Proceedings of the 28th IEEE
EMBS Annual International Conference
New York City, USA, Aug 30-Sept 3, 2006

Fully-Integrated Heart Rate Variability Monitoring System with an Efficient Memory

Xiaoyue Wang, Mingqi Chen, Luca Macchiarulo and Olga Boric-Lubecke
Department of Electrical Engineering, University of Hawaii at Manoa
2540 Dole Street, Honolulu, HI 96822, USA
xiaoyue@hawaii.edu

***Abstract*—Heart rate variability is a strong indicator of a number of medical conditions. Current HRV systems typically determine R-R intervals from pre-recorded ECG signals, which include a large amount of redundant data. In this paper we describe a more efficient HRV monitoring and assessment system on chip. By applying digital techniques to store the difference between every two adjacent R-R intervals in a single-port synchronous, high-performance SRAM, up to 24 hours of continuous ECG data can be stored on chip with a fixed resolution of 1 ms. The system has been tested for functionality, synthesized and laid out in a commercial 0.18 µm CMOS process in a 2.5x2.5 mm^2 hardware core with less than 155 µW power consumption. Such a system can enable HRV monitoring with home based health care and implantable devices.**

***Index Terms*—Digital signal processing, Heart rate variability, ECG data compression, System on chip, and SRAM.**

I. Introduction

The significance of using heart rate variability (HRV) measures as important quantitative indices of cardiovascular control became apparent when they were confirmed as effective diagnostic tools and strong independent predictors of mortality for diseases related to cardiovascular function and regulation in the late 80's [1]-[7]. Measurement of HRV provides a noninvasive method to obtain reliable information on autonomic modulation of heart rate. A compact, low-power, real-time HRV assessment system could provide a valuable feature for implantable and portable cardiac monitoring and intervention devices. A microprocessor HRV monitoring system based on QRS detector for ambulatory monitoring was described in [8], and a system on a chip (SoC) for HRV monitoring and assessment has been described in [9]. The SoC approach applies digital techniques to measure RR intervals from ECG signals, and stores HRV measures in an internal 16x8 register file histogram memory [9].

The HRV SoC design described in [9] has a modest memory capability due to the limitations of 0.5 µm CMOS process that was used for the design, and thus does not allow storage of long-term ECG signals. In addition, even though most R-R intervals exhibit low variations, such redundancy is not exploited to compress the information. Also, the resolution of the 128x8 register file is dependent on the R-R interval lengths, due to truncation of the binary representation. These issues are resolved in a new 0.18 µm SoC design described in this paper.

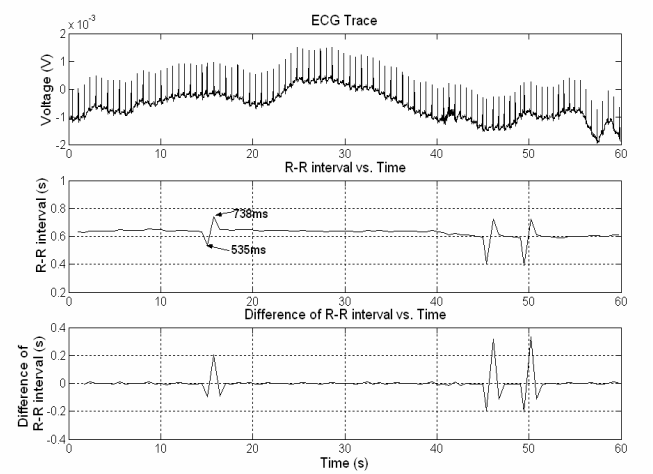

Fig. 1. Pre-recoded ECG trace from Physionet database (top), corresponding R-R intervals (middle) and difference of R-R intervals (bottom).

To illustrate the typical redundancy of HRV data, Fig. 1 shows a 1-minute pre-recorded ECG trace (the top graph) obtained from PTB Diagnostic ECG Database [10], the corresponding R-R intervals (the middle graph) and the difference between every two adjacent R-R intervals (the bottom graph). The bottom graph shows that abrupt variations (called abnormal heart beats in the following) account for less than 10% of the complete dataset, indicating a high degree of redundancy if full R-R intervals are recorded. In this paper, we present an efficient HRV SoC, which stores the difference of every two adjacent intervals instead of full R-R intervals. This difference information is stored in a high-density/high-speed single-ported SRAM of 8196 words x 64 bits, which is designed to take full advantage of an advanced commercial CMOS process and its design libraries. As a result, a much longer, and more clinically relevant, set of intervals can be recorded without increasing the chip size.

The following sections describe the algorithm of the efficient RAM, the SoC design, the chip implementation in a commercial 0.18 µm CMOS technology, the verification of the hardware core functionality using pre-recorded ECG signals shown in Fig. 1 from the PTB Diagnostic ECG Database [10] and its physical synthesis.

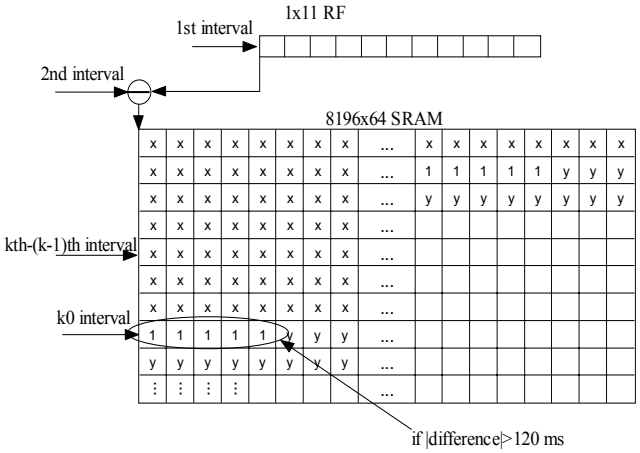

Fig. 2. The illustration of the efficient SRAM algorithm.

II. Efficient Ram Algorithm

Since typical human heart rate ranges from 30 to 150 beats per minute, the beat to beat intervals range from 0.4 s to 2 s. In [9], the R-R intervals are stored in a 128 x 8 register file using a straightforward storing algorithm, thus stored intervals vary from 1.6 ms to 8 ms. Therefore, assuming the worst case of 2 s beat to beat interval, 11 bits are needed when a 1ms resolution is required. In the 128x8 register file described in [9], the last three digits of the intervals are ignored, which greatly reduces the system accuracy.

After taking into account that the differences between every two adjacent R-R intervals rarely exceed ±100 ms as shown in Fig. 1, we can increase the resolution to 1 ms by writing only the differences of the R-R intervals. We separate the R-R intervals into two categories, normal and abnormal ones (as explained above). The differences are written one by one, occupying 8 or 16 digits according to the category of the intervals.

For a resolution of 1 ms, we use an 11-bit binary to represent the heart rate of 30 beats per minute. For the differences of R-R intervals within ±100 ms, we can use an 8-bit binary to store the data. We employ a register to store the first 11-bit interval and a 8196 words x 64 bits single-ported synchronous SRAM to store the differences of every two adjacent R-R intervals. The algorithm flow is described in the following steps (Fig. 2):

1) The first interval is stored by a separate 11- bit register.
2) An 8196x64 SRAM is separated into eight sections. Each section includes 8196 words x 8 bits. The RAM has a write mask capability that allows us to write the data at the selection of 8 bits. Hence we set the write mask word to select one section which is used to store the differences of every two adjacent R-R intervals vertically.
3) As illustrated in Fig. 2, if the absolute value of the difference between the current interval and the previous interval is smaller than 120 ms (1111000 in binary), the 8-bit value representing the difference in sign and magnitude is written into the selected section of the current cell of the RAM.
4) If the absolute value of the difference between the current and the previous interval is larger than or equal to 120 ms, the first five digits of the current cell are set to 1 as a flag. Then the three most significant digits of the current interval are written into the last three digits of the current cell and the remaining eight digits of the current interval are written into the next cell without subtraction. Hence we use two cells to store a value of an abnormal R-R interval.
5) When it comes to the end of the last address of the SRAM, the following section, if one is still available, of the SRAM will be selected by the write enable mask to store the data. At last, when the RAM is full, a corresponding flag is generated.

Fig. 2 illustrates the algorithm flow: SRAM bits indicated with 'x' represent normal intervals, while the two cases with 'y' are examples of abnormal intervals. The distinction between the two types of interval is guaranteed by the fact that no normal interval will ever have its five most significant bits set to 1.

Based on the previous observation, a read unit can follow the following procedure to retrieve the R-R intervals:

1) Read the first interval from the appropriate register.
2) According to the address in sequence, starting from the first cell of the selected section of the 8196 x 64 SRAM, if the left five digits of the first cell of this 8196x8 section are not all ones, take the sum of the first interval and the first cell as the second interval. If the 5-digit flag are all 1, indicating an abnormal interval, the last 3-digit in the current cell and the 8-digit in next cell are combined to generate the next datum.
3) Repeat step 2 until the end of the memory address space.
4) If the section read is not the last one, address the next 8-bit section of the memory and reset the address to 0, going to step 2.
5) Otherwise, issue an end read flag.

This algorithm ensures that the accuracy of stored data is at least 1 ms, even if most data are stored using the same amount of memory employed in [9].

Assuming that around 99% of the R-R intervals are regular intervals, while the other 1% are the abnormal ones (for example as indicated in Fig. 1), we can estimate the upper bound on the memory savings. With 99% of intervals represented with 8 bits and 1% of intervals represented with 16 bits, (99% x 8/11 + 1% x 16/11=73.46%), we estimate 26.54% memory space reduction as compared to writing all intervals with 11 bits.

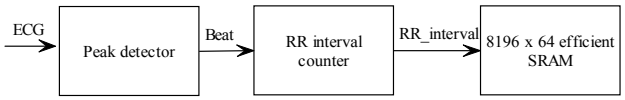

Fig. 3. SoC block diagram.

III. SoC Design

The SoC block diagram is shown in Fig. 3. It consists of the peak detector, R-R interval counter, and the efficient 8196 words x 64 bits SRAM. The peak detector converts the digitized ECG input signals into narrow pulses which identify the R-peaks of the input heart signals. Then the R-R interval counter detects the output pulses from the peak detector, and obtains the R-R intervals for 8196 x 64 SRAM. The efficient SRAM discussed in previous section allows recording of up to 24 hours data of R-R intervals continuously.

A. Peak Detector

The R wave is detected from the ECG signals and converted into narrow square pulses in the three following steps: 1) setup a threshold value to eliminate other peaks by comparing the input data with the threshold value within the small time interval, 2) determine the position of the peak by looking for the maximum value in this interval, and 3) generate a square pulse after obtaining the position of the peak. The threshold value depends on the DC level and amplitude of the input ECG signals and is thus unknown ahead of time. Therefore, the peak detector must initialize an adjustable threshold to accommodate variations in the input signals. The details of the peak detector design are discussed in [9].

B. R-R Interval Counter

The RR-interval counter receives the beats coming from the peak detector and outputs an 11-bit data which counts the number of clock periods between every two adjacent R-peaks for 8196 x 64 SRAM. The 11-bit R-R interval counter can detect the RR-intervals up to 2047 clock cycles, which is two seconds of range for R-R intervals.

C. Efficient 8196x64 SRAM

The single-port synchronous RAM is a fully static memory, which has four pins, namely, write enable (WEN), chip enable (CEN), data in (D) and data out (Q).

The finite state machine block diagrams implemented in VHDL, illustrate the efficient SRAM writing and reading algorithms in Fig. 4 (a) and Fig. 4 (b).

In Fig. 4 (a) which illustrates the state transfer of writing status, the state s0w initializes all signals. The next two states, s1w and s1w2, wait for the coming two consecutive R-R intervals. The difference of the two R-R intervals is calculated at s2w, the state s3w prepares control signals for writing the differences to RAM. The next two states, s4w1 and s4w2, set the signal WEN and the address for writing.

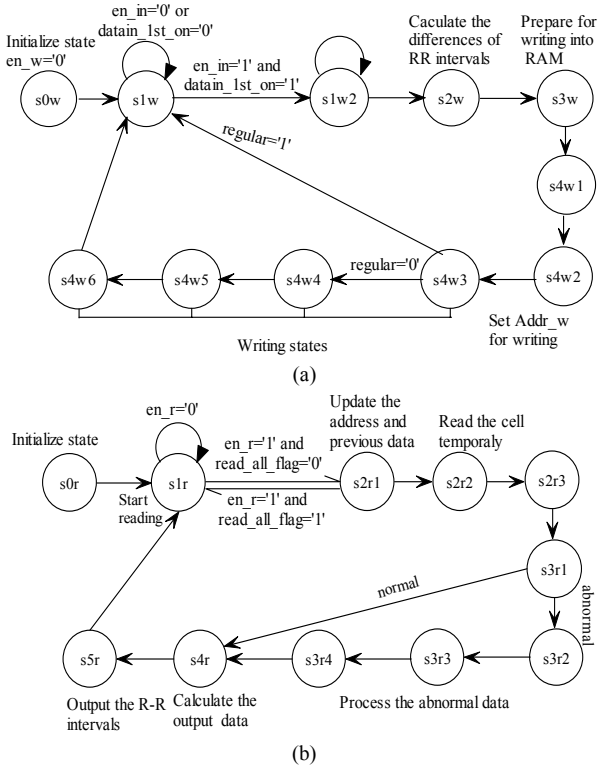

Fig. 4. Finite state machine of RAM (a) writing (b) reading protocol.

When the R-R interval is a normal one, the state s4w3 goes back to s1w; if the interval is an abnormal one, the state will transfer to s4w4, together with s4w5, s4w6, which are three writing states to store the complete abnormal R-R intervals associated with flags. Finally s4w6 will transfer back to s1w state.

In Fig. 4 (b), illustrating the reading protocol, s0r is also the initializing state. From s1r, it starts retrieving the R-R intervals from the RAM. S2r, s2r2 and s2r3 will update the address when it finishes reading one cell, then the state moves on to s3r1 which is used to judge if the value of the current cell is normal or abnormal. If it is a normal value, s3r1 will jump to s4r to calculate the output data, then goes to next state s5r to output the retrieved R-R interval; if it is abnormal value, s3r1 state will keep on processing the data through s3r2, s3r3 and s3r4, and then moves on to s4r and s5r. Finally it goes back to s1r to complete one cycle.

IV. Verification and Layout of the SoC

The hardware core whose design is detailed in the previous sections has been described in synthesizable VHDL targeted for an implementation using a commercial 0.18 μm technology and its associated memory libraries. After a pre-synthesis verification of the functionality, we proceeded with a synthesis using Synopsys' Design Compiler (TM). The cell and memory occupation after synthesis gives an overall area of 2.5x2.5 mm^2 for the entire system.

The synthesis results show that the standard cell logic

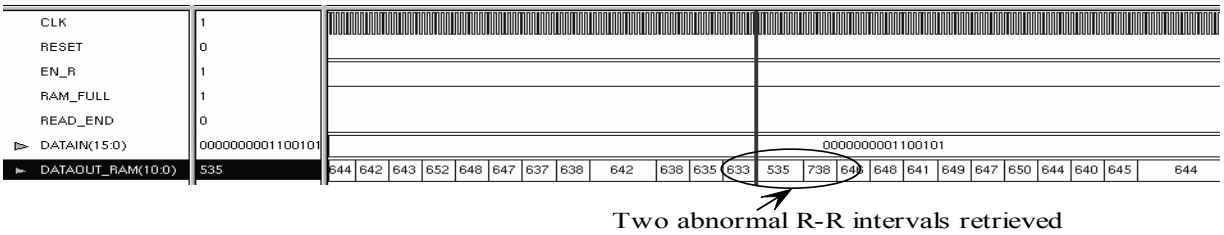

Fig. 5. VHDL post-synthesis simulation.

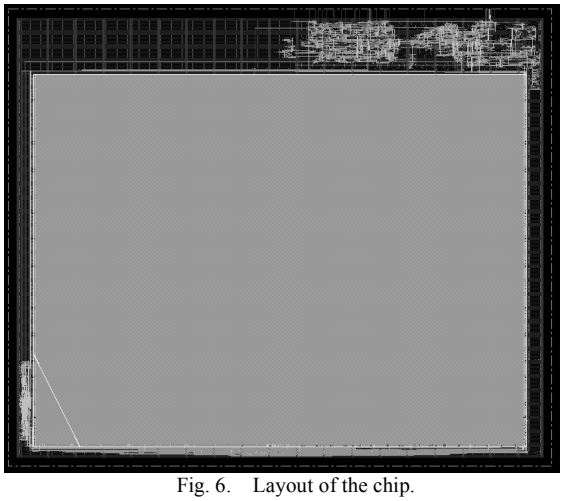

Fig. 6. Layout of the chip.

used in the design is a small fraction of the overall area, which is mainly occupied by the memory. This indicates that there is plenty of space for incorporating more functionality (such as histogram classification, triangular index, standard deviation of R-R intervals, and possibly spectral measures) in a single chip. As expected, the timing constraints are not significant, and this implies a light clock tree with low power dissipation. The dynamic power dissipation reported after synthesis (without the clock tree, but see the previous observation) is 5.5 nW, which is much smaller than the cell leakage power of 155 µW.

Fig. 5 shows the output of VHDL post-synthesis simulation of digital HRV chip which illustrates that R-R intervals are retrieved successfully. A pre-recorder ECG trace shown in Fig. 1 from Physionet database [10] was used for this test. The two output intervals 535 ms, 738 ms which are highlighted in both Fig. 1 and Fig. 5 are abnormal R-R intervals. The output retrieved R-R intervals are exactly the same with the input intervals, which indicates that the efficient algorithm for SRAM enables us to obtain all R-R intervals regardless of their classification.

With the result from the synthesis, physical design of the chip has been completed using Cadence SoC Encounter (TM). The final result, that includes power and clock routing, is shown in Fig. 6. It is apparent that the standard cell area is underutilized, even though the ratio between standard cell and memory block has been kept to a minimum, further emphasizing the availability of silicon area to incorporate more complex functions. The overall laid out core has dimensions of 2.5 mm by 2.5 mm.

V. Conclusion

We have demonstrated a digital heart rate variability monitoring system on chip with an efficient RAM in a commercial 0.18µm process using VHDL, with a power consumption of less than 155 µW. We have presented the algorithm of the efficient RAM, the design of the system on chip, verification and the layout. The RAM employed an efficient algorithm to store the differences between every two adjacent R-R intervals, thus saving memory space. The SoC has the capability to store 24 hours of continuous ECG data with a fixed resolution of 1ms. The chip has a core area of 2.5x2.5 mm^2. Such a system could be used with implantable devices or home based monitoring systems, such as pacemakers and Holter ECG monitors.

Acknowledgement

The authors would like to thank Wansuree Massagram of the University of Hawaii at Manoa for useful discussions.

References

[1] M. Malik, A. Camm and Members of the Task Force, "Heart rate variability: standards of measurement, physiological interpretation, and clinical use," Circulation, vol. 93, pp.1043-1065, 1996.

[2] R. Kitney, S. Byrne, M. Edmonds, R. Watkings and V. C. Roberts, "Heart rate variability in the assessment of autonomic diabetic neuropathy," *Automedica*, vol. 4 pp. 155-167, 1982.

[3] R. Freeman, J. P. Saul, M. Roberts, R. Berger, C. Broadbridge, and R. Cohen, "Spectral analysis of heart rate in diabetic neuropathy," *Arch Neurul.*, vol. 48 pp. 185-190, 1991.

[4] J. Bigger, J. Fleiss, R. Steinman, L. Rolnitzky, R. Kleiger and J. Rottman, "Frequency domain measures of heart period variability and mortality after myocardial infarction," *Circulation*, vol. 85 pp. 164-171, 1992.

[5] M. Malik, T. Farrell, T. Cripps and A. Camm, "Heart rate variability in relation to prognosis after myocardial infraction: selection of optimal processing techniques," *Eur. Heart J.*, vol. 10, pp. 1060-1074, 1989.

[6] M. Pagani, G. Malfatto, S. Pierini, R. Casati, A. Masu, M. Poli, S. Guzzetti, F. Lombardi, S. Cerutti and A. Malliani, "Spectral analysis of heart rate variability in the assessment of autonomic diabetic neuropathy," *J.Auton. Nerv. Syst.*, vol. 23 pp. 143-153, 1988.

[7] R. Barbieri, E. C. Matten, and E. N. Brown, "Instantaneous monitoring of heart rate variability," *Proc. 25th Annu. Int. Conf. IEEE EMBS*, vol. 1, 17-21, Sep. 2003

[8] A. Ruha, S. Sallinen, and S. Nissilä, "A Real-Time Microprocessor QRS Detector System with a 1-ms Timing Accuracy for the Measurement of Ambulatory HRV," *IEEE Trans. Biomed. Eng.*, vol. 44. no. 3, pp. 159-167, Mar. 1997.

[9] W. Massagram, O. Boric-Lubecke, L. Macchiarulo, and M. Chen, "Heart Rate Variability Monitoring and Assessment System on Chip," *Proc. 27th Annu. Int. Conf. of the IEEE EMBS, Shanghai, China,* Sep. 2005.

[10] http://www.physionet.org/physiobank/database/ptbdb/