Jurnal Teknologi

Full Paper

EFFICIENT QRS COMPLEX DETECTION ALGORITHM IMPLEMENTATION ON SOC-BASED EMBEDDED SYSTEM

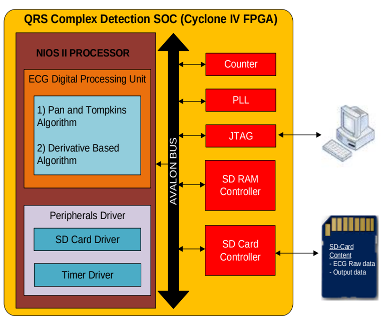
Muhammad Amin Hashim[a], Yuan Wen Hau[b]*, Rabia Bakhteri[c]

[a]Faculty of Electrical Engineering, Universiti Teknologi Malaysia, 81310 UTM Johor Bahru, Johor, Malaysia
[b]Faculty of Biosciences and Medical Engineering, Universiti Teknologi Malaysia, 81310 UTM Johor Bahru, Johor, Malaysia
[c]Sightline Innovation Inc., Winnipeg, Manitoba, Canada

Article history
Received
31 October 2015
Received in revised form
1 March 2016
Accepted
28 March 2016

*Corresponding author
hauyuanwen@utm.my

Graphical abstract



Abstract

This paper studies two different Electrocardiography (ECG) preprocessing algorithms, namely Pan and Tompkins (PT) and Derivative Based (DB) algorithm, which is crucial of QRS complex detection in cardiovascular disease detection. Both algorithms are compared in terms of QRS detection accuracy and computation timing performance, with implementation on System-on-Chip (SoC) based embedded system that prototype on Altera DE2-115 Field Programmable Gate Array (FPGA) platform as embedded software. Both algorithms are tested with 30 minutes ECG data from each of 48 different patient records obtain from MIT-BIH arrhythmia database. Results show that PT algorithm achieve 98.15% accuracy with 56.33 seconds computation while DB algorithm achieve 96.74% with only 22.14 seconds processing time. Based on the study, an optimized PT algorithm with improvement on Moving Windows Integrator (MWI) has been proposed to accelerate its computation. Result shows that the proposed optimized Moving Windows Integrator algorithm achieves 9.5 times speed up than original MWI while retaining its QRS detection accuracy.

Keywords: Derivative based algorithm, electrocardiography, pan and tompkins algorithm, moving windows integrator, system-on-chip

Abstrak

Artikel ini mengkaji dua algoritma pemprosesan Electrokardiografi, iaitu Pan and Tompkin (PT) dan Derivative Based (DB) algoritma, yang amat penting dalam mengesan kompleks QRS untuk mengenal pasti penyakit kardiovaskular. Kedua-dua algoritma ini dibanding dari segi kejituan pengesanan kompleks QRS dan prestasi masa pemprosessan dengan implimentasi ke dalam system-dalam-cip system terbenam yang telah diprototaipkan ke dalam platfom Altera DE2-115 Field Programmable Gate Array (FPGA). Kedua-dua algoritma telah diuji dengan 30 minit ECG rekod daripada 48 pesakit berbeza yang telah diperolehi daripada databes MIT-BIH arritmia. Keputusan menunjukkan algoritma PT mencapai 98.15% ketepatan dengan masa pemprosessan 56.33 saat manakala algoritma DB pula mencapai ketepatan 96.74% dengan hanya 22.14 saat waktu pemprosessan. Daripada kajian ini, satu pengoptimuman PT algoritma telah dicadangkan untuk meningkatkan prestasi Moving Windows Integrator (MWI). Keputusan menunjukkan Windows Moving Intergration yang telah dioptimumkan mencapai kelajuan sehingga 9.5 kali ganda berbanding MWI asal dengan pengekalan kejituan algoritma PT.

Kata kunci: Algoritma berasaskan terbitan, electrokardiografi, pan dan tompkins algoritma, tetingkap bergerak penyepadu, sistem-dalam-cip

1.0 INTRODUCTION

Heart is the most important vital organ in human body. It ensures oxygenated blood and nutrient can be supplied to entire human body. Heart failure has been one of the top diseases that cause human death. According to World Health Organization (WHO), it is estimated that 17.3 millions of people dies each year caused by heart failure. This represents one third of entire global human death in worldwide.
Electrocardiography (ECG) is a record that contains the information of heart's electrical conduction activity. Heart muscles contract and relax due to the electricity originated by 2 nodes on the heart, so-called Sinoatrial (SA) node and Atrioventricular (AV) node. This contraction cycle produces blood pressure so that blood can flow throughout the human body. Thus ECG record of heart electrical conductivity is commonly used to detect early sign of cardiovascular diseases (CVDs), such as Premature Ventricular Heartbeat, Premature Atrial Heartbeat, Left Bundle Branch Block, Right Bundle Branch Block, Heart block, ischemia and many more.

Some of these abnormal heartbeats indicate a dangerous symptom and can cause sudden death if early treatment is not provided promptly [1]. However many of these abnormal symptoms does not appear continuously but only occur at random time, thus a portable ECG device with accurate classification is desired to enable frequent monitoring of patient heart condition [2-4]. In addition to that, for accurate heart disease classification, QRS complex detection algorithm is a must and has commonly been used by many researchers [5-7]. Thus choosing an efficient and optimized algorithm for QRS complex detection is crucial especially when the ECG device is implemented on embedded system which has hardware constraint.

This paper studies two QRS detection algorithms, so-called Pan and Tompkins (PT) algorithm and Derivative Based (DB) algorithm. They are compared in terms of QRS detection accuracy and computation timing performance which target to System-on-Chip (SoC) based embedded system architecture. This paper also proposes an optimization of PT algorithm by improving the Moving Windows Integration (MWI) to accelerate its computation for effective embedded system implementation while retaining its QRS detection accuracy. All the timing performance results are compared by implementing both algorithm on Altera DE2-115 FPGA platform to form a low cost, power effective, portable and lightweight ECG device prototype.

This paper is organized as below. Section 1 has introduced the statistics of CVDs and its relation with QRS detection of ECG signal. Section 2 depicts the ECG signal and related work of QRS detection. Section 3 explains the detail of both algorithms including the optimized PT algorithm. Section 4 describes the implementation of ECG processing system on SoC-based embedded system architecture and its FPGA prototyping platform. The results of QRS detection accuracy and timing performance is presented in Section 5. Lastly, conclusion and recommendation of future work is presented in Section 6.

2.0 ECG SIGNAL AND RELATED WORKS

An ECG shows the heart's electrical conductivity. There are few events of interest of an ECG signal such as P wave, QRS complex and T wave. In brief, P wave represents atrial depolarization (contraction), QRS complex represents ventricular depolarization, and T wave represents ventricular repolarization. Figure 1 shows an example of Lead 2 ECG.

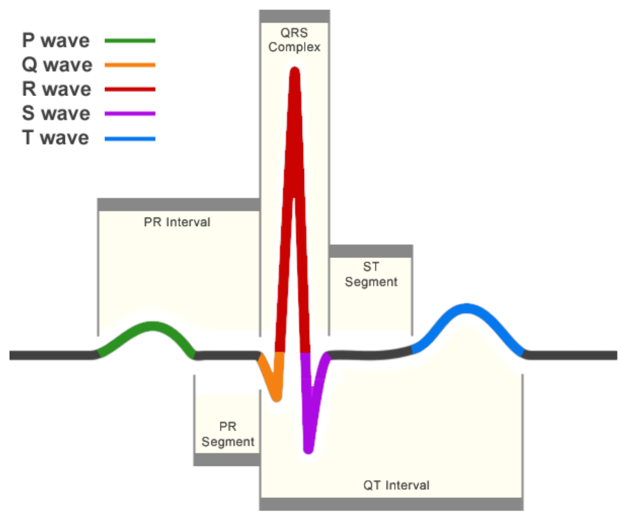


Figure 1 P, Q, R, S, T waves in lead II Electrocardiogram

There are many QRS complex detection algorithms have been proposed by researchers worldwide such as Derivative based algorithm [8], Pan and Tompkins algorithm, Difference Operation Method (DOM) [9], ANN-based QRS detection [10], Hidden Markov model QRS detection [11], Wavelet-transform based QRS detection, Genetic algorithm [12] and many more.

Derivative based algorithm was proposed by Balda *et al.* [8]. This algorithm applies two orders of derivative equation to the ECG signal before smoothing to 3 points signal. This technique is very efficient in term of computation though it is not optimized for noisy signal. Pan and Tompkins algorithm [13] is based on filtering, derivative and squaring operations. It is a breakthrough during that time due to its ability to detect QRS complex with high accuracy.

Wavelets based QRS detection decomposes the ECG signal to different frequency band before applying threshold to obtained characteristic points.

In neural networks QRS detection, the system predicts the current signal before it can determine the appropriate filter to remove the noise and decision making [10]. Genetic algorithm try to find the best polynomial coefficient and then the best parameter for classification [12].

Based on the aforementioned algorithms, we have selected Pan and Tompkins and Derivative Based algorithm due to their low complexity and thus suitable for embedded system implementation.

3.0 QRS DETECTION ALGORITHM

This section describes the detail of Pan and Tompkins (PT) and Derivative based (DB) algorithms which aim to be implemented on Altera DE2-115 FPGA board as SoC-based embedded system for ECG processing. The proposed optimization of PT algorithm for effective Moving Window Integration computation is also presented in this section.

3.1 Original Pan and Tompkins Algorithm with Adaptive Threshold

Pan and Tompkins (PT) algorithm is one of the most common algorithm for QRS detection, which requires the ECG data sampling frequency at 200Hz [13-16]. Figure 2 shows the data flow diagram of PT algorithm which consists of band pass filter, differentiation, squaring, moving window integration, and adaptive threshold detection.

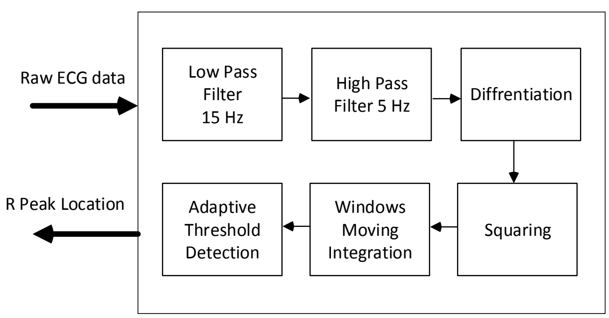


Figure 2 Pan and Tompkins Algorithm

Firstly, the raw ECG data is filtered by a 15 Hz low pass filter follow by a 5 Hz high pass filter as shown in Equation 1 and Equation 2, respectively. These two filters form a band pass filter to remove unnecessary low and high frequency noise signal such as muscle noise and baseline wanders [17].

$$Y(N) = 2Y(N) - Y(N\text{-}2) + X(N) - 2X(N\text{-}6) + X(N - 12) \quad (1)$$

$$Y(N) = Y(N-1) + X(N) - X(N - 32) \quad (2)$$

Next, it is differentiated as shown in equation 3 to highlight the slope information of QRS complex which usually contains the steepest slope compared to the other peaks.

$$Y\,(N) = (1/8)\ (2X(N) + X(N-1) - X(N-3) - 2X(N-4)) \quad (3)$$

The differentiated output is then squared to maximize the amplitude difference of QRS complex with other peaks as shown in equation 4. It is also aims to convert all the signal amplitude values become positive values.

$$Y(N) = [X(N)]^2 \quad (4)$$

After that, the squared output signal passes through a Moving Windows Integrator (MWI) to smooth the signal by removing the fluctuations in signal peaks. It is done by summing a number of data points and calculate the average value of each window. The number of data points in each window is 32 and the complete equation is shown in equation (5).

$$Y(T) = (1/32)\ [Y(N) + Y(N\text{-}1)+\ldots+Y(N\text{-}32)] \quad (5)$$

Lastly, the QRS complex can be determined by applying a threshold to the output of moving window integrator. In this work, an adaptive voltage threshold is applied based on technique proposed in [18]. Compare to the fixed threshold, adaptive threshold does not need to be manually set prior of the ECG processing, but is automatically set after processing the first few seconds of early ECG recording which function as the parameter training.

Initially, every peak is considered as either Noise Peak or Signal Peak. An initial value of Signal Threshold and Noise Threshold is then generated for QRS detection accordingly. Please note that these threshold values are not fixed and will keep changing and adapt from time to time along the ECG data processing. Using other words, whenever the ECG data is changed along the ECG record, the threshold values will be updated automatically accordingly. In this work, the first 2 seconds of an ECG data were used for parameter training to compute initial parameter value as shown in equations (6) to (9).

$$\text{Signal Peak} = \text{MAX (training set)} \quad (6)$$
$$\text{Signal Threshold} = \text{Signal Peak}/3 \quad (7)$$
$$\text{Noise Peak} = \text{MEAN (training set)} \quad (8)$$
$$\text{Noise Threshold} = \text{Noise Peak}/2 \quad (9)$$

After the first 2 seconds of parameter training, the parameter will keep changing along the ECG data processing to set the adaptive threshold according to equation (10) to (12). If a QRS complex is detected, which the value is larger than Signal Threshold, then the algorithm will skip 0.2 second, this is to prevent double QRS detection at the nearest location which would be physically impossible.

$$\text{Signal Peak} = 0.125\ (\text{Current Peak}) + 0.875\ \text{Signal Peak} \quad (10)$$

$$\text{Signal Threshold} = \text{Noise Peak} + 0.25(\text{Signal Peak} - \text{Noise Peak}) \quad (11)$$

$$\text{Noise Threshold} = \text{Signal Threshold}/2 \quad (12)$$

On the other hand, if the detected beat is fall within the range of Signal Threshold and Noise Threshold, the system needs to be aware of this peak. Hence the Noise Peak, Noise Threshold and Signal Threshold parameters need to be adapted and change according to equation (13) to (15).

$$\text{Noise Peak} = 0.125\ (\text{Current Peak}) + 0.875\ (\text{Noise Peak}) \quad (13)$$

$$\text{Signal Threshold} = \text{Noise Peak} + 0.25(\text{Signal Peak} - \text{Noise Peak}) \quad (14)$$

$$\text{Nosie Threshold} = \text{Signal Threshold}/2 \quad (15)$$

3.2 Derivative Based Algorithm with Adaptive Threshold

Derivative based (DB) algorithm is another popular algorithm to compute the location of QRS complex in a raw ECG data. The algorithm is proposed by [8, 17, 19, 20] and has been commonly adapted by researchers. Figure 3 shows the data flow diagram of DB algorithm which consists of first and second differentiation, weighting and combining, smoothing, and adaptive threshold detection.

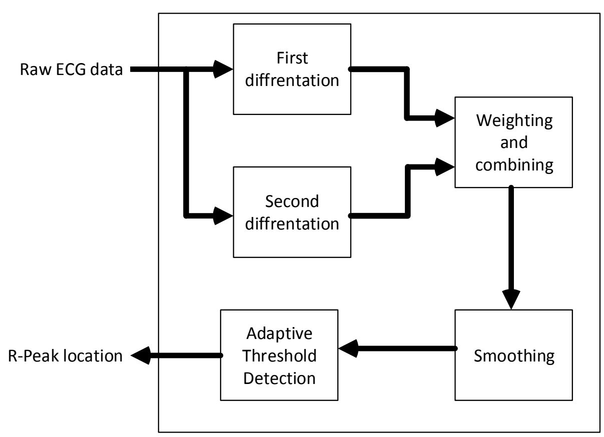


Figure 3 Derivative Based Algorithm

Firstly, the raw ECG data passes through the first and second order differentiation as given in equations (16) and (17), respectively.

$$Y_1[N] = |X(N) - X(N\text{-}2)| \quad (16)$$

$$Y_2(N) = |X(N) - 2X(N\text{-}2) + X(N\text{-}4)| \quad (17)$$

The outputs of both differentiation are then combined with different bias weight proposed by [15] which aims to maximize the difference of QRS complex with other peaks.

$$Y_3[N] = 1.3Y_1[N] + 1.1\ Y_2[N] \quad (18)$$

To prevent false beat detection which cause by noise signal, the data is smoothen according to equation (19).

$$Y4(N) = (1/8)[Y3(N\text{-}0) + \ \ldots\ Y3(N\text{-}7)] \quad (19)$$

The same adaptive threshold technique is then applied according to equation (6) to (15) to detect the location of QRS complex.

3.3 Optimized Pan and Tompkins algorithm with Adaptive Threshold

This section proposes an optimized implementation of Pan and Tompkins algorithm by improving Moving Average Integrator operation to accelerate its computation. Firstly, a temporary variable is declared to store the sum of N=32 data points of the first window as shown in equation (20).

$$V = X(0)) + X(1)) + \ldots + X(32) \quad (20)$$

The first output data point is then computed according to equation (21).

$$Y[0] = (1/32)\ (V) \quad (21)$$

The temporary variable is then updated by subtracting the first ECG data point of the previous window and adding the next consecutive ECG data point for the next consecutive moving window as shown in equation (22).

$$V = V - X(N\text{+}1) + X(N\text{-}32\text{-}1) \quad (22)$$

The updated temporary variable is then used to compute next data point as shown in equation (23).

$$Y[N] = (1/32)(V) \quad (23)$$

By introducing this optimized PT algorithm of modifying MWI by utilizing a temporary variable, it helps to store redundant ECG intermediate data points of each window to avoid repetitive computation. By only subtracting and addition of one ECG data point, respectively, instead of full window size calculation, this greatly reduced the computation overhead in each window iteration along the ECG record. This optimization is particularly crucial if a long ECG data records was processed.

Figure 4 shows the C code fragment of the original Moving Windows Integrator based on Equation (5).

```
for (i=0; i <= 360000; i++)
{         for (n=0; n < 32; n++)
          {
                    Y[i] = X[n+i];
          }
          Y[i] = Y[i]/32;
}
```

Figure 4 C code fragment of Original Moving Windows Integrator Algorithm

Figure 5 shows the C code fragment of Optimized Moving Window Algorithm based on equation (20) to (23).

```
    V = 0;          //Temporary variable initialization

//Pre-computation of the first window of 32 data points
    for (i=0; i<32; i++)
    {
        V = V + X(i);
    }
    Y(0) = V/32;

    //Computation of the next consecutive windows
    for (i = 1; i<360000; i++)
    {
       V = V - X(i-1)+ X(i+31);
       Y(i) = V/32;
    }
```

Figure 5 C code fragment of Optimized Moving Windows Integrator Algorithm

From C code shown in Figures 4 and 5, it can be observed that the original Moving Windows Integrator executes total of 33 operations for each window of 30 minutes or 360,000 data points. However, in the optimized Moving Windows Integrator algorithm, we only need to pre-compute the intermediate data point once and store it into a temporary variable, then it could be reused for next data point computation. As a result, this optimization leave us with only 3 operations for each 360000 data points and eliminate unnecessary 30 redundant operations for each window.

4.0 DESIGN AND IMPLEMENTATION OF ECG QRS DETECTION ALGORITHM

Figure 6 shows the System-on-Chip (SoC) based embedded system architecture of ECG processing implemented on Altera DE2-115 FPGA board.

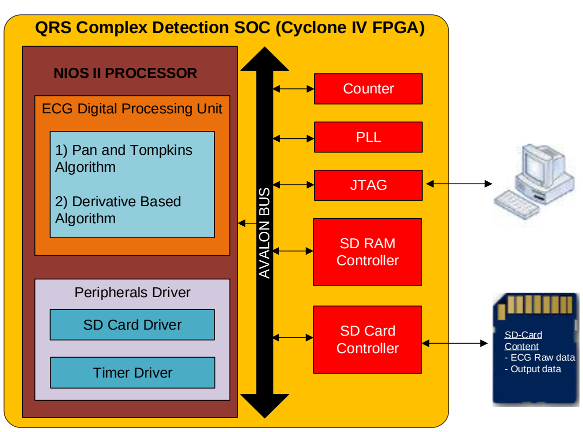


Figure 6 System on Chip Implemented Architecture

The SoC architecture consists of few standard peripherals which is designed using Altera Quartus II and Qsys EDA software. The NIOS II processor with full features configuration acts as the system master controller and executes the ECG QRS complex detection algorithms implemented as embedded software. It also executes the peripheral device drivers such as SD-Card and SDRAM external memory module.

In terms of memory module, due to ECG processing algorithm requires a large temporary data storage to store the intermediate result, it will be insufficient if the system architecture only rely on on-chip memory. As a result, a SDRAM controller is added so that Nios II Processor can utilize the 124 MB off-chip SDRAM memory on DE2-115 FPGA board as program memory and data memory. In addition to that, a SD-Card controller is added to interface with external SD-Card which used to store the input offline ECG data obtained from MIT-BIH database. After execution of each ECG QRS detection algorithm, the detected QRS complex as output data is stored in the SD-Card for post analysis, such as output graph plotting using Matlab software. Besides, the location of detected QRS complex of each algorithm is also compared with actual QRS complex location provided from MIT-BIH for detection accuracy analysis.

The system running frequency is set to 120 MHz by adding and configuring a Phase Lock Loop (PLL) to change the input clock frequency of 50 MHz generated from the crystal on the FPGA board. There is also a 32-bit high resolution timer used to count the clock cycle during algorithm execution for computation timing performance measurement.

5.0 RESULTS AND DISCUSSION

This section discusses the analysis result of Pan and Tompkins, Derivative Based Algorithm, as well as optimized Pan and Tompkins algorithms, in terms of QRS detection accuracy and computation timing performance. The system is designed based on SoC-based embedded system architecture and implemented on Altera DE2-115 FPGA board which utilize Cyclone IV FPGA chip. The test data is offline ECG data obtained from MIT-BIH database which consists of total 48 patient's records.

5.1 QRS Detection Accuracy and Timing Performance of Original PT algorithm with Adaptive Threshold

Figure **7** shows the output graph plotted using Matlab software of each intermediate process of Pan and Tompkins algorithm, from ECG raw data to detected QRS complex after moving window integration. The result of detected QRS complex of PT algorithm is also compared with actual QRS location annotated by MIT-BIH database for QRS detection accuracy analysis as shown in Table 1.

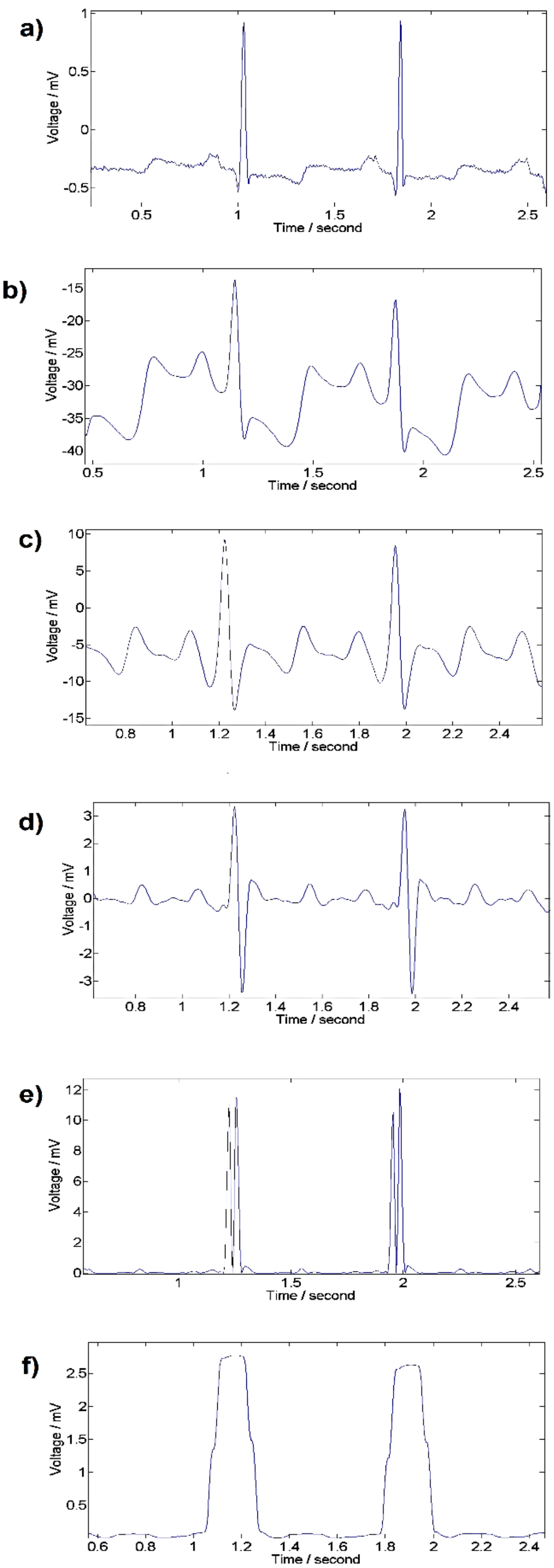


Figure 7 a) ECG raw data, b) 15 Hz low pass filter output, c) 5Hz High Pass Filter output, d) Differentiation output, e) Squaring output, and f) Moving Window Integrator output

Table 1 QRS detection accuracy of PT algorithm

Patient Record	Total Actual Number Peaks	Total Peaks Detected	Accuracy (%)
#100	2273	2273	100.00
#101	1865	1868	99.83
#102	2187	2188	99.95
#103	2084	2079	99.76
#104	2230	2261	98.63
#105	2572	2684	95.83
#106	2027	2015	99.41
#107	2137	2131	99.72
#108	1763	1980	89.04
#109	2532	2527	99.80
#111	2124	2129	99.77
#112	2539	2543	99.84
#113	1795	1795	100.00
#114	1879	1878	99.95
#115	1953	1953	100.00
#116	2412	2384	98.84
#117	1535	1536	99.93
#118	2275	2279	99.82
#119	1987	1988	99.95
#121	1863	1862	99.95
#122	2476	2476	100.00
#123	1518	1520	99.87
#124	1619	1615	99.75
#200	2601	2603	99.92
#201	1963	1984	98.94
#202	2136	2130	99.72
#203	2982	2968	99.53
#205	2656	2654	99.92
#207	1862	2255	82.57
#208	2956	2951	99.83
#209	3004	3016	99.60
#210	2647	2619	98.94
#212	2748	2749	99.96
#213	3251	3249	99.94
#214	2262	2256	99.73
#215	3363	3362	99.97
#217	2208	2207	99.95
#219	2154	2155	99.95
#220	2048	2048	100.00
#221	2427	2422	99.79
#222	2484	2602	95.46
#223	2605	2596	99.65
#228	2053	1565	76.23
#230	2256	2256	100.00
#231	1886	1571	83.30
#232	1780	1788	99.55
#233	3079	3076	99.90
#234	2753	2752	99.96
Average	**2288**	**2242**	**98.15**

It can be observed from Table 1 QRS detection accuracy of PT algorithmthat the lowest QRS detection accuracy of PT algorithm is fall on ECG record #228 with only 76.23%. It is because the record #228 contains high amplitude Premature Ventricular Contraction beats which each amplitude value is too high compare to the normal heart beat. As a result, whenver the adaptive algorithm encounter the arrhythmia beat, the threshold value is adjusted to high value. When the adptive threshold technique encounters the normal heartbeat again, it requires some additional for threshold readjustment and hence missed several QRS detection. However, the PT algorithm achieves overall average QRS complex detection accuracy of 98.15% of all 48 patient records available on MIT-BIH arrhythmia database.

In terms of timing performance, the clock cycle of each intermediate processing block of Pan and Tompkins algorithm is measured based on one patient record or 30 minutes ECG record obtained from arrhythmia MIT-BIH database. The number of cycle clock is then divided by SoC based ECG embedded system operating frequency (120MHz). The timing performance breakdown of 30 minutes ECG data is as shown in Figure 8.

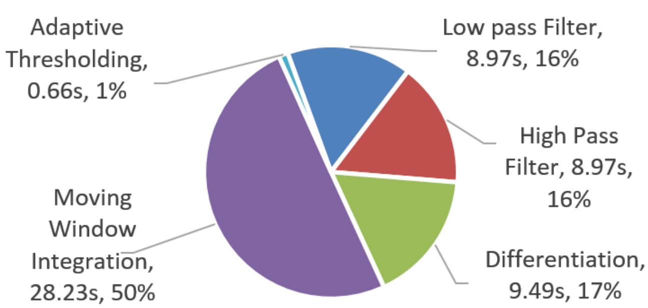


Figure 8 Timing analysis of Pan and Tompkins algorithm

It can be observed from Figure 8 that the total computation time of ECG QRS detection using PT algorithm of 30 minutes ECG data is 56.33 seconds. The moving window integration alone has consumed 28.23 seconds in computation which represent almost 50% of overall computation. This shows that the moving window integration is the computation bottleneck of the overall system performance.

5.2 QRS Detection Accuracy and Timing Performance of DB Algorithm with Adaptive Threshold

Figure 9 shows the computation output graph of each intermediate processing block of derivative based algorithm, from ECG raw data to detect QRS As PT algorithm, the QRS detection result of DB algorithm of all 48 ECG records is also compared with actual QRS annotation file provided in MIT-BIH database as shown in Table 2.

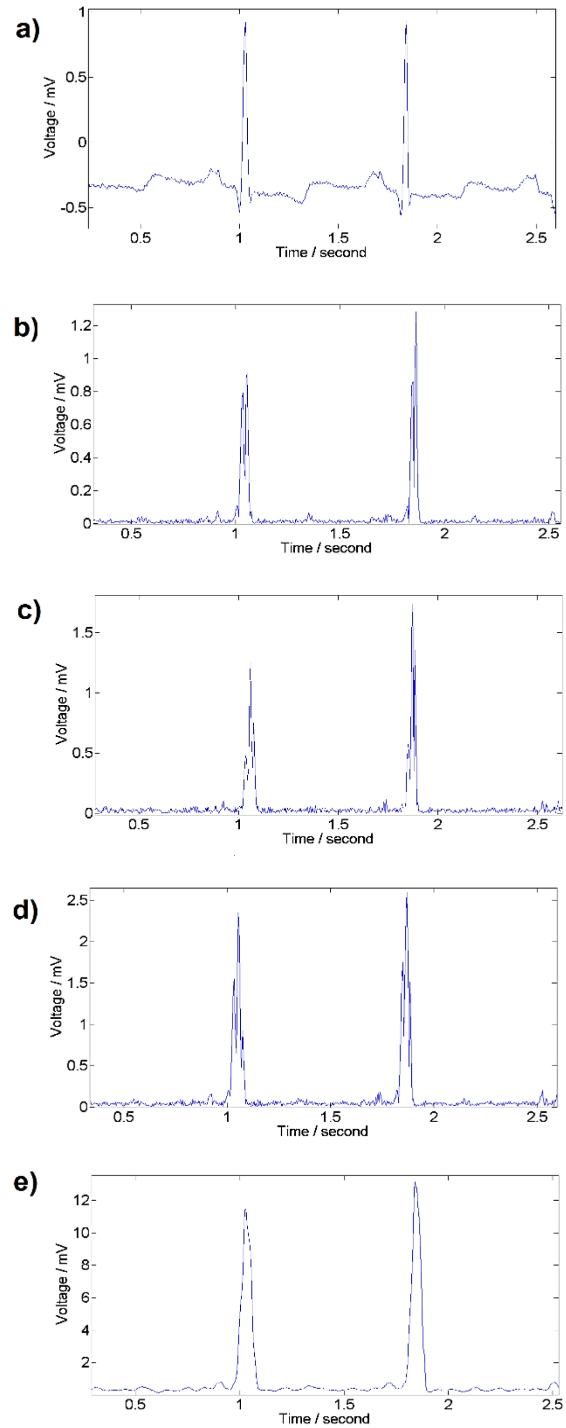


Figure 9 a) ECG Raw data b) First differentiation output c) Second differentiation output d) Scaling and Weighting output e) Smoothing output complex after smoothing operation

Table 2 QRS detection accuracy of DB algorithm

Patient Record	Total Actual Number Peaks	Total Peaks Detected	Accuracy (%)
#100	2273	2273	100.00
#101	1865	1874	99.52
#102	2187	2186	99.95
#103	2084	2084	100.00
#104	2230	2356	98.84
#105	2572	2640	97.42
#106	2027	1870	92.25
#107	2137	2131	99.72
#108	1763	2648	66.57
#109	2532	2558	98.98
#111	2124	2133	99.58
#112	2539	2546	99.73
#113	1795	1794	99.94
#114	1879	1901	98.84
#115	1953	1953	100.00
#116	2412	2394	99.25
#117	1535	1540	99.68
#118	2275	2296	99.08
#119	1987	1989	99.90
#121	1863	2041	91.27
#122	2476	2477	99.96
#123	1518	1515	99.80
#124	1619	1621	99.87
#200	2601	2831	91.87
#201	1963	1898	96.69
#202	2136	2132	99.81
#203	2982	3245	91.89
#205	2656	2643	99.51
#207	1862	2381	78.20
#208	2956	2679	90.63
#209	3004	3013	99.70
#210	2647	2684	98.62
#212	2748	2753	99.82
#213	3251	3229	99.32
#214	2262	2272	99.56
#215	3363	3369	99.82
#217	2208	2204	99.82
#219	2154	2151	99.86
#220	2048	2048	100.00
#221	2427	2177	89.70
#222	2484	2490	99.76
#223	2605	2609	99.85
#228	2053	2362	86.88
#230	2256	2262	99.73
#231	1886	1571	83.30
#232	1780	1793	99.27
#233	3079	3072	99.72
#234	2753	2748	99.82
Average	**2288**	**2321**	**96.73**

It can be observed from Table 1 QRS detection accuracy of PT algorithmthat the lowest QRS detection accuracy of DB algorithm is fall on ECG record #108 with only 66.57%. It is due to the record #108 is a noisy data which can be filtered using PT algorithm, but DB algorithm only use the differentiation method to obtain slope information only. Nevertheless, the DB algorithm still achieves average QRS complex detection accuracy of 96.73% for all 48 patient ECG records available on MIT-BIH arrhythmia database.

The computation timing performance of DB algorithm execution is presented in pie chart as shown in Figure 10 for 30 minutes ECG data or 1 patient ECG record. It can be observed from Figure 10 that the total time of QRS detection using DB algorithm for 30 minutes ECG data is 22.14 seconds. The longest operation of DB algorithm is smoothing operation which consumes 8.27 seconds or 37% of overall computation time. However, DB algorithm almost consume 60% less of the total computation time compare to the PT algorithm.

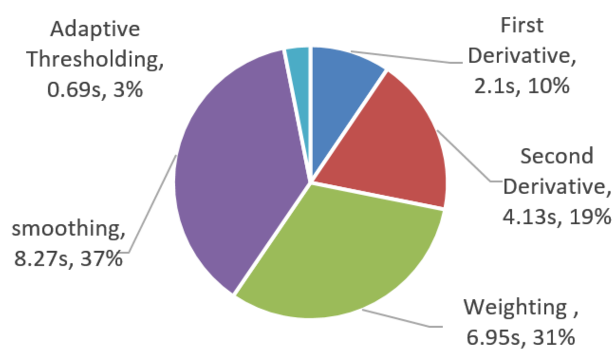


Figure 10 Timing Analysis of Derivative Based Algorithm

5.3 QRS Detection Accuracy a Timing Performance of Optimized PT Algorithm with Adaptive Threshold

In original Pan and Tompkins algorithm, moving window integration takes the longest time to compute which represents almost 50% of overall algorithm computation time. Considering the ECG record is 30 minutes with 200Hz sampling rate, it requires computation of 33 operations of each data point of total 360,000 data points.

However, with the optimization of the moving window integration, the operation is reduced from 33 operations to only 3 operation of each window of 360,000 data points while maintaining its QRS detection accuracy.

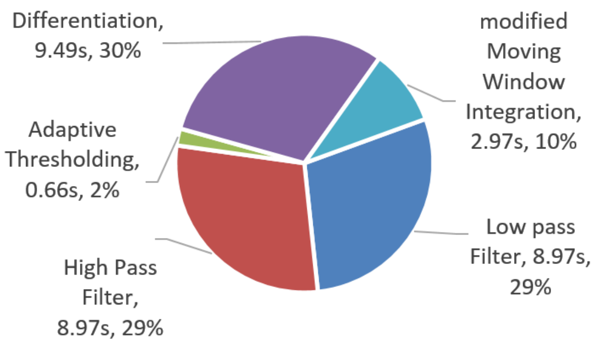


Figure 11 Timing analysis of Optimized Pan and Tompkins with Modified Moving Windows Integrator

Both original and optimized PT algorithm give the exactly same QRS detection accuracies of 98.15%. In terms of comparison of moving window integration alone, the optimized MWI consumes only 2.97 seconds compared to original MWI which are 28.23 seconds. This shows the modified MWI achieve 9.5 times speed faster from the original. However, in terms of overall timing performance, it can be observed that the optimized PT algorithm consumes only 31.06 seconds of total computation time compared to original PT algorithm of 56.33 seconds. This shows the overall system performance improvement of 44.86%.

In [3], Open Source ECG Analysis Software (OSEA) have been implemented on Xilinx Virtex-II pro embedded system board for QRS complex detection. This system use FPGA soft-processor by Xilinx and run at 100MHz. In embedded software execution, this system can process 30 minutes ECG data in 43.25 seconds.

In [21], an CWT-based QRS detection system have been implemented. Using low power embedded system which run only at 800KHz, it manage to process 3 seconds ECG data by 1.5 Million clock cycle which translated to 18.75 minutes (or 1125 seconds) to process 30 minutes of ECG data. This low power consumption thus more suitable for emergency ambulatory system instead of long hour ECG embedded processing.

Figure 3 summarizes and benchmarks the result of QRS detection accuracy and timing performance of all algorithms on 30 minutes ECG data.

6.0 CONCLUSION

This paper has presented result study of original and optimized version of Pan and Tompkins algorithms, as well as derivative based algorithm in ECG QRS complex detection. All these algorithms have been implemented on SoC-based embedded system architecture which prototype on Altera DE2-115 FPGA that house a Cyclone IV FPGA chip. The comparison and analysis is focus on QRS detection accuracy and computation timing performance, which the algorithm is implemented as embedded software. It has been tested by offline ECG data of 48 patients obtained from MIT-BIH database. QRS detection accuracy analysis result shows that the Pan and Tompkins algorithm yield better accuracies which are 98.15% compared to Derivative Based algorithm of 96.73%.

Table 3 QRS Detection Accuracy and Timing performance of three Algorithms on 30 minutes ECG data

Algorithm	Experiment Setup	Accuracy	Average Computation Time (s)
Original PT Algorithm	▪ Altera DE2-115 FPGA	98.15%	56.33
DB Algorithm	▪ Nios II Processor (full)	96.73%	22.14
Optimized PT algorithm	▪ f_{max}: 120 MHz	98.15%	31.06
CWT-based algorithm [21]	▪ f_{max}: 800Khz embedded system	99.81%	1125.00
OSEA [3]	▪ Xilinx Virtex-ii ▪ f_{max}: 100MHz	99.45%	43.25

Though Derivative Based algorithm use less time to process a 30 minutes ECG data which is 22.33 seconds compared to 56.5 seconds of PT algorithm, it is vulnerable to ECG data with a lot of noise which the detection accuracy could drop drastically. To improve the performance of Pan and Tompkins algorithm, an optimization of Moving Windows Integration has been proposed which shows 9.5 time faster than original Moving Windows Integration and overall system performance improvement of 44.86% while maintaining the same QRS detection accuracy.

In future we can apply these algorithms as ECG preprocessing technique for CVD detection and classification. Taking advantage of SoC and FPGA technology, the system functionality could be partitioned and system architecture could be designed using hardware/software co-design approach to strike the balance of design constraints of biomedical embedded system, in terms of computation timing performance, logic area and power consumption.

Acknowledgement

This work is supported by the Ministry of Science, Technology & Innovation of Malaysia (MOSTI) under Science Fund Grant 06-01-06-SF1098 (UTM Vote No. 4S061).

References

[1] Pedersen, O. D., H. Bagger, N. Keller, B. Marchant, L. Kober, and C. Torp-Pedersen. 2001. Efficacy of Dofetilide in the Treatment of Atrial Fibrillation-Flutter in Patients with Reduced Left Ventricular Function a Danish Investigations of Arrhythmia and Mortality on Dofetilide (DIAMOND) Substudy. *Circulation.* 104(3): 292-296.

[2] Acharya, R., A. Kumar, P. Bhat, C. Lim, N. Kannathal, and S. Krishnan. 2004. Classification of Cardiac Abnormalities using Heart Rate Signals. *Medical and Biological Engineering and Computing.* 42(3): 288-293.

[3] Cvikl, M., and A. Zemva. 2010. FPGA-Oriented HW/SW Implementation of ECG Beat Detection and Classification Algorithm. *Digital Signal Processing.* 20(1): 238-248.

[4] Jatmiko, W., P. Mursanto, A. Febrian, M. Fajar, W. Anggoro, R. Rambe, S. Eka. 2011. Arrhythmia Classification from Wavelet Feature on FGPA. *International Symposium on Micro-Nanomechatronics and Human Science (MHS).*

[5] Chang, M.-C., Z.-X. Lin, C.-W. Chang, H.-I. Chan, and W.-S. Feng. 2004. Design of a System-On-Chip for ECG Signal Processing*. The 2004 IEEE Asia-Pacific Conference on Circuits and Systems, 2004. Proceedings.*

[6] Jin, Z., Y. Sun, and A. C. Cheng. 2009. Predicting Cardiovascular Disease from Real-Time Electrocardiographic Monitoring: An Adaptive Machine Learning Approach on a Cell Phone. *Annual International Conference of the IEEE Engineering in Medicine and Biology Society.*

[7] Patel, A. M., P. K. Gakare, and A. Cheeran. 2012. Real Time ECG Feature Extraction and Arrhythmia Detection on a Mobile Platform. *International Journal of Computing Appication.* 44: 40-45.

[8] Balda, R., G. Diller, E. Deardorff, J. Doue, and P. Hsieh. 1977. The HP ECG Analysis Program. *Trends in Computer-Processed Electrocardiograms*. 197-205.

[9] Vijaya, G., V. Kumar, and H. Verma. 1998. ANN-Based QRS-Complex Analysis of ECG. *Journal of Medical Engineering & Technoloy*. 22(4): 160-167.

[10] Benitez, D., P. Gaydecki, A. Zaidi, and A. Fitzpatrick. 2000. A New QRS Detection Algorithm Based on the Hilbert Transform. *Conference on Computers in Cardiology.*

[11] Kadambe, S., R. Murray, and G. F. Boudreaux-Bartels. 1999. Wavelet Transform-Based QRS Complex Detector. *IEEE Transactions on Biomedical Engineering. 46*(7): 838-848.

[12] Poli, R., Cagnoni, S., & Valli, G. 1995. Genetic Design of Optimum Linear and Nonlinear QRS Detectors. *IEEE Transactions On Biomedical Engineerin.* 42(11): 1137-1141.

[13] Pan, J., & Tompkins, W. J. 1985. A Real-Time QRS Detection Algorithm. *IEEE Transactions on Biomedical Engineering. (*3): 230-236.

[14] Lipponen, J. A., & Tarvainen, M. P. 2013. Advanced Maternal ECG Removal and Noise Reduction for Application of Fetal QRS Detection. *Computing in Cardiology Conference (CINC).*

[15] Das, S. 2012. *Acquisition, Processing and Analysis of Normal, Diseased and Music Stimulated ECG Signals.* Jadavpur University Kolkata.

[16] Gradl, S., Kugler, P., Lohmuller, C., & Eskofier, B. 2012. Real-Time ECG Monitoring and Arrhythmia Detection using Android-Based Mobile Devices. *Annual International Conference of the IEEE Engineering in Medicine and Biology Society.*

[17] Friesen, G. M., Jannett, T. C., Jadallah, M. A., Yates, S. L., Quint, S. R., & Nagle, H. T. 1990. A Comparison of the Noise Sensitivity of Nine QRS Detection Algorithms. *IEEE Transactions on Biomedical Engineerin.* 37(1): 85-98.

[18] Hamilton, P. S., & Tompkins, W. J. 1986. Quantitative Investigation of QRS Detection Rules using the MIT/BIH Arrhythmia Database. *IEEE Transactions on Biomedical Engineering.* 12: 1157-1165.

[19] Bansal, D., Khan, M., & Salhan, A. 2009. A Review of Measurement and Analysis of Heart Rate Variability. *International Conference on Computer and Automation Engineering.*

[20] Sufi, F., Fang, Q., & Cosic, I. 2007. ECG RR Peak Detection on Mobile Phones. *29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society.*

[21] Romero, I., Grundlehner, B., Penders, J., Huisken, J., & Yassin, Y. H. 2009. Low-Power Robust Beat Detection in Ambulatory Cardiac Monitoring*. IEEE Biomedical Circuits and Systems Conference.*

[22] Yeh, Y. C., & Wang, W.-J. 2008. QRS Complexes Detection for ECG Signal: the Difference Operation Method. *Computer Methods and Programs in Biomedicine*. 91(3): 245-254.