An Efficient Filter Design for Active Noise Control System

Silpa R. Sajeev[1], Shinoj K. Sukumaran [2]

[1] PG Scholar, Dept. of Electronics and Communication Engineering, Government Engineering College, Idukki, Kerala, India
[2]Assistant Professor, Dept. of Electronics and Communication Engineering, Government Engineering College, Idukki, Kerala, India

---***---

Abstract - *Active noise control (ANC) is a method to eliminate noise in many applications such as industries, automobiles, home appliances, etc. The system mainly consists of an adaptive filter section and is implemented using a series of adders and multipliers. In this work, several adders and multipliers are compared and the optimized ones are chosen for speed and area efficiency. In the proposed system, conventional adders and multipliers are replaced with high speed area efficient spanning tree adders and modified Vedic multipliers. Comparison of the proposed system with the existing one shows a hardware efficiency of 4.8% and delay reduction of 40.41%. The system simulation is validated in Xilinx 14.2 ISE and synthesized in Xilinx Virtex-5 FPGA.*

Key Words: **Active Noise Control, Vedic multipliers, Parallel Prefix Adders and Xilinx**

1. INTRODUCTION

Active noise control (ANC) is a method that is used to suppress audio noise. This system uses an array of secondary sources to produce an antinoise wave, which cancels the noise wave. The antinoise signal is a wave with same amplitude and opposite phase to that of the noise signal. ANC works on the principle of superposition. The anti noise signal will be made to superimpose with the noise signal, and effectively they will cancel and a noise free output will be obtained. The secondary sources will be connected by an appropriate signal processing algorithm. Many applications find use of ANC such as in industries, automobiles etc.

Many modifications were introduced in this field of ANC. Conventional active noise cancelling (ANC) systems often perform well in reducing the low-frequency noise. Due to the popularity of digital filters, the existing ANC systems often use high-speed digital signal processors to cancel disturbing noise. But, a digital controller often has additional delay due to several processes such as that in DSPs, analog to digital converters (ADC), digital to analog converters (DAC), etc. This delay may affect several design constraints in ANC systems and affects the stability and performance of the system. This work focuses on providing an area and delay efficient filter design for ANC system. The conventional adders and multipliers were also replaced by efficient systems, increasing the overall performance of the system.

2. CONVENTIONAL ACTIVE NOISE CONTROL SYSTEM

Active noise control (ANC) employs an electro-acoustic system to cancel the primary noise based on the principle of superposition. The system block diagram is shown in Fig -1.

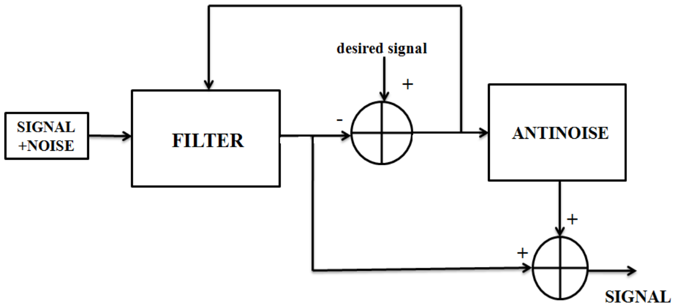

Fig - 1: Block diagram of the conventional system

The main part of the system is a filter section to extract the signal. The input to the filter is a signal contaminated with noise. The filter out may be incorrect due to the presence of noise. The filter out will be subtracted from a desired signal in order to get the error signal. The filter weights will be updated to adapt to signal. An antinoise signal will be generated to combine with the filter output to obtain the error free signal. Main portion of the system are adders and an adaptive filter section.

2.1 Adaptive Filter

Adaptive filters are linear filters with a provision to update filter weights according to certain optimization algorithm. It consists of a filter section (IIR/FIR) and a weight updating part. The basic circuit of an adaptive filter is shown in Fig -2. The filter normally gives the convolution of input and weights as the output y(n). The aim is to get y(n) so close to desired signal d(n). As 'n' changes, the system adapts the weights so as to get y(n) a close estimation of d(n).

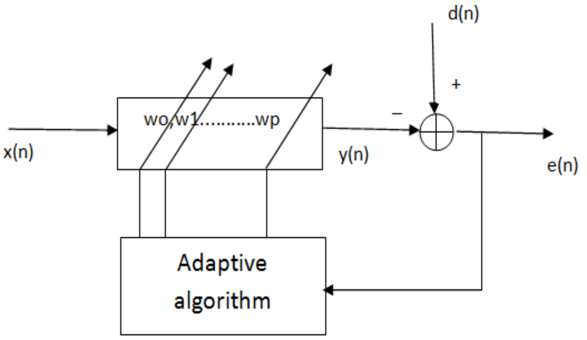

Fig - 2: Adaptive Filter Circuit

In the filter section, a controller provides the necessary control signals and addresses to the storage buffers. It consists of two counters, enabled to count from 0 to M-1, where M is the filter length. Here, the filter length is taken as 24. Therefore, the counters will count from 0 to 23. One comparator is included to disable the count for one cycle, so that the new value can be stored every 24th cycle. The hardware of controller circuit is shown in Fig -3.

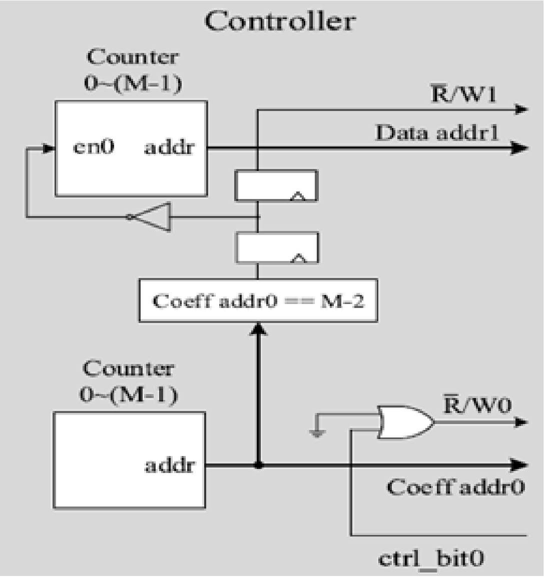

Fig - 3: Controller circuit [1]

Storage buffers are the main elements of a filter to store the data and filter coefficients. In order to accomplish the basic filter operation, the coefficients need to be fetched from the buffer sequentially from memory location 0 to M-1. The data has to be also taken from the buffer. So, whenever a new data enters the line, each data value stored in the linear buffer has to be moved by one place, and thereby increases the performance overhead. Therefore, a circular buffer has been used. The circular buffer outputs the old data and replaces the old data by the new one. Therefore, only one data needs to be replaced and hence, effectively reduces the delay.

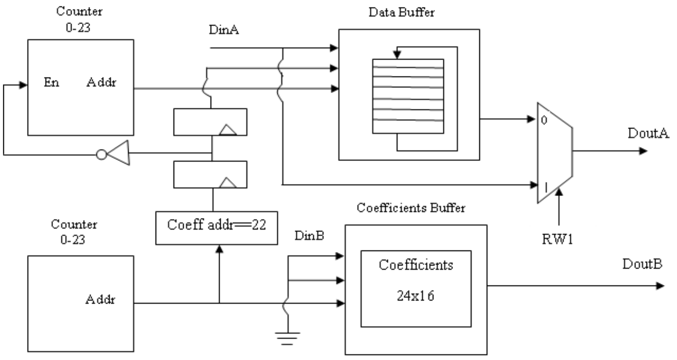

Fig – 4: Hardware architecture of circular buffer [1]

The controller provides necessary addresses and control signals. When the R/W signal is enabled, the new data will be entered into the buffer. The old data will be taken out in one clock cycle and the data will be entered during the next clock cycle only. In order to get the new data during the same cycle itself, a 16 bit 2x1mux is used.

The most essential part of the filter is the Multiply Accumulate (MAC) unit for implementing the filter equations. The data and coefficients will be simultaneously taken from the memory to the registers simultaneously and then multiplied using an array multiplier. The partial products will be then accumulated using a carry look- ahead adder in order to satisfy the filter equation.

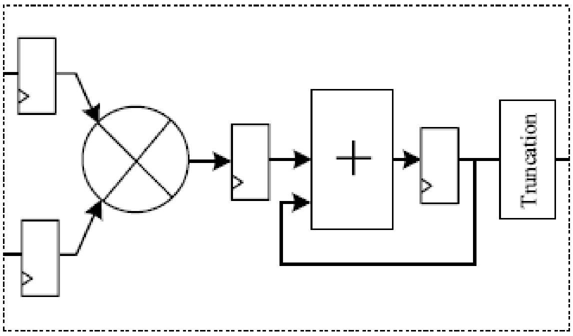

Fig – 5: Multiply Accumulate Unit hardware [1]

The weights of the filter are updated in order to adapt to the situation. The new weights are calculated according to the equation:

$$w(n+1) = w(n) + \mu e(n)x(n) \qquad \text{....... (1)}$$

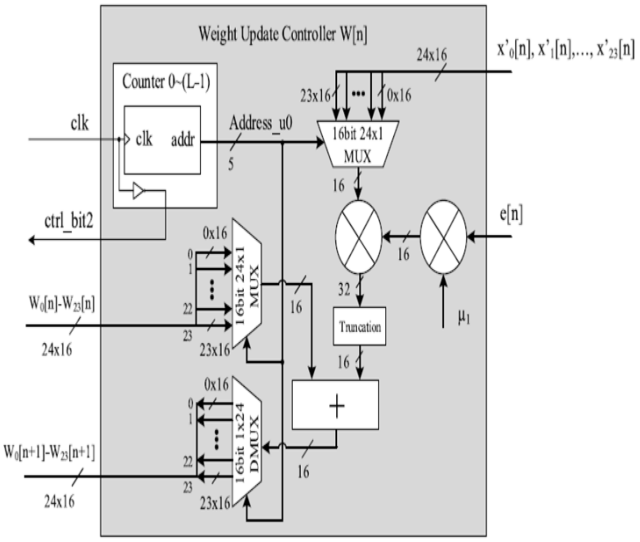

Fig – 6: Weight updation block[1]

The weight updation block consists of a two 16 bit 24x1 multiplexers are used to select the corresponding data and weight. A counter provides the necessary select signal. The weight updation is performed and stored to the corresponding address with the help of a 16bit 1x24 bit demultiplexer.

2.2 Adders

The adders used in the system are carry look-ahead (CLA) adders. The generate and propagate signals are calculated as:

P =A ^ B ..... (2)

G =A & B ..... (3)

The sum will be calculated as:

$$S_i = A_i \hat{} B_i \hat{} C_{i-1} \quad ..... (4)$$

3. PROPOSED SYSTEM

The adders in the whole system are replaced with high speed spanning tree adder and the filter section is modifies by replacing all adders and multipliers by spanning tree adders and modified Vedic multipliers for improvement in area and delay efficiency.

3.1 Vedic multiplier

Vedic multiplication is an ancient method mentioned in Adarvaveda[2]. It provides a fast and easier method for multiplication of two numbers. Normal Vedic multiplication is done using the concept of Urdhva-tiryakbyham. To multiply two 2-bit numbers, the vertical and cross wise operant bit products are taken as shown in Fig -7. The LSBs of the two operands will be multiplied to get the LSB of the product, the cross products of the bits will be taken for getting the next MSB, and finally, the product of the MSBs along with the carry will give the MSB of the product. This method was considered speeder than conventional multipliers.

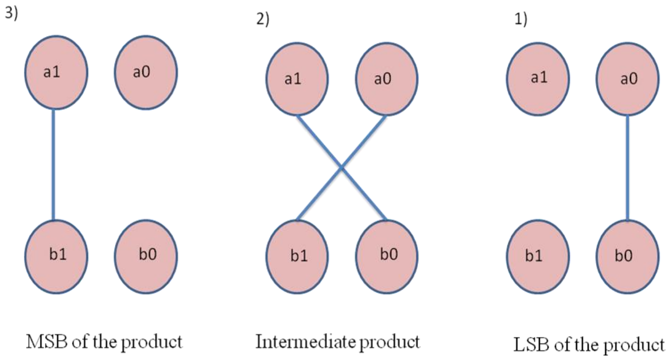

Fig – 7: Vedic multiplication of two 2-bit numbers [2]

The multiplication of two 3-bit numbers is depicted in Fig -8 which involves 5 steps. The comparison of Fig. 7 and 8 shows that the number of steps involved in multiplication will increase as the bit size increases. Therefore, the ancient Vedic multiplication has been modified.

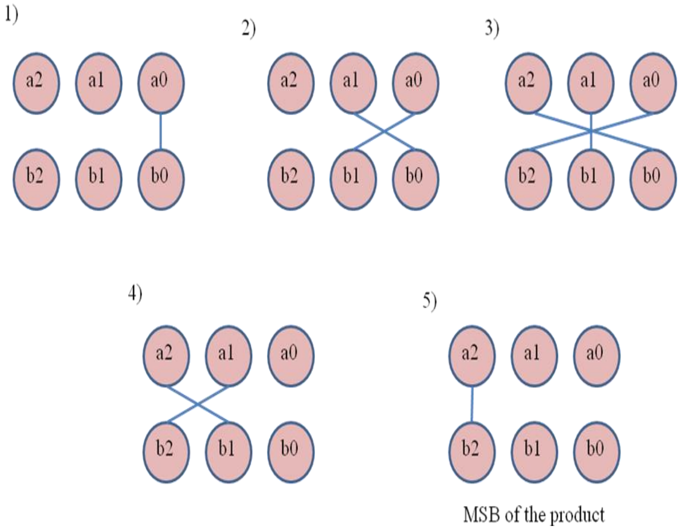

Fig –8: Vedic multiplication of two 3-bit numbers [2]

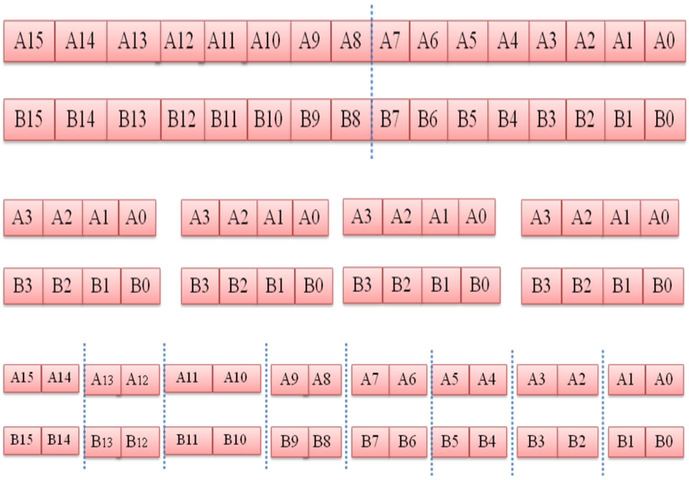
A15 A14 A13 A12 A11 A10 A9 A8 A7 A6 A5 A4 A3 A2 A1 A0
B15 B14 B13 B12 B11 B10 B9 B8 B7 B6 B5 B4 B3 B2 B1 B0
A3 A2 A1 A0
B3 B2 B1 B0

Fig – 9: Modified Vedic multiplication of two 16-bit numbers.

The adders in the whole system are replaced with spanning tree adder. Spanning tree adder is a type of parallel prefix adder (PPA) which has been used to improve the area, delay and power efficiency [3]. The two main elements of a parallel prefix adder are black cells (BC) and grey cells (GC). The number of steps involved in generating the carry signal is reduced in PPAs as compared to CLA. In PPAs, the delay is logarithmically proportional to the adder width.

PPA consists of 3 stages: a precomputation stage, prefix stage and final computation stage. In the precomputation stage, the generate and propagate signals are generated for the two inputs using the equations (1) and (2). In the prefix stage, bitwise generate and propagate signals are generated using the black cells and grey cells. The BCs consists of two AND operations and one OR operation, giving both propagate and generate signals, whereas the GCs have only one AND gate and one OR gate giving the generate signals alone. Only the generate signals are necessary during the final stage for carry generation. Therefore GCs are used in final stage so that the area can be reduced. Finally, the sum will be generated. Fig -10 shows the circuit of a spanning tree adder. It consists of 16 full adders, 10 black cells and 3 grey cells.

4. RESULTS AND DISCUSSION

The system is designed using Xilinx ISE 14.2 Design Suite and implemented in Spartan-3E FPGA Basys2 board. Simulation result, RTL, schematic, device utilization summary and power consumed by the system is discussed in this section.

The array multiplier and Vedic multiplier were compared and results were obtained. The comparison is shown in chart -1 . The number of slice LUTs, IOBs and slices has been reduced drastically providing an overall reduction in area. The system delay has also been reduced. From the graph, it can be seen that an area efficiency of 89.2% and delay reduction of 79.6% has been achieved for the multiplier module.

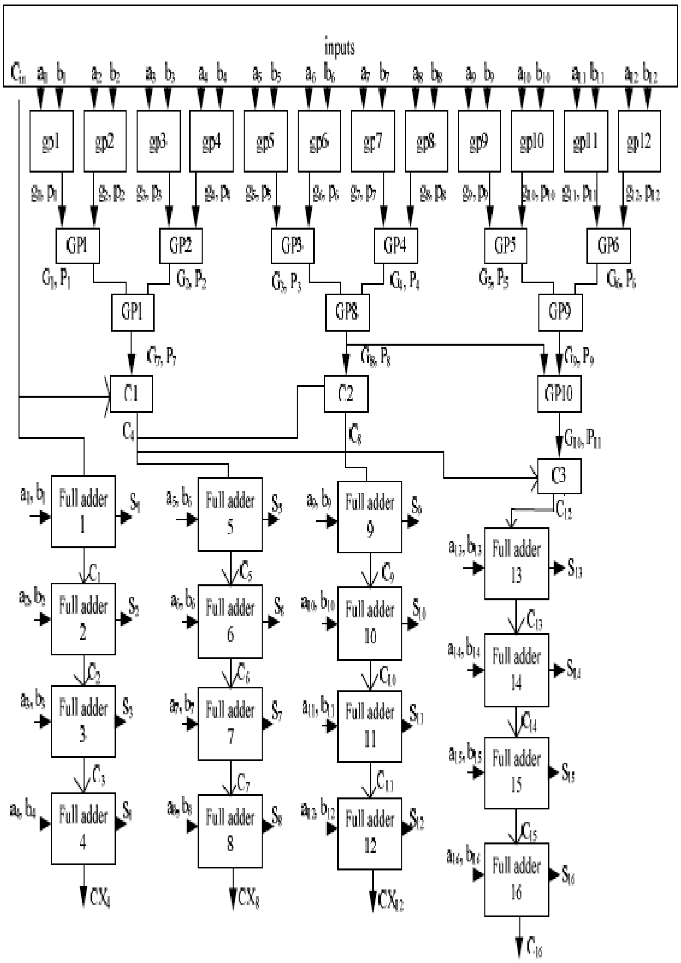

Fig – 10: Spanning tree adder [3]

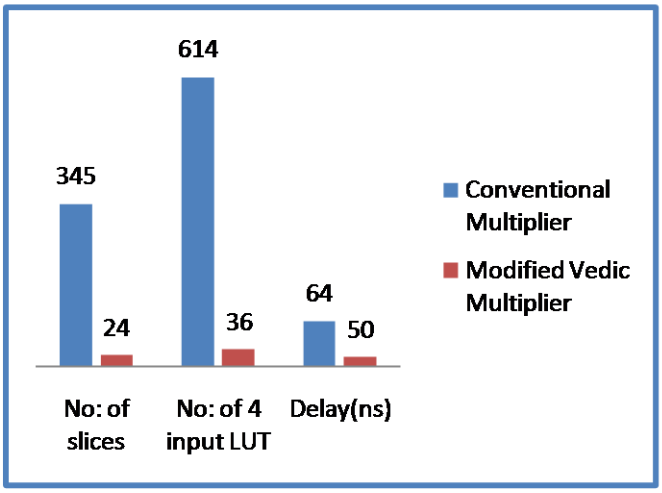

Chart - 1: Area and delay comparison of multipliers.

The comparison between four 16 bit parallel prefix adders- Koggestone adder, Brent Kung adder, Sparse Koggestone adder and Spanning tree adders- with a 16- bit CLA was done. All the parallel prefix adders utilizes less area than CLA, but spanning tree adders shows optimization in both area and delay. STA shows 16.5% area efficiency and 21% delay reduction.

The conventional ANC system and modified system has been compared and the results are shown in Table -1. The number of slice LUTs has been reduced from 746 to 710, which shows an area improvement. Also, there is a large reduction in case of delay, dropping from 22.382ns to 13.337ns.

Table - 1: Comparison of conventional and modified ANC system

Properties	Conventional system	Modified system
No: of slices	872	872
No: of slice LUTs	746	710
No: of bonded IOBs	35	35
Delay(ns)	22.382	13.337

Chart -2 shows the comparison between the conventional and modified systems. The modified system shows a hardware efficiency of 4.8% and delay reduction of 40.41%.

The RTL of the systems are also taken, which shows the interconnection between various blocks. RTL of proposed ANC system is shown in Fig - 11. It consists of the controller circuit comprising of the counters and comparators, data buffer, coefficient buffer, the multiply accumulate unit, weight updation block with its multiplexers and demultiplexers, the Vedic multiplier and adders. The 16 bit vedic multiplier consists of an 8-bit multiplier with a 4 bit multiplier. A 2 bit multiplier is the basic block of all other multipliers. The adder block shows the spanning tree adder, with 16 full adders, 10 black cells and 3grey cells.

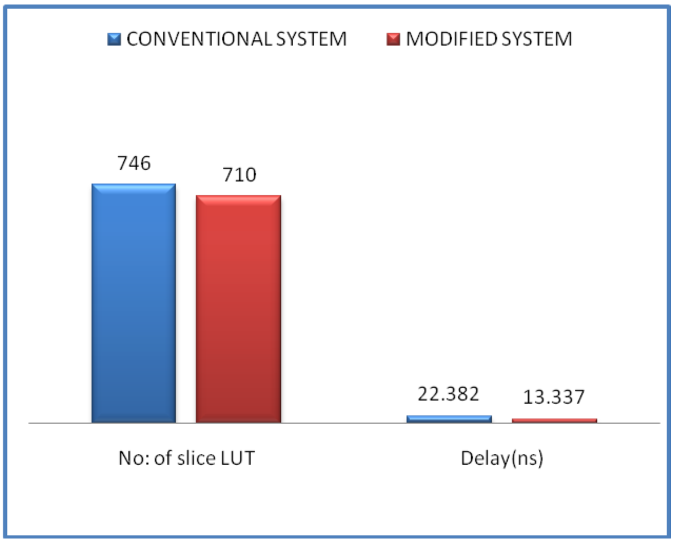

Chart – 2: Comparison of conventional and modified ANC systems.

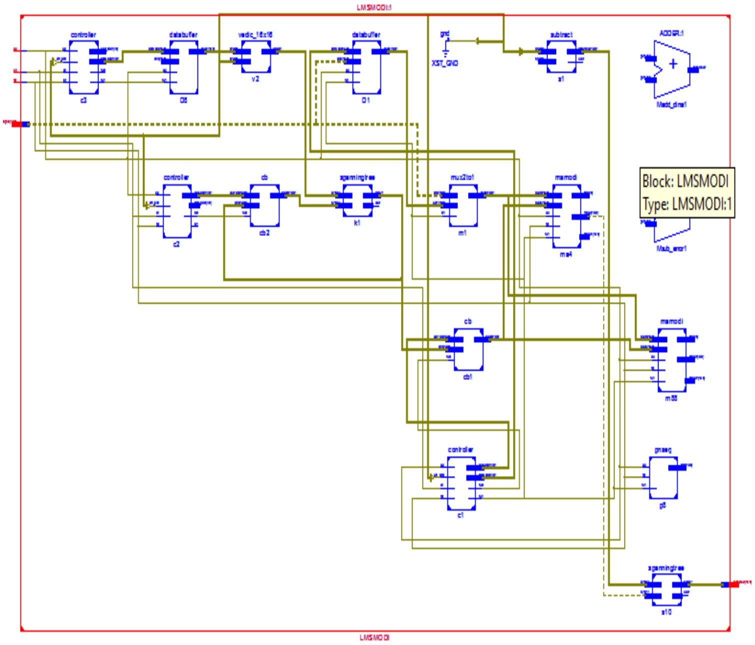

Fig –11: RTL of modified ANC system

5. CONCLUSIONS

Active noise control is a method used for noise suppression. An efficient filter for active noise control system has been proposed. The multipliers and adders of the system are replaced with high speed modified Vedic multipliers and spanning tree adders. Several parallel prefix adders have been studied and finally spanning tree adders are chosen for efficiency. A comparison of conventional and proposed active noise control system is done. A hardware efficiency of 4.8% and delay reduction of 40.41% is obtained.

REFERENCES

[1] Hong-Son Vu and Kuan-Hung Chen, "A Low-Power Broad-Bandwidth Noise Cancellation VLSI Circuit Design for In-Ear Headphones.", IEEE *Trans. Very Large Scale Integration (VLSI) Systems, vol. 24, no. 6, pp. 2013-2025, June 2016.*

[2] Udit Narula, Rajan Tripathi and Garima Wakhle,"High Speed 16-bit Digital Vedic Multiplier using FPGA " in *2nd International Conference on Computing for Sustainable Global Development ,2015,pp. 121-124.*

[3] Sudheer Kumar Yezerla and B Rajendra Naik,"Design and Estimation of delay, power and area for Parallel prefix adders", *in Proceedings of 2014 RAECS UIET Punjab University Chandigarh, 06 - 08 March, 2014*

[4] S. M. Kuo and D. R. Morgan, "Active noise control: A tutorial review,"*Proc. IEEE, vol. 87, no. 6, pp. 943–973, Jun. 1999.*

[5] X. Kong and S. M. Kuo, "Study of causality constraint on feedforward active noise control systems," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process., vol. 46, no. 2, pp. 183–186, Feb. 1999.*

[6] Y. Song, Y. Gong, and S. M. Kuo, "A robust hybrid feedback active noise cancellation headset," *IEEE Trans. Speech Audio Process., vol. 13, no. 4,* pp. 607–617, Jul. 2005.

[7] S. M. Kuo, S. Mitra, and W.-S. Gan, "Active noise control system for headphone applications," *IEEE Trans. Control Syst. Technol., vol. 14,* no. 2, pp. 331–335, Mar. 2006.

[8] C.-Y. Chang and S.-T. Li, "Active noise control in headsets by using a low-cost microcontroller," *IEEE Trans. Ind. Electron., vol. 58, no. 5,*pp. 1936–1942, May 2011.

[9] L. Zhang, L. Wu, and X. Qiu, "An intuitive approach for feedback active noise controller design," *Appl. Acoust., vol. 74, no. 1, pp. 160–168,* Jan. 2013.

[10]L. Wu, X. Qiu, and Y. Guo, "A simplified adaptive feedback active noise control system," *Appl. Acoust., vol. 81, pp. 40–46, Jul. 2014.*

[11]K.-K. Shyu, C.-Y. Ho, and C.-Y. Chang, "A study on using microcontroller to design active noise control systems," in *Proc. IEEE Asia Pacific Conf. Circuits Syst. (APCCAS), Nov. 2014, pp. 443–446.*

[12]Douglas S.C, Introduction to Adaptive Filters, Digital Signal Processing Handbook., CRC Press LLC, 1999.

[13]Implementation of 4 bit array multiplier using Verilog HDL and its testing on the Spartan 2 FPGA.

[14] Advanced Digital Design with Verilog HDL ,second edition-Micheal D Ciletti.