An FPGA Implementation of a Lossless Electrocardiogram Compressor based on Prediction and Golomb-Rice Coding for Telemedicine Applications

Moab Mariz Meira, José Antônio Gomes de Lima and Leonardo Vidal Batista

***Resumo*—Aplicações de telemedicina estão passando por um impressionante desenvolvimento nos últimos anos devido aos avanços em telecomunicações. Este trabalho descreve uma implementação em FPGA de um compressor sem perdas de eletrocardiogramas baseado em predição e codificação de Golomb-Rice, as funcionalidades dos blocos principais da arquitetura do compressor estão apresentadas.**

***Palavras-Chave*—Eletrocardiograma, telemedicina, FPGA, compressão de dados, codificação de Golomb-Rice.**

***Abstract*—Telemedicine applications are experiencing an impressive development in the last years due to advances in telecommunications. This work describes an FPGA implementation of a lossless electrocardiogram compressor based on prediction and Golomb-Rice coding, the functionalities of the compressor main blocks, the simulation and systhesis methodology are presented.**

***Index Terms*—Electrocardiogram, telemedicine, FPGA, data compression, Golomb-Rice coding.**

I. INTRODUCTION

Telemedicine uses telecommunication technologies to exchange relevant medical information among geographically separated institutions, physicians and patients. Telemedicine can be used in many fundamental stages, such as diagnosis, therapy and follow-up, thus addressing the access, quality, and cost problems of modern health care services. In the store-and-forward modality of telemedicine, patient data are captured, stored and sent to a specialist who asynchronously analyzes the registers. In the real-time modality, the specialist immediately receives the data during the capture process, and in many cases simultaneously communicates with the patient [1, 2].

Telemedicine has been intensively applied in Cardiology, with the implantation of systems to remotely monitor the cardiac activity of patients at home or even during ambulance transportation [3]. The electrocardiogram (ECG) is the most used physiological signal for noninvasive investigation of the cardiovascular activity, and the extensive use of digital ECG produces large amounts of data. Since it is often necessary in telemedicine applications to store or transmit ECG records, efficient compression techniques are important to reduce transmission time or required storage capacity. In fact, efficient data compression techniques, combined with advances in computing, data storage and high-speed communication systems open new horizons for remote medical sensing [2, 3].

Especially critical are long duration (24 or even 48 hours) *Holter* exams. The data generated in one Holter exam can surpass 1G bytes. Holter devices must have good storage capacity, in addition to reduced dimensions and low power dissipation in order to be comfortably carried by patients [4]. With the advent of low-cost mobile communication networks, the popularization of similar devices with real-time telemonitoring capabilities is expected. In the future, such portable devices could continuously transmit medical telemetry to monitoring systems and to health professionals [2].

Several ECG compression methods have been developed in the past 30 years, and average compression ratios (CR) ranging approximately from 2:1 up to 50:1 have been reported [5, 6, 7]. CR is defined as the ratio between the number of bits in the original signal and the number of bits used to represent the compressed signal.

Most of the reported techniques are *lossy compressors*, and therefore do not allow perfect reconstruction of the original signal from its compressed representation. In many cases, however, lossless compression is largely preferred by health professionals [8], due to the fear that the distortions introduced by lossy compression may induce misdiagnoses.

A recent study investigated more than 30 lossless compression methods resulting from combinations of three predictors, five integer-to-integer wavelet transforms, and Huffman, Golomb, LZW and LZ78 coders [9]. A very simple compressor based on first-order linear prediction and Golomb coding reached a competitive CR in comparison with the other evaluated methods [9]. The CR achieved by this simple

Moab Mariz Meira, José Antônio Gomes de Lima and Leonardo Vidal Batista, Departamento de Informática, Universidade Federal da Paraíba, João Pessoa, PB, Brasil, E-mail: marizmeira@bol.com.br. This work was supported by CNPq, a governmental Brazilian institution dedicated to scientific and technological development.

compressor was 58% greater than the CR achieved by the general-purpose compression utility WinZip® 7.0 running in maximum compression mode. Moreover, Golomb-Rice coding, which is a simplification of Golomb coding more suitable to hardware implementation, performs close to Golomb coding when used in combination with prediction techniques [10].

This paper presents an FPGA implementation of a lossless ECG compressor based on prediction and Golomb-Rice coding. Section II presents the compression model adopted in the present work, along with some fundamental concepts; section III describes in detail the architecture of the compressor and decompressor; section IV presents the simulations; section V shows synthesis results in terms of FPGA devices; the conclusions are presented in section VI.

II. COMPRESSION MODEL AND FUNDAMENTAL CONCEPTS

Figure 1 presents the adopted compression model. Initially, the original signal $\mathbf{x}_o$ is processed by a decorrelation stage, whose objective is to reduce the inter-sample correlation. The coding stage tries to code the decorrelated signal $\mathbf{x}_d$ using a reduced number of bits. The result of the coding is the binary sequence $\mathbf{x}_c$.

Entropy coders depend on a statistical model describing the sequence to be coded. When samples are strongly correlated, high compression rates can be achieved if the coder is based on a contextual model that appropriately captures the correlations.

However, there are theoretical and practical problems associated with the use of contextual models, related to memory requirements and estimation of conditional probabilities distributions [11]. If the decorrelation stage substantially reduces the inter-sample correlations, simple non-contextual models can lead to a good coding performance.

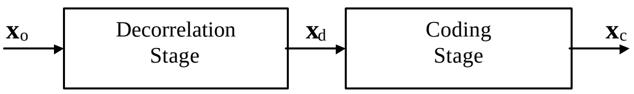

Fig. 1. Generic compression model

A. *Prediction*

The decorrelation stage used in the proposed compression is based on prediction. Prediction techniques defines an estimate $\tilde{x}_n$ for a given sample $\boldsymbol{x_n}$ of the signal using past samples $x_{n-1}x_{n-2}x_{n-3...}$. The sample $\boldsymbol{x_n}$ is substituted by the prediction error, $e_n = x_n - \tilde{x}_n$. The first-order prediction adopted in this work is simply:

$$\tilde{x}_n = x_{n-1} \qquad (1)$$

B. *Golomb-Rice Coding*

Golomb-Rice coding provides a simple way to code any non-negative integer n [10]. Given a non-negative integer parameter k, the Golomb-Rice code for n is the unary code for n shifted k bits to the right, concatenated with the k less significant bits of n. For one-sided geometric distributions (OSGD), Golomb-Rice codes generally achieve nearly optimal bit rates [10].

It is generally accepted that prediction errors are well modeled by Laplacian distributions or, in the case of integer values, by two-sided geometric distributions (TSGD) [12]. A TSGD can be transformed in an approximation to an OSGD by the following mapping of values [12]:

$$M(v) = \begin{cases} 2v & v \geq 0 \\ 2|v| - 1 & v < 0 \end{cases} \qquad (2)$$

C. *Calculation of the parameter k*

Given a sequence whose elements follow a Laplacian distribution, the optimum Golomb-Rice parameter k to encode the sequence obtained after the mapping in (2) can be estimated as $k = \lceil \log_2 a' \rceil$, where a' is the average of the absolute values of the sequence before mapping [12].

In the proposed method, parameter k is incrementally calculated as shown in Equation 3 [12].

$$n \cdot 2^k \geq a, \qquad (3)$$

where n is the number of samples seen so far, and a is the sum of the absolute values of these samples.

III. COMPRESSOR ARCHITECTURE IMPLEMENTED IN FPGA

The compressor is composed by two main blocks, the PREDICTOR block and the CODER block, as shown in Figure 2. A 16 bits sample is read by the PREDICTOR block, which computes the prediction error. The CODER block performs the Golomb-Rice coding and generates the codeword, bit by bit, on the "code" output.

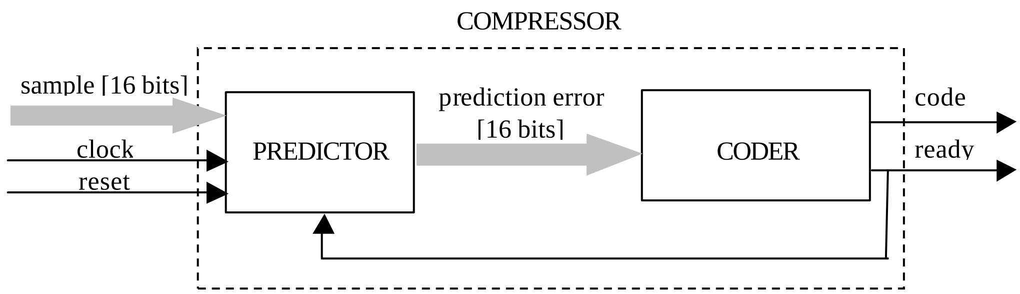

Fig. 2. Compressor Architecture

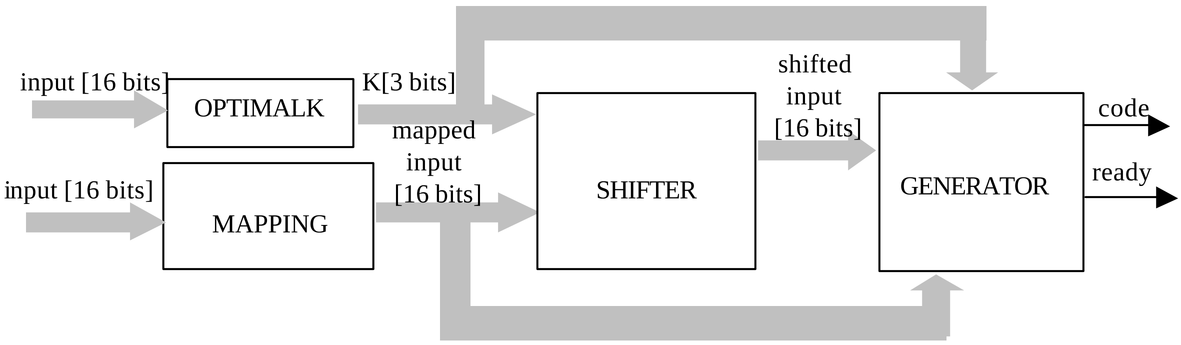

Fig. 3. CODER block

As seen in Figure 3, the following blocks compose the CODER: OPTIMALK block, MAPPING block, SHIFTER block and GENERATOR block.

The OPTIMALK block incrementally calculates the optimal *k* value according to Equation 3; the MAPPING block maps the prediction error according to the Equation 2; the SHIFTER block shifts the mapped prediction error, according to the current *k* value. The GENERATOR block performs the unary coding of the data provided by the SHIFTER block and sends it serially to the output, followed by the *k* less significative bits of the mapped prediction error, generating the Golomb-Rice codeword. The GENERATOR block synchronize the PREDICTOR block to accept a new sample data.

IV. Decompressor Architecture Implemented in FPGA

The decompressor is composed of two main blocks, the IPREDICTOR block and the DECODER block, as shown in Figure 4. The DECODER block performs the decoding stage from an incoming coded input. The IPREDICTOR block obtains the original samples from the prediction errors calculated by the PREDICTOR block.

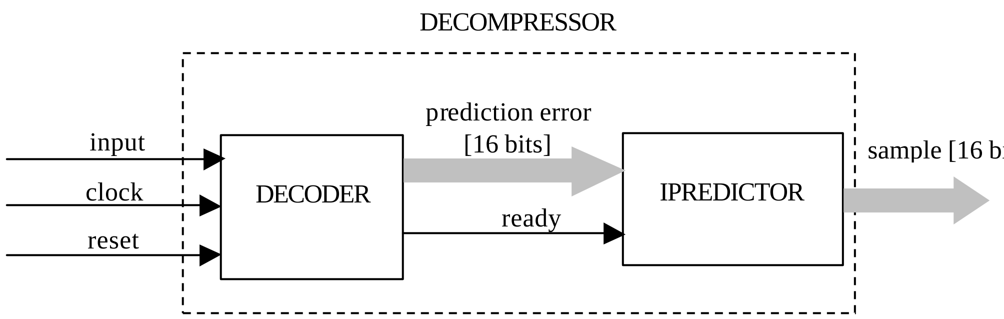

Fig. 4. Architecture of the Decompressor

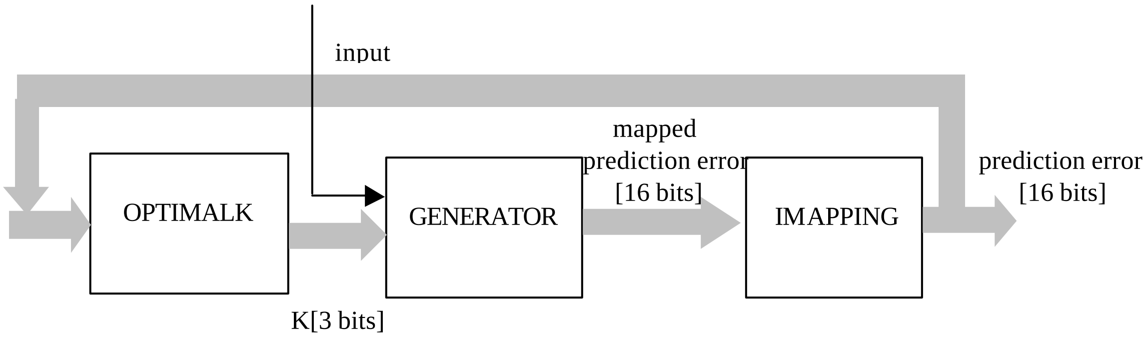

Fig. 5. DECODER block

Figure 4 shows that the DECODER block decodes an input (codeword), reconstructs the prediction error, and sends it to the IPREDICTOR block, which reconstructs the original sample in the decompressor output.

As shown in Figure 5, the DECODER is composed of the following blocks: OPTIMALK block, IMAPPING block and GENERATOR block.

The OPTIMALK block incrementally calculates the optimal *k* according to the Equation 3. The IMAPPING block obtains the original prediction error from its mapped representation. The GENERATOR block performs the Golomb-Rice decoding.

V. SYNTHESIS AND SIMULATION METHODOLOGY

The compressor and the decompressor were described in VHDL and their general architectures were divided in blocks as shown in Figure 2 and Figure 4. The blocks were divided in sub-blocks. At sub-block level, behavioral descriptions were made and, at block level, structural descriptions connecting the sub-blocks were constructed and validated using the MAX+plus II 10.0, software from Altera[13, 14].

Figure 6 presents a simulation of a sample coded by the compressor. Initially the input is processed by the prediction block and the prediction error is calculated (A). Since it is the first sample, the predecessor sample is set to zero. Then the value of the sample is mapped (B), and a new *k* is calculated (C). After the mapping, the sample is shifted *k* bits to the right (D) and then the *ready* signal goes up, indicating the beginning of the coding, and each bit of the code is sent to the output in the rising edge of the clock (E). Finally, the coding ends and another input is read (F). The new input is processed by the predictor using the previous sample and the process goes on (G).

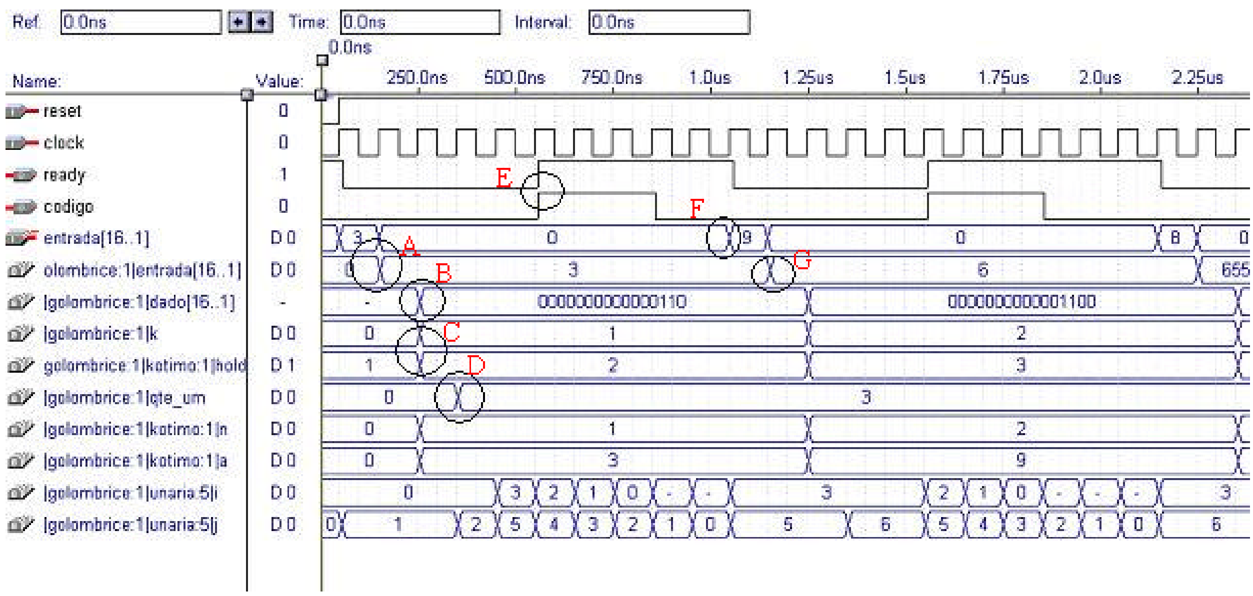

Fig. 6. Simulation of the coding of a sample

Figure 7 presents a simulation of an input decoded by the decompressor.

In the beginning of the decoding of the input, bits are read one by one in the rising edge of the clock (A), and a mapped prediction error is generated (B). Then, the prediction error is obtained from its mapped representation and the *ready* signal goes up indicating to the next block that the prediction error is ready to be processed (C). After the activation of the *ready* signal, the IPREDICTOR reconstructs the original sample. Since it is the first sample, the predecessor sample is set to zero. The decoded sample is used to calculate a new k (D) and then a second codeword is decoded and the process goes on (E).

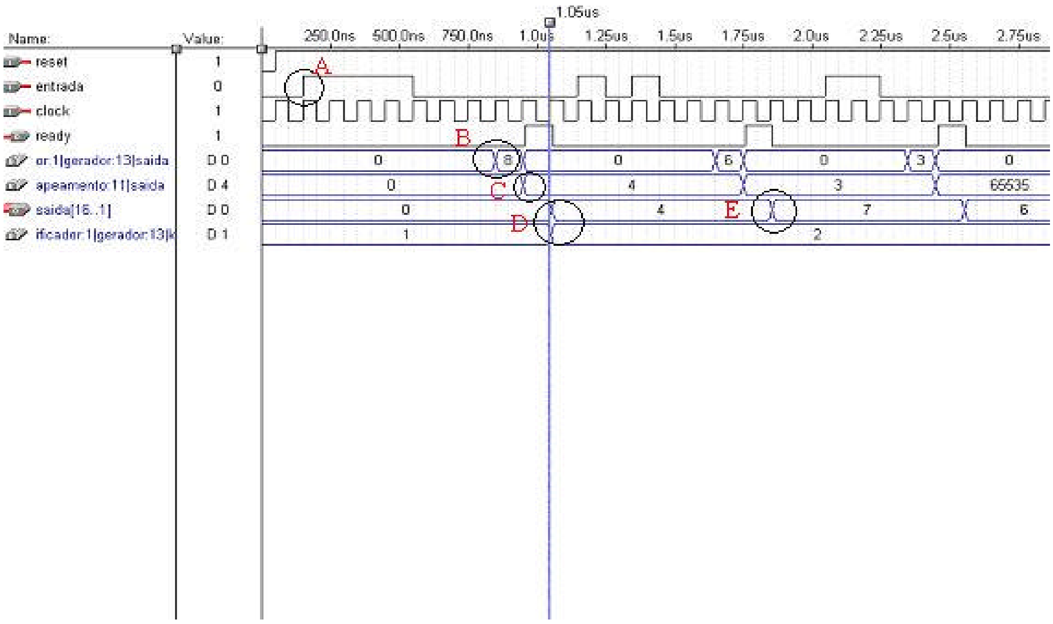

Fig. 7. Simulation of the decoding of a sample

VI. Synthesis Results

The descriptions of the compressor and decompressor were synthesized using Alteras' tools [13, 14]. Results in terms of number of logic cells (LC) memory utilized, frequency of operation and device used are showed in Table I (compressor) and Table II (decompressor).

TABLE I
COMPRESSOR SYNTHESIS RESULTS

Device	Number of LC	Percent of the device used	Memory utilized	Clock Frequency (MHz)
EPF10K30ETC144-1	791	45 %	-	21.45
EPF10K20TC144-3	791	68%	-	10.23
EP1K30TC144-1	793	45%	-	19.68

TABLE II
DECOMPRESSOR SYNTHESIS RESULTS

Device	Number of LC	Percent of the device used	Memory utilized	Clock Frequency (MHz)
EPF10K30ETC144-1	681	39 %	-	39.68
EPF10K20TC144-3	681	59%	-	18.08
EP1K30TC144-1	681	39%	-	37.03

VII. CONCLUSIONS AND FUTURE WORKS

The design of an FPGA implementation of a lossless ECG compressor based on prediction and Golomb -Rice coding has been described. The results show that all the compressor/decompressor functions can be implemented in a single and simple FPGA chip. The clock frequencies presented in Tables I and II shows that a real-time implementation of the method is possible, and the reduced number of logic cells indicates that the compressor/decompressor can be embedded in portable telemedicine devices.

The compressor was designed for ECG compression, and tested with ECG records from MIT-BIH Arrhythmia Database [15]. However, it could be used to compress various other one-dimensional signals similar to ECG (less than 12 bits per sample and high inter-sample correlation). This includes many biological signals, such as electroencephalogram and blood pressure signals, and those typically registered in multi-modal records such as polysomnograms.

New FPGA families allows the implementation of more complex cores with significantly improved performance, thus encouraging further researches to optimize this project design adding new functionalities for telemedicine devices, such as error-correcting capabilities.

REFERENCES

[1] J.J. McKee, N.E. Evans, and F.J. Owens, "Digital Transmission of 12-Lead Electrocardiograms and Duplex Speech in the Telephone Bandwidth," *Journal of Telemedicine and Telecare*, Vol.2, No.1, 1996, pp.42-49.

[2] Ari T. Adler. A Cost-Effective Portable Telemedicine Kit for Use in Developing Countries. *Master thesis*. Massachusetts Institute of Technology, May 2000.

[3] José García, Ignacio Martínez, Leif Sörnmo, Salvador Olmos, Angel Mur, and Pablo Laguna. "Remote Processing Server for ECG-Based Clinical Diagnosis Support." *IEEE Transactions On Information Technology In Biomedicine*, VOL. 6, NO. 4, DECEMBER 2002, p. 277-284.

[4] Antoniol, G.; and Tonella, P. "EEG Data Compression Techniques." *IEEE Transactions on Biomedical Engineering*, v. 44, n. 2, p. 105-114, 1997.

[5] Lee, H.; and Buckley, K. M. "ECG Data Compression Using Cut and Align Beats Approach and 2-D Transforms." *IEEE Transactions on Biomedical Engineering*, v. 46, n. 5, p. 556-564, March 1999.

[6] Batista, L. V.; Melcher, E. U. K.; and Carvalho, L. C. "Compression of ECG Signals by Optimized Quantization of Discrete Cosine Transform Coefficients." *Medical Engineering & Physics*, v. 23, n. 2, p. 127-134, 2001.

[7] Jalaleddine, S. M. S; Hutchens, C. G.; Strattan, R. D.; and Coberly, W. A. "ECG Data Compression Techniques - A Unified Approach." *IEEE Transactions on Biomedical Engineering*, v. 37, n. 4, p. 329-343, 1990.

[8] Bilgin, A.; Zweig, G.; and Marcellin, M. W. "Efficient Lossless Coding of Medical Image Volumes Using Reversible Integer Wavelet Transforms." *Proc. of 1998 Data Compression Conference*, p. 428-437, Snowbird, Utah, March 1998.

[9] Batista, L.V.; Meira, M.M.; Patrício, F.Z.A.; Carvalho, L.C.; e lima, J.A.G. "Compressão sem Perdas de Sinais Eletrocardiográficos" *Workshop de Informática Médica, 2003*

[10] Seroussi, G.; and Weinberg, M.J. "On Adaptive Strategies for an Extended Family of Golomb-type Codes." *Proceedings of the Data Compression Conference*, Snowbird, p.131-140, March, 1997.

[11] Bell, T. C.; Cleary, J. G.; and Witten, I. H. *Text Compression*, Englewood Cliffs:Prentice Hall, 1990.

[12] Weinberger, M. J.; Seroussi, G.; and Sapiro, G. LOCO-I: "A Low Complexity, Context-Based, Lossless Image Compression Algorithm." *Proceedings of the IEEE Data Compression Conference*, Snowbird, p. 140-149, March, 1996.

[13] Altera Corporation. "Altera Data Book", 1995

[14] Altera Corporation. "Data Book and Max + Plus II Getting Started", 1997

[15] Moody, G.B.; and Mark, R.G. *MIT-BIH Arrhythmia Database Directory*. Second Edition, BMEC TR010(revised), Massachusetts Institute of Technology, Biomedical Engineering Center, August 1988.