

# Modeling and Simulation of Frequency Response Masking FIR Filter Bank using Approximate Multiplier for Hearing Aid Application

Raghavachari Ramya<sup>1</sup>, Sridharan Moorthi\*<sup>2</sup>

<sup>1</sup>) VLSI Systems Research Laboratory, Department of Electrical and Electronics Engineering, National Institute of Technology, Tiruchirapalli, INDIA

E-mail: [407114003@nitt.edu](mailto:407114003@nitt.edu)

<sup>2</sup>) VLSI Systems Research Laboratory, Department of Electrical and Electronics Engineering, National Institute of Technology, Tiruchirapalli, INDIA

E-mail: [srimoorthi@nitt.edu](mailto:srimoorthi@nitt.edu)

Received July 20,2018; Revised December 15,2018; Published December 31,2018

**Abstract:** The tremendous increase in the use of portable electronic devices is due to the development in the fields of signal processing and electronic technology. These battery operated devices needs reduction in power consumption with increased performance and long battery life. Since CMOS technology scaling fast approaches its physical limit of minimum supply voltage and smaller feature size, the hardware designer has to opt for new multiplier architectures for achieving low power and high speed performance. This paper proposes an area and power efficient approximate multiplier architecture. The error metrics and circuit characteristics are estimated to verify its performance advantage over other approximate multipliers. Using frequency response masking approach, a 6-band non-uniform digital FIR filter bank is developed using approximate multiplier for hearing aid application. Audiogram matching is done with audiograms of two different types of hearing losses and the matching error is computed. Simulation results show that the audiogram matching error falls within +/- 4 dB range.

**Keywords:** Approximate Multiplier, Audiogram, Filter Bank, Hearing Aid, Wordlength

## 1. INTRODUCTION

Audio signal processing to devise low power and low cost hearing aids become one of the major application areas in digital signal processing. It should also be noted that a large population affected by hearing defects of different levels requires medical assistance in the form of hearing aids. The hearing aid selectively amplifies the audio sounds such that the processed sound matches with that of one's audiogram. The signal processing block involved in the design process is the digital filters.

Digital filter banks are networks of digital filters. The filter bank separates an input signal into many sub-band signals. These sub-band signals can be independently processed according to the requirements and can also be combined together to generate the desired output signal. Non-uniform filter banks are more desirable because they can better match the physiological properties or the perceptual properties of the human ear. The major arithmetic block in digital filters is the Multiply and Accumulate (MAC) unit. They are the largest power consuming units in digital filters. Since the speeding up and energy reduction in VLSI technology fast approaching its physical limits [1-2], new approaches in digital design of arithmetic circuits are mandated for high speed, low power VLSI systems. Thus, to achieve improved performance in terms of speed and power consumption it is essential to adopt power efficient

---

\* Corresponding author: [srimoorthi@nitt.edu](mailto:srimoorthi@nitt.edu)

hardware design of multiplier and adder to enhance the overall system performance. The adder and multiplier unit decide the speed, size and power dissipation of the MAC unit and in turn of the filter bank module.

Approximate computing represents a paradigm shift in low-power VLSI design for error resilient signal processing applications. For multiplication involving large numbers, truncation and allowance of a small magnitude of error in the generation of partial products will improve speed and reduce power consumption in the multiplier to a large extent. Digital signal processing for speech processing, multimedia, graphics, and computer vision can accommodate approximation methods to generate the outputs of multiplication and accumulation process. This is possible because of the inherent error resilience of above mentioned application areas.

This paper proposes an area and power efficient approximate integer multiplier architecture. Both unsigned and signed multiplier architectures are developed and their performances were studied. A 6-band non-uniform Finite Impulse Response (FIR) filter bank is developed using the Frequency Response Masking (FRM) technique and used to demonstrate the efficacy of the approximate multiplier in applications like hearing aids. The computationally intensive MAC operation of FIR filtering is performed using the proposed approximate multiplier. The approximate multiplier architecture is modeled using Verilog HDL. The circuit characterization is done by evaluation of the area, power, and delay performance of the circuit. The error characterization of the proposed design is performed using standard error characterization techniques and compared with other similar approximate designs.

The paper is organized as follows. The related research in approximate multiplier design and filter bank design for hearing aid are reviewed in section 2. The description of the proposed multiplier architecture and its error characteristics are presented in Section 3. Section 4 discusses the hardware implementation of the proposed approximate multiplier and its circuit characteristics are discussed. Section 5 deals with the implementation of FRM based FIR filter bank for error tolerant hearing aid application using the proposed multiplier unit and audiogram matching for two different hearing losses are analyzed. Finally, the conclusions are presented in Section 6.

## 2. RELATED WORKS

In this section, we describe some prior research that focuses on approximate multiplier design and filter bank design for hearing aid applications.

The difficult design problem in FIR filter design is to design filters with sharp transition band with less hardware complexity. This can be overcome by using FRM design method [3-4]. Frequency Response Masking technique is the most efficient method for the realization of digital FIR filters with sharp transition bandwidth. The advantage of FRM technique is that the filter has a very sparse coefficient vector and hence the hardware complexity is low. Also, it has guaranteed stability and linear phase response. Ying Wei and Yong Lian [5] proposed an 8-band non-uniform digital FIR filter bank for hearing aid applications using FRM approach. Two half-band filters are used as prototype filters and the filter bank provides a minimum stop band attenuation of 80 dB. The filter bank is tested with audiograms for different hearing losses and matching errors are within  $\pm 5$  dB. Nisha Haridas and Elizabeth Elias proposed a set of variable bandwidth filters for hearing aid design using Farrow structure [6]. A fixed number of bands are generated from the variable bandwidth filters by spectral shifting of the required bandwidth response. Each filter of the bank is tuned to match different categories of audiograms. The reported matching errors are within 3 dB for a 6-band filter bank and better matching is achieved with higher order filter banks. Deng [7] proposed a three-channel variable filter bank for digital hearing aid applications. A normalized analog Chebyshev type-I low pass filter is used as a prototype filter. Using analog frequency

transformations and a modified bilinear transformation approach, variable low pass filter, variable band pass filter, and variable high pass filter are obtained from the prototype filter. These filters have adjustable gains and band edge frequencies which can be independently tuned for matching various hearing loss patterns. The variable filter bank was designed using Infinite Impulse Response (IIR) digital filters, which introduces non-linear phase to the system. The audiogram matching experiments shows that the maximum matching error is below 3.44 dB. A 16-band low power non-uniform FIR filter bank designed using multiplier less approach for a digital hearing instrument was proposed by Setiawan, Lesmana, and Gwee [8]. The power reduction is mainly due to the replacement of multipliers with multiplexers and adder arrays and restricting the wordlength of filter coefficient and data word. Minimum attenuation of 60 dB is provided in the stop band for high programmability of magnitude gain. Ying Wei and Debao Liu [9] proposed a reconfigurable FIR filter bank with adjustable sidebands. The FRM technique used reduces the computational complexity and is able to show better results for the matching error. Interpolation and decimation techniques are used to make the filter bank reconfigurable. A quasi-ANSI S1.11 1/3 octave filter bank with 18 bands is proposed in [10] for hearing aid design with significant reduction in group delay and number of multiplications per sample. Interpolated FIR technique is adopted for reducing the computational complexity. Ying Wei and Yinfeng Wang [11] proposed an adjustable filter bank for personalized hearing aids. Fractional interpolation technique along with symmetric filters and complementary filters are utilized to reduce the complexity. The filter bank can meet different types of hearing loss with acceptable delay.

Moving away from the exact arithmetic towards the approximate arithmetic for filter bank design gives enough opportunity to develop power and area efficient circuits and systems for error tolerant audio signal processing applications like portable hearing aids. Discussions of the recent developments in the field of approximate multipliers are elaborated.

Several error tolerant approximate adders and multipliers have been proposed in the literature [12-15]. Most of these multiplier designs use truncated multiplication method. Truncation reduces the complexity of the multiplier unit by computing the most-significant bits of the product only. In order to reduce the error introduced by the truncation process, a correction factor may be added [16]. As more columns are eliminated, the error introduced by truncation also increases. Kyaw et al. [17] developed an Error tolerant multiplier (ETM) in which the operands are split into a multiplication part with higher order bits and a non-multiplication with lower order bits. Every bit position from left to right is checked for the lower order bits and if either or both the operands are '1', all the bit positions are set to '1' from that bit onwards. For the higher order bits, normal multiplication operation is performed. The major drawback of the ETM is the relatively large magnitude of the relative error. Parag Kulkarni et al [18] proposed a  $2 \times 2$  underdesigned approximate multiplier block and used this block to construct large inaccurate multipliers. The inaccurate multipliers have shown to achieve an appreciable power savings over an accurate multiplier. The performance of the underdesigned multiplier decreases in terms of area and power with increasing wordlength. Reza Zendegani et al [19] presented three hardware implementations of approximate multipliers based on rounding of the inputs in the form of  $2^n$  was proposed. One unsigned and two signed Rounding-Based Approximate Multiplier (RoBA) architectures are implemented. In [20], two variants of signed 16-bit approximate radix-8 booth multipliers are developed using approximate recoding adder logic with and without truncation of a number of less significant bits in the partial products. Venkatachalam & Ko [21] proposed two variants of approximate multipliers. The partial products of the multiplier are altered using generate and propagate signals and the accumulation of generate signals is done column wise. Approximate 4:2 compressors and adders are used to accumulate the remaining partial products. In the first variant, approximation is applied to all the columns of the partial product, and in variant 2, approximation is not applied to most significant column of partial products. Even though Multiplier design2 has lower relative error compared to multiplier design1, it offers less area

and power savings as compared to multiplier design1. Narayanamoorthy et al [22] proposed Dynamic Segment Method (DSM), Static Segment Method (SSM), Enhanced Static Segment Method (ESSM) based approximation multipliers for various DSP and classification applications. A scalable Dynamic Range Unbiased Multiplier (DRUM) for approximate applications such as image filtering, JPEG compression, perceptron classifier is proposed in [23]. Both [22] and [23] follows an approach for approximating the input operands based on the output of a complex leading one detection (LOD) circuit. Even though DRUM offers better accuracy than [22], the power and area gets increased due to the complex steering logic employed. The steering logic circuit extracts a predefined number of consecutive bits starting from the leading '1' bit to be retained for further processing. As the size of the input grows, the complexity of the steering logic also increases. We propose an approximate multiplier circuit that has the same accuracy as that of DSM and better circuit characteristic than both DSM and DRUM.

In this paper we target power and area reduction by reducing the wordlength of the input operands. The wordlength reduction is performed by right shifting and the amount of right shift required to truncate the operands is derived from the most-significant  $N/2$ -bits of the inputs. The approach retains a block of most significant information carrying  $N/2$ -bits beginning with the leading one for multiplication. This ensures much accurate result compared to a truncation process. The reduced circuit complexity and error characteristics thus achieved in the design of the approximate multiplier makes it very attractive for ASIC implementation.

### 3. PROPOSED APPROXIMATE MULTIPLIER

The data wordlength affects the design parameters like speed, area, and power [24] in VLSI circuits. Increased speed and reduced power consumption in DSP circuits can be achieved using data wordlength reduction in various arithmetic operations. This reduces the switching activity of CMOS circuits and in turn the power consumption. The multiplier unit decides these performance parameters for DSP circuits. The data wordlength reduction can be applied to one or both the inputs of the multiplier and this greatly reduces the area and power in the multiplier unit.

The main idea in the design of proposed multiplier lies in the fact that simple right shifting operation is exploited to reduce the wordlength of the input operands for multiplication with good accuracy along with reduced area and power compared to similar designs. The approximate multiplier architecture can be represented by three sub-blocks as shown in Fig 3.1.

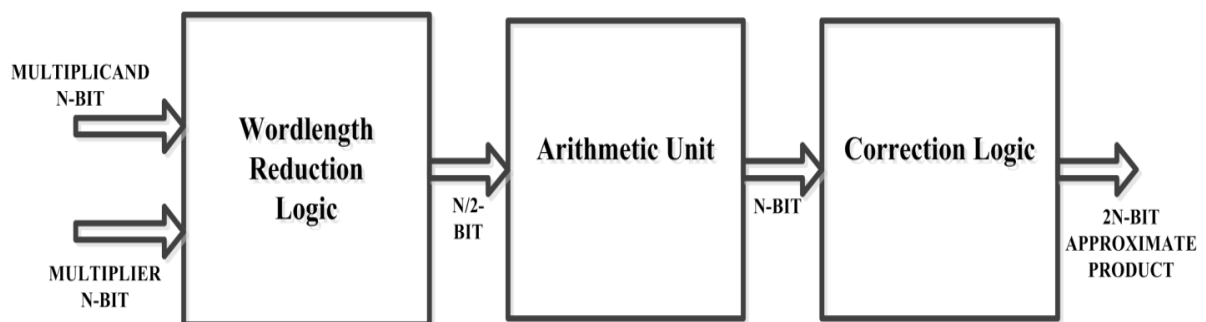


Fig.3.1 General Block Diagram of Approximate Multiplier

The first sub-block is the wordlength reduction logic which reduces the input operand wordlength. Second sub-block is an arithmetic unit which is an exact multiplier block of  $N/2$ -bit wordlength and the last sub-block is a correction logic block to compensate for the reduction of input operand wordlength. In the proposed approximate multiplier, the

wordlength of the N-bit input operands are reduced to N/2-bits and multiplication is done with a single N/2-bit multiplier instead of N-bit multiplier.

The block diagram of unsigned approximate multiplier is shown in Fig.3.2. The wordlength reduction logic is composed of following blocks: N/2-bit priority encoder, a binary to excess one converter, 2:1 MUX, and N-bit right barrel shifter. The priority encoder receives only the most-significant N/2-bit as the input. The location of one bit position present in the most-significant N/2-bit of the input i.e.  $A[N-1:N/2]$  and  $X[N-1:N/2]$  are encoded by the N/2-bit priority encoder. In order to obtain the amount of shift needed to obtain the truncated input operands, a binary-to-excess one converter and a 2:1 multiplexer (MUX) are used. The select lines needed for the 2:1 MUX are obtained by performing the bit-wise OR operation of the most significant N/2-bit of the input operand. In case of input operands with wordlength less than N/2-bits, the select value will be set as '0' and the 2:1 MUX select the input as '0' and the right barrel shifter will not provide any shift and input operands are not truncated. For operand size greater than N/2-bits, then select value will be set as '1' and the 2:1 MUX will select the required shift amount from the binary-to-excess one converter. The input operands are truncated to N/2-bit wordlength using right barrel shifters and multiplication is performed using N/2-bit multiplier. The correction logic is made up of 2N-bit left-barrel shifter which expands the truncated product to a 2N-bit number by left-shifting. The barrel shifter left shifts the product by an amount equal to the number of bit positions which is the sum of right shifts applied to both the input operands.

The design can be extended for signed numbers by inserting a two's complement block at the input of each branch. At the output the product value may be negated if necessary. For further simplifying the circuit complexity, the maximum negative input of magnitude  $-2^N$  is left out from the computation and the resulting simplified architecture of proposed signed approximate multiplier is shown in Fig.3.3. The signed multiplier comprises of a two's complement block at the input and output, N/2-bit sign-extension encoder, inverter based control logic, right and left Barrel shifters, unsigned exact multiplier block of N/2-bit wordlength. The multiplier receives the inputs in signed two's complement format. The sign of the input operand is determined and if the sign bit is set, the two's complement block determines the absolute value of the negative operand. A modified priority encoder is used as a sign-extension encoder to calculate the effective size of the positive operands. The effective size of the positive operands is computed by finding the number of sign-extension bits (zeros) immediately to the right of the most-significant sign bit. For a N-bit multiplier,  $N/2 + \log_2(N/2)$  priority encoder is required since the multiplier block is of N/2-bit wordlength. The count of the sign-extension bits is passed to the control logic block which is used to calculate the amount of shift needed to truncate the inputs. If the size of the input operand is below N/2-bits, then shift operation need not be performed and operands are not truncated. In case of operand wordlength above N/2-bits, the right barrel shifter right shifts the input operands to the required number of places as decided by the control logic to obtain the N/2-bit input operand. Now the effective size of the input operands is limited to wordlength of size N/2-bit instead of N-bits and the multiplication is performed as an  $N/2 \times N/2$ -bit multiplication. Unsigned Exact Wallace Tree multiplier is adopted for the fundamental multiplier block. In order to compensate for the N/2-bit truncated multiplication, the correction logic block is used. The obtained truncated product is left shifted appropriately to compensate for the truncation performed. Depending on the sign of the input operands, the unsigned product is negated to obtain the final signed product as the output.

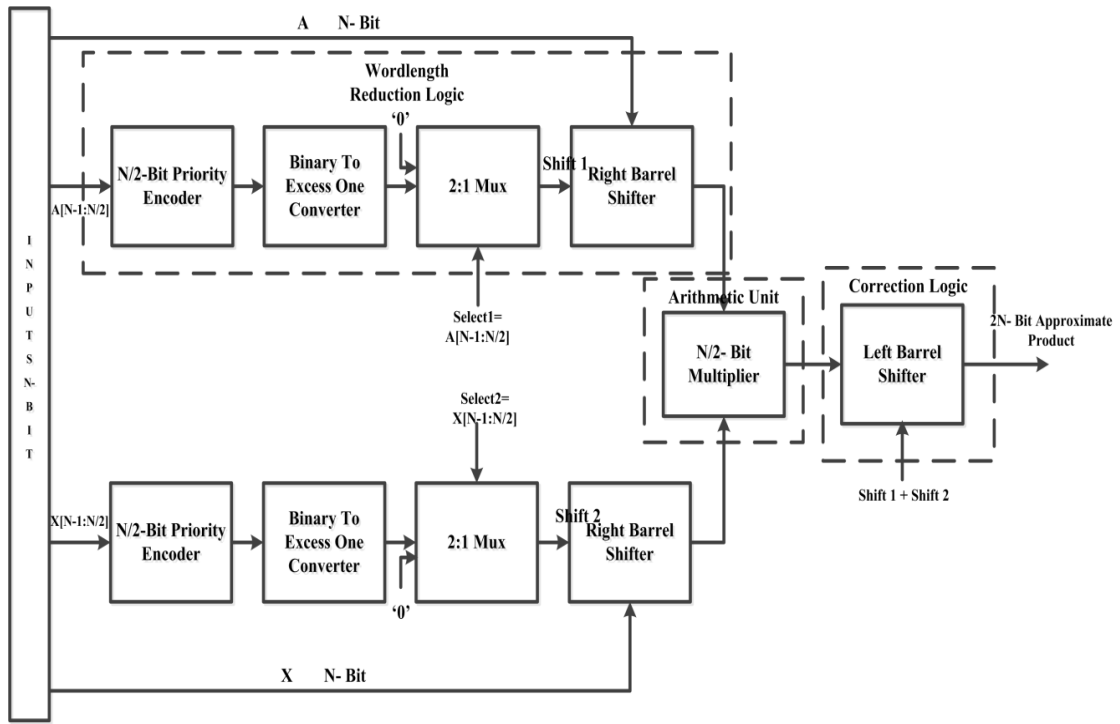


Fig. 3.2 Block Diagram of Unsigned Approximate Multiplier

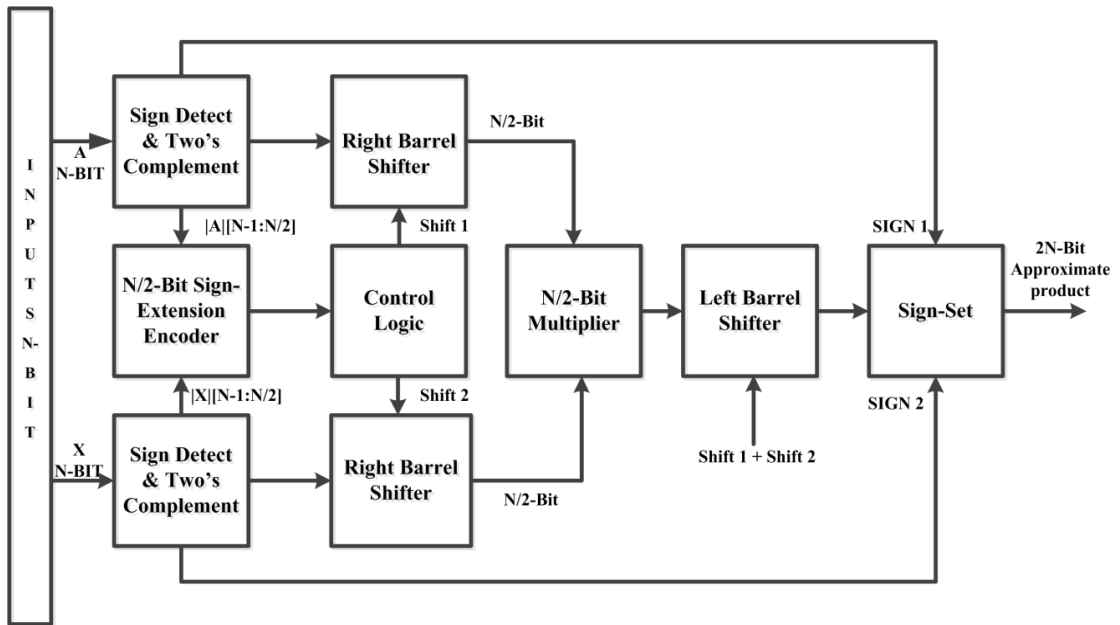


Fig. 3.3 Block Diagram of Signed Approximate Multiplier

3.1. Illustration of Approximate Multiplication of two 16-bit signed numbers

As an example, a multiplication of two signed 16-bit numbers with multiplicand  $A = -259$  and multiplier  $X = 517$  is shown in Fig. 3.4. The MSB of the inputs are used to determine the sign of the operands. Since the sign bit of  $A$  is '1' the operand is negative and hence the absolute value of  $A$  is calculated and used for computation. Hence the value of  $A$  becomes 259 and  $X = 517$ . The control logic will decide the amount of shift needed for both the inputs based on the wordlength of the input. The right barrel shifter provides 1-bit right shift for operand  $A$  and 2-bit right shift for operand  $X$ . After right shifting, the operands are truncated

to 8-bit wordlength i.e.,  $A' = 129$  and  $X' = 129$  and process of multiplication is illustrated in Fig 3.4.

A	1111	1110	1111	1101
A	0000	0001	0000	0011
X	0000	0010	0000	0101

---

Truncated Input A'		1000	0001
Truncated Input X'		1000	0001

---

Truncated Product P'	0100	0001	0000	0001
----------------------	------	------	------	------

---

AP	1111	1111	1111	1101	1111	0111	1111	1000
----	------	------	------	------	------	------	------	------

EP	1111	1111	1111	1101	1111	0100	1111	0001
----	------	------	------	------	------	------	------	------

**Fig. 3.4** Illustration of multiplication of two 16-bit signed Numbers  $A = -257$  and  $X = 513$ ;

AP: Approximate signed product, EP: Exact Product

The truncated product  $P'$  is expanded to 32-bits by left-shifting three bits to obtain the unsigned approximate product. Since the multiplicand is negative, two's complement of the unsigned product is taken and approximate signed product,

$$AP = 1111\ 1111\ 1111\ 1101\ 1111\ 0111\ 1111\ 1000 = -133128$$

is obtained in place of the exact product value,

$$EP = 1111\ 1111\ 1111\ 1101\ 1111\ 0100\ 1111\ 0001 = -133903.$$

### 3. 2. Error Characteristics of the proposed system

The error metrics such as Error rate, normalized mean error distance (NMED), absolute value of mean relative error distance (MRED) and percentage mean accuracy are used to evaluate the performance of approximate multipliers [15] [25]. The definitions of the various error metrics are given as follows:

1. Error Distance (ED): For adders and multipliers, ED is the absolute difference between the accurate output ( $M$ ) and the approximated output ( $M'$ ).

$$ED = |M - M'|$$

2. Mean Error Distance (MED): It is computed by taking the average value of all possible EDs

$$MED = \frac{1}{N} \sum_{i=0}^N ED_i$$

Where  $N$  is the total number of samples, and  $ED_i$  is the error distance in the  $i^{\text{th}}$  value.

3. Normalized mean error distance (NMED): It is the normalization of the mean error distance by the maximum output of the accurate multiplier.

$$NMED = MED / M_{max}, \text{ where } M_{max} \text{ is the maximum accurate product}$$

4. Mean Relative Error Distance (MRED): MRED is computed as the average value of all possible relative error distances and is defined as:

$$MRED = \frac{1}{N} \sum_{i=0}^N \frac{ED_i}{M_i}$$

where  $ED_i$  and  $M_i$  are the error distances and the accurate output of the  $i^{\text{th}}$  input

5. Mean Accuracy: It is defined as 100-MRED.

6. Error Rate (ER): It is defined as the probability of producing incorrect outputs for different combination of inputs.

$$\text{Error Rate} = \text{Number of Incorrect outputs} / \text{Total number of outputs}$$

In order to evaluate the error performance of the proposed multipliers two sets of hundred thousand random numbers are generated with uniform probability and the multiplication is performed. The error metrics for different 16-bit approximate multipliers were evaluated and summarized in Table 3.1.

**Table 3.1** Arithmetic Accuracy Comparison of Proposed 16-bit Multiplier with State-of Art Designs

<b>Unsigned Designs</b>	<b>Error Rate (%)</b>	<b>NMED (%)</b>	<b>MRED (%)</b>	<b>Mean Accuracy (%)</b>
Proposed unsigned Approximate Multiplier	99.95	0.13	0.53	99.47
Underdesigned Multiplier UDM[18]	80.85	1.37	3.33	96.67
Unsigned RoBA [19]	99.96	0.69	2.93	97.07
Venkatachalam and Ko Multiplier Design1 [21]	99.80	1.78	7.63	92.37
DSM8×8 [22]	99.95	0.13	0.53	99.47
DRUM8 [23]	99.98	0.09	0.36	99.64
<b>Signed Designs</b>	<b>Error Rate (%)</b>	<b>NMED (%)</b>	<b>MRED (%)</b>	<b>Mean Accuracy (%)</b>
Proposed signed Approximate Multiplier	99.87	0.032	0.52	99.48
Signed RoBA [19]	99.90	0.172	2.88	97.12
Approximate signed RoBA [19]	99.94	0.173	2.89	97.11

The results tabulated in Table 3.1 shows, the current design of the approximate multiplier provides highest accuracy in terms of various error metrics. The proposed unsigned multiplier achieves the same error performance as that of the DSM8. This is due to the same functionality of the wordlength reduction logic even though the hardware implementation differs in them. Due to the unbiased nature of error distribution of the DRUM8 design, it shows a better error



performance than the proposed multiplier. Comparing with the unsigned RoBA design, the error performance of DSM8 is shown to be better [19]. The proposed multiplier also shows better error performance comparing with RoBA designs. Smaller the values of NMED and MRED shows that the proposed approximate multiplier gives higher accuracy over underdesigned multiplier, RoBA, Multiplier design1 architectures. The relative error is also calculated for inputs of various wordlength and is tabulated in Table 3.2.

**Table 3. 2** Variation of Relative Error (RE) of outputs in percentage for various wordlength for the proposed Approximate Multiplier

Multiplier Type	Input wordlength	RE <0.5%	RE <1%	RE <2%	RE <5%	RE <10%	RE <20%
Unsigned	8-bit	3.79 %	5.4%	9.6%	31.7%	79.8%	100%
	16-bit	47%	97 %	100%			
	32-bit	100%					
Signed	8-bit	9.3%	10.72%	16.1%	41.7%	85.6%	100%
	16-bit	48.6%	97.23%	100%			
	32-bit	100 %					

Table 3.2 gives the percentage of outputs with the relative error less than the specified percentage for various wordlength such as 8-bit, 16-bit, and 32-bit proposed approximate designs. It must also be noted that the relative error reduces as wordlength of the operand increases. For a 16-bit design, the relative error is less than 2% and that for a 32-bit multiplier, it is less than 0.5%. The difference between the exact and approximate multiplier almost vanishes for 32-bit signal processing applications and it is negligible for 16-bit applications.

#### 4. HARDWARE IMPLEMENTATION OF THE PROPOSED APPROXIMATE MULTIPLIER

The proposed 16-bit approximate multiplier architectures are modeled using Verilog HDL and synthesized in cadence RTL Compiler using generic PDK (gPDK) 90-nm CMOS technology with typical library settings. The functionality of the proposed multipliers is verified using Cadence NCSIM and all the designs are synthesized in RTL compiler with proper timing constraints. The post-synthesis circuit performance characteristics such as power, area and critical-path delay are tabulated in Table 4.1. The power-delay-product (PDP) is a measure of energy and is defined as the product of average power and the corresponding delay of the circuit. Since leakage currents are also contributing to the total power, better parameters for characterizing circuit performance are Energy or Power-Delay Product (PDP), and Area-Delay Product (ADP) [15] [26]. Hence the compound metrics such as PDP, and ADP were also computed and tabulated in Table 4.1 for performance comparison.

Table 4.1 gives the post-synthesis circuit characteristics of various 16-bit multipliers. The circuit characteristics of the proposed design are compared against exact multiplier architectures like Wallace tree (exact unsigned), and Baugh-Wooley multiplier (exact signed) architectures. For the comparison, a few of the state-of-art approximate multiplier architectures are considered and implemented using gPDK 90-nm CMOS technology with the same timing constraints. The synthesized gate level netlist is used to extract the layout of the proposed approximate multiplier using Cadence SOC Encounter and the physical layout of the proposed 16-bit multiplier is shown in Fig.4.1.

**Table 4.1.** Post synthesis performance characteristics of various 16-bit multipliers

<b>Unsigned Designs</b>	<b>Power (<math>\mu</math>W)</b>	<b>Delay (ns)</b>	<b>PDP (pJ)</b>	<b>Area (<math>\mu</math>m<sup>2</sup>)</b>	<b>ADP (<math>\mu</math>m<sup>2</sup>. ns)</b>
Proposed unsigned Multiplier	303.47	3.97	1.21	3533	14026
Underdesigned Multiplier [18]	760.49	4.05	3.07	6241	25276
Unsigned RoBA [19]	235.97	4.84	1.14	4522	21886
Venkatachalam & Ko Multiplier 1[21]	425.75	4.38	1.86	4527	19828
DSM 8 $\times$ 8 [22]	402.71	4.44	1.78	3548	15753
DRUM8 Segment [23]	424.83	4.64	1.96	3806	17659
Wallace Tree (Unsigned Exact)	871.72	4.08	3.56	7012	28609
<b>Signed Designs</b>	<b>Power (<math>\mu</math>W)</b>	<b>Delay (ns)</b>	<b>PDP (pJ)</b>	<b>Area (<math>\mu</math>m<sup>2</sup>)</b>	<b>ADP (<math>\mu</math>m<sup>2</sup>. ns)</b>
Proposed signed Multiplier	414.02	5.34	2.21	4012	21424
Signed RoBA [19]	541.17	5.24	2.83	5640	29553
Approximate Signed RoBA[19]	537.40	5.15	2.76	5210	26831
Baugh-Wooley (Signed Exact)	769.07	4.56	3.51	6679	30456

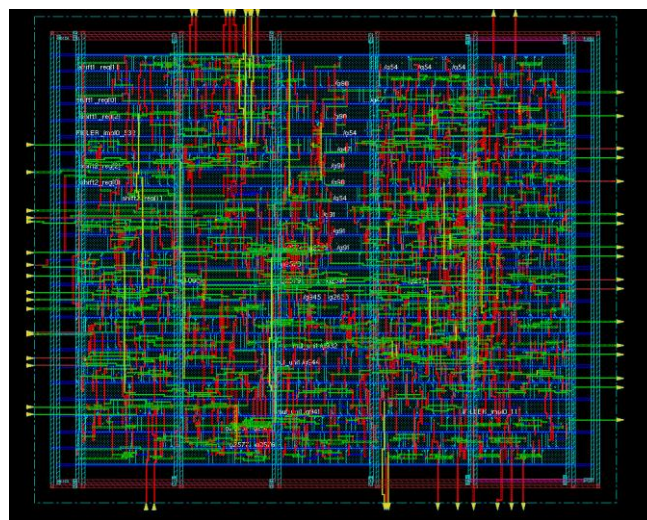
The tabulated result in Table 4.1. shows that the proposed unsigned and signed multiplier architectures have smaller values of area and power consumption (except that unsigned RoBA has lower power consumption) than other existing state-of-art multipliers, due to hardware efficient wordlength reduction logic employed, thereby their PDPs and ADPs are also small. The area requirement of the proposed approximate unsigned (signed) multiplier is 50% (40%) lower than that of exact Wallace (Baugh-Wooley) multiplier. The proposed 16-bit unsigned (signed) multiplier consumes 65% (45%) less than the total power consumed by exact Wallace (Baugh-Wooley) multiplier. The above results show the efficiency of the multiplier in terms of area and power compared to standard designs. The Energy or Power-Delay Product (PDP), and ADP of the proposed unsigned (signed) approximate multiplier are about 66% (37%), 51% (30%), lower than that of exact Wallace (Baugh-Wooley) multiplier.

Table 4.2 illustrates the ranking of approximate multipliers in terms of both circuit metrics and error metrics such as PDP, ADP, NMED and MRED. The results reveal that the proposed unsigned multiplier gives minimum area, power, and ADP; While DRUM8 gives lowest NMED and MRED values and unsigned RoBA design gives lowest PDP value in comparison to all multipliers. But the NMED and MRED error metrics of unsigned RoBA is higher than the proposed, DSM8, and DRUM8 designs. The other two multipliers have comparatively less performance in terms of design and circuit metrics.

**Table 4.2** Ranking of Approximate Multipliers in terms of Design Metrics and Error Metrics

<b>Unsigned Designs</b>	<b>PDP</b>	<b>ADP</b>	<b>NMED (%)</b>	<b>MRED (%)</b>
Proposed Unsigned Multiplier	2	1	1	2
Underdesigned Multiplier	6	6	4	4
Unsigned RoBA	1	5	3	3
Venkatachalam & Ko Multiplier Design 1	4	4	5	5
DSM 8×8	3	2	2	2
DRUM8 segment	5	3	1	1
<b>Signed Designs</b>	<b>PDP</b>	<b>ADP</b>	<b>NMED (%)</b>	<b>MRED (%)</b>
Proposed Signed Multiplier	1	1	1	1
Signed RoBA	3	3	2	2
Approximate Signed RoBA	2	2	3	3

The proposed signed multiplier gives lowest PDP, ADP, NMED, and MRED values in comparison with signed-RoBA and approximate signed RoBA architectures. The lowest value of PDP and ADP is due to the reduced complexity wordlength reduction logic employed in the multiplier design.

**Fig 4.1** Layout of proposed 16-bit approximate multiplier

## 5. APPLICATION- APPROXIMATE FIR FILTER BANK

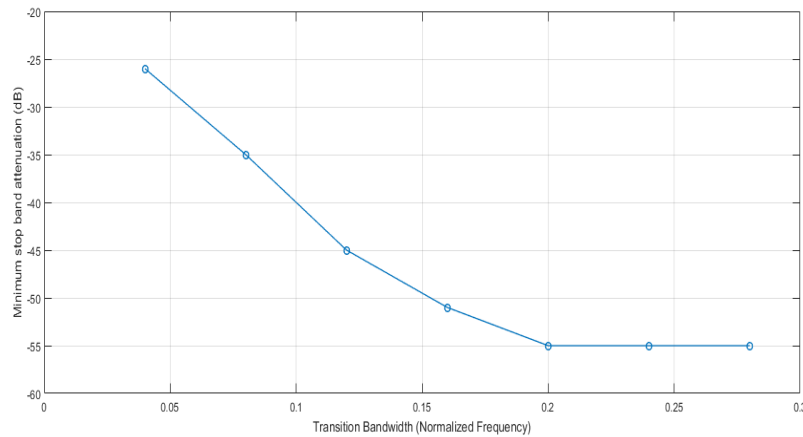
The design method followed in the design of filter banks is the Frequency Response Masking approach introduced in [3]. The method finds wide acceptance in the design of FIR filters with sharp transition band. The filter bank proposed covers the frequency ranges from 0 to 8 KHz with 6 non-uniform bands. The sampling frequency selected as 16 KHz. The prototype filter is  $H(z)$  and the interpolated filters  $H(z^4)$  and  $H(z^2)$  are used for the filter design. The

higher order filters appropriately cascaded to generate the desired frequency response when masked with  $H(z)$ . The transfer functions of different sub-bands in the 6-band filter bank are listed in the Table 5.1.

**Table 5.1** Transfer Functions for Each Sub-Bands

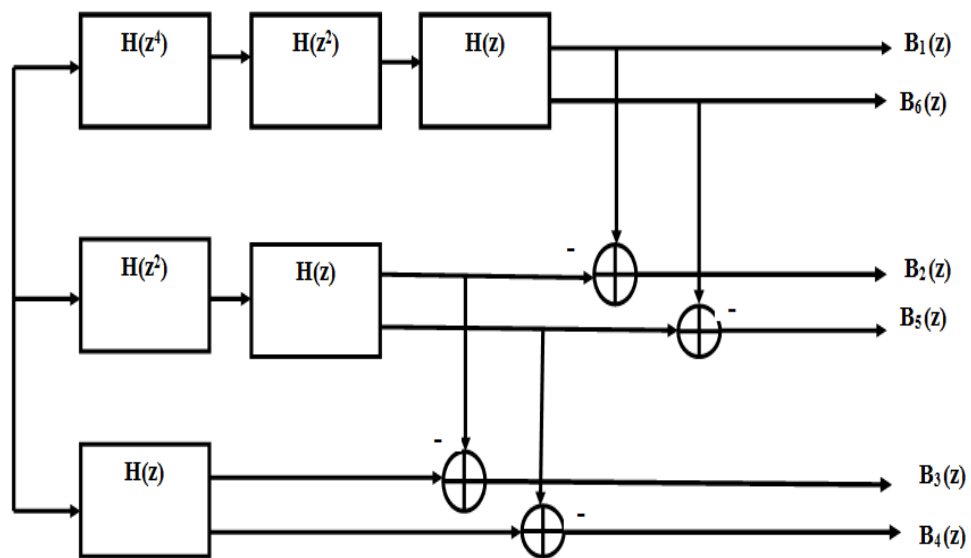
Sub-Band	Transfer function
B <sub>1</sub>	$H(z^4)H(z^2)H(z)$
B <sub>2</sub>	$H(z^2)H(z)- H(z^4)H(z^2)H(z)$
B <sub>3</sub>	$H(z)- H(z^2)H(z)$
B <sub>4</sub>	$H_c(z)-H(z^2)H_c(z)$
B <sub>5</sub>	$H(z^2)H_c(z)-H(z^4)H(z^2)H_c(z)$
B <sub>6</sub>	$H(z^4)H(z^2)H_c(z)$

The prototype filter  $H(z)$  is designed in MATLAB using the least square method. The dependency of stop band attenuation on transition bandwidth was verified and is shown in Fig.5.1. for a 40 tap filter. The simulation study to identify the dependence of transition bandwidth on stop band attenuation shows that a minimum transition bandwidth of 0.2 radians should be maintained to achieve the 50 dB stop band attenuation.



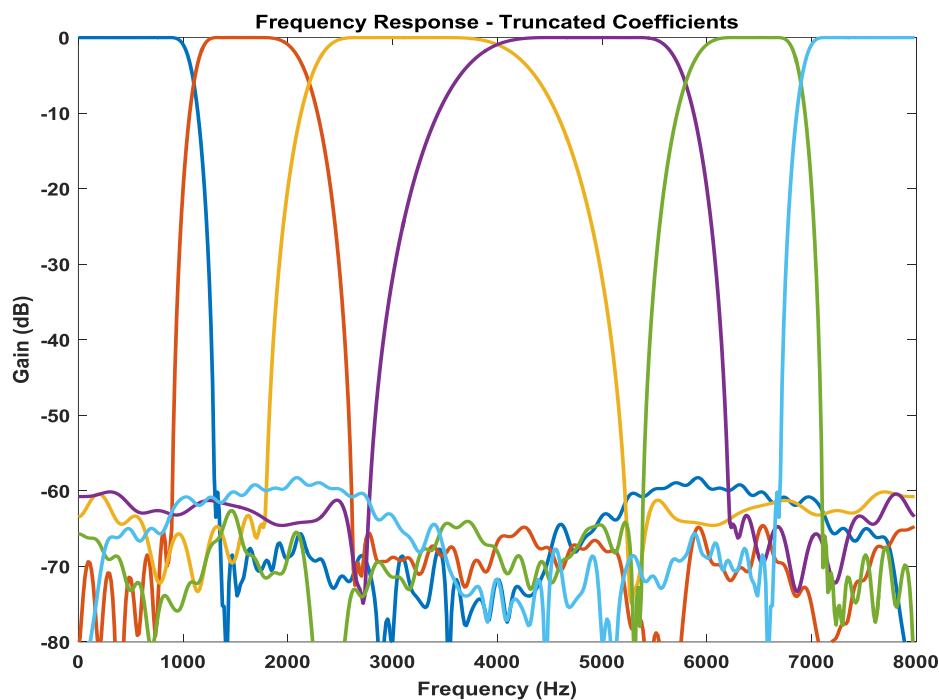
**Fig. 5.1** The variation of minimum stop-band attenuation with increasing Transition Bandwidth

The normalized transition bandwidth of the prototype  $H(z)$  is fixed as 0.2 and the order of the filter is selected as 40 to ensure a minimum stop band attenuation of 50 dB which is sufficient for hearing aids to compensate for the hearing loss satisfying the dynamic range requirements of the hearing impaired person. The structure of the 6-band filter bank is shown in Fig.5.2. The lower bands are complementary bands of the upper bands formed by replacing  $H(z)$  with its complement  $H_c(z)$  as the masking filter. The multipliers used in the circuit are integer multipliers. Hence the integer filter coefficients are generated by scaling the real filter coefficients by a scaling factor  $2^n$  and rounded off to the nearest integer. The filtering process is the convolution of impulse response of the filter with the input audio samples.



**Fig. 5.2** Structure of 6-band Filter Bank

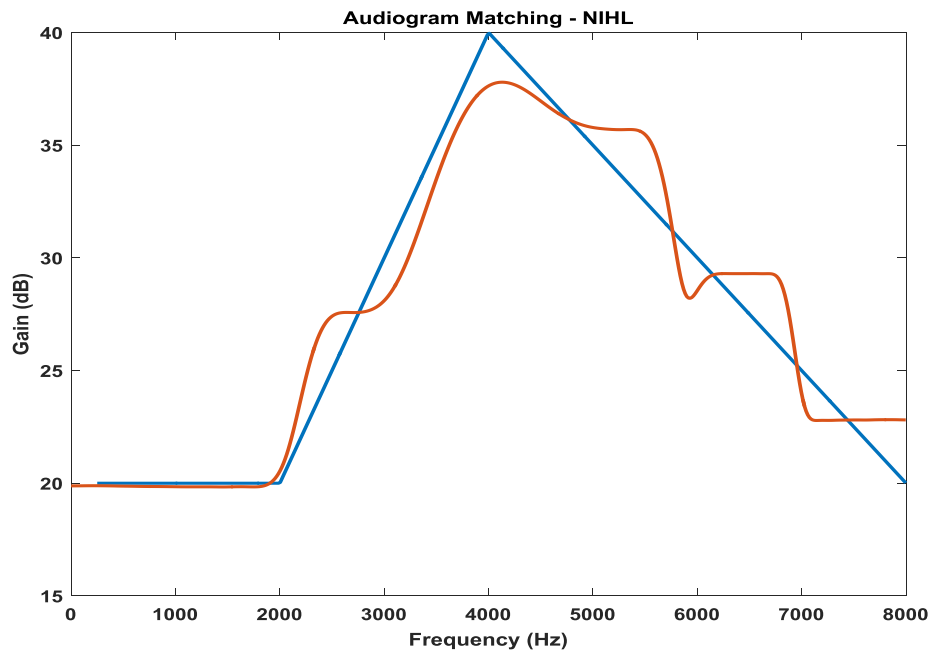
Approximate multipliers are used to compute the MAC operation of the convolution sum which in effect can be interpreted as the convolution of approximate input samples with an approximate impulse response. The approximate filter thus results in a frequency response with less attenuation in the stop band. The frequency response of the 6-band filter bank designed in MATLAB is shown in Fig.5.3.



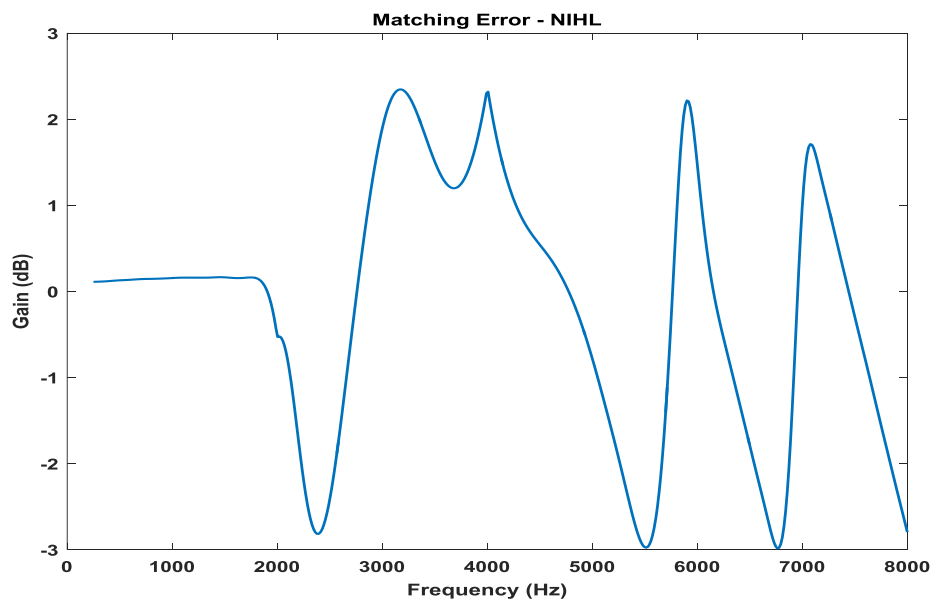
**Fig. 5.3** The frequency response of the 6-Band Approximate Filter Bank

### 5.1. Audiogram Matching

The 6-band approximate FIR filter bank is simulated in MATLAB and the gains of the different bands are adjusted to match the audiogram of hearing impaired. Fig. 5.4a shows the audiogram matching result of the experiment. The audiogram of a patient with Noise Induced Hearing Loss (NIHL) is selected for the experiment. The results show that the proposed 6-band approximate filter is a suitable one for the implementation of the hearing aids. The matching error is plotted in Fig. 5.4b and is within  $\pm 3$  dB.



(a)

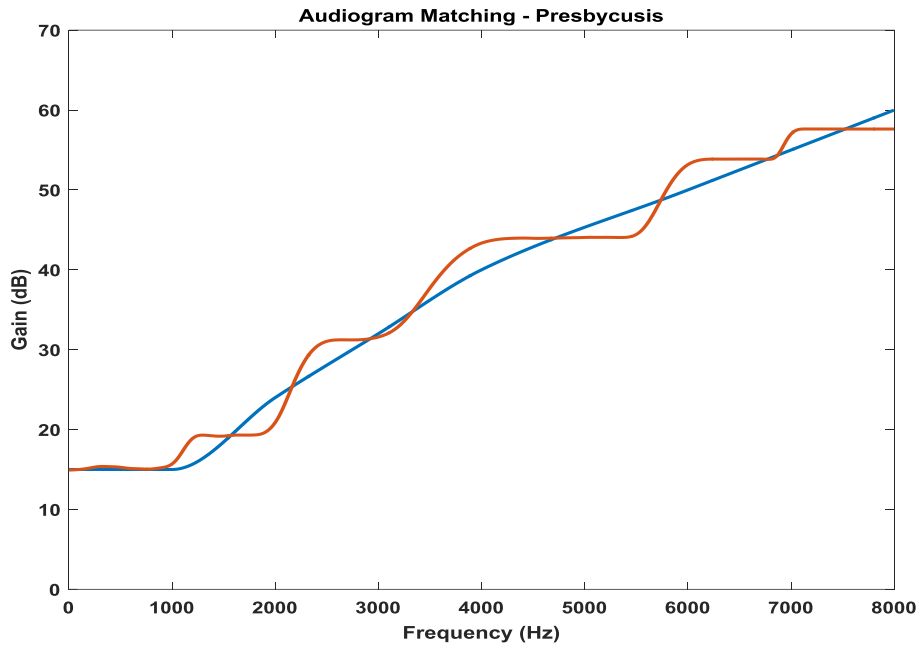


(b)

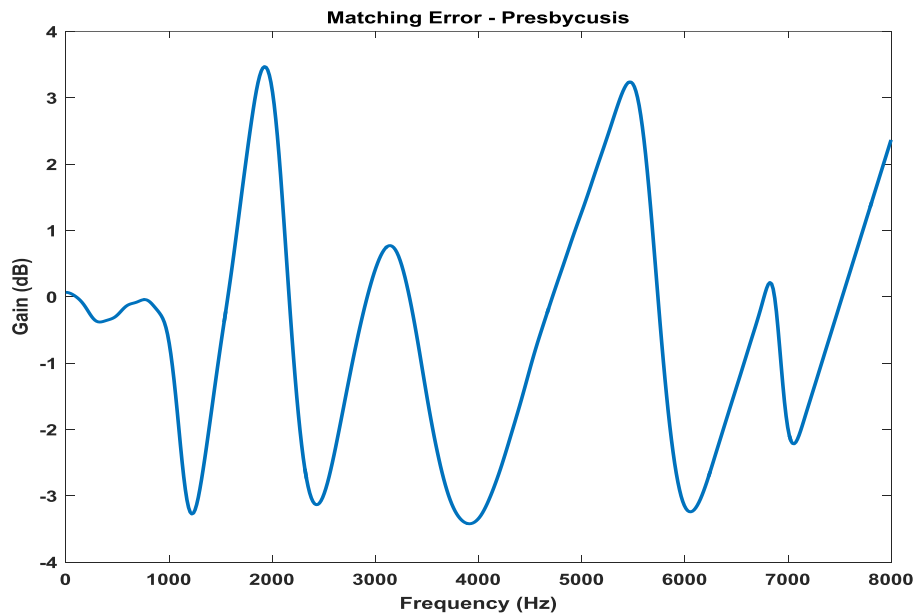
**Fig.5.4** Matching Example 1

- (a) NIHL audiogram matching for the proposed Approximate filter bank
- (b) Plot of Matching Error

Another matching experiment is performed using old age related presbycusis hearing loss. Fig 5.5a shows the plot of audiogram matching and the corresponding matching error plot are shown in Fig. 5.5b. Results show that matching error can be adjusted within +/- 4dB for the particular case considered.



(a)



(b)

**Fig. 5.5** Matching Example 2

(a) Presbycusis audiogram matching for the proposed Approximate filter bank

(b) Plot of Matching Error

### 5.2. Area and Power Savings

The area and power savings achieved for the approximate filter bank are evaluated when using proposed approximate multiplier for the digital hearing aid application. Table 5.2 gives the area and power savings achieved in the 6-band digital filter bank designed using with and without the utilization of proposed approximate multiplier. The filter-bank is synthesized in RTL compiler and post-layout design parameters are taken from cadence SOC Encounter.

**Table 5.2** Area and Power savings achieved for the proposed Approximate Filter Bank

Exact Design		Proposed Approximate Design		Savings Achieved	
Area ( $\mu\text{m}^2$ )	Power (mW)	Area ( $\mu\text{m}^2$ )	Power (mW)	Area (%)	Power (%)
487233	7.170	374268	6.17	23	14

The power and area results show the advantage of the proposed approximate design in comparison with exact design. The approximate filter bank consumes 14% less power and 23 % less area than the filter bank designed using exact multiplier.

## 6. CONCLUSION

In this paper, we proposed an area and power efficient approximate multiplier based on simple shifting operation to approximate a number for multiplication. The proposed multiplier consumes less area, power, and in energy compared to exact and other recently published approximate multipliers. By adopting frequency response masking approach, a 6-band non-uniform digital FIR filter bank is developed using approximate multiplier for hearing aid application. Audiogram matching is done with audiograms of two different types of hearing losses and the matching error is computed. It is found that the design offer comparable matching error performance with that of hearing aids implemented with exact arithmetic filter banks reported elsewhere. The large attenuation in the stop band ensures high programmability of the magnitude response of the filter bank, which may be utilized to attain arbitrary reduction of the matching error. The customary low clock rates of digital audio devices makes allowance for time sharing of the resources which may further improve the hardware efficiency.

## REFERENCES

1. Itoh, K., Yamaoka, M., & Oshima, T. (2010). Adaptive circuits for the 0.5-V nanoscale CMOS era. *IEICE transactions on electronics*, 93(3), 216-233, <https://doi.org/10.1587/transele.E93.C.216>
2. Nowak, E. J. (2002). Maintaining the benefits of CMOS scaling when scaling bogs down. *IBM Journal of Research and Development*, 46(2.3), 169-180, <https://doi.org/10.1147/rd.462.0169>
3. Lim, Y. (1986). Frequency-response masking approach for the synthesis of sharp linear phase digital filters, *IEEE transactions on circuits and systems*, 33(4), 357-364, <https://doi.org/10.1109/TCS.1986.1085930>
4. Lian, Y., Zhang, L., & Ko, C.C. (2001). An improved frequency response masking approach for designing sharp FIR filters, *Signal processing*, 81(12), 2573-2581, [https://doi.org/10.1016/S0165-1684\(01\)00149-9](https://doi.org/10.1016/S0165-1684(01)00149-9)
5. Lian, Y., & Wei, Y. (2005). A computationally efficient non uniform FIR digital filter bank for hearing aid, *IEEE Transactions on Circuits and Systems I: Regular Papers*, 52 (12), 2754-2762, <https://doi.org/10.1109/TCSI.2005.857871>
6. Haridas, N., & Elias, E. (2016). Efficient variable bandwidth filters for digital hearing aid using Farrow structure, *Journal of advanced research*, 7(2), 255-262, <https://doi.org/10.1016/j.jare.2015.06.002>



7. Deng, T.B. (2010). Three-channel variable filter-bank for digital hearing aids, *IET signal processing*, 4(2), 181-196, <https://doi.org/10.1049/iet-spr.2008.0164>
8. Setiawan, R., Lesmana, V.P., & Gwee, B.H. (2005). Design and Implementation of A Low Power FIR Filter Bank, *Journal of The institution of Engineers, Singapore*, 45(5), 77-87.
9. Wei, Y., & Liu, D. (2013). A reconfigurable digital filterbank for hearing-aid systems with a variety of sound wave decomposition plans. *IEEE transactions on Biomedical Engineering*, 60(6), 1628-1635, <https://doi.org/10.1109/TBME.2013.2240681>
10. Lin, C. H., Chang, K. C., Chuang, M. H., & Liu, C. W. (2012). Design and implementation of 18-band Quasi-ANSI S1.11 1/3-octave filter bank for digital hearing aids. In *2012 IEEE International Symposium on VLSI Design, Automation, and Test (VLSI-DAT)*, pp. 1-4, <https://doi.org/10.1109/VLSI-DAT.2012.6212620>
11. Wei, Y., & Wang, Y. (2015). Design of low complexity adjustable filter bank for personalized hearing aid solutions. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(5), 923-931, <https://doi.org/10.1109/TASLP.2015.2409774>
12. Jiang, H., Han, J., & Lombardi, F. (2015). A comparative review and evaluation of approximate adders. In *Proceedings of the 25th edition on Great Lakes Symposium on VLSI*, pp. 343-348, <https://doi.org/10.1145/2742060.2743760>
13. Zhu, N., Goh, W. L., Zhang, W., Yeo, K. S., & Kong, Z. H. (2010). Design of low-power high-speed truncation-error-tolerant adder and its application in digital signal processing. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 18(8), 1225-1229, <https://doi.org/10.1109/TVLSI.2009.2020591>
14. Jiang, H., Liu, C., Maheshwari, N., Lombardi, F., & Han, J. (2016) A comparative evaluation of approximate multipliers. In *2016 IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH)*, Beijing, China, 191-196, <https://doi.org/10.1145/2950067.2950068>
15. Jiang, H., Liu, C., Liu, L., Lombardi, F., & Han, J. (2017). A review, classification, and comparative evaluation of approximate arithmetic circuits. *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, 13(4), 60:1–60:34, <https://doi.org/10.1145/3094124>
16. Schulte, M.J., & Swartzlander, E. (1993). Truncated multiplication with correction constant [for DSP]. In *Proceedings of IEEE Workshop on VLSI Signal Processing VI*, Veldhoven, Netherlands, 388–396, <https://doi.org/10.1109/VLSISP.1993.404467>
17. Kyaw, K. Y., Goh, W. L., & Yeo, K. S. (2010). Low-power high-speed multiplier for error-tolerant application. In *2010 IEEE International Conference on Electron Devices and Solid-State Circuits (EDSSC)*, pp.1-4, <https://doi.org/10.1109/edssc.2010.5713751>
18. Kulkarni, P., Gupta, P., & Ercegovic, M. (2011, January). Trading accuracy for power with underdesigned multiplier architecture. In *Proceeding of 24th International Conference on VLSI Design*, Chennai, India, 346–351, <https://doi.org/10.1109/VLSID.2011.51>
19. Zendegani, R., Kamal, M., Bahadori, M., Afzali-Kusha, A., & Pedram, M. (2017). RoBA multiplier: A rounding-based approximate multiplier for high-speed yet energy-efficient digital signal processing. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 25(2), 393-401, <https://doi.org/10.1109/TVLSI.2016.2587696>
20. Jiang, H., Han, J., Qiao, F. & Lombardi, F. (2016). Approximate radix-8 booth multipliers for low-power and high-performance operation. *IEEE Transactions on Computers*, 65(8), 2638-2644, <https://doi.org/10.1109/TC.2015.2493547>

21. Venkatachalam, S., & Ko, S.B. (2017). Design of Power and Area Efficient Approximate Multipliers. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 25(5), 1782-1786, <https://doi.org/10.1109/TVLSI.2016.2643639>
22. Narayanamoorthy, S., Moghaddam, H.A., Liu, Z., Park, T., & Kim, N.S. (2015). Energy-efficient approximate multiplication for digital signal processing and classification applications. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 23(6), 1180-1184, <https://doi.org/10.1109/TVLSI.2014.2333366>
23. Hashemi, S., Bahar, R., & Reda, S. (2015). Drum: A dynamic range unbiased multiplier for approximate applications. In *Proceedings of the IEEE/ACM International Conference on Computer Aided Design (ICCAD), Austin, TX, USA*, 418–425, <https://doi.org/10.1109/ICCAD.2015.7372600>
24. Chandrakasan, A. P., Potkonjak, M., Mehra, R., Rabaey, J., & Brodersen, R. W. (1995). Optimizing power using transformations. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 14(1), 12-31, <https://doi.org/10.1109/43.363126>
25. Liang, J., Han, J., & Lombardi, F. (2013). New metrics for the reliability of approximate and probabilistic adders. *IEEE Transactions on Computers*, 62(9), 1760-1771, <https://doi.org/10.1109/TC.2012.146>
26. Sengupta, D., & Saleh, R. (2007). Generalized power-delay metrics in deep submicron CMOS designs. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 26(1), 183-189, <https://doi.org/10.1109/TCAD.2006.883926>