VOL. 11, NO. 13, JULY 2016 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences


www.arpnjournals.com

ANALYSIS OF A PEAK DETECTION ALGORITHM USING SYSTEM-ON-CHIP ARCHITECTURE

Lim Chun Keat[1], Asral Bahari Jambek[1] and Uda Hashim[2]
[1]School of Microelectronic Engineering, Universiti Malaysia Perlis, Perlis, Malaysia
[2]Insituite of Nano Electronic Engineering, Universiti Malaysia Perlis, Perlis, Malaysia
E-Mail: wally.ck24@gmail.com

ABSTRACT

Peak detection is widely used in many signal-processing applications, since it allows automatic signal processing and produces faster result for users. In this work, an analysis of a peak detection algorithm implemented into a field-programmable gate array (FPGA) is discussed. A system-on-chip (SoC) hardware architecture was designed using the Altera platform to analyse the system data flow. A set of 1000 random data was executed by the peak detection algorithm in the SoC architecture. The output result obtained from the architecture were verified with the result obtained from a Matlab simulation. Based on the power consumption report, the reported power disipation of the system architecture is 202.79 mW.

Keywords: peak detection, microprocessor system, FPGA.

INTRODUCTION

Various peak detection algorithms have been proposed for better performance and efficiency as well as application requirement. A few examples of peak detection algorithm techniques are threshold, filter, and wavelet [5, 6, 8]. However, the performance and efficiency of peak detection algorithms are depending on application such as the QRS detection algorithm for electrocardiogram (ECG) signal [3, 4, 9].

A peak detection algorithm is a set of steps or a formula that is used to find one or more maximum values from a signal or a set of continuous data. They are widely used in many sectors such as the health science, chemical, communications, and military sectors [2–7,9–11]. In the health science sector, peak detection is used for heart arrhythmia detection based on ECG signal, arterial blood pressure (ABP) signal analysis, biosensor-based glucose concentration detection and blood cell analysis [3, 4, 6, 7, 9, 11].

In previous research work, the algorithm has been implemented into electronic systems, such as microprocessor systems or digital systems, to provide real-time detection [3-6, 8, 9, 11]. Implementing an algorithm within these systems allows it to improve upon the system performance and shorten the time required to carry out a specific task. Furthermore, a more accurate and precise output can be produced by using peak detection algorithm. Through very large scale integration (VLSI) approach, the algorithm is converted into an architecture which can be integrated with other system modules to form a system-on-chip (SoC) [3, 5, 6, 9]. This approach can be used to create portable and miniature devices; a wearable ECG monitoring device would be one such example [1].

In this paper, a study on a peak detection algorithm implemented in a microprocessor system is discussed. The microprocessor system is pre-loaded with a set of input data and the system is analysed through simulation and tested in hardware. Literature review section discusses several peak detection systems, including their architecture, algorithm and application. Methodology section presents the methodology of implementing a microprocessor system with peak detection algorithm within field-programmable gate array (FPGA). In results and discussion section, the simulation and experimental results of the peak detection algorithm run in the microprocessor system are discussed. Finally, Conclusion section concludes this paper.

LITERATURE REVIEW

Peak detection algorithms have been a particularly interesting topic for researchers. In one study, researcher applied a peak detection algorithm for heart arrhythmia detection into FPGA-based system [3]. The authors applied the basic QRS complexes detection algorithm, developed by Ahlstrom and Tompkins, to an FPGA. The system architecture for the proposed system is illustrated in Figure-1. The proposed system contains signal busses, a pro-program counter (PRO-PC),a program counter (PC), a control ROM, input, output, registers A and B, an arithmetic logic unit (ALU), a pointer (PTR), a parameter block (P) and random access memory (RAM). The architecture was synthesized and simulated, and the output was compared with the simulation results from Matlab software. The proposed system was implemented in Xilinx Spartan XC2S150 FPGA board with 1MHz clock frequency.

In paper [11], an embedded system based on an automatic hematology analyser, using a peak detection method, is introduced as illustrated in Figure-2. The analyser is used in blood cell analysis for medical treatment and prevention. Blood cell analysis is carried out by classifying and counting the white blood cell, using peak detection on the signal pulse from the test module. The embedded system design is illustrated in Figure-3. The system is separated into control system and management systems. The control system is used for signal acquisition, modulation, and as an actuator controller. It consists of a 32-bit digital signal processor (DSP), an actuator drive, a signal modulation circuit, an analogue-to-digital converter (ADC) and a complex

programmable logic device (CPLD) interface. The management system provides human-computer interaction, data management, data transmission, and system parameter setting. The system is operated by a microprocessor system that runs Linux operating system (OS), with a MiniGUI graphics interface and a SQLite database. The control system and the management system communicate using a serial peripheral interface (SPI).

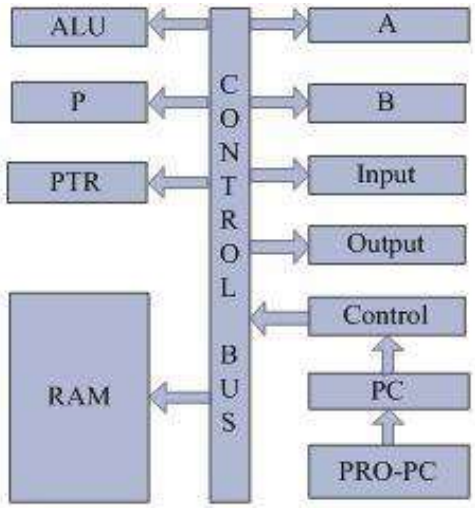

Figure-1. Block diagram of the system for heart arrhythmia detection [3].

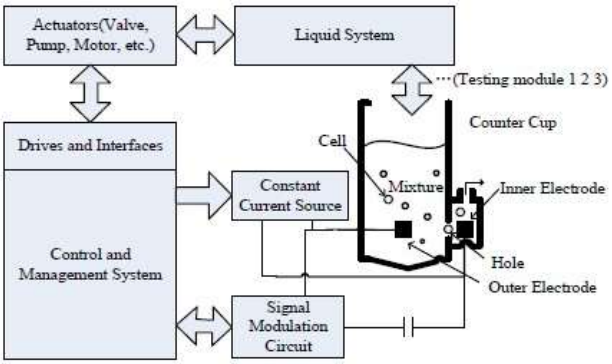

Figure-2. The automatic hematology analyser system [11].

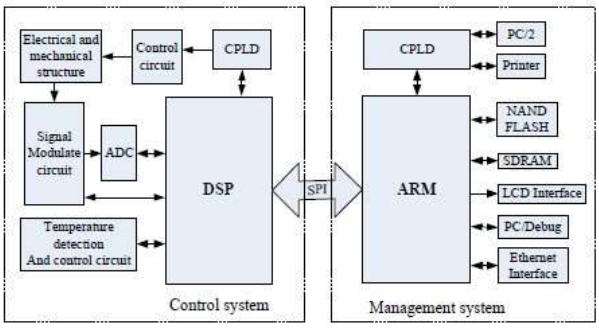

Figure-3. Block diagram of the embedded system design for automatic hematology analyser [11].

The authors in paper [5] implemented an FPGA-based digital signal processing system for radiation spectroscopy. The system consists of an ADC and an FPGA that performs the DSP. The digital signal processing system is illustrated in Figure-4. A peak detection algorithm was implemented into the FPGA, where it sampled the desired input signal and determined the pulse height and rise time. This information was used to identify the radiation energy, detecting the gamma-ray interaction depth and particle identification. Trapezoidal filtering, digital constant-fractional discrimination (digital CFD) and binomial filtering were used as parts of the digital signal processing method. The results is stored in an output buffer which could be accessed by personal computer through a PCI Express interface.

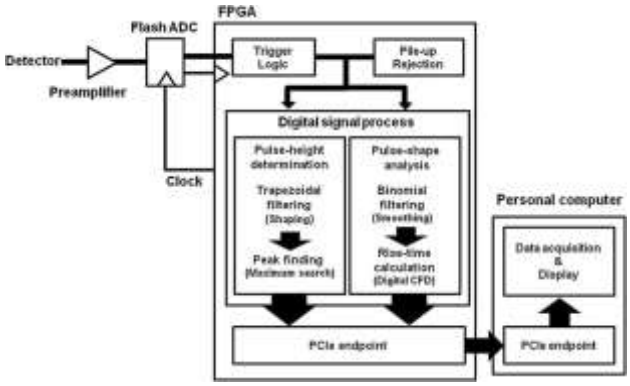

Figure-4. Block diagram of the digital signal processing system for radiation spectroscopy [5].

In paper [6], a unique peak detection algorithm for bioelectronics glucose sensor is introduced as shown in Figure-5. The bioelectronics glucose sensor is a multi-electrode array (MEA), which is used to detect glucose concentration. The interaction between the electrodes is captured in a form of electrical signal. The data acquisition and processing are based on a peak detection algorithm. The algorithm contains two different parts: stationary wavelet transform (SWT) filter action potential (AP) detection, and the slow-wave- (SW-) based processing. SWT filter AP detection is used for high frequency signals while SW-based processing is used for low frequency signals. The algorithm flow is plotted in Figure-5. The algorithm was implemented into the FPGA and the results were stored in a PC or accessed via USB.

A peak detection algorithm for portable multi-modal nano-bio sensor system is introduced by the authors of paper [8]. A multi-channel or multi-modal scheme was implemented into the nano-bio sensor system, as illustrated in Figure-6. It consisted of a signal condition circuit, an ADC, a DSP, a control unit, a DAC, a control interface and a digital interface. The proposed system consisted of four channel ADCs, a signal conditional circuit, a ZigBee wireless communication module, and an 8051 microcontroller. An auto-threshold peak detection algorithm was implemented in the proposed system, due to its low computation cost.

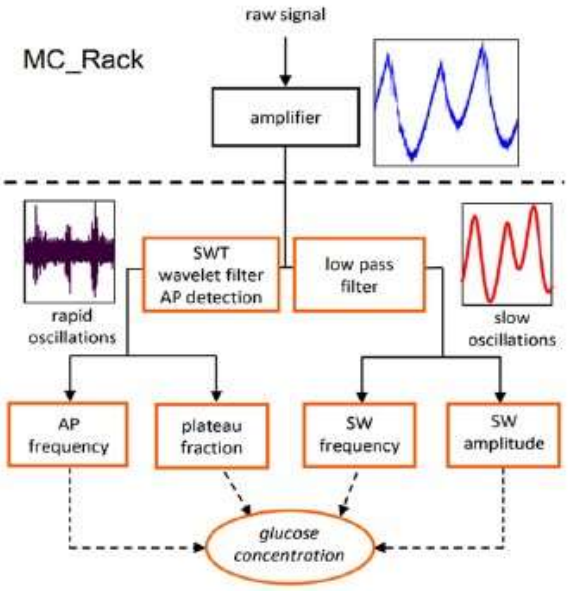

Figure-5. Data processing flow for glucose concentration [6].

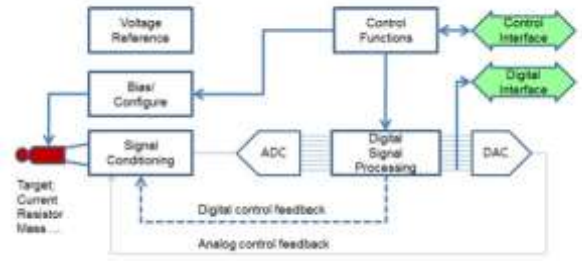

Figure-6. Block diagram of the nano-bio sensor controller [8].

Based on the previous discussion, the main features and the differences between these peak detection systems are tabulated in Table 1. From the table, it is clear that most of the systems used a dedicated architecture to run their algorithms. This improves system performance and reduces any unnecessary components. These systems were typically designed together with conventional processors, such as the ARM and 8051 processors. Most system used FPGA as a platform, due to its flexibility. However, hard processors are also a popular choice for systems that require high performance.

Table-1. Comparison of different peak detection systems.

	[3]	[11]	[5]	[6]	[8]
Processor	Dedicated	DSP, ARM	Dedicated	Dedicated	8051
Algorithm	QRS Complexes Detection	Peak detection (peak counting)	Peak, pulse height, rise time detection	Action Potential detection	Peak detection
Application	Heart Arrhythmia	Hematology	Radiation spectro-scopy	Glucose conc.	Nano-bio sensor
Output	Yes	USB, LCD, SD card	PCI Express interface to PC	USB, PC	RF, UART, USB, LCD
Hardware	FPGA	DSP & µP chip	FPGA	FPGA	µP chip
ADC	-	Yes	AD9254	Yes	Yes

In this paper, a simple peak detection algorithm is implemented into a microprocessor system. The purpose of the work is to analyse the performance and accuracy of the algorithm in the microprocessor system. An experiment is carried out to test the peak detection algorithm using pre-load benchmark data in the on-chip memory.

METHODOLOGY

In this work, a peak detection system is proposed to analyse the data flow in a microprocessor system. The microprocessor system utilizes an Altera Nios II processor system, as illustrated in Figure-7. The system consists of an economy version Nios II processor, Avalon busses, 64 kB on-chip memory, 32 MB synchronous dynamic random access memory (SDRAM) with an SDRAM driver, a Joint Test Action Group (JTAG) universal asynchronous receiver transmitter (UART), general purpose input and output (GPIO), and a 16x2-character LCD with driver. The system architecture is designed and synthesized using Qsys tool: a system integration tool from Quartus II software. The architecture was programmed into a Cyclone II EP2C70F896C7 FPGA chip in an Altera DE2-70 board, using a Quartus II programmer.

The peak detection algorithm was implemented to mimic the "findpeaks" function in the Matlab software. This algorithm is used to find the local maxima and local minima of a signal. Figure-8 presents the state diagram of the peak detection flow. The algorithm starts by identifying whether the data are increasing or decreasing. After the first datum, the "UP" state or "DOWN" state is selected. The change of state will depend on the subsequent datum. An equal state - represent by the "UP="

state and "DOWN=" state - is selected when the current datum and previous datum have the same value. Whenever there is a transition from a low to a high value, or vice versa, the local maximum or local minimum is detected, respectively. The dashed line arrow indicates that the previous datum is a local maximum while the dotted line arrow indicates that the previous datum is local minimum.

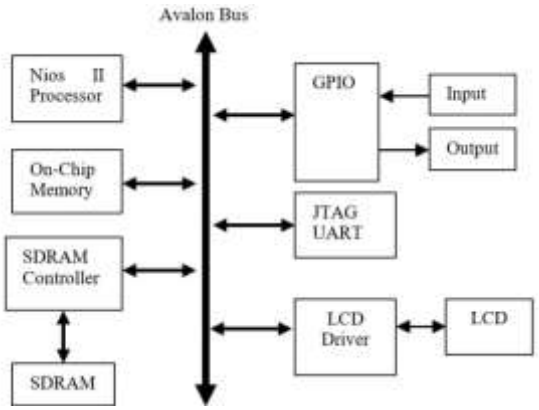

Figure-7. Block diagram of the proposed peak detection system.

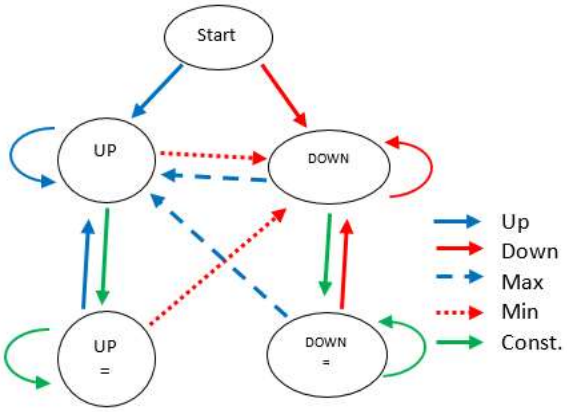

Figure-8. State diagram of the peak detection algorithm.

To evaluate the accuracy of our system, first we generated a random data set using Matlab. Then, these data were analysed, first using Matlab's built-in "findpeaks" command. Next, the same task was carried out using our own microprocessor system. The results from our microprocessor system were then compared with the Matlab results to measure their accuracy. One thousand data were generated randomly, using the Matlab software. Then, Matlab's "findpeaks" command was applied to the data to identify the number of peaks. The result are illustrated in Figure-9. To evaluate the effectiveness of the microprocessor system, the same data were converted into hexadecimal, which were then loaded into the microprocessor system.

The peak detection algorithm is coded in a C language for a Nios II processor. The pseudocode is shown in Figure-10. The pseudocode was designed based on the state diagram of the peak detection algorithm in Figure-8. In the pseudocode, only the detected peak number, the peak amplitude and distance between the peaks are record. The source code was then compiled and the hex file built using Nios II Software Build Tools (SBT). The algorithm and data were loaded into the on-chip memory of a Nios II processor system, through the Nios II SBT.

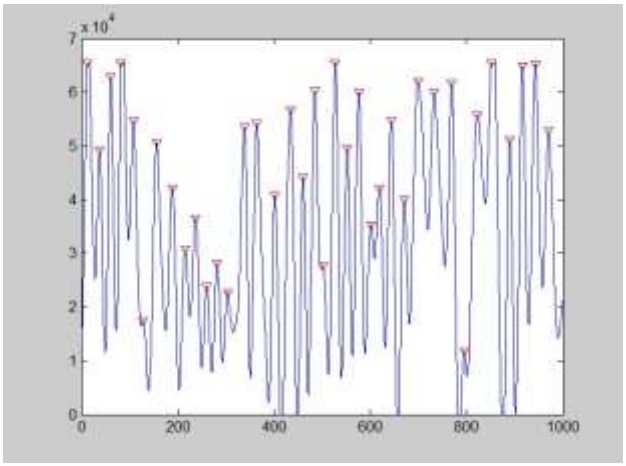

Figure-9. Detected peaks from the 1000 data plotted using the Matlab software.

Figure-10. Pseudocode of the peak detection algorithm.

RESULT AND DISCUSSIONS

This section discusses our experimental results in order to verify the performance of our peak detection system using a microprocessor. First, we discuss the peak detection obtained from Matlab's "findpeaks" command. Then, we will present the results based on our own microprocessor system. Figure-9 shows the result analysis using the Matlab software; the red triangle in the graph is the detected peak. Based on this analysis, 37 local maxima or peaks is detected using the "findpeaks" command.

To verify our microprocessor system, a simulation was carried out using Altera ModelSim to

analyse the data flow in the microprocessor system with the algorithm running inside. Figure-11 shows the signal flow of a read operation on an on-chip memory, using Nios II processor. In the Figure-11, the processor carries out a read operation on a memory address of 0x0873 to retrieve 16-bit data with value of 0x592B. The data value stored in the address is 0x6D26592B. In the read operation on the on-chip memory, the *chip select* signal was set high in the beginning. Then, the *byteenable* signal was set to 0011b (in binary) and the *address* signal was set as selected. The *byteenable* signal was set to allow byte read. In this operation, lower 16-bits was selected. When the chip select signal was set low, the *address* signal was latched and the *readdata* signal received data, based on the address signal.

Figure-12 shows the signal sent between the microprocessor system and the SDRAM, when storing the information concerning a detected peak on the SDRAM. When a peak is detected, the microprocessor will store the sequential order of the peak, the amplitude of the peak and the distance between the current peak and previous peak. The waveform in Figure-12 shows 2nd peak, with an amplitude of 49,366 (0xC0D6 in hexadecimal) and a distance from the previous detected peak of 26 data (0x001A in hexadecimal). The number of the peak is stored in SDRAM address location 0x0003, while the amplitude of the peak is stored in SDRAM address location 0x0004 and the distance between peaks is stored in SDRAM location 0x0005.

The waveform in Figure-12 is separated into three sections, which are shown in Figure-13, Figure-14 and Figure-15. In Figure-13, the waveform shows the write operation as carried out on the SDRAM at memory location of 0x0003, with the data value of 0x0002. This write operation is carried out to record the number of detected peaks. In the process of writing the data to the SDRAM by microprocessor, the *CS* (chip select) signal is set low to enable access to the SDRAM. The *ADDR* (address) signal is set, and the *RAS* (row access strobe) signal is set low to select the row of memory to be accessed. The *RAS* signal is set high once the address is set. Next, the *ADDR* signal is set, and the *CAS* (column access strobe) signal is set low to select the column of memory to be accessed. At the same time, the *DQ* (data) signal is set with the value to be store and *WE* (write enable) signal is set low for the write operation.

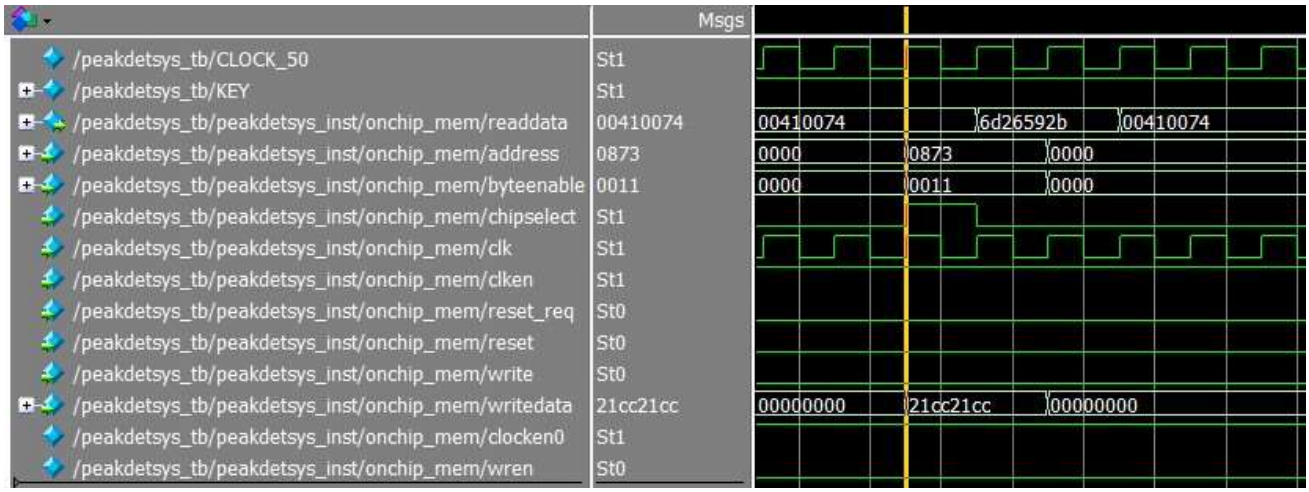

Figure-11. Waveform of the read operation on the on-chip memory by a Nios II processor.

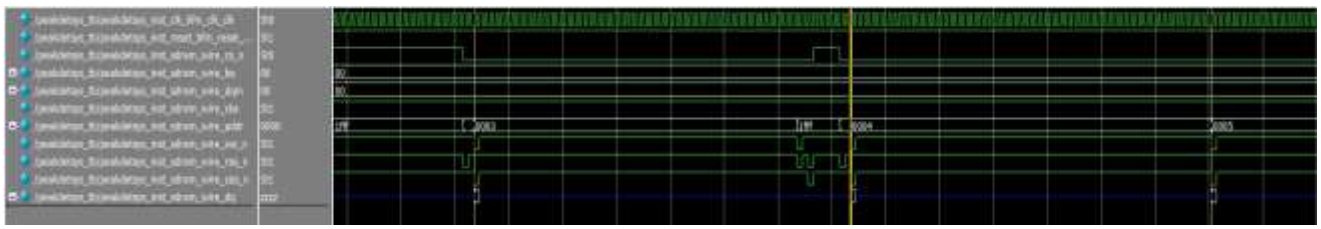

Figure-12. Waveform of the write operation on the SDRAM by a Nios II processor.

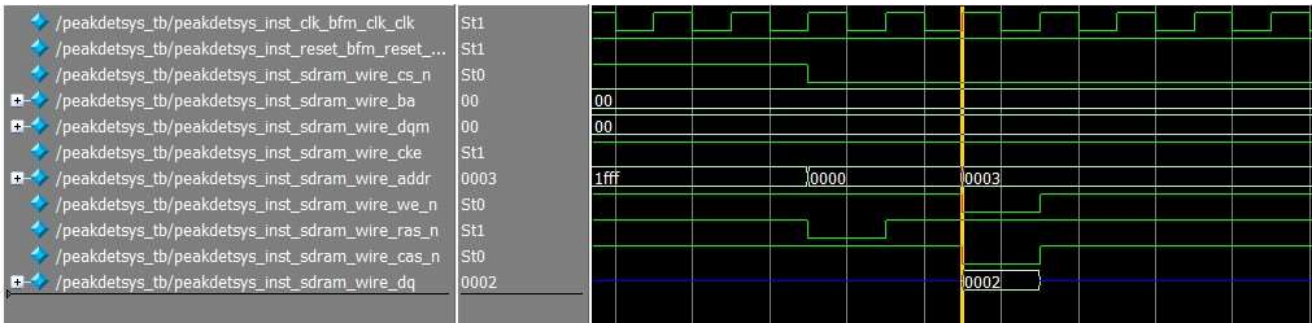

Figure-13. Waveform of the write operation on SDRAM memory location 0x0003, with value 0x0002.

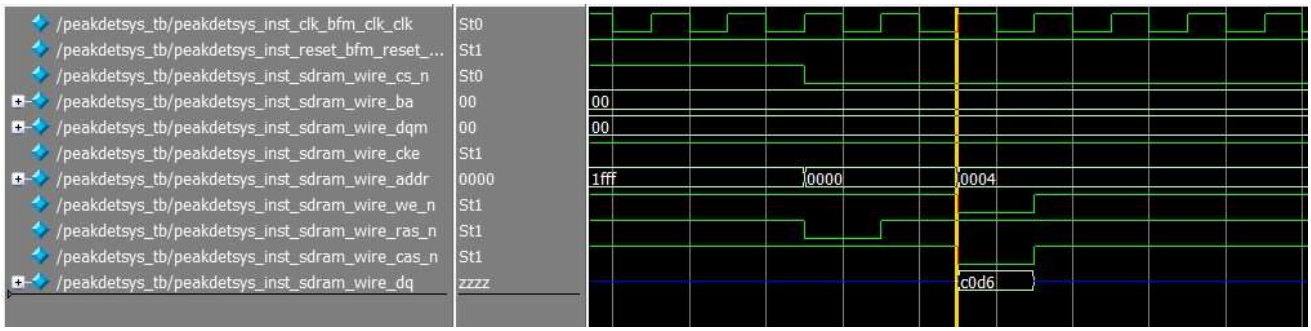

Figure-14. Waveform of the write operation on SDRAM memory location 0x0004, with value 0xC0D6.

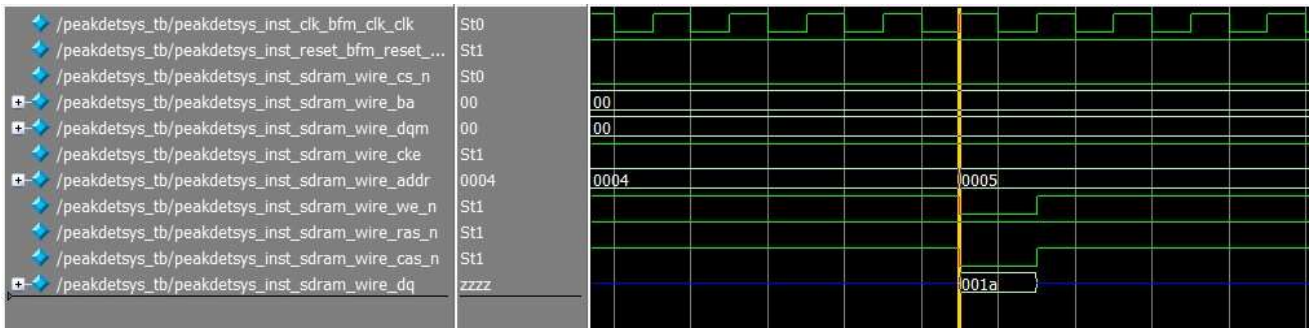

Figure-15. Waveform of the write operation on SDRAM memory location 0x0005, with value 0x001A.

Once the write operation is completed, the *CAS* signal and *WE* signal are both set high. The transition of each signal is based on a positive edge clock pulse and the signal will only transition after a cycle of the clock pulse. The waveform in Figure-14 shows that the amplitude of the current peak is recorded by storing the 49,366 peak value in hexadecimal (0xC0D6) in memory location 0x0004. In Figure-15, the waveform of the write operation shows the data value 0x001A is stored in memory location 0x0005, for the purposes of recording the distance between the current and previous detected peaks.

The microprocessor system in the Altera DE2-70 board has the algorithm and set of 1000 data added to it, by loading the generated hex file from the Nios II SBT to the on-chip memory. The Nios II console that communicates with the Nios II processor system showed that there was a total of 37 peaks detected from the data set. Figure-16 shows the data and peaks detected by the Matlab "findpeaks" command (the detected peaks are marked with red triangle) and our microprocessor system (the detected peaks are marked with magenta square). These results prove that the algorithm run in our microprocessor system is able to obtain the same results as those obtained from the simulation by the Matlab software. The output of the Nios II console is shown in Figure-17. The console shows the number of peaks detected, the distance between the detected peak and the amplitude of the peaks in hexadecimal. Besides this, the microprocessor system also contains a 16x2-character LCD, which was used to present the result on-board. A push button was used to display the different results on the 16x2-character LCD. The on-board results-displaying LCD is shown in Figure-18.

An analysis of the power consumption of the system architecture was carried out using the PowerPlay Power Analyser in the Quartus II software. The report showed that the total thermal power dissipation for the architecture was 202.79 mW. The core static power dissipation was 154.98 mW and the I/O thermal power dissipation was 47.10 mW. The power of the system was

analysed using a default toggle rate of 12.5 % for input and output signal, and a toggle rate for remaining signals based on vectorless estimation.

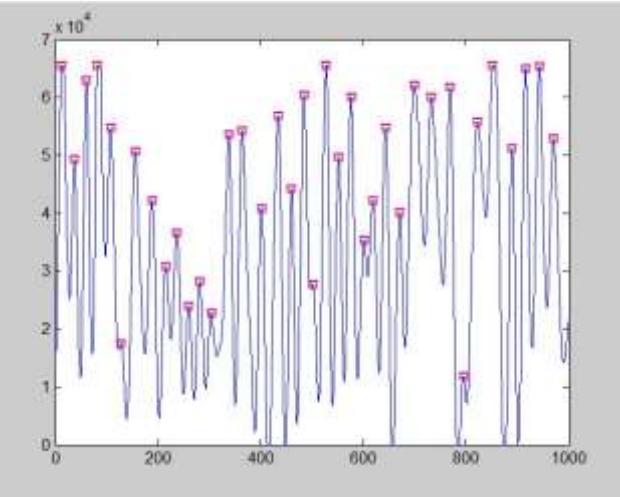

Figure-16. Detected peak results from the "findpeaks" command and results from the microprocessor system.

Figure-17. Results of detected peak display in the Nios II console.

Figure-18. Results of detected peak display on the on-board LCD.

CONCLUSIONS

In this paper, a simple peak detection algorithm was implemented in a NIOS II microprocessor system. Several peak detection systems with different applications were reviewed and compared. A set of 1000 data was pre-loaded into the memory for the microprocessor system to run the peak detection algorithm. An analysis of the data flow in the microprocessor system was carried out via a simulation in Altera ModelSim. The microprocessor system architecture was loaded into an Altera DE2-70 board for testing purpose. After testing the algorithm in the microprocessor system, the results showed that 37 peaks were detected from 1000 data, which were the same results as those obtained from the simulation using the Matlab software. Based on the power consumption report, the estimated power required by the system architecture was 202.79 mW. In future work, a real-time peak detection microprocessor system with ADC and a dedicated on-chip sensor input will be studied.

ACKNOWLEDGEMENT

This research was supported in part by the Fundamental Research Grant Scheme, Ministry of Higher Education, Malaysia (FRGS Phase 1/2014).

REFERENCES

[1] C.J. Deepu, X. Zhang, W.S. Liew, D.L.T. Wong, Y. Lian. 2013. An ECG-SoC with 535nW/channel lossless data compression for wearable sensors, in: Proc. 2013 IEEE Asian Solid-State Circuits Conf. A-SSCC 2013. pp. 145-148.

[2] M.P. Fok, Y.D.Y. Deng, P.R. Prucnal. 2009. Optical auto-correlation peak discriminator for optical CDMA signals detection, in: 2009 Conf. Lasers Electro-Optics 2009 Conf. Quantum Electron. Laser Sci. Conf. pp. 1-2.

[3] S.M.R. Islam, A.F.M.N. Uddin, M.B. Hossain, M.I. Khan. 2013. FPGA based Heart Arrhythmia's Detection Algorithm. ACEEE Int. J. on Information Technology. 3(1): 9-14.

[4] H. Kim, S. Kim, N. Van Helleputte, A. Artes, M. Konijnenburg, J. Huisken, *et al.* 2014. A configurable and low-power mixed signal SoC for portable ECG monitoring applications. IEEE Transactions on Biomedical Circuits and Systems. 8(2): 257-267.

[5] P.S. Lee, C.S. Lee, J.H. Lee. 2013. Development of FPGA-based digital signal processing system for radiation spectroscopy. Radiation Measurements. 48(1): 12-17.

[6] Q.V. Nguyen, A. Caro, M. Raoux, A. Quotb, J.B. Floderer, Y. Bornat, *et al*. 2013. A novel bioelectronic glucose sensor to process distinct electrical activities of pancreatic beta-cells, in: Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBS. pp. 172-175.

[7] A. Pachauri, M. Bhuyan. 2011. ABP peak detection using energy analysis technique, in: 2011 Int. Conf. Multimedia, Signal Process. Commun. Technol. pp. 36-39.

[8] J. Park, J. Song, H. Kim, D. Ryu. 2013. Peak Detection for Portable Multi-modal Nano-bio Sensor System. International Journal of Bio-Science and Bio-Technology. 5(3): 135-142.

[9] O.C. Pin, A.B. Jambek, S. Yaacob. 2014. Circuit architectures reviews for portable ECG signal analyzer, in: 2014 2nd Int. Conf. Electron. Des. Penang: pp. 261-264.

[10] G. Prema, P. Gayatri. 2014. Blind spectrum sensing method for OFDM signal detection in Cognitive Radio communications, in: 2014 Int. Conf. Commun. Netw. Technol. pp. 42-47.

[11] P. Zhao, F. Fang, X. Ma, Z. Dong. 2014. Embedded system design and implementation in novel automatic hematology analyzer, in: 2014 IEEE Int. Conf. Mechatronics Autom. pp. 952-957.