Microcontroller-Based Real-Time QRS Detection

YING SUN, PHD, SETH SUPPAPPOLA, THOMAS A. WRUBLEWSKI

The authors describe the design of a system for real-time detection of QRS complexes in the electrocardiogram based on a single-chip microcontroller (Motorola 68HC811). A systematic analysis of the instrumentation requirements for QRS detection and of the various design techniques is also given. Detection algorithms using different nonlinear transforms for the enhancement of QRS complexes are evaluated by using the ECG database of the American Heart Association. The results show that the nonlinear transform involving multiplication of three adjacent, sign-consistent differences in the time domain gives a good performance and a quick response. When implemented with an appropriate sampling rate, this algorithm is also capable of rejecting pacemaker spikes. The eight-bit single-chip microcontroller provides sufficient throughput and shows a satisfactory performance. Implementation of multiple detection algorithms in the same system improves flexibility and reliability. The low chip count in the design also favors maintainability and cost–effectiveness. (BIOMEDICAL INSTRUMENTATION & TECHNOLOGY 1992;26:477–484)

Detection of QRS complexes in the electrocardiogram (ECG) is necessary for the operation of many medical instruments. For example, cardiac pacemakers sense the electrical activities in the atrium and ventricle.[1] The delivery of electrical stimulus is often inhibited when a spontaneous R wave is detected. For imaging the pumping heart using either x-rays[2] or magnetic resonance,[3] ECG-gated acquisition is common, either to compensate for the low temporal resolution of an imaging system or to enhance the signal-to-noise ratio by averaging images over several cardiac cycles. When an intraaortic balloon pump is used to assist a failing heart,[4] the inflation and deflation of the balloon must be synchronized with the cardiac cycle; the R wave of the ECG provides the key reference for the timing control of an intraaortic balloon pump.

Although much research has been devoted to the development of both analog and digital QRS detectors,[5] reliable detection of QRS complexes remains a challenge. QRS detection in the clinical environment is difficult because: 1) The wave shape of the QRS complex varies significantly in health and in disease.[6] 2) Several other signals in the ECG recording, such as an elevated T wave or a premature ventricular contraction, may be morphologically similar to a QRS complex. 3) Various types of noise (e.g., caused by a pacemaker, an electrosurgical tool, power line interference, or patient motion) interfere with QRS detection. 4) The quality and the frequency bandwidth of recorded ECGs may vary significantly, depending upon the purpose of the recording (diagnosis or monitoring) and the recording environment (e.g., in an ambulance, in an operating room, during exercise or at rest, with or without telemetry).[7]

Conventional QRS detectors based upon analog electronics generally fail to provide the flexibility of handling large variations in the ECG wave shape and in the signal–noise condition. A computer-based QRS detector, on the other hand, may suffer from slowness in response because of the throughput limitation imposed by the underlying hardware and the involvement of software for implementing the detection algorithm. The increasing complexity of hardware and the involvement of software not only change the design of a QRS detection algorithm dramatically but also raise concerns about reliability and fault tolerance.

The purpose of this study was twofold. First, the instrumentation requirements for QRS detection were systematically analyzed, to provide a guideline for the design of a computer-based QRS detector. Second, a real-time QRS detector based on a single-chip microcontroller (the Motorola 68HC811) was developed to exemplify techniques for algorithm design, performance evaluation, and enhancement of system reliability and maintainability.

Received from the Department of Electrical Engineering, University of Rhode Island, Kingston, Rhode Island (YS, SS) and Boston Scientific Corporation, Mansfield, Massachusetts (TAW).

Address correspondence and reprint requests to Dr. Sun: Department of Electrical Engineering, University of Rhode Island, Kingston, RI 02881.

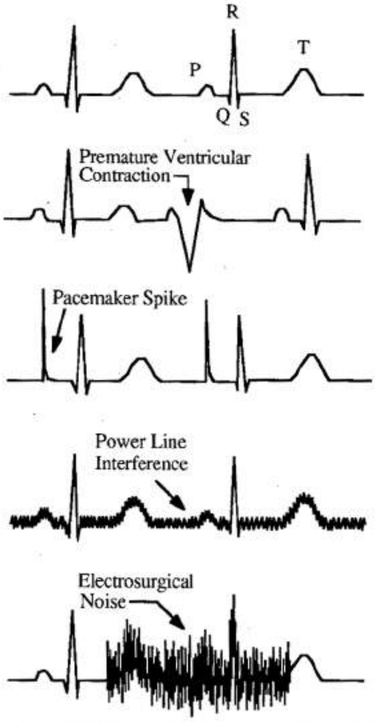

Figure 1. A typical ECG waveform and different types of interference.

INSTRUMENTATION REQUIREMENTS FOR QRS DETECTION

The detection algorithm is the most important component in the design. The hardware and software specifications must be developed on the basis of, and in support of, the detection algorithm. However, it is unlikely that any single QRS detection algorithm will satisfy all instrumentation requirements and perform satisfactorily in a variety of clinical environments. The following instrumentation requirements should be considered in the algorithm design.

Real-time

A real-time algorithm determines whether or not a QRS complex is present while the ECG is sampled on a continuing basis. More specifically, the execution time of the detection software must not exceed one sampling period. A real-time algorithm must be causal, i.e., only present and past data may be used. By contrast, a non–real-time algorithm processes data previously acquired and stored; thus, it does not have to be causal. The ECG has a frequency bandwidth between 0.05 Hz and 150 Hz. However, the frequency bandwidth of recorded ECGs may be reduced by the recording device. For example, the ECG channel from a bedside monitor may have a cutoff frequency as low as 40 Hz. A common sampling rate for QRS detection is around 250 Hz. This imposes an upper bound of 4 ms on the execution time of the detection software.

Response Time

Some instruments may require an early decision with a detection point before the peak of the R wave, while others accept a late decision with a detection point after the appearance of the complete QRS complex. The requirement for response time determines how many samples on the QRS complex can be used in the decision process. There is a tradeoff between response time and robustness. The detection decision is more robust when more information is gathered from the QRS complex at the sacrifice of response time.

Noise Rejection

The types of noise that are present and how they should be rejected are also important considerations in the algorithm design. Different types of noise in recorded ECGs are illustrated in Figure 1. Some instruments do not trigger on premature ventricular contractions (PVCs), while others do. Rejection of pacemaker spikes is often required, while in some circumstances of ventricular pacing, the pacemaker spike is the only available signal to trigger the instrument because normal R waves are absent. Thus, the noise-rejection capability not only is an important feature to implement but also should be implemented such that user intervention is possible.

Receiver Operating Characteristics

Certain parameters in a detection algorithm affect the sensitivity of QRS detection. If sufficient test data are available, the receiver operating characteristics (ROC) of a detection algorithm can be established by plotting the true detection rate vs. the false detection rate.[8] If other conditions remain unchanged, increasing the sensitivity of a detection algorithm increases the true detection rate and decreases the false detection rate. In the case of QRS detection, the appropriate operating point on the ROC curve depends on the instrument and its application. For example, if the deflation of an intraaortic balloon is triggered by the R wave, a false negative (missed detection) is less tolerable than a false positive (spurious detection). This is because a missed detection of an R wave would obstruct left ventricular ejection by leaving the balloon inflated during systole. On the other hand, a spurious detection would deflate the balloon prematurely and would only decrease the *assistance* provided by the intraaortic balloon.

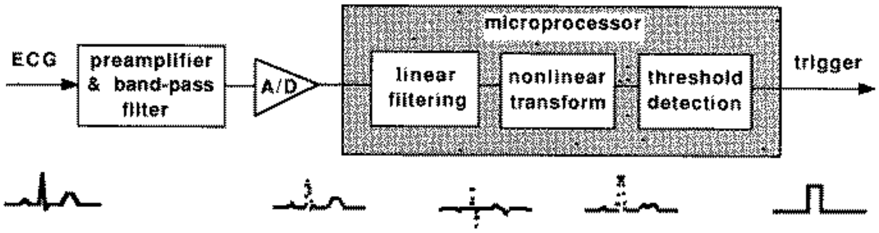

Figure 2. Block diagram of a computer-based QRS detector and waveforms of a processed ECG at various stages.

Dynamic Range

The dynamic range of a QRS detection algorithm is defined as the range of the input signal magnitude within which the algorithm gives an acceptable performance. In the clinical environment, the magnitudes of ECGs at the input of QRS detectors may vary over a considerable range. An algorithm is more robust if it has a relatively broad dynamic range. For digital QRS detection, the dynamic range depends not only on the algorithm itself but also on the degree of quantization of the input signal.

QRS DETECTION ALGORITHMS

Review of Past Research

As Figure 2 shows, a typical QRS detection algorithm contains three stages of processing: linear filtering, nonlinear transform, and decision.[7,9] The linear-filtering stage usually includes an operation of digital differentiation. The nonlinear-transform stage traditionally involves a moving average of the signal squares, which provides an estimate of the signal energy. Then, the output of the nonlinear transform is compared with a threshold in the decision stage.

The appropriate linear filtering on the ECG for the purpose of QRS detection has been thoroughly investigated.[5,10] A band-pass filter with a pass band between 5 and 15 Hz is generally acceptable. A 60-Hz (or 50-Hz) notch filter is also useful for suppressing power-line interference. However, linear filtering alone is not sufficient to eliminate other types of noise, including motion artifacts, pacemaker spikes, and electrosurgical interference. These types of noise typically have frequency spectra overlapping in part with that of the QRS complex. Thus, suppressing these interferences by low-pass filtering would also obscure the R wave itself. An appropriate nonlinear transform tuned to the wave shape of the QRS complex, on the other hand, can significantly enhance the signal-to-noise ratio prior to threshold detection. This is why the nonlinear transform plays a very important role in QRS detection.

The frequency domain behavior of linear filtering can be completely characterized by using Fourier analysis. By contrast, analytical tools for studying the time-domain nonlinear operations on signals are very limited. Some heuristic-based nonlinear operations[11] and morphologic filters[12] have been applied to the problem of QRS detection in the past. Our preliminary study[13] has shown that two-point backward difference multiplication can be used as a nonlinear transform for early detection of R waves. However, the use of various nonlinear transforms to suppress noise and to enhance QRS complexes remains an area where further investigation is needed.

Proposed Nonlinear Transforms

We systematically compared the QRS detection algorithms based on five nonlinear transforms. All five transforms involve some operations of multiplying consecutive samples. To provide a common ground for comparison, the backward difference is used as the linear filtering stage. Let u_n denote the current ECG sample. The derivative of the ECG is approximated by the first-order backward difference, x_n, given by:

$$x_n = u_n - u_{n-1}$$

Let y_n denote the output from the nonlinear transform. The five nonlinear transforms under investigation are defined as follows:

$$y_n = x^2_n \tag{2}$$

$$y_n = x_n x_{n-1} \tag{3}$$

$$y_n = \begin{cases} x_n x_{n-1}, & \text{if } x_n x_{n-1} > 0 \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

$$y_n = x_n x_{n-1} x_{n-2} \tag{5}$$

$$y_n = \begin{cases} x_n x_{n-1} x_{n-2}, & \text{if } x_n, x_{n-1}, x_{n-2} \\ & \quad \text{sign consistent} \\ 0, & \quad \text{otherwise} \end{cases} \tag{6}$$

Adaptive Thresholding

An adaptive threshold is used in the decision stage. A refractory period (typically 100 ms) is set immediately after the detection of an R wave to prevent multiple triggers on the same QRS complex. The adaptive threshold is used to increase the sensitivity of QRS detection as the elapsed time from the last R wave increases. Immediately following the refractory period the threshold for detecting the next R wave is set at the maximum of y_n during the past refractory period. From this point on, the threshold is halved whenever a fixed time period elapses. This predetermined time period is called the time constant for threshold decay. The threshold is not allowed to decay below a lower bound. The lower bound "floats" above the noise level by an adaptation scheme based on the noise level of previous cycles. The average

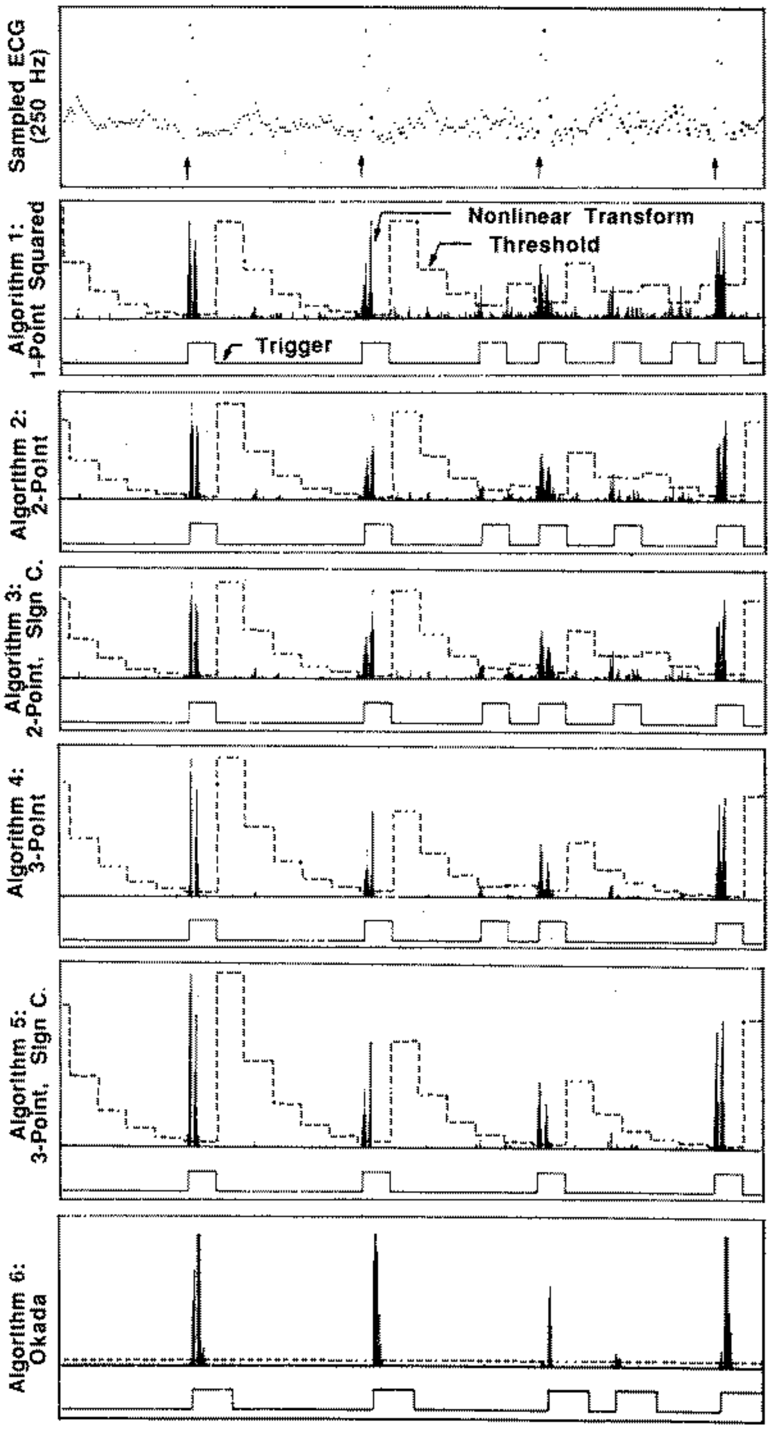

Figure 3. A 2.5-second segment of an ECG signal from the AHA ECG database (tape #6203, channel 1, beats 247-250) and the results from the six QRS detection algorithms. Arrows in the ECG panel show the onsets of R waves marked by annotation.

of y_n during the last cardiac cycle (excluding the refractory period) is computed. The lower bound is updated by the average y_n plus half of the previous lower bound. This averaging scheme is equivalent to a first-order infinite impulse response (IIR) filter.

Comparison with an Existing Algorithm

To provide a performance reference to the above five algorithms, we also implemented the algorithm developed by Okada.[11] The Okada algorithm was chosen because its response time is comparable to that of the proposed algorithms. Some other algorithms[7,9] may provide better performance, but they either are not for real time or have much longer response times (the detection point can be as late as a few hundreds of milliseconds after the R wave). The Okada algorithm contains several nonlinear operations, including the multiplication of each point by the square of a moving average of several adjacent points. Its computational complexity, in terms of processing time, is about four times that of the three-point with sign consistency algorithm (equation 6). The Okada algorithm employs a constant threshold throughout the cardiac cycle; the threshold is set at 1/32 of the maximum of the nonlinear transform output.

Evaluation of Algorithm Performance

The ECG database compiled by the American Heart Association was used to test the algorithms. Figure 3 shows examples of the QRS detection performances of the six algorithms. A segment of ECG signal sampled at 250 Hz is shown at the top. For each algorithm the output from the nonlinear transform is plotted, overlapping with the adaptive threshold. The trigger output from the threshold detection is also shown. Due to the presence of noise in the last two beats, spurious triggers are observed for all algorithms except the three-point with sign consistency algorithm (algorithm 5). The three-point algorithms (algorithms 4 and 5) reject the noise better than do the one-point and two-point algorithms (algorithms 1 to 3), as indicated by the cleaner baseline and the clearer separation between the signal and the background. The effect of the sign consistency constraint is the elimination of some sparsely occurring noise spikes. The Okada algorithm also has good noise-rejection capability; the spurious trigger could be avoided if the threshold adaptation scheme were used.

Receiver Operating Characteristics with Respect to Dynamic Range

To provide a quantitative assessment of the algorithms' performances, the false positives and false negatives for each algorithm were recorded when the quantization level was varied between six and 12 bits. The digital version of the AHA Arrhythmia ECG database was sampled at 250 Hz with 12-bit quantization. The sampling rate is fixed and cannot be varied, but the degree of quantization can be easily increased by dropping the low-order bits. The AHA ECG database contains 80 tapes. The following eight tapes were used in the evaluation: 1207, 2202, 3202, 4204, 5208, 6203, 7208, and 8209. One tape from each series was included; the selection of tapes was otherwise arbitrary. In all, 34,060 beats were included in the evaluation. A valid detection must fall into the window that begins 50 ms before and ends 100 ms after the R wave marked by the AHA annotation. The window is shifted by the average processing delay for each algorithm so that, so far as detection accuracy is concerned, an algorithm is not penalized for slowness in response. The method used for performance evaluation has been described in detail.[14]

Table 1. Detection Performances of Three Algorithms Operating at Their Best Quantization Levels

Algorithm	Quantization	1 − FN* (%)	FP† (%)	FN + FP (%)	Delay (ms)
3-point, sign consistency	8-bit	99.2	2.9	3.6	2.4
2-point	10-bit	94.3	36.8	42.5	−2.2
Okada	12-bit	97.6	1.4	3.8	32.4

*False negative.
†False positive.

Only algorithm 5 (three-point with sign consistency) and algorithm 6 (Okada's algorithm) performed satisfactorily. Algorithms 1 to 4 are, in general, too sensitive; although they gave reasonable true detection rates, their false detection rates were very high. For simplicity, in addition to algorithms 5 and 6, we show only the result of algorithm 3 (two-point), as a representative for the rest of the algorithms. Table 1 summarizes the detection performances of the three algorithm, each operating at its best quantization level. The false-positive (FP) rate and the false-negative (FN) rate are defined as:

$$\text{FP} = \frac{\text{number of spurious triggers}}{\text{number of beats analyzed}} \times 100(\%) \qquad (7)$$

$$\text{FN} = \frac{\text{number of missed triggers}}{\text{number of beats analyzed}} \times 100(\%) \qquad (8)$$

The false detection rate is the same as the FP rate. The true detection rate is defined by (1 − FN). A single performance index (FP + FN) is formed by weighting

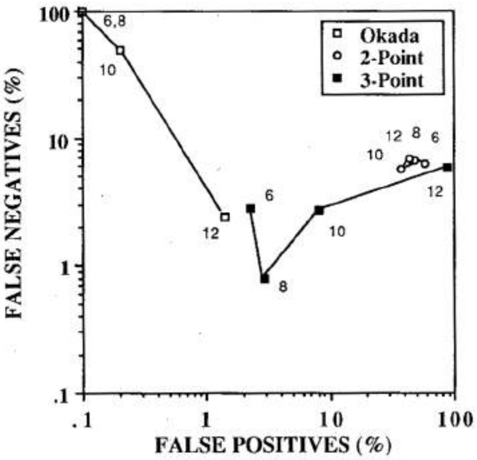

Figure 4. False negatives vs. false positives for Okada's algorithm (Okada), the two-point algorithm (2-point), and the three-point with sign consistency algorithm (3-point). The degrees of quantization are varied as shown by the numbers of bits adjacent to individual plotted points.

FN and FP equally. The best performance was that of the three-point with sign consistency algorithm (3.6%), closely followed by the Okada algorithm (3.8%).

To further probe the issue of FN–FP tradeoffs, Figure 4 shows FN versus FP for each of the three algorithms. Notice that, by definition, the ROC curve is plotted with (1 − FN) versus FP. We use FN instead of (1 − FN) because points corresponding to low FN and low FP are shown with better resolution on the logarithmic scales. In Figure 4 the theoretical best performance corresponds to the point at origin. The closer a point is to the origin, the better the performance is. If a point is closer to the FP axis than the FN axis, the algorithm gives more spurious triggers than missed triggers, which is the case for the three-point with sign consistency algorithm. The three-point with sign consistency algorithm shows a relatively broad dynamic range. In fact, a higher degree of quantization seems to help prevent spurious detections; the best performance was that obtained with eight-bit quantization in this test. The point of detection was, on average, 2.4 ms after the annotation mark. The Okada algorithm, on the other hand, generally showed more false negatives than false positives. Its dynamic range is relatively narrow; performance deteriorated abruptly as the number of bits dropped below 12. It also has a longer response time; the point of detection was, on average, 32.4 ms after the annotation mark.

MICROCONTROLLER-BASED QRS DETECTOR

Hardware

As Figure 5 shows, the QRS detector consists of three components, i.e., a single-chip microcomputer (Motorola 68HC811E2), a crystal oscillator, and a power-on reset circuit. The 68HC811 chip contains an eight-bit, 6,800-family microprocessor. This processor supports eight-bit integer arithmetic, including multiplication and division. No floating-point instruction is available. When running under the "single-chip" mode, it becomes a self-contained computer system equipped with on-chip random-access memory (RAM), electrically erasable programmable read-only memory (EEPROM), and a 16-bit timer. It communicates with the outside world via serial and parallel I/O ports and an eight-bit, eight-channel analog-to-digital converter.

Development System

The development system consists of an IBM Personal Computer and a Motorola 6811 evaluation module (M68HC11EVM). An assembly source program is created on the PC and cross-assembled into an ASCII-coded (Motorola S-record) machine program. The program is down-loaded to the evaluation module via an RS-232 connection. The evaluation module has the in-circuit emulation capability that allows one to test and debug the program using the same I/O environment as the target device has. The evaluation module is also capable of erasing, programming, and verifying the contents of

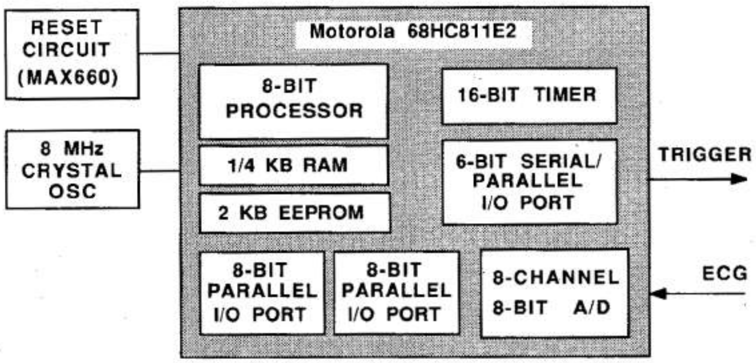

Figure 5. System hardware for the QRS detector based on the 68HC811 microcontroller.

EEPROM on a 68HC811 chip. On the target device (i.e., the QRS detector) an 8-MHz crystal oscillator chip is used. This clock is internally divided by 4, and the resulting system clock rate is 2 MHz. External switches are connected to one of the parallel I/O ports. These switches are used to select different detection algorithms stored in the EEPROM.

Software

A main program is initiated upon the power-on reset. The main program sets the 16-bit timer to generate periodic interruptions at the chosen sampling rate. The QRS detection algorithm is implemented in an interrupt service routine. The following is the pseudo code for an algorithm using equation 4 as the nonlinear transform:

```
1.0 Reset the timer interrupt flag.
2.0 Assign old $u_n$ to $u_{n-1}$ and get new $u_n$ from A/D. Start
    A/D for next sample.
3.0 Assign old $x_n$ to $x_{n-1}$ and compute $x_n = u_n - u_{n-1}$.
4.0 IF $x_n$ and $x_{n-1}$ have the same sign
    THEN
       4.1 $y_n = x_n \times x_{n-1}$.
    ELSE
       4.2 $y_n = 0$.
5.0 IF refractory = TRUE
    THEN
       5.1 threshold = maximum ($y_n$, threshold).
       5.2 Count the refractory period, IF expired
    THEN
          5.2.1 refractory = FALSE.
          5.2.2 trigger = 0.
    ELSE
       5.3 Count the threshold decay period, IF expired
    THEN
          5.3.1 Reset decay counter.
          5.3.2 threshold = threshold/2.
       5.4 IF $y_n$ > threshold
    THEN
          5.4.1 refractory = TRUE.
          5.4.1 Reset refractory counter.
          5.4.2 trigger = 1.
6.0 Output trigger.
7.0 Return from interrupt.
```

Evaluation

Three QRS-detection algorithms were implemented on a 68HC811E2 microcontroller. The first (three-point with sign consistency) employs equation 6 and a sampling rate of 200 Hz. The second (two-point) employs equation 3 and a sampling rate of 500 Hz. The second algorithm does not distinguish a pacemaker spike from an R wave, while the first does. The third algorithm is the Okada algorithm implemented with a sampling rate of 250 Hz. The sampling rate was chosen for each algorithm to achieve its best performance. Because we were unable to optimize the sampling rate for each algorithm using the digital AHA ECG database, the sampling rates were chosen empirically and are not necessarily optimal.

Via an external switch connected to one of the parallel I/O ports of the microcontroller, the selection of which algorithm to use is entered. The three programs combined occupy 872 bytes, less than half of the total on-

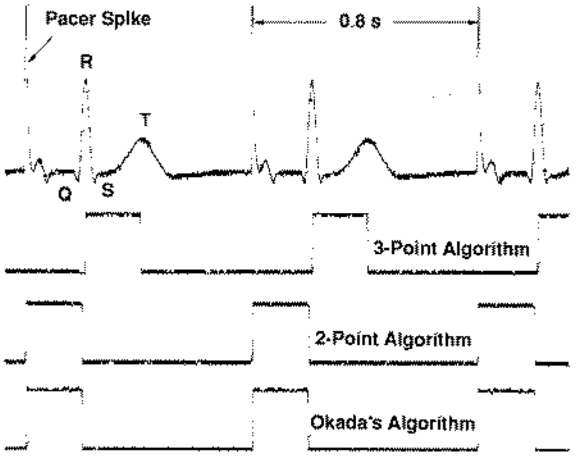

Figure 6. Screen dump from a digital oscilloscope that shows the trigger outputs of the microcontroller-based QRS detector with three detection algorithms. The ECG and pacemaker spikes are generated by a patient dynamics simulator.

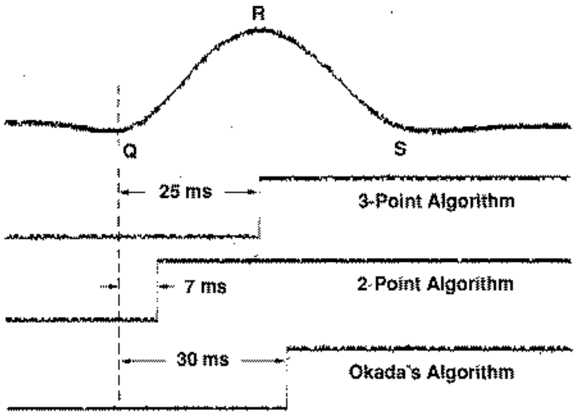

Figure 7. Comparison of the response times of the three QRS detection algorithms. The waveform on the top is a QRS complex shown on an expanded time scale. The elapsed time from the Q wave to the detection point is shown for each QRS detection algorithm.

chip EEPROM. The execution time for any of the three algorithms is less than 0.2 ms.

Figure 6 shows an example of the ECG input and the trigger outputs for the three algorithms. The ECG signal with pacemaker spikes was generated by a patient dynamics simulator (Model 130, Cardiac Assist, BSC, Mansfield, MA). The three-point with sign consistency algorithm rejects pacemaker spikes and triggers on R waves. Both the two-point algorithm and the Okada algorithm trigger on pacemaker spikes. Figure 7 shows the differences in the detection points for the three algorithms. In this example, the two-point algorithm gives the fastest response, with a detection point at 7 ms after the Q wave. The three-point with sign consistency algorithm triggers around the peak of the R wave, at 25 ms after the Q wave. The Okada algorithm triggers shortly after the R wave, at 30 ms after the Q wave.

CONCLUSION

The detection algorithm is the most critical component in the design of a computer-based QRS detector, and strongly influences the architecture of the hardware. Specifically, we have arrived at the following conclusions: 1) Certain nonlinear transforms on the ECG signal and heuristics for setting the detection threshold improve the sensitivity and reliability of QRS detection. The incorporation of these transforms and heuristics into the detection algorithm is much easier with software than with hardware. 2) Among the algorithms under investigation, only the three-point with sign consistency and the Okada algorithm performed acceptably. The former tended to give more false positives than false negatives, while the latter behaved in reverse. The three-point with sign consistency algorithm was superior in the categories of response time, computational simplicity, and dynamic range. 3) With a proper detection algorithm an eight-bit single-chip microcontroller provides sufficient computing power and reliable performance for real-time QRS detection. 4) Multiple detection algorithms can be implemented on the same chip to cope with waveform variations in recorded ECGs. 5) The number of discrete components of the hardware can be as low as three. The low chip count together with the availability of on-chip electrically erasable programmable ROM make the system very maintainable and cost-effective. 6) Although not demonstrated in this study, fault tolerance can be easily achieved with multiple algorithms in the same processor and multiple processors in the same system. ■

REFERENCES

1. *Tarjan PP, Berstein AD. An engineering overview of cardiac pacing. IEEE Eng Med Biol. 1984;9:10-4.*
2. *Masuda Y, Yoshida H, Morooka N, et al. ECG synchronized computed tomography in clinical evaluation of total regional cardiac motion: comparison of postmyocardial infarction to normal hearts by rapid sequential imaging. Am Heart J. 1982;103:230-8.*
3. *Higgins CB, McNamara M. Magnetic resonance imaging of ischemic heart disease. Prog Cardiovas Dis. 1986;28:257-66.*
4. *Weber KT, Janicki JS. Intraaortic balloon counterpulsation. Ann Thorac Surg. 1974;17:602-36.*
5. *Thakor NV, Webster JG, Tompkins WJ. Optimal QRS detector. Med Biol Eng Comput. 1983;21:334-50.*
6. *Van Bemmel JH. Estimation of biological signal parameters in dynamic environment. In: Inbar GF, ed. Signal analysis and pattern recognition in biomedical engineering. New York: John Wiley and Sons, 1974;63-92.*
7. *Pahlm O, Sörnmo L. Software QRS detection in ambulatory monitoring — a review. Med Biol Eng Comput. 1984;22:289-97.*
8. *Van Trees HL. Detection, estimation, and modulation theory. New York: John Wiley and Sons, 1968;37-47.*
9. *Hamilton PS, Tompkins WJ. Quantitative investigation of QRS detection rules using the MIT/BIH arrhythmia database. IEEE Trans Biomed Eng. 1986;33:1157-65.*
10. *Pan J, Tompkins WJ. A real-time QRS detection algorithm. IEEE Trans Biomed Eng. 1985;32:230-6.*
11. *Okada M. A digital filter for the QRS complex detection. IEEE Trans Biomed Eng. 1979;26:700-3.*
12. *Chu CH, Delp EJ. Impulsive noise suppression and background normalization of electrocardiogram signals using morphological operators. IEEE Trans Biomed Eng. 1989;36:262-73.*
13. *Wrublewski TA, Sun Y, Beyer JA. Real-time early detection of R waves of ECG signals. In: Proc IEEE Eng Med Biol Society 11th Annual International Conference, Piscataway, NJ: IEEE (Order No. 89CH2770), 1989;38-9.*
14. *Suppappola S. Digital QRS detection. Master's thesis, Department of Electrical Engineering, University of Rhode Island, 1990.*