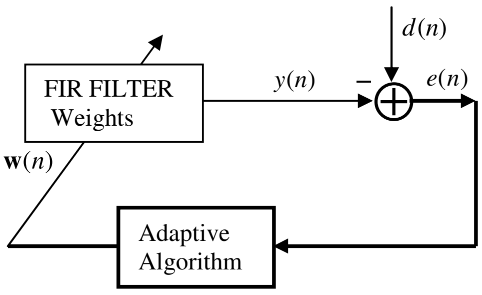

Fig. 1. LMS algorithm to update FIR filter weights

Let $\mathbf{w}$ and $\mathbf{x}(n)$ are filter coefficient and input vector of N-tap filter,

$$\mathbf{w} = [w_0, w_1, \ldots\ldots\ldots, w_{N-1}]^T$$
$$\mathbf{x}(n) = [x(n), x(n-1), \ldots\ldots\ldots, x(n-N+1)]^T \qquad (1)$$

The output signal of the adaptive filter, $y(n)$, can be computed as the inner product of $\mathbf{w}(n)$ and $\mathbf{x}(n)$.

$$y(n) = \sum_{m=0}^{N-1} w_m(n)x(n-m) = \mathbf{w}^T(n)\mathbf{x}(n) \qquad (2)$$

The error signal $e(n)$ is expressed as the difference between the desired response $d(n)$ and the filter output $y(n)$, i.e,

$$e(n) = d(n) - \mathbf{w}^T(n)\mathbf{x}(n) \qquad (3)$$

The weight vector is updated iteratively in such manner that the mean-square error (MSE), ε is minimized.

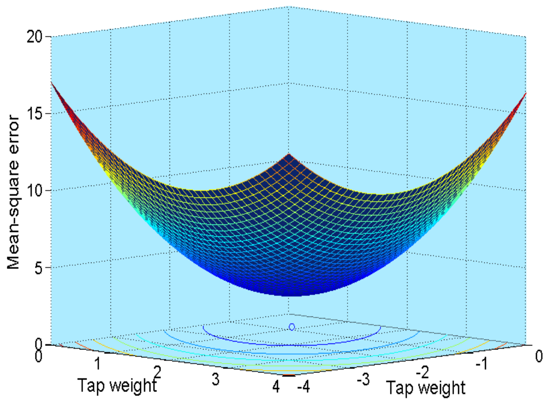

Fig. 2 Error surface curve to derive optimal weight of a FIR filter

$$\varepsilon \equiv E\{e^2(n)\}. \qquad (4)$$

$$= E\{d^2(n)\} - 2\mathbf{w}^T\mathbf{p} + \mathbf{w}^T\mathbf{R}\mathbf{w} \qquad (5)$$

where $\mathbf{R} \equiv E[\mathbf{x}(n)\mathbf{x}^T(n)]$ is the input auto-correlation matrix and $\mathbf{p} \equiv E[\mathbf{x}(n)d(n)]$ is the cross-correlation vector. Solving the equation, $\nabla_w \varepsilon = 0$, the optimal weights of a FIR filter can be derived.

$$\mathbf{w}_{opt} = \mathbf{R}^{-1}\mathbf{p}. \qquad (6)$$

The direct calculation of the optimum filter weight ($\mathbf{w}_{opt}$) from $\mathbf{w}_{opt} = \mathbf{R}^{-1}\mathbf{p}$ is not practicable as matrix inversion involves huge complex multiplications. To reduce hardware overhead in terms of multiplier, steepest descent search method can be used for the design. The expression for steepest descent search method can be written for n-th index as

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \frac{\mu}{2} \times \nabla_w \varepsilon \Big|_{w=w(n)} \qquad (7)$$

where μ is step-size which is responsible for determining the convergence speed of the design. Putting $\mathbf{R}$ and $\mathbf{p}$ in equation (7), we get the simplest LMS algorithm in the following form:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu\, \mathbf{x}(n)\, e(n) \qquad (8)$$

The pipelined implementation of conventional LMS algorithm is difficult because of its recursive behavior. That is why delayed LMS (DLMS) algorithm has been used. The feedback error, $e(n)$ generated at the n-th iteration cannot be available at the same iteration for weight updating process. The delay is referred to adaptation delay. If the adaptation delay is generated for L numbers of pipelined stages, the error in DLMS algorithm will be corresponds to $(n-L)_{th}$ iteration for updating current filter weight in place of recent most error. Therefore, the iterative equation of delayed LMS algorithm can be expressed as:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu\, \mathbf{x}(n-L)\; e(n-L) \tag{9}$$

The iterative LMS algorithm is required to be modified to include CORDIC processing element in the proposed architecture [19]. Let the tap weight w_k be

$$w_k = A_k \sin\theta_k \tag{10}$$

w_k satisfies, $-A_k \le w_k \le +A_k$, and maps uniquely to a θ_k in the interval $[-\pi/2, +\pi/2]$. The function $\sin\theta_k$ is a monotonically increasing continuous function of θ_k, as it lies within $[-\pi/2 \le \theta_k \le +\pi/2]$, i.e. $\partial\varepsilon/\partial\theta_k$ has the same sign as that of $\partial\varepsilon/\partial w_k$ everywhere within the hypercube resulting in local minima or maxima less MSE. Using the equation (10), we can show the variation of error w.r.t angle variation as follows:

$$\partial\varepsilon/\partial\theta_k = \partial\varepsilon/\partial w_k \,.\, A_k \cos\theta_k \tag{11}$$

$$\nabla_\theta \varepsilon = \Delta . \nabla_w \varepsilon \tag{12}$$

Here Δ is $N \times N$ diagonal matrix whose k-th term can be given by: $\Delta_{k.k} = A_k \cos\theta_k$, $k = 0,1,\ldots\ldots\ldots, N-1$

Using the equations (6) and (12), we get the following

$$\boldsymbol{\theta}(i+1) = \boldsymbol{\theta}(i) - \frac{\mu}{2} \nabla_\theta \varepsilon \,\big|_{\boldsymbol{\theta}=\boldsymbol{\theta}(i)} \tag{13}$$

where, $\nabla_\theta \varepsilon = -2\Delta(\,\mathbf{p\text{-}Rw})$

With replacing $\mathbf{p}$ and $\mathbf{R}$ by $\mathbf{x}(n)d(n)$ and $\mathbf{x}(n)\mathbf{x}^T(n)$ in equation (13), we get the trigonometric form of LMS algorithm as:

$$\boldsymbol{\theta}(n+1) = \boldsymbol{\theta}(n) + \mu\boldsymbol{\Delta}(n)\mathbf{x}(n)e(n) \tag{14}$$

$$e(n) = d(n) - \sum_{k=0}^{N-1} \sin\theta_k(n)\, x(n-k)$$

The above form of LMS algorithm can be realized using CORDIC processing element. The sine/cosine terms generated by CORDIC can be utilized for filtering as well as updating filter coefficients in the equalizer.

III. Review of CORDIC Algorithm and Its Design

A. *CORDIC Algorithm revisited*

The CORDIC algorithm deals with the decomposition of the desired rotation angle into the weighted sum of a set of predefined elementary rotation angles. Each of them can be accomplished with simple shift-add operation for a desired rotational angle θ. It can be represented for M iterations of an input vector $(x,y)^T$ setting initial conditions: $x_0 = x$, $y_0 = y$, and $z_0 = \theta$ as $z_f = \theta - \sum_{i=0}^{M-1} \delta_i \alpha_i$. If $z_f = 0$ holds, then

$$\theta = \sum_{i=0}^{M-1} \delta_i \alpha_i \tag{15}$$

i.e. the total accumulated rotation angle is equal to θ. δ_i, $0 \le i \le M-1$, denote a sequence of ±1s that determine the direction of each elementary rotation. When M is the total number of elementary rotation angles, i-th angle α_i is given by:

$$\alpha_{m,i} = \frac{1}{\sqrt{m}} \tan^{-1}[\sqrt{m}\, 2^{-s(m,i)}] = \begin{cases} 2^{-s(0,i)} \\ \tan^{-1} 2^{-s(1,i)} \\ \tanh^{-1} 2^{-s(-1,i)} \end{cases} \tag{16}$$

where $m = 0$, 1 and -1 correspond to the rotation operation in linear, circular, and hyperbolic coordinate system respectively. For a given value of θ, the CORDIC iteration is given by:

$$\begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} = \begin{bmatrix} 1 & -\delta_i 2^{-i} \\ \delta_i 2^{-i} & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} \tag{17}$$

and $z_{i+1} = z_i - \delta_i \alpha_i$

where $\alpha_i = \tan^{-1} 2^{-i}$. To bring a unit vector to desired angle θ, the CORDIC algorithm gives known recursive rotations to the vector. The known rotational values are shown in **Table. I** as a Pre-Computed angle. Once the vector is at desired angle, the outcome of the X and Y coordinates of the vector are equal to $Cos\theta$ and $Sin\theta$ respectively. Let a unit vector in iteration 'i' is rotated by some angle θ_i, then the recursively updated equations are generated in the following form:

$$x_{i+1} = x_i \cos\delta_i\alpha_i - y_i \sin\delta_i\alpha_i$$

$$y_{i+1} = y_i \cos\delta_i\alpha_i + x_i \sin\delta_i\alpha_i \tag{18}$$

The above equation can be simplified and written as

$$x_{i+1} = \cos\delta_i\alpha_i(x_i - y_i \tan\delta_i\alpha_i)$$

$$y_{i+1} = \cos\delta_i\alpha_i(y_i + x_i \tan\delta_i\alpha_i) \tag{19}$$

The $\tan\alpha_i = \pm 2^{-i}$. So multiplication is converted into an arithmetic right shift. Since cosine is an even function, therefore $\cos(\alpha) = \cos(-\alpha)$. The iterative equation (19) can be reduced to

$$x_{i+1} = K_i(x_i - y_i\delta_i 2^{-i})$$

$$y_{i+1} = K_i(y_i + x_i\delta_i 2^{-i}) \tag{20}$$

where $K_i = \cos\left(\arctan 2^{-i}\right) = 1\Big/\sqrt{(1+2^{-2i})}$ is known as scale factor for each iteration. If M iterations are performed, then scale factor, K, is defined as the multiplication of every K_i.

$$K = \prod_{i=0}^{M-1} K_i = \prod_{i=0}^{M-1} 1\Big/\sqrt{(1+2^{-2i})}\,. \qquad (21)$$

The elementary functions sine and cosine can be computed using the rotation mode of the CORDIC algorithm if the initial vector starts at $(|K|,0)$ with unit length. The final outputs of the CORDIC for the given input values $x_0 = 1, y_0 = 0$ and $z_0 = \theta$ are as follows:

$$x_f = K\cos\theta\ ,\ y_f = K\sin\theta \ \text{ and } \ z_f = 0\,. \qquad (22)$$

Since the scale factor is constant for a given number of rotations, $x_0 = 1/K$ can be set to get purely $\sin\theta$ and $\cos\theta$ values.

B. *Implementation of pipelined CORDIC:*

In Pipelined CORDIC architecture, a number of rotational modules are incorporated and each module is responsible for one elementary rotation. The modules are cascaded through intermediate latches (**Fig. 3**). Every stage within the pipelined CORDIC architecture, only adder/subtraction is used. The shift operations are hardwired using permanent oblique bus connections to perform multiplications by 2^{-i}. The pre-computed values of i-th iteration angle α_i required at each module can be generated through required sequence of binary data within V_{cc} and G_{nd} using trivial hardware arrangements.

The delay is adjusted by using proper bit-length in the shift register. In the design of CORDIC module, adders and subtractor contribute the critical path. Since carry propagation delay in full adders is the source of delay, Carry Save Adder (CSA) was the obvious choice for the design. The use of these adders reduces the stage delay significantly [19]. With the pipelining architecture, the propagation delay of the multiplier is the total delay of a single adder.

So ultimately the throughput of the architecture is increased significantly as the throughput is given by: "*1/(delay due to a single adder)*". If an iterative implementation of the CORDIC is used, the processor would take several clock cycles to give output for a given input. But in the pipelined architecture, each pipeline stage takes exactly one clock cycle to pass one output. Again, to keep rotation angles within $-\pi/2 \le \theta_i \le +\pi/2$, the transformation of angles in different quadrant complementing corresponding MSBs has been illustrated in **Fig. 4**. A detailed study of various errors in pipelined CORDIC have been discussed in Mandal et al. [20]. The total error generated in the CORDIC block which uses 12 bits in its fractional part. Therefore, the upper limit of the total quantization error can be taken as 2^{-12} with considering scale factor K and iteration number n, the equation can be given by:

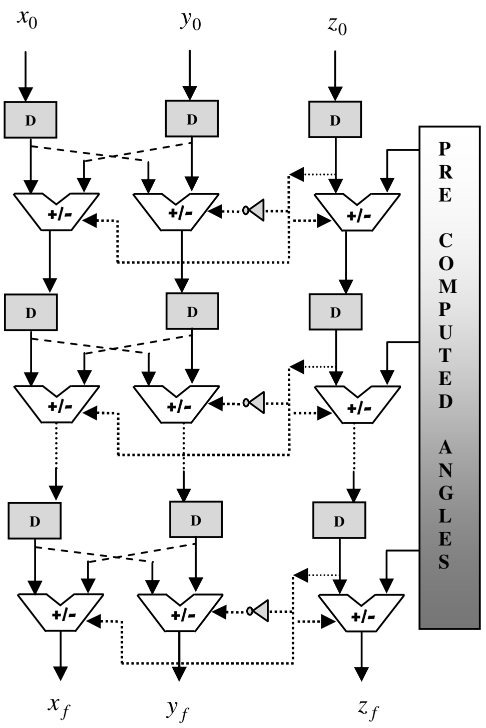

Fig. 3. The pipelined CORDIC

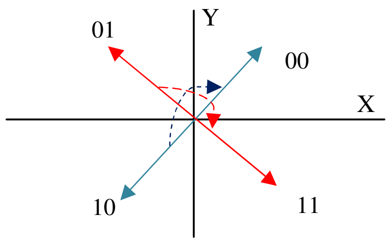

Fig. 4 Quadrant transformation for CORDIC operation

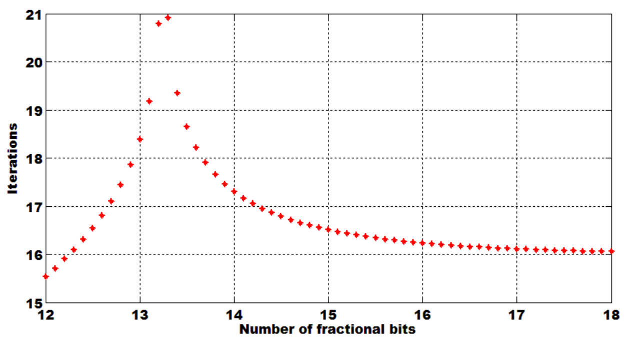

Fig. 5 Word-length in fractional bits Vs number of iterations

TABLE I
COMPARISON WITH DIFFERENT CORDIC ARCHITECTURES BASED ON HARDWARE SYNTHESIS [21]

Designs	Area (sq. um)	Clock (ns)	TPT (per µs)	Latency (ns)	ACT (ns)	ADP
Conventional CORDIC	2987	2.52	36.07	27.72	27.72	82799
Fixed rotation CORDIC	2471	2.49	40.16	24.90	24.90	61527
Interleaved scaling	2674	2.56	55.8	17.92	17.92	47918
Separated scaling	4074	2.29	109.17	16.03	9.16	37317
Bi-rotation cascade	5819	2.21	226.24	15.47	4.42	25719
Proposed design	**3142**	**1.98**	**505.05**	**12.31**	**1.98**	**38678**

TPT stands for throughput, ACT stands for average computation time and ADP stand for area-delay product.

$$\frac{1}{2^{n-1}} * |v^*| + K * \sqrt{2} * 2^{-b}\left(1 + \sum_{j=0}^{n-1}\prod_{i=j}^{n-1}\sqrt{(1+2^{-2i})}\right) + 2^{-b}$$

$$\leq 2^{-12} \qquad (23)$$

The above inequality is simulated in MATLAB to find out fractional bits of the internal word length of the CORDIC. The **Fig. 5** shows that when number of bits less than 14, the response of the CORDIC becomes complex and when bit number exceeds 14, the design works well. In view of the internal addition and subsequent error performance, 16 bit design has been preferred as fractional bits become almost independent of number of iterations at ≥ 16.

The time complexities of the proposed design and other various types of CORDIC architectures which have been published in recent times are compared. Unlike the proposed design, the conventional and few other designs available in the literatures uses adders, registers, barrel shifters and MUX which contribute toward hardware complexity. It has been seen that shifting operations using barrel-shifter for maximum of S shifts for the word-length can be implemented by $\lceil \log_2(S+1) \rceil$ stages of 2:1 MUX. Few designs have used ROM memory to store *arctan* angles as well as control signals for CORDIC operations. Therefore, the hardware complexity of the design also increases with increase in required word-length which leads to maximum number of shift operations for the design. We have compared our design with the other published work [21] in respect of clock period, throughput, latency and area which have been shown in **Table I** and **Table II** using TSMC 90-nm library.

TABLE II
TIME COMPLEXITY OF DIFFERENT CORDIC ARCHITECTURES [21]

Conventional CORDIC	$T_A + T_{FF} + 5T_{MX}$
Fixed rotation CORDIC	$T_A + T_{FF} + 5T_{MX}$
Interleaved scaling	$T_A + T_{FF} + 5T_{MX}$
Separated scaling	$T_A + T_{FF} + 4T_{MX}$
Bi-rotation cascade	$T_A + T_{FF} + 2T_{MX}$
Proposed design	T_A

T_A, T_{FF} and T_{MX} stand for addition time, Flip-Flop delay and delays in 2:1 MUX respectively.

IV. ARCHITECTURE OF ADFE WITH CORDIC

The adaptive decision feedback equalizer consists of feed forward filter (FFF), feedback filter (FBF), decision device and error computation unit. A simple block diagram of ADFE has been shown in **Fig.** 6.

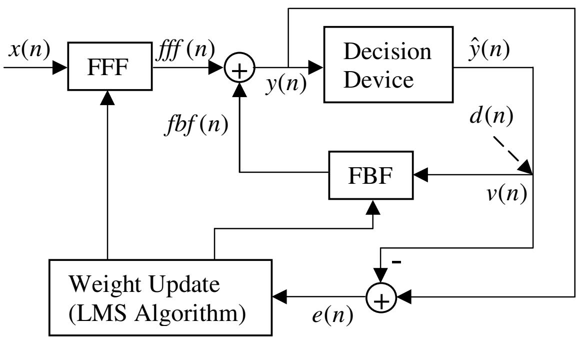

Fig. 6 Block diagram of a basic ADFE

A signal $x(n)$ arrives at the input to FFF filter. A decision device has been incorporated to give a decision based on $y(n)$ which is nothing but a sum of output of both feed forward and feedback filter. The decision result which is made on the input signal already detected previously is passed to FBF as an input. The error generated from the $y(n)$, $\hat{y}(n)$ and $d(n)$ is remain responsible for updating weights of the filters are being used in the design. The FFF in our design is a linear transversal filter which is implemented with a finite duration impulse response (FIR) filter with adjustable coefficients. The decisions made on the equalizer signal are fedback via a second transversal filter [22]. The elaborate internal design of proposed ADFE can be seen in **Fig. 7** which shows the error computation and weight updating phenomena through CORDIC based FFF and trivial multiplier based FBF filter respectively. If the detected symbols are stationary, then the ISI contributed by these symbols can be cancelled exactly by subtracting past symbol values with appropriate weight from the equalizer output. But channel response may not be same always. Therefore, the forward and feedback filter should adjust its coefficients simultaneously to counter the variation in channel response by minimizing the mean square error. At initial operation, the coefficients of the equalizer are needed to

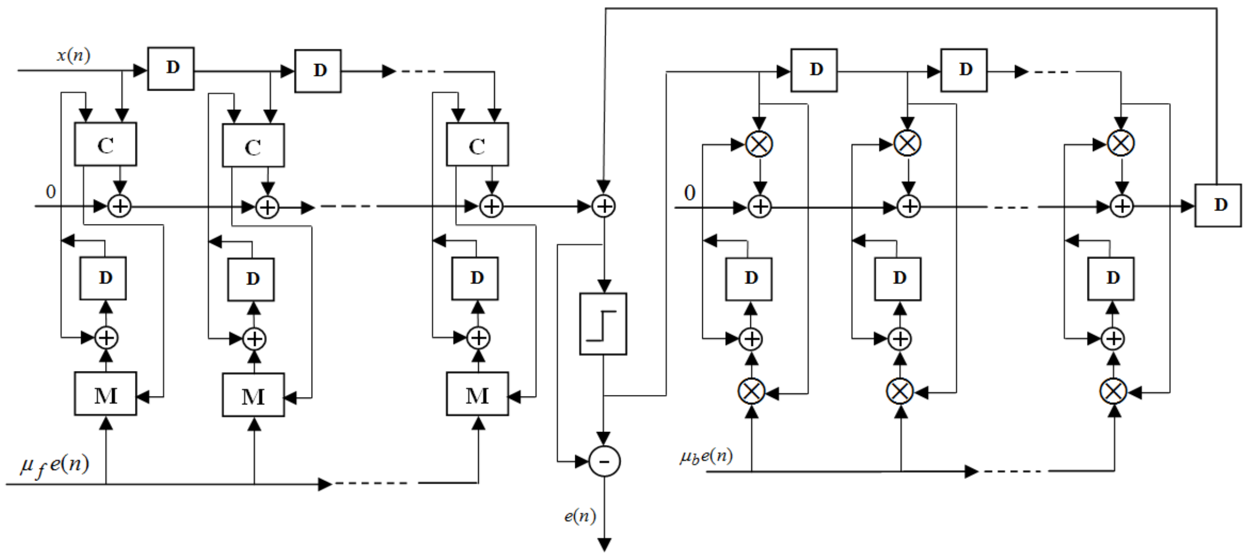

Fig. 7. Adaptive decision feedback equalizer using FFF and FBF filter

be adjusted as per the channel response and therefore, a known sequence of symbols named as training sequence is transmitted through the desired channel for this purpose. The design uses pipelined CORDIC unit as a main processing element. In circular rotation mode, CORDIC unit generates $\sin\theta$ and $\cos\theta$ which are the main ingredients for filtering and weight updating respectively in the proposed design. The pipelined technique has been adopted throughout the architecture to implement delayed version of trigonometric form of LMS algorithm in FFF and simple delayed LMS (DLMS) for FBF. The reason behind that is the inputs to the FBF are the transmitted symbols whose bit lengths are very small and hardware required for multiplication is trivial. But it is not the case for FFF filter. Adaptation delay in FFF naturally more as compared to FBF. μ_f and μ_b are the step size used to compute the weight update term of FFF and FBF respectively to synchronize the error output.

Modified Booth encoding (MBE) algorithm has been widely used in implementation of multiplier due to its efficiency in reduction of partial products by half. The presented multiplication operations are performed using three steps as shown in **Fig 8 (a)**. All the partial products are generated in first step. For this purpose, popular MBE scheme has been employed and has not been discussed in detail in this design. The second step deals with reduction of all partial products in parallel structure using 4-to-2 adder. The third step uses fast adder named Carry Look-Ahead (CLA) to add the two numbers obtained from second stage to generate final product [23]. The reduction of partial products using 4-to-2 adder has been shown in **Fig. 8 (b)**. There are four 4-to-2 adder have been used to accommodate 16 partial product

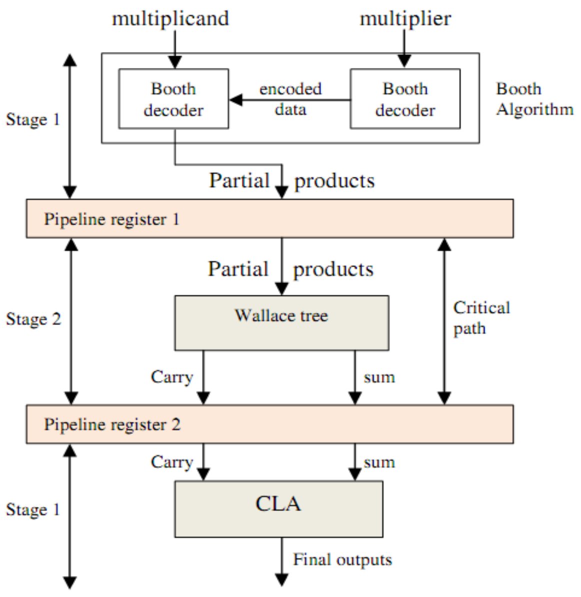

Fig. 8 (a) 3-stage pipelined multiplier

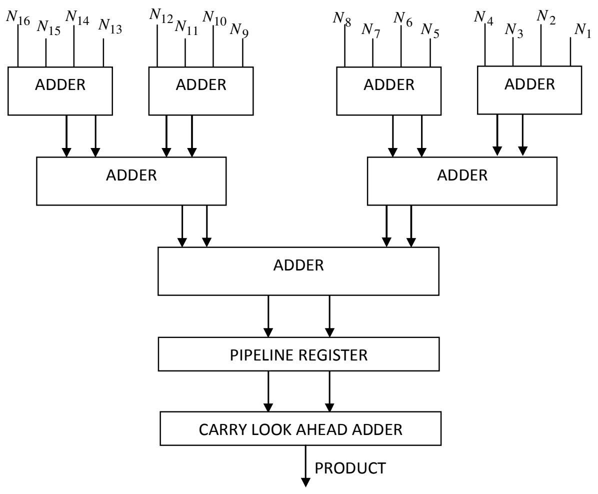

Fig 8(b). Partial product reduction scheme in association with final fast adder generated from the input coming from CORDIC as well as error generator circuits. Each 4-to-2 adder receives four inputs and gives two outputs in terms of sum and carry. The adder cell consists of carry generator which receives three inputs from previous adder cell and a parity generator which generates a control signal for the two MUX to be selected.

The multiplier has been synthesized on Xilinx 13.4 using target device Spartan 3E xc3s250e-5-pq208. The resource utilization summary has been shown in **Table III.** The table shows the area efficiency of the design.

TABLE III
DEVICE UTILIZATION BY MULTIPLIER

Number of Slices:	45	out of	2448	1%
Number of Slice Flip Flops:	39	out of	4896	0%
Number of 4 input LUTs:	80	out of	4896	1%
Number used as logic:	63			
Number used as Shift registers:	17			
Number of IOs:	36			
Number of bonded IOBs:	36	out of	158	22%
Number of GCLKs:	1	out of	24	4%

V. PERFORMANCE ANALYSIS AND DISCUSSION

It is well known that pipelined CORDIC itself has got good convergence property which has been efficiently used in this application. Exact convergence analysis of proposed architecture using trigonometric LMS algorithm is not possible due to presence of nonlinearity of the filter as well as angle update equation. Using the iteration error generated in the equalizer, approximate convergence studies have been carried out.

The input sequence to the channel is considered to be QPSK data source. The generated signal is transmitted over a communication channel. The channel is complex and infected by noise which is additive and white Gaussian (AWGN). During the journey through the channel, the symbols are get corrupted due to various channel imperfection. A standard constellation has been utilized for training sequence for the equalizer. The equalizer is used to minimize the error at the receiver end by adjusting the filter weights in such a way to recover the constellation. The simulation environment has been created by utilizing total data 3000, training symbols 2000 and corresponding signal-to-noise ratio 30 dB. A 15-tap adaptive filter with centre placed at 8-th tap has been taken for simulation studies. The equalizer delay is set at 8. The coherent form of equalization has been considered where channel estimation is carried out to determine the amplitude as well as phase distortion created by channel imperfections or ISI. The tap weights are initialized to zero. The step size which is the guiding force for fast or slow conversion of the system is appropriately taken care. If $\mu \leq 0.001$ the convergent speed of the algorithm will be very slow and too large μ can cause instability. MATLAB Simulation for equalization on the received corrupted symbols and corresponding convergence performance of ADFE in terms of learning curves of the design at $\mu = 0.001$ and $\mu = 0.014$ has been shown in **Fig. 9** and **Fig 10** respectively.

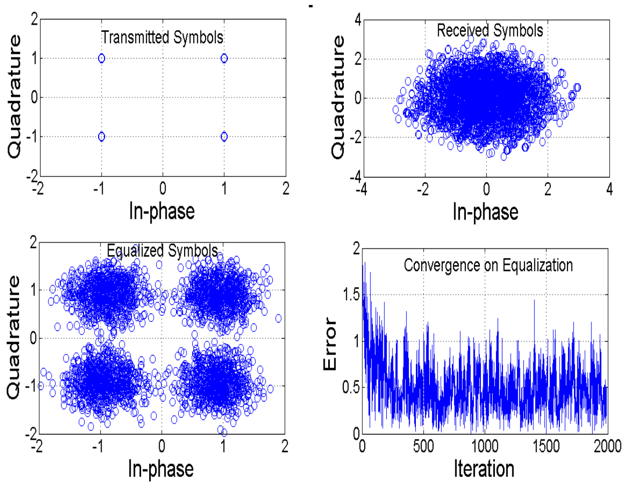

Fig. 9 Equalization on channel imperfection at μ=0.001

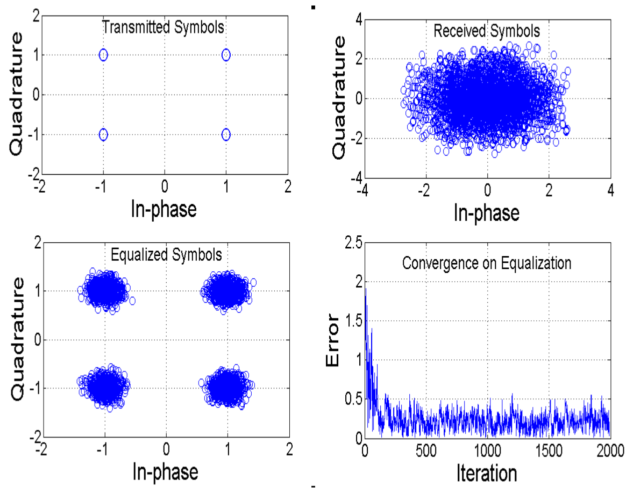

Fig. 10. Equalization on channel imperfection at μ=0.014

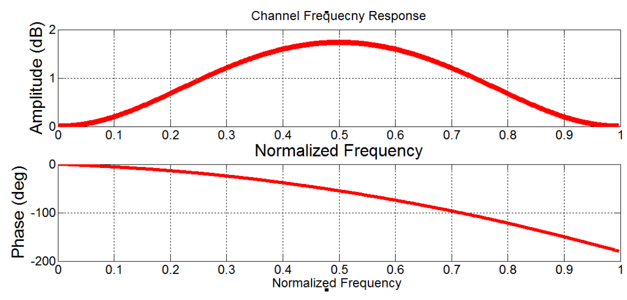

Fig. 11 (a) The response of mismatched channel

The proposed design has been tested using randomly generated signal as an input for the equalizer which is to be used in mismatched channel. The equalization of mismatched channel has been shown in **Fig.11 (a)** whereas, **Fig.11 (b)** shows the equalizer response which is essentially inverse of the given channel response.

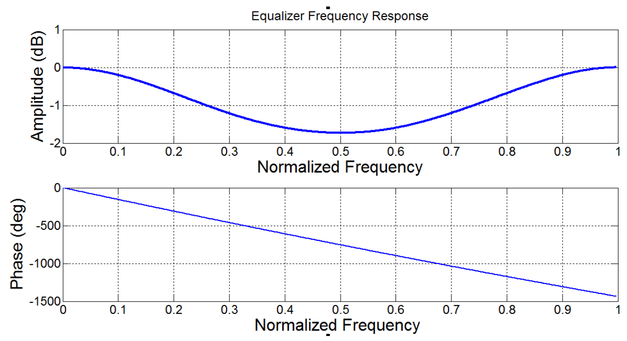

Fig. 11 (b) The equalizer response of the mismatched channel

VI. CONCLUSION

This paper presents realization of CORDIC based adaptive feedback equalizer using trigonometric form of LMS algorithm in mitigating severe Inter Symbol Interference (ISI) in AWGN channel. Based on the trigonometric reformulation of LMS algorithm, low-complexity ADFE architecture is designed. 50% of multipliers have been replaced in FFF by CORDIC unit to facilitate in easy implementation for weight updating and filtering purpose. To verify the validity of the proposed architecture, extensive MATLAB simulations have been carried out and the results shows the suitability of the design in respect of high speed channel equalization.

REFERENCES

[1] R. Mishra and A. Mandal, "Coordinate rotation algorithm based non-linear Adaptive Decision Feedback Equalizer", In *IEEE 2012 9th International Multi-Conference on Systems, Signals and Devices (SSD),* pp. 1-5, 2012.

[2] L. Fan, C. He, D. Wang and L. Jiang, "Efficient robust adaptive decision feedback equalizer for large delay sparse channel", *IEEE Transactions on Consumer Electronics,* vol. *51, no.* 2, pp. 449-456, 2005.

[3] J. E. Volder, "The CORDIC Trigonometric Computing Technique", *IRE Transactions on Electronic Computing*, vol. EC-8, pp. 330-334, Sept, 1959.

[4] Y. H. Hu, "CORDIC-based VLSI architectures for digital signal processing." *IEEE Signal Processing Magazine,* vol. 9, no. 3, pp. 16-35, 1992.

[5] P. K. Meher, et al., "50 years of CORDIC: Algorithms, architectures, and application*s", IEEE Transactions on Circuits and Systems I: Regular Papers,* vol. 56, No. 9, pp. 1893-1907, 2009.

[6] Y. H. Hu, "On the convergence of the CORDIC adaptive lattice filtering (CALF) algorithm." *IEEE Transactions on Signal Processing,* vol. 46, no. 7, pp. 1861-1871, 1998.

[7] M. D. Yang, A. Y. Wu and J. T. Lai, "Fast convergent pipelined adaptive DFE architecture using post-cursor processing filter technique", *IEEE Transactions on Circuits and Systems II: Express Briefs,* , vol. *51, no.* 2, pp. 57-60, 2004.

[8] M. Magarini, L. Barletta and A. Spalvieri, "Efficient computation of the feedback filter for the hybrid decision feedback equalizer in highly dispersive channels". *IEEE Transactions on Wireless Communications,* vol. *11, no.* 6, pp. 2245-2253, 2012.

[9] A. M. Chan, and G. W. Wornell, "A class of block-iterative equalizers for intersymbol interference channels: Fixed channel results", *IEEE Transactions on Communications,* vol. *49, no.* 11, pp. 1966-1976, 2001.

[10] Y. C. Lin, S. J. Jou and M. T. Shiue, "High throughput concurrent lookahead adaptive decision feedback equalizer". *IET circuits, devices & systems*, vol. *6, no.* 1, pp. 52-62, 2012.

[11] R. Lopez-Valcarce, "Realizable linear and decision feedback equalizers: properties and connections", *IEEE Transactions on Signal Processing,* vol. *52, no.* 3, pp. 757-773, 2004.

[12] Á. Knapp and L. Pap, "General Performance Analysis of Binary Fading Channels with Measurement Based Feedback Channel Equalization", *Infocommunications Journal*, vol. VI, no. 1, pp. 1-9, 2014.

[13] A. Wu, C. K. Ng, and K. C. Tang. "Modified Booth pipelined multiplication."*Electronics Letters*, vol. 34, no. 12, pp. 1179-1180, 1998.

[14] F. Elguibaly "A fast parallel multiplier-accumulator using the modified Booth algorithm." *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing,* vol. 47, no. 9, pp. 902-908, 2000.

[15] R. S. Waters and E. E. Swartzlander, "A reduced complexity Wallace multiplier reduction," IEEE Transactions on Computers, vol. 59, no. 8, pp. 1134–1137, 2010.

[16] D. Galbi, et al. *U.S. Patent No. 4,901,270*, 1990, Washington, DC: U.S. Patent and Trademark Office.

[17] S.-R. Kuang, J.-P. Wang and C.-Y. Guo, "Modified booth multipliers with a regular partial product array," IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 56, no. 5, pp. 404–408, 2009.

[18] Y. Hu, "The Quantization Effects of the CORDIC Algorithm", *IEEE Trans. on Signal Processing*, vol. 40, No. 4, pp. 834-844, 1992.

[19] M. Chakraborty, A. S. Dhar and S. Pervin, "CORDIC realization of the transversal adaptive filter using a trigonometric LMS algorithm", In *IEEE International Conference on Acoustics, Speech, and Signal Processing,* Vol. 2, pp. 1225-1228, 2001.

[20] A. Mandal and R. Mishra, "FPGA Implementation of Pipelined CORDIC for Digital Demodulation in FMCW Radar", *Infocommunications Journal,* 2013, vol. V, no. 2, p. 17-23.

[21] P. K. Meher and S. Y. Park, "CORDIC designs for fixed angle of rotation", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems,* vol. *21, no.* 2, pp. 217-228, 2013.

[22] N. R. Shanbhag and K. K. Parhi, "Pipelined adaptive DFE architectures using relaxed look-ahead" *IEEE Transactions on Signal Processing, vol. 43, no.* 6, pp. 1368-1385, 1995.

[23] A. Mandal, R. Mishra, B. K. Kaushik and N. Z. Rizvi, "Design of LMS Adaptive Radar Detector for Non-homogeneous Interferences", *IETE Technical Review*, vol. 32, 2015. DOI: 10.1080/02564602.2015.1093436.

Amritakar Mandal received his M.Tech with specialization in VLSI from Shobhit University, India in 2009. He is currently working toward the Ph.D degree in the School of Information and Communication Technology, Gautam Buddha University, India. He has vast experience in Electronic Warfare (EW) and Surveillance Radar systems. His research interests include Radar Signal Processing and High Speed VLSI Design for Communication Systems.

Rajesh Mishra is currently working as Assistant Professor in School of Information and Communication Technology, Gautam Buddha University Greater Noida, Delhi NCR (India). He received his BE (Electronics Eng.), M. Tech and Ph. D. degree (Reliability Eng.) from Reliability Engineering Centre, IIT Kharagpur (India) in year 2000, 2004, and 2009 respectively. He has research interest in the area of reliability engineering, layout design for capacitated networks, and network optimization. He has published papers in several international journals of repute.