

# Hardware Implementation of Low Power Audio Sub word Sorter Unit for High Security Transmission

P Karthigaikumar MIACSIT, MIAENG<sup>1</sup>, K Baskaran MIEEE<sup>2</sup>

**Abstract**— The security of audio data in high end communication applications like satellites and radars is an issue of concern these days. Designing a processor at the chip level for this requirement is by itself a challenge to VLSI engineers. This paper aims to design a HDL based novel audio subword sorter unit, which is less complex in structure and highly efficient in terms of security. In this paper, we examine the hardware implementation of powerful permutation instruction group (GRP) in low power. This is done at the integrated chip (IC-level) using Verilog HDL and can be implemented in FPGA. To our knowledge this is the first audio subword sorter unit implemented in FPGA

**Index Terms**— Cryptography, permutation, audio subword sorter, Network Security, multimedia

## I. INTRODUCTION

In most cases, multimedia data are packed into subwords of one or two bytes that are processed in parallel in word oriented processors as per the single instruction multiple data (SIMD) [5] [8]. This is called subword parallelism. In order to fully exploit the subword parallel operations, the subwords need to be efficiently rearranged inside the registers in order to enhance the permutation. Efficient handling of permutations [9] is also needed for the software implementation of cryptographic algorithms in order to achieve the needed throughput. The selection of efficient permutation instructions and design of fast permutation units have gained a lot of interest [6][7][9][10].

As per GRP, the data bits that are associated with the control bits equal to one are concentrated to the left side of the output. Similarly, the data bits that are associated with the control bits equal to zero are concentrated on the right side of the output. This action resembles a sorting operation for the control bits, where the largest bits that are equal to one are gathered to the left. Therefore the problem of designing a hardware unit that executes GRP is equivalent to the design of a sorting network that sorts the control bits and moves the data bits appropriately. Recent work has until designing a permutation unit for audio data sorting which is given in [4].

## II. ENHANCED MERGE SORTER NETWORK (EMS)

As in enhanced bi-tonic sorting network EBSN [3], the Merge Sorter Network shown in fig.1 contains connected sub

networks called Enhanced Merge Sorters (EMS) whose purpose is analogous to the EBS unit. Also only  $n/2$  MSB's are sufficient to guide the data rearrangement. The  $n/2$  most significant control bits that are equal to one, suggest that no exchange be done at these locations because the data present in those locations should not move and is fixed. If any of the bits of the  $n/2$  MSB in the control word are zero, then their corresponding  $n/2$  LSB bits cannot be swapped. This is because the locations are of no relative significance and once swapped the corresponding bits in  $n/2$  LSB become relatively insignificant. Hence it is not intended to bring them back to right side again. A signal called 'barrier' is introduced after the first level and which depict these two constraints. The structure of the EMSN is shown in fig.1.

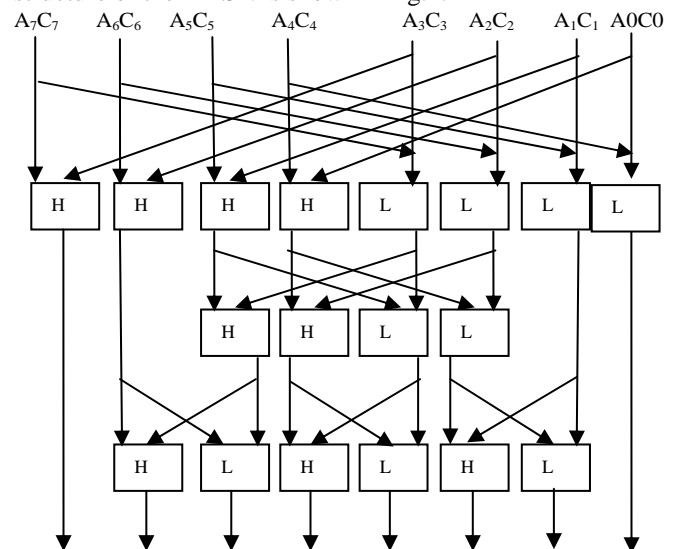


Fig. 1 Structure of Enhanced Merge Sorter Network

## III. MODIFIED ENHANCED MERGE SORTER NETWORK (MEMS)

In this paper, the audio data is a 16 bit signal coming into the system in real time. This data is divided into two halves since the processor is an 8 bit processor. The advantage of 8 bit processor over higher bit processor is given in [11]. Each half is fed to the Enhanced Merge Sorting as the input and the sorted data is sent to the transmission line. The main idea of the enhancement is to incorporate some video code signals into the locations which are free and are of no relative significance.

The main strength of Enhanced Merge Sorting Network [3] lies in the control word. If even one bit of the control word is changed, the whole process and the network structure

Manuscript received January 5, 2009.

<sup>1</sup> Assistant Professor (SG) in Electronics and Communication Engineering Department, Karunya University Coimbatore, India..

<sup>2</sup> Assistant Professor in Computer Science and Engineering Department, Government college of Technology, Coimbatore, India.

changes. The control word used in the paper has five '1's and three '0's. Hence at the output of the left and the right data path, there would be uneven number of relatively insignificant bits available. But in this paper, it is needed to have even number of locations free so that the video bits can be incorporated and reconfigurability [1][2] can be applied. Hence the control word is changed to have even number of one's and zero's and it in turn demands that the whole structure be changed. Hence a new structure is designed for the new control word.

According to the method, the data bits are masked with the control bits to obtain the left data path. To obtain the right data path, the data bits are masked with the inverted control bits. But since the control word has been changed, the same structure cannot be applied to the left and the right data paths. Hence a new structure is designed for each data path.

One modification has been made to design the structure. The second constraint in the 'barrier' is that if the bits in the control word are '0', then their corresponding swap locations be swapped. This is done to take care that the relatively insignificant bits that are sent to the right side(LSB side) are not brought back. This constraint has been removed in the enhancement with the argument that instead of blocking the relatively insignificant bits in the beginning itself, they can be pushed to the right in subsequent levels of the design.

Another modification that has been brought out is to introduce the 'barrier' signal in the first level itself. This would help so that the relatively significant bits are not moved.

The paper [3] expounds only the sorting technique at the transmitter side. It does not give any method to retrieve (decrypt) the data at the receiving end. New proposed structures have been designed and introduced so that the data can be retrieved back. Separate structures have been designed for decryption of the left and right data paths.

**IV. IMPLEMENTATION DETAILS OF LEFT DATA PATH AT THE TRANSMITTER END**

In the first level positions 1-5, 2-6, 3-7, 4-8 are checked. So  $A_5 A_4$  and  $A_1 A_0$  are interchanged since they have different control bits. A 'swap' signal is generated denoting the locations which have been swapped. In the second level, positions 3-6, 4-7, 5-8 are checked. Hence  $A_1 A_0 A_3$  and  $A_2 A_5 A_4$  are swapped. In the third level, positions 4-6, 5-7 are checked. Hence  $A_0 A_4$  and  $A_1 A_5$  are swapped. The structure for this implementation is shown in fig 2.

Data	$A_7$	$A_6$	$A_5$	$A_4$	$A_3$	$A_2$	$A_1$	$A_0$
Control	1	1	0	0	0	1	1	0
Barrier	1	1	0	0	0	0	0	0
Eq. Control	1	1	0	0	0	1	1	1

**LEVEL 1**

Data	$A_7$	$A_6$	$A_1$	$A_0$	$A_3$	$A_2$	$A_5$	$A_4$
Swap	0	0	1	1	0	0	1	1
Barrier	1	1	0	0	0	0	1	1

**LEVEL 2**

Data	$A_7$	$A_6$	$A_2$	$A_0$	$A_4$	$A_1$	$A_5$	$A_3$
Swap	0	0	1	0	1	1	0	1
Barrier	1	1	0	0	0	0	1	1

**LEVEL 3**

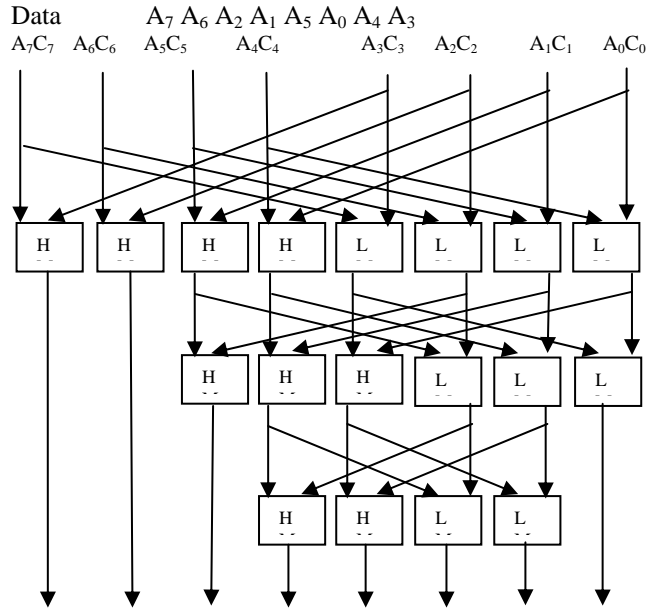


Fig. 2 Structure of MEMS of the left data path at the transmitter end

**V. IMPLEMENTATION DETAILS OF LEFT DATA PATH AT THE RECEIVER END**

In the first level positions 3-6, 4-7 are checked. So  $A_0 A_4$  and  $A_2 A_1$  are interchanged since they have different control bits. A 'swap' signal is generated denoting the locations which have been swapped. If the swap bits are different, the bits positions are relatively incorrect. In the second level, position 3-5 is checked. Hence  $A_0$  and  $A_5$  are swapped. In the third level, position 5-8 is checked. Hence  $A_0$  and  $A_3$  are swapped. The structure for this implementation is shown in fig 3.

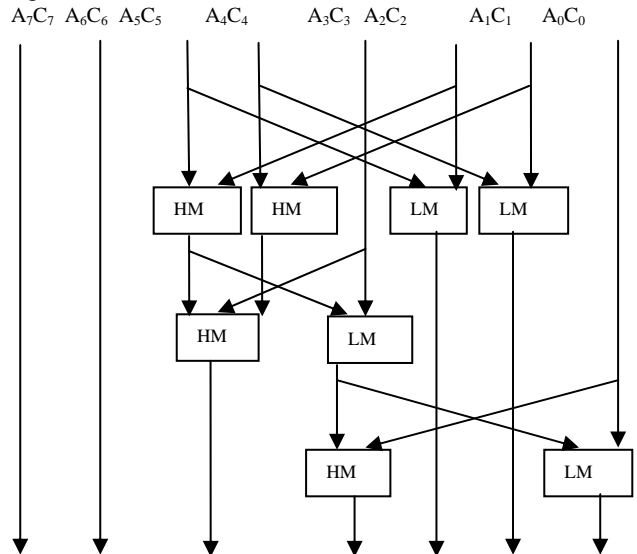


Fig. 3 Structure of MEMS of the left data path at the receiver end

Data	$A_7$	$A_6$	$A_2$	$A_1$	$A_5$	$A_0$	$A_4$	$A_3$
Barrier	1	1	0	0	0	0	1	1
Eq. Control	1	1	0	0	0	1	1	1

**LEVEL 1**

Data	$A_7$	$A_6$	$A_0$	$A_4$	$A_5$	$A_2$	$A_1$	$A_3$
------	-------	-------	-------	-------	-------	-------	-------	-------

Swap 0 0 1 1 0 1 1 0

**LEVEL 2**

Data A<sub>7</sub> A<sub>6</sub> A<sub>5</sub> A<sub>4</sub> A<sub>0</sub> A<sub>2</sub> A<sub>1</sub> A<sub>3</sub>  
Swap 0 0 1 0 1 0 0 0

**LEVEL 3**

Data A<sub>7</sub> A<sub>6</sub> A<sub>5</sub> A<sub>4</sub> A<sub>3</sub> A<sub>2</sub> A<sub>1</sub> A<sub>0</sub>

**VI. IMPLEMENTATION DETAILS OF RIGHT DATA PATH AT THE TRANSMITTER END**

In the first level positions 1-5, 2-6, 3-7, 4-8 are checked. So A<sub>7</sub> and A<sub>3</sub> are interchanged since they have different control bits. A 'swap' signal is generated denoting the locations which have been swapped. In the second level, position 2-5 is checked. Hence A<sub>6</sub> and A<sub>7</sub> are swapped. In the third level, positions 1-7, 2-8 are checked. Hence A<sub>0</sub> and A<sub>7</sub> are swapped. The structure for this implementation is shown in fig 4.

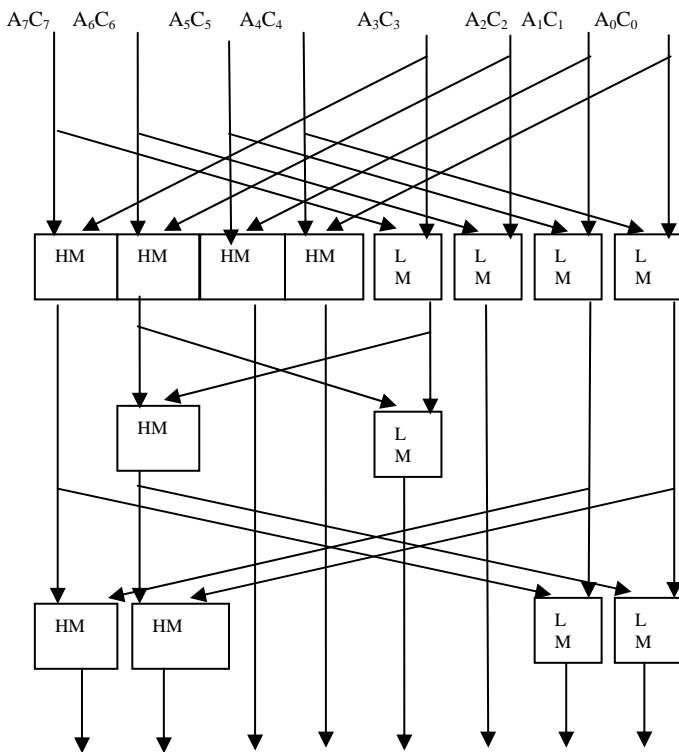


Fig. 4 Structure of MEMS of the right data path at the transmitter end

Data A<sub>7</sub> A<sub>6</sub> A<sub>2</sub> A<sub>1</sub> A<sub>5</sub> A<sub>0</sub> A<sub>4</sub> A<sub>3</sub>  
Inv. Control 0 0 1 1 1 0 0 1

**LEVEL 1**

Data A<sub>3</sub> A<sub>6</sub> A<sub>5</sub> A<sub>4</sub> A<sub>7</sub> A<sub>2</sub> A<sub>1</sub> A<sub>0</sub>  
Swap 1 0 0 0 1 0 0 0

**LEVEL 2**

Data A<sub>3</sub> A<sub>7</sub> A<sub>5</sub> A<sub>4</sub> A<sub>6</sub> A<sub>2</sub> A<sub>1</sub> A<sub>0</sub>  
Swap 0 1 0 0 1 0 0 0

**LEVEL 3**

Data A<sub>3</sub> A<sub>0</sub> A<sub>5</sub> A<sub>4</sub> A<sub>6</sub> A<sub>2</sub> A<sub>1</sub> A<sub>7</sub>  
Sent Data A<sub>5</sub> A<sub>4</sub> A<sub>3</sub> A<sub>0</sub> A<sub>6</sub> A<sub>2</sub> A<sub>1</sub> A<sub>7</sub>

**VII. IMPLEMENTATION DETAILS OF RIGHT DATA PATH AT THE RECEIVER END**

In the first level positions 1-3, 2-4 are checked. So A<sub>5</sub> A<sub>4</sub> and A<sub>3</sub> A<sub>0</sub> are interchanged since they have different control bits. In the second level, position 1-8 is checked. Hence A<sub>3</sub> and A<sub>7</sub> are swapped. In the third level, position 5-8 is checked. Hence A<sub>6</sub> and A<sub>3</sub> are swapped. In the fourth level, position 2-8 is checked. Hence A<sub>0</sub> and A<sub>6</sub> are swapped. The structure for this implementation is shown in fig 5.

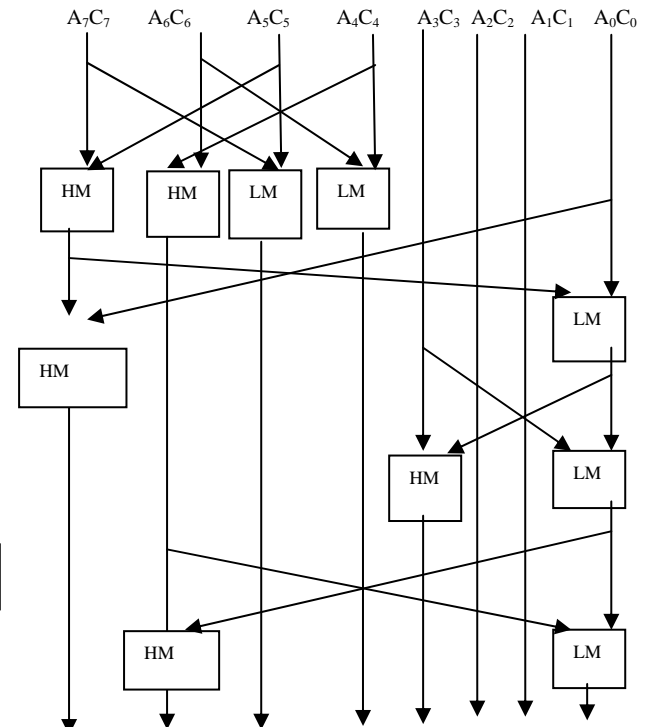


Fig. 5 Structure of MEMS of the right data path at the receiver end

Data A<sub>5</sub> A<sub>4</sub> A<sub>3</sub> A<sub>0</sub> A<sub>6</sub> A<sub>2</sub> A<sub>1</sub> A<sub>7</sub>  
Inv. Control 0 0 1 1 1 0 0 1

**LEVEL 1**

Data A<sub>3</sub> A<sub>0</sub> A<sub>5</sub> A<sub>4</sub> A<sub>6</sub> A<sub>2</sub> A<sub>1</sub> A<sub>7</sub>  
Swap 1 0 0 0 1 0 0 0

**LEVEL 2**

Data A<sub>7</sub> A<sub>0</sub> A<sub>5</sub> A<sub>4</sub> A<sub>6</sub> A<sub>2</sub> A<sub>1</sub> A<sub>3</sub>  
Swap 0 1 0 0 1 0 0 0

**LEVEL 3**

Data A<sub>7</sub> A<sub>0</sub> A<sub>5</sub> A<sub>4</sub> A<sub>3</sub> A<sub>2</sub> A<sub>1</sub> A<sub>6</sub>  
Swap 1 0 0 0 1 0 0 0

**LEVEL 4**

Data A<sub>7</sub> A<sub>6</sub> A<sub>5</sub> A<sub>4</sub> A<sub>3</sub> A<sub>2</sub> A<sub>1</sub> A<sub>0</sub>

**VIII. RESULT ANALYSIS AND DISCUSSIONS**

The following modifications are made in [3],  
1. The control word is of even number of 1's and 0's. Hence the video signals can be incorporated into the four data paths evenly along with audio signals.  
2. New structure has been designed for both the left and right data path separately. The designed structures are simpler and more efficient than the ones existing [3].

3. Control word is the basic element of security in this design. Changing even one bit of the control word would alter the process completely. Hence the intruder cannot get the data easily.

4. Decryption of the data paths at the receiver end has also been introduced. Separate structures have been designed for this. This has not been mentioned in [3].

Since no paper is appeared earlier related to FPGA implementation of audio sorter, we have written the code for [3] which deals only with encryption and its corresponding synthesis report converted into transistor level netlist and is implemented in Tanner EDA Tool and the power is calculated. Then our proposed algorithm which deals with both encryption and decryption code is taken and its corresponding synthesis report converted into transistor level netlist and is implemented in Tanner EDA Tool and the power is compared with power consumption of [3]. The result is shown in table.1 and power waveforms are shown in fig 6 and 7 respectively.

The command to find the power in Tanner Tool EDA is

```
.model pmos pmos
.model nmos nmos
.tran 4n 400n
.print p(v /node number/)
```

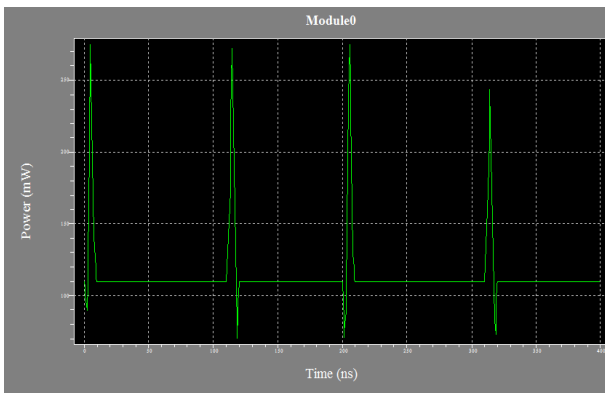


Fig.6 Power waveform for the Enhanced Merge Sorting Network from Tanner Tool

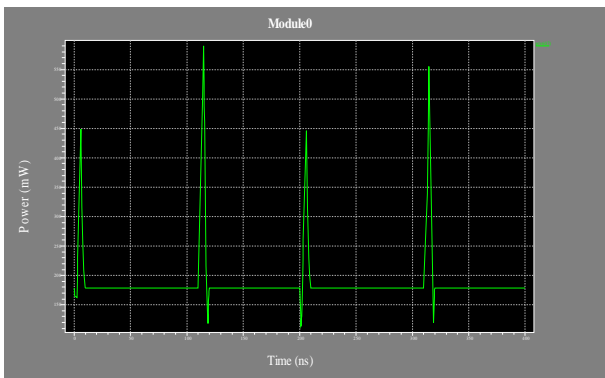


Fig.6 Power waveform for the proposed MEMS algorithm

TABLE 1 EXPERIMENTAL RESULTS FOR POWER CALCULATIONS

PAPER	POWER CONSUMPTION(mW)
[3] (Encryption)	270
Proposed algorithm	600

(Encryption & Decryption)

## IX. CONCLUSION

The paper propose a novel scheme of “Audio permutation Sorter Unit” for audio applications. It exploits the concept of reconfigurable computing which is a recent technique of VLSI system design. The audio sorting algorithms have been designed at the chip level by using hardware description language and thereby implement the design in a single chip. The low power techniques are introduced to optimize the design for utmost real time reliability. This paper is a real time application and can be incorporated in various fields like satellite data security, radar echo pulse data security, TV systems security and the like to provide image and audio security.

## ACKNOWLEDGMENT

The authors would like to thank the Karunya University management for providing all the tools and support to carry out the research successfully

## REFERENCES

- [1] Jer Min Jou, Yun Lung Lee, Chen Yen Lin and Chien Ming Sun, “A Novel Reconfigurable computation unit for DSP applications”, IEEE comp. society annual symp. on VLSI, ISVLSI’07, pp 439-444, 9-11 March 2007.
- [2] Navid Lashkarian, Ed Hemphi, Helen Tarn, Hemang Parekh and Chris Dick, “Reconfigurable Digital Front End Hardware for wire less base-station transmitters: Analysis, Design and FPGA implementation”, IEEE transactions on circuits and systems, vol 54, No. 8, pp 1666-1677, Aug 2007.
- [3] Giorgos Dimitrakopoulos, Christos Mavrokefalidis, Kostas Galanopoulos and Dimitris Niolos, “Sorter based permutation units for Media-Enhanced Processors” IEEE Transactions on VLSI systems, vol 15, no. 6, pp 711-715, June 2007.
- [4] Ismail Kadayif, Partho Nath, Mahmut Kandemir and Anand Sivasubramaniam, “Reducing data TLB power via compiler directed Address Generation”, IEEE transactions on Comp. Aided design of IC’s and systems, vol 26, no 2, pp 312-324, Feb 2007.
- [5] T. Conte etal., “Challenges to combining general purpose and multimedia processors” IEEE computer, Vol. 30, no 12, pp.33-37, Dec 1997.
- [6] Z.J.Shi, “Bit permutaion instructions: Architeture, Implementation and cryptographic properties”, Ph.D Dissertation, Electr. Engg. Dept. Princeton Univ., Princeton, NJ, 2004.
- [7] Z.J.Shi and R.B.Lee, “Implementation Complexity of bit permutation instructions”, in Proc.Asilomar Conf. Signals Stt. Comput, pp 879-886, 2003.
- [8] I. Kuroda and T. Nishitani, “ Multimedia processors”, Proc. IEEE, Vol.86, no 6, pp 1203-1221, Jun 1998.
- [9] R.B.Lee, Z.Shi, X.Yang, “Efficient permutation instructions for fast software cryptography” IEEE Micro, vol 21, no. 6, pp 56-69, Nov/Dec 2001.
- [10] X.Yang, and R.B.Lee, “Fast subword permutation units using omega and flip network stages”, in Proc. IEEE Int. Conf. Comput. Design, pp 15-22m, 2000
- [11] P Karthigaikumar, Dr. K. Basakaran and Praveen babu, “A Novel argument to use 8 bit processor for low power media application” International Conference on IMECS, Vol 1, pp 301-306, March 2008.



Mr. P. Karthigaikumar received his Bachelor of Engineering degree in Electrical and Electronics Engineering from the Bharathiar University, India in 1999 and his Master of Engineering degree with Distinction in Applied Electronics from Bharathiar University, India in 2003. He is pursuing Ph. D degree in Anna University-Coimbatore, India from 2007, focusing on Media Security processor. He is the member of International Association of Engineers (MIAENG) and member of International Association of Computer sciences

and Information Technology (MIACSIT). He joined Karunya University, Coimbatore, India in 2000. He is now Assistant Professor (SG) in Electronics and Communication Engineering.

Dr. K. Baskaran received his Bachelor of Engineering degree in Electrical and Electronics Engineering from the Annamalai University, India in 1989, Master of Engineering degree in Computer Science Engineering from Bharathiar University, India in 2002, and Ph. D degree from Anna University-Chennai, India in 2006. He is a member of IEEE and member of ISTE. He is now Assistant Professor in Computer Science and Engineering, Government college of Technology, Coimbatore, India