HARDWARE SOFTWARE CO-SIMULATION OF MOTION ESTIMATION IN H.264 ENCODER

S. Valarmathi[1], R. Vani[2] and Dr. M. Sangeetha[1]

[1]Dept. of ECE, Karpaga Vinayaga College of Engg. and Technology,
Kanchipuram.
valars88@gmail.com, sang_gok@yahoo.com,
[2]Anna University of Technology, Chennai.
vani_gowtham@yahoo.com

ABSTRACT

This paper proposes about motion estimation in H.264/AVC encoder. Compared with standards such as MPEG-2 and MPEG-4 Visual, H.264 can deliver better image quality at the same compressed bit rate or at a lower bit rate. The increase in compression efficiency comes at the expense of increase in complexity, which is a fact that must be overcome. An efficient Co-design methodology is required, where the encoder software application is highly optimized and structured in a very modular and efficient manner, so as to allow its most complex and time consuming operations to be offloaded to dedicated hardware accelerators. The Motion Estimation algorithm is the most computationally intensive part of the encoder which is simulated using MATLAB. The hardware/software co-simulation is done using system generator tool and implemented using Xilinx FPGA Spartan 3E for different scanning methods.

KEYWORDS

H.264/AVC, Hardware accelerators, Motion Estimation, Scanning methods.

1. INTRODUCTION

H.264/AVC is an industry standard for compressing video. The H.264/AVC offers better compression efficiency and greater flexibility in compressing, transmitting and storing video. Compared with standards such as MPEG-2 and MPEG-4 visual, H.264/AVC can deliver better image quality at the same compressed bit rate (or) deliver the same image quality at a lower bit rate. H.264 is widely used for video applications, for example in HDTV, but also in mobile video and internet application. It is a standard developed by the ITU-T video coding Experts Group (VCEG) together with ISO. Hardware H.264 encoder can be an ASIC or FPGA which is a general programmable chip [1, 2, and 7].

An encoder converts video into a compressed format and a decoder converts compressed video back into an uncompressed format. H.264 video encoder carries out prediction, transform and encoding processes to produce a compressed H.264 bit stream. An H.264 Video Decoder carries out the complementary process of decoding, inverse transform and reconstruction to produce a video sequence. H.264/AVC standard allows for seven modes with variable block sizes.16x16 pixel macroblock can be divided in smaller blocks to yield better compression efficiency. The

Natarajan Meghanathan, et al. (Eds): ITCS, SIP, JSE-2012, CS & IT 04, pp. 263–271, 2012.

DOI : 10.5121/csit.2012.2124

increasing video resolutions and the increasing demand for real-time encoding require the use of faster processors. However, power consumption should be kept to a minimum.

1.1. Motion Estimation

Motion Estimation is a part of ‘inter coding’ technique. Inter coding refers to a mechanism of finding ‘co-relation’ between two frames (still images), which are not far away from each other as far as the order of occurrence is concerned, one called the reference frame and the other called current frame, and then encoding the information which is a function of this ‘co-relation’ instead of the frame itself [4] . Motion Estimation is the basis of inter coding, which exploits temporal redundancy between the video frames, to scope massive visual information compression as shown in Fig 1. Motion estimation is part of prediction step. The vector between the position of the macroblock in the current frame and position of best matching block in the previous frame is called Motion Vector.

Motion estimation is the processes of determining motion vectors that describe the transformation from one 2D image to another usually from adjacent frames in a video sequence. The Motion vectors may be represented by a translational model or many other models that can approximate the motion of a real video camera such as rotation and translation in all three dimensions and zoom.

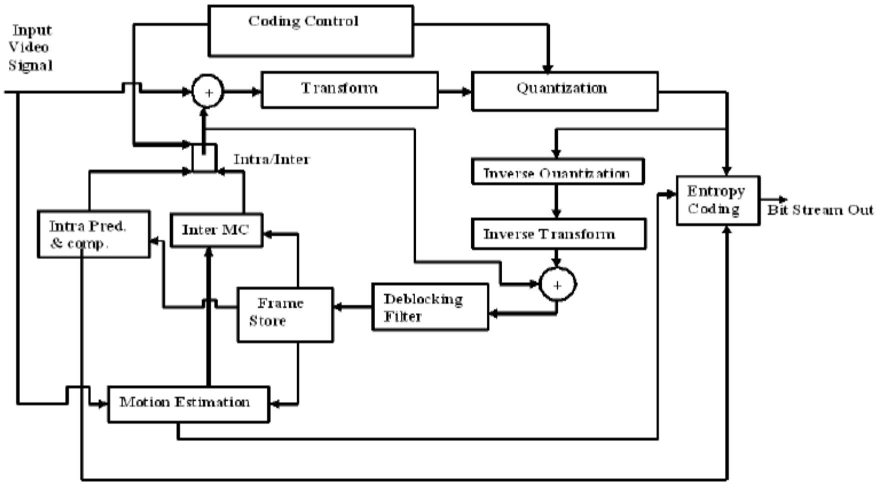


Figure 1. Block diagram of H.264 encoder

1.2. Target platforms

The software required is MATLAB 7.0 Version. MATLAB is a high level technical computing language and interactive environment for algorithm development, data visualization, data analysis and numeric computation. Using the MATLAB product, you can solve technical computing problems faster than with traditional programming languages such as C, C++ and FORTRAN. One special characteristics of MATLAB 7.0 is that, it has direct link to the FPGA through Xilinx blockset in Xilinx 10.1, through system generator by creating JTAG.

The hardware required is FPGA target platform SPARTAN 3e with Xilinx 10.1. The Spartan-3E family of Field-Programmable Gate Arrays (FPGAs) is specifically designed to meet the needs of high volume, cost-sensitive consume electronic applications. Because of their exceptionally low cost, Spartan-3E FPGAs are ideally suited to a wide range of consumer electronics applications, including broadband access, home networking, display/projection, and digital television equipment. The Spartan-3E family is a superior alternative to mask programmed ASICs. Xilinx System Generator is a MATLAB Simulink blockset that facilitates the design and targeting of Xilinx FPGAs. Within MATLAB, designers can both target a Xilinx FPGA hardware platform and verify the hardware output, making it easier for an algorithm developer to make the leap into hardware and a firmware developer to better grasp the algorithm.

The rest of this paper is organized as follows: Section 2 gives a brief overview of the related work. The section 3 gives the Motion Estimation algorithm. Scanning methods are discussed in section 4. Co-design and Co-Simulation approaches are focussed on section 5. Implemented Results are discussed in Section 6. Section 7 gives the Conclusion and final Suggestions for future work.

2. RELATED WORK

Complex data flow control are major challenges in high definition integer motion estimation hardware implementation [6]. The proposed multi-resolution motion estimation algorithm reached a good balance between complexity and performance with rate distortion optimized variable block size motion estimation support [20, 22]. Performance and complexity jointly optimized integer motion estimation (IME) engine which is the biggest challenge in HD video encoder architecture. System throughput burden, memory bandwidth, and hardware cost are huge challenges in HD FSBM based ME architecture due to the large search window size requirement [15]. In modern video coding standards, for example H.264, fractional-pixel motion estimation (ME) is implemented. Many fast integer-pixel ME algorithms have been developed to reduce the computational complexity of integer-pixel ME [16]. A parallel motion estimation algorithm based on stream processing is discussed in [19]. Many approaches are explored to enable high data reuse efficiency and computation parallelism for GPUs or other programmable processors. In [5], a novel prediction method is introduced which is based on multiple search centers, which can efficiently improve the prediction accuracy. In a typical video coding system, the computational complexity of ME is approximately 50% to 90% of the total. Fast algorithms based on statistical learning to reduce the computational cost involved in three main components in H.264 encoder, i.e., intermode decision, multi-reference motion estimation (ME), and intra-mode prediction [7]. Since the execution speed is a major challenge for real time applications, several methods have been proposed for fast intermode decision in H.264 recently [9]. In contrast to intra prediction, the partition of a MB into several smaller-size blocks for better ME is called intermode decision. In [1, 8], not only the rate distortion performance is considered but also its implementation complexity using an efficient hardware/software Co-design.

Mode decision is a process such that for each block-size, bit-rate and distortion are calculated by actually encoding and decoding the video. Therefore, the encoder can achieve the best Rate Distortion (RD) performance, at the expense of calculation complexity. Intraprediction use previously decoded, spatially-local macroblocks to predict the next macroblock and it works well for low-detail images [1, 22]. The combination of using only one reference frame, 16×16 block size, the de-blocking filter, and turning off the RDO option, achieves the best balance between computation complexity and picture quality [22]. The tests mainly aimed at finding how rate-distortion optimization and different block size partitions affect the compression performance of the transcoded videos. H.264/AVC employs rate distortion optimization (RDO) technique to determine the optimal mode from a set of coding modes with different block sizes. On the contrary, selecting a smaller block size usually results in a smaller residual [14]. If RDO calculation is performed on each possible coding mode, its extremely high computational complexity makes it a bottleneck for a real-time implementation of H.264/AVC encoder

3. MOTION ESTIMATION ALGORITHMS

The method for finding motion vectors can be categorized into pixel based methods (DIRECT) and feature based method (INDIRECT). Evaluation metrics for direct methods are Sum of Absolute Differences (SAD), sum of squared difference (SSD) and sum of absolute transformed difference (SATD) [1]. Variable Block Size (VBS) ME allows different MVs for different sub-blocks and can achieve better matching for all sub-blocks and higher coding efficiency than Fixed Block Size ME (FBSME). VBSME has good RD performance compared with FBSME, but it has huge computational requirement and irregular memory access making it hard for efficient hardware implementation. In [22], H.264 allows a 16 × 16 MB to be partitioned into seven kinds of sub-blocks as shown in Fig 2.

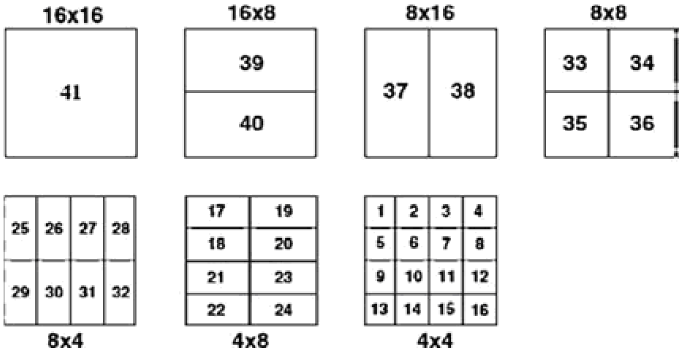


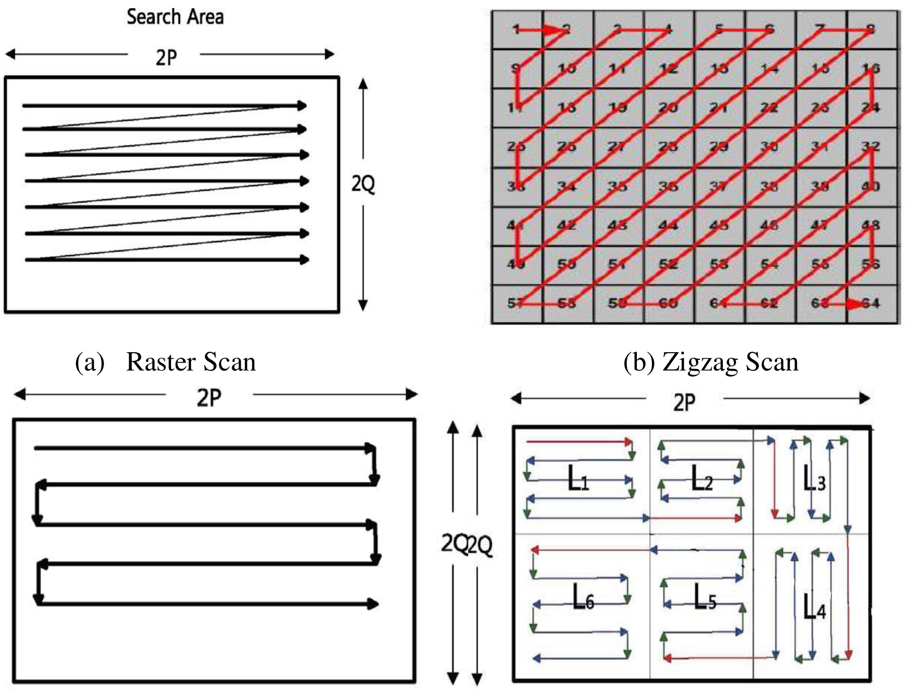
Figure 2. Variable block size of H.264 encoder

4. SCANNING METHODS

Different scanning methods and search patterns are discussed in this section. 3SS yields better speedup when compared to FS, DS ME algorithms, by taking a Leon3 uniprocessor video encoding system as the reference platform. The quality of fast ME algorithms have the following relations: DS~CDS~4SS~HEXBS.

In Raster Scan, the search locations in the first row are scanned from left to right, followed by the second row from left to right, and so on. Raster Scan as shown in fig 3(a) is effective in reusing data horizontally with relatively high data re-use ratio but with redundant loading. The encoding consists of two phases: Zigzag scanning is shown in fig 3(b), to convert the 2D image matrix into a 1–D vector and share generation. The decoding also consists of two phases: secret image recovery and Inverse zigzag scanning to convert 1–D vector to 2D image matrix.

The data re-usability is improved slightly in some architecture by another scanning order called Snake Scan as shown in Fig. 3(c). Snake Scan processes the first row from left to right, then the second row from right to left, and then the third row from left to right, and so on. A novel scanning order called Smart Snake (SS) is proposed in [5], which can achieve variable data re-use ratios and minimum redundant data loading. Search window is divided into an array of non-overlapping rectangular sub-regions that span the search window as shown in Fig. 3(d). In each rectangular sub-region, Snake Scan is performed to achieve significantly higher data re-use. After one sub-region is searched, it will move into an adjacent region and Snake Scan will be applied again. In different sub-regions, Snake Scan may be performed from top to bottom (L1), or from bottom to top (L2). It may start from left and end at right (L1, L2, L3), or start from right and end at left (L4, L5, L6). It may be horizontal (L1, L2) or vertical (L3, L4).



(a) Raster Scan (b) Zigzag Scan

(c) Snake Scan (d) Smart Snake Scan

Figure 3. Scanning methods

5. Co-Design and Co-Simulation Approaches

The emphasizes of Co-Design is on the area of system specification, hardware software partitioning, architectural design, and the iteration in between the software and hardware as the design proceeds to next stage. The hardware and software co design makes it possible. A Multi Core H.264 video encoder is proposed in [13, 19], by applying a novel hardware software co-design methodology which is suitable for implementing complexity video coding embedded systems. The hardware and the software components of the system are designed together to obtain the intended Performance levels. At the hardware level, the designer must select the system CPU, hardware accelerators, peripheral devices, memory and the corresponding interconnection structure. One method is to represent all systems in HDL. The second method is to use a simulator that supports all different HDLs used. The third method is to use different simulators for each system and verify the integrated system using co-simulation. Co-simulation is useful in HW/SW co-design. Co-simulation can be done by either connecting two simulators known as direct coupling or by the use of a co-simulation backplane. The co-simulation allows for designing in much short iteration while verifying functional behaviour.

6. Implemented Results

An important coding tool of H.264 is the variable block size matching algorithm for the ME (Motion Estimation).The motion estimator has two inputs: a macroblock from the current frame and a search area from the previous frame. The residual of current and reference frame is simulated using Xilinx blockset. The hardware/software co-simulation is implemented using Xilinx Spartan 3E FPGA by creating JTAG. The result is shown in fig 4.

Figure 4. Residual of Current & Reference frame with HW/SW Co-Simulation

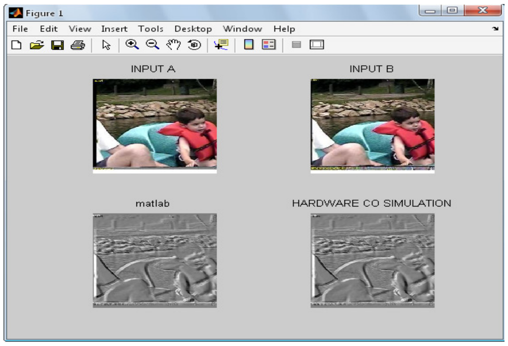
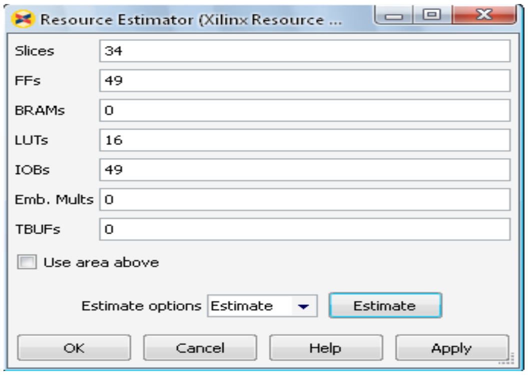


Figure 5. Xilinx resource estimator for HW/SW Co-Simulation



Slices	34
FFs	49
BRAMs	0
LUTs	16
IOBs	49
Emb. Mults	0
TBUFs	0

□ Use area above

Estimate options: Estimate

The interface between hardware and software environment is done using System Generator. Hardware utilization of motion estimation is shown in Fig 5. The current and reference frame are estimated by evaluation metrics of direct method Motion Estimation Algorithm using Sum of Absolute Differences. The Simulated result and the implemented hardware/software co-simulation are shown in fig 6. The peak signal-to-noise ratio (PSNR) between two images is 45.57dB which is used as a quality measurement.

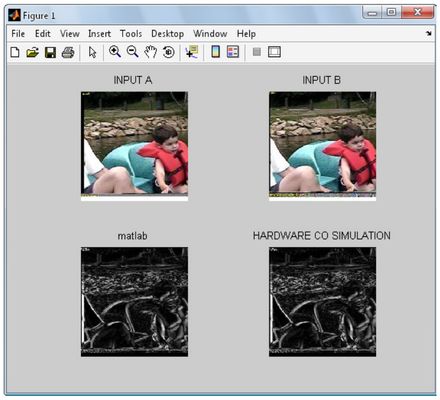


Figure 6. Residual of Current & reference frame using SAD with HW/SW Co-Simulation

7. CONCLUSION AND FUTURE WORK

Motion estimation in H.264 encoder is simulated using MATLAB with Zigzag Scanning method. The ideal hardware / software tool produces automatically a set of high quality partitions in short, predictable computation time allowing the designer to interact. Hardware software co-simulation is implemented in Xilinx Spartan 3E kit using system generator as interface.

Development of encoder is still a challenging issue, particularly for real time applications. For improving speed up factors, power consumption and rate distortion of motion estimation algorithm, HW/SW partitioning techniques can be done. The software part of H.264 encoder can be compiled by LABVIEW and executed in LABVIEW embedded module of ARM processor to increase the speed of the execution time. The most intensive part of H.264 encoder is considered as hardware, for which we need to improve the performance and reduce the power consumption. At the same time, the design can be extended to produce the better compressed image quality at lower bit rate.

REFERENCES

[1] Zhuo Zhao and Ping Liang, “A statistical analysis of H.264/AVC FME mode reduction”, IEEE transactions on circuits and systems for video technology, (2011)

[2] Huitao Gu, Shuming Chen, Shuwei Sun, Shenggang Chen, Xiaowen Chen,The 11th International Conference on Parallel and Distributed Computing, Applications and Technologies, “Multiple Search Centers Based Fast Motion Estimation Algorithm for H.264/AVC”, (2011).

[3] Zhenyu Liu, Member, IEEE, Junwei Zhou, Dongsheng Wang, Member, IEEE, and Takeshi Ikenaga, Member, “Register Length Analysis and VLSI Optimization of VBS Hadamard Transform in H.264/AVC”, (2011)

[4] Chen-Kuo Chiang, Student Member, IEEE, Wei-Hau Pan, Chiuan Hwang, Shin-Shan Zhuang,and Shang-Hong Lai, Member, “Fast H.264 Encoding Based on Statistical Learning”, (2011)

[5] Xing Wen, Lu Fang and Jiali Li: “Novel RD-Optimized VBSME with Matching Highly Data Re Usable Hardware Architecture”. In: IEEE Transactions on Circuits and Systems for Video Technology, Vol. 21, No. 2, pp. 206–219 (2011).

[6] Meihua GU, Ningmei YU, Lei ZHU, Wenhua: “High Throughput and Cost Efficient VLSI Architecture of Integer Motion Estimation for H.264/AVC”. In: Journal of Computational Information Systems, pp 1310–1318 (2011).

[7] Xiaomin Wu, Weizhang Xu, Nanhao Zhu, Zhanxin Yang , China, “A Fast Motion Estimation Algorithm for H.264”, International Conference on Signal Acquisition and Processing (2010).

[8] Haithem Chaouch, Salah DHAHRI, Abdelkrim ZITOUNI and Rached TOURKI Laboratory of Electronics and Micro-Electronics (LAB-IT06) Faculty of Sciences of Monastir Monastir, Tunisia,International Conference on Design & Technology of Integrated Systems in Nanoscale Era, “Low Power Architecture of Motion Estimation and Efficient Intra Prediction Based on HardwareDesign for H.264”, (2010).

[9] Zhi Liu, Member, IEEE, Liquan Shen, and Zhaoyang Zhang, “An Efficient Intermode Decision Algorithm Based on Motion Homogeneity for H.264/AVC” (2010).

[10] Haibing Yin, Huizhu Jia, Honggang Qi, Xianghu Ji, Xiaodong Xie, and Wen Gao, “A Hardware-Efficient Multi-Resolution Block Matching Algorithm and Its VLSI Architecture for High Definition MPEG-Like Video Encoders”, (2010).

[11] Chih-Hung Kuo, Member, IEEE, Li-Chuan Chang, Student Member, IEEE, Kuan-Wei Fan,and Bin- Da Liu, Fellow, "Hardware/Software Codesign of a Low-Cost Rate Control Scheme for H.264/AVC", (2010).

[12] Heng-Yao Lin, Kuan-Hsien Wu, Bin-Da Liu, Fellow, IEEE, and Jar-Ferr Yang, Fellow, "An Efficient VLSI Architecture for Transform-Based Intra Prediction in H.264/AVC", (2010).

[13] Tiago Dias, Nuno Roma, Leonel Sousa: "Hardware/Software Co-Design Of H.264/AVC Encoders For Multi-Core Embedded Systems" (2010).

[14] Qiang Tang, Student Member, IEEE, and Panos Nasiopoulos, Member, "Efficient Motion Re-Estimation with Rate-Distortion Optimization for MPEG-2 to H.264/AVC Transcoding", (2010).

[15] G. A. Ruiz and J. A. Michell: "An Efficient VLSI Architecture of Fractional Motion Estimation in H.264 for HDTV." In: J Sign Process Syst, Springer (2010).

[16] Ka-Ho Ng, Lai-Man Po, Shen-Yuan Li, Ka-Man Wong and Li-Ping Wang Department of Electronic Engineering, City University of Hong Kong, Hong Kong SAR, China, "A Linear Prediction Based Fractional-Pixel Motion Estimation Algorithm" (2009).

[17] Camille Mazataud, Benny Bing: "A Practical Survey of H.264 Capabilities". In: Seventh Annual Communication Networks and Services Research Conference, pp 25–32 (2009).

[18] L. Smit, G. Rauwerda, A. Molderink, P. Wolkotte, and G. Smit: "Implementation of a 2-D 8×8 IDCT on the reconfigurable Montium core". In: International Conference on Field Programmable Logic and Applications, pp 562–566 (2007).

[19] C.-Y. Lee, Y.-C. Lin, C.-L. Wu, C.-H. Chang, Y.-M. Tsao, and S.-Y. Chien: "Multi Pass and Frame Parallel algorithms of Motion Estimation in H.264/AVC for generic GPU". In: International Conference on Multimedia and Expo, pp 1603–1606 (2007).

[20] Y. Song, Z. Liu, T. Ikenaga, and S. Goto: "Ultra low complexity fast Variable Block Size Motion Estimation algorithm in H.264/AVC". In: IEEE International Conference on Multimedia and Expo, pp. 376–379 (2007).

[21] Nukhet Ozbek, Turhan Tunali: "A Survey on the H.264/AVC Standard". In: Turk JElec Engin, VOL.13, NO.3 (2005).

[22] Li Zhang, Wen Gao: "Improved FFSBM Algorithm and its VLSI Architecture For Variable Block Size Motion Estimation Of H.264". In: Proceedings of 2005 International Symposium on Intelligent Signal Processing and Communication Systems, (2005).

Authors

Valarmathi received B.E. degree in Electronics and Communication Engineering from Karpaga Vinayaga College of Engineering & Technology, Kanchipuram. She is currently doing Master degree with specialization of Embedded Systems in the same institution affiliated to Anna University of Technology, Chennai. Also currently working on the project based on Motion Estimation Algorithm in H.264/AVC encoder.

Vani received B.E. degree in Electronics and Communication Engineering from Bharathiar University, Coimbatore and M.E. degree in Communication Systems from Anna University, Chennai. She is currently pursuing Ph.D. in Anna University of Technology, Chennai and also working as Assistant Professor in Department of Electronics and Communication Engineering, Meenakshi College of Engineering, Chennai.

Sangeetha Marikkannan born in 1975 in Tamilnadu, India, received her B.E. degree from the Bharathidasan University, Trichy, in 1996 and M.E. degree from the University of Madras in 1999. She is currently with Karpaga Vinayaga College of Engineering and Technology in the Department of Electronics and Communication Engineering and Ph.D. degree with the Anna University, Chennai, India. She is a member of the IEEE and ISTE. Her research interests are Hardware/Software Co-design and High Level Synthesis.