VIPER: a powerful tool for the real-time calculation of inner products for biomedical signal processing

J. A. van Alsté   E. D. Luursema*

Bio-information Group, Department of Electrical Engineering, Twente University of Technology, PO Box 217, 7500 AE Enschede, The Netherlands

1 Introduction

COMPUTING time is a well known limitation in real-time signal processing in biomedical engineering. This is especially true when microcomputers are involved in the calculation of signal-processing operations, e.g. digital filters (OPPENHEIM and SCHAFER, 1975). Much effort has been put into the reduction of the number of time-consuming mathematical operations such as multiplications, often with considerable concessions to the performance of the algorithms concerned, to be able to implement them in small computer systems (WATANABE *et al.*, 1980).

We are interested in the real-time processing of electrocardiograms (ECGs) of exercising patients. Therefore we wish to high-pass filter three orthogonal ECG leads using linear phase filters and also to calculate correlation functions between incoming ECG beats and a number of beat templates (VAN ALSTÉ *et al.*, 1981). The mathematical operations involved with these functions are shown in eqns. 1 and 2.

$$\text{convolution sum: } y(k) = \sum_{i=0}^{N-1} h(i) \cdot x(k-i) . \quad . \quad . \text{(1)}$$

where

$x(k)$ = sampled input (ECG lead) signal.
$h(i) = i = 0, ..., N-1$, filter impulse response
$y(k)$ = filtered signal.

correlation coefficient:

$$r(k) = \frac{\sum_{i=0}^{N-1} x(k-i) \cdot y(N-1-i)}{\left\{ \sum_{i=0}^{N-1} x(k-i)^2 \sum_{i=0}^{N-1} y(i)^2 \right\}^{1/2}} \quad . \quad . \quad . \text{(2)}$$

* *Currently with the Dr. Neher Laboratories of The Netherlands Postal & Telecommunication Services, Leidschendam*

where

$x(k)$ is   the   input   signal   (ECG), $k = ... -3, -2, -1, 0, 1, 2, 3, ...$
$y(n), n = 0, ..., N-1$, (ECG beat) template
Both x and y are expected to have a zero mean at the observed interval.

A common mathematical operation in eqns. 1 and 2 and further signal processing is the inner product of two vectors having an equal length, as defined in eqn. 3.

$$\text{inner product } U \cdot V = \sum_{i=0}^{N-1} u_i v_i \quad . \quad . \quad . \quad . \text{(3)}$$

where

$$U = u_0, u_1, ..., u_{N-1}$$

and

$$V = v_0, v_1, ..., v_{N-1}$$

The calculation of an inner product is very time-consuming because of the large number of multiplications involved. In the case of filtering a real-time signal applying eqn. 1, N multiplications have to be performed for every new input signal sample $x(k)$.

To relieve the central processing unit of the computer from these inner vector product calculations we designed special hardware called VIPER, which stands for Vector Inner Product Equipment for Real time. VIPER operates in parallel with a computer system and is connected to it by means of parallel interfaces.

VIPER calculates a number of inner products of vectors consisting of 16-bit integer-valued arrays. All vectors are loaded by the computer and vector elements can be changed at any time. This implies that not only input signals but also impulse responses and templates can be changed during processing, which allows flexible signal processing. In its present form VIPER has a 4 ms cycle, which means that it is suitable for real-time processing of signals with a maximum

sampling rate of 250 Hz. In one cycle all programmed inner products are obtained, and data are exchanged with the external computer. The 4096 multiplications and additions included can be used for a maximum of 64 different inner products of two vectors each with 64 elements, or 16 products of vectors with a length of 256 elements, or a free-choice combination of these vector lengths.

2 Principle of operation

The main component of VIPER is the multiplier/accumulator, which performs a multiplication and accumulation operation of two 16-bit numbers in less than 200 ns. The task of the other components is to feed this integrated circuit with the proper data and to extract the results. The principle of operation will be discussed using the simplified block diagram of Fig. 1.

All vectors and the product results are stored in a random access memory (RAM) where they can be read or written by the external computer. The external computer has access to VIPER's RAM during the 2 ms communication period of VIPER's 4 ms cycle.

During the other 2 ms calculation period the RAM addresses of all vector elements needed are generated, the multiplier/accumulator is fed with the proper vector elements, the inner products are calculated and the results are stored on the appropriate generated RAM addresses.
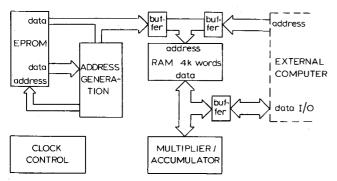

Fig. 1 *Simplified block diagram of VIPER*

Microcode stored in EPROM controls these operations.

The elements of a vector have to be stored at successive addresses. The microcode provides the lowest address of a vector memory field from which the address generation logic generates the actual element address.

For example, the address of the ith element $a(i)$ of a vector with length N is composed in the following way:

$$a(i) = v + (n + i) \bmod_N \qquad \ldots \ldots \ldots (4)$$

where v, the vector starting address, is the lowest address of the RAM space used for storage of the vector concerned and is directly obtained from the EPROM. n, the newest element index, indicates where the most recent element of the vector is stored, relative to v. This index is used when a vector is arranged as a circular buffer for input signals that have to be processed continuously. $\bmod_N$ represents the mathematical modulo N operation.

The clock control unit provides the synchronisation and clock signals necessary for the various hardware components. The possibility of filling and processing of vectors that are arranged in memory as a circular buffer makes it easy to update a vector in real time with new samples of an input signal. Hence, in this case, the communication between VIPER and the external computer is restricted to the exchange of the new samples and results.

3 Circuit description

The block diagram of Fig. 1 represents the total apparatus. The multiplier/accumulator comprises a single large-scale integrated circuit, the TDC 1010 J manufactured by TRW (*TRW*).

The multiplier obtains its data from and stores its results in a 4096 × 16-bit RAM via a 16-bit databus that is also connected to a bidirectional buffer. This buffer enables data storage and retrieval by an external device, such as the 16-bit parallel input/output port of a computer system. The required 12-bit memory addresses are then applied by the external computer to the address bus.

The generation of the addresses of the vector elements actually used is under the control of a microcode program
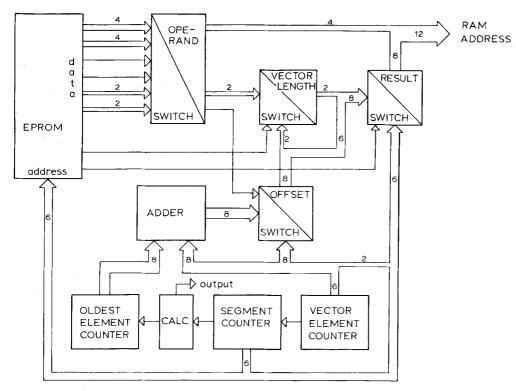

Fig. 2 *Extended block diagram of the address generation logic. Refer to the text for details*

stored in EPROM. This microcode contains the following information:

(a) the base addresses of the vectors involved in a vector product
(b) the vector length, being either 256 or 64 elements
(c) the vector storage mode as a circular or linear buffer
(d) the address where the product result should be stored.

Separate EPROM words are used for vector and result address specification. Four different microcode programs can be chosen by using a front panel switch which selects the two most important EPROM address bits. The logic used for the address generation will be described referring to Fig. 2. First its block diagram is explained for vector lengths of 64 elements.

The vector element counter is incremented every 488 ns and counts for every product from 0 to 63 giving the actual sample index number. The segment counter indicates which product has to be calculated, and its value is also used as the low-order bits of the actual product result address. After $64 \times 64 \times 488$ ns $= 2{\cdot}0$ ms the calculation flip-flop (calc) becomes low and access to the RAM is given to the external device. The counting continues and after another $2{\cdot}0$ ms calc becomes high again and the vector products are calculated again. Also, the oldest element counter is incremented. This counter in combination with the adder and the offset switch allows the arrangement of vector elements in a circular buffer, starting with the oldest vector element, as assigned by the oldest element pointer. Therefore, the external device has to renew the oldest vector element of the floating vectors every 4 ms. If vector elements have to be arranged in a linear buffer the offset switch ignores the oldest element pointer.

The vector length switch decides whether the two most important address bits are taken from the EPROM microcode (length 64) or from the segment counter (length 256). The result switch normally puts the vector addresses through but, when a product result has to be stored, four bits from the microcode are extended with bits from the segment counter to form the result address.

The vector product results are internally calculated in 32-bit plus an extra 3 for overflow (TRW), which are used to provide a front-panel overflow warning. The inner product results are rounded off to their most significant 16 bits. Optimal use of these 16-bit results can be obtained by scaling the vector elements.

The circuits are constructed using standard low-power Schottky TTL integrated circuits ($Texas\ Instruments$, 1983).

4 Interfacing

Interfacing VIPER to a computer system can be performed simply by a 12-bit parallel output port for RAM addressing and a bidirectional 16-bit parallel port for RAM

input and output. A one-bit calc signal is output for synchronising with the calculation/communication phases. A one-bit output signal for synchronising the cyclic buffer pointer is also provided. Two handshake lines for either of the two parallel ports are used.

The communication between VIPER and the external computer is only restricted by the RAM access time of approximately 250 ns. The total RAM can be exchanged in one communication period of 2 ms. In practice the exchanged data will be limited by the data transport speed of the external computer. When the result of a vector product is needed again as a vector element for another VIPER operation, then the external computer has to take care of the transportation of the data in the RAM of VIPER.

5 Conclusions

The real-time inner product equipment VIPER has been in full operation for more than a year and works satisfactorily as a hardware extension of an LSI 11/23 computer system from Digital Equipment Corporation. It has been found to be very useful in real-time signal processing and real-time control, especially when impulse responses or templates have to be adaptive.

Its easy interfacing makes it suitable to process it parallel to all microcomputers that can be equipped with parallel interfaces.

The calculation time can be halved by using faster types of RAM and EPROM, resulting in the real-time processing of signals sampled at a maximum rate of 500 Hz.

References

OPPENHEIM, A. V. and SCHAFER, R. W. (1975) $Digital\ signal\ processing$. Prentice-Hall, Englewood Cliffs, New Jersey.

$Texas\ Instruments$ (1983) $The\ TTL\ data\ book\ for\ design\ engineers$, vol. 1. ISBN 3-88078-037-4.

TRW Multiplier-accumulator parallel 16-bit. Model: TDC1010J. Data sheet from TRW, Redondo Beach, California.

VAN ALSTÉ, J. A., VAN ECK, W. and HERRMANN, O. E. (1981) Methods for exercise electrocardiography in patients unable to perform leg exercise: rowing ergometry, robust averaging and linear phase filtering. $Computers\ in\ cardiology$, IEEE Computer Society, Florence, Italy, 465–468.

WATANABE, K., BHARGAVA, V. and FROELICHER, V. (1980) Computer analysis of the exercise ECG: a review. $Progr.\ Cardiovasc.\ Dis.$, 22, 423–446.