Adaptive Sampling for ECG Detection Based on Compression Dictionary

Zhongyun Yuan, Jong Hak Kim, and Jun Dong Cho

***Abstract*—This paper presents an adaptive sampling method for electrocardiogram (ECG) signal detection. First, by employing the strings matching process with compression dictionary, we recognize each segment of ECG with different characteristics. Then, based on the non-uniform sampling strategy, the sampling rate is determined adaptively. As the results of simulation indicated, our approach reconstructed the ECG signal at an optimized sampling rate with the guarantee of ECG integrity. Compared with the existing adaptive sampling technique, our approach acquires an ECG signal at a 30% lower sampling rate. Finally, the experiment exhibits its superiority in terms of energy efficiency and memory capacity performance.**

***Index Terms*—Adaptive sampling, ECG, redundant processing, dictionary-based compression, FPGA**

I. INTRODUCTION

Nowadays, society is facing a challenge of the aged high-health-risk populations, health care problems become a hot issue. Meanwhile, current medical innovation model is shifted from the traditional hospital-centered one toward family, community, and workplace-centered one. Fig. 1 exhibits a typical architecture of a modern bio-sensor network system. Along with technique developments in semiconductors and wireless sensor networks (WSNs), bio-sensor systems have become more portable, low-power, intelligent and micro-sized such as Mobihealth [4] and Intel [5].

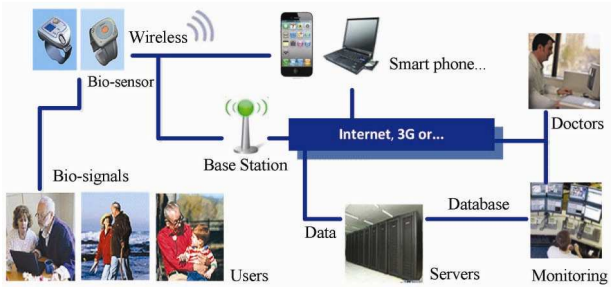

Fig. 1. Typical architecture of the modern bio-sensor systems.

In the wireless nodes, the most power consumption is from data transmission. For example, the power consumption of one bit of data transmitted is approximately equal to running 1000 CPU codes on a sensor node [9, 10]. A large amount of samples certainly aggravates the energy burden of data transmission. Thus, adaptive sampling and data compression technologies have been widely accepted as efficient methods to satisfy low-power requirements in WSNs applications. In this paper, we focus on the adaptive sampling of ECG signal through smart selection of sampling rates, and thus we significantly reduce the sampling cost.

Due to the inclusion of a smaller number of samples, the sampling speed is expected to be low. Nonetheless, the Nyquist-rate limitation cannot be avoided on account of the occurrence of aliasing [6] such that ambiguity occurs in the measurement. Consequently, the speed of frequency measurement should be adopted carefully. In the comparison of various relevant algorithms and techniques [1-3, 7], the traditional sampling rate adjustment is based on the analog signal analysis or statistical characteristics frameworks, which require a complex processing. Unfortunately, any algorithm

Manuscript received Apr. 3, 2013; accepted Oct. 1, 2013
Department of Electrical and Computer Engineering, Sungkyunkwan University, Suwon, Korea
E-mail : jdcho@skku.edu

implemented in WSN suffers from nodes resource (such as power consumption and size) limitations. Thus, an efficient sampling algorithm should be designed for WSN with low computation complexity and high resource utilization. In this paper, we utilize the existing data compression processing as the adaptive sampling control logic, so as to avoid including an additional algorithm for adaptive scheme.

The organization of this paper is as follows. Section II introduces the related works and our motivation. Section III explains our adaptive sampling scheme, and shows our proposed algorithm framework with hardware platform implementation. Section IV discusses the simulation results. Finally, Section V concludes the paper.

II. BACKGROUND AND RELATED WORK

1. Adaptive ECG Acquisition

In the time-domain sampling theorem, Nyquist-Shannon defines the minimum sampling rate: $f_{sampling}$ guaranteeing the reconstruction of the sampled signal [6], and $f_{sampling} \geq f_k$. where f_k is the maximum frequency in the power spectrum of the signal. For example, Fig. 2 exhibits a typical ECG transient signal. If we realize the rate selection of sampling based on the Nyquist-Shannon theory, this signal can be divided into two parts in the time domain: the low sampling rate segment f_{low} and high sampling rate segment f_{high} (in the *QRS* complex segment). For sampling the classical transient ECG signal in Fig. 2, the desired scheme is given in Table 1 [1-3]. The sampling scheme usually adopts 12bit sampling precision (f_{high} > 500samples/s and f_{low} > 64samples/s). Note that we need to use non-uniform sampling taking the Nyquist-Shannon theorem into consideration. In other words, the sampling rate of the *QRS* complex (f_{high}) is higher than the others (f_{low}). Because of the requirement of integrity signal reconstruction, a large number of samples are applied in the *QRS* complex segment. Whereas, the smaller number of samples are demanded by the other segments. Except for the *QRS* segment, many stationary samples containing dominant redundant data are not necessary for reconstruction [7]. Furthermore, we can easily find that the sample rate can be related to the redundancy of samples (Fig. 4).

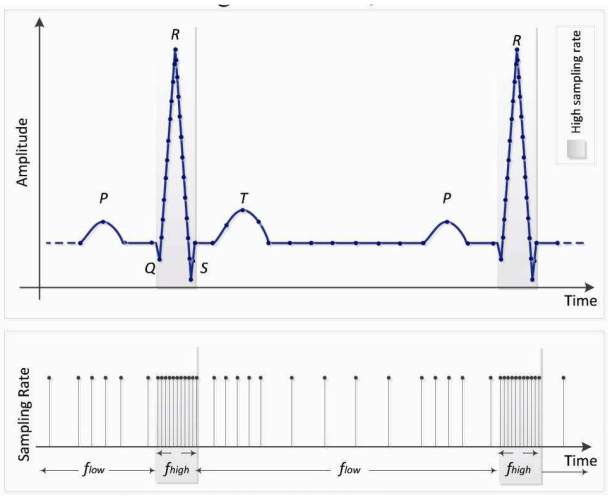

Fig. 2. Adaptive sampling principle of the ECG signal.

Table 1. Non-uniform sampling schemes in [1-3]

Non-uniform sampling	Reference [1]	Reference [2]	Reference [3]	Our scheme
f_{high}	1000Hz	512Hz	1000Hz	800Hz
f_{low}	100Hz	64Hz	200Hz	25-100Hz
Precision	12bit	12bit	12bit	12bit

2. Adaptive Sampling Model

In various data acquisition designs, an efficient sampling and compression processing is essential. It can significantly reduce the size of redundant samples, with the benefit of extra energy savings in DSP and transmission. For example, in our ECG signal detection cases in Fig. 3, there are three types of sampling schemes. Scheme *A* is based on a uniform sampling framework that adopts a fixed sampling rate. Compared with the other two adaptive sampling methods *B* and *C*, more samples are accumulated in *A,* which required extra hardware memory resources and energy.

In Fig. 3, *B* illustrates a classical adaptive sampling scheme in ECG signal detection, which is realized on a thresholding principle [2, 15]. When the raw analog signal is compared to a threshold voltage value, if the signal level is above the threshold, a fast sampling rate f_{high} will be adopted. Otherwise, a slow rate f_{low} will be fed back with low-cost samples. However, the thresholding principle scheme certainly increases the complexity and adds an additional analog circuit [16]. For the size-restricted ECG senor nodes, any unnecessary resource cost may make system unavailable. Different

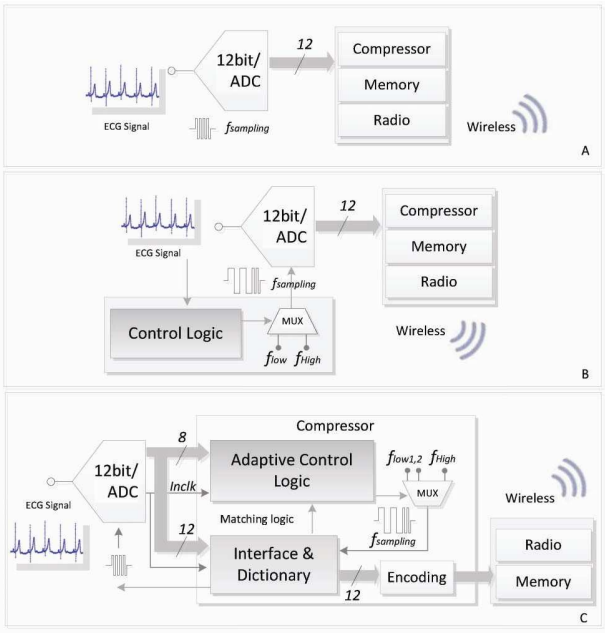

Fig. 3. (a) Uniform sampling, (b) Traditional adaptive sampling [2], (c) Proposed architecture of an adaptive sampling.

from the traditional adaptive sampling scheme, our proposed method of adaptive sampling (Fig. 3 *C*) depends on dictionary's redundant analysis with an existing compression processing. Finally, based on the feedback of the rate control logic, the required samples can be acquired with an adaptive synchronous rate. Let us compare the scheme *B* and *C* as shown in Fig. 3. In order to complete an adaptive sampling, the scheme *B* needs an additional analog circuit. Such as, H. J. Kim et al. [2] designed the architecture consisted of a differentiator, a comparator, and a multiplexer. Instead, the proposed scheme *C* setups an adaptive logic without any extra hardware, which re-utilizes an existing compression algorithm as a core of sampling strategy.

As shown in Fig. 3, our sampling scheme *C* is similar to the scheme *B* in terms of sampling rate setting. Both of them adopt the non-uniform sampling (using a multi-rate of sampling). For instance, if a large number of repetitive samples occur (such as samples between *T* and *P* segment), a lower sampling rate f_{low} is employed. In contrast, in the *QRS* complex, if the samples redundancy significantly decreases, then the current low sampling rate cannot satisfy the requirement of *QRS* reconstruction. Thus a higher rate f_{high} is prerequisite. In scheme *C*, another main difference with the scheme *B* is that we set a more flexible rate based on the matching length of strings. According to these matching results of each string in the dictionary, the sampling rate can be classified into f_{high}, f_{low}, and f_{deep}.

In order to illustrate our motivation, Fig. 4 describes the relationship between the adaptive sampling and the responded compressing coding clock. Fig. 4(a) shows adaptive sampling mode for ECG measurement. Fig. 4(b) exhibits a LZW-based coding clock for ECG0328.dat. In the LZW processing, coding clock reflects the state of matching operation in dictionary. D. Salomon [14] also points that matching operation presents the redundant state of samples. Based on these operations, the transient impulse is accompanied by the low redundancy. For example, if we compress an ECG in Fig. 4, the *QRS* complex segment can be distinguished from the other stable state segments, since the coding clock gives a high-frequent responding for *QRS*. Eventually, according to this coding clock in Fig. 4(b), we can relate the data compressing with the adaptive sampling. By doing so, our sampling estimates sampling rate on different segments (such as *QRS*, *P* and *T* segment).

Table 2 exhibits a comparison result of match processing between the initial part of the ECG and the *QRS* complex in a group of ECG samples (ECG0328.dat in Fig. 8). Also the table shows the maximum length of string in processing and code number. Table 2 shows a

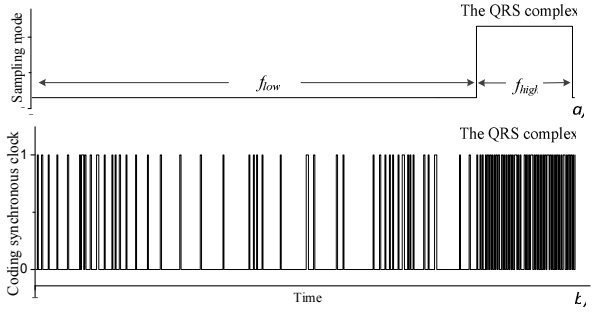

Fig. 4. The principle illumination of proposed scheme (a) Adaptive sampling strategy for ECG0328.dat signal, (b) LZW compression coding clock to ECG0328.dat. *Note: For the f_{high} sampling period (the *QRS* complex segment), the compression coding clock also give a high frequency response.

Table 2. Dictionary operation in the different ECG segments

Dictionary operation	Matched	Mismatched	String max-length	Codes number
Before *QRS*	144	8	12byte	48
QRS	3	40	2byte	43

matching result of ECG signal (ECG0328.dat). In the *QRS* complex segment, the signal samples have low redundancy. Thus, in the dictionary operations, there are 3 matched operations and 40 mismatched operations. In contrast, in the segment between two *QRSs*, there are 144 matched operations and 8 mismatched operations included in compression processing. Because of the inherent repetitive samples and high redundancy, a large number of stable data exists in this segment, and thus samples are frequently matched in the processing of look-up dictionary. Moreover, the length of string is grown, and thus, as shown in Table 2, the string max-length is 12byte. Here, the string length is another key factor for sampling, since the string length reflects the degree of accumulation of redundant data in dictionary.

III. ALGORITHM AND IMPLEMENTATION

1. Overall Algorithm

Fig. 5 exhibits our sampling scheme procedure. The counter i is a matching ratio of samples with the entry in the redundant dictionary. The counter j is a mismatching ratio of samples in the entry of redundant dictionary. The *string* > k represents the trigger value of lowest sampling. Other sampling parameters include f_{high} = 800 samples/s, f_{low} = 100 samples/s, and f_{deep} = 25 samples/s.

1) *Mode 1* ($f_{sampling}=f_{low}$): this mode's control logic is based on the matching result in the redundant dictionary. If samples have a high matching ratio (counter $i>n$), a low sampling rate logic $f_{sampling}=f_{low}$ will be fed back. For example, for the initial samples of an ECG signal in Fig. 2 before the *QRS* complex sensing, a large number of repetitive samples are accumulated. The sampling matching ratio is high. Hence, a low sampling rate ($f_{sampling}=f_{low}$) is adopted.

2) *Mode 2* ($f_{sampling}=f_{high}$): the triggering of this mode control depends on the mismatching ratio ($j>m$). For example, as shown in Table 2, most samples do not match in the redundant dictionary, when the *QRS* complex is sensed. Hence, a high sampling rate $f_{sampling} = f_{high}$ should be permitted. In addition, in order to avoid the *QRS* complex segment accumulated in the dictionary, after testing every *QRS* sampling, the redundant dictionary must be cleaned, by erasing all saved strings.

3) *Mode 3* ($f_{sampling}=f_{deep}$): After an $f_{sampling}=f_{low}$ mode is adopted, the algorithm should check the length of matching samples. If the length of matching samples is out of a trigger value (length of *string>k)* for a single string, a lower sampling rate $f_{sampling}=f_{deep}$ mode should be adopted. For example, with the given amount of repetitive samples, the matched sample length is significant grown. Such samples represent in Table 2 a higher redundancy occurrence. Hence, a deep sparse sampling mode ($f_{sampling}=f_{deep}$) should be adopted. Next, we describe our approach as follows:

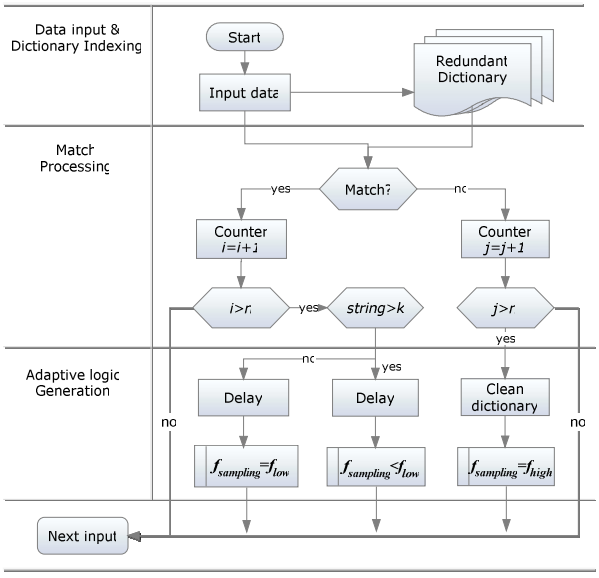

Fig. 5. The flow chart of our ECG signals sampling strategy.

Adaptive Sampling Algorithm

Setp 1 Initialize the dictionary parameter, setting $f_{sampling}=f_{high}$ mode.

Setp 2 Generate the string and index value, completing a lookup-based operation in the dictionary (LZW dictionary).

Setp 3 Compare the string value with the dictionary value. If there are matching, set $i=i$, $j=0$. Otherwise, set $i=0$, $j=j$, and jump to Step 5.

Setp 4 *Set matching counter* $i=i+1$. If $i>n$, enter into $f_{sampling}=f_{low}$ mode with a delay and *reset* $i=0$, *and* jump to Step 6. Otherwise, stay in the original sampling mode, and go back to Step 2.

Setp 5 Set mismatching counter $j=j+1$. If $j>m$, enter into $f_{sampling}=f_{high}$ mode, and reset $j=0$. Otherwise, stay the original sampling mode. Go back to Step 2, starting a new match processing.

Setp 6 If the matched sample length overflows the trigger value *string*>k, then enter into $f_{sampling}=f_{deep}$ mode with a delay. Otherwise, stay in the original sampling mode. Go back to Step 2.

2. FPGA Implementation

Many dictionary-based compression algorithms have been developed specifically for data acquisition or sensor networks [10-12]. Lempel-Ziv-Welch (LZW) [14] compression is the most popular in the field of lossless data compression due to its simplicity. The LZW compression replaces strings of samples with a set of single codes, and improves data redundancy quality to reduce data space, not significantly increasing the computation complexity. Thus LZW is a commonly effective method for dealing with highly redundant data processing.

Fig. 6 illustrates the LZW compressor architecture on FPGA platform. Since FPGA resources are constrained, the dictionary's available size in this platform is limited to *256×(9+9+8)* bits. The dictionary storage memory is constrained so as to satisfy the ECG signal detection application. Because under a sampling rate $f_{sampling}$= 800samples/s, the stable samples segment of the ECG (before the *QRS* complex) includes the code count less than 200 in compression, while the dictionary update cycle is 255 for a *256×(9+9+8)*bits-sized dictionary. Note that the dictionary cannot be filled up before the QRS complex has arrived. This guarantees that the matching operation can work steadily, since the dictionary would not be reset, and these samples can be saved in the same dictionary. All of the f_{low} segment samples can be contained in the same single dictionary until the f_{high} trigger has arrived.

As shown in Fig. 6, the compressor is decomposed into three stages. The first stage generates a dictionary index value. At the beginning, the $index_1$ value and other related parameters are generated based on the hash function. After that, the algorithm moves on to the dictionary matching part. According to the result of the dictionary matching *comp1_comp2* in the current index, the conflict judgment output selection logic to *multiplexer array1*. For instance, when the index conflict does not occur, $index_1$ would be available in *multiplexer array1*. Otherwise, the first stage would output $index_2$ or $index_3$.

The second stage performs dictionary-based string matching in parallel. As illustrated in Fig. 6 *comparator array1* matches the first byte against data of *appendcode (index)*, and *comparator array2* matches the second byte

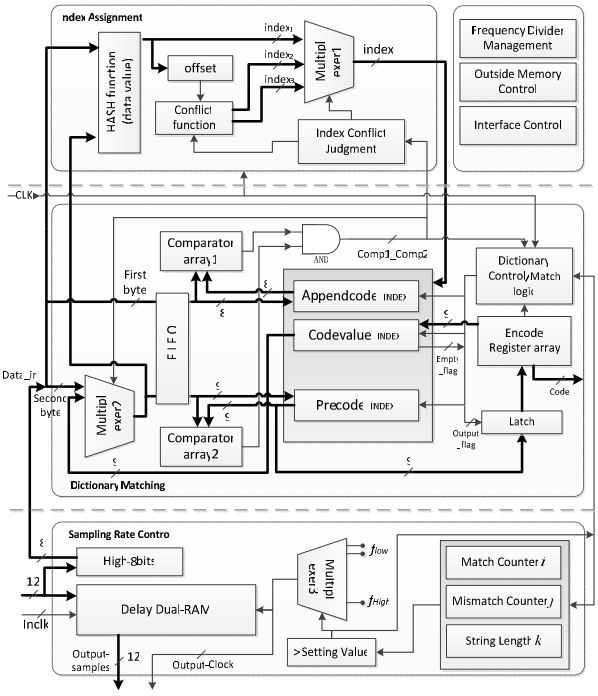

Fig. 6. Compressor architecture on a FPGA platform.

or *codevalue (index)* against data of *percode (index)*. The matching results *comp1_comp2* and *empty_flag* are used to determine whether to push these two data (string) into the dual-port RAM dictionary. If *codevalue (index)* is *0* in the dictionary, then the *empty_flag* is set to *1*, implying the dictionary is empty. Otherwise, according to *comp1_comp2*, the processor can easily distinguish the current compression state between the match pattern and the conflict pattern. Then the compressor decides to generate a new string with the next byte, or go back to the index assignment and count a new index value. During dictionary matching, the *percode(index)* input value is synchronously decided by these two patterns in *multiplexer array2*. Whether the latch is enabled depends on the *output_flag* value. When *output_flag* is *1*, the compressor output *percode* from the encoder register (as shown in Fig. 6).

The last stage of the FPGA implementation is the feedback logic control. In our proposed hardware architecture, according to matching results in the match *Counter i*, mismatch *Counter j* and string length *k*, the sampling rate control block can generate a dynamic frequency based on the different sampling mode (*mode1*, *mode2* and *mode3*). And then samples are outputted from

dual-RAM in real-time. Here, dual-RAM owns two main function one is temporarily stored to samples. Input samples should be temporarily stored in the delay-RAM a priori, and wait until the correct sampling rate is assigned from *multiplexer array3*. This task is common in feedback-sampling-rate designs, which normally ensure signal integrity in trigger and variable-sampling periods, such as the delay buffer application presented in [2]. Other is a delay function that guarantees the *QRS* complex integrity when the f_{high} sampling is triggered. We trigger a high sampling rate for the *QRS* complex after the trigger value. However, we also need to set the same high sampling rate for the boundary samples in the front part of the *QRS* complex which are below the trigger value. Therefore, we need the dual-RAM to temporarily restore these samples in the front part of the *QRS* complex until a higher sampling rate control logic (f_{high}) come.

The implementation was done using VHDL on the Quartus II compiler and simulator. Fig. 7 shows the RTL viewer map on Quartus II, the simulation and the hardware platforms. Our program synthesized platform is implemented on an Altera chip EP1K. For the complexity of the compression block in the FPGA resources, the total memory block utilization is up to 78% of the FPGA storage cells, since a FIFO and double-port RAM dictionary is integrated into the processor, and logic elements occupy 21% of the logic cells.

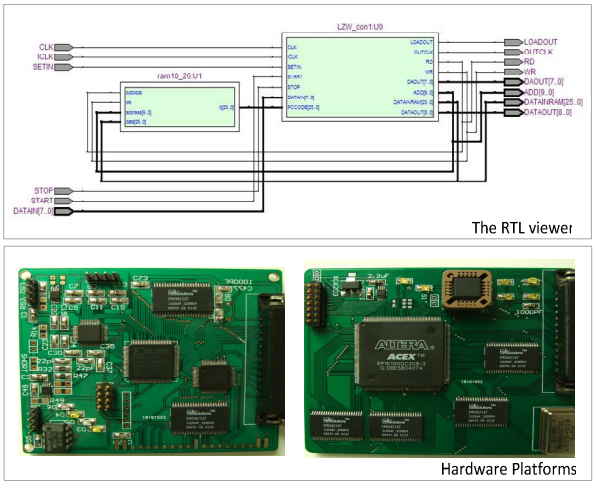

Fig. 7. The RTL viewer technology map of compressor on Quartus II and hardware simulation platforms.

IV. EXPERIMENTAL RESULTS

Fig. 8 shows comparison results of a uniform sampling with our implemented variable sampling rate scheme for an ECG transient signal samples (ECG0328.data and ECG0329.data). According to Fig. 8, we find ECG signal can be integrally reconstructed without any loss based on these three sampling schemes. However, their sampling efficiency are different. More detailed information is illustrated in Table 3. Next, we briefly describe the experimental results.

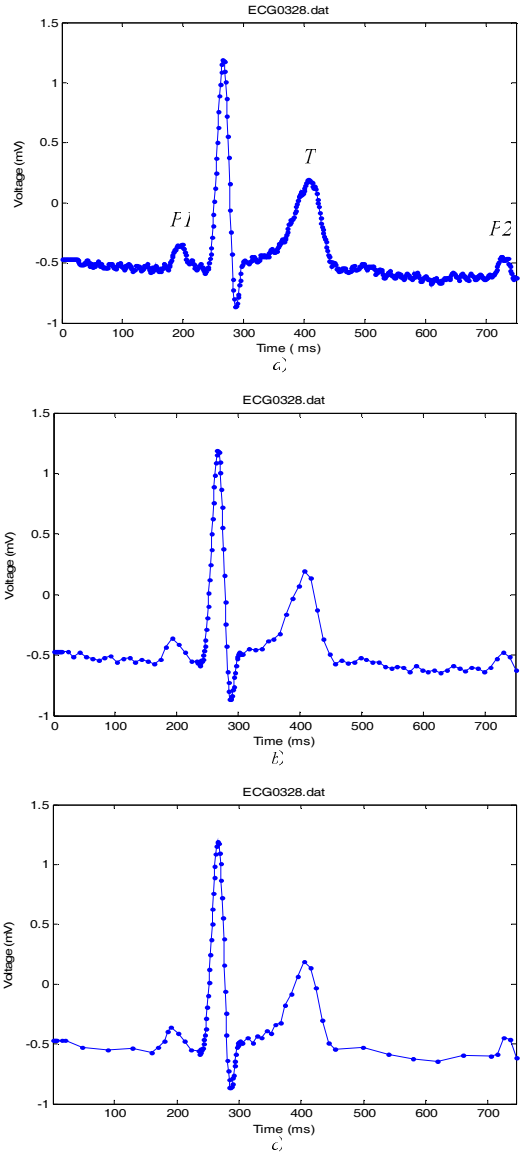

Fig. 8. Comparison of the ECG represented in (a) uniform sampling rate, in, (b) adaptive sampling rate level on f_{high}=800 Hz and f_{low}= 100 Hz, in (c) adaptive sampling rate level on f_{high}= 800 Hz, f_{low}=100 Hz and f_{deep}=25 Hz.

Table 3. Comparison results of the different sampling schemes.

ECG0328/ samples	Uniform sampling i)	Adaptive sampling ii)	Adaptive sampling iii)
QRS complex	42	42	42
*P*1	30	6	7
*P*2	31	6	5
T	82	13	11
Others	415	72	43
Total	600	139	108

Fig. 8(a) is set to the uniform sampling whose fixed sampling rate is 800 samples/s. Based on our algorithm, Fig. 8(b) adopts the non-uniform sampling scheme, and the sampling rate setting as follows: f_{high}=800 samples/s and f_{low}= 100 samples/s without f_{deep}. The scheme in Fig. 8(b) is used as a related reference for traditional adaptive sampling. Performance of the scheme is close to the traditional adaptive sampling application in the ECG signal detection [1-3]. Result of our scheme is shown in Fig. 8(c). The deep adaptive sampling rate is divided into f_{high} = 800 samples/s, f_{low}=100 samples/s and f_{deep} = 25 samples/s. Therefore, we obtain a lower number of samples. Table 3 illustrates the numner of samples in the different segments of the ECG signal, according to the result shown in Fig. 8.

As shown in Table 3, the scheme *c*) significantly decreased the number of samples in the most segment samples, except for the *QRS* complex. The sampling is done without any integrity loss in ECG signal detection. Compared with the uniform sampling scheme in scheme *a*), our proposed strategy reduced a large amount of the samples (more than 80%) in the ECG sensor nodes before performing compression and transmission. For example, if we measure ECG0328 during 1 hour with uniform sampling (without adaptive sampling), system required 5.8 Mbyte hard-merory before the transmission. However, based on our scheme, 1.1 Mbyte hard-merory was required. The memory access space was reduced by 5 times. According to P. Smith [13], Bluetooth's Power per bit is 0.153 uW/bit, and Wifi's Power per bit is 0.00525 uW/bit. Table 4 exhibits the wireless transmit power consumption comparison among these sampling schemes for ECG0328 (Fig. 8) sampling in 10,000ms.

Fig. 9 shows the effectiveness of the proposed adaptive sampling scheme on ECG detection platform. We compared our scheme with an existing adaptive architecture [2] that includes an additional analog circuits

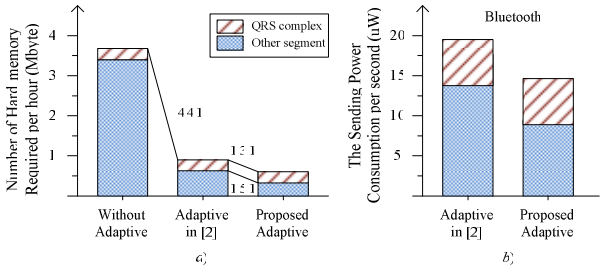

Fig. 9. Performance of the hardware memory and transmission power consumption comparison between [2] suggested technique and our proposed adaptive sampling.

Table 4. The sending power consumption comparison results of the different sampling schemes for ECG0328 in 10 seconds.

Sampling Scheme & power consumption	Uniform sampling *a*) /mW	Adaptive sampling *b*) /mW	Adaptive sampling *c*) /mW
Bluetooth	19.584	4.572	3.452
WIFI	0.672	0.156	0.117

(a differentiator, a comparator, and a multiplexer), and two rate levels f_L and f_H (respectively 64 Hz and 512 Hz). As shown in Fig. 9, we use the same sampling rate levels as [2]. Finally, as Fig. 9 exhibits, this reduction in sampling rate reduces the hard-memory resource and the energy consumption of the transmission.

V. CONCLUSIONS

In this paper, we propsoed an adaptive sampling strategy for ECG measurement, implementing on an FPGA platform. Our algorithm re-utilized the existing data compression logic (real-time strings matching logic) to improve the sampling efficiency. As simulation results indicated, our approach reconstructed the ECG signal at a lower sampling rate with guaranteeing ECG integrity. Our sampling algorithm significantly decreased the sampling rate by 80% compared with a unifrom sampling strategy. Additionally, the hardware memory space and the wireless transmission power were reduced, so that efficiency approximately improved 5 times than the uniform sampling. As a future research, we need to generalize our measurement model to apply to extensive applications.

ACKNOWLEDGMENTS

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (NO. 2013-031352), and the Ministry of Trade, Industry and Energy (MOTIE) through the Special Education program for Industrial Convergence.

REFERENCES

[1] S. Feizi, G. Angelopoulos, V. K. Goyal, and M. Medard, "Energy-Efficient Time-Stampless Adaptive Nonuniform Sampling," *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Process.*, 3809-3812, 2012.

[2] H. J. Kim, C. V. Hoof, and R. F. Yazicioglu, "A Mixed Signal ECG Processing Platform with an Adaptive Sampling ADC for Portable Monitoring Application," *33rd annual international conference of the IEEE EMBS.,* 2196-2199, 2011.

[3] P.Augustyniak, "ECG Sampling Rate Controlled by Signal Contents," *4th Symbiosis*, 128- 131, 2001.

[4] D. Konstantas, A. V. Halteren1, R. Bults1, K.Wac, and R.Herzog, "Mobihealth: Ambulant Patient Monitoring Over Public Wireless Networks," *Mediterranean Conference on Medical and Biological Engineering MEDICO.*, 107-122, 2004.

[5] C. R. Baker, K. Armijo, S. Belka, M. Benhabib, V. Bhargava, and N. Burkhart, "Wireless Sensor Networks for Home Health Care," *21st International Conference on Advanced Information Networking and Applications Workshops*, 832-837, 2007.

[6] Rebizant, Waldemar, Szafran, Janusz, Wiszniewski, and Andrzej, "Digital Signal Processing in Power System Protection and Control," *SBN 978-0-85729-8010*, 2011.

[7] V. Almenar and A. Albiol, "A New Adaptive Scheme for ECG Enhancement," *Signal Processing.,* 253-263, 1999.

[8] L. Chan and C. L. Wang, "VLSI Implementation of Wavelet-based Electrocardiogram Compression and Decompression " *Journal of Medical and Biological Engineering*, 331-338, 2011.

[9] C. Alippi, G. Anastasi, M. D. Francesco, and M. Roveri, "An adaptive sampling algorithm for effective energy management in wireless sensor networks with energy-hungry sensor," *Instrumentation and Measurement.*, 335-344, 2010.

[10] F. Chen, F. Wen, and H. D. Jia, "Algorithm of Data Compression Based on Multiple Principal Component Analysis over the WSN," *6th Wireless Communications Networking and Mobile Computing.,* 1-4, , 2010.

[11] Zhou Yanli, Fan Xiaoping, Liu Shaoqiang, and Xiong Zheyuan, "Improved LZW Algorithm of Lossless Data Compression for WSN," *3rd Computer Science and Information Technology,* 523-527, 2010.

[12] Huan Zhang, Xiaoping Fan, Shaoqiang Liu, and Zhi Zhong, "Design and Realization of Improved LZW Algorithm for Wireless Sensor Networks," *Information Science and Technology Conference.*, 671-675, 2011.

[13] P. Smith, "Comparisons between Low Power Wireless Technologies, Bluetooth low energy, ANT, ANT+, RF4CE, ZigBee, WiFi, Nike+, IrDA and NFC," *HBU Marketing document*, CSR plc, 2011.

[14] D. Salomon, "Data Comparisons," 2nd Edition, *Springer*, New York, 2000.

[15] J. Lim, Y. Cho, and J. Choi, "A 9- bit ADC with a Wide-Range Sample-and-Hold Amplifier," *Journal of Semiconductor Technology and Science.,* Vol.4, 280-285, 2004.

[16] S. Nakaya and Y. Nakamura "Adaptive Sensing of ECG Signals using R-R Interval Prediction," *35th Annual International Conference of the IEEE EMBS.,* 9-12, 2013.

Zhongyun Yuan was born in Shan Xi, China, on 1982. He received the B.S. degree and M.S. degree in the Department of Electronic and Electrical Engineering from North University of China, in 2005 and 2008, respectively. He is currently pursuing the Ph.D. degree in the Department of Electrical and Computer Engineering, Sungkyunkwan University, Suwon, Korea. His interests include compressive sampling and image processing.

Jong Hak Kim received the B.S. degree in radio communication engineering from the Kyunghee University, Suwon, Korea, in 2009, the M.S. degree from the Department of Electrical and Computer Engineering, Sungkyun-kwan University, in 2012, and he is studying for a Ph. D degree at Sungkyunkwan University. He is interested in the efficient low power and real-time processing system for mobile equipment and currently study image processing algorithms and hardware implementation for stereo-system.

Jun Dong Cho received the B.S. degree from the Department of Electronic Engineering, Sungkyunkwan University, Suwon, Korea, in 1980, the M.S. degree from the Department of Computer Science, Polytechnic University Brooklyn, New York, in 1989, and the Ph.D. degree from the Department of Computer Sciences, Northwestern University, Evanston, in 1993, respectively. He was a CAD engineer and Team Leader in Samsung Electronics Company, from 1983 to 1987, and a Senior Technical Staff in Samsung Electronics Company, from 1993 to 1995. In 1995, he joined the Department of Electrical and Computer Engineering, Sungkyunkwan University, Suwon, Korea, where he is currently Professor. His research interests include Low Power Design, 3D Image Processing, System on chip of Software Defined Radio and Multimedia Applications. Prof. Cho is a senior member of IEEE.