Original scientific paper
https://doi.org/10.33180/InfMIDEM2019.406

Informacije MIDEM
Journal of Microelectronics,
Electronic Components and Materials
Vol. 49, No. 4(2019), 241 – 254

Design of an Optimized Twin Mode Reconfigurable Adaptive FIR Filter Architecture for Speech Signal Processing

Padmapriya S.[1], Jagadeeswari M.[1], Lakshmi Prabha V.[2]

[1]Sri Ramakrishna Engineering College, Department of ECE, Coimbatore, Tamilnadu, India
[2]PSG College of Technology, Department of ECE, Coimbatore, Tamilnadu, India

Abstract: Reconfigurability, low complexity and low power are the key requirements of FIR filters employed in multi-standard wireless communication systems. Digital Filters are used to filter the audio data stream and increase the reliability of speech signal. Therefore, it is imperative to design an area optimized and low power based reconfigurable FIR filter architectures.
The reconfigurable architecture designed in this research is capable of achieving lower adaptation-delay and area-delay-power efficient implementation of a Delayed Least Mean Square (DLMS) adaptive filter with reversible logic gates. The Optimized Adaptive Reconfigurable (OAR) FIR filter architectures are proposed. The optimized architectures are implemented across the combinational blocks by reducing the pipeline delays, sampling period, energy consumption and area, to increase the Power-Delay Product (PDP) and Energy Per Sample (EPS).The noisy speech signals are used for verifying the efficiency of the proposed architectures. By implementing the proposed scheme in signal corrupted by various real-time noises at different Signal to Noise Ratios (SNRs), the efficiency of the architecture is verified.

Keywords: Adaptive Filter; Least Mean Square Algorithm; Reconfigurable Filtering; Speech Signal Processing

Dizajn optimiranega nastavljivega FIR filtra v za procesiranje govornega signala

Izvleček: Nastavljivost, enostavnost in nizka poraba so glavne zahteve FIR filtrov v multistandardnih sistemih brezžične komunikacije. Digitalni filtri se uorabljajo za filtriranje zvokovnega prenosa in izboljšanje zanesljivosti govora v signal. Zato je potrebno načrtovati majhne filtre z nizko porabo energije. Predlagana arhitektura je sposobna dosegati nizko porabo in majhno uporabo prostora za DLMS filter z vrati z reverzno logiko. Optimizirana arhitektura je implementirana v bloke za zmanjševanje zakasnitev, period vzorčenja in porabe energije ter prostora in povečanja PDP in EPS. Za verifikacijo so bili uporabljeni številni vzorci govora z veliko šuma v realnem času.

Ključne besede: adaptiven filter; algoritem najmanjših povprečnih kvadratov; procesiranje govornega signala

** Corresponding Author's e-mail: padmapriya.s@srec.ac.in*

1 Introduction

FIR digital filters are widely used as a main tool in various signal processing and image processing applications [1]. Most portable electronic devices such as cellular phones, personal digital assistants, and hearing aids require DSP for high performance. The miniaturization of handheld devices with good performance increases the demand for sophisticated DSP algorithm implementations that are area optimized and consume as little power as possible.

An important branch of signal and information processing is Adaptive Signal Processing. The relationship between two signals is modeled in an iterative manner with an adaptive filter in real time. A filter can be realized either as a set of program instructions running on

an arithmetic processing device such as a microprocessor or a DSP chip, or in a semi-custom or custom VLSI integrated circuit. The fundamental operation of an adaptive filter depends on the specific physical realization that it takes. Adaptive signal processing is rapidly developing, and widely emphasized by scholars at home and abroad. It expands the application range of digital signal processing [2].

Least mean square error (LMS) algorithm, proposed by Widrow in 1985 [18], is a key algorithm for adaptive signal processing. It is widely applied because of characteristics, including good stability, computational efficiency and easy implementation.

Ting et al.[3] have proposed a fine-grained pipelined design to limit the critical path to the maximum of one addition time, which supports high sampling frequency. But it involves a lot of area overhead for pipelining and higher power consumption than the systolic architecture for Delayed Least Mean Square Adaptive Digital Filter (DLMS ADF) proposed by Van &Feng [4].

To reduce the number of adaptation delays, Meher&Maheshwari [5] have proposed a 2-bit multiplication cell, and used that with an efficient adder tree for pipelined inner-product computation to minimize the critical path and area without increasing the number of adaptation delays. For achieving lower adaptation-delay and area-delay-power efficient implementation, a novel partial product generator and an optimization of previously reported design have been proposed by Park &Meher [6]. It is found to be more efficient in terms of the Power Delay product (PDP) and Energy Per Sample (EPS).

In the above works, the critical-path analysis and necessary design considerations are not taken into account. The designs of Meher& Park[7, 8] still consume higher area, which could be substantially reduced. A critical-path analysis of the LMS adaptive filter, conditional signed carry-save accumulation to reduce the sampling period, and area complexity and low power consumption by fast bit clock for carry-save accumulation were presented by Meher& Park [8]. An architecture for the LMS adaptive filter with minimal use of pipeline stages was derived, which results in less area complexity and power consumption without compromising the desired processing throughput. Fixed-point implementation of LMS adaptive filter was proposed by Meher & Park [9], which uses an efficient implementation of multiplications and Computation sharing inner-product makes a novel design of Partial Product Generator (PPG) Block.

The Block Least Mean Square (BLMS) ADF proposed by Clark et al. [10] and Baghel&Shaik[11] is one of the useful derivatives of the LMS ADF for fast and computationally-efficient implementation of ADFs. BLMS ADF accepts a block of input for computing a block of output and updates the weights using a block of errors in every training cycle.

Baghel&Shaik [12] had suggested a Distributed-Arithmetic (DA)-based structure for FPGA implementation of BLMS ADFs. A low complexity design using a single Multiply-ACcumulate (MAC) cell for the computation of filter output and weight-increment term supporting a low sampling rate has been proposed by Jayashri et al.[13] for BLMS ADFs.

The throughput of the DA-LMS ADFs could be slow for real-time applications due to the bit-serial nature of DA computation. The scheme offers the sharing of Look Up Table (LUT) for the computation of both filter output and weight-increment term, but this scheme cannot be applied to derive a DA-based structure for BLMS ADFs, because separate Inner-Product Computation (IPC) is performed for calculation of filter output and weight-increment term of BLMS ADF. In LMS ADF, IPC is performed to calculate the filter output. Mohanty&Meher [14, 15] derived a DA formulation of BLMS algorithm where both convolution and correlation are performed using a common LUT for the computation of filter outputs and weight increment terms. LUT words and adders are significantly saved, which constitute the major hardware components in DA-based computing structures. A parallel architecture for the implementation of DA-based BLMSADF is derived.

Mohanty et al [16] proposed an architecture, scalable for higher filter lengths and block sizes based on DA. The maximal sharing of parallel LUT Update operation and LUT contents was also proposed. However, the structure complexity increases with filter length and block size.

Tasleem Khan & Shaik [17] proposed pipelined DA optimal-complexity structures based least-mean-square (LMS) adaptive filter. Offset-Binary-Coding (OBC) combinations of input samples was implemented to reduce the complexity of proposed structures on hardware. A novel low-complexity implementations for the offset term, weight update block and shift-accumulate unit are also proposed.

2 Adaptive filtering and principle

2.1 Adaptive filter

2.1.1 Adaptive filtering principle

The FIR Filter weights are updated by the Widrow-Hoff Least Mean Square (LMS) algorithm proposed by Widrow&Stearns[18] due to its low complexity, stability

and satisfactory convergence performance analyzed by Haykin&Widrow [19]. The output signal is compared to a second signal d_n, called the desired response signal, by subtracting the two samples at time n. The weights of the LMS adaptive filter during n^{th} iteration are updated according to the following equations.

$$W_{n+1} = W_n + \mu e_n X_n \quad (1)$$

where μ is the step size,

$$e_n = d_n - y_n \quad (2)$$

$$y_n = W_n^T X_n \quad (3)$$

With the input vector X_n and weight vector W_n at the n^{th} iteration given by

$$X_n = [x_n, x_{n-1}, \ldots\ldots x_{n-N+1}]^T$$
$$W_n = [w_n(0), w_n(1), \ldots\ldots, w_n(N-1)]^T$$

Where d_n is the desired response, y_n is the filter output and e_n denotes the error signal. The error signal is fed into a procedure which alters or adapts the parameters of the filter from time *n* to time (*n*+1) in a well-defined manner. As the time index n is incremented, it is hoped that the output of the adaptive filter becomes a better and better match to the desired response signal through the adaptation process, such that the magnitude of e_n decreases over time. In the adaptive filtering task, adaptation refers to the method by which the parameters of the system are changed from time index n to time index (*n*+1). The number and types of parameters within this system depend on the computational structure chosen for the system.The error e_n becomes available after *m* cycles, where *m* is called the adaptation-delay for pipelined designs with *m* pipeline stages. For an N^{th} -order FIR filter, the generation of each output sample y_n takes *N*+1 Multiply-Accumulate (MAC) operations.

2.1.2 Adaptive filtering with minimal delay and pipelining
In the direct form LMS adaptive filter the critical-path for computing the inner-product (in order to obtain the filter output) is long. Pipelined implementation is required to reduce the critical-path when the desired sample period is exceeded. Due to the recursive structure of the conventional LMS algorithm, pipelining is not supported. Meyer &Agrawal [20] ,Long et al. [21] proposed Delayed LMS (DLMS) algorithm, which supports pipelined implementation. In DLMS, the correction terms for updating the filter weights of the current iteration are calculated from the error corresponding to the past iteration. Many architectures were proposed by Ramanathan&Visvanathan [22], Van &Feng [4],Yi et al. [23] and Ting et al. [3] to reduce the adaptation delay but the area overhead and high-power consumptions were the trade-offs in all the designs. To overcome the drawbacks of the above methods, a 2-bit multiplication cell was proposed by Meher&Maheshwari [5] and implemented in an adder-tree for pipelined inner-product computation.

2.1.3 Drawbacks of the pipelined adaptive filters
The algorithms based on pipelined LMS have high computational complexity. For higher-order filters, the adaptation delay increases. There exists a trade-off between the adaptation delay, area, and power consumption. However, it occupies an area without increasing the number of adaptation-delays. Hence, Area Optimized and Low Power Adaptive Reconfigurable FIR filter architecture has been proposed in this research. In the proposed adaptive architecture, thresholding is performed to support the reconfiguration in the existing adaptive designs. The proposed adaptive reconfigurable architectures are simple to design and suitable for filtering speech signals.

The Proposed Optimized Adaptive Reconfiguration (OAR) FIR Filter consumes less power, the area overhead is reduced, and the adaptation delay is decreased. Thresholding is done with the help of the adaptive filter coefficient values.

3 Delayed least mean square (DLMS) algorithm

Pipeline implementation is not favored when the sampling rate is high, due to the delay in availability of the feedback error for updating the weights according to the LMS algorithm. Cohen et al.[24] and Parhi [25] have proposed the DLMS algorithm for pipeline implementation. In DLMS, the error of the past iteration is used for calculating the correct terms for updating the filter weights of the current iteration.

3.1 Implementation of direct-form DLMS algorithm:

The error computation path is implemented in m pipelined stages, the latency of error computation is m cycles. The error computed by the structure at the n^{th} cycle is e_{n-m}. The error is used with the input samples delayed by *m* cycles to generate the weight-increment term. The weight-update equation of the DLMS algorithm is given by

$$W_{n+1} = W_n + \mu e_{n-m} X_{n-m} \quad (4)$$

Where $e_{n-m} = d_{n-m} - y_{n-m}$ and $y_n = W_n^T X_n$

A generalized block diagram of direct form DLMS adaptive filter is shown in Fig. 1. It consists of an error-computation block as shown in Fig. 2 and a weight-update block as shown in Fig. 3. The number of delays m shown in Figure 1 corresponds to the pipeline delays introduced due to pipelining of the error computation block.

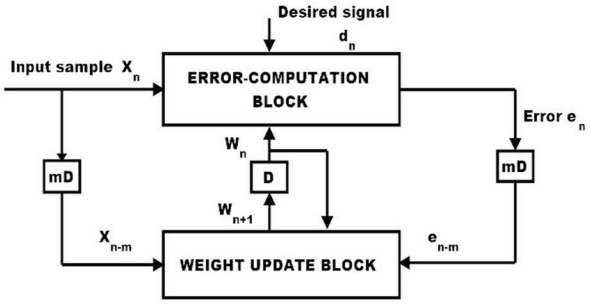


Figure 1: Generalized block diagram of Direct-form DLMS Adaptive Filter

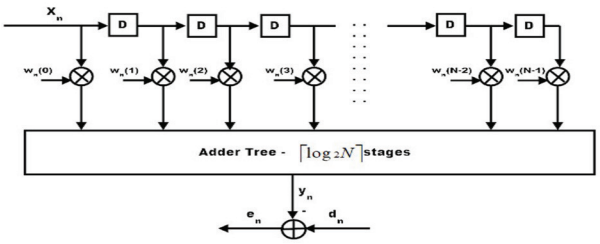


Figure 2: Error Computation block of Direct-form DLMS Adaptive Filter

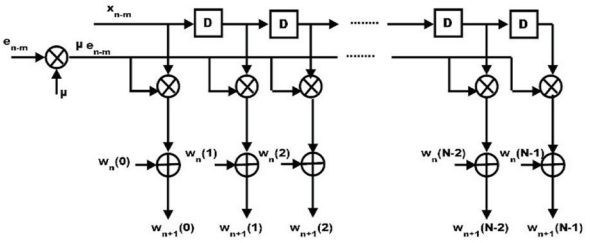


Figure 3: Weight-update block of Direct-form DLMS Adaptive Filter

3.2 Structure of error computation block

The structure of Error-Computation Unit of N-tap DLMS adaptive filter is shown in Fig.4. It consists of *N* 2-bit Partial Product Generators (PPG) corresponding to *N* multipliers, a cluster of *L/2* binary adder-trees followed by a single shift add tree.

3.3 Structure of partial product generator

The structure of each PPG is shown in Fig. 5. It involves L/2 2-to-3 decoders and the same number of AND-OR

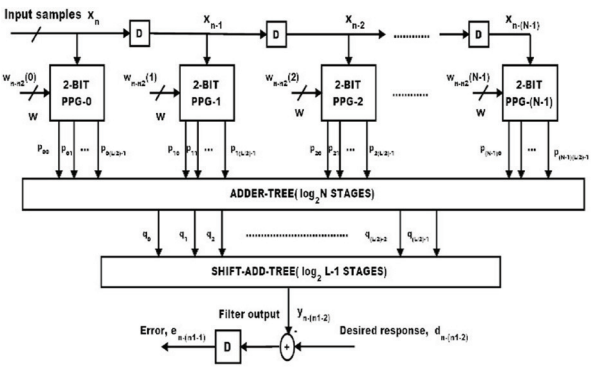


Figure 4: Structure of Error-Computation Block

Cells(AOC). Input is taken as even word-length. Each of the 2-to-3 decoders takes a 2-bit digit (u_1, u_0) as input and produces three outputs $b_0 = u_0.u_1$, $b_1 = \overline{u}_0.u_1$, and $b_2 = u_0.u_1$, such that $b_0 = 1$ for $(u_1u_0) = 1$, $b_1 = 1$ for $(u_1u_0) = 2$, and $b_2 = 1$ for $(u_1u_0) = 3$. The decoder outputs b_0, b_1 and b_2 along with *w*, 2*w* and 3*w* are fed to an AOC, where *w*, 2*w* and 3*w* are in 2's complement representation and sign-extended to have (*w*+2) bits each. While computing the partial product corresponding to the Most-Significant Digit (MSD), i.e., ($u_{L-1}u_{L-2}$) of the input sample, the AOC-(*L*/2-1) is fed with *w*, -2*w* and -*w* as input while considering the sign of the input.

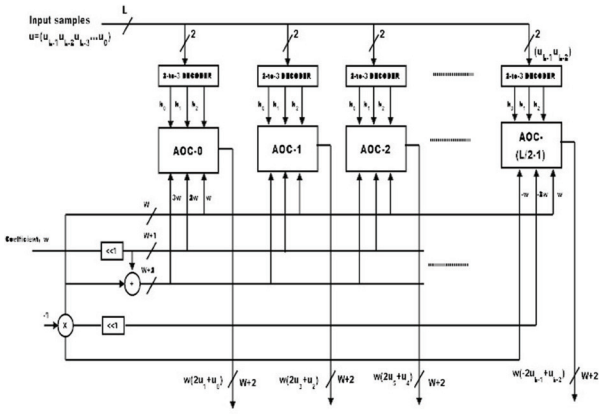


Figure 5: Structure of Partial Product Generator (PPG)

3.4 Structure of AND-OR cells

The structure and function of AOC is depicted in Fig.6. Each AOC consists of 3 AND cells and 2 OR cells. The structure and function of AND cells and OR cells are depicted by Fig. 6(b) and 6(c) respectively. Each AND cell takes *n*-bit input D and a single bit input *b*, and consists of *n* AND gates. It distributes all *n*-bits of input D to its *n* AND gates as one of the inputs. The other inputs of all the *n* AND gates are fed with the single-bit input *b*. Each OR cell similarly takes a pair of *n*-bit input words,

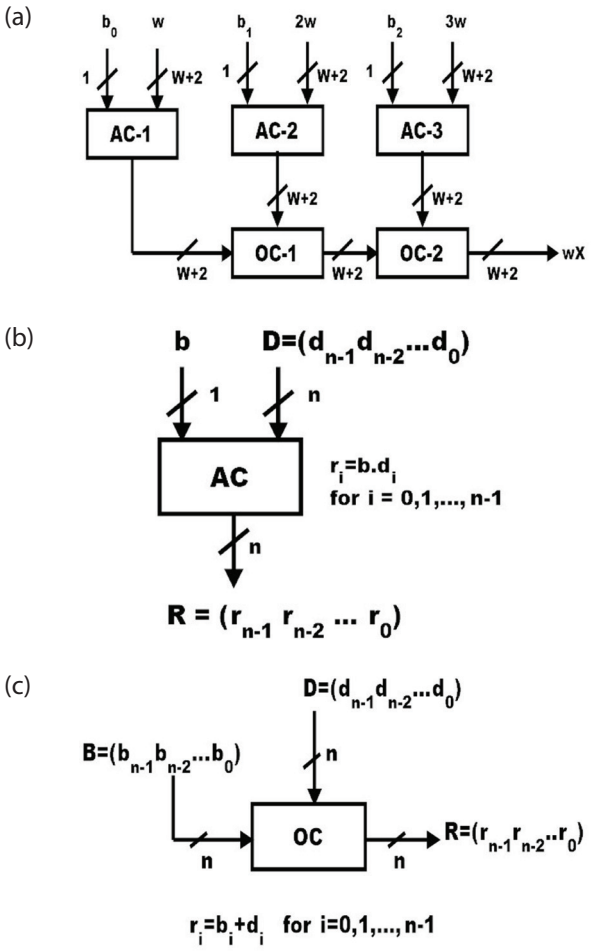


Figure 6: (a) Structure of AND/OR cell, (b) AND gate and (c) OR gate

and has *n* OR gates. A pair of bits in the same bit-position in B and D is fed to the same OR gate.

The output of an AOC is *w*, 2*w* and 3*w* corresponding to the decimal values 1, 2 and 3 of the 2-bit input (u_1u_0) respectively. The decoder along with the AOC performs a multiplication of input operand w with two-bit digit (u_1u_0), such that the PPG of Fig. 5 performs *L*/2 parallel multiplications of input word *w* with a 2-bit digit to produce *L*/2 partial products of the product word *wu*.

3.5 Proposed latched carry select adder (LCSA)

Carry Select Adder is used in the proposed reconfigurable FIR filter since it is the fast adder. D-latch with enable signal is introduced in the design. Latches are used to store one bit information. As long as the enable signal is asserted, the outputs are affected by the inputs.

The architecture of carry select adder with D-latch consists of five groups of different bit size Ripple Carry Adders (RCA) and D-latch. The Carry Select Adder (CSA) proposed by Ramkumar &Kittur [26] has two RCAs , one for carry input C_{in}=1 and another RCA for carry input C_{in}=0. In the proposed LCSA, instead of using two separate adders in the regular CSA, the proposed method uses only one RCA block which results in the reduction of area and power consumption. Another RCA block is replaced with a D-latch. Each of the two additions is performed in one clock cycle. When clock goes high, addition for carry input C_{in}=1 is performed. Otherwise the carry input is assumed as zero and sum is stored in adder itself.

The latch is used to store the sum and carry for C_{in}=1. As shown in Fig.7.Carry out from least significant bit adder is used as a control signal for multiplexer to select the final output carry and sum of the n-bit adder.The Fig.7 shows the n-bit adder in which the LSB adder is a 2-bits wide ripple carry adder.

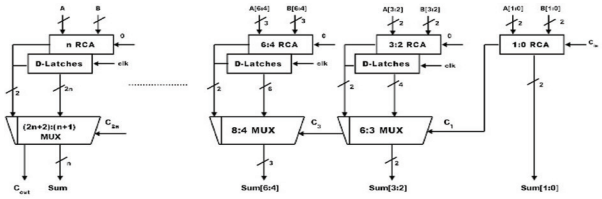


Figure 7: Proposed n-bit Latched Carry Select Adder

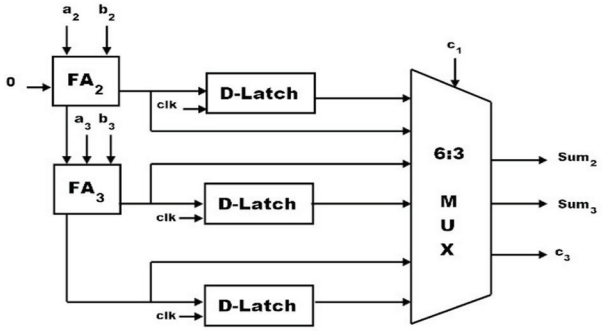


Figure 8: Internal structure of group 2

The upper half of the adder i.e, most significant part is (*n*-2) -bits wide which works according to the clock. Whenever clock goes high addition for carry input one is performed. When clock goes low then carry input is assumed as zero and sum is stored in adder itself.

From the Fig. 7, latch is used to store the sum and carry for C_{in}=1. Carry out from the previous stage i.e, least significant bit adder is used as control signal for multiplexer to select final output carry and sum of the n-bit adder. If the actual carry input is one, then computed sum and carry latch is accessed and for carry input zero MSB adder is accessed. C_{out} is the output carry.

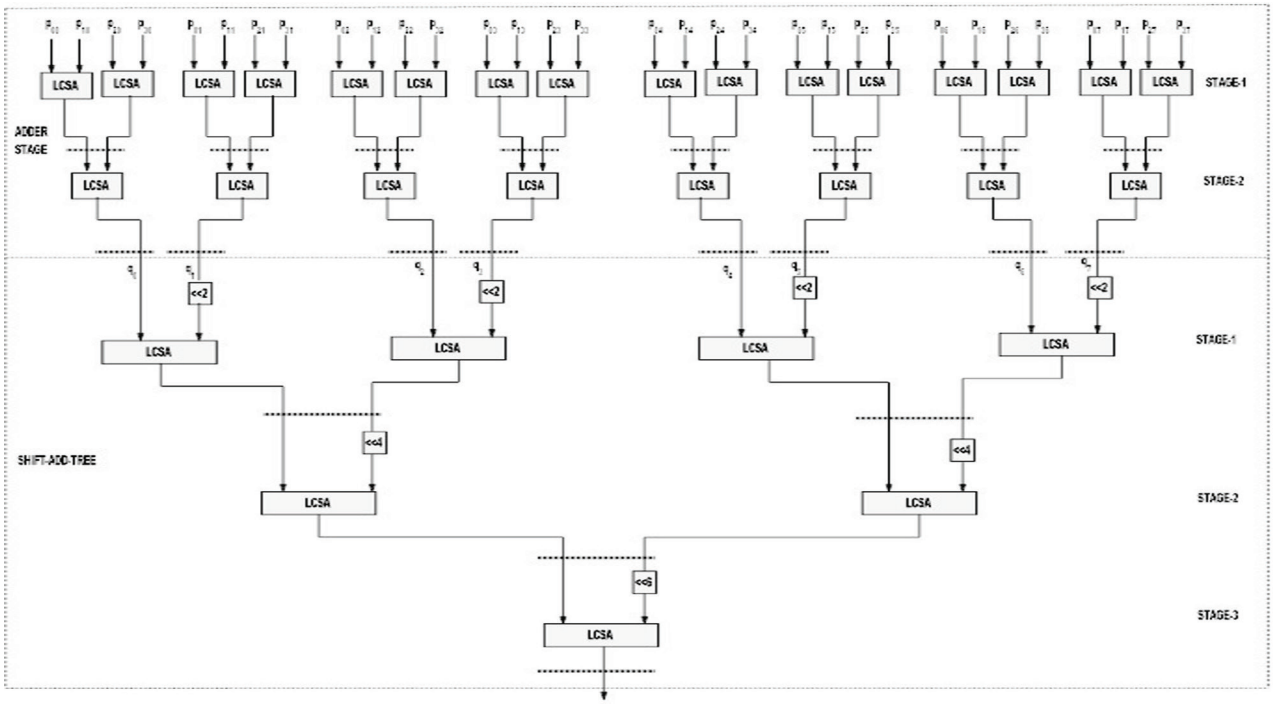


Figure 9: The adder structure of the filtering unit for N = 8 and L = 16

Fig.8 shows the internal structure of group 2 of the proposed n-bit LCSA. The group 2 performed the two bit additions which are a_2 with b_2 and a_3 with b_3. This is done by two Full Adders (FA) named FA_2 and FA_3 respectively. The third input to the full adder FA_2 is the clock instead of the carry and the third input to the full adder FA_3 is the carry output from FA_2. The group 2 structure has three D-Latches in which two are used to store the Sum_2 and Sum_3 from FA_2 and FA_3 respectively and the last one is used to store carry. Multiplexer is used for selecting the actual sum and carry from the previous stage. The 6:3 multiplexer is the combination of 2:1 multiplexers.When the clock is low, a_2 and b_2 are added with carry input equal to zero. Because of low clock, the D-Latch is not enabled. When the clock is high, the addition is performed with carry input equal to one. All the D-Latches are enabled and store the sum and carry for carry input equal to one. According to the value of C_{in} whether it is 0 or 1, the multiplexer selects the actual sum and carry.

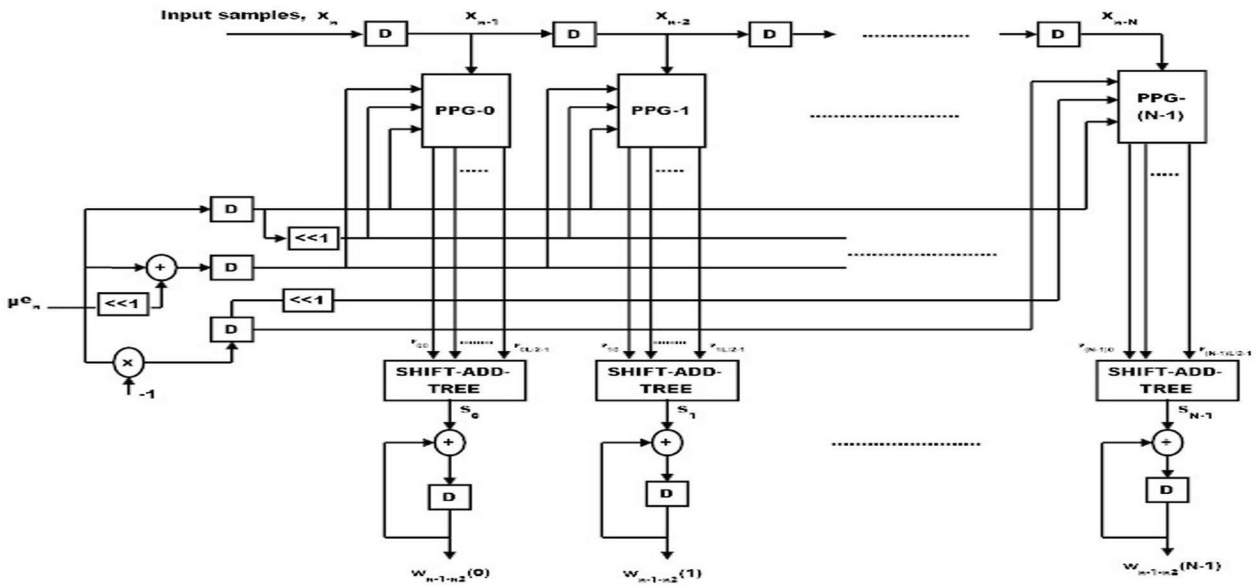


Figure 10: Structure of weight-update block

3.5.1 Structure of adder-tree

The shift-add operation is used for obtaining the desired inner product but the adder size increases as the word length increases. *N*-1 additions are required to add *N* product values. To avoid such increase in word-size of the adders, all the *N* partial products of the same place value are added from all the *N* Partial Product Generator (PPG)s by one adder-tree.

All the *L*/2 partial products generated by each of the *N* PPGs are thus added by *L*/2 binary Carry Select adder trees. The output of the *L*/2 adder-trees are then added by a shift-add-tree according to their place values. Each of the binary Carry Select adder-trees requires $\log_2 N$ stages of adders to add *N* partial products, and the shift-add-tree requires $\log_2 L$-1 stages of adders to add *L*/2 outputs of *L*/2 binary Carry Select adder-trees. The addition scheme for the error-computation block for 8-tap filter and input word-size *L*=16 is shown in Fig. 9. For *N*=8 and *L*=16, the adder-network requires eight binary carry select adder-trees with two stages and a three-stage shift-add tree. Pipeline latches (represented by dashed line) are introduced to reduce the critical-path to one addition time. Pipelining is performed by a feed-forward cut-set retiming of error-computation block .

3.6 Structure of weight-update block

The proposed structure for weight-update block is shown in Fig. 10. It performs *N* multiply accumulate operations of the form $(\mu \times e) \times x_i + w_i$ to update *N* filter weights. The step-size 'μ' is taken as a negative power of two to realize the multiplication with recently available error only by a shift operation. Each of the MAC units therefore performs the multiplication of shifted value of error with the delayed input samples 'x_i' followed by the additions with the corresponding old weight values 'w_i'. All the *N* multiplications for the MAC operations are performed by *N* PPGs followed by *N* shift-adder trees. Each of the PPGs generates *L*/2 partial products corresponding to product of recent shifted error value $\mu \times e$ with *L*/2 2-bit digits of the input word x_i, where the subexpression $3\mu \times e$ is shared within the multiplier. Since the scaled error $(\mu \times e)$ is multiplied with all the *N* delayed input values in the weight-update block, this subexpression can be shared across all the multipliers as well. This leads to a substantial reduction of adder complexity. The final outputs of MAC units constitute the desired updated weights to be used as inputs to the error-computation block as well as the weight-update block for the next iteration.

3.7 Filter coefficient monitoring

In the proposed adaptive reconfigurable FIR filter, the filter order is changed depending on the amplitude of the input signal as well as the filter coefficients. In Linear symmetric FIR filter, the middle filter coefficient has the largest value. The threshold of the filter coefficient and the input is fixed by considering the average sum of the first half of the filter coefficients. The filter coefficients will vary depending on the characteristics of the FIR filter. The filter input and the filter coefficient are denoted as X_n and C_k respectively. Same threshold is used for the input and filter coefficient. The threshold value is denoted as *Th*.

3.8 Decision block

The amplitude of the input samples as well as the filter coefficients are monitored by using the Decision Block, also referred to as the Amplitude Detector (AD) as shown in Fig.11.

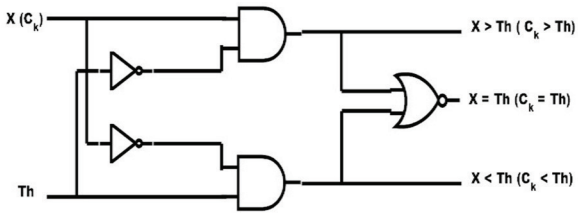


Figure 11: Amplitude Detector (AD)

The proposed optimized Adaptive Reconfigurable FIR Filter works in two modes. When the input signal X as well as the filter coefficient C_k are both smaller than the threshold (average of the filter coeffcients) then multiplierless implementation of the adaptive FIR Filter is carried out. In the other case, if both X and C_k are greater than the threshold value(Th) the area optimized implementation is used. In runtime the modes are varied dynamically. The speech signal with noise inclusion are considered for analysis. This signal value (x) will be var-

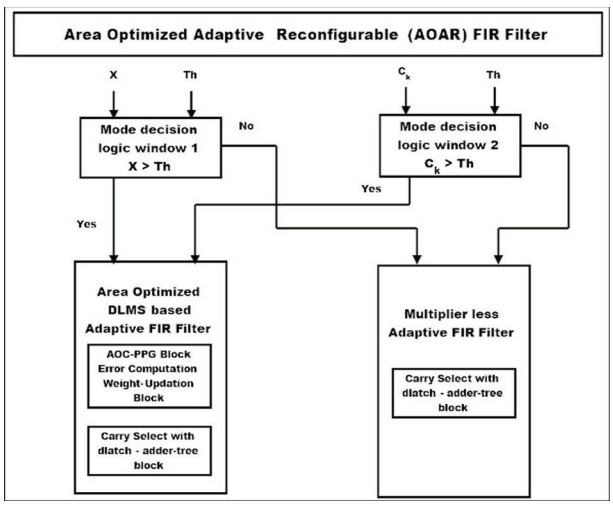


Figure 12: Decision block diagram of Proposed Optimized Adaptive Reconfigurable (OAR) FIR Filter

ying. The modes are switched depending on the input signal variation. The threshold value can be changed depending on the designer's considerations. AD can be implemented using a comparator. The decision diagram of the Proposed Optimized Adaptive Reconfigurable FIR filter is shown in Fig.12.

The filter order can be changed depending on the amplitude of both filter coefficients and the inputs. The product of the data inputs of the filter, with the coefficients, has large variations in amplitude. When the data sample multiplied with the coefficient is small, the FIR filter with multiplierless implementation is used. When the samples as well as the coefficients are large, the proposed Latched Carry Select Adder (LCSA) based Optimized Adaptive FIR filter is used.

3.9 Multiplier control signal decision(MCSD) window

When the number of filter taps increases the switching problem occurs because the multipliers must be turned on/off. Multiplier Control Signal Decision (MCSD) window is used to solve the switching problem. The filter coefficients under this window alone are monitored and the decision is made.

The amplitude of the input samples as well as the filter coefficients are monitored by using AD. The proposed architecture of Reconfigurable FIR filter consists of two modes of operation. When the input sample value X_n and the filter coefficient C_k are less than *Th*, then multiplierless FIR filter is implemented. Otherwise, it performs area optimized FIR filtering operation.

4 Experimental evaluation

4.1 Performance validation metrics

Data arrival time (DAT) in ns
Data arrival time is the time required for the data to propagate from a source sequential circuit, through combinational logic and routing and arrive at the destination sequential circuit (which must happen before the next clock edge occurs).

Minimum sample period (MSP) in ns
The minimum sampling rate is the minimum period at which a signal can be sampled without introducing errors, which is twice the highest frequency present in the signal.

Area delay product (ADP) (sq.mm × ns)
Area Delay Product is defined as the product of area occupied by the design and Minimum Sampling Period.

$$\text{ADP} = \text{Area} \times \text{MSP} \tag{5}$$

Energy per sample (EPS) (nW× ns)
Energy Per Sample is defined as the product of Total power consumed by the design and Minimum Sampling Period.

$$\text{EPS} = \text{Total power} \times \text{MSP} \tag{6}$$

Maximum sampling frequency (MSF) (MHz)
Maximum Sampling Frequency is defined as the reciprocal of the Minimum Sampling Period percentage.

$$\text{MSF} = \frac{1}{\text{MSP}} \times 100 \tag{7}$$

Maximum usable frequency (MUF) (MHz)
Maximum usable frequency (MUF) is the highest frequency that can be used between two ends of an architecture.

For discussions, as a metric of power savings, Power Consumption Ratio, P_r is used. P_r is the ratio of the proposed reconfigurable filter power consumption to the existing filter power consumption.

$$P_r = \left(\frac{P_{\text{Pr}\,oposed}}{P_{existing}} \right) \tag{8}$$

The Power Saving Ratio (PSR) is defined as $(1 - P_r)$.

A significant factor that has a major effect on the proposed filter performance and power consumption is *Th*. Increasing the value of *Th* result in greater power savings. On the other hand, if *Th* is too small, power savings become trivial, but area optimization is carried out.

The Area Saving Ratio (ASR) is defined as $(1 - A_r)$. A_r is the area utilization ratio. A_r is defined as the ratio of the proposed reconfigurable filter area and the existing filter area.

$$A_r = \left(\frac{A_{\text{Pr}\,oposed}}{A_{existing}} \right) \tag{9}$$

Mean Square Error (MSE) Garcia [27] of the filter output is used as a metric of filter performance degradation.

$$M = \frac{1}{n} \sum_{i=1}^{n} (y - y_1)^2 \tag{10}$$

Where *n* is the number of samples, *y* is the expected output and y_1 is the proposed reconfigurable filter output.

The comparison is made between the existing adaptive filter (named as expected output *y*) and the proposed adaptive reconfigurable filter output (named as *y*1).

The degradation in filter performance is analysed by Signal Power to MSE Ratio of the filter output (SMR). SMR realized by Proakis [28] is defined as the ratio of the desired signal to the distorted error signal power, measured in dB.

4.2 Evaluation datasets description

The following database signals are used for analysing the performance of the Proposed Adaptive Reconfigurable FIR filter.

NOIZEUS - A Noisy Speech Corpus: The noisy database contains 30 IEEE sentences, produced by three male and three female speakers corrupted by eight different real-world noises at different SNRs. The noise signals were added to the speech signals at SNRs of 0dB, 5dB, 10dB and 15dB. (Speech Processing Lab [29]).

Speech Enhancement and Assessment Resource (SpEAR) database: The SpEAR database contains carefully selected samples of noise corrupted speech with clean speech references. Lombard speech samples, in which live speech signal recorded in a noisy environment and monaural recordings, in which two recorded speech signals are acoustically combined and re-recorded. (SpEAR Noisy Speech Database Beta Release v1.0 [30]).

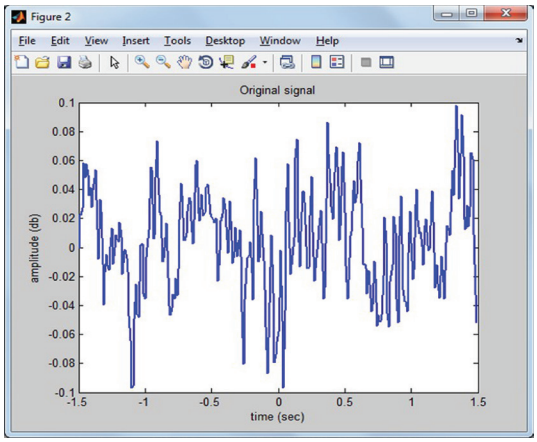
TIMIT: TIMIT is an acronym composed by TI (Texas Instruments) and MIT.TIMIT provides useful speech for both the acoustic and phonetic aspects. The database contains 6300 utterances produced by 630 speakers which includes both male and female speakers of the main US regional variety in .wav format (Becchetti& Ricotta [31]).

Table 1: Comparison of various speech database in terms of MSE measure for Bohman filter characteristics.

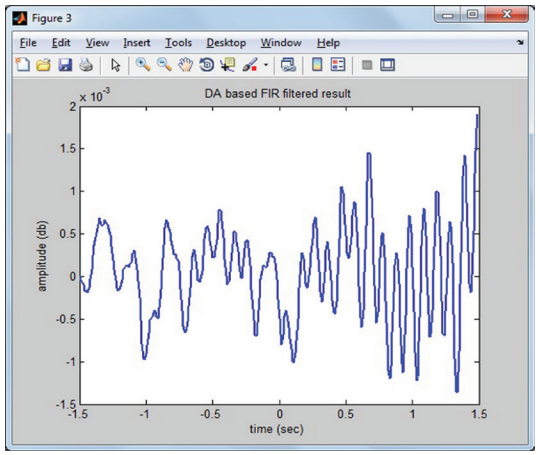
Speech database	MSE measure for Bohman filter characteristics		
	8 taps	16 taps	32 taps
	MSE	MSE	MSE
NOIZEUS sp12_airport_sn15.wav	0.29	0.37	0.5
SPEAR Scholars_f16r1_16.wav	0.274	0.358	0.491
TIMIT bigtips_16.wav	0.266	0.339	0.482
ITU A_eng_f1.wav	0.258	0.326	0.467

ITU test database: International Telecommunication Union, ITU-T Recommendation P56. The dataset includes 16 recorded sentences in each of 20 languages and sentences recorded in the laboratories of some ITU members. (International Telecommunication Union [32).

(a)



(b)



(c)

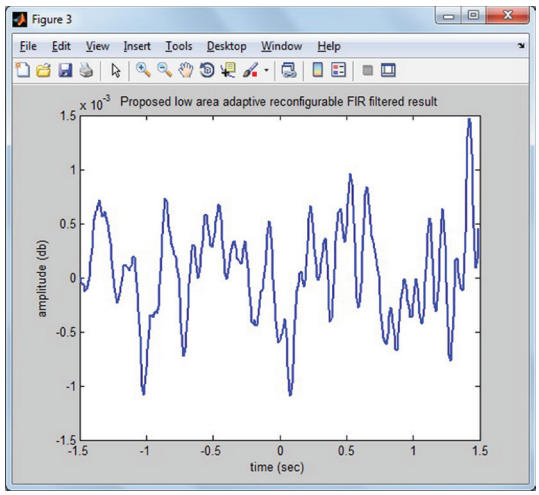


Figure 13: MATLAB simulation results (a) original noisy input signal from spear database (Scholars_f16r1_16.wav noise addition from f16 flight), (b) DA based Adaptive Filter method (Park&Meher [8]), (c) Proposed OAR FIR Filter result.

5 Result analysis and discussion

The performance of the Proposed Optimized Adaptive Reconfigurable (OAR) FIR filter has been analyzed and discussed in the following sections.

The MSE was analyzed for various filter types (8, 16 and 32 taps) namely, equiripple characteristics, least square characteristics, Hamming window characteristics, and Bohman window characteristics. Bohman characteristics results in smaller MSE when compared to other filter characteristics.

5.1 MSE analysis

Table 1 shows the Comparison of various speech databases in terms of MSE measure for Bohman filter characteristics. Bohman filter characteristics gives better performance. This is the reason for choosing the Bohman characteristics for comparison with various speech database. It is inferred from Table 1 that among the four databases considered, ITU database results in a smaller MSE value. The Mean Square Error (MSE) of the filter output is used as a metric of filter performance degradation. As the number of taps increases, the MSE value also increases.

Among the four speech database considered, NOIZEUS database contains noise at different SNRs. The performance measure of the architecture is analysed and tabulated by considering the NOIZEUS database.

MSE1 is the MSE value of the DA based adaptive filter design. MSE2 is the MSE value of the New DA formulation of Block LMS design. MSE3 is the MSE value of the Area-Delay-Power efficient LMS Adaptive filter. MSE4 is the MSE value of the Proposed design of the OAR-FIR filter. Table 2 shows the MSE performance measure for various filter taps with respect to different filter characteristics. MSE values are tabulated for the NOIZEUS database input "sp12_airport_sn15.wav", considering the window monitoring with window size m = 4. It is inferred from the table that the Bohman window filter characteristics results in smaller MSE when compared to other filter characteristics. As the number of taps increases, the MSE value also increases. Table 3 shows the Power Analysis of the 32 Taps Proposed OAR-FIR filter using the Bohman filter characteristics for various speech databases.Table 5 shows the Power Saving Ratio of the Optimized Adaptive Reconfigurable FIR filter design. Fig. 13 shows the MATLAB Simulation of existing as well as proposed Filter techniques.

5.2 Power Analysis

PSR_{OAWC1}, PSR_{OAWC2} and PSR_{OAWC3} are the Power Saving Ratios of the Optimized Adaptive Reconfigurable FIR Filters designs. Equations (11-16) give the expressions for Power Saving Ratios and Power consumption ratios.

$$PSR_{OA1} = (1 - P_{rOA1}) \tag{11}$$

Table 2: MSE measure considering the Window based Coefficient Monitoring of the Proposed Area Optimized Adaptive Reconfigurable FIR filter with NOIZEUS speech as input with window size m = 4 for various pass band, stop band and critical frequencies

Type	**8 taps** $(\omega_p = 0.10, \omega_s = 0.38)$				**16 taps** $(\omega_p = 0.10, \omega_s = 0.26)$				**32 taps** $(\omega_p = 0.10, \omega_s = 0.20)$			
	MSE1	**MSE2**	**MSE3**	**MSE4**	**MSE1**	**MSE2**	**MSE3**	**MSE4**	**MSE1**	**MSE2**	**MSE3**	**MSE4**
EQ	0.46	0.44	0.42	0.4	0.51	0.48	0.45	0.42	0.58	0.554	0.542	0.52
LS	0.455	0.428	0.416	0.39	0.509	0.471	0.438	0.4	0.566	0.538	0.528	0.51
Type	**8 taps** $(\omega_p = 0.10, \omega_s = 0.20)$				**16 taps** $(\omega_p = 0.10, \omega_s = 0.155)$				**32 taps** $(\omega_p = 0.10, \omega_s = 0.135)$			
	MSE1	**MSE2**	**MSE3**	**MSE4**	**MSE1**	**MSE2**	**MSE3**	**MSE4**	**MSE1**	**MSE2**	**MSE3**	**MSE4**
EQ	0.39	0.382	0.37	0.35	0.544	0.521	0.5	0.483	0.636	0.623	0.59	0.576
LS	0.383	0.373	0.352	0.354	0.52	0.506	0.491	0.476	0.594	0.582	0.572	0.56
Type	**8 taps** $(\omega_c = 0.12)$				**16 taps** $(\omega_c = 0.12)$				**32 taps** $(\omega_c = 0.12)$			
	MSE1	**MSE2**	**MSE3**	**MSE4**	**MSE1**	**MSE2**	**MSE3**	**MSE4**	**MSE1**	**MSE2**	**MSE3**	**MSE4**
Ham	0.372	0.35	0.331	0.3	0.46	0.449	0.424	0.401	0.56	0.553	0.532	0.521
Boh	0.368	0.348	0.324	0.29	0.423	0.39	0.386	0.37	0.551	0.535	0.52	0.5

*EQ – equiripple characteristics, LS – least square characteristics, Ham –Hamming window characteristics, Boh – Bohman window characteristics

Table 3: Power Analysis of the 32 Taps Proposed OAR-FIR considering the Bohman filter characteristics for different speech database.

Speech Database	Power results of Bohman Filter Characteristics											
	DA based Adaptive FIR filter [8]			DA based formulation of BLMS [15]			Optimal Pipelined DA based LMS Adaptive filter [17]			Proposed OAR-FIR		
	LP (nW)	DP (nW)	TP (nW)	LP (nW)	DP (nW)	TP (nW)	LP (nW)	DP (nW)	TP (nW)	LP (nW)	DP (nW)	TP (nW)
NOIZEUS sp12_airport_sn15.wav	0.014	67.393	67.407	0.013	64.684	64.697	0.013	60.967	60.98	0.011	56.323	56.334
SPEAR Scholars_f16r1_16.wav	0.015	70.237	70.252	0.014	67.105	67.119	0.014	63.884	63.898	0.012	62.798	62.81
TIMIT bigtips_16.wav	0.015	69.852	69.867	0.014	65.989	66.003	0.013	61.498	61.511	0.011	60.594	60.605
ITU A_eng_f1.wav	0.015	72.568	72.583	0.015	68.432	68.447	0.014	65.587	65.601	0.013	64.852	64.865

LP : Leakage Power in nano watts; DP : Dynamic Power in nano watts; TP : Total Power in nano watts

where P_{rOA1} is the Power consumption ratio for OAR design between the Proposed Optimized Adaptive Reconfigurable FIR filter and DA based Adaptive filter design.

$$P_{rOA1} = \frac{P_{ProposedOAR-FIR}}{P_{DAbasedAdaptiveFilter}} \tag{12}$$

$$PSR_{OA2} = (1 - P_{rOA2}) \tag{13}$$

where P_{rOA2} is the Power consumption ratio for OAR design between the Proposed Optimized Adaptive Reconfigurable FIR filter and DA based formulation of Block LMS.

$$P_{rOA2} = \frac{P_{\mathrm{Pr}\,oposedOAR-FIR}}{P_{DAbasedformulationofBLockLMS}} \tag{14}$$

$$PSR_{OA3} = (1 - P_{rOA3}) \tag{15}$$

where P_{rOA3} is the Power consumption ratio for OAR design between the Proposed Optimized Adaptive Reconfigurable FIR filter and Optimal pipelined DA based Adaptive filter.

$$P_{rOA3} = \frac{P_{\mathrm{Pr}\,oposedOAR-FIR}}{P_{OptimalpipelinedDAbasedAdaptiveFilter}} \tag{16}$$

When comparing the various filter designs, the Power Saving Ratio is high for 32 Taps filter with NOIZEUS speech signal as input (which implies that, the multiplierless implementation is done while switching the modes). PSR_{OA1} is equal to 19.3941% as shown in Table 3.

5.3 Area analysis

The Area Saving Ratios of the Optimized Adaptive Reconfigurable FIR Filter are ASR_{OA1}, ASR_{OA2}, and ASR_{OA3}. They are calculated using Equations (17) – (22).

$$ASR_{OA1} = (1 - A_{rOA1}) \tag{17}$$

where A_{rOA1} is the Area utilization ratio for OAR design between the Proposed Optimized Adaptive Reconfigurable FIR filter and DA based Adaptive FIR filter.

$$A_{rOA1} = \frac{A_{\mathrm{Pr}\,oposedOAR-FIR}}{A_{DAbasedAdptiveFIRFilter}} \tag{18}$$

$$ASR_{OA2} = (1 - A_{rOA2}) \tag{19}$$

where A_{rOA2} is the Area utilization ratio for OAR design between the Proposed Optimized Adaptive Reconfigurable FIR filter and DA based formulation of Block LMS Adaptive FIR filter.

$$A_{rOA2} = \frac{A_{\mathrm{Pr}\,oposedOAR-FIR}}{A_{DAbasedFormulationofBLockLMSAdaptiveFIRFilter}} \tag{20}$$

$$ASR_{OA3} = (1 - A_{rOA3}) \tag{21}$$

where A_{rOA3} is the Area utilization ratio for OAR design between the Proposed Optimized Adaptive Reconfigurable FIR filter and Optimal DA Based Adaptive FIR filter.

Table 4: Power Saving Ratio (PSR) of the Optimized Adaptive Reconfigurable (OAR) FIR Filter

Speech Database	8Taps			16 Taps			32 Taps		
	PSR_{OA1}	PSR_{OA2}	PSR_{OA3}	PSR_{OA1}	PSR_{OA2}	PSR_{OA3}	PSR_{OA1}	PSR_{OA2}	PSR_{OA3}
NOIZEUS sp12_airport_sn15.wav	11.2561	8.7283	5.3286	13.8712	8.0073	2.8308	**19.3941**	16.0177	10.8986
SPEAR Scholars_f16r1_16.wav	13.1335	10.0808	7.5644	15.5653	8.6317	3.0864	16.2870	12.3795	7.9626
TIMIT bigtips_16.wav	12.0153	8.6477	2.8417	13.7687	8.1919	2.8235	18.9817	14.2387	7.9758
ITU A_eng_f1.wav	14.0047	9.6366	7.4794	14.7373	8.5797	2.8797	16.1442	11.0771	7.2194

Table 5: Comparison of area for OAR-FIR filter design

Area Comparison of OAR-FIR design in mm²												
Speech Database	**8 Taps**				**16 Taps**				**32 Taps**			
	C1	**C2**	**C3**	**P1**	**C1**	**C2**	**C3**	**P1**	**C1**	**C2**	**C3**	**P1**
NOIZEUS sp12_airport_sn15.wav	0.094	0.091	0.088	**0.044**	0.185	0.181	0.177	**0.104**	0.347	0.338	0.300	**0.126**
SPEAR Scholars_f16r1_16.wav	0.0783	0.075	0.072	**0.064**	0.158	0.155	0.152	**0.133**	0.287	0.276	0.264	**0.244**
TIMIT bigtips_16.wav	0.076	0.072	0.068	**0.054**	0.144	0.141	0.137	**0.124**	0.256	0.238	0.210	**0.197**
ITU A_eng_f1.wav	0.068	0.064	0.058	**0.085**	0.120	0.118	0.106	**0.176**	0.218	0.196	0.147	**0.268**

C1 –DA based Adaptive FIR filtermethod; **C2** – DA based formulation of Block LMS Adaptive FIR filter;
C3 –Optimal DA based Adaptive FIR filter; **P1** - Proposed OAR-FIR

Table 6: SMR analysis for OAR designs

Speech Database	SMR (dB) analysis		
	8 Taps	16 Taps	32 Taps
NOIZEUS sp12_airport_sn15.wav	41.1620	98.2594	121.73
SPEAR Scholars_f16r1_16.wav	39.7810	96.6564	119.7759
TIMIT bigtips_16.wav	38.0451	97.3598	117.4377
ITU A_eng_f1.wav	34.7751	95.0797	116.3468

$$A_{rOA3} = \frac{A_{\text{Pr}\,oposedOAR-FIR}}{A_{OptimalDAbasedAdaptiveFIRFilter}} \quad (22)$$

Table 4 shows the area comparison of OAR-FIR Filter design.

5.4 Signal power to mean square error ratio analysis

Table 6 presents the SMR results of proposed OAR-FIR designs. The SMR is analyzed for 8 taps, 16 taps and 32 taps FIR filter. For most of the cases the SMR value is larger than 35dB, meaning that the MSE is almost ne-

Table 7: Performance comparison of adaptive filter characteristics based on synthesis using TSMC 180-nm library

Design	DAT (ns)	MUF (MHz)	ADP (sq.mm × ns)	EPS (nW × ns)	MSP (ns)	MSF (MHz)
DA based Adaptive Filter	1.35	740	0.29484	3779.3916	38.8	25.773
DA based formulation of BLMS	1.25	800	0.2455	2665.5164	41.2	24.271
Optimal DA based Adaptive filter	1.19	840	0.175168	2591.65	42.5	23.529
Proposed OAR-FIR	1.15	869	**0.14582**	**2287.4614**	42.1	23.752

glectable. For the speech applications, if SMR is comparable or larger than the signal to quantization error power ratio or the Signal to Noise Ratio (SNR) of a given system, which is usually less than 30 dB (Proakis[27]).
DAT: Data Arrival Time in ns; MSP: Minimum Sample Period in ns; ADP: Area Delay Product expressed in (sq. mm × ns); EPS: Energy Per Sample expressed in (nW× ns); MSF: Maximum Sampling Frequency expressed in (MHz) and MUF: Maximum Usable Frequency expressed in (MHz)

From the Table 7 it is inferred that the Proposed OAR-FIR design has the lowest ADP and EPS of about 0.14582 sq.mm × ns and 2287.4614 nW × ns respectively, when compared to the other proposed designs and existing methods. From the analysis, it is found that the proposed Adaptive reconfigurable FIR filter is more efficient in terms of power and area when compared to the conventional reconfigurable designs.

6 Conclusion

An OAR-FIR Filter design has been proposed. The proposed architecture allows an efficient trade-off between the filter performance and computation energy. In the proposed reconfigurable filter, the input data and the filter coefficients are monitored. Maximum Power Saving Ratio and Area Saving Ratio of about 19.3941% and 41.9413% respectively is achieved for OAR-FIR filter with 32 taps (NOIZEUS input and ITU input respectively). The MCSD window size is fixed (here m = 4) and those filter coefficients lying within the window alone are monitored, the multiplier turning off is reduced, which leads to a smaller Power and area saving ratio. The proposed architecture design gives better performance compared to existing designs.

In the future, the proposed approach can be applied to other areas of signal processing, where a trade-off between power savings and filter performance degradation is considered.

7 References

1. Proakis, J.G., Manolakis, D.K.: Digital Signal Processing: Principles, Algorithms, and Applications. Macmillan Publishing Company (2007).
2. Ranran Liu., Hong xiang Xu., Enxing Zheng., Yi feng Jiang.:Adaptive filtering for intelligent sensing speech based on multi-rate LMS algorithm, Cluster Computing, Springer, pp.1-11(2017). https://doi.org/10.1007/s10586-017-0871-y
3. Ting, L.K., Woods, R., Cowan, C.F.N.: Virtex FPGA implementation of a pipelined adaptive LMS predictor for electronic support measures receivers, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 13, no. 1, pp. 86-99 (2005). https://doi.org/10.1109/TVLSI.2004.840403
4. Van, L.D., Feng, W.S.: An efficient systolic architecture for the DLMS adaptive filter and its applications, IEEE Transactions on Circuits and Systems II, Analog and Digital Signal Processing, vol. 48, no. 4, pp. 359-366(2001). https://doi.org/10.1109/82.933794
5. Meher, P.K., Maheshwari, M.: A high-speed FIR adaptive filter architecture using a modified delayed LMS algorithm, in Proceedings of IEEE International Symposium on Circuits and Systems, pp. 121-124(2011). https://doi.org/10.1109/ISCAS.2011.5937516
6. Park, S.Y., Meher, P.K.: Low-Power, High-Throughput, and Low-Area Adaptive FIR Filter Based on Distributed Arithmetic, IEEE Transactions on Circuits and Systems-II: Express Briefs, vol.60, no.6, pp. 346-350 (2013). https://doi.org/10.1109/TCSII.2013.2251968
7. Meher, P.K., Park, S.Y.: Low adaptation-delay LMS adaptive filter part-II: An optimized architecture, in Proceedings of 54th IEEE International Midwest Symposium Circuits and Systems, pp. 1-4 (2011). https://doi.org/10.1109/MWSCAS.2011.6026643
8. Meher, P.K., Park, S.Y.: Critical-Path Analysis and Low-Complexity Implementation of the LMS Adaptive Algorithm, IEEE Transactions on Circuits and Systems I: Regular Papers, vol.63, no.3, pp. 778-788 (2013). https://doi.org/10.1109/TCSI.2013.2284173
9. Meher, P.K., Park, S.Y.: Area-Delay-Power Efficient Fixed-Point LMS Adaptive Filter With Low Adaptation-Delay, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 22, No. 2, pp.362-371 (2014). https://doi.org/10.1109/TVLSI.2013.2239321
10. Clark, G.A., Mitra, S.K.,Parker, S.:Block implementation of adaptive digital filters, IEEE Transactions on Acoustics, Speech and Signal Processing, vol. 29, no.3, pp. 744-752 (1981). https://doi.org/10.1109/TASSP.1981.1163603
11. Baghel, S., Shaik, R.: FPGA implementation of Fast Block LMS adaptive filter using Distributed Arithmetic for high-throughput, in Proceedings of International Conference on Communications and Signal Processing, pp. 443-447 (2011). https://doi.org/10.1109/iccsp.2011.5739356
12. Baghel, S., Shaik, R.:Low power and less complex implementation of fast block LMS adaptive filter using distributed arithmetic, in Proceedings of

IEEE Students Technology Symposium, pp. 214-219 (2011).
https://doi.org/10.1109/TECHSYM.2011.5783848
13. Jayashri, R., Chitra, H., Kusuma, S., Pavitra, A.V.,Chandrakanth, V.:Memory based architecture to implement simplified block LMS algorithm on FPGA, in Proceedings of International Conference on Communications and Signal Processing , pp. 179-183 (2011).
https://doi.org/10.1109/ICCSP.2011.5739296
14. Mohanty, B.K., Meher.P.K.: A High-Performance Energy-Efficient Architecture for FIR Adaptive Filter Based on New Distributed Arithmetic Formulation of Block LMS Algorithm, IEEE Transactions on Signal Processing, Vol. 61, No. 4, pp. 921-932 (2013).
https://doi.org/10.1109/TSP.2012.2226453
15. Mohanty, B.K., Meher, P.K.: A High-Performance FIR Filter Architecture for Fixed and Reconfigurable Applications, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 24, no.2, pp. 444-452 (2015).
https://doi.org/10.1109/TVLSI.2015.2412556
16. Mohanty, B.K., Meher, P.K.,Patel, S.K.: LUT Optimization for Distributed Arithmetic-Based Block Least Mean Square Adaptive Filter, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol.24 , Issue: 5, pp. 1926-1935 (2016).
https://doi.org/10.1109/TVLSI.2015.2472964
17. Khan, M.T.,Shaik,R.A.:Optimal Complexity Architectures for Pipelined Distributed Arithmetic-Based LMS Adaptive Filter, IEEE Transactions on Circuits and Systems I: Regular Papers, Volume: 66 , Issue: 2, pp. 630-642 (2019).
https://doi.org/10.1109/TCSI.2018.2867291
18. Widrow, B., Stearns, S.D.:Adaptive Signal Processing. Englewood Cliffs, NJ, USA: Prentice-Hall (1985).
19. Haykin, S, Widrow, B.:Least-Mean-Square Adaptive Filters, Hoboken, NJ: Wiley-Interscience (2003).
20. Meyer, M.D., Agrawal, D.P.: A modular pipelined implementation of a delayed LMS transversal adaptive filter, in Proc. IEEE International Symposium on Circuits and Systems, vol.3, pp.1943-1946 (1990).
21. Long, G.H., Ling, F., Proakis, J.G.:The LMS algorithm with delayed coefficient adaptation', IEEE Transactions on Acoustics, Speech and Signal Processing, vol. 37, no.9, pp. 1397-1405(1989).
https://doi.org/10.1109/29.31293
22. Ramanathan, S., Visvanathan, V.: A systolic architecture for LMS adaptive filtering with minimal adaptation delay, in Proc. International Conference on Very Large Scale Integration (VLSI) Design, pp. 286-289(1996).
https://doi.org/10.1109/ICVD.1996.489612
23 Yi, Y., Woods, R., Ting, L.K., Cowan, C.F.N.: High Speed FPGA-Based Implementations of Delayed-LMS Filters', Journal of Very Large Scale Integration (VLSI) Signal Processing, Springer, vol. 39, no. 1, pp. 113-131(2005).
https://doi.org/10.1023/B:VLSI.0000047275.54691.be
24. Cohen, R.H., Herzberg, H., Beery, Y.: Delayed adaptive LMS filtering: current results', in Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, Albuquerque, NM, vol.3, pp. 1273-1276(1990).
https://doi.org/10.1109/ICASSP.1990.115604
25. Parhi, K.K.:VLSI Digital Signal Processing Systems – Design and Implementation', Wiley(1999).
26. Ram Kumar., Kittur, H.M.: Low-Power and Area-Efficient Carry Select Adder, IEEE Transactions on Very Large Scale Integration (VLSI) systems, vol.20, no.2, pp.371-375(2011).
https://doi.org/10.1109/TVLSI.2010.2101621
27 Garcia, A.L.:Probability, Statics and Random Processes for Electrical Engineering', Upper Saddle River, NJ: Pearson Education(2009).
28. Proakis, J.G.:Digital Communications, 3rd ed. New York: McGraw- Hill(1995).
29. Speech Processing Lab, UT Dallas. Available from: <http://ecs.utdallas.edu/loizou/speech/noizeus/>. (2008).
30. SpEAR Noisy Speech Database Beta Release v1.0. CSLU, Oregon Graduate Institute of Science and Technology. Available from: <http://www.cslu.ogi.edu/nsel/data/SpEAR_database.html>(2009).
31. Becchetti, C., Ricotta, L.P.: Speech Recognition Theory and C++ Implementation', By John Wiley &Sons (1999).
32. International Telecommunication Union. Available from: <http://www.itu.int/net/itu-t/sigdb/menu.aspx>.(2015).

Arrived: 21. 08. 2019
Accepted: 13. 01. 2020