Proceedings of the 2012 IEEE 12th International Conference on Bioinformatics & Bioengineering (BIBE), Larnaca, Cyprus, 11-13 November 2012

An Appendix to Embedded Designs of QRS Detectors

Radovan Stojanović, Saša Knežević
Faculty of Electrical Engineering
University of Montenegro
Podgorica, Montenegro
stox@ac.me, sasaknezevic@live.com

Dejan Karadaglić
School of Engineering and Built Environment
Glasgow Caledonian University
Glasgow, United Kingdom
Dejan.Karadaglic@gcu.ac.uk

Abstract—**The Wavelet Transform in its discrete form has been widely applied in the field of biomedical signals. Typically, its calculation is performed off-line and calculation systems usually suffer from limited autonomy, bulkiness and obtrusiveness. A noticeable surge in industrial, research and academic interest into telemedicine and medical embedded systems, has been happening recently, where miniature, low-cost, autonomous and ultra-low-power devices play a major role. Such devices are usually based on Microcontrollers (MCs) or Field Programmable Gate Arrays (FPGAs), and in addition to other tasks they need to perform signal processing, very often in real-time. This paper presents a methodology to perform on-line QRS detector on MC's and FPGA's platforms. After the theoretical considerations on wavelets and their optimization in term of integer arithmetic, the computation architectures for both technologies are described. At the end, the presentation of obtained results during intensive tests on real signals is given. Similar approach can be applied to other signals, where the embedded implementation of wavelets can be of a benefit.**

Keywords-wavelet transform; microcontroller; FPGA; QRS;

I. INTRODUCTION

The Fourier Transform (FT) contains only globally averaged information and inherently has the drawback to obscure transient or location specific features within the signal. This limitation can be remedied to some extent by application of Short Time Fourier transform (STFT), which uses a sliding time window of fixed length to localize the analysis. Among other avialable time–frequency methods, the most promising seems to be Wavelet Transform (WT).

In contrast to FT, which is restricted to the use of a sinusoid, WT uses a variety of basic functions, known as wavelets [1]. In its discrete form (DWT), based on an orthogonal wavelet, it is particularly useful in signal compression applications. Another advantage of WT is the ability to reduce noise in signal, using successively the procedures of decomposition, thresholding, and signal reconstruction.

DWT analysis has been broadly applied to a range of biomedical signals, including ECG, EMG, EEG, PPG, clinical sounds, respiratory patterns, blood pressure trends and DNA sequences [2]. Most typically, it is calculated off-line by custom-design software or general mathematical tools, like MATLAB or similar. The input data can be prerecorded on special data buses and formats, like MIT-BIH, or taken from logger devices, like holters or memory cards. These circumstances limit the flexibility and transferability of computation systems.

Nowadays the field of telemedicine and medical embedding systems are among the fastest growing areas, and the systems here are based on miniature, low-cost, autonomous and ultra-low-power devices, where MCs and FPGAs perform majority of tasks. In addition to digitalization, data storage and communication these devices need to perform real-time signal processing in all three domains: time, frequency and time-frequency. It is not a trivial task considering the algorithmic complexities and limited performances of MCs and FPGAs in terms of arithmetic power, memory resources, consumption budget, etc. Undobtedly, the whole this field would enormously benefit in the optimization of calculation algorithms, including those DWT based, and their efficient and effective implementation on MCs and FPGAs platforms.

In this paper we show a way to optimize DWT transform for its calculation by general purpose, low-cost and low-power MCs and FPGAs. The principle can be implemented on any MC, and in our case we selected the MSP430 from Texas Instruments (TI) as a target platform. In the case of FPGA, the detector is implemented in Altera's Cyclone EP1C12Q240 chip.

In this paper, firstly we present theoretical background on wavelets and their integer arithmetic optimizations, and later we elaborate both realized computation architectures for real-time QRS complex extraction. That is followed by the presentation of the verification and real testing results. Although the approach has only been verified on ECG, it can be applied to any signal, where embedded implementation of WT is desired.

II. METHODOLOGY

A. DWT

978-1-4673-4358-9/12/$31.00 ©2012 IEEE

In practice, DWT is obtained by letting signal X(n) through a Low-Pass (L_d) and a High-Pass (H_d) filters successively according to the Mallat's decomposition scheme shown in Fig. 1 [3]. For each decomposition level *i*, the L_d and H_d filters are followed by downsampling operator $\downarrow 2$, which is, in fact, the downsampling by a factor of 2. The coefficients $CA_i(n)$ (approximations) and $CD_i(n)$ (details) are outputs from level *i*. The reconstruction consists of upsampling by a factor of 2 and filtering through filters L_r and H_r. The coefficients for L_d, H_d, L_r, H_r filters can be choosen from the simplest, such as Haar, over Daubechies up to those such as Quadratic Spline. They have different vector lengths and, usually, floating point nature.

The Haar wavelet is considered to be the simplest one where L_d=[1/sqrt(2),1/sqrt(2)], H_d=[1/sqrt(2),-1/sqrt(2)], L_r=[sqrt(2)/2,sqrt(2)/2] and H_r=[-sqrt(2)/2,sqrt(2)/2] filters are of two elements wide with real coefficients. Although very simple in principle, Haar Transform (HT) is still complicated for practical implementation on MCs.

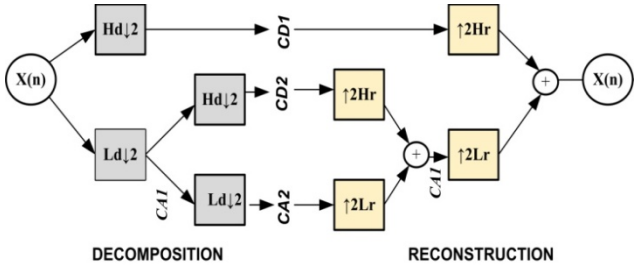


Figure 1. Wavelet decomposition and reconstruction scheme.

However, like any other WT, HT can be generalized to an integer-to-integer version. One of proposed techniques [4] is in form of S Transform (ST), which Forward (FST) and Reverse (RST) versions are defined as:

$$CA_1[n] = \left\lfloor \frac{1}{2} X[2n] + \frac{1}{2} X[2n+1] \right\rfloor, \quad (1)$$

$$CD_1[n] = X[2n] - X[2n+1], \quad (2)$$

$$X[2n] = CA_1[n] + \left\lfloor \frac{CD_1[n] + 1}{2} \right\rfloor, \quad (3)$$

$$X[2n+1] = CA_1[n] - \left\lfloor \frac{CD_1[n]}{2} \right\rfloor, \quad (4)$$

where $\lfloor\ \rfloor$ denotes rounding operator. The above equations employ only basic arithmetic units like adder/subtractor and shifter, because $\lfloor x/2 \rfloor \cong (x >> 1)$.

B. WT in QRS Detection

As any other WT, FST is capable of extracting the QRS-complex within ECG signal by using Mallat's decomposition scheme. The $CD_i(n)$ coefficients across the scales show that the peak of the QRS complex corresponds to the zero crossing (ZC) between two modulus maxima within $CD_i(n)$ [2], [5]. Fig. 2 illustrates the decomposition of real discrete ECG signal $X(n)$ up to the 4th level, ($CD_1(n)$, $CD_2(n)$, $CD_3(n)$ and $CD_4(n)$) by using FST. For each decomposition level, the QRS complex produces two modulus maxima (min and max) with opposite signs, with a ZC between.

Presented method for QRS detection is very robust and allows direct application on the raw ECG data because the frequency domain filtering is performed implicitly by computing the coefficients. For example, the original signal becomes clear at 4th decomposition level, as shown in Fig. 2. The built-in filtering is an important feature of wavelet decomposition applied in QRS detection.

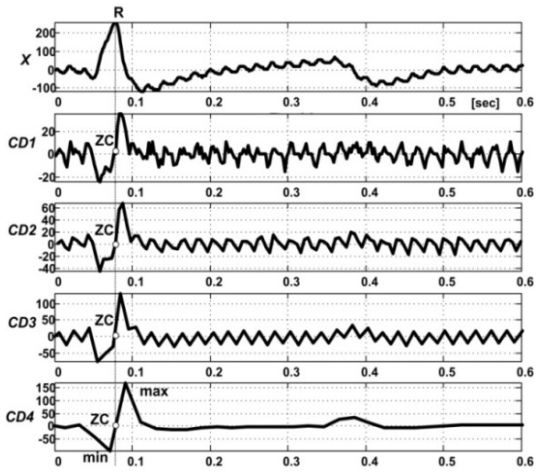


Figure 2. QRS detection using wavelet decomposition. Signal X[n] is sampled with 800 Hz. $CD_i(n)$ are the details after i[th] decomposition level.

In practice the selection of the most suitable decomposition level/levels is a challenging task. 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[5]. Another complication is the acquisition of certain thresholds for finding the modulus maxima, because the values of thresholds differ, usually, from one level to another.

All mentioned restrictions and complications restict the method to off-line use and put heavy demand on the computing resources.

III. EMBEDDED IMPLEMENTATION

In embedded applications the signal processing algorithms are usually implemented in MCs or FPGAs. Many of the authors favour one of the ways on the expense of the other. The most realistic and balanced approach is to confirm that each of both has own advantages and disadvantages. As example, FPGA consists of reconfigurable logic, I/O and interconnections blocks and differs from MCs and DSP processors, which are Von Neumann types of machine. In addition to the parallelism, the advantages of later 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). Although much faster, the FPGAs are also quite expensive, bulky and more difficult for rapid prototyping and implementation in small series. In this project we decided the above optimized wavelet algorithms to implement in both technologies.

A. MC implementation of QRS detector

As MC platform, the MSP430 from TI [6] is selected. It belongs to a family of ultra-low power microcontrollers optimized for using in portable battery powered devices like medical ones. As a target chip, the MSP430F169 has been selected, having 16-bit RISC CPU, 16-bit registers, two 16-bits timers, fast 12-bit A/D converter with 8 external input channels, dual 12-bit D/A converter, USART, I2C, DMA, and 48 I /O pins, etc.

MSP430F169 on-chip architecture for real-time QRS detection is shown in Fig. 3. The analog ECG signal is fed to the channel of ADC. After digitalization and software processing the real-time output signals are obtained in different forms. Analog form of details $CD_N(n)$ and $CD_{N-1}(n)$, N=4, generated by DAC through the pins P6.6 and P6.7. Pulse form, where each pulse indicates detected QRS complex, pin P1.0. ASCII form through the USART's TX pin, where the each record represents the actual RR interval in ms.

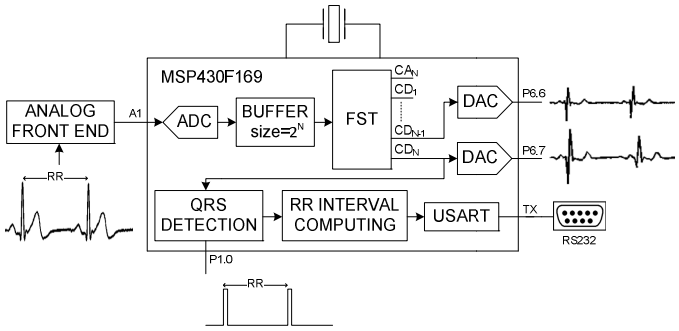


Figure 3. MSP430F169 architecture for QRS detection.

The real-time implementation of forward and reverse wavelet transform in software is done through the FST and RST, because of their simplicity and fast calculation. Before its processing, the signal is digitalized by 12bits A/D converter. The sampling frequency is set to be 800Hz. The A/D conversion is performed in an interrupt routine. Between the interrupts the MSP430 MCU switches in sleep mode.

After A/D conversion, each sample is stored in a circular buffer of lenght 2^N, where N presents the number of decomposition levels. When the buffer is full enough the FST is calculated, while the buffer continues to accept new samples. Then, the detail coefficients $CD_i(n)$ are examined on ZC, which is situated between modulus maxima of opposite signs. As it is mentioned before, the most suitable levels for QRS detection are 3^{th} and 4^{th}, coresponding to $CD_3(n)$ and $CD_4(n)$. In this approach the $CD_4(n)$ is prefered. Negative and positive modulus maxima are determined by adaptive thresholding technique. Five successive vectors of 50 CD_4 coefficients are examined. For each of them the maximum $M_jmax=max(CD_4(1..50))$ and minimum $M_jmin=min(CD_4(1..50))$ values are calculated, $j=1...5$. Then the negative (T_1) and positive (T_2) thresholds are defined as:

$$T_1 = \frac{1}{4}\left(\frac{1}{5}\Sigma_{j=1}^{5} M_j min\right), \quad T_2 = \frac{1}{4}\left(\frac{1}{5}\Sigma_{j=1}^{5} M_j max\right) \qquad (5)$$

The process repeats with four old vectors and one new vector. ZC is detected by finding the coefficients associated to the condition $CD_4(n-1)<0$ and $CD_4(n)> 0$.

B. FPGA implementation of QRS detector

Simplified diagram of proposed FPGA system for QRS complex detection is given in Fig. 4. Wavelet decomposition is realized with pipelined L-H cells [7]. 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 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.

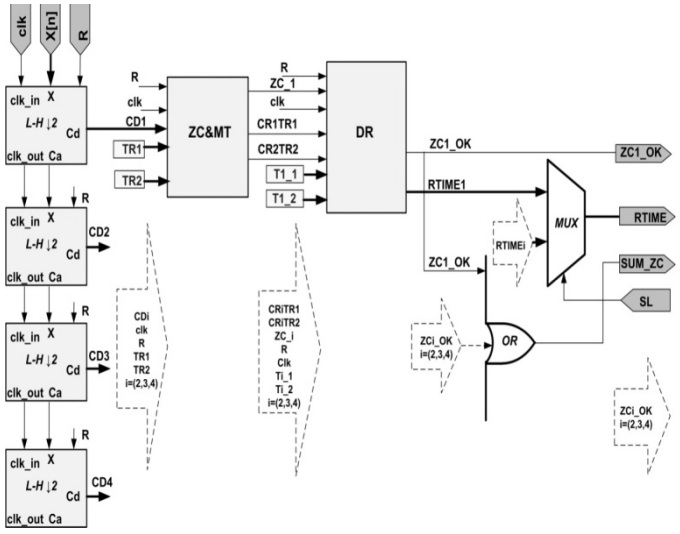


Figure 4. FPGA architecture for QRS detector.

L-H cell from Fig. 5 presents the core unit in decomposition scheme and in fact is a digital expression of equations (1) and (2). The delay line is implemented in the form of 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 ($\downarrow 2$) 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 fed to the input X of the further cell and so on. The calculation time per Ca and Cd is 2clk. The signal clk_out is led to clk_in of next cell. As shown, it is a way to calculate equations (1) and (2) without multipliers.

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.

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). Differently from MCs case, here TR2=abs(TR1)=k*Max(X(n)), where k varies in the range as 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 sucessive samples such that CD(n-1)<0 and CD(n)>0.

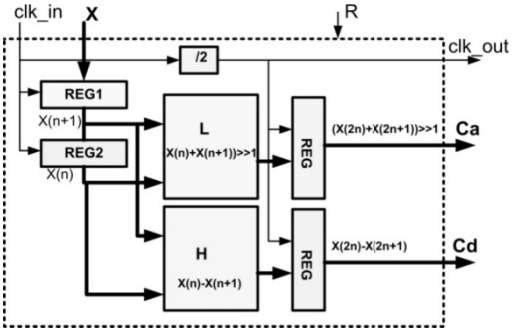


Figure 5. Architecture of L-H cell .

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 a 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* will be 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 will be 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 because this time corresponds to the position of *R* peak. The storing of current time when R peak happened (*RTIMEi*) 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. A possible complication may arise from the 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.

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.

In Fig. 7 the simulated scenario assumes all four decomposition levels working in parallel. Test signal of 1024 samples is acquired by sampling frequency of 400Hz (2.5ms), in total duration of 2.5s. The thresholds *TR1* and *TR2* are adjusted to -22 and 22. The R peak is detected at all decomposition levels. ZC1_OK, ZC2_OK, ZC3_OK and ZC4_OK are summed in *SUM_ZC*. Because of the accuracy reasons the *RTIME* was captured from 1^{st} level, *RTIME =RTIME1*. Previously the same signal was passed through identical MATLAB detector, measuring R-R intervals. As shown, the good matching is obtained, 0.9915s against 0.9900s and 0.9330s against 0.9150s respectively.

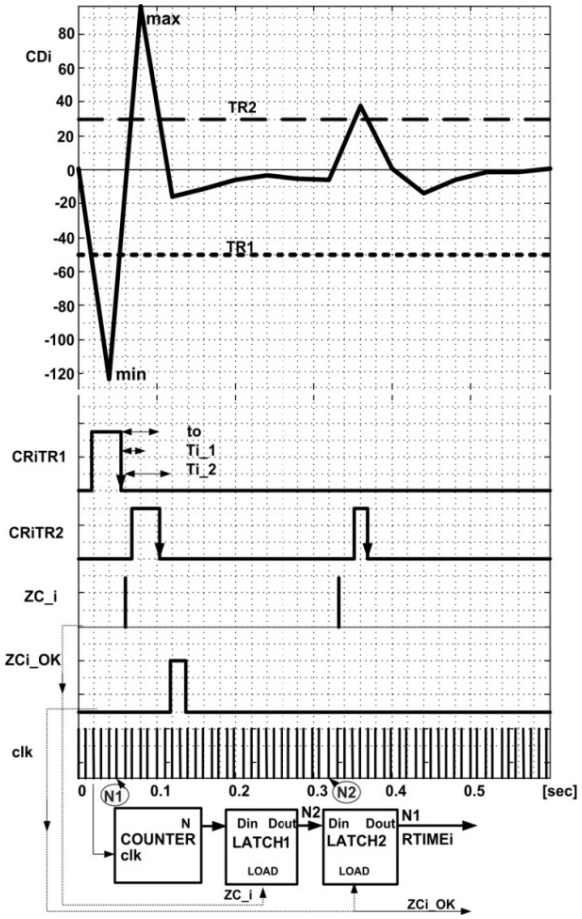


Figure.6. Illustration of the algorithm for finding modulus maxima and zero crossing in FPGA. Bottom is a decision making technique.

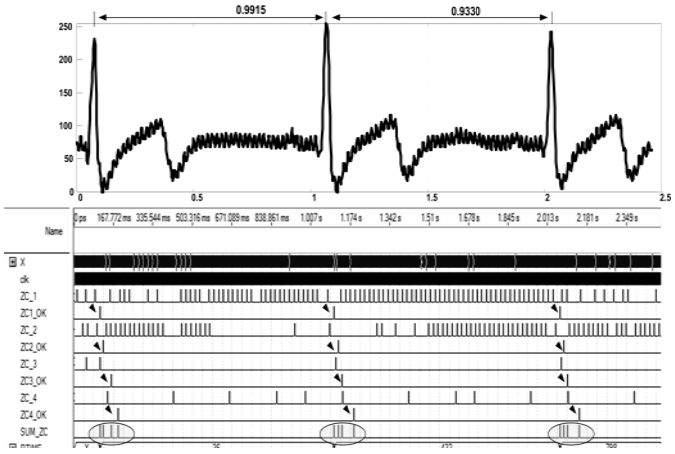


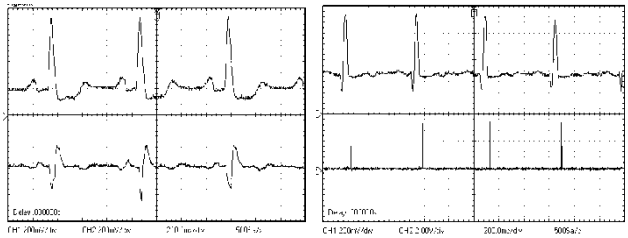
Figure 7. Simulation results. 4 decomposition levels, fs=400Hz.

IV. RESULTS

In the case of MCs, on-chip verification is implemented by specially designed tool-set based on ELVIS II$^+$ NI Platform [8], microcontroller MSP430 board (Olimex MSP430-P169) and personal computer (PC). MC code is developed in IAR Embedded Workbench Compiler and uploaded to chip. The ECG signals are taken from MIT-BIH database and emulated in analog form by ELVIS platform via LabView virtual instrument.

MSP430 chip accepts the emulated signals, performs FST and RST and QRS detection in real-time. It returns the different signals: analog form of detail coefficients $CD_4(n)$ and $CD_3(n)$, the digital signal with pulses in places of QRS complexes and serial RS232 data in form of ASCII strings with RR intervals in ms. The analog and digital output signals are observed by oscilloscope, while ASCII strings are received by serial terminal emulator.

The original ECG signal and corresponding detail coefficients $CD_4(n)$ taken from P6.7 are shown in Fig. 8 a), while Fig. 9 b) presents an ECG signal and impulses from pin P1.0 which correspond to the QRS complexes. In order to determine the QRS detection accuracy the 11,094 heart beats within 5 characteristic files are observed (MIT-BIH Records 101, 103, 202, 230, 234). The average detection rate was about 99.1%.



a) b)

Figure 8. a) Analog forms of ECG signal and CD4(n) coefficients calculated by MSP430F169 MC on-line. b) QRS complexes and corresponding impulses obtained as a result of on-line wavelet based QRS detector.

In the case of FPGA, the system was designed in Altera's Quartus 9.1 tool. 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. 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 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 hardware interrupts on microcontroller. In the 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.

To assess the design efficiency in the term of silicon consumption the system and its components were considered up to 4^{th} decomposition level. 1 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. These are very good optimization results even for a low-capacity FPGA chip. In order to determine the detection accuracy, the same data set as in case of MC is considered. The average detection rate was 93.23% for using 3^{rd}&4^{th} levels in parallel.

V. CONCLUSION

WT is a reliable method for processing of physiological signals. After certain optimizations in terms of integer arithmetics, it can be easily implemented on general purpose MCs and FPGAs. An approach for the implemention of QRS detection in both technologies using WT is elaborated and discussed. The systems work in real-time, above required detection accuracy and denoising performances. The same principle can also be applied in the case of other signals .

REFERENCES

[1] A. Graps, "An Introduction to Wavelets," Computing in Science & Engineering, vol. 2, no. 2, IEEE press, 1995. pp. 50-61.

[2] P. S. Addison, Wavelet transforms and the ECG: a review, Physiological Measurements, vol. 26, 2005, pp. 155–199.

[3] S. G. Mallat, A theory for multiresolution signal decomposition: the wavelet representation. IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 11, no. 7, 1989, pp. 674-693.

[4] R. Calderbank, I. Daubechies, W. Sweldens and B. L. Yeod, Wavelet Transforms That Map Integers to Integers. Applied and Computational Harmonic Analysis, vol. 5, no. 3, 1998, pp. 332-369.

[5] C. Li, C. Zheng, and C. Tai, Detection of ECG characteristic points using wavelet transform. IEEE Transactions on Biomedical Engineering, vol. 42, no. 1, 1995, pp. 21-28.

[6] Internet: http://www.ti.com, Jan., 2012.

[7] R. Stojanović, D. Karadaglić, M. Mirković, D. Milošević, "A FPGA system for QRS complex detection based on Integer Wavelet Transform," Measurement Science Review (ISSN:1335-8871), vol. 11, issue 4, June 2011, pp. 131-138.

[8] Internet: http://www.ni.com/pdf/manuals/372590b.pdf, Jan., 2012.

[9] Internet: http://www.altera.com, Jan., 2012.