Part V: Some elegant designs based on the use of recursive running sum filters

- This pile contains two articles on how generate elegant products with the aid of running sum filter.

- What to read for the examination: Why the recursive structures in these articles are so effective? When do they work in a proper manner? That is all folks!

A Digital Filter Chip for ECG Signal Processing

Tommi Raita-aho, Tapio Saramäki, and Olli Vainio, *Member, IEEE*

Abstract— A VLSI implementation of a linear-phase digital filter for ECG signal processing has been designed. With a sampling rate of 100 Hz, the passband is from 0.5 Hz to 49.5 Hz with 0.5-dB ripple. The filter architecture is based on the use of recursive running-sum blocks, resulting in a very low computational complexity. Module generators have been used in the layout design for high integration density. The circuit has been designed for a 2.0-μm double-metal CMOS technology, having about 34 000 transistors and a 15.43-mm^2 chip area.

I. INTRODUCTION

THE electrocardiogram (ECG) is an electrical signal caused by the functioning of the heart. The amplitude is typically a few millivolts. The ECG signal can be used to count the heart rate as well as for various diagnostic purposes in medicine. The time-domain characteristics and artificial generation of ECG signals are described, for instance, in [1]. Recent results of applying modern signal-processing methods to ECG signal analysis are given in [2].

In practical ECG measurements, the primary signal is often contaminated by relatively strong disturbances, which must be removed before the signal is registered. The varying ECG contact potentials and breathing artefacts (below 0.5 Hz) cause unwanted baseline drift. Especially in stress ECG recordings, this drift may sometimes make the recording impossible. Another problem is the mains frequency (50 Hz) noise which occurs within the clinically important frequency range.

Up to now, the harmful baseline drift is usually removed from the measurements using analog high-pass filters. Because of their highly nonlinear phase response, the passband cutoff frequency is usually selected to be only one-tenth of the heart rate. With these filters it is not possible to effectively remove low-frequency noise without distorting the ECG signal shape. The same is true for conventional infinite-impulse-response (IIR) digital filters. Analog notch filters are very sensitive to component aging and temperature variations, thus making it necessary to adjust the Q-value frequently. Digital filters do not suffer from this problem.

In the case of linear-phase filters, like linear-phase finite-impulse-response (FIR) filters, it is possible to increase the cutoff frequency up to the fundamental heart rate without causing any distortion to the ECG signal. The linear-phase performance of FIR filters enables us to make the stopband deeper and wider compared with nonlinear-phase filters. Thus linear-phase FIR filters give us a potential tool of effectively eliminating low-frequency noise components. In [3] and [4], special computationally efficient FIR filter designs have been

proposed for eliminating low-frequency noise up to 0.4 Hz. Using a transfer function of the form $H(z^4)$ and a sampling rate of 200 Hz, these designs attenuate simultaneously 50-Hz and 100-Hz disturbances.

The abovementioned filters are easily implementable using general-purpose medium-speed microprocessors. However, for practical VLSI implementations, they have a large number of delay elements, making the overall silicon area large. Another disadvantage is a long delay caused by the filter to the ECG signal. An efficient VLSI implementation can be achieved by constructing the overall filter using easily VLSI-realizable building blocks. It has turned out that the recursive running-sum filter is such a block, requiring one multiplier and two adders regardless of the length of the filter. This paper concentrates on designing a filter for meeting the criteria stated in [7] and [8]. It is required that the filter provides narrow stopbands around 0 Hz and 50 Hz for the sampling rate of 100 Hz. The proposed tailored design requires no general multipliers, and the required silicon area is only 25 percent of what the VLSI design considered in [7] requires. This filter is aimed to be used in heart rate monitoring where the sampling rate of 100 Hz is sufficient. However, by slight modifications given in this paper, it can equally well be used for the sampling rate of 200 Hz. In this case, the whole clinically important frequency range of the ECG signal could be processed.

The proposed digital filter chip is attractive for battery-operated portable heart rate monitors where the power consumption and the physical size of the device are critical. In such applications, the entire system must fit into a small space, which is only possible by utilizing VLSI technology.

II. FILTER SPECIFICATIONS

The sampling rate of the ECG signal was specified in the application under consideration to be $f_s = 100\,\text{Hz}$ [7], [8]. The filter passband is (0.5 Hz, 49.5 Hz), the peak-to-peak passband ripple is 0.5 dB, and it is desired to have notches at the zero frequency and at 50 Hz. The minimum length of a conventional minimax FIR filter to meet these criteria is 281. Because of the symmetry of the design, every second multiplier value is zero. By exploiting the coefficient symmetry, this design requires 71 general multipliers. This figure is high for practical VLSI implementations, where the implementation of a general multiplier is very space consuming. Adders are not free, either. A single multiplier–accumulator is typically multiplexed in time between the FIR filter taps. However, this easily leads to large designs because of all the necessary control logic and coefficient memory, in addition to the data memory. In order to avoid the use of general multipliers, we propose in the
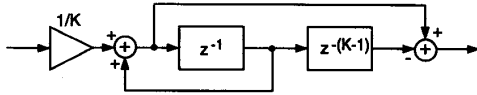
Fig. 1. Efficient implementation of the recursive running-sum filter $G_K(z)$.

next section a multiplier-free design which also has very few adders.

III. FILTER DESIGN

From the VLSI implementation point of view, the simplest FIR filter is the recursive running-sum (RRS) filter. The desired implementation form is obtained by expressing an averaging FIR filter transfer function of length K in the following recursive form:

$$G_K(z) = \frac{1}{K} \sum_{n=0}^{K-1} z^{-n} = \frac{1}{K} \frac{1 - z^{-K}}{1 - z^{-1}}. \tag{1}$$

An efficient implementation of $G_K(z)$ is depicted in Fig. 1. It requires only two adders and one multiplier $(1/K)$ regardless of the value of K. We note that if modulo arithmetic (e.g., 1's or 2's complement arithmetic) and the worst case scaling (corresponds to peak scaling in this case) are used, then the output of the RRS filter implemented as shown in Fig. 1 is correct even though internal overflows may occur. For details, see, e.g., [5]. This implementation is very attractive as, in this case, the filter does not need initial resetting, and the effect of temporary soft errors vanishes automatically from the output in a finite time.

The frequency response of this filter is given by

$$G_K(e^{jw}) = e^{-j(K-1)\omega/2} G_K(\omega) \tag{2a}$$

where[1]

$$G_K(\omega) = \frac{1}{K} \frac{\sin(K\omega/2)}{\sin(\omega/2)}. \tag{2b}$$

The delay is thus $(K - 1)/2$ samples, and $G_K(\omega)$ is characterized by the following properties. It provides $K - 1$ zeros on the unit circle, located at $\omega = \pm 2\pi k/K$ for $k = 1, 2, \cdots, \lfloor(K-1)/2\rfloor$ ($\lfloor x \rfloor$ stands for the integer part of x) and at $\omega = \pi$ for K even (see Fig. 2(a)). $G_K(\omega)$ is monotonically decaying from unity to zero in the interval $[0, 2\pi/K]$. On $[2\pi/K, \pi]$, $G_K(\omega)$ oscillates around zero, and the maximum deviation for zero occurs between the first two zeros and is approximately 0.22 regardless of the value of K.

A symmetric bandpass filter having two very narrow stopband regions and a rather small passband variation can be generated on the basis of $G_K(z)$ in the following three steps.
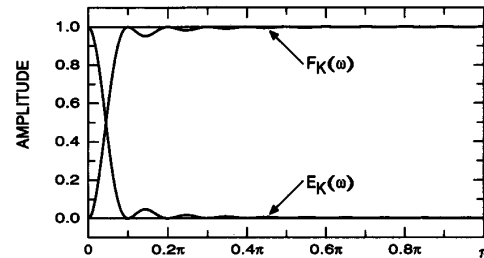
Step 1: Cascade two RRS filters of length K yielding the following transfer function:

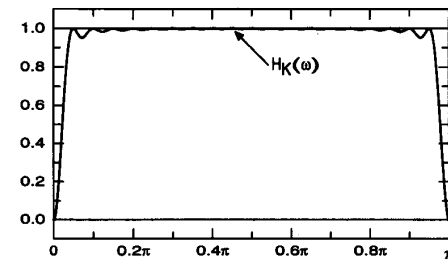$$E_K(z) = [G_K(z)]^2 = \left[\frac{1}{K} \frac{1 - z^{-K}}{1 - z^{-1}}\right]^2. \tag{3}$$

[1] For simplicity, we use $H(\omega)$ to denote the zero-phase response of $H(z)$ throughout this paper.
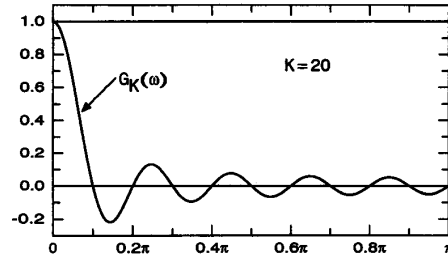
(a)

(b)

(c)

Fig. 2. Generation of the proposed filter $H_K(z)$ using a recursive running-sum filter $G_K(z)$ as a basic building block.

The frequency response of this filter is given by

$$E_K(e^{j\omega}) = e^{-j(K-1)\omega} E_K(\omega) \tag{4a}$$

where

$$E_K(\omega) = [G_K(\omega)]^2 = \left[\frac{1}{K} \frac{\sin(K\omega/2)}{\sin(\omega/2)}\right]^2. \tag{4b}$$

This $E_K(\omega)$ has double zeros at the frequencies where $G_K(\omega)$ has single zeros, making $E_K(\omega)$ nonnegative in the interval $[2\pi/K, \pi]$ (see Fig. 2(b)). The maximum value in this interval is approximately $(0.22)^2 \approx 0.048$.

Step 2: Subtract $E_K(z)$ from $z^{-(K-1)}$ yielding

$$F_K(z) = z^{-(K-1)} - E_K(z)$$

$$= z^{-(K-1)} - \left[\frac{1}{K} \frac{1 - z^{-K}}{1 - z^{-1}}\right]^2. \tag{5}$$

The frequency response of this filter is given by

$$F_K(e^{j\omega}) = e^{-j(K-1)\omega} F_K(\omega) \tag{6a}$$

where

$$F_K(\omega) = 1 - E_K(\omega) = 1 - \left[\frac{1}{K} \frac{\sin(K\omega/2)}{\sin(\omega/2)}\right]^2. \tag{6b}$$

From the above equation it follows that $F_K(z)$ and $E_K(z)$ are complementary filters, and $F_K(\omega)$ is monotonically increasing from zero to unity on $[0, 2\pi/K]$ and varies on $[2\pi/K, \pi]$ within unity and $1 - 0.048 = 0.952$ (see Fig. 2(b)).

Step 3: Replace z^{-1} by z^{-2} in $F_K(z)$ giving

$$H_K(z) = F_K(z^2) = z^{-2(K-1)} - \left[\frac{1}{K}\frac{1 - z^{-2K}}{1 - z^{-2}}\right]^2. \quad (7)$$

The resulting frequency response is

$$H_K(e^{j\omega}) = e^{-j2(K-1)\omega}H_K(\omega) \quad (8a)$$

where

$$H_K(\omega) = F_K(2\omega) = 1 - \left[\frac{1}{K}\frac{\sin(K\omega)}{\sin(\omega)}\right]^2. \quad (8b)$$

$H_K(\omega)$ is thus a frequency-axis compressed version of $F_K(\omega)$ such that the interval $[0, 2\pi]$ is shrunk onto $[0, \pi]$ (see Fig. 2(c)). $H_K(\omega)$ behaves on $[0, \pi/2]$ in the same manner as $F_K(\omega)$ on $[0, \pi]$. Because of the periodicity, $F_K(2\pi - \omega) = F_K(\omega)$. From this it follows that $H_K(\pi - \omega) = H_K(\omega)$ so that $H(\omega)$ is symmetric around $\omega = \pi/2$. Hence, $H_K(\omega)$ is monotonically increasing from zero to unity (decreasing from unity to zero) on $[0, \pi/K]([\pi - \pi/K, \pi])$, and it stays within the limits 0.952 and 1 on $[\pi/K, \pi - \pi/K]$. The frequency points on $[0, \pi/K]$ and $[\pi - \pi/K, \pi]$ where $H_K(\omega)$ achieves the value 0.952 are approximately $0.81\pi/K$ and $\pi - 0.81\pi/K$, respectively. If these points are regarded as passband edges, the resulting passband amplitude variation becomes 0.43 dB. For the maximum allowable variation of 0.5 dB, the passband region widens approximately to $[0.8\pi/K, \pi - 0.8\pi/K]$.

For the sampling rate of 100 Hz, the desired bandpass filter with passband edges of 0.5 Hz and 49.5 Hz is obtained by selecting $K = 80$. In this case, in terms of the normalized angular frequency $\omega = 2\pi f/f_s$, the edges are $\pi/100$ and $99\pi/100$ (f is the real frequency, and f_s is the sampling rate). In order to avoid a separate implementation of the delay term $z^{-2(K-1)}$, we rewrite $H_K(z)$ as given by (7) in the form

$$H_K(z) = z^{-2(K-1)} - A_K(z)B_K(z) \quad (9a)$$

where

$$A_K(z) = \tfrac{1}{4} + \tfrac{1}{2}z^{-K} + \tfrac{1}{4}z^{-2K} \quad (9b)$$

and

$$B_K(z) = \left[\frac{1}{K/2}\frac{1 - z^{-K}}{1 - z^{-2}}\right]^2. \quad (9c)$$

In the above, we have utilized the identity $1 - z^{-2K} = (1 + z^{-K})(1 - z^{-K})$. Overflows can be avoided in the overall filter by further rewriting $B_K(z)$ as

$$B_K(z) = \left[\frac{1}{64}\frac{1 - z^{-K}}{1 - z^{-2}}\right]\left[\frac{1}{32}\frac{1 - z^{-K}}{1 - z^{-2}}\right]\left[\frac{64 \cdot 32}{(K/2)^2}\right]. \quad (10)$$

The resulting filter implementation where the delay term $z^{-2(K-1)}$ and $A_K(z)$ share the same delays is shown in Fig. 3. It requires only one general multiplier of the value $64 \cdot 32/(K/2)^2 = 1.28$. This multiplier value can be quantized
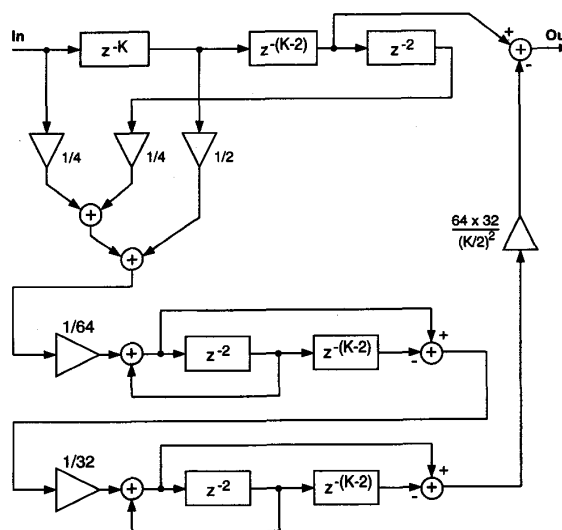

Fig. 3. Block diagram of the integrated circuit filter. $K = 80$.

to $1 + 2^{-2} + 2^{-5} = 1.28125$ without significantly affecting the amplitude response of the filter. The resulting response is given in Fig. 4. The abovementioned quantization has, in fact, a favorable effect. Instead of having (double) zeros at 0 Hz and 50 Hz, the resulting filter has zeros at 0.02 Hz and 49.98 Hz, and the attenuation at 0 Hz and 50 Hz is more than 60 dB. The resulting stopband regions are thus slightly wider which makes it possible to attenuate more effectively the wandering mains frequency interference (in Western European countries, the frequency variation is typically less than ±2%). It is interesting to observe from Fig. 4 that the largest passband ripples occur near the band edges, and these ripples become negligible in the center of the passband. The length of the impulse response of the above design is 317, which is only 13 percent higher than that (281) of the conventional minimax equivalent which requires 71 general multipliers.

It should be noted that if the sampling rate is increased to 200 Hz to preserve and analyze the whole clinically important frequency range of the ECG signal from fundamental heart rate to 80 Hz, the same design can be used. The only modification is to replace each unit delay in this design by double delay units. The resulting filter behaves in the frequency range from 0 to 50 Hz exactly in the same manner as the original design, as shown in Fig. 5. In the range from 50 Hz to 100 Hz, the behavior is the same as in the range 0 to 50 Hz because of the symmetry of the response around 50 Hz.

IV. CHIP DESIGN AND IMPLEMENTATION

The proposed filter is very attractive for VLSI implementation because it consists mainly of delays, adders, subtracters, and shift operations. It needs no general multipliers. Two's complement arithmetic is used; the word length of data input and output is 12 bits. Eighteen-bit accuracy is used internally in the recursive running-sum sections in order to keep the noise level generated in these sections below the dynamic range of
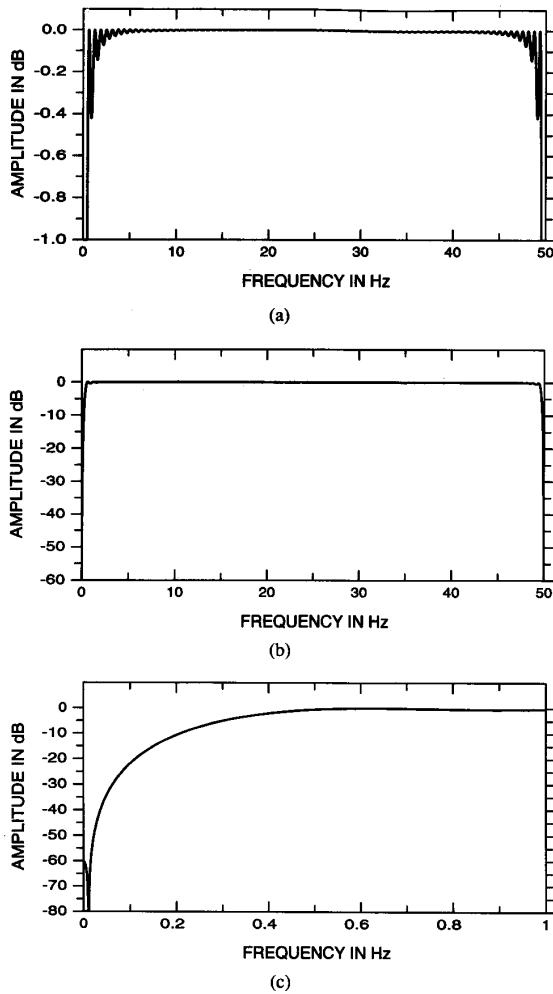
Fig. 4. Details of the amplitude response for the proposed design. (a) Passband details. (b) Overall response. (c) Low-frequency details.
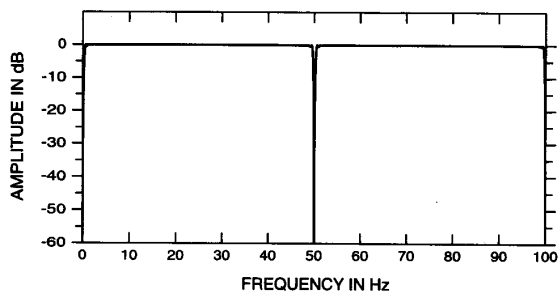

Fig. 5. The amplitude response obtained by replacing all unit delays in the proposed filter design with double delays and increasing the sampling rate to 200 Hz.

12-bit words. This corresponds to the situation where there is only one effective noise source at the filter output.

Because of the low clock rate, a bit-serial implementation would easily give a sufficient performance. In this case, however, the bit-parallel approach results in almost the same
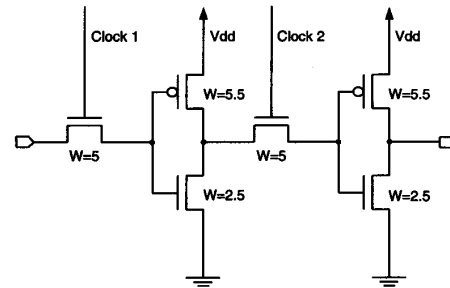

Fig. 6. Implementation of the dynamic two-phase clocked latch. All transistors have a length of $2.0\,\mu\text{m}$, and the widths are shown.

chip size, because more than half of the chip area is occupied by data storage registers, and their size is not affected by the arithmetic blocks. The bit-parallel implementation of the filter needs no control logic besides the two-phase clock and is therefore faster to design than a bit-serial structure. Furthermore, it was desired to have a bit-parallel chip interface. The bit-parallel design approach was thus adopted, utilizing module generators.

A. Circuit Elements

The hardware blocks of the ECG filter circuit are dynamic D-latches, adders, subtracters, buffers and a clock generator. The implementation technology is 2.0-μm CMOS with two metal layers. The target sampling rate of the filter is 100 Hz which is not too low for the dynamic two-phase clocked D-latches which were used in the delay lines. Each master-slave latch consists of two inverters and two n-type pass transistors, as shown in Fig. 6. In order to verify the proper functioning of the dynamic latches at 100 Hz, similar latches processed on the same technology were successfully tested for such low clock rates prior to finalizing this design.

The two nonoverlapping clock phases are generated from the master input clock by an internal clock generator. Strong clock buffering is used locally because of the heavy load caused by the large shift registers. Both clock phases are driving approximately 4800 n-transistors.

The power-of-two coefficients in the bit-parallel environment are implemented simply as hard-wired shifts. The coefficient $1.28125 = 1 + 2^{-2} + 2^{-5}$ is implemented with two shifts and two adders. The adders and subtracters are constructed by cascading single-bit full adder cells, which have been optimized for the area rather than for the speed.

B. Module Generators

The layout design was carried out in the GDT environment.[2] The software consists of the Led editor for schematic and layout editing, the mixed-mode simulator Lsim, the functional description compiler Mc, and some additional utilities. The structure and geometry of layouts and schematics are presented using the L-language in a universal L-database. Functional descriptions are written using another C-like language, the M-

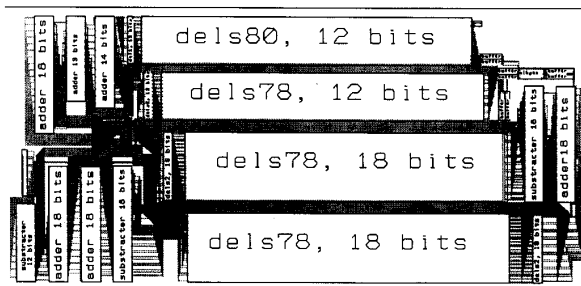[2] Generator Development Tools from Mentor Graphics.
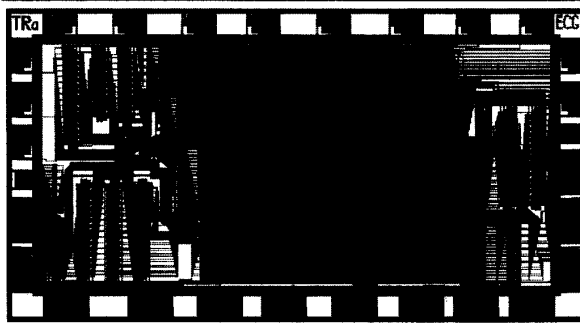
Fig. 7. Floor plan of the ECG filter chip.

Fig. 8. Chip layout plot.

(a)

(b)

Fig. 9. Simulated operation of the ECG filter. (a) Input signal, contaminated by a 50-Hz disturbance which is about 15 dB stronger than the primary ECG signal. (b) Filtered result.

language. With the Lsim simulator, it is possible to simulate the compiled M-language descriptions, netlists from Led, or a mixture of these and SPICE netlists. The L-language is also used for writing full custom generators which utilize manually optimized cells [6].

The following parameterized module generators were developed for implementing the functional units.

dels-G: Generator for bit-parallel delay lines. Its parameters are:

> *bits:* the number of parallel bits;
> *delays:* the number of unit delays;
> *M-model:* a flag indicating whether an M-language model is used in the simulation.

adder-G: Generator for bit-parallel ripple-carry adders with parameters:

> *bits:* the number of bits of the two input words;
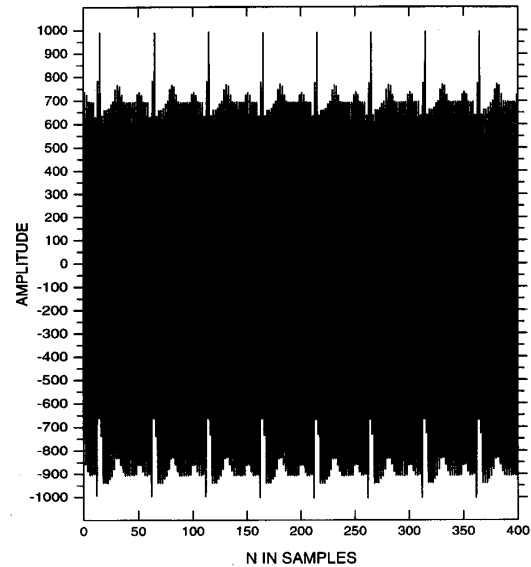> *m1m2:* a flag indicating whether both metal1 and metal2 layers are needed for the I/O terminals.

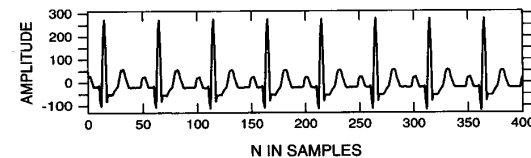*subtracter-G:* Generator for bit-parallel ripple-carry subtracters with parameters:

> *bits:* the number of bits of the two input words;
> *m1m2:* a flag indicating whether both metal1 and metal2 layers are needed for the I/O terminals.

These layout generators can be adapted to any double-metal CMOS technology by changing the basic cells which are used as layout building blocks. The wiring between blocks was generated using the channel router of GDT, utilizing a separate code in the L-language. The floor plan of Fig. 7 results after generating the layout of the main blocks and the wiring. A plot of the chip layout is shown in Fig. 8.

C. Design Verification

After completing the layout, the design rules were checked using the Lrc rule checking program. Functional simulations were carried out in order to verify the correct operation for sinusoidal inputs of different frequencies. This was done using a netlist extracted from the layout. In all cases, the steady-state output is reached after 317 clock cycles, corresponding to the length of the impulse response of the filter.

Furthermore, the amplitude response of the filter was computed from the simulated impulse response, and it was seen to be in accordance with the theoretical response.

Simulated operation of the filter with an ECG signal is demonstrated in Fig. 9. The input signal, contaminated by a strong 50-Hz component, is shown in Fig. 9(a), and the result of filtering is shown in Fig. 9(b).

D. The Chip

The chip was implemented on a 2.0-μm double-metal CMOS technology. It has about 34 000 transistors, and the chip size is 5.37 mm $\times$ 2.87 mm $= 15.43$ mm^2. The chip area is only about 25% of the area of a previously designed ECG filter chip [7]. The first silicon has been fabricated but did not work completely due to well problems.

The chip fits into a 28-pin package since it only needs the 12-bit input and output interfaces, V_{dd}, ground, and the clock

input. One pin is left for a test output, taken from a shift register.

V. CONCLUSIONS AND DISCUSSION

A digital filter chip for ECG signal processing has been designed. The small chip area is a result of using a computationally efficient architecture based on recursive running-sum structures. The proposed design concept is suitable to a module-generator-based VLSI design.

The filter was designed for the sampling rate of 100 Hz. If the sampling rate of 200 Hz is desired to be used, then the same design can be used by replacing each unit delay by two unit delays. This modification would increase the overall silicon area from $15.43 \, \text{mm}^2$ to approximately $21 \, \text{mm}^2$. Another alternative is to apply switching and two chips in such a way that every second sample is fed into one chip and the remaining samples into another chip. The desired overall output is then obtained by interlacing the outputs of the two chips.

In order to create a complete ECG signal-processing system, we also need an A/D converter with good linearity and negligible phase distortion. The oversampled sigma-delta modulator-based approach is recommended for the A/D conversion [9]. In a sigma-delta converter, a linear-phase decimating digital filter can be used for attenuating high-frequency components of the input signal, acting also as an antialias filter. Because of oversampling, the requirements on analog antialias filtering are relaxed (in most cases, a first-order RC filter is sufficient). VLSI-implementable sigma-delta converters with a very narrow transition band are available; thus the baseband signals are not attenuated in the conversion process.

Besides ECG signal processing, the proposed digital filter design and construction approach could be applied also to other kinds of biomedical signal processing where bandpass filtering is needed, for instance, electromyogram (EMG) measurements.

REFERENCES

[1] P. Le-Huy, E. Yvroud, and J.-L. Dion, "A versatile cardiac arrhythmia simulator," *IEEE Trans. Instrum. Meas.*, vol. IM-36, pp. 934–939, Dec. 1987.

[2] U. C. Niranjan and I. S. N. Murthy, "ECG component delineation by Prony's method," *Signal Processing*, vol. 31, pp. 191–202, Mar. 1993.

[3] K.-P. Estola and T. Saramäki, "Efficient microprocessor-based linear-phase FIR filters for ECG-signal processing," in *Proc. 6th European Conf. Circuit Theory and Design*, Stuttgart, FRG, Sept. 1983, pp. 420–422.

[4] K.-P. Estola and T. Saramäki, "High performance FIR filters without multipliers for ECG signal processing," in *Proc. 1984 IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Montreal, Canada, May 1984, pp. 1462–1465.

[5] T. Saramäki, Y. Neuvo, and S. K. Mitra, "Design of computationally efficient interpolated FIR filters," *IEEE Trans. Circuits Syst.*, vol. CAS-35, pp. 70–88, Jan. 1988.

[6] J. Nurmi, T. Karema, O. Vainio, and H. Tenhunen, "Generator development techniques supporting efficient DSP system design," in *Proc. NORSILC/NORCHIP'90 Seminar*, Lund, Sweden, Oct. 1990.

[7] M. Williams, J. Nurmi, and H. Tenhunen, "ASIC design of digital ECG filter," in *Proc. 2nd Annu. IEEE ASIC Seminar and Exhibit*, Rochester, NY, Sept. 1989, pp. P7-3.1–P7-3.4.

[8] L. T. Sheffield *et al.*, "Recommendations for standards of instrumentation and practice in the use of ambulatory electrocardiography," *Circulation*, vol. 71, no. 3, Mar. 1985.

[9] J. C. Candy and G. C. Temes, Eds., *Oversampled Sigma Delta Converters: Theory, Design, and Simulations*. Piscataway, NJ: IEEE Press, 1991.
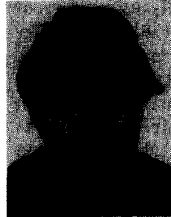
Tommi Raita-aho received the degree of Diploma Engineer from the Tampere University of Technology, Tampere, Finland, in 1993.

He is currently working on a research project in the Signal Processing Laboratory, Tampere University of Technology. His research interests are in module generator based VLSI circuit design and DSP core processor implementations.

Tapio Saramäki was born in Orivesi, Finland, on June 12, 1953. He received the Diploma Engineer degree (with honors) and the Doctor of Technology degree (with honors) in electrical engineering from the Tampere University of Technology, Tampere, Finland, in 1978 and 1981, respectively.

Since 1977 he has been with the Department of Electrical Engineering, Tampere University of Technology. From 1979 to 1981 he served as a Research Assistant and from 1982 to 1986 as a Research Fellow, both financed by the Academy of Finland. Since 1987, he has held various research and teaching positions at the Tampere University of Technology. Currently he is a Docent of Teleommunications and an Assistant Professor of Signal Processing. He is also a founding member and a system-level designer of VLSI Solution Oy specializing in VLSI implementations of sigma-delta modulators and signal processing algorithms. In 1982, 1985, 1986, and 1990 he was a Visiting Research Scholar at the University of California, Santa Barbara. He has authored 100 international journal and conference articles. In 1987 he received the Guillemin-Cauer Award for the best paper of the *IEEE Transactions on Circuits and Systems*. His research interests are in the areas of digital signal processing, approximation theory, and VLSI implementations of signal processing algorithms. He is a founding member of the Median-Free Group International.
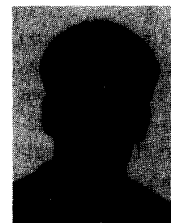
Olli Vainio (S'84–M'88) received the degrees of Diploma Engineer and Doctor of Technology in electrical engineering from the Tampere University of Technology, Tampere, Finland, in 1984 and 1988, respectively.

Since 1983 he has held research and teaching positions at the Tampere University of Technology and the Academy of Finland. During 1986–1987 he was a Visiting Scholar at the University of California, Santa Barbara. He now holds the position of Senior Research Fellow of the Academy of Finland. His research interests are in VLSI signal processing and microelectronics.

Recursive Implementation of FIR Differentiators with Optimum Noise Attenuation

Olli Vainio, *Senior Member, IEEE*, Markku Renfors, *Senior Member, IEEE*, and Tapio Saramäki

Abstract—We introduce a computationally efficient recursive implementation of digital finite impulse response (FIR) filters for estimating the rate of change or slope of digitized signals. The proposed FIR differentiator is characterized by the optimal attenuation of white noise and an efficient suppression of upper-band frequencies. The basic structure needs only one multiplier, which becomes a power of two with an appropriate selection of the length of the impulse response. The structure does not need resetting and recovers from any bit errors. For long filters, sampling rate reduction by decimation gives further computational savings.

Index Terms—Differentiation, digital filter wordlength effects, finite impulse response (FIR) digital filters.

I. INTRODUCTION

DIFFERENTIATION of a signal gives an estimate of its instantaneous rate of change or slope. Such a task is frequently encountered in digital signal processing, including instrumentation and measurement applications. For instance, in biomedical engineering, differentiation is used in motion analysis and in ventricular pressure measurements [1], [2].

The ideal differentiator would have a magnitude response which is proportional to the frequency [3]. This can be seen by differentiating the signal $\sin n\omega$ with respect to n, giving the signal $\omega \cos n\omega$. The ideal discrete-time differentiator has the following frequency response:

$$H(e^{j\omega}) = j\omega, \quad |\omega| \leq \pi. \tag{1}$$

The output for every bandlimited input would be the derivative of the input. In practical implementations, the ideal differentiator must be approximated because the ideal impulse response would be noncausal and of infinite length. One possible approximation is the first-order difference operation where the input–output relation is simply given by $y(n) = x(n) - x(n - 1)$. The corresponding frequency response is given by

$$H(e^{j\omega}) = 1 - \cos \omega + j \sin \omega \tag{2}$$

which closely approximates that of (1) for small frequencies, but is significantly different when ω approaches π. When more samples are included in the approximation, different design objectives can be targeted.

Differentiator approximations have been presented for two classes of signals: piecewise polynomial (time domain constraint) and bandlimited signals (frequency domain constraint). Design approaches used earlier include the error minimization at low frequencies [4], [5], minimax or equiripple designs [6], least-mean-square error designs [7], and differentiation based on stochastic models [8]. Infinite impulse response (IIR) filter-based halfband differentiator designs are described in [9].

The ideal differentiator, as well as many of its approximations, has highpass characteristics, which causes undesirable noise amplification. This is particularly disturbing in applications where the signal is contaminated by high noise. The objective of this paper is to enhance the first-order difference by expanding it to a finite impulse response (FIR) filter, which is optimized for attenuating white noise as well as possible. This leads to a differentiator filter which closely approximates the ideal differentiator at low frequencies but attenuates the upper-band frequencies. Both the stopband width and attenuation depend on the filter length, as does the noise attenuation. The resulting filter is suitable to estimating the signal slope even under noisy conditions. Furthermore, our approach allows a very efficient implementation structure that needs only one multiplier. This single multiplier becomes a power of two with an appropriate selection of the length of the impulse response. The structure does not need resetting and recovers from any bit errors. For long filters, sampling rate reduction by decimation gives further computational savings, especially in the number of delay elements required in the implementation. The resulting filter structure has a potential of being implemented as a VLSI circuit with a very small silicon area.

II. DERIVATION OF THE FILTER COEFFICIENTS

The task is to design an FIR differentiator for estimating the rate of change or slope of the input signal while providing the optimum white noise attenuation. Assuming that the additive noise has a flat power spectrum, the noise power gain is given by

$$F = \sum_{n=0}^{N-1} h^2(n) \tag{3}$$

where $h(n)$, $n = 0, \ldots, N - 1$, are the tap coefficients of the FIR filter and N is the filter length [10]. The overall optimization problem is to minimize the quantity F with respect to the tap coefficients subject to the following two

constraints:

$$g_0 = \sum_{n=0}^{N-1} h(n) = 0 \qquad (4)$$

and

$$g_1 = \sum_{n=0}^{N-1} nh(n) = -1. \qquad (5)$$

Constraint (4) means that filter response at the zero frequency is equal to zero. Thus a constant level at the filter input produces a zero output. Constraint (5), in turn, means that the filter produces a unity response in the case of the input ramp signal, $x(n) = n$.

Optimization of the above-constrained problem can be conveniently carried out using the method of Lagrange multipliers. This is a technique which has been often used to solve for FIR filter coefficients in closed form, including differentiator designs [5]. When applying this technique, the problem is to minimize the following function:

$$L(h(0), h(1), \ldots, h(N-1), \lambda_0, \lambda_1) = F + \lambda_0 g_0 + \lambda_1 g_1 \quad (6)$$

where λ_0 and λ_1 are the Lagrange coefficients. The minimum of this function is found by setting the partial derivatives with respect to all of the arguments equal to zero. As the first step, we have

$$\frac{\partial L}{\partial h(n)} = 2h(n) + \lambda_0 + \lambda_1 n = 0. \qquad (7)$$

This is solved for $h(n)$, which is substituted into (4) and (5). The resulting pair of equations is then solved for λ_0 and λ_1, which are substituted into (7), finally giving

$$h(n) = \frac{6}{N(N+1)}\left(1 - \frac{2n}{N-1}\right), \quad n = 0, \ldots, N-1. \quad (8)$$

The same solution could be found using linear regression [11]. The terms of the impulse response are found on a straight line, as shown in Fig. 1, which also shows the corresponding amplitude response. Depending on whether N is odd or even, the differentiator is, respectively, a Type III or a Type IV linear-phase filter [10] with antisymmetric impulse response.
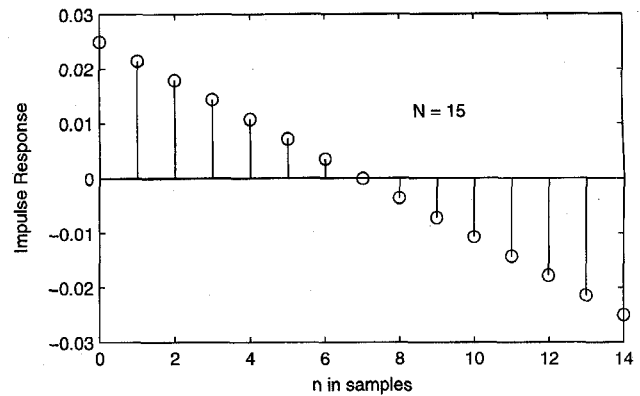
For an efficient implementation of the filter, the tap coefficients are scaled to integers. For N odd, we multiply the impulse-response coefficient values from (8) by $N(N^2-1)/12$ yielding

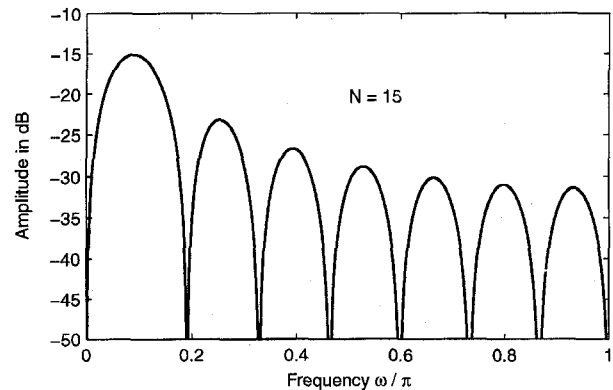$$h(n) = \frac{N-1}{2} - n, \quad n = 0, \ldots, N-1. \qquad (9)$$

The resulting FIR differentiator filter implements the algorithm

$$y(n) = S \sum_{k=0}^{N-1} h(k)x(n-k) \qquad (10)$$

where the scaling factor S depends on the maximum slope of the input signal, word length, filter length, and the desired maximum output level, as will be explained in the following.

Fig. 1. Impulse and amplitude responses for the optimal FIR differentiator with $N = 15$.

III. RECURSIVE IMPLEMENTATION

Recursive structures provide efficient implementations for some FIR filters [12]–[14]. This section describes a dedicated structure which implements the FIR differentiator approximation derived above.

The proposed recursive structure is shown in Fig. 2. The structure can be understood on the basis of the well-known recursive running-sum structure [15]–[17], which implements an averaging filter where the impulse response terms are equal-valued. The accumulator on the left-hand side of Fig. 2 and the extra summing nodes have the effect of converting the impulse response to a ramp such as that shown in Fig. 1. This is shown in the following, where we use $u(n)$ to denote the unit step signal [10].

Ignoring the scaling coefficients S_1 and S_2, the impulse responses from the input to the outputs of the branches labeled A, B, and C in Fig. 2 are given by

$$h_A(n) = \frac{N+1}{2}u(n) \qquad (11a)$$

$$h_B(n) = \frac{N+1}{2}u(n-N) + (n+1-N)u(n-N) - u(n-N) \qquad (11b)$$
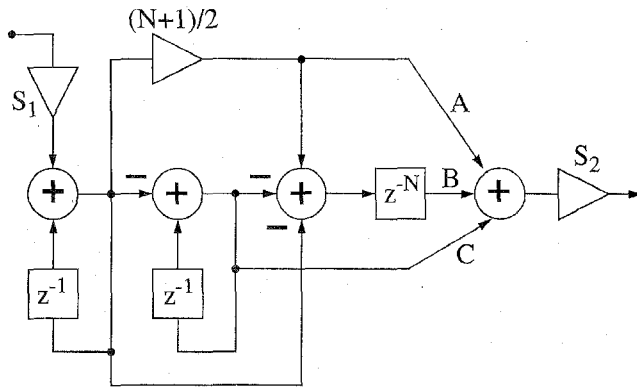
and

$$h_C(n) = -(n+1)u(n) \qquad (11c)$$

Fig. 2. Recursive implementation of the proposed FIR differentiator.

Fig. 3. Noisy test signal.

respectively. Combining these terms, the composite impulse response takes the following form:

$$h(n) = \frac{N+1}{2}u(n) - (n+1)u(n) + \frac{N-1}{2}u(n-N) \\ + (n+1-N)u(n-N) \tag{12}$$

which corresponds to the coefficient values of (9).

For large values of N, this structure gives significant computational savings when compared with the direct-form FIR structure since the number of arithmetic operations does not depend on N. The coefficient $(N+1)/2$ is an integer if N is odd. In many cases $(N+1)/2$ can be chosen to be a power of two, in which case the multiplication becomes a simple arithmetic shift.

Except for the scaling coefficients S_1 and S_2, the arithmetic operations include only additions, subtractions, and an integer multiplication (for N odd). This guarantees that the recursive structure of Fig. 2 can be made working in the desired manner provided that modulo arithmetic (for example, one's or two's complement arithmetic) is used. This arithmetic has the useful property that when several numbers are added together, overflows are allowed in intermediate additions as long as the final result is within the desired range. On the other hand, multiplication by an integer-valued coefficient corresponds to additions. Thus the structure of Fig. 2 works properly if the signal after multiplication by S_1 is simply scaled in such a way that the signal before multiplying by S_2 is within the given range. Coefficient S_2 is only used for adjusting the output scale. Another attractive feature is that the proposed structure does not need initial resetting and it automatically recovers in a finite time from temporary miscalculations which may be caused by power surges, system start-up, etc.

The largest absolute signal level encountered at the input of S_2 is given by

$$w_{\max} = S_1|\alpha|\frac{N(N^2-1)}{12} \tag{13}$$

where α is the largest allowed positive or negative input slope. This level is achieved for the input signal given by $x(n) = \alpha n$, where α is either positive or negative. For a fixed-point $(1+b)$-bit arithmetic, S_1 is set such that
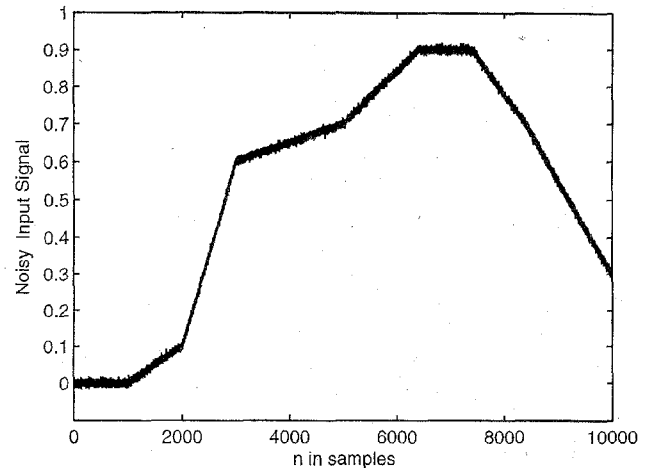
$$w_{\max} \leq 1 - 2^{-b} \tag{14}$$

in order to guarantee the desired operation of the proposed structure, yielding

$$S_1 \leq \frac{12(1-2^{-b})}{|\alpha|N(N^2-1)}. \tag{15}$$

S_2, in turn, is selected to be

$$S_2 = \frac{|\beta|}{w_{\max}} \tag{16}$$

where β is the desired output value corresponding to the largest absolute rate of change of the input signal.

IV. ILLUSTRATIVE EXAMPLE

This section illustrates, by means of an example, the attractive properties of the proposed recursive differentiator.

We consider an input signal $x(n)$ consisting of straight lines going through nine corner points $(n_k, x(n_k))$ with $n_1 = 0$, $x(n_1) = 0$; $n_2 = 1000$, $x(n_2) = 0$; $n_3 = 2000$, $x(n_3) = 0.1$; $n_4 = 3000$, $x(n_4) = 0.6$; $n_5 = 5000$, $x(n_5) = 0.7$; $n_6 = 6400$, $x(n_6) = 0.9$; $n_7 = 7400$, $x(n_7) = 0.9$; $n_8 = 8400$, $x(n_8) = 0.7$; and $n_9 = 10000$, $x(n_9) = 0.3$. This signal is contaminated by white Gaussian noise with deviation of 0.005, as shown in Fig. 3. The highest slope occurring between the corner points $(2000, 0.1)$ and $(3000, 0.6)$ is $\alpha = 0.0005$.

Fig. 4 shows the outputs of the proposed scaled differentiators with $N = 63$, $N = 127$, and $N = 255$. In these cases, no actual multipliers are needed in the implementation. For these differentiators, $S = S_1 S_2 = 12/[|\alpha|N(N^2-1)]$ with $\alpha = 0.0005$ [cf., (10), (15), and (16)]. This makes the output for the highest signal slope $\alpha = 0.0005$ equal to $\beta = 1$. As seen from Fig. 4, the noise attenuation is improved by increasing the filter length at the expense of wider transitions between the constant output levels. The impulse and amplitude responses for $N = 63$ and $N = 255$ are shown in Figs. 5 and 6, respectively.
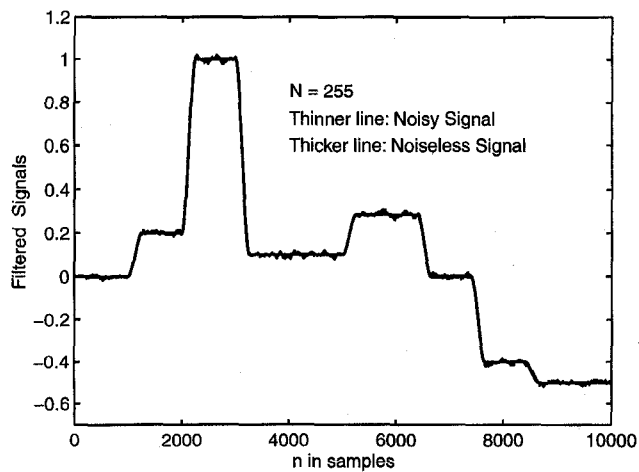
The signal-to-noise ratios (SNR's), as given by $10\log_{10}(1/F)$ with F defined by (1), for the these unscaled differentiators with impulse responses given by (8) are 43.2, 52.3, and 61.4 dB, respectively. Notice, however, that due
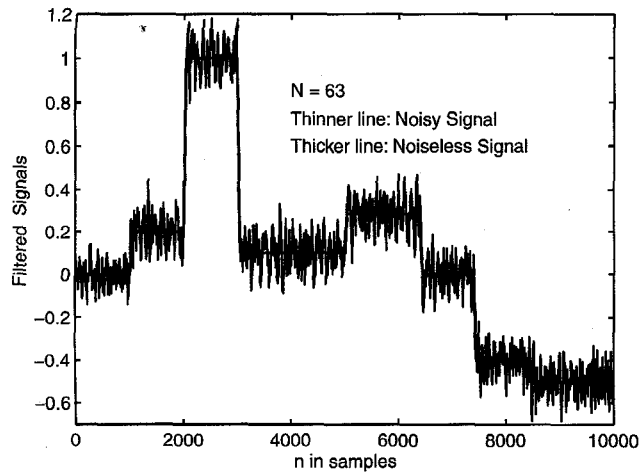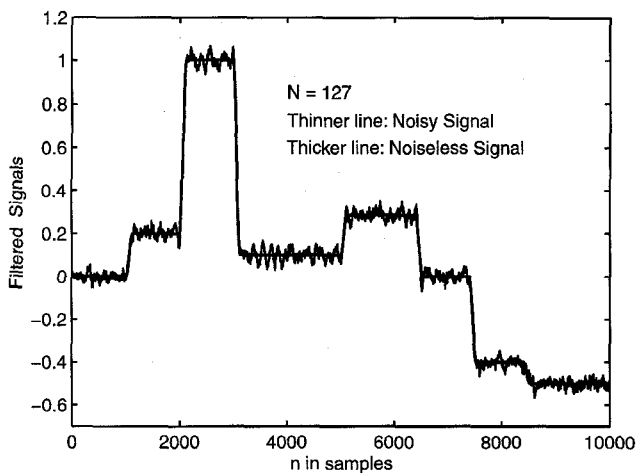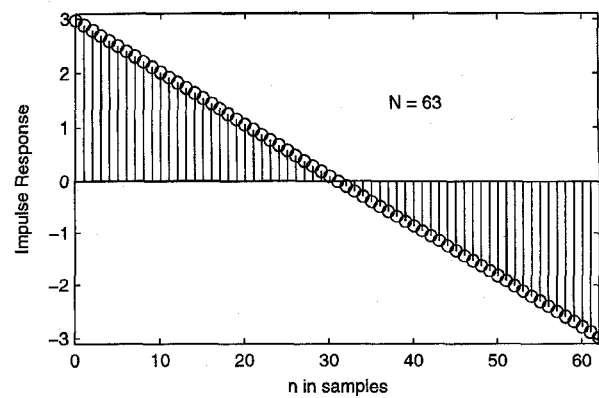
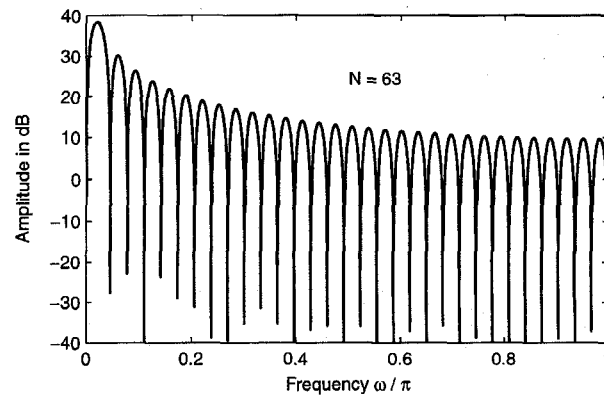Fig. 4. Filtered output signals for the scaled proposed differentiators of different orders.

Fig. 5. Impulse and amplitude responses for the scaled differentiator of length 63 used for filtering the example test signal.
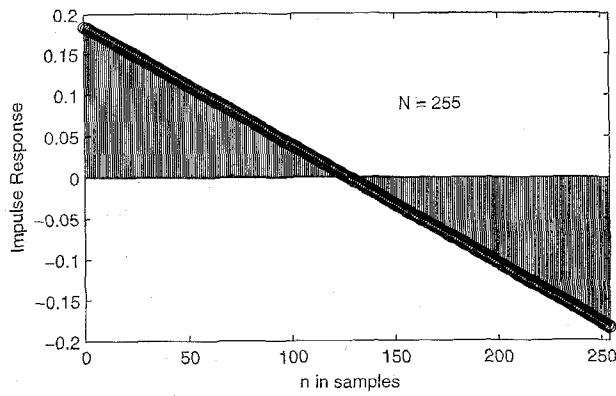
using the Remez multiple exchange algorithm [10], [16] with the desired and weighting functions being $D(\omega) = \omega$ and $W(\omega) = 1/\omega$, respectively. The scaled filter is obtained by multiplying the resulting coefficient values by $1/\alpha = 2000$ to produce the output value of unity for the slope of $\alpha = 0.0005$. The output of this filter is shown in Fig. 7. As seen from this figure, the output noise dominates the desired output signal, making it totally invisible. This clearly shows the efficiency of the proposed differentiator over the minimax design.
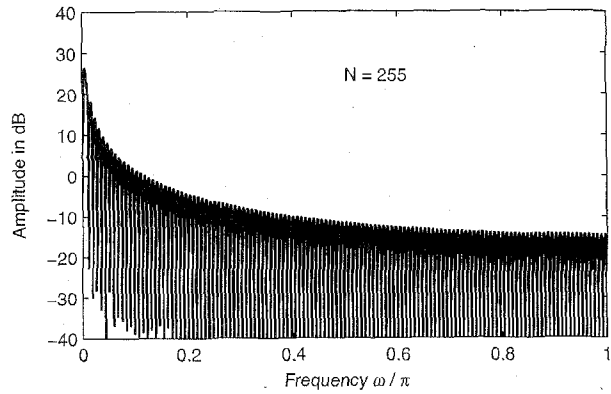
V. DECIMATION

If not all the output samples are needed, then the number of delays in the overall differentiator implementation can be reduced and the dynamics increased by lowering the sampling rate using decimation before the differentiator part. This also leads to savings in the computational workload and allows more relaxed specifications for antialias filtering. On the other hand, the bandwidth of accurate differentiator approximation is reduced by the decimation factor.

For sampling rate reduction by an integer factor of M, a suitable decimator is a cascade of two averaging filters of length M having the following transfer function:

$$D_M(z) = \left[\sum_{n=0}^{M-1} z^{-n} \right]^2 = \left[\frac{1 - z^{-M}}{1 - z^{-1}} \right]^2. \qquad (17)$$

to the scaling used in this example, the noise gain in each case is 66 dB-SNR.

For comparison purposes, the same input data has been filtered by the optimum minimax linear-phase differentiator of length 256. This filter has been first optimized for $0 \leq \omega \leq \pi$

(a)

(b)

Fig. 6. Impulse and amplitude responses for the scaled differentiator of length 255 used for filtering the example test signal.
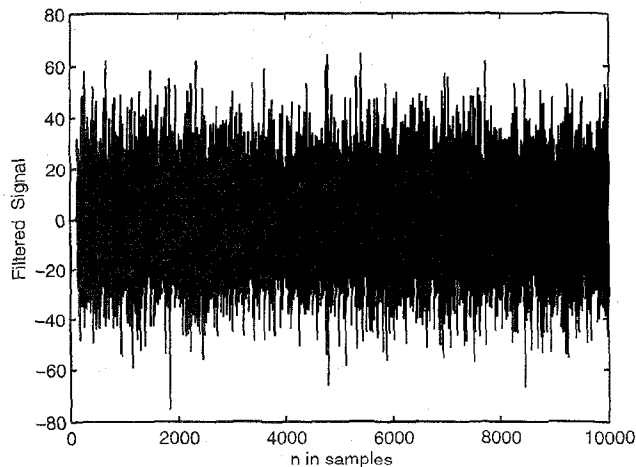

Fig. 7. Filtered output signals for the scaled minimax differentiator of length 256.

An efficient implementation for this decimator is shown in Fig. 8 [18].

If after the decimation the proposed differentiator with transfer function $H(z)$ is used, then the overall system is equivalent to the system where the input signal is first filtered by the single-stage filter with transfer function
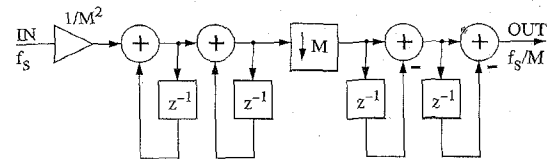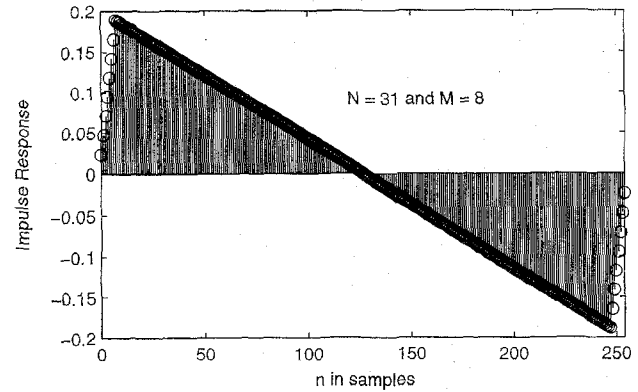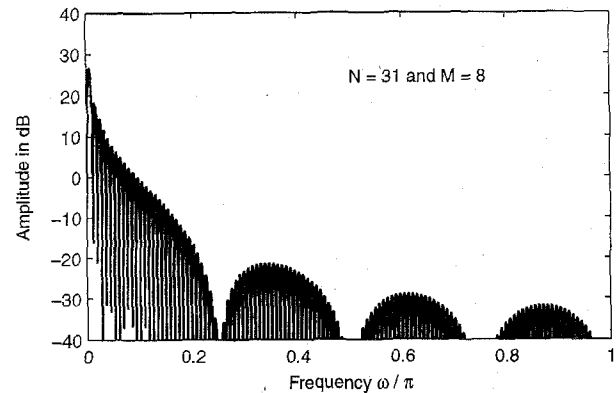
$$F(z) = H(z^M)D_M(z) \qquad (18)$$

Fig. 8. A decimating filter.

(a)

(b)

Fig. 9. Impulse and amplitude responses for $F(z)$ given by (17) and (18) with $N = 31$ and $M = 8$.

and then every Mth output is evaluated. If the length of $H(z)$ is N, then the overall length of $F(z)$ is $M(N + 1) - 1$. The suitability of the above $D_M(z)$ to decimation purposes is based on the fact that the impulse response of $F(z)$ follows a straight line except for the very beginning and end of the impulse response. This is illustrated in Fig. 9, which shows the impulse and amplitude responses for $F(z)$ for $N = 31$ and $M = 8$. The length of the overall $F(z)$ is 255. When comparing these responses with the corresponding responses for $N = 255$, shown in Fig. 6, two basic facts are observed. First, the impulse responses are very similar except for the very beginning and end of the response. Second, at low frequencies, the amplitude responses are very similar. This suggests that the filtering performances of these two alternatives are also quite similar. This is, in fact, true when comparing the filtering results for $N = 255$ in Fig. 4 with those of Fig. 10. Fig. 10 shows the filtered output data in the case where decimation is not actually performed. In terms
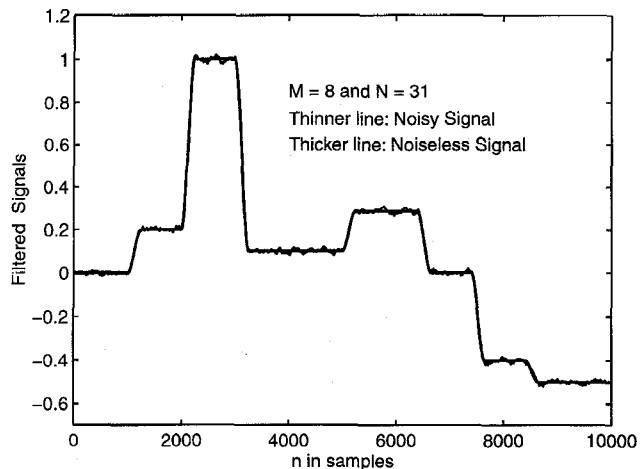
Fig. 10. Filtered output signals for the differentiator with impulse and amplitude responses shown in Fig. 9. In the actual implementation using decimation, only every Mth output is evaluated.

of noise attenuation capability, the cascade of a decimator with decimation rate M and a differentiator of length N thus approximately corresponds to a differentiator of length $M(N+1) - 1$. For scaling purposes, decimation increases the parameter α by a factor of M.

VI. CONCLUSION

An efficient and practical implementation structure was proposed for FIR differentiator approximations with optimal white noise attenuation. The structure is attractive on microcontrollers, ASIC's, and other platforms with limited arithmetic capacity.

REFERENCES

[1] S. Usui and I. Amidror, "Digital low-pass differentiation for biological signal processing," *IEEE Trans. Biomed. Eng.*, vol. BME-29, pp. 686–693, Oct. 1982.

[2] A. E. Marble, C. M. McIntyre, R. Hastings-James, and C. W. Hor, "A comparison of digital algorithms used in computing the derivative of left ventricular pressure," *IEEE Trans. Biomed. Eng.*, vol. BME-28, pp. 524–529, July 1981.

[3] P. A. Lynn and W. F. Fuerst, *Introductory Digital Signal Processing with Computer Applications.* Chichester, UK: Wiley, 1994, pp. 159–163.

[4] B. Kumar and S. C. Dutta Roy, "Design of digital differentiators for low frequencies," *Proc. IEEE*, vol. 76, pp. 287–289, Mar. 1988.

[5] G. W. Medlin and J. W. Adams, "A new design technique for maximally linear differentiators," in *Proc. 1989 IEEE Int. Conf. Acoustics, Speech, Signal Processing*, Glasgow, Scotland, May 1989, pp. 825–828.

[6] L. R. Rabiner and R. W. Schafer, "On the behavior of minimax relative error FIR digital differentiators," *Bell Syst. Tech. J.*, vol. 53, pp. 333–361, Feb. 1974.

[7] S. Sunder and V. Ramachandran, "Design of equiripple nonrecursive digital differentiators and Hilbert Transformers using a weighted least-squares technique," *IEEE Trans. Signal Processing*, vol. 42, pp. 2504–2509, Sept. 1994.

[8] B. Carlsson, A. Ahlen, and M. Sternad, "Optimal differentiation based on stochastic signal models," *IEEE Trans. Signal Processing*, vol. 39, pp. 341–353, Feb. 1991.

[9] R. Pintelon and J. Schoukens, "Real-time integration and differentiation of analog signals by means of digital filtering," *IEEE Trans. Instrum. Meas.*, vol. 39, pp. 923–927, Dec. 1990.

[10] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing.* Englewood Cliffs, NJ: Prentice-Hall, 1989, App. A.

[11] E. Kreyszig, *Advanced Engineering Mathematics.* New York: Wiley, 1983.

[12] G. F. Boudreaux and T. W. Parks, "Thinning digital filters: A piecewise-exponential approximation approach," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-31, pp. 105–113, Feb. 1983.

[13] S. Chu and C. S. Burrus, "Efficient recursive realizations of FIR filters, part I: The filter structures," *Circuits Syst. Signal Process.*, vol. 3, no. 1, pp. 3–20, 1984.

[14] T. Saramäki and A. T. Fam, "Properties and structures of linear-phase FIR filters based on switching and resetting of IIR filters," in *Proc. 1990 IEEE Int. Symp. Circuits Systems*, New Orleans, LA, May 1990, pp. 3275–3278.

[15] T. Saramäki, Y. Neuvo, and S. K. Mitra, "Design of computationally efficient interpolated FIR filters," *IEEE Trans. Circuits Syst.*, vol. 35, pp. 70–88, Jan. 1988.

[16] T. Saramäki, "Finite impulse response filter design," Chapter 4 in *Handbook for Digital Signal Processing*, S. K. Mitra and J. F. Kaiser, Eds. New York: Wiley, 1993, pp. 155–277.

[17] T. Raita-aho, T. Saramäki, and O. Vainio, "A digital filter chip for ECG signal processing," *IEEE Trans. Instrum. Meas.*, vol. 43, pp. 644–649, Aug. 1994.

[18] T. Saramäki and H. Tenhunen, "Efficient VLSI-realizable decimators for a sigma-delta analog-to-digital converter," in *Proc. 1988 IEEE Int. Symp. Circuits Systems*, Espoo, Finland, June 1988, pp. 1525–1528; reprinted in *Oversampling Delta-Sigma Data Converters: Theory, Design, and Simulations*, J. C. Candy and G. C. Temes, Ed. New York: IEEE Press, 1991, pp. 471–474.

Olli Vainio (S'84-M'88-SM'94), for a photograph and biography, see this issue p. 1201.

Markku Renfors (S'77-M'82-SM'90) was born in Suoniemi, Finland, on January 21, 1953. He received the Diploma Engineer, Licentiate of Technology, and Doctor of Technology degrees from Tampere University of Technology (TUT) in 1978, 1981, and 1982, respectively.

He held various research and teaching positions at TUT from 1976 to 1988. From 1988 to 1991, he was a Design Manager in the area of video signal processing, specifically HDTV, at Nokia Research Centre and Nokia Consumer Electronics. Since 1992 he has been a Professor of Telecommunications at TUT. His main research area is signal processing algorithms for digital receivers.

Tapio Saramäki was born in Orivesi, Finland, on June 12, 1953. He received the Diploma Engineer degree (with honors) and the Doctor of Technology degree (with honors) in electrical engineering from Tampere University of Technology (TUT), Tampere, Finland, in 1978 and 1981, respectively.

Since 1977, he has been with the Department of Electrical Engineering, TUT. From 1979 to 1981, he was a Research Assistant, and from 1982 to 1986, a Research Fellow; both positions were financed by the Academy of Finland. Since 1987, he has held various research and teaching positions at TUT. Currently, he is a Docent of Telecommunications and an Associate Professor of Signal Processing, as well as the Head of the Signal Processing Laboratory, TUT. He is also a cofounder and a System-Level Designer with VLSI Solution Oy, specializing in VLSI implementations of sigma–delta modulators and signal processing algorithms. In 1982, 1985, 1986, and 1990, he was a Visiting Research Scholar at the University of California, Santa Barbara. He has written more than 100 international journal and conference articles, three international book chapters, and holds four patents. His research interests are in the areas of digital signal processing, approximation theory, and VLSI implementations of signal processing algorithms.

Dr. Saramäki received the 1987 Guillemin-Cauer Award for the best paper in the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS. He is a founding member of the Median-Free Group International.