EURASIP Journal on Applied Signal Processing 2005:18, 3015–3025

A Low-Power Two-Digit Multi-dimensional Logarithmic Number System Filterbank Architecture for a Digital Hearing Aid

Roberto Muscedere
Research Centre for Integrated Microsystems (RCIM), University of Windsor, ON, Canada N9B 3P4
Email: rmusced@uwindsor.ca

Vassil Dimitrov
Advanced Technology Information Processing Systems (ATIPS) Laboratory, University of Calgary, AB, Canada T2N 1N4
Email: dimitrov@atips.ca

Graham Jullien
Advanced Technology Information Processing Systems (ATIPS) Laboratory, University of Calgary, AB, Canada T2N 1N4
Email: jullien@atips.ca

William Miller
Research Centre for Integrated Microsystems (RCIM), University of Windsor, ON, Canada N9B 3P4
Email: wmiller@uwindsor.ca

Received 30 April 2004; Revised 7 December 2004

This paper addresses the implementation of a filterbank for digital hearing aids using a multi-dimensional logarithmic number system (MDLNS). The MDLNS, which has similar properties to the classical logarithmic number system (LNS), provides more degrees of freedom than the LNS by virtue of having two, or more, orthogonal bases and the ability to use multiple MDLNS components or digits. The logarithmic properties of the MDLNS also allow for reduced complexity multiplication and large dynamic range, and a multiple-digit MDLNS provides a considerable reduction in hardware complexity compared to a conventional LNS approach. We discuss an improved design for a two-digit 2D MDLNS filterbank implementation which reduces power and area by over two times compared to the original design.

Keywords and phrases: logarithmic number system, double-base number system, multi-dimensional logarithmic number system, filterbank, low power, hearing aids or instruments.

1. INTRODUCTION

Digital signal processing for hearing aids is providing possibilities for new signal processing strategies to compensate for hearing loss [1]. Hearing loss compensation in a typical digital hearing instrument is performed by separating the input signal into multiple frequency bands which are then compressed to allow the amplification of low-level signals while maintaining the amplitude of high-level signals. We therefore require a processor that is able to both perform linear processing (band separation) and nonlinear processing (signal compression). In order to be able to adequately represent the very low-level signals that are subject to the maximum amplification in the processor, very large word lengths are required, and floating-point representation is quite usual in this regard [2]. To be practically usable in a completely-in-canal (CIC) device [3], the digital circuitry needs to fulfill the joint requirements of low-power consumption and small size. The multi-dimensional logarithmic number system (MDLNS) is a recently developed number system [4] that appears to be a good candidate for implementing hearing instrument processors. Although the logarithmic number system (LNS) [5] has been previously considered for digital hearing-aid processors [6], this research presents an exploration of the MDLNS for digital hearing-aid circuitry. As with the LNS, the MDLNS provides a reduction in the size of the number representation, but the MDLNS promises a lower-cost (area × power) implementation of the arithmetic operations required in both the linear and nonlinear domains of filtering and compression. In this research, we apply the MDLNS to the construction of a finite impulse response (FIR) filterbank; a major component of any digital hearing-aid processor. Most binary implementations of filterbanks for hearing instruments either use a modulated DFT or interpolated FIR filter (IFIR) approach to perform the signal separation because they reduce the number of multiplications. With MDLNS a binary multiplication component is never used, only addition/subtraction components. Therefore, a simple FIR filter structure can be easily implemented in the MDLNS for use in separating the input signal. We have previously done so and fabricated the filterbank design and achieved promising results [7]. However, the published design was a first attempt and in this paper we will use recently developed MDLNS techniques [8] to considerably improve the performance of the filterbank design.

We start by defining the MDLNS [4], demonstrating its logarithmic-like properties, and then discussing its application to the filterbank construction. We will then discuss the filterbank specifications, our original design, the improvements made, and how they reduce the resource and power consumption of the new implementation.

2. MDLNS REPRESENTATION

2.1. Definition

The MDLNS representation of a number differs somewhat from the traditional fixed radix form of linear representation. In a fixed radix positional system, a number is represented in the form

$$\chi = \sum_{i=0}^{N} m_i \cdot r^i, \tag{1}$$

where N is the number of digits, $m \in \{0, 1, \ldots, r-1\}$, i is an integer, and r is the radix. For example, in the decimal system $r = 10$, and in the binary system $r = 2$.

In the logarithmic number system (LNS), a number is represented by

$$x = s \cdot 2^a, \tag{2}$$

where a is an arbitrary real number and $s \in \{-1, 0, 1\}$. Note that the ability to set the sign to -1 and 0 allows an exact representation of 0 or negative numbers (not representable using logarithms).

A multi-dimensional logarithmic number system is based on computing with exponents of multiple base representations (or representations with s-integers [9]). In this paper, we will restrict ourselves to 2DLNS systems. A single-digit 2DLNS represents unsigned numbers in the form

$$x \approx 2^a \cdot 3^b, \tag{3}$$

where a and b are signed integers. A 2DLNS is defined more generally as

$$x \approx \sum_{i=1}^{n} s_i \cdot 2^{a_i} \cdot D^{b_i}, \tag{4}$$

where n is the number of digits, and D is the second base (and not necessarily an integer). We often refer to b_i as the nonbinary exponent, and we will drop the index i, where it is obvious by context. We define R as the constrained precision of the nonbinary exponent (i.e., $b_i = \{-2^{R-1}, \ldots, 2^{R-1} - 1\}$).

We may look at this representation as a two-dimensional generalization of the binary logarithmic number representation. The important advantage of this generalization is that the binary and second-base exponents are operated on independently from each other, with an attendant reduction in complexity of the implementation hardware. As an example, a VLSI architecture for inner-product computation with the MDLNS proposed in [4, 10] has an area complexity dependent entirely on the dynamic range of the second-base exponents. Providing that the range of the second-base exponent is smaller than the LNS dynamic range for equivalent precision, then we have the potential for a large reduction in the MDLNS hardware compared to that required by the LNS. We can capitalize on this potential improvement by placing design constraints on the second-base exponent size. For example, if we want to represent digital filter coefficients in the MDLNS, then we can design the coefficients in such a way that the second-base exponent is minimized; an integer programming task [11]. Although this approach is sound and can produce modest improvements, generalizing the representation to multi-dimensions and/or multiple digits has the potential to bring about very large reductions in hardware complexity of DSP implementations.

2.2. Mathematical operations

To summarize, a 2DLNS representation provides a triple, $\{s_i, a_i, b_i\}$, for each digit, where s_i is the sign bit and a_i, b_i are the exponents of the binary and nonbinary bases, and a number x is approximated by (4).

Multiplication and division

MDLNS multiplication and division are the simplest of the arithmetic operations. The equations for multiplication and division, given a single-digit 2DLNS representation of $x = \{s_x, a_x, b_x\}$ and $y = \{s_y, a_y, b_y\}$, are [12]

$$\begin{aligned} x \cdot y &= \{s_x \cdot s_y, a_x + a_y, b_x + b_y\}, \\ \frac{x}{y} &= \{s_x \cdot s_y, a_x - a_y, b_x - b_y\}. \end{aligned} \tag{5}$$

The above two equations show that single-digit 2DLNS multiplication can be implemented in hardware using two independent binary adders and simple logic for the sign correction. As we start to add digits to the representation, we will face the equivalent of implementing multiplication with the addition of partial products. A two-digit representation will produce four independent partial products that will have to be added, and since addition is an expensive operation, we try to optimize this process as much as possible (we will show an optimized structure later).

Addition and subtraction

Unfortunately, as with logarithms, addition and subtraction operations are not as simple as multiplication and division operations. Traditionally, addition and subtraction must be handled through a set of identities and lookup tables. The identities are [12]

$$\begin{aligned}(2^{a_1}\cdot D^{b_1})+(2^{a_2}\cdot D^{b_2}) &= (2^{a_1}\cdot D^{b_1})\cdot(1+2^{a_2-a_1}\cdot D^{b_2-b_1})\\ &\approx (2^{a_1}\cdot D^{b_1})\cdot\Phi(a_2-a_1,b_2-b_1),\\ (2^{a_1}\cdot D^{b_1})-(2^{a_2}\cdot D^{b_2}) &= (2^{a_1}\cdot D^{b_1})\cdot(1-2^{a_2-a_1}\cdot D^{b_2-b_1})\\ &\approx (2^{a_1}\cdot D^{b_1})\cdot\Psi(a_2-a_1,b_2-b_1).\end{aligned}\tag{6}$$

The operators Φ and Ψ are lookup tables (LUTs) that store the precomputed 2DLNS values of

$$\begin{aligned}\Phi(x,y)&=1+(2^x\cdot D^y),\\ \Psi(x,y)&=1-(2^x\cdot D^y).\end{aligned}\tag{7}$$

The use of large LUTs, implemented through the use of ROMs, for the evaluation of addition and subtraction operations, is the traditional approach in systems such as the LNS [13]. This technique is only feasible for very small ranges of 2DLNS numbers. It is more practical, in most cases, to convert the 2DLNS numbers to binary and perform the addition and subtraction using a binary representation.

The conversions from 2DLNS to binary will still require an LUT, but one that is much smaller than required for handling 2DLNS addition and subtraction. The LUT is used to convert the second-base portion of the 2DLNS number into a binary representation. Therefore, the size of the LUT is dependent on the number of bits used to represent the second-base exponent.

Multidigit MDLNS arithmetic

Multidigit MDLNS arithmetic is simply an extension of the single-digit MDLNS arithmetic, and is necessary when numbers are represented by more than one MDLNS digit. When performing a computation using multidigit MDLNS, each digit can be treated as an independent MDLNS number and the operations handled separately. For example, if X and Y are two-digit MDLNS numbers such that $X = x_1 + x_2$ and $Y = y_1 + y_2$, then

$$\begin{aligned}X\cdot Y &= (x_1+x_2)(y_1+y_2)\\ &= (x_1\cdot y_1)+(x_1\cdot y_2)+(x_2\cdot y_1)+(x_2\cdot y_2),\end{aligned}\tag{8}$$

where x_i and y_i are single-digit MDLNS numbers. The independence of the arithmetic operations is very important, as it allows for parallel architectures.

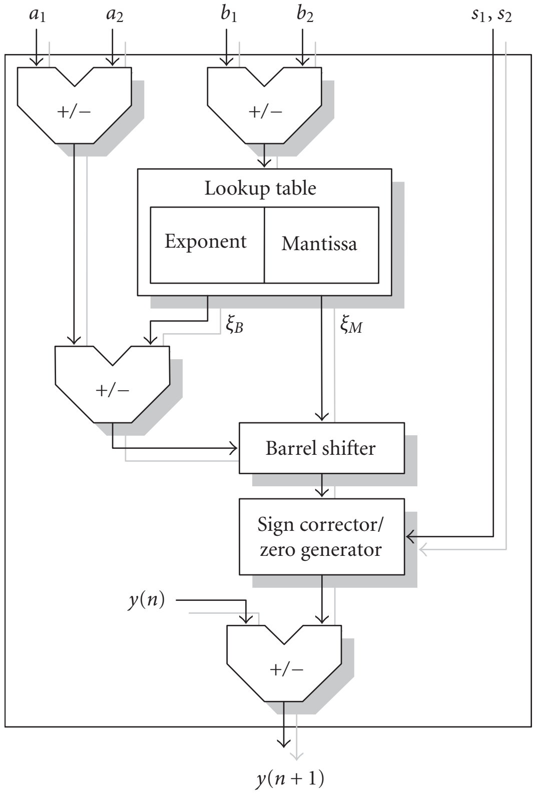

Figure 1: Single-digit 2DLNS inner product computation unit.

2.3. Hardware complexity

In order to provide complexity results for the 2DLNS inner-product computation unit, we expand on the inner-product processor architecture initially developed for the single-digit 2DLNS [12]. The processor can be used in a filter for one-dimensional convolution [14].

Single-digit computational unit

Figure 1 shows the structure of the proposed single-digit computation unit (CU). Since we do not wish to retain the 2DLNS representation of the accumulated output, and also since the CU is feedforward, we can use the 2DLNS domain for the coefficient multiplication and a binary representation for the accumulated output.

The computation performed by the CU is given in (9):

$$\begin{aligned}y(n+1) &= (s_1\cdot 2^{a_1}\cdot D^{b_1})\times(s_2\cdot 2^{a_2}\cdot D^{b_2})+y(n)\\ &= (s_1\cdot s_2)\times(2^{(a_1+a_2)}\cdot D^{(b_1+b_2)})+y(n).\end{aligned}\tag{9}$$

The multiplication is performed by small parallel adders for each of the data and coefficient base exponents. The addition output for the nonbinary exponent is the input address for an LUT (ROM). This table produces an equivalent floating-point value for the product of the nonbinary base raised to the exponent sum, as shown below:

$$D^{(b_1+b_2)}\approx 2^{\xi_B}\cdot\xi_M.\tag{10}$$

Table 1: Octave bands of human hearing and their characteristics.

Octave	Frequency range	Characteristics
1st	20–40 Hz	Low bass—these frequencies add fullness, power, and boom to sound. Lowest notes of bass, piano, and tuba fall into this category.
2nd	40–80 Hz	
3rd	80–160 Hz	Upper bass—these frequencies provide a balance in the structure of sound. Without them, sound is thin. The lower tones of the cello, trombone, and rhythm sections produce sounds in this range.
4th	160–320 Hz	
5th	320–640 Hz	Midrange—sounds get their intensity from this range of frequencies. Fundamentals and lower harmonics of most sound sources fall into this category.
6th	640–1280 Hz	
7th	1280–2560 Hz	
8th	2560–5120 Hz	Upper midrange—humans hear this range of frequencies best. 3000–3500 Hz contains information which improves the intelligibility of speech and lyrics. If this band is incorrectly processed, sound becomes unpleasant. Frequencies above 3500 Hz give sound realism and clarity. Listeners perceive sound in this section of this octave (and up to about 6000 Hz in the 9th octave) as being close. Thus 3500–6000 Hz is known as the presence range.
9th	5120–10 240 Hz	Treble—frequencies in this range give sound sparkle and brilliance. Most humans do not hear much beyond 16 000 Hz.
10th	10 240–20 480 Hz	

We find that the size of the exponents of the nonbinary base in a 2DLNS representation (where there are at least two-digits) is usually very small, which acts to exponentially reduce the hardware complexity of the CU (assuming that it is dominated by the size of the LUT).

3. ORIGINAL 2DLNS FILTERBANK DESIGN

As noted above, the 2DLNS inner product CU can be used to create an FIR filter. By using a controller circuit (state machine), we can easily schedule the data flow of the two input operands (from RAM/ROM components) and accumulation output of the CU in order to implement an MDLNS filterbank. However, before implementing any design, the constraints of a hearing instrument filterbank should be known in order to build a competitive design.

Frequency range

The frequency range of human hearing is from 20 Hz to 20 kHz [15] (see Table 1). Because of the octave-band characteristic of human hearing, good quality sound can still be achieved with half the frequency range covered. In our filterbank design, we sample the audio input at 16 kHz assuming that the input is bandlimited to 8 kHz. This will cover more than the first eight octaves, as summarized in Table 1.

Number of channels or banks

Another important constraint is the frequency resolution. The monitoring of hearing loss is accomplished through the generation of audiograms, which record measurements at eight different frequencies. Therefore, 8 channels is an acceptable resolution for hearing instruments with more resolution at lower frequencies because of the octave characteristic of human hearing [1]. This approach is used in [16]. However, in the design discussed here, we apply an efficient 2DLNS architecture to a filterbank with equally spaced filters which results in perfectly flat overall magnitude response and a reduction in filter coefficients. We note, however, that the 2DLNS can be used in any filterbank design (including octave separation filters) with similar gains to those obtained with our current design.

Stopband attenuation

The stopband attenuation in each channel determines the gain range of the hearing instrument, and at least 50 dB of gain adjustment in each bank are required. The order of the filter is proportional to stopband attenuation and passband ripple. When the order of the filter increases, the group delay and implementation cost increases. Therefore, the tradeoff between these parameters should be well adjusted to achieve an optimum design [15]. For our design, we chose a 0.01 dB passband ripple and stopband attenuation of 60 dB.

Linear phase

In a compression system, gain changes are dynamic. This may cause anomalies in the overall frequency response if phase differences exist between adjacent bands. To avoid these undesirable frequency response notches or peaks at the

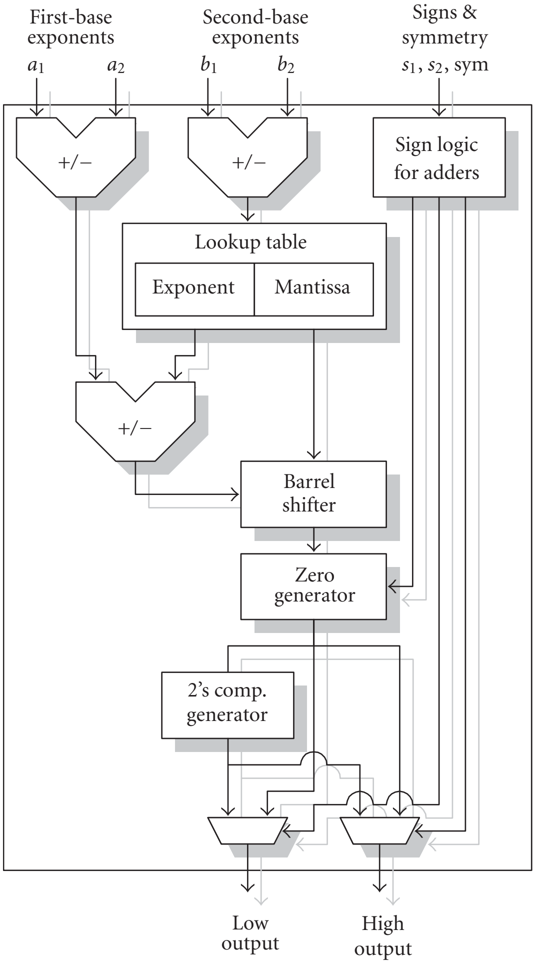

FIGURE 2: Dual 2DLNS processor for symmetrical filters (w/o accumulator).

band edges (which frequently occur in analog systems), it is necessary to constrain the filter channel impulse responses to be linear phase and of equal delay.

From the above constraints, we chose an 8-band linear phase filterbank with a 0.01 dB passband ripple and a 60 dB stopband attenuation. These values are comparable to those found in commercial hearing instrument processors [17].

Dual inner-product computational unit

A major advantage of choosing filters that are equally spaced with identical bandwidths and overlaps is that they are symmetrical allowing a perfectly flat composite magnitude response (0 dB) across the whole frequency range and duplication of the magnitude of coefficients between the low and high bands. Since the coefficients are shared, the inner-product CU can be modified to process both the low and high filters at the same time. Since only the magnitude of the coefficients may be different (depending on the symmetry of the filters), only the final binary accumulator need be duplicated to output each band (see Figure 2). As we have previously stated, although some hearing instruments use different bandwidths for the filterbanks (e.g., larger for the low pass, smaller for the high pass), using symmetric filters saves resources over nonsymmetrical filters in an FIR implementation. By using enough filter bands, custom-tailoring of bandwidths for the individual user should not be necessary.

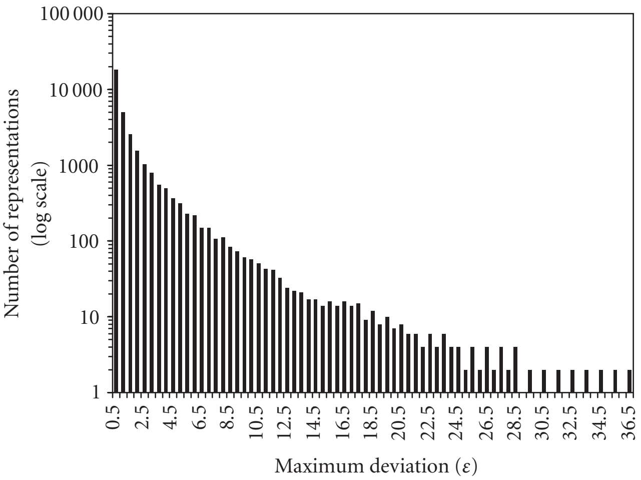

FIGURE 3: Histogram of error in coefficient optimized high/low 2DLNS input mapping.

Choice of the 2DLNS second base

Using the 8 separate equal bands, filters were designed using Matlab ("fir1" function with a Kaiser window). Eight 75-tap filters were deemed acceptable with a 0.0128 dB passband ripple and 58.9 dB stopband attenuation (these are worst-case results for all the filters in the filterbank). The specifications are met with 89 coefficients. Of the 600 coefficients generated, only 132 of them are unique in magnitude which simplifies the search for an optimal base with a minimum value of R. In the case of the above filter specifications, with an optimal base of 1.28308348549366 and $R = 2$, the filterbank responses are slightly worse with a 0.0176 dB passband ripple and a 57.7 dB stopband attenuation. As R is increased, the specifications are matched to that of the Matlab 64-bit floating-point values. Clearly, however, we need to keep R as low as possible.

Binary-to-2DLNS conversion

The input data (16-bit signed) is converted to 2DLNS via a high/low serial implementation [18] with the second-base exponents limited from −14 to 14. The limit is adjusted from −16 to 15 ($R = 5$) so that overflow never occurs when the input data is multiplied with the coefficients ($R = 2$). By limiting the exponents in this way, the representation is used to its fullest. Of 32 768 possible representations, the high/low converter generates 18 348 error-free (56% with $\varepsilon < 0.5$) representations. The remaining 14 420 representations have errors from 0.5 to 37 in which the frequency decreases almost logarithmically (see Figure 3).

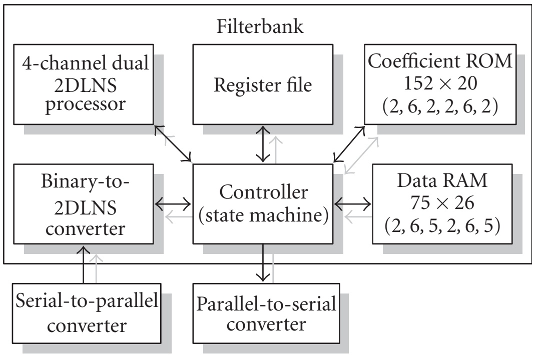
Filterbank
4-channel dual 2DLNS processor
Register file
Coefficient ROM 152 × 20 (2, 6, 2, 2, 6, 2)
Binary-to-2DLNS converter
Controller (state machine)
Data RAM 75 × 26 (2, 6, 5, 2, 6, 5)
Serial-to-parallel converter
Parallel-to-serial converter

Figure 4: Filterbank structure.

Serial architecture

Since the filterbank is intended for audio (sampling frequency of 16 kHz) and low-power operation, a serial implementation is favorable to minimize both power and area. Assuming that two of the 600 coefficients are processed each cycle, an operating clock of $16\,000\,\text{Hz} \cdot 600/2 = 4\,800\,000\,\text{Hz}$ or 4.8 MHz is required. The controller is therefore used to move data from the controller into a RAM where 75 values are multiplied with 75 coefficients and accumulated (see Figure 4). Serial-to/from-parallel converters are used to reduce the I/O pad count since the design would otherwise be I/O bound (i.e., the silicon area inside the pad ring is much larger than required by the processing circuitry).

Full details of the original design can be found in [19]. The design core is 1 mm × 1 mm and 1.67 mm × 1.67 mm including I/O pads (see Figure 5).

4. IMPROVED 2DLNS FILTERBANK DESIGN

Our original filterbank design was intended to show that the MDLNS could be used for this particular application and possibly save power in the process. Although the design was essentially a collection of existing MDLNS building blocks, the power results were encouraging enough for us to work on the new design presented in this paper.

Filterbank scalability

The controller for the original system is fixed to process the eight 75-tap filters, and is not easily scalable to process more coefficients or filters. For example, adjusting the filter to handle 89-tap filters or 10 bands would require significant coding and retesting. The improved filterbank controller is capable of processing any even number of filter bands and any odd number of coefficients. The architecture uses "smart" counters which generate dynamic references reducing the overall driving logic. The address path to the SRAM is fully utilized eliminating conditional counters and maximizing memory efficiency. These filterbank parameters are applied before synthesis to generate a static controller. A dynamic controller is quite achievable when run-time loading of the parameters and filterbank coefficients is desirable (assuming the memory capacities are large enough).

Dual-port-to-single-port SRAM

The original filterbank controller uses a third-party blackbox 256 × 32 dual-port RAM of which only 75 × 26 elements are used. The dual-port RAM component in the original design was used simply because it was smaller in area and used less power than any other single-port RAM component available to our design group. Unfortunately the controller performs both read and write operations on the same cycle which makes the design unusable for a single-port RAM. Since dual-port RAMs are generally twice the area of single-port RAMs, and consequently consume more power, the improved filterbank uses synchronized input data storage and processing in the same cycle to allow the use of a single-port RAM. With the appropriately sized single-port SRAM we obtain significant reductions in silicon area and power consumption.

SRAM operation

The original filterbank controller operates the RAM on the opposite of the system clock to guarantee that the inputs are stable (see Figure 6).

This is not necessary in our new design since the SRAM contains its own built in latches (edge triggered D flip-flops) which have zero hold time. Coding for a component which has its own input latches is possible in the Verilog hardware description language, we use by mirroring the synchronous and asynchronous logic (see Figure 7).

Operating the SRAM at the opposite clock of the system is not favorable since it will cause more logic transitions at both the beginning of, and halfway through, the cycle which consume more power (see Figure 8).

Operating the SRAM at the same clock as the system will remove invalid stable states between clock phases thus reducing the power (see Figure 9).

Maintenance clock cycles

The original filterbank required 13 additional cycles to perform maintenance operations (reset counters, memory pointers, etc.). These extra cycles contribute to increased power consumption, additional logic cells, and scalability issues (i.e., more coefficients and bands require more cycles). The new filterbank controller schedules arithmetic operations, multiplexes data paths, and pipelines information to eliminate any maintenance cycles. The system can now operate at the optimum 4.8 MHz clock rate, processing an input every 300 cycles or at a 16 kHz sampling frequency.

Channel accumulator delay

The four-channel dual 2DLNS processor in the original design first generates the signed-binary representation of the data multiplied by the coefficient (as in the DBNS/2DLNS inner-product CU used for an earlier hybrid chip [14]) for each channel and then adds them together. For the high-pass filter, the sum of these channels may, depending on the symmetry, have to be negated once before accumulation.

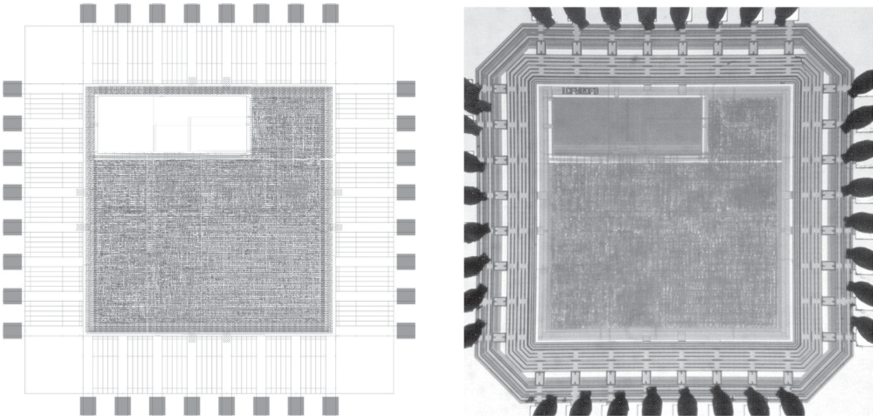

Figure 5: Screen copy and micrograph of the 2DLNS filterbank.

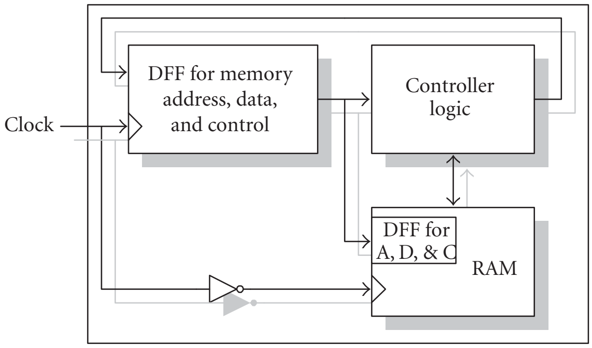

Figure 6: Two-phase clock controlling memory latches.

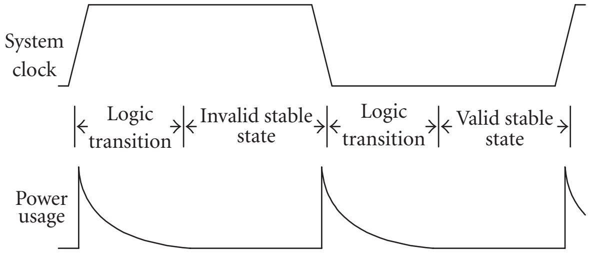

Figure 8: Two-phase clock power consumption.

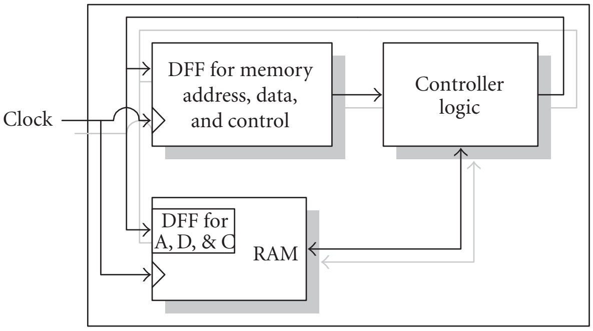

Figure 7: One-phase clock controlling memory latches.

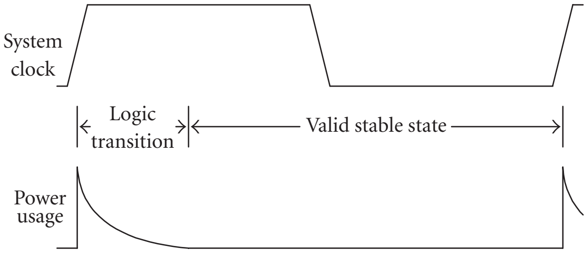

Figure 9: Single-phase clock power consumption.

These two negating operations add extra delay, logic, and power requirements. In total, 5 two's complement generators and 5 adder components are used to merge all the channels. The worst-case delay from multiplication to final accumulation is 5 arithmetic operations.

New one sign-bit architecture

The data path of the dual 2DLNS processor (shown in Figure 2) is affected significantly by the signs of the operands. The required sign correction operation comes at a cost of additional logic and power. Since our particular filterbank architecture requires additional processing to be performed after the dual 2DLNS processor, it is possible to use the common single sign-bit binary representation for the intermediate results. We have therefore developed a new 2DLNS sign system to reduce the processing path of the 2DLNS inner-product CU while producing a single sign-bit binary representation.

Our original 2DLNS notation uses two bits to represent the sign for each digit (−1, 0, and 1). There are only three of four states used, one of which (zero) only represents a single value. Using two sign bits results in having nearly 50

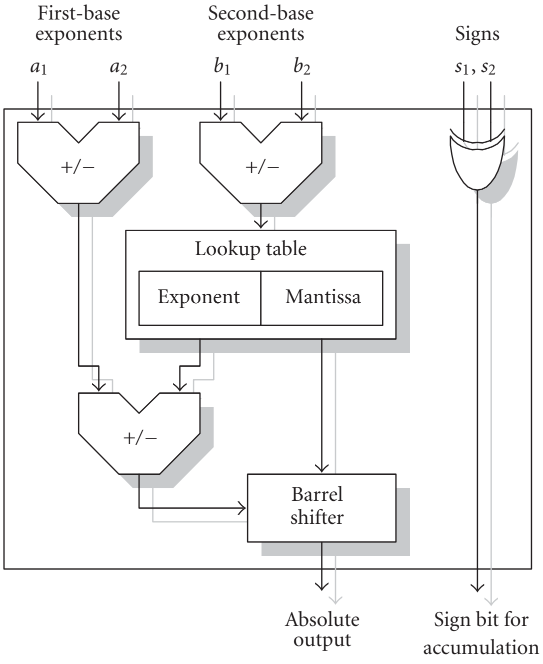

Figure 10: Dual one-bit sign 2DLNS processor.

percent of the representation space unused. To improve this ratio, only a single sign bit is needed to represent the most used cases (−1 and 1). We now represent zero by setting the nonbase two indices to their most negative values (i.e., $b = -2^{R-1}$). This allows us to reduce the circuitry of the system while maintaining the independent processing of the indices and this modification is easily integrated into the existing two-bit sign architecture. This special case for zero still leaves us with unused representation space, but not nearly as much as with the two-bit sign system.

By using the one sign-bit architecture for our filterbank, the word lengths for the 2DLNS representation of the coefficients and data are reduced by 2 bits. The 2DLNS processor is improved since it no longer needs to handle the negative or special zero case; only the absolute output is required. The coefficient and data signs are simply XORed to produce the output sign which is used along with the absolute output to determine the final sum (see Figure 10).

Four-channel accumulator

The four-channel and output accumulation process is simplified with a single sign bit by using only 5 adder/subtractor components and simple logic to coordinate the proper series of operations (see Figure 11). The delay is reduced to 3 arithmetic operations and the logic is also reduced since an adder/subtractor component is smaller than a separate adder and 2's complement generator.

Data and coefficient representations

Using the single sign bit simplifies the implementation of the filterbank, however, it limits the 2DLNS filterbank coefficients since one of the second-base exponent states is used to represent zero. With $R = 2$ the range of the coefficient nonbinary exponent is now from −1 to 1 which reduces the filterbank responses to a 0.0213 dB passband ripple and a 55.9 dB stopband attenuation. To better meet the specifications, we can either use more coefficients or increase R. With $R = 3$ the range on the nonbinary exponent is from −3 to 3 which improves the filterbank responses to 0.0134 dB for the passband ripple and 59.1 dB for the stopband attenuation. Although increasing R for the coefficients improves the filterbank response, the data representation nonbinary base index is reduced from 29 (−14 to 14) to 25 (−12 to 12) states. This will reduce the number of unique representations for the filter input data, and we can therefore expect a larger error than that shown in the original design (Figure 3). The single sign bit reduces hardware in this case, but increases representational error.

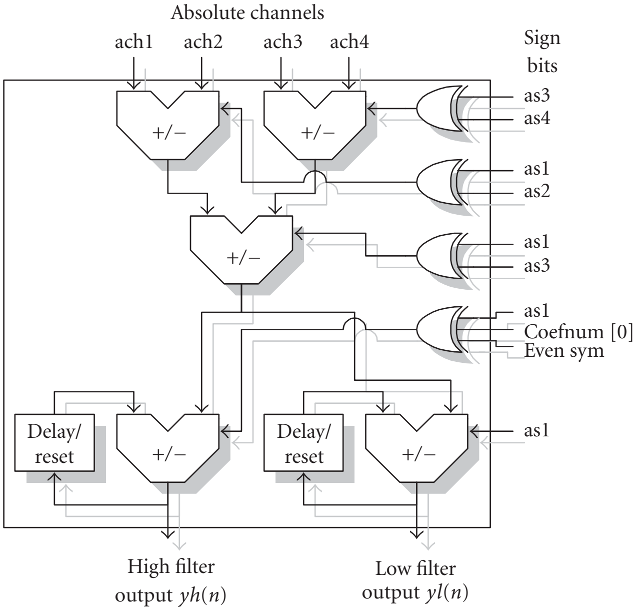

Figure 11: One-bit sign four-channel accumulator.

Optimal input data mapping

An alternative approach was taken where we optimized the nonbinary base for the input data (exponent range from −12 to 12) rather than the filter coefficients. The coefficients were then mapped using that base ($D = 0.92024380912663017$) with $R = 3$ obtaining better filterbank responses (0.0137 dB passband ripple and 58.2 dB stopband attenuation) than those of the original 2DLNS filter design and similar to those using an optimal coefficient base and $R = 3$. Using this approach, the input data mapping is improved with 19 513 error-free representations of the total 32 768 (59.5%) (about 3.5% more than the original design). More importantly, the maximum error of any of the input data representations is below 6 (see Figure 12). By optimizing the representation for a single sign bit, the accuracy of the input data is considerably improved without changing the filterbank response

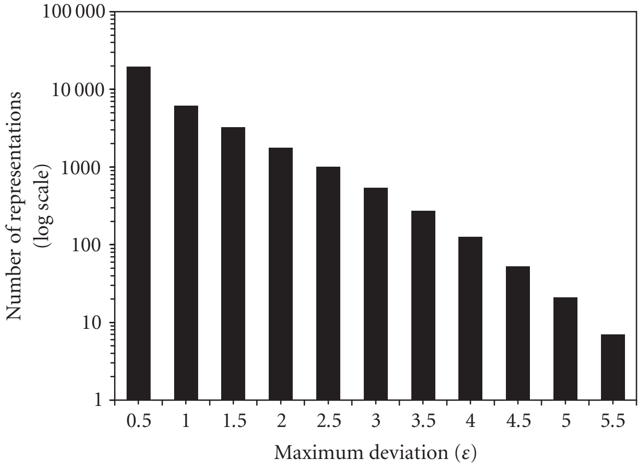

Figure 12: Histogram of error in data optimized high/low 2DLNS input mapping.

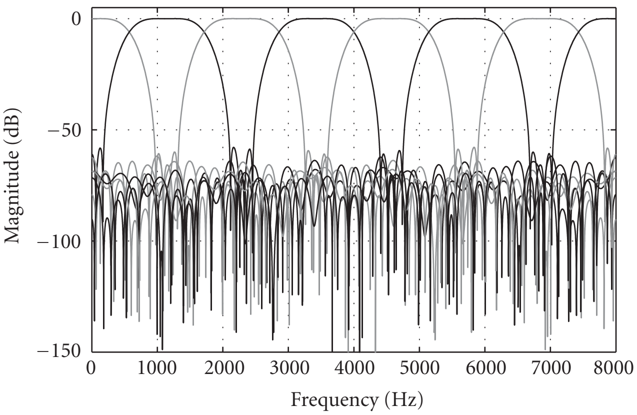

Figure 13: Improved MDLNS filterbank frequency response.

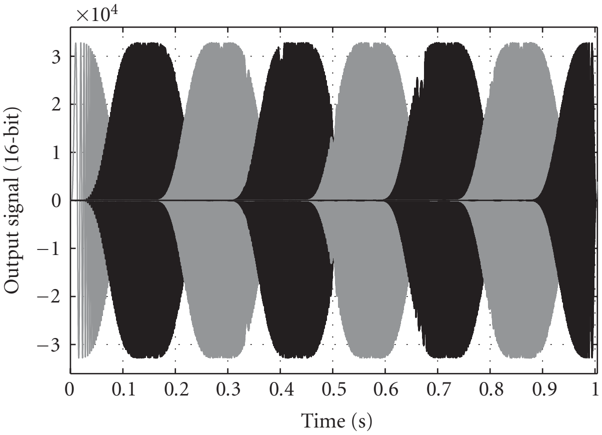

Figure 14: MDLNS filterbank output of an 8 kHz chirp signal.

significantly. The single sign-bit 2DLNS processor will also reduce interconnect and area/logic as well as power consumption.

5. RESULTS AND COMPARISONS

The improved MDLNS filterbank simulated frequency response is shown in Figure 13 and the simulated output of an 8 kHz chirp signal is shown in Figure 14.

The original MDLNS filterbank was designed using Verilog, synthesised with Synopsys Design Compiler (using worst-case models), placed with Cadence AreaPDP, routed with Cadence Silicon Ensemble, and fabricated in a 1.6 V TSMC 0.18 μm CMOS process. At the time of writing this paper, we have not yet fabricated the new design. We can, however, estimate the core size to be 555 μm × 555 μm (a little more than the quarter of the size of the original) by assuming the same cell placement ratio as the original filterbank. We also assume the power measurements are fairly accurate since the original filterbank simulated measurements were close to the test results using the same process parameters. The design statistics and percentage savings between the original and improved filterbanks can be found in Table 2, with considerable reductions in area, number of logic cells, interconnects, and power consumption.

For comparison purposes, we look at two recently published designs. A 16-bank linearly spaced filter, with a 40 dB stopband attenuation, using an FFT approach [20] has a power consumption of 1 mW at 1.8 V in a 0.18 μm CMOS process. If we scale this 16-bank design to an 8-bank design, we could conservatively estimate the power to be about half or 500 μW. A 7-bank logarithmically spaced filter with a 50 dB stopband attenuation, using an IFIR approach [16], has a power consumption of 471 μW at 1.55 V in a low-power 0.7 μm CMOS process. Our design appears competitive at 316 μW, but it is important to point out that the design presented here only uses a generic 0.18 μm "black-box" standard cell library. Due to proprietary restrictions, we are not allowed to modify or improve the performance of any of these cells. We are currently unable to obtain access to low-power standard cell libraries, since they are not generally distributed to universities.

We would also like to note that our power estimates are based on the worst-case performance of the filterbank (i.e., a maximum amplitude, chirp input). Our best-case measurements estimate the filterbank will require less than 180 μW when idle (i.e., a low amplitude, low-frequency input).

As a final note, we have recently developed a process for adding/subtracting MDLNS digits entirely within the MDLNS (no conversion to/from binary is required) [8]. We are optimistic that this approach will lower the power consumption even more than shown in the design presented here. This may also open the possibility of using MDLNS for further signal processing (i.e., compression) since the signal channels will remain in the MDLNS representation after frequency separation.

Table 2: Area, cell, net, and power comparison between original and improved filterbank (excludes SRAM).

Design	Total cell area (μm^2)	Logic cells	Interconnects	Estimated power at 1.6 V @ 4.8 MHz (μW)
Original	184 965	7005	5759	708
Improved	53 716	3742	4877	316
Savings	71.0%	46.6%	15.3%	55.4%

6. CONCLUSIONS

In this paper, we have discussed an improved 2DLNS filterbank architecture for applications in a CIC hearing-aid systems. For this application, the size, power, linear phase, and flat overall magnitude response are important constraints for the filterbank design. We have discovered that the 2DLNS offers significant advantages over the standard binary system, mainly through overhead reduction achieved by not using multipliers. The 2DLNS filterbank has linear phase with a perfectly flat overall magnitude response; a considerable improvement over IFIR filterbank designs. By applying newly developed MDLNS architectures and circuit optimizations to an existing design, the power and performance of the filterbank are shown to be quite competitive with IFIR and DFT binary implementations based on recently published designs. We have also commented on some very recent work that may allow even more reductions in power consumption.

ACKNOWLEDGMENTS

The authors would like to acknowledge financial support from the Natural Sciences and Engineering Research Council (NSERC) of Canada, the Micronet Network of Centres of Excellence, and Gennum Corporation. The authors also acknowledge the important contribution of CMC Microsystems for their equipment, software loan, and fabrication services.

REFERENCES

[1] J. Agnew, “An overview of digital signal processing in hearing instruments,” *Hearing Review*, July 1997.

[2] R. Muscedere, G. A. Jullien, V. S. Dimitrov, and W. C. Miller, “Nonlinear signal processing using index calculus DBNS arithmetic,” in *Advanced Signal Processing Algorithms, Architectures, and Implementations X*, F. T. Luk, Ed., vol. 4116 of *Proceedings of SPIE*, pp. 247–257, San Diego, Calif, USA, August 2000.

[3] “Paragon Digital^{TM} Two-Channel DSP Systems,” Gennum Corporation, Burlington, Ontario, Canada, Document no. 14437-1, September 2001.

[4] V. S. Dimitrov, J. Eskritt, L. Imbert, G. A. Jullien, and W. C. Miller, “The use of the multi-dimensional logarithmic number system in DSP applications,” in *Proc. 15th IEEE Symposium on Computer Arithmetic (Arith ’01)*, pp. 247–254, Vail, Colo, USA, June 2001.

[5] E. E. Swartzlander and A. G. Alexopoulos, “The sign/logarithm number system,” *IEEE Trans. Comput.*, vol. 24, no. 12, pp. 1238–1242, 1975.

[6] T. J. Sullivan, R. E. Morley Jr., and G. L. Engel, “A VLSI FIR digital signal processor using logarithmic arithmetic,” in *Proc. IEEE Workshop on VLSI Signal Processing*, pp. 276–280, VLSI Signal Processing-III, IEEE Press, Monterey, Calif, USA, November 1988.

[7] H. Li, R. Muscedere, V. S. Dimitrov, and G. A. Jullien, “The application of 2-D logarithms to low-power hearing-aid processors,” in *Proc. 45th IEEE International Midwest Symposium on Circuits and Systems (MWSCAS ’02)*, vol. 3, pp. 13–16, Tulsa, Okla, USA, August 2002.

[8] R. Muscedere, *Difficult operations in the multi-dimensional logarithmic number system*, Ph.D. thesis, University of Windsor, Windsor, Ontario, Canada, 2003.

[9] B. M. M. de Weger, *Algorithms for Diophantine Equations*, vol. 65 of *CWI Tracts*, Centrum voor Wiskunde en Informatica, Amsterdam, the Netherlands, 1989.

[10] S. Sadeghi-Emamchaie, G. A. Jullien, V. S. Dimitrov, and W. C. Miller, “Digital arithmetic using cellular neural networks,” *Journal of Circuits, Systems and Computers*, vol. 6, no. 8, pp. 515–535, 1998.

[11] G. A. Jullien, V. S. Dimitrov, B. Li, W. C. Miller, A. Lee, and M. Ahmadi, “A hybrid DBNS processor for DSP computation,” in *Proc. IEEE International Symposium on Circuits and Systems (ISCAS ’99)*, vol. 1, pp. 5–8, Orlando, Fla, USA, May–June 1999.

[12] V. S. Dimitrov, G. A. Jullien, and W. C. Miller, “Theory and applications of the double-base number system,” *IEEE Trans. Comput.*, vol. 48, no. 10, pp. 1098–1106, 1999.

[13] M. G. Arnold, T. A. Bailey, J. R. Cowles, and J. J. Cupal, “Redundant logarithmic arithmetic,” *IEEE Trans. Comput.*, vol. 39, no. 8, pp. 1077–1086, 1990.

[14] S. J. Eskritt, “Inner product computational architectures using the double base number system,” M.S. thesis, University of Windsor, Windsor, Ontario, Canada, 2001.

[15] E. Onat, “DSP algorithms for digital hearing instruments,” M.S. thesis, University of Windsor, Windsor, Ontario, Canada, 2001.

[16] L. S. Nielsen and J. Sparsø, “Designing asynchronous circuits for low power: an IFIR filter bank for a digital hearing aid,” *Proc. IEEE*, vol. 87, no. 2, pp. 268–281, 1999.

[17] “DUET DIGITAL^{TM} Advanced DSP System with FRONTWAVE®,” Gennum Corporation, Burlington, Ontario, Canada, Document no. 20352-1, December 2003.

[18] R. Muscedere, V. S. Dimitrov, G. A. Jullien, and W. C. Miller, “Efficient techniques for binary-to-multidigit multidimensional logarithmic number system conversion using range-addressable look-up tables,” *IEEE Trans. Comput.*, vol. 54, no. 3, pp. 257–271, 2005, Special Issue on Computer Arithmetic.

[19] H. Li, “A 2-digit multi-dimensional logarithmic number system filterbank processor for a digital hearing aid,” M.S. thesis, University of Windsor, Windsor, Ontario, Canada, 2003.

[20] T. Schneider, R. Brennan, P. Balsiger, A. Heubi, and F. Pellandini, “An ultra low-power programmable DSP system for hearing aids and other audio applications,” in *Proc. International Conference on Signal Processing Applications and Technology (ICSPAT ’99)*, Orlando, Fla, USA, November 1999.

Roberto Muscedere was born in Windsor, Ontario, Canada, in 1973. He received his B.A.S. degree in 1996, M.A.S. degree in 1999, and Ph.D. degree in 2003, all from the University of Windsor in electrical engineering. During this time, he also managed the microelectronics computing environment at the Research Centre for Integrated Microsystems (formally VLSI Research Group), the University of Windsor. He is currently an Assistant Professor in the Electrical and Computer Engineering Department, the University of Windsor. His research areas include the implementation of high-performance and low-power VLSI circuits, full and semicustom VLSI design, computer arithmetic, HDL synthesis, and digital signal processing.

Vassil Dimitrov was born in Plovdiv, Bulgaria, in 1964. He received the Ph.D. degree in mathematics in 1995 from the Mathematical Institute of the Bulgarian Academy of Sciences. Since then he has spent two years as a Postdoctoral Fellow at the VLSI Research Group, University of Windsor, Canada, one year as a Research Scientist at Reliable Software Technologies, Virginia, USA, and one year as a Chief Research Scientist at the Laboratory of Signal Processing and Computer Technology, Helsinki University of Technology, Finland. Between July 2000 and June 2001, he held an Associate Professor position in the Department of Electrical and Computer Engineering, the University of Windsor, Canada, and since July 2001 he has been an Associate Professor in the Department of Electrical and Computer Engineering, the University of Calgary, Alberta, Canada. His main research interests include DSP algorithms, cryptography, algorithmic number theory, and related topics. He is a Member of the New York Academy of Sciences.

Graham Jullien was educated in the United Kingdom, receiving a B.Tech. degree, in electrical engineering, from the University of Loughborough, Loughborough, UK, in 1965, the M.S. degree from the University of Birmingham, Birmingham, UK, in 1967, and the Ph.D. degree from the Aston University, Birmingham, UK, in 1969. From 1961 to 1966, he was a Student Engineer and Data Processing Engineer at English Electric Computers, Kidsgrove, UK. From 1975 to 1976, he was a Visiting Senior Research Engineer at the Central Research Laboratories, EMI Ltd., Hayes, UK. From 1969 until 2000, he was with the Department of Electrical and Computer Engineering, the University of Windsor, Ontario, Canada, where he held the rank of a University Professor and was the Director of the VLSI Research Group. Since January, 2001, he has been with the Department of Electrical and Computer Engineering, the University of Calgary, where he holds the iCORE Research Chair in advanced technology information processing systems. He is a Member of the Board of Directors of CMC Microsystems and is a Member of the Steering Committee and Board of Directors of the Micronet Network of Centres of Excellence. He has published widely in the fields of digital signal processing, computer arithmetic, neural networks, and VLSI systems, and teaches courses in related areas. He has served on the technical committees of many international conferences, and he currently serves on the Editorial Board of the Journal of VLSI Signal Processing; he is a past Associate Editor of the IEEE Transactions on Computers. He hosted and was a Program Cochair of the 11th IEEE Symposium on Computer Arithmetic, Program Chair for the 8th Great Lakes Symposium on VLSI, and Technical Program Chair for the 1999 Asilomar Conference on Signals, Systems and Computers. He was a General Chair for the 2003 Asilomar Conference and was General Cochair of the International Workshop on System-on-Chip for Real-Time Systems, Calgary, Alberta, 2003.

William Miller received the B.S.E. degree from the University of Michigan, Ann Arbor, and the M.A.S. and Ph.D. degrees from the University of Waterloo, Waterloo, Ontario, Canada, all in electrical engineering. He is a Professor of electrical and computer engineering at the University of Windsor, Windsor, Ontario, Canada, and is the Director of the Research Centre for Integrated Microsystems at the university. His interests include electronics, digital signal processing, neural networks, microelectronics, and microelectromechanical systems (MEMS). He has authored or coauthored over 240 research papers in refereed journals and conference proceedings. He is carrying out research in the design of MEMS devices for hearing instrument applications as part of a research collaboration with the Gennum Corporation of Burlington, Ontario. He is currently the Vice-Chairman of the Board of Directors of the Canadian Microelectronics Corporation (CMC), a not-for-profit corporation delivering a national research infrastructure support program to microsystems researchers in universities across Canada. He is a registered professional engineer (P. Eng.) in the province of Ontario.