A DSP BASED PROCESSING SYSTEM FOR ELECTROCARDIOGRAPHIC SIGNALS

Francisc Iacob, Cornel Popescu
Computer Department, “Politehnica” University of Bucharest, Romania
e-mail: iacob@cs.pub.ro

Abstract. The paper presents an electrocardiographic processing system based on a DSP module connected to a PC microcomputer. It describes some important aspects concerning the digital signal processing, especially the ECG waves. The general structure and the operating mode of a development system based on the floating point DSP96002 from Motorola, connected to an IBM PC computer, is presented in this paper. The processing program comprises a powerful algorithm for the electrocardiogram wave detection, analysis and diagnosis. The paper describes an efficient algorithm for the ECG waves recognition. Also, the program uses the Fast Fourier Transform in the ECG signal processing. **Key words**: ECG processing, pattern recognition, FFT, DSP.

ANALOG SIGNAL PROCESSING

In the real world all phenomena are analog (sound, light, heat, electricity, magnetism). In order to process them with a computer, the measuring system uses transducers to convert the analog levels into electrical voltage signals. Then an analog-to-digital converter (ADC) converts these electrical signals into digital values what are processed with DSP, which executes specific algorithms. The numerical results may be displayed on the computer screen in various forms or some digital results may be converted with digital-to-analog converter (DAC) in electrical signals and may control the real world phenomena (for example a technologic process) using special devices.

The first step in the analog signal processing is the selection of a set of discrete points in the continuum of real world phenomena . This process is called sampling. The sampling rate is determined by the amount of signal information that is needed for processing the signals adequately for a given application. Let d(t) the digital representation of an analog signal f(t), consisting of unquantized pulses at intervals of T, so that:

$$\begin{aligned} d(kT) &= f(kT) && k = 0,1,2,\ldots,\infty \\ d(t) &= 0 && if\ t \neq kT \end{aligned}$$

It can be shown that the Fourier transform D(w) of d(t) is related to F(w) the Fourier transform of f(t) by the equation:

$$D(w) = \sum_{n=-\infty}^{\infty} F(w - \frac{2\pi n}{T})$$

In other words, the Fourier transform of d(t) consists of a summation of shifts of the Fourier transform of f(t).

Let w_{max} be the largest value of w for which F(w) is nonzero. The Shannon’s sampling theorem says that f(t) can be reconstructed exactly from the samples if the time between samples is less than:

$$T < \frac{\pi}{w_{\max}}$$

i.e. the sampling frequency is at least twice the size of w_{max}.

The sampled analog signal is represented in the computer by a finite number of bits (quantization). This process is equivalent with the mapping from the real numbers into a range of integers.

For real time processing of signals the DSP must perform all processing tasks required by the current sample in the time between two successive samples. Processing speed is influenced by algorithm complexity. Complex algorithms need more processing tasks and require faster processing because the time between samples is fixed. The minimum required DSP speed may be calculated with the following relation:

$$S = \frac{N}{T} \qquad (in\ MOPS)$$

where S is the DSP speed, N is the total number of processing operations between two successive samples and T is the time interval between two samples.

The analyzing system for recognition of the ECG waves is based on a set of rules concerning the definitions of the basic parameters. Here are presented the most important of them. The basic mark of an ECG wave is the QRS complex. The earliest QRS deflection which is above the reference level is labeled R. Any downward deflection which precedes R is labeled Q. When R is absent and the QRS complex consists of a single downward deflection, this deflection should be labeled QS. In statistical studies QS, Q and S deflections should be considered separately.

The voltage of an upward (QRS) deflection (R wave, R’, R”, etc.) is determined by measuring the vertical distance between the upper edge of the trace at the beginning of the QRS interval and the upper edge of the trace at the peak of the deflection.

The S-T junction or J is indicating the point or shoulder which marks the end of the QRS complex, when steep slopes of the QRS deflections are more or less abruptly replaced by the more gradual slopes which precede or comprise the first limb of the T wave. The reference level for the measurement of the displacement of the S-T junction (or J) is the P-R segment at the beginning of QRS.

The measurements of the ECG intervals are very important in the pattern recognition process. The P-R interval is measured from the beginning of the P wave to the beginning of the QRS whether this is represented by a Q wave or an R wave. The longest interval found in the bipolar or unipolar extremity leads is regarded as the P-R interval. The QRS interval is measured from the beginning (Q or R) to the end of the QRS group of deflections. The longest interval found in the unipolar or bipolar extremity leads is regarded as most nearly correct. The basic parameters of an ECG wave are presented in Figure 1.

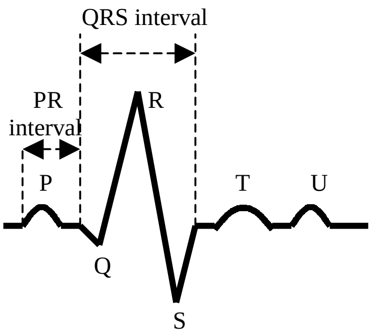

Figure 1. Basic parameters of an ECG wave.

DSP DEVELOPMENT SYSTEM

DSPs (Digital Signal Processors) are special processors with hardware and software capabilities for high-speed numeric processing applications of data representing analog signals, providing all the advantages associated with numerical processing: speed, accuracy, increased noise immunity, greater dynamic range, flexibility and programmability.

DSP chips can be divided into four categories: building-block circuits, application-specific integrated circuits (ASICs), algorithm specific integrated circuits and general purpose DSPs.

The first category provides multipliers/accumulators, address generators and sequencers, offering advantages in performance and flexibility.

The second category (DSP ASICs) includes gate arrays and cell-based circuits, providing performance and flexibility, but in addition minimum board space and low power dissipation.

The third category is represented by algorithm specific DSPs that execute specific algorithms (image processing algorithms, transforms, finite impulse response (FIR) or infinite impulse response (IIR) filters).

The general purpose DSPs represent the fourth category. They have general processing features including data arithmetic logic units for integers and floating point numbers, address generation units, on chip RAM and ROM, data and address buses and I/O ports. These DSPs may be used in a large array of applications with low costs.

The development system proposed in this paper is based on DSP96002 from Motorola. It is a 32-bit IEEE floating point general purpose digital signal processor. At 33MHz clock it can execute 16.5 million instructions per second (Mips) and 49.5 million floating point instructions per second (MFLOPS).

The major components of the DSP 96002 processor are data buses, address buses, data ALU, address generation unit, X data memory, Y data memory, program control and system stack, program memory, port A and port B external bus interfaces, internal bus switch, bit manipulation unit and I/O interfaces [1]. Data movement on the chip occurs over five bidirectional 32-bit buses: X Data Bus (XDB), Y Data Bus (YDB), Global Data Bus (GDB), DMA Data Bus (DDB) and Program Data Bus (PDB). Address are specified on X Address Bus (XAB) for internal X Data Memory, on Y Address Bus (YAB) for internal Y Data Memory and on Program Address Bus for internal Program Memory. The Data ALU consists of ten 96-bit general purpose registers, a 32-bit barrel shifter, a 32-bit adder and a 32-bit parallel multiplier. It performs all of the arithmetic and logical operations on data operands, implementing the IEEE 754 binary floating-point arithmetic. The X Data Memory and Y Data Memory may contain both data RAM (512x32 bits) and ROM (1024x32 bits). The Program Control and System Stack performs instruction prefetch, instruction decoding and exception processing. It has a 32-bit program counter to address the internal Program Memory (1024x32 bits) and external memory (up to 4MB). The DSP96002 has two identical external bus interfaces, each with 32 bit of data and 32 bit of address to access external Data and Program Memory and I/O devices. The bloc diagram of the DSP development system is shown in Figure 2.

The DSP96002 is connected to the host computer via the OnCE (On Chip Emulator) serial debug port. It contains four pins: DR# (Debug Request) causes the DSP96002 to enter the debug mode and wait commands from the debug serial input line. DSCK (Debug Serial Clock) receives the serial clock supplied by the host. DSI (Data Serial Input) receives the serial data and commands from the host computer. DSO (Data Serial Output) provides the output data.

The analog-to-digital converter is represented by a DSP56ADC16 chip and a 16-bit shift register. The DSP56ADC16 achieves 16 bit accuracy at output data rates up to 100 kHz and requires an input clock frequency 128 times the output sample rate [2]. In this application the device carries out analog to digital conversion on 16 bits for each 12 ECG leads at 400 samples/sec. When the least significant bit of a word arrives, the parallel word is latched into a three-state buffer and the DSP96002 may read the data anytime during the transmission of the subsequent data word. The control logic provides a Frame Sync Input (FSI) to the ADC initiating the data transmission, address decoding that enables the DSP 96002 to read the three-state buffer and write the output data buffer and interrupt request to the processor when the output buffer is loaded. The acquisition time interval for D II and V 2 leads is 10 seconds (long leads) and for D I, D III, avr, avl, avf, V 1, V 3, V 4, V 5 and V 6 leads is 2 seconds (short leads).

The development system uses only one analog-to-digital converter because the input signal frequency is not high. In order to provide the 12 ECG leads to the converter the system needs an input analog multiplexer, controlled by a selection register, loaded with the analog input code.

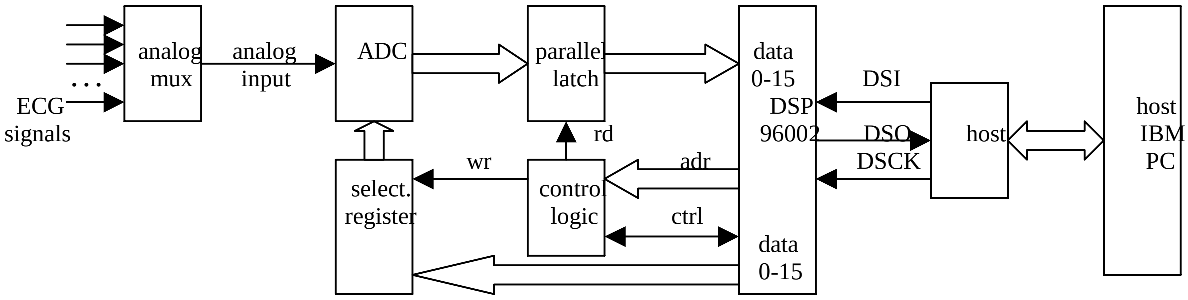

Figure 2. DSP based processing system block diagram.

ECG SIGNAL PROCESSING

A very powerful algorithm was implemented to ECG waves recognition. This algorithm bases on a geometric rule set that describes the features of electrocardiogram leads. The input data are represented by a sample value array corresponding to the 12 standard leads. The output data of the pattern recognition program is represented by an array of numeric values for ECG waveform parameters.

There are two kinds of parameter vectors:

-time vectors: a vector element is represented by the sample number k, from the acquisition beginning when the interesting event was occurred. The corresponding time interval can be calculated using the following equation: $t = k / f$ (s), where f represents the sampling frequency;

-amplitude vectors: a vector element is represented by the numeric value provided by the analog to digital converter, that codifies the ECG bipolar signal amplitude at one moment. The voltage value is given by: $u = v * 5 / 65536$ (V).

For the QRS complex detection, a short part of the basic algorithm used for ECG waves recognition is described here in a pseudocode language:

search forward for a steep slope from the 50-th sample;
search forward for a peak / connection;
search backward for the beginning of a positive / negative slope;
{the start of QRS complex}
if slope=positive **then**
  search forward the R peak;
5: R peak **found**;
  if end(R)=peak **then**
    if peak<0 **then**
      S wave **found**;
    search forward the end of the slope;
    if end=peak **then**
      R’ wave **found**;
      J point **found**
    else
      J point **found**
  else
    J point **found**
else {slope=negative}
  search forward the Q peak;
  search forward the end of the slope;
  if end=peak **then**
    goto 5{label}
  else
    J point **found**;

The vectors obtained by the recognition program represent the input data for the diagnosis program. It will determine ventricular conduction defects, preexcitation, ventricular enlargement and atrio-ventricular conduction disorders. Figure 3 presents the electrocardiographic waves for an experimental data set.

The working menu of ECG processing program [3] is displayed on the top line of the screen, and comprises the following commands: INFORM, THRESHOLDS, MEASURE, DATA, PROCESSING, GRAPHICS and QUIT. Each command contains options that are displayed by activating the command. Their significance is as follows:

INFORM displays informations about patients and files.

THRESHOLDS contains the numerical parameters for filtering and pattern recognition.

MEASURE allows sampling of the experimental data from the electrocardiograph.

DATA allows saving of the sampled data, loading of previously saved data, listing of the data existing in the program memory.

PROCESSING selects the processing modes for the sampled data. After the data acquisition, the user can optionally command the data filtering. There is three filtering procedures :

-a procedure that eliminates peaks with a duration or amplitude less than a threshold value ;

-a procedure that performs ECG base line identification ;

-a procedure that eliminates the ECG device and patient movement noises. Each waveform is inspected and each consecutive five sample set is analysed and compared with standard patterns, performing correction in case of mismatch.

GRAPHICS allows graphical display on the screen of the electrocardiogram.

QUIT allows leaving or initialising the program.

The program allows calculate FFT for the long leads (D II and V 2). The frequency domain approach is very interesting and offers new means for ECG processing.

The FFT calculations are accomplished in real time on the DSP96002 processor and consist of the following processing module: an assembler macro to generate the “twiddle” factors (the

Figure 3. Electrocardiographic signals for an experimental data set.

weighting factors), an assembler macro to normally reorder the bit-reversed output sequence, an assembler macro to implement an FFT butterfly using both the X-Memory and the Y-Memory of the processor, a procedure to compute an FFT's group offset and the number of groups per stage, a procedure to compute the butterfly coefficient of each group of the FFT's stages and an assembler macro to implement a complete N-point FFT.

CONCLUSIONS

The paper discusses a very powerful way to process analog signals using a DSP based development system connected to a host computer (a PC platform). The system receives digital values based on samples of the ECG signal, calculates in real time the FFT of these samples and provides values that represent the pattern recognition program inputs. The high speed arithmetic and logical hardware is programmed to rapidly execute the algorithm modeling the Fast Fourier Transformation.

A significant advantage of the floating point DSP is obtained. Using a floating point DSP code can be compacted. It is possible to squeeze all of the program code into the chip on board RAM. In comparison fixed point device would use considerably more code to implement the same operations. A simple FFT, for example, requires almost three times the amount of code as the same operation performed in floating point. As the signal increases in frequency, more code must be added to do all of the overflow checking. Fixed point operations also make it very difficult to write FFT algorithms for very large transforms. For example, in a fixed point operation it's much more difficult to write a 1000-point FFT algorithm than is to write a 128-bit algorithm. With the floating point chip, very generic software can thus be used regardless of which mode the DSP is running.

REFERENCES

[1] ***, "DSP 96002 User's Manual", Motorola, 1989
[2] M.Sharkawy, "Signal Processing, Image Processing and Graphics Applications with Motorola's DSP 96002 Processor", PrenticeHall PTR, 1994
[3] F.Iacob, "IBM PC Interface for ECG Processing", The 9-th International Conference on Control Systems and Computer Science, Bucharest, 25-28 May 1993, vol.II, pg.310-315