10.2478/v10048-011-0024-z
MEASUREMENT SCIENCE REVIEW, Volume 11, No. 4, 2011

A FPGA system for QRS complex detection based on Integer Wavelet Transform

R. Stojanović[1], D. Karadaglić[2], M. Mirković[1] and D. Milošević[3]

[1]Faculty of Electrical Engineering, University of Montenegro, Džordža Vašingtona bb, 81000, Podgorica, Montenegro, stox@ac.me
[2]University of Manchester, Oxford Road, M13 9PL, Manchester, United Kingdom, Dejan.Karadaglic@manchester.ac.uk
[3]University of Kragujevac, Technical Faculty Čačak, Serbia, danijela.tfc@gmail.com

Due to complexity of their mathematical computation, many QRS detectors are implemented in software and cannot operate in real time. The paper presents a real-time hardware based solution for this task. To filter ECG signal and to extract QRS complex it employs the Integer Wavelet Transform. The system includes several components and is incorporated in a single FPGA chip what makes it suitable for direct embedding in medical instruments or wearable health care devices. It has sufficient accuracy (about 95%), showing remarkable noise immunity and low cost. Additionally, each system component is composed of several identical blocks/cells what makes the design highly generic. The capacity of today existing FPGAs allows even dozens of detectors to be placed in a single chip. After the theoretical introduction of wavelets and the review of their application in QRS detection, it will be shown how some basic wavelets can be optimized for easy hardware implementation. For this purpose the migration to the integer arithmetic and additional simplifications in calculations has to be done. Further, the system architecture will be presented with the demonstrations in both, software simulation and real testing. At the end, the working performances and preliminary results will be outlined and discussed. The same principle can be applied with other signals where the hardware implementation of wavelet transform can be of benefit.

Keywords: QRS detection, FPGA, Wavelet Transform.

1. INTRODUCTION

THE QRS COMPLEX is the most important segment in electrocardiogram (ECG) signal and its shape, moment of occurrence and frequency gives us indispensable information about heart condition. Many methods were attempted for QRS complex detection, and an extensive review of the approaches proposed in the last decade can be found in [1].

The results of studies of these approaches have demonstrated that the Wavelet Transform (WT) in its discrete form - Discrete Wavelet Transform (DWT) - is the most reliable method among all of them to extract critical points from the ECG signal. However, the majority of published algorithms are heavily processor demanding, therefore do not allow real-time operation on a personal computer.

In the recent years, significant research, academic and industrial attention has been turned to the Telemedicine and Wearable Health Care (WHC) systems. Such systems are based on autonomous ultra-low-power devices capable of performing real-time sampling, signal processing and wireless transfer for what they employ general purpose microprocessors/microcontrollers and very often, in last time, Programmable Logic Devices (PLDs) like Field Programmable Gate Arrays (FPGAs).

A FPGA consists of reconfigurable logic, I/O and interconnections blocks and differs from microcontrollers and DSP processors, which are Von Neumann type of machines. In addition to the parallelism, other advantages include, but not limit to: higher throughput, low price, flexibility of design, testing and rapid prototyping as well as an ability to transform digital design directly to Application Specific Integrated Circuits (ASICs). Therefore, there is an emphasised need to implement bulk of the software algorithms for biomedical signal processing in this technology.

There are few articles about implementing QRS detection in FPGA. References [2] and [3] implement Pan-Tompkins algorithm, while [4] uses the wavelets. They employ complex filtering and mathematical calculations as well as sophisticated state-machine blocks. It results in a complex design and huge requirements on silicon resources.

In this paper a cost and performance effective design methodology will be proposed, which combines the positive characteristics of Wavelets in QRS detecting and FPGAs in hardware-based signal processing. The final developed system works in real time, does not require memory and is modular in principle. During the design and construction, the care has been taken to minimise total number of adjustable parameters.

In Section 2 the related theoretical background on WT with emphasis to its optimization in integer arithmetic will be elaborated briefly. The principle of WT based detection of QRS complexes will be explained in Section 3. Section 4 will deal with proposed design approach from system and component point of view with focus to software simulation and verification of proper work. The testing procedure and preliminary results will be presented in Section 5.

2. THEORETICAL BACKGROUND

A. Wavelet transform

WT is a time-scale representation that has been used successfully in a broad range of applications, and most typically in signal compressions. Recently, the WT has been applied to several problems in electrocardiology,

including data compression, denoising, analysis of ventricular late potentials and detection of ECG characteristic points. It is a linear operation that decomposes signal into a number of scales related to frequency components and analyses each scale with a certain resolution. WT uses a short time interval for evaluating higher frequencies and a long time interval for lower frequencies. Because of this property, high frequency components of short duration can be observed successfully by WT. One of the advantages of the WT is ability to decompose signal at various resolutions, what allows an accurate feature extraction from non-stationary signals like ECG.

Analytically, the WT in its continuous form, for signal *f(t)* is defined by:

$$W(a,b) = \int_{-\infty}^{\infty} f(t)\psi_{a,b}(t)dt \tag{1}$$

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}}\psi^*(\frac{t-b}{a}) \tag{2}$$

Where * denotes complex conjugation and $\psi_{a,b}(t)$ is window function called the mother wavelet, a is a scale factor and b is a translation factor. Here, $\psi^*(\frac{t-b}{a})$ is a shifted and scaled version of a mother wavelet which is used as a basis for wavelet decomposition. However, the continuous wavelet transform provides certain amount of redundant information.

DWT is sufficient for most practical applications providing enough information and offering a significant reduction in the computation time. For a discrete function *f*(*n*) the DWT is given by:

$$W(a,b) = C(j,k) = \sum_{n\in Z} f(n)\psi_{j,k}(n) \tag{3}$$

where $\psi_{j,k}(n)$ presents a discrete wavelet defined as $\psi_{j,k}(n) = 2^{\frac{-j}{2}}\psi(2^{-j}n-k)$. The parameters a, b are defined as $a = 2^j$ and $b = 2^j k$.

In practice, DWT is computed by passing the signal through a Low-Pass (L) and a High-Pass (H) filters successively according to the Mallat's decomposition scheme [5], as shown in Fig. 1. For each decomposition level *i*, the L and H filters are followed by downsampling operator $\downarrow 2$ which can be expressed as $(X\downarrow 2)[n]=X[2n]$, what is in fact the reduction of sampling rate by 2. Therefore, the outputs *CAi(n)* and *CDi(n)* from one decomposition level, in our case the first one, *i=1*, are given by:

$$CA_1[n] = \sum_k L[k]X[2n+k] \tag{4}$$

$$CD_1[n] = \sum_k H[k]X[2n+k] \tag{5}$$

where n and k denote discrete time coefficients and X stands for the given signal. In the second decomposition level the $CA_1[n]$ plays a role of $x[n]$ and so on. *CAi(n)* and *CDi(n)* are usually called the *approximations* and the *details*, respectively.

The coefficients for L and H filters can vary from the simplest, such as Haar, over Doubisch up to those such as Quadratic Spline, having different vector lengths and, usually, floating point values. The L and H filters, within L-H cell, Fig.1, can be easily implemented in hardware only in the case of simplification their transfer functions. It means both; the reduction of vector length and escaping floating point calculations. Thereby, the positive characteristics of the wavelets should be retained.

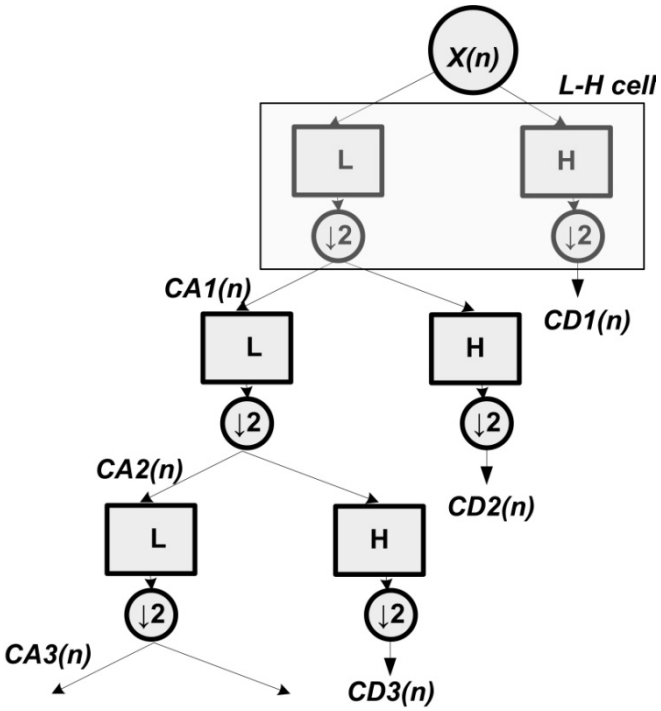

Fig.1. Wavelet decomposition scheme.

B. Wavelet transform

The Haar wavelet is considered to be the simplest one. In discrete form it is related to the Haar transform, which serves as a prototype for all wavelet transforms. The L and H filters are of two elements wide, $L = \left[\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right]$, $H = \left[\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}\right]$. The $CA_1(n)$ and $CD_1(n)$ are given by:

$$CA_1[n] = \frac{1}{\sqrt{2}}X[2n] + \frac{1}{\sqrt{2}}X[2n+1] \tag{6}$$

$$CD_1[n] = \frac{1}{\sqrt{2}}X[2n] - \frac{1}{\sqrt{2}}X[2n+1] \tag{7}$$

This transform has a number of advantages; it is (i) conceptually simple, (ii) fast, (iii) memory efficient, since it can be calculated in a place without a temporary array. Also, it is reversible without the edge effect which can be a problem with some other Wavelet transforms.

The Haar transform also has its limitations, which can be of a problem in some applications, mainly in signal compression and noise removal from audio or video signal. But, in our case this is not an issue.

C. Integer Haar Transform

Although very simple in principle, the Haar transform in above form is still complicated enough for hardware implementation, because of the floating point nature. However, like any other wavelet transform, it can be generalized to an integer to integer version as shown in the excellent reference [6]. Its integer form is known as S transform [7]:

$$CA_1[n] = \left\lfloor \frac{1}{2}X[2n] + \frac{1}{2}X[2n+1] \right\rfloor \tag{8}$$

$$CD_1[n] = X[2n] - X[2n+1] \tag{9}$$

Where $\lfloor\ \rfloor$ denotes rounding operator.

On the binary level the division with 2 and rounding are performed by right shifting for one position, >>1, which is very helpful circumstance in the term of digital logic implementation. With this, equation (8) becomes:

$$CD_1[n] = (X[2n] + X[2n+1]) >> 1 \tag{10}$$

As seen, the $CA_1[n]$ is calculated by an adder and shifter and the $CD_1[n]$ by a subtractor, excluding the necessity for floating point multipliers. Now, the L-H cell from Fig.1 should be very easy implemented in digital logic. The rounding to the nearest integer does not implicate the significant error in signal reconstruction [6].

3. Wavelet Decomposition and QRS Detection

WT is capable of distinguishing the QRS-complex within ECG signal by using decomposition scheme from Fig.1, [8], [9]. The *CD* coefficients across the scales show that the peak of the QRS complex corresponds to the zero crossing between two modulus maxima.

Fig.2 illustrates the decomposition of real ECG signal *X(n)* up to the 4^{th} level , (*CD1, CD2, CD3* and *CD4*), using Haar wavelet from (6) and (7). For each decomposition level, the QRS complex produces two modulus maxima with opposite signs, with a zero crossing (ZC) between.

This method for QRS detection is very robust and allows direct application over the raw ECG data because the frequency domain filtering is performed implicitly by computing the coefficients. Fig.2 shows the clear signal at 4^{th} decomposition level. The built-in filtering is an important additional contribution of wavelet decomposition in QRS detection.

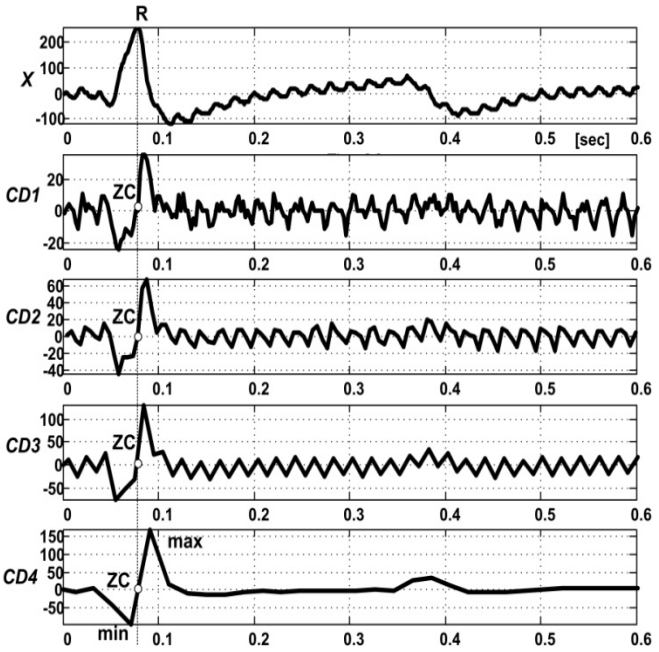

Fig.2. QRS detection using wavelet decomposition. Signal *X[n]* sampled with frequency of 800 Hz.

Usually, the modulus maxima are found by thresholding techniques and the threshold values vary from one scale to another. As an example, according to [9] the thresholds are proportional to the RMS value of the WT and can be calculated as *RMS(Di)* for *i=1,2* and *3* and *0.5RMS(Di)* for *i=4*.

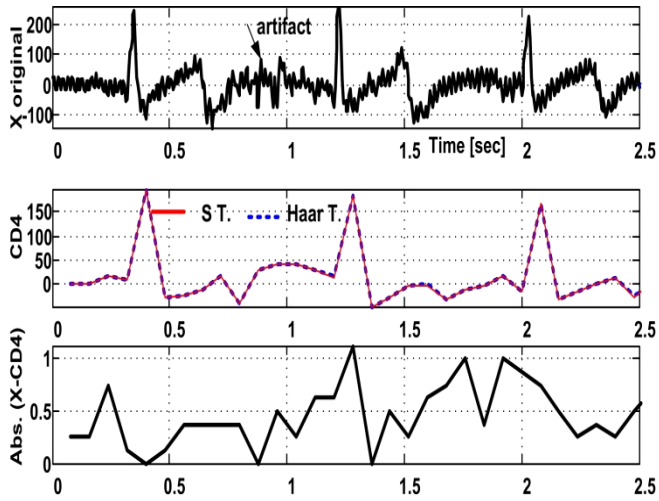

Fig.3. Matching the S Transform and Haar Transform.

In the practical applications the challenge is the selection of the most suitable decomposition level/levels. Most of the energy for QRS complex lies between 3Hz and 40Hz. Translated to WT it means somewhere between scales 2^3 and 2^4, with the largest at 2^4. The energy of motion artifacts and baseline wander (i.e., noise) increases for scales greater than 2^5. Article [9] states that most energies of a typical QRS-complex are at scales 2^3 and 2^4, and the energy at scale

2^3 is the largest. According to [11], for QRS-complex with high frequency components, the energy at scale 2^2 is larger than that at scale 2^3. The authors recommend mainly the scales 2^3 to 2^4 for satisfactory detection.

Another complication is the acquisition of certain thresholds for finding the modulus maxima. the values of thresholds differ usually from one level to another. The third obstacle is the change of decision rules according to circumstances and it is a very ambiguous and time-consuming task. Mentioned restrictions confine the method to off-line use and put heavy demand on the computing resources.

In the presented research, instead of classical Wavelet Transform, additionally optimised Integer Wavelet Transform (S Transform) is used for the task of QRS detection. Fig.3 shows that the absolute error is less than one count at 4^{th} coefficient when using S transform given by (9) and (10) instead of Haar transform given by (6) and (7).

4. THE DESIGN APPROACH

A. System Architecture

Simplified diagram of the proposed methodology for QRS complex detection is given in Fig.4. Wavelet decomposition is realized with pipelined L-H cells. Each cell is followed by branch consisting of two circuits/blocks: *Zero Crossing and Modulus Thresholding (ZC&MT) and Decision Rules (DR)*. The outputs from ZC&MT and DR are fed to the MUX and summing (OR) circuits. The input to the system is the ECG vector *X[n],* driven in by system clock *clk*, while the main outputs are the true zero crossings (*ZCi_OK*), summing of true zero crossings (*SUM_ZC*) and vector of time stamps which corresponds to the position of *R* peak (*RTIME*) in time scale. *SUM_ZC* and *RTIME* are obtained by summation of *ZCi_OKs* and multiplexing *RTIMEi* by address *SL*. *R* presents the asynchronous reset which puts all outputs to zero. The threshold parameters *TR1* and *TR2* as well as parameters *Ti_1* and *Ti_2* related to the QRS inclination times are stored in configuration registers or internal Read Only Memory (ROM). As seen, the architecture can be extended very easily to the desired decomposition level by adding a new branch of L-H, ZC&MT and DR. Also, in the case of reduction, unnecessary branches can be omitted.

B. Wavelet decomposition

L-H cell from Fig.5 is the core unit in decomposition scheme from Fig.4. It presents digital expression of equations (9) and (10). The delay line is implemented by cascade of *REG1* and *REG2*. The ((*X(n)+X(n+1))>>1*)) and *X(n)-X(n+1)* computations are done for each sample. The signal *clk_out=clk_in/2* is used for downsampling (↓2) and stores *X(2n)-X(2n+1)* and ((*X(2n)+X(2n+1))>>1*)) values in adequate output registers REGs. *Ca* output from the previous cell is the input *X* to the further cell and so on. The calculation time per *Ca* and *Cd* is *2clk*. The signal *clk_out* presents a *clk_in* for the next cell. The implementation of L-H cell in Fig.5 is a very fast way to calculate equations (9) and (10) without multipliers.

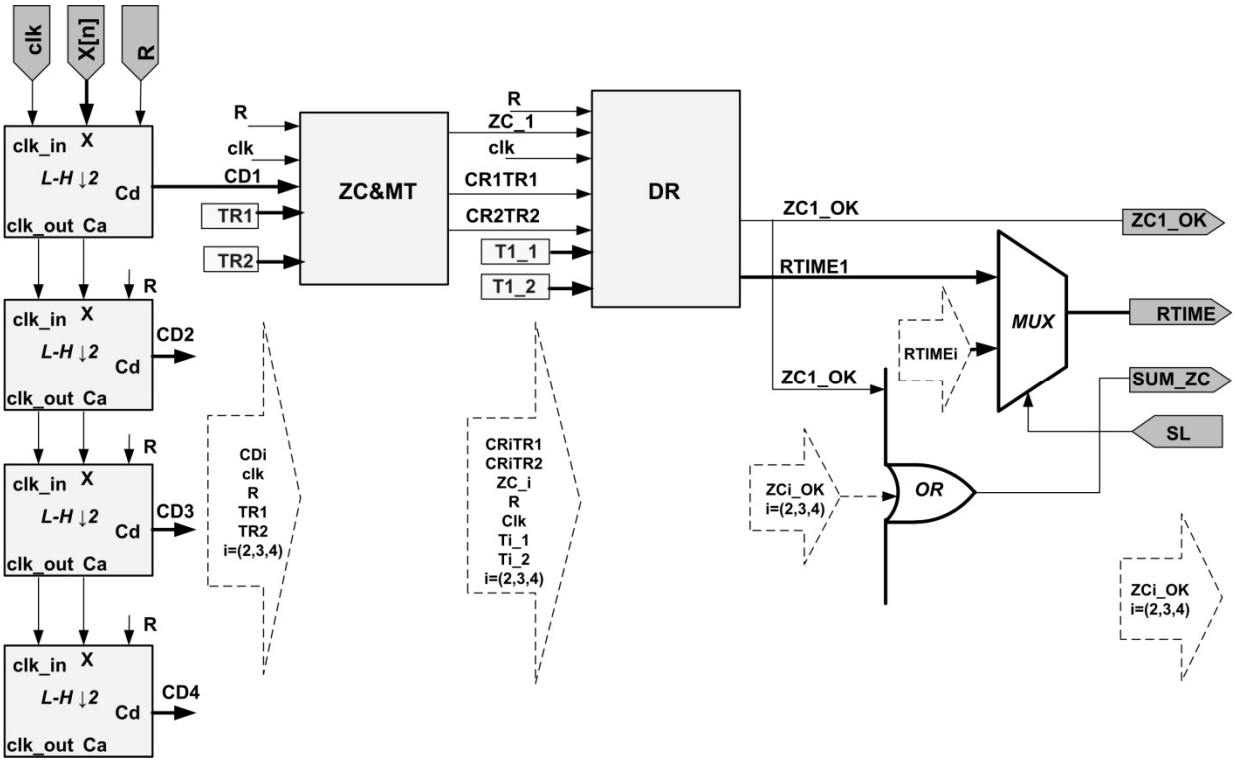

Fig.4. Simplified architecture of FPGA based QRS detection

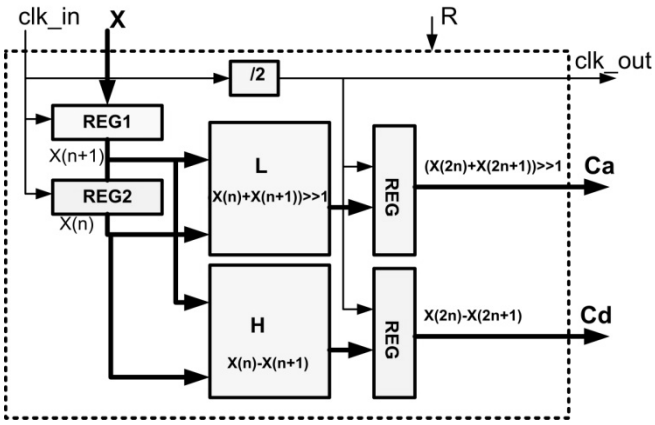

Fig.5. Architecture of L-H cell .

C. Zero crossing and modulus thresholding

In order to detect zero crossing (*ZC_i*) for the chosen coefficients *CDi*, it is necessary to recognize the modulus maxima pair, Fig.6 (up). Instead of using a classical way based on the local max-min finding ZC&MT technique has been employed in this paper.

Negative (min) and positive (max) peaks are isolated by negative and positive thresholds *TR1* and *TR2*, producing the digital signals *CRiTR1* and *CRiTR2*, with zero crossing *ZC_i* between, Fig.6 (middle). Usually, $TR2=abs(TR1)=k*Max(X(n))$, where k varies between $0.05<k<0.5$. For the purpose of demonstration in Fig.6 $TR2<abs(TR1)$ is selected. The signal *ZC_i* is found by scanning successive samples such that $CD(n-1)<0$ and $CD(n)>0$.

D. Decision making

For each decomposition level the Decision Rules (DR) circuit receives the signals *CRiTR1*, *CRiTR2* and *ZC_i*, from ZC&MT, Fig.6(middle). The detection of true zero crossing begins on the falling edge of signal *CRiTR1* by starting the counter driven by system clock CLK, Fig.6 (down). The counter has been previously reset by rising edge of *CRiTR1*. The counter stops on the falling edge of *CRiTR2* latching the time t_0. If $Ti\_1<t_0<Ti\_2$ the detected *ZC_i* is declared to be the true one. Only then the trigger *ZCi_OK* is generated. If the rising edge of *CRiTR2* is not detected within time *Ti_2*, the detected *ZC_i* is considered to be false, the counter cleared, and trigger *ZCi_OK* remains zero. If *ZC_i* is declared to be true it is necessary to store the time when it occurred (*RTIMEi),* because this time corresponds to the position of *R* peak. It is ensured by double latch, Fig.6 (down); first when *ZC_i* happened, whether it is true or not, and then when the *ZC_i* is declared as true, namely *ZCi_OK* generated. By multiplexor MUX, Fig.4, the *RTIME* will be taken from *RTIMEi* which corresponds to the better accuracy. As an example, if *R* peak is detected at all four levels the *RTIME1* will be forwarded out. As an alternative the mean value, *mean(RTIMEi, RTIMEi, RTIMEi, RTIMEi),* can be considered.

Here, a possible complication may arise from selection of parameters *Ti_1* and *Ti_2* which depend on the inclination and duration times within the QRS complex. These values are recalculated in integer values considering frequency of system clock (clk). In developed system $30ms<T4\_1<50ms$, $90ms<T4\_2<150ms$, $15ms<T3\_1<25ms$ and $45ms<T3\_2<75ms$ are used for 4^{th} and 3^{rd} levels.

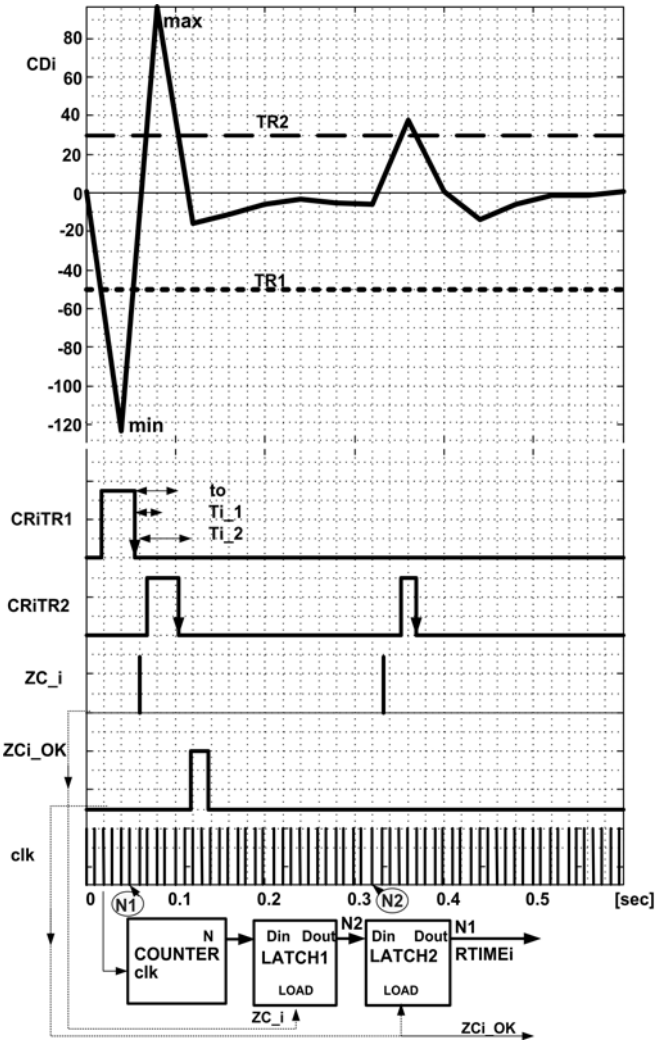

Fig.6. Graphical illustration of the algorithm for finding modulus maxima and zero crossing. Bottom is a decision making technique.

Generally, the system can be implemented till highest level, in our case 4^{th}. The signals *ZC1_OK*, *ZC2_OK, ZC3_OK* and *ZC4_OK* with corresponding values *RTIME1, RTIME2, RTIME3* and *RTIME4* have to be considered. A number of *ZCi_OK* within *SUM_ZC* and adequate *RTIMEi* are sufficient information for decision. In practice two levels (3^{rd} *and* 4^{th}) are enough for satisfactory detection rate, using the signals *ZC1_OK*, *ZC2_OK, RTIME1* and *RTIME2.* In this case the resources are less occupied, and thresholds are easier to adjust.

E. Simulation

The simulation is performed using real ECG samples brought to the system input *X[n]*. For this purpose a separately circuit/block (simulation generator) is designed and embedded into same chip. It consists of an up-counter which addresses the FPGA's ROM in incremental order. The counter is driven by clock whose frequency is identical

to the ECG sampling frequency. The ROM is initialized through memory initialization (*.mif) file, which is in fact an array of test samples *X[n], X[n+1]....* Thus, the simulation process is synchronized using the real samples. After the simulation and verification, before chip configuration, the simulation generator can be removed.

Fig.7 shows the timing diagrams in characteristic points for *3rd* decomposition level. The ECG segment with R peak was isolated as a test sequence. The *TR1* and *TR2* were adjusted at -30 and 30 and signals *D3, CR3TR1, CR3TR2, ZC_3, ZC3_OK* and *RTIME* observed. As seen, *ZC3_OK* is generated and *RTIME=RTIME3* captured. This means the successful detection of R peak.

In Fig.8 the scenario simulated was as all four decomposition levels worked in parallel. Test signal of 1024 samples was acquired in sampling frequency of 400Hz (2.5ms), in total duration of 2.5s. The thresholds *TR1* and *TR2* were adjusted to -22 and 22. The R peak has been detected at all decomposition levels. ZC1_OK, ZC2_OK, ZC3_OK and ZC4_OK are summed in *SUM_ZC*. Because of the accuracy reason the *RTIME* was captured from *1st* level, *RTIME =RTIME1*. Previously the same signal had been passed through identical MATLAB detector, measuring R-R intervals. As seen, the good matching is obtained, 0.9915s against 0.9900s and 0.9330s against 0.9150s respectively.

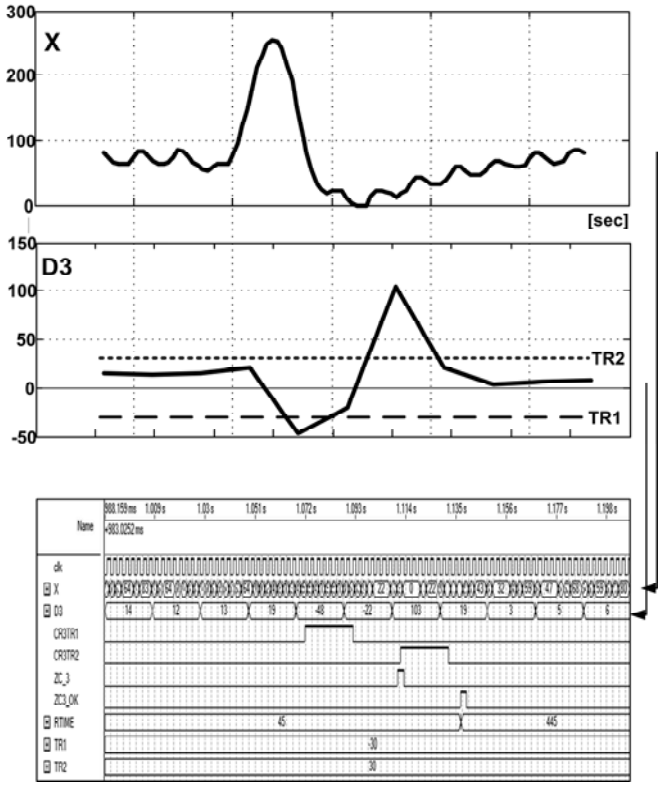

Fig.7. Simulation of decomposition, ZC&MT and DR blocks for ECG sequence with R peak. 400Hz sampling frequency.

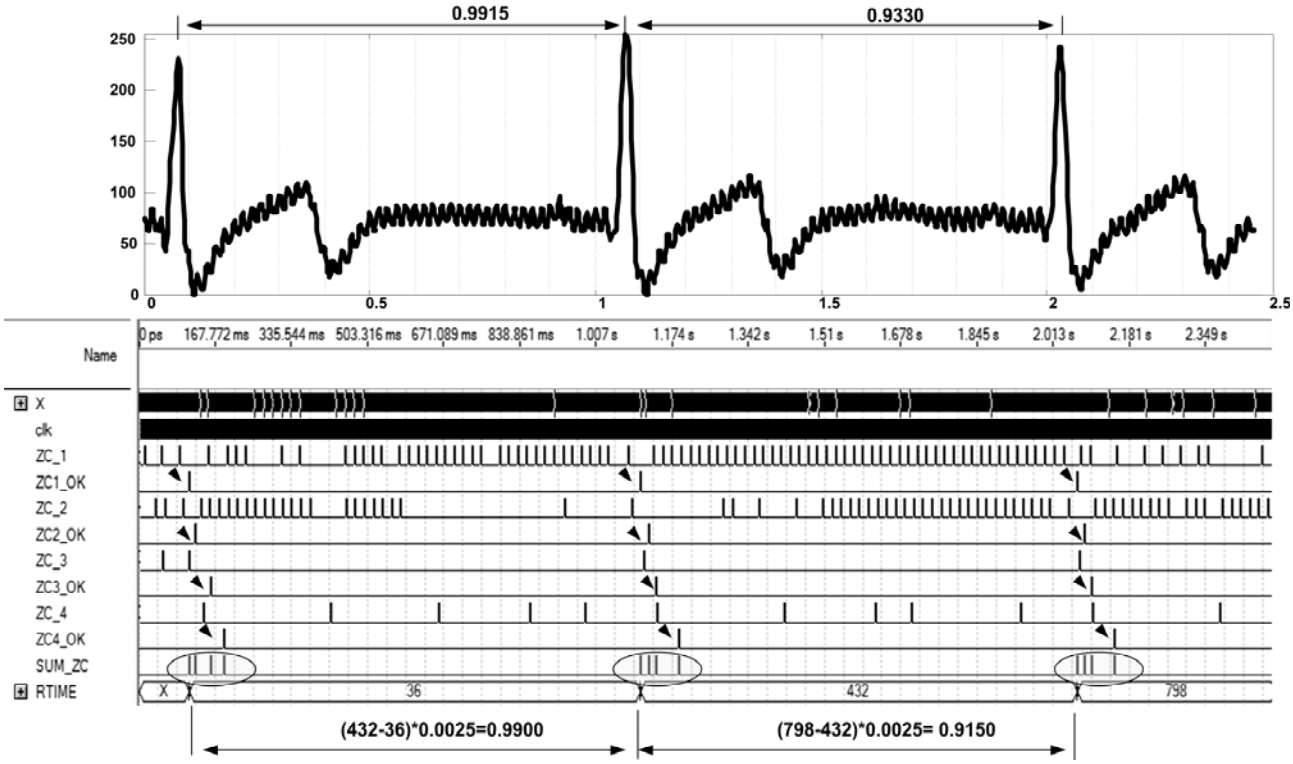

Fig.8. Illustration of simulation results. 4 decomposition levels. 400Hz sampling frequency.

F. Hardware occupancy

To asses the design efficiency in the term of silicon consumption the system and its components were considered till 4^{th} decomposition level, Table 1. One L-H cell occupied 32 LCs (Logic Cells), 4(L-H) cells 174 LCs, while overall system with 4(L-H) cells and 4(ZC&MT+DR) blocks and one MUX occupied 651 LCs. The table also gives the other combinations. These are very good optimization results even for low-capacity FPGA chips.

Table 1. Occupation of silicon resources in function of system complexity.

Configuration	Number of LCs (Logic Cells) out of 5980
L-H	32
2(L-H)	76
3(L-H)	120
4(L-H)	174
4(L-H)+ZC&MT+DR	301
4(L-H)+2(ZC&MT+DR)	385
4(L-H)+3(ZC&MT+DR)	459
4(L-H)+4(ZC&MT+DR)	533
4(L-H)+4(ZC&MT+DR)+MUX	651

5. TESTING AND RESULTS

The system was designed in Altera's Quartus 9.1 development environment. As a target chip FPGA Cyclone EP1C12Q240 was chosen. All components were implemented in Very High Speed Integrated Circuit Hardware Description (VHDL) Language, compiled, simulated, verified, converted to the symbols and then integrated into system. The simulation results were discussed in Section 4.E. After simulation and verification, the 4^{th} level system is configured in a target chip. The characteristic signals were taken out from 3^{rd}, 4^{th} or both levels (3^{rd}&4^{th}).

On-chip functionality was verified by specially designed tool set based on MC (microcontroller) board and personal computer (PC). The board communicated with PC via serial port. Microcontroller (ATMEL's AVR ATMega32) was receiving digital samples *X[n]* from PC, forwarding then them to the FPGA chip via I/O bus and simultaneously generating control and threshold signals (*clk, SL, TR1, TR2, T1_3, T2_3, T1_4, T2_4,*). The signal *SUM_ZC* from FPGA chip caused the hardware interrupts on microcontroller. In interrupt routine the contexts of registers RTIME3 and RTIME4 were captured and sent back to PC. MATLAB virtual interface managed the process between PC and MC board and performed the signal processing and statistics.

In order to determine the detection accuracy, the student volunteers recorded their ECGs by portable handheld ECG monitor/logger MD100B (Med Choice) in three lead configuration (LA-RA-LL). The noise was caused by main voltage (50Hz) and artifacts. The recorded signals were transferred to PC and stored in *.txt raw files as 8bits integers. There were totally 105,200 heart beats within 10 files/records assigned to each student. From each record the signal/array $Xi[n]$ (i=1:10) was formed and fed to the FPGA detector. The particular detection error rate *DERi* was defined as:

$$DER_i[\%] = 100(1 - \frac{NFP + NFN}{TN}) \qquad (11)$$

where are: *NFP*- number of false positives in $Xi[n]$, *NFN*- number of false negatives in $Xi[n]$ and *TN*- total number of complexes in $Xi[n]$. The averaged accuracy

$$ADER[\%] = \frac{1}{10}\sum_{i=1}^{i=10} DER_i[\%] \qquad (12)$$

was about 95.23% for using 3^{rd}&4^{th} levels in parallel, Table 2. It had been assumed that R peak appeared if the signal *ZCi_OK* occurred at *CD3* or *CD4* or at both of them.

Table 2. Partial and average accuracy.

Records Xi(n)	Beats in Xi(n)	NFP+NFN	DERi(%)
1	9200	442	95,19565
2	10400	532	94,88462
3	8300	457	94,49398
4	11200	695	93,79464
5	12600	687	94,54762
6	11800	369	96,87288
7	9600	526	94,52083
8	8700	276	96,82759
9	12000	636	94,7
10	11400	409	96,41228
Sums/ ADER[%]	105200	5029	95,22501

In the terms of hardware performances the system was evaluated against two parameters: occupancy of silicon resources (Section 4.E) and operation speed. The most demanding 4^{th} level configuration occupies around *11%* resources of Cyclone EP1C12Q240 chip. It means that at least 8 systems for QRS detection can be implemented in this device. For higher-capacity chips that number could be much higher. Maximum operation frequency for system clock is 27.23MHz that is much more than the strongest requirements regarding sampling frequency for ECG signal.

6. CONCLUSION

Wavelet Transform is a reliable method for detection QRS complexes within ECG signal. Together with distinguishing QRS complexes it performs signal filtering. Additionally it can be easily implemented in hardware, after certain optimizations in term of integer arithmetic. This paper

presents the hardware system based on these principles. The QRS detector was implemented in a single chip from FPGA family. It has accuracy of about 95% and boasts noise immunity and low cost. Because of the design optimization and capacity of FPGA, dozens detectors can be placed in same chip and directly embedded in medical instruments or wearable health care devices. After theoretically explanation, overall design methodology was presented including description of the components and the system as a whole. Also the software simulation and the verification are performed. Further, the testing procedures and preliminary results were elaborated and discussed. The same principle could be applied with other signals where required hardware implementation of wavelet transform.

ACKNOWLEDGMENT

The work presented was funded mainly by the Montenegrin Ministry of Science. The authors are thankful for the support.

REFERENCES

[1] Kohler, B.U., Hennig, C., Orglmeister, R. (2002). The principles of software QRS detection. *IEEE Enginering in Medicine and Biology,* 21, 42-57.

[2] Pavlatos, C., Dimopoulos, A., Manis, G., Papakonstantinou, G. (2005). Hardware implementation of Pan & Tompkins QRS detection algorithm. In: *Proceedings of the EMBEC'05 Conference,* November 2005, Prague, Czech Republic. Retrieved Jun 6, 2011, from http://mule.cslab.ece.ntua.gr/docs/c8.pdf.

[3] Chio, I.I., Mang, I.V., Peng, U.M. (2008). ECG QRS complex detection with programmable hardware. In: *30th Annual International IEEE EMBS Conference*, 20-24 August 2008. Vancouver, Canada, 2920-2923. Retrieved Jun 6, 2011, from http://www.bme.eee.umac.mo/PublishedPaper/IEEE%20EMBS%20Conference%202008.pdf.

[4] Shukla, A., Macchiarulo, L. (2008). Exploring and Prototyping Designs for Biomedical Applications. *Xcell Journal,* Third Quarter (65), 18-21.

[5] Mallat, S.G. (1989). A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 11 (7), 674-693.

[6] Calderbanka, A.R., Daubechiesb, I., Sweldensc, W., Yeod, B.L. (1998). Wavelet transforms that map integers to integers. *Applied and Computational Harmonic Analysis*, 5 (3), 332-369.

[7] Said, A., Pearlman, W.A. (1996). An image multiresolution representation for lossless and lossy compression. *IEEE Trans. Image Processing*, 5, 1303-1310.

[8] Walker, J.S. (1999). *A Primer on Wavelets and Scientific Applications*. Boca Raton: CRC Press.

[9] Li, C., Zheng, C., Tai, C. (1995). Detection of ECG characteristic points using wavelet transform. *IEEE Trans. Biomedical Engineering*, 42 (1), 21-28.

[10] Almeida, R., Martnez, J.P., Olmos, S., Rocha, A.P., Laguna, P. (2004). A wavelet-based ECG delineator: Evaluation on standard databases. *IEEE Trans. Biomedical Engineering*, 51 (4), 570-581.

[11] Thakor, N.V, Webster, J.G., Tompkins, W.J. (1984). Estimation of QRS complex power spectra for design of a QRS filter. *IEEE Trans. Biomedical Engineering*, 31 (11), 702-706.

[12] Altera Corp. (1995-2011). Retrieved Jun 6, 2011, from www.altera.com.

Received June 17, 2011.
Accepted September 9, 2011.