FPGA enhanced implementation of ECG QRS complex detection algorithm

Sahel Alouneh, Mohammad Al-Shayeji, Sa’ed Abed, M.D. Samrajesh and Sari Sultan

***Abstract*—** Heart disease is one of the leaders in fatal diseases. Detecting disease in rudimentary stages leads to auspicious recovery rates. Analyzing the heart rate is a vital tool to detect abnormalities and malfunctions of the heart. Heart rate analyzers are used to identify the QRS complex of the Electrocardiograph ECG. Portable battery-operated ECG hardware devices has been developed in order to increase early detection— user can get real-time results without visiting the hospital, and these results can be sent remotely to the physician. These devices face many challenges such as delay, power, area, and cost in order to be used in daily life. In this paper, QRS complex design targeting FPGA has been developed, enhanced and optimized in terms of delay, power, and area, leveraging leading hardware QRS complex detection algorithm using Synopsis tool. The algorithm is downloaded on Altera’s FPGA board. The results show that our enhancements speed up the system, which is vital for real-time application. Moreover, the enhancement has succeeded in minimizing the power and the area. The design achieves the best average of 24% power reduction, 8.9% latency reduction and 10.5% area reduction compared to the original QRS algorithm. These optimizations should enhance and facilitate QRS complex on chip designs.

***Keywords*—** FPGA, Optimization Techniques, Complex Detection, QRS Chip.

I. Introduction

Cardiovascular diseases are considered a leading causes of death [1]. Early detection of cardiovascular diseases play a vital role in increasing survival rates since treatment in rudimentary stages is auspicious. For the last two decades, biomedical equipment’s for heart rate monitoring have leveraged microcomputer-based systems. Portable devices are increasingly used for analysis with monitors in the Intensive Care Units (ICU) to deliver continuous measurements of the heart rate for the medical stuff. Battery-operated devices such as wireless ECG sensors are used in order to assess the heart function without the need to visit the clinic, hence, promoting the chances of early cardiovascular disease detection.

Primarily, ECG main task is robust detection for QRS complex [2]-[4]. Performing ECG using hardware based devices has been addressed by many researchers who pointed that such devices are capable of performing medical services efficiently [5]-[7]. Different systems have been employed to execute signal processing functions such as 12-lead off-line ECG analysis, Holter tape analysis, and real-time patient monitoring. All these application require robust detection of the QRS complex. QRS detection is a computationally intensive operation; many algorithms implemented in software and could not operate in real-time. On the other hand, real-time hardware based solutions have been developed using Application Specific Integrated Circuit (ASIC), and Field Programmable Gate array (FPGA). Hardware designs achieved better results over software designs [8]-[10]. In addition, power optimization for QRS designs has been implemented on FPGA in order to minimize power consumption, which is an advantage for battery-operated devices [1], [5]. Moreover mixed hardware/software designs proposed and achieved 17 times faster speed than software designs [9].

Synopsys's Design Compiler is a well-known synthesis tools used in the ASIC industry. The two main parts of the tool are the Design Analyzer (DA), which implements a Graphical User Interface (GUI) and the Design Compiler (DC), which implements a Command Line Interface (CLI). The FPGA devices are specified using a Hardware Description Language (HDL). Due to their speed and circuit density, FPGAs are essential in the ASIC areas. The goal of this work is to implement real-time QRS detection algorithms, targeting ASIC and FPGA devices, leveraging the leading HDL algorithms while optimizing the design for three major parameters: delay, power, and area.

A designer implementing a circuit on an FPGA must have access to CAD tools for that type of FPGA. The design process, as depicted in Fig. 1, starts by defining the goals of design and selecting the suitable compilation strategy. Consequently, enhancement techniques are applied on the design and results analysis. Deploying the design to FPGA logic blocks or a circuit will then take place using technology mapping program: to minimize number of blocks and suitable placement strategy for blocks in FPGA array in order to minimize interconnects length. After wiring and selecting programmable switches, the programming unit to configure the final FPGA board will invoke the output of the CAD.

Sahel Alouneh is with German Jordanian University, Amman, Jordan ((corresponding author e-mail: sahel.alouneh@gju.edu.jo).

Mohammad Al-Shayeji is with Kuwait University, Kuwait. (e-mail: alshayej@eng.kuniv.edu.kw).

Sa’ed Abed is with Kuwait University, Kuwait (e-mail: s.abed@ku.edu.kw).

M.D. Samrajesh is with Different Media, Kuwait (e-mail: sam@differentmedia-kw.com).

Sari Sultan is with CARDO Systems, Amman, Jordan (e-mail: engSariSultan@gmail.com).

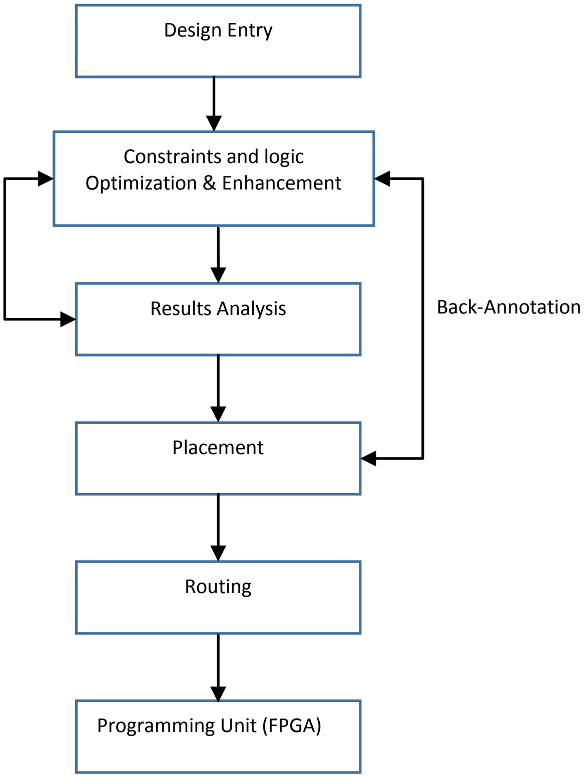


Fig.1 Design Flow Process Diagram

In context, the contributions of this paper are as follow:

- Leveraging leading QRS algorithm in the field efficiently.
- Enhance the algorithms in terms of delay, power, and area.
- Increase battery-operated QRS complex devices feasibility, including, minimizing power, and area.

In this paper, a setup file is used to ascertain the target library (xdc_virtex2-5.db) and other variables essential for the synthesis. Design Compiler (DC) has numerous options, which allow the designer to take advantage of using them: making clocks, adding constraints to the design, analyzing a design and others. We focus on the delay, power, and area enhancements and optimizations. During this stage, the DC generates a netlist file based on the target technology library that accomplishes the timing and the area goals.

Moreover, we present VHDL code for QRS integrator that consists of an algorithm that describes its functionality. The code is transformed from behavioral model to synthesizable RTL. Next, dynamic verification is done to validate the synthesizable RTL against the behavioral model.

To validate the verification procedure, a test-bench environment is implemented. Next, the codes are synthesized for minimum latency, power, and area. The two codes were downloaded to FPGA board. Finally, a comparison with the original code and the improved code is performed and the results showed significant enhancement in applying our proposed enhancement techniques. Simulations on Altera's Quartus-II FPGA simulator showed that our proposed design achieves an average of 24% power reduction, 8.9% latency reduction and 10.5% area reduction compared to the original QRS code.

The rest of this paper is organized as follows; Section 2 presents the closest related work to ours. Section 3 presents the proposed enhancement technique. It describes the synthesis results and analysis of the original and improved code for minimum latency, area and power for different cycle times. Section 4 focuses on the comparisons and comments on the obtained results. Finally, Section 5 concludes the paper and presents some suggested future works.

II. Related Work

Several QRS detection algorithms have been proposed and implemented [8]-[10]. However, little has been done to optimize the various algorithms.

Optimal wavelet based efficient QRS complex detection algorithm is presented in [11]. The QRS complex localization uses maximum detection and peak classification algorithm. The effects of various wavelet functions such as time, frequency localization and linearity of QRS detection are examined. The algorithm applies ECG registrations from the MIT-BIH database. The evaluation results showed that the wavelets enhance the average detection ratio.

An estimation of QRS power spectra for design of a QRS filter is presented in [12]. The paper presented spectral analysis of ECG waveforms. Their analysis showed that the QRS complex could be isolated from other interfering signals. Signal-to-noise is used as band-pass filter to maximize the ratio. The coherence function is computed from the signal-to-noise ratio. The authors observed that the band-pass filter with a frequency of 17 Hz and a Q of 5 produces the best signal-to-noise ratio.

A wavelet transformation based QRS design using FPGA is presented in [13]. The proposed design applies the parallelism of FPGA thereby offers better throughput, with minimized sample rate as several hundred samples per second. However, the detection accuracy is lower when compared with other algorithms.

A Very-Large-Scale Integration (VLSI) morphology based QRS complex detector is presented in [14]. The baseline wandering and background noise are removed by a mathematical morphological method. A low-pass filter is used to enhance the QRS complex and enhance the signal-to-noise ratio. Evaluation is done using the standard MIT-BIH Arrhythmia database and wearable workout ECG data. Results showed that the detector offered a higher detection rate.

An optimization of ECG classification using feature selection is presented in [15]. To reduce computational resources Sequential Forward Floating Search (SFFS) algorithm is applied with an updated criterion function index that uses linear discriminants. Moreover, the subset returned is further evaluated using a Multilayer Perceptron (MLP) to assess the robustness of the model. Their performance analysis showed that the proposed method reduces classification complexity.

A QRS complex detection algorithm for wearable and wireless electrocardiogram recording device is presented in [16]. The proposed algorithm can automatically shift between single-channel and multi-channel investigation mode. The algorithm was evaluated using MI T-BIH Arrhythmia Database (MITDB). The overall detection performance is increased by the inclusion of the addition channel on the MITDB. Thus, the method is considered better than a single-channel approach using an arbitrary channel.

A QRS complex detection using integer wavelet transformation is presented in [17]. The system includes many modules that are incorporated in a single FPGA chip. The single chip can be embedded in portable medical instruments or wearable health care devices. Software simulation and the verification are performed on the proposed method. The results show that the proposed method has an accuracy of about 95%, offers noise immunity and low cost.

A peak detection system for heart rate and heart rate variability calculation using FPGA is presented in [18]. The paper claims that the design of the system is both simple and economical, as it requires low cost hardware for implementation. The data used is from the standard Atrial Fibrillation (AF) termination challenge database. The implementation of QRS detection algorithm integrates a slope vector waveform analysis. This offers a quick and accurate search of the location of the peak of the R wave. In addition, the paper claims that the computations are substantially less when compared with methods such as a wavelet based analysis.

An FPGA based implementation of a QRS detector is presented in [19]. The decimation filters and the QRS detection algorithm were modeled and simulated before being tested on hardware. The algorithms were then converted to VHDL to target FPGA. The paper provides new FPGA implementations of QRS detection systems suitable for ASIC mapping. Moreover, the paper investigates the trade-offs in energy consumption and the intricacy of high resolution encoding versus filter elimination necessities.

A hardware based implementation Of QRS complex detection using FPGA is presented in [20]. The paper applies mathematical morphological to extract information with low signal-to-noise ratio by removal of baseline wandering and background noise from original ECG image. Leveraging multi-pixel modulus accumulation as low-pass filter to determine the maximum value of QRS. The paper concludes that the proposed method detected the QRS of ECG, even under noise, baseline drift and large P/T waves.

A QRS complex detection algorithm using FPGA implementation for fetal and maternal heart rates is presented in [21]. The procedure is built on cross correlation, adaptive thresholding and statistical characteristics in the time domain. The evaluation results showed that the model could extract both maternal and fetal heart rates employing a single-lead configuration.

A validation of low-power and wearable ECG patch is presented in [22]. The paper measured and compared the functionality of hardware performance of a proprietary developed ECG patch with a medical gold standard 24-h Holter device. They used standardized AF algorithm on the observations and the results showed a good correlation between both devices. The paper concluded that the new ECG patch has the same performance as a medical gold standard Holter. Moreover, improvements in electronics contraction and patch optimization can further improve stability on the chest wall and physical activities.

An optimization of QRS chip using Synopsys tool is presented in [23], [24] to minimize latency and area. Minor modifications were done to save two clock cycles and few units.

In this paper, we focus on enhancing the QRS chip to minimize latency and area as a continuation to the work in [23], [24] by applying more enhancements and optimizations. Moreover, as more devices are currently handheld and wearable thereby operated using battery, energy conservation is becoming an important challenge. Our proposed enhancement method is analyzed to measure the energy consumption in the device by downloading the design into FPGA board.Final Stage

III. Proposed Enhancement Technique

The proposed enhancement technique consists of various phases as shown in Fig. 2. The QRS detection algorithm is primarily implemented in VHDL to define the operational functionality of the heart rate analyzer chip by detecting the characteristic of Q-R-S points of the ECG data stream; Altera’s Quartus II is used so that the system can be synthesized into an FPGA device with real timing, area and power values. The VHDL code can be downloaded into an FPGA for physical realization or as the netlist file in a design flow to be developed into a custom ASIC device.

The original VHDL code is then improved in the next phase. Changes are made in the original code for synthesizing using the Synopsys tool. The impact in terms of timing and functionality are observed. The next step is the testing phase using a test bench to verify its behavior functionality. The synthesis phase for better timing, area and power is applied. Finally, the improved code performance in terms of functionality, latency, power consumption and area represented as total logic elements is determined, analyzed and compared.

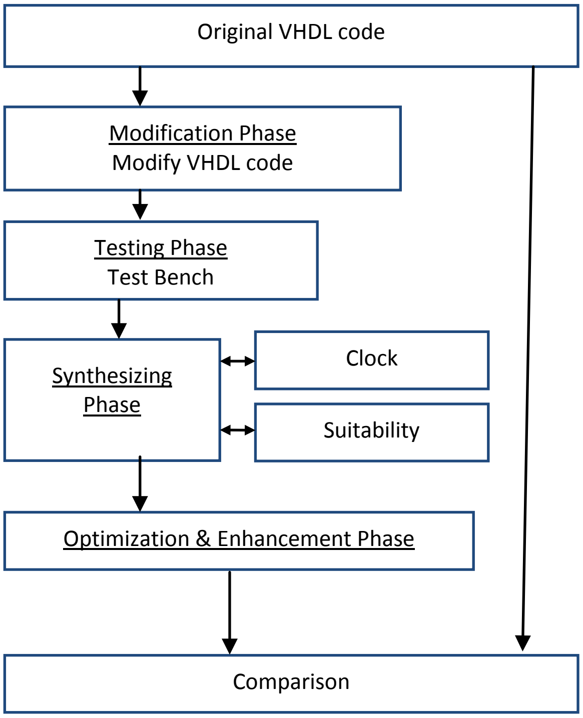


Fig .2: Architecture of the Proposed Enhancement

A. Behavioral Verification

The VHDL code describing the behavioral functionality of QRS detection algorithm is implemented and named as an original code. The QRS algorithm has two main signals: the ready and write enable signals. The ready signal is invoked in every loop and changed with every clock cycle using if statement to check the write enable signal. In case the write enable signal becomes inactive, then the ready signal is activated to read new data. On the other hand, if the write enable signal becomes active in the next clock cycle, the data can be read and the ready signal becomes inactive.

Then, the VHDL code was improved to reduce some operators and enhance the performance. We apply some architectural optimization and enhancement techniques based on schedule and binding on the code to minimize the latency, area and power. The enhancement of these objectives reduces the computations of the design space for different cycle-time values. The cycle-time is constrained to meet a specific value to satisfy the design requirements. The code is named as an improved code. To ensure that the original behavioral functionality is not altered in the integrator chip, a test bench is written and a timing simulation validates the equivalence of both designs.

The test bench was executed for 5000 ns for both the original QRS code and improved QRS code and the simulation results shows that they are comparable. Figs. 3–6 show snap shots of the simulation-timing diagram.

Next, we apply certain optimizations and enhancements on the original code and then synthesize the improved code for minimizing latency, area and power for several values of cycle-time to satisfy the design specifications and requirements. The (area/latency/power) can be then explored by solving suitable scheduling problems. The results are then compared with the original. The following subsections discuss the complete results analysis.

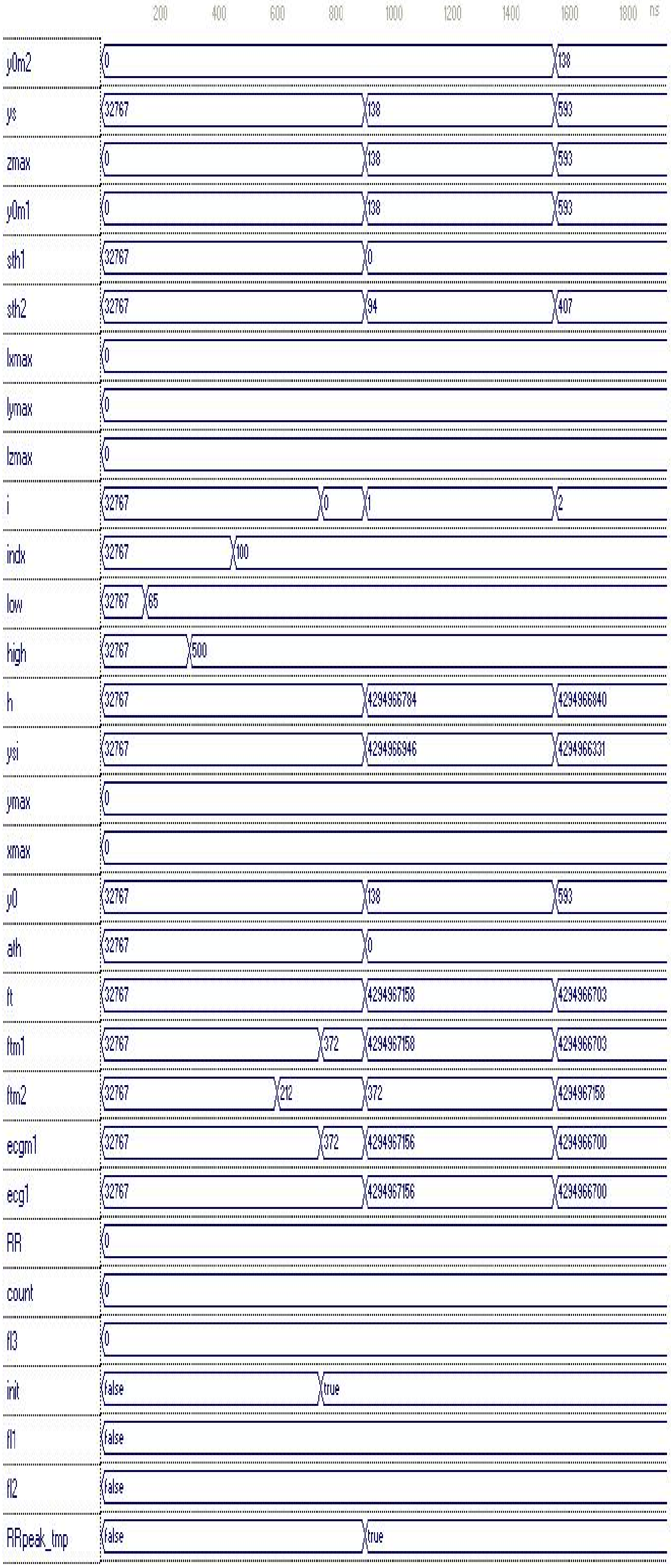


Fig.3: Simulation Timing of the Original Code (0-1800ns)

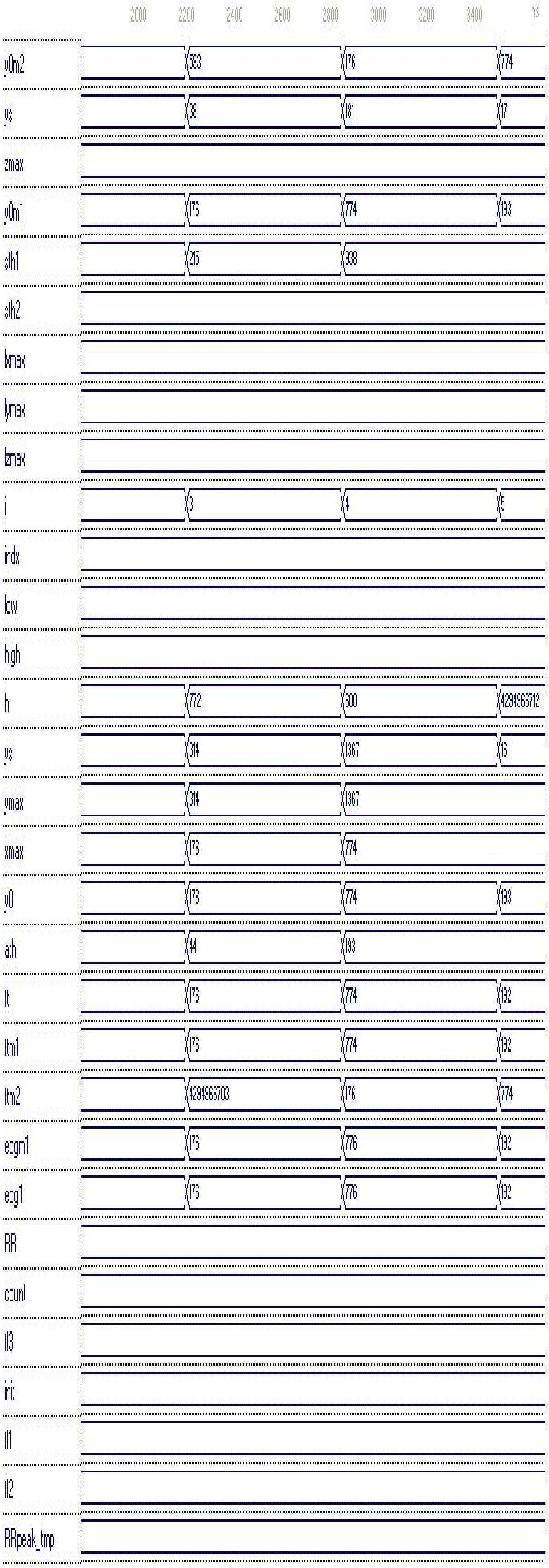


Fig. 4: Simulation Timing of the Improved Code (1800-3600ns)

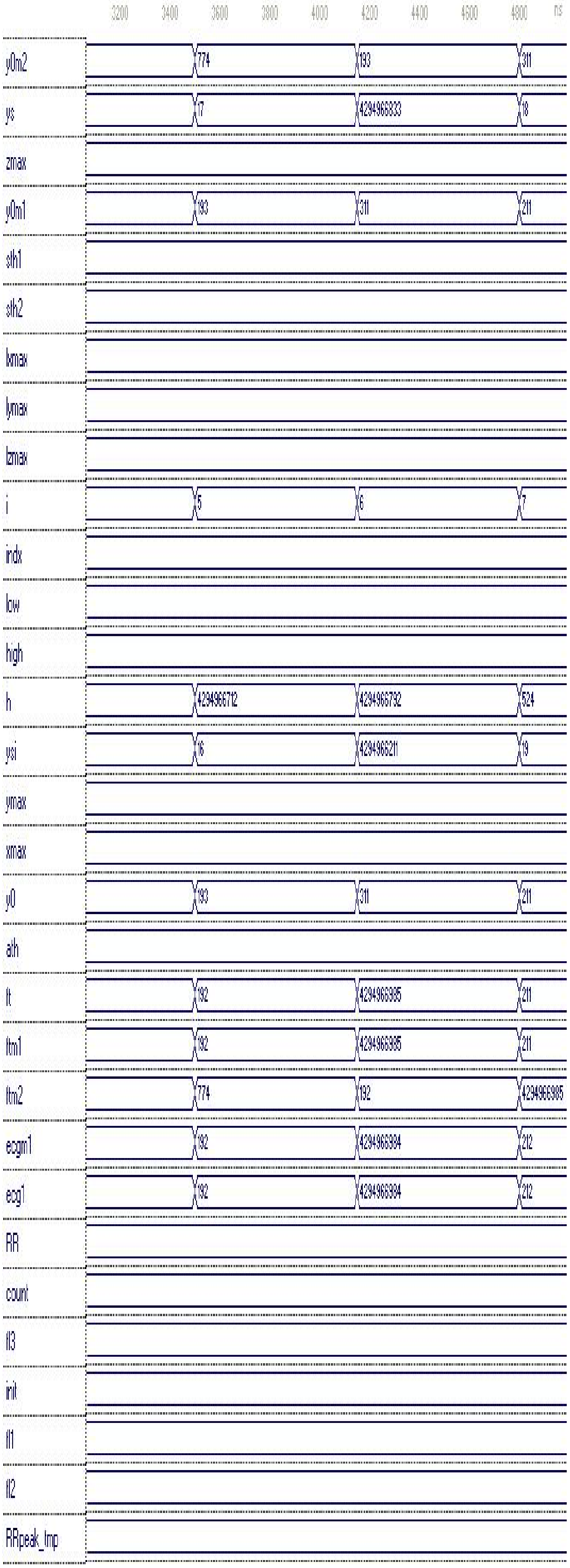


Fig. 5: Simulation Timing of the Original Code (3200-5000ns)

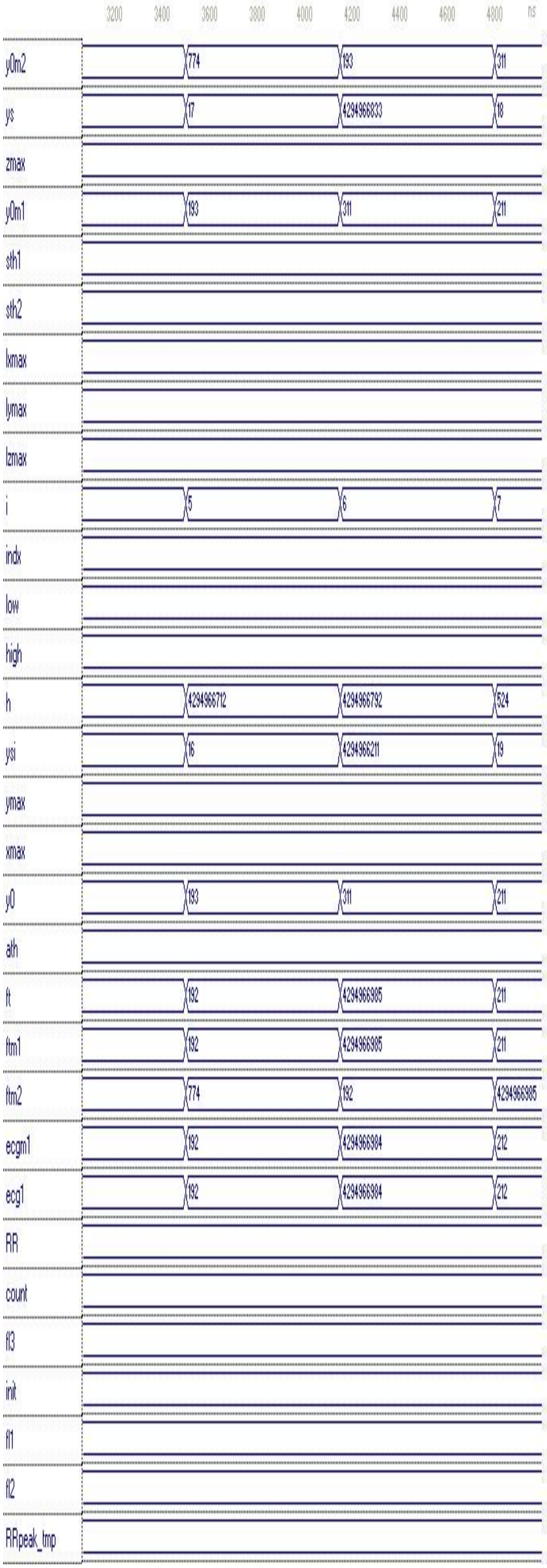


Fig. 6: Simulation Timing Diagram of the Improved Code (3200-5000ns)

In this stage, several cycle-times are selected and the scheduling problem provides us for calculating the (area/latency/power) trade-off points. Solutions to the minimum-latency scheduling problem and to the minimum-resource scheduling problem provide the extreme points of the design space. In between these two extreme solutions, other solutions can be found for solving the minimum-latency resource constrained scheduling problems or the minimum-resources latency constrained scheduling problems for different values of the cycle-time.

Moreover, synthesis is performed using Synopsys tool on the scheduling followed by binding as an initial step to minimize the latency. Later using the extend_latency option with the schedule commands is done to minimize the area. The effect of the power is studied and analyzed based on scheduling and binding.

B. *Enhancing the Design Latency*

At this stage, the main goal is to minimize the latency of the design. Here, scheduling is performed before binding. Binding has impact based on scheduling due to the amount of resource shared and the parallelism of the processes. A minimum latency is attained by applying different cycle-times iteratively and the results compared to get a minimum latency for appropriate cycle-time (i.e. minimum slack).

Behavioral enhancement is a set of semantic transformations that minimize the amount of information required to stipulate the partial order of tasks. Optimization techniques done by the DC automatically selects the best structure for the design based on the high-level architecture. One among the methods that the DC applies is Arithmetic Expression Optimization (AEO). The DC uses the characteristics of arithmetic operators (i.e. associative, commutative, and parenthesis) to rearrange an expression. Other arithmetic enhancements were carried out such as merging cascaded adders with a carry, using tree height reduction arrangement to reduce the delay, applying resource sharing and using operator strength reduction to replace integer division of power 2 by shifts so that it produces an optimized implementation [24].

On the other hand, the compiler was unable to handle or apply few optimizations such as managing hierarchies. The technique that we have applied in enhancing the original code is by balancing the branch statements. We removed the most demanding operations out of the branch legs, which allowed the compiler to do resource sharing more efficiently.

Table I shows the latency/Area trade-off of synthesizing the original and improved code for different values of the cycle-time using the timing estimation report.

From Table I, we note that the latency and area are 37-clock cycle and 1565 units (combinational and sequential area), respectively for the original code and 35-clock cycle and 1484 units, respectively for the improved one at cycle-time of 10 ns. The achieved results are significant since the slack time is positive and at an acceptable point to satisfy the timing goal. In addition, selecting a clock cycle = 10 does not generate any

Table I. Latency/Area Trade-off for Different Cycle-Time

	Minimum Latency (Original Code)			Minimum Latency (Improved Code)		
CC	Latency (ns)	Area (unit)	Slack	Latency (ns)	Area (unit)	Slack
8	50	1836.5	0.07	44	1853	0.42
9	38	1977	0.02	35	1883	0.02
10	**37**	**1565**	**0.88**	**35**	**1484**	**0.35**
15	35	1540.5	4.44	35	1551	5.98
20	35	1567.5	5.7	35	1523	8.12
25	35	1560.5	10.98	35	1651	12.13
30	35	1440	16.20	35	1688	14.81

multi-cycle operations as in the case of clock cycle = 8. Based on this, the appropriate speed that satisfies the chip requirements is around 100MHz.

C. Enhancing the Design Area

The goal is to minimize the area of the design using an extend_latency option and the allocation_effort option with scheduling commands; binding is performed first and then scheduling. In this case, operation pair with shared resources cannot execute simultaneously. The best area achieved is by applying appropriate cycle-times iteratively and comparing the results for the best minimum area at minimum slack time. Table II shows the area/latency trade-off of synthesizing the original code for different values of the cycle-time.

Table II. Area/Latency Trade-off for Different Cycle-Time

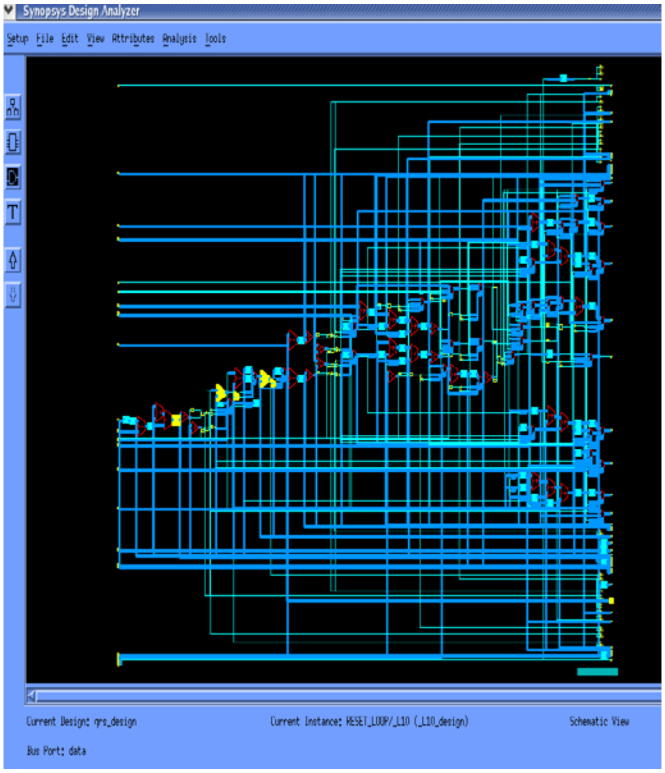
	Minimum Area (Original Code)			Minimum Area (Improved Code)		
CC	Latency (ns)	Area (unit)	Slack	Latency (ns)	Area (unit)	Slack
8	45	2352	0.02	41	2954.5	0.02
9	**45**	**1776**	**1.38**	**41**	**1591**	**1.08**
10	45	1738	2.38	41	1591	2.08
11	45	1738	7.38	41	1591	7.08
15	45	1738	12.38	41	1591	12.08
20	45	1738	17.38	41	1591	17.08
25	45	1738	22.38	41	1591	22.08

Table II shows that at cycle-time of 9 ns, the latency and area values are 45-clock cycle and 1776 units for the original code and 41-clock cycle and 1591 units for the improved one with an acceptable slack time. The following parameters are used in the timing and area estimation report for the improved code; the overhead is due to steering, storage, and control logic as shown in Table III.

Table III. Area/Latency Parameters for the Improved code

Parameters	Minimum Latency	Minimum Area
Cycle Margin	2.32	2.32
FSM	0.50	0.50
MUX	1.00	1.00
FF	0.82	0.82
Chained Operation	10	9
Multi-cycle Operations	10	9
Controller states	35	41

The scheduled output file represents the circuit after applying behavioral synthesis; a hardware language represents it where each component (data path, controller, register and other modules) has two parts: firstly the interface, which defines its input and output ports and secondly the architecture that defines its structure. Instances of these modules are created and used to build the complete design. Fig. 7 shows the critical path (marked in yellow color) of the design generated using design analyzer.



Current Design: qrs_design

Current Instance: RESET_LOOP/_L10 (_L10_design)

Schematic View

Bus Port: data

Fig. 7: Critical Path of the Improved code

D. Enhancing the Design Power

There has been several power reduction techniques used to enhance the power such as clock gating [25] and gate level power optimization. The method employed by this paper for power reduction of QRS detection algorithm is restricted by the latency/area enhancement applied previously. Taking into consideration that a major amount of dynamic power is paid in the distribution of the clock due to the highest toggle rate of the clock, an enhancement is done to reduce the power consumption by turning off the clocks when they are not needed.

The performance parameters such as power and area estimation are reported using Quartus II software. The power consumption is essential for ECG applications. In this proposed technique, the power consumption is determined based on Altera’s FPGA implementation under the latency/area constraints as shown in Table IV. The total power dissipated in case of enhanced latency is better than the enhanced area by a factor of 23.9%. This reduction in power helps in extending the life of the battery of wearable ECG devices.

Table IV. Various Power Parameters for the Improved code

Parameters	Minimum Latency	Minimum Area
Dynamic Thermal power dissipation	6.17 mw	8.36 mw
Static Power dissipation	38.13 mw	47.37 mw
I/O power Dissipation	17.15 mw	25 mw
Total power dissipation	61.45 mw	80.73 mw

IV. Explanation and Comparison of the Results

Based on the results obtained from analyzing the QRS chip, for minimum latency, the original code needs 37 clock cycles at 10 ns cycle-time and an area of 1565 units while the improved code needs 35 clock cycles and an area of 1484 units. Hence, the improved code offers a saving of two clock cycles and 81 units in area. Among these clock cycles, 24 clock cycles were needed for external synchronization and reading data from an outside source, (two clock cycles for each data read and each takes 12-clock cycles for synchronizations and reads). In addition, due to data dependency for some operations and few arithmetic expressions, the enhancement was very limited in these 24 clock cycles. Therefore, this leaves us with only 13 clock cycles that can be achieved in the 37 clock cycles needed for the original code.

Two clock cycles from the 13 clock cycles were saved by carrying out enhancements and optimizations. The big save is the saving of one adder, which has an individual area of 72.5 units using target library xdc_virtex2-5.db. However, it should be noted from the summary reports that number of 16-bit registers increase by 1 for the improved code. In general, the area was reduced despite the area overhead of the register. Taking intensive operations out of the branch legs makes them available for resource sharing and this helps in reducing the latency. The design archives 5.5% reduction in latency and 5.2% in area at 10 ns.

For minimum area design, the best area was 1776 units with latency of 45 clocks while the improved code has an area of 1591 units and a latency of 41 clocks at 9 ns. The design archives 8.9% reduction in latency and 10.5% in area at 9 ns.

Regarding the power dissipation, it is affected mainly by the area. Therefore, when applying binding followed by scheduling, the area is increased for the original code by a factor of 11.9% and 7%, however applying scheduling followed binding leads to a better reduction in power by a factor of 23.9% at 10 ns latency.

V. Conclusion

QRS detection is frequently used for heart rate analysis. Most prevalent ECG property extraction algorithms apply QRS at the beginning stage. In this paper, we apply several cycle-times on QRS chip using Synopsys tool to synthesize and run scheduling and then binding to reduce the latency, the area and the power. The QRS integrator chip synthesis is done on both the original and the improved code. The enhancement has appeared when minimizing the latency where the design archives 5.5% reduction in latency, 10.5% in area and 24% in power consumption. The proposed enhancement methodology is also applicable to other applications.

As a part of future work, we plan to apply other enhancement techniques to improve the code efficiency and its performance. Moreover, cases in which enhancement could be justified in the term of project complexity can be investigated and analyzed.

References

[1] A. Alwan, “Global status report on noncommunicable diseases 2010,” *World Health Organization*, pp. 1-176, 2010.

[2] S. Kurakula, A.S.D.P. Sudhansh, R. Paily, and S. Dandapat, “Design of QRS detection and heart rate estimation system on fpga,” *Advances in Computing and Communication, Communications in Computer and Information Science*, vol. 193, pp 165-174, 2011.

[3] D. S. Benitez, P. A. Gaydecki, A. Zaidi and A. P. Fitzpatrick, "A new qrs detection algorithm based on the Hilbert transform", *Computers in Cardiology*, vol.27, pp. 399-406, 2000.

[4] N. M. Arzeno, Z. D. Deng and C. S. Poon, "Analysis of first derivative based qrs detection algorithms", *IEEE transaction in Biomedical Engineering*, Vol.55, No.2, pp. 478-484, February 2008.

[5] J. Kovacevic, R. Stojanovic, D. Karadaglic, B. Asanin, Z. Kovacevic, Z. Bundalo, and F. Softic, "FPGA low-power implementation of qrs detectors," *IEEE 3rd Mediterranean Conference on Embedded Computing (MECO)*, pp. 98-101, 2014.

[6] P. Mundhe, and A. K. Pathrikar, "An overview of implementation of efficient qrs complex detector with fpga," *International Journal of Advanced Research in Computer and Communication Engineering-IJARCCE*, vol. 2, no. 10, pp. 4041-4043, 2013.

[7] M. Lascu, and D. Lascu, “LabVIEW event detection using pan-tompkins algorithm,” *Proceedings of the 7th WSEAS International Conference on Signal, Speech and Image Processing (SSIP'07)*, Lang Congyan (Ed.). *World Scientific and Engineering Academy and Society (WSEAS)*, Stevens Point, Wisconsin, USA, pp. 32-37, 2007.

[8] K.A Azad, Z.M. Darus, and M.A. Ali, "Testable ASIC design for a fuzzy logic based qrs complex detector," *Proceeding of ICSE '98. IEEE International Conference on Semiconductor Electronics*, pp.175-178, 1998.

[9] M. Cvikl, and A. Zemva, “FPGA-oriented HW/SW implementation of ecg beat detection and classification algorithm,” *Digital Signal Processing*, vol. 20, no. 1, pp. 238–248, 2010.

[10] H. K. Chatterjee, R. Gupta, J. N. Bera, and M. Mitra, "An fpga implementation of real-time qrs detection," *2nd International Conference on Computer and Communication Technology (ICCCT)*, pp. 274-279, 2011.

[11] G. Selvakumar, and K. B. Bagan, “An efficient qrs complex detection algorithm using optimal wavelet,” *WSEAS Transactions on Signal Processing*, 2(8), 1069-1073, 2006.

[12] N. V. Thakor, J. G. Webster, and W. J. Tompkins, “Estimation of qrs complex power spectra for design of a qrs filter,” *IEEE Transactions on Biomedical Engineering*, vol. 11, 702-706, 1984.

[13] C. I. Ieong, M. I. Vai, and P. U. Mak, “Ecg qrs recognition with programmable hardware,” *2nd IEEE International Conference on Bioinformatics and Biomedical Engineering (ICBBE*), pp. 2028-2031, May 2008,

[14] C. F. Zhang, and T. W. Bae, “Vlsi friendly ecg qrs complex detector for body sensor networks,” *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 2(1), 52-59, 2012.

[15] T. Mar, S. Zaunseder, J. P. Martinez, M. Llamedo, and R. Poll, “Optimization of ecg classification by means of feature selection,” *IEEE Transactions on Biomedical Engineering*, 58(8), 2168-2177, 2011.

[16] D. B. Nielsen, K. Egstrup, J. Branebjerg, G. B. Andersen, and H. B. Sorensen, “Automatic qrs complex detection algorithm designed for a novel wearable, wireless electrocardiogram recording device,” I*EEE Annual International Conference of the Engineering in Medicine and Biology Society (EMBC)*, pp. 2913-2916, August 2012.

[17] R. Stojanović, D. Karadaglić, M. Mirković, and D. Milošević, "An fpga system for qrs complex detection based on integer wavelet transform," *Measurement Science Review*, vol. 11, Issue 4, pp. 131-138, June 2011.

[18] N. Nagpal, M. Chawla, and D. Phillips, “Simple fpga based ecg R wave peak detection system for heart rate and heart rate variability calculation,” *International Journal of Engineering Research and Applications*, pp. 33-36, 2014

[19] A. G. James, and D. J. Farrell, “An fpga implementation of a digital qrs detector,” *Proceedings of the International Conference on Signal Processing and Technologies (ICSPAT)*, pp. 1-6, 1999.

[20] M. Mahalakshmi, and M. Vaijayanthi, “Hardware implementation of ecg qrs complex detection using fpga,” *IOSR Journal of Computer Engineering*, vol. 3, pp. 20-28, 2014.

[21] M. I.Ibrahimy, M. B. I. Reaz, F. Mohd-Yasin, T. H. Khoon, and A. F. Ismail, “Fetal qrs complex detection algorithm for fpga implementation,” *International Conference on Computational Intelligence for Modelling, Control and Automation, and IEEE International Conference on Intelligent Agents, Web Technologies and Internet Commerce*, vol. 1, pp. 846-850, November 2005.

[22] T. Torfs, C. J. Smeets, D. Geng, T. Berset, J. Van der Auwera, P. Vandervoort, and L. Grieten, “Clinical validation of a low-power and wearable ecg patch for long term full-disclosure monitoring,” J*ournal of electrocardiology*, 47(6), 881-889, 2014.

[23] S. Abed, and S. Alouneh, “Optimization on qrs chip for minimum latency,” *Proc. of the 14th Inter. Conf. on Software Engineering, Parallel and Distributed Systems (SEPADS'15), Advances in Software Engineering and Systems*. Dubai, UAE, pp. 150-155, Feb. 22-24, 2015.

[24] S. Abed, A. Al-Khasawneh, and S. Alouneh, “Synthesizing a heart rate analyzer chip qrs”, *International Journal of Advancements in Computing Technology (IJACT)*, vol. 5, no. 11, pp. 126-137, *Advanced Institute of Convergence Information Technology*, DOI: 10.4156/ijact.vol5.issue11.14, July 2013.

[25] M. Pedram, “Power minimization in IC design: principles and applications”, *ACM Transactions on Design Automation of Electronic Systems*, vol. 1, no. 1, pp. 3–56, January 1996.

Sahel Alouneh received his B.Sc. in electrical and computer engineering from Jordan University of Science and Technology, Jordan in 2000. His M.Sc. and Ph.D were obtained from Concordia University in 2004 and 2008 respectively.

He is currently working at the German Jordanian University as an associate professor in the Computer Engineering Department. In addition, he worked as the director of the computer center. He brings more than ten years of professional and academic expertise in computer networks, fault tolerance, software engineering and security. He has also held many lead positions in network security and assurance at GJU. He was appointed as the computer center at GJU. He was appointed as vice dean for Scientific Research and Graduate Studies deanship at GJU. His research interest in includes computer networks fault tolerance, security, software engineering, and computer hardware design.

Mohammad H. Al Shayeji, received his B.Sc. (Computer Engineering), from the University of Miami, and M.S. (Computer Science) from University of Central Florida. He got his Ph.D. (Computer Science) from the University of Southern California.

He is working as Assistant Professor in Computer Engineering Department, College of Computing Sciences and Engineering, Kuwait University. He has several publications in different international Journals and Conferences. His research interest includes VOD, Video Servers, Multimedia DMS, and Distributed Systems.

Sa’ed Abed received his B.Sc. and M.Sc. in Computer Engineering from Jordan University of Science and Technology (J.U.S.T.), Jordan in 1994 and 1996, respectively. In 2008, he received his Ph.D. in Computer Engineering from Concordia University, Canada.

He has previously worked as lecturer at King Faisal University in Saudi Arabia from 1997 until 2003. He joined Hashemite University, Jordan, as Assistant Professor from 2008 until 2014. Currently, Dr. Abed is an assistant professor in the Department of Computer Engineering at Kuwait University. His research interests include Formal Methods, Theorem Proving, Model Checking, SAT-Solvers, VLSI Design and Automation, Synthesis, Computer Architecture and Fault Tolerance. Dr. Abed also served as a reviewer for various international conferences and journals.

M.D Samrajesh received his B.Sc, from Bharathiar University, and Master of Computer Applications from Bharathidasan University. He obtained his M.Phil in the field of Computer Science from Manonmaniam Sundaranar University.

He has many publications in various national and international Journals and Conferences. His research interest includes Distributed Systems, Video-on-Demand, and Software Engineering.

Sari Sultan earned his B.Sc. in Computer Engineering from the Hashemite University, Jordan. Currently he is a researcher who published multiple journal, conference papers. Also nominated as top finalist in multiple conferences such as KAUST WEP 2012 (KSA), and 5th NTP (Jordan). In addition, he is an instructor for ethical hacking, as he trained more than 2000 IT Professionals in Jordan, and Tunisia. He also hold ANSI accredited certifications such as Certified Ethical Hacker CEH, and Computer Hacking Forensic Investigator CHFI. His research interest involves computer architecture, VLSI and Network.