Digital Receivers and Transmitters Using Polyphase Filter Banks for Wireless Communications

Fredric J. Harris, *Fellow, IEEE*, Chris Dick, *Member, IEEE*, and Michael Rice, *Senior Member, IEEE*

***Abstract*—This paper provides a tutorial overview of multichannel wireless digital receivers and the relationships between channel bandwidth, channel separation, and channel sample rate. The overview makes liberal use of figures to support the underlying mathematics. A multichannel digital receiver simultaneously down-convert a set of frequency-division-multiplexed (FDM) channels residing in a single sampled data signal stream. In a similar way, a multichannel digital transmitter simultaneously up-converts a number of baseband signals to assemble a set of FDM channels in a single sampled data signal stream. The polyphase filter bank has become the architecture of choice to efficiently accomplish these tasks. This architecture uses three interacting processes to assemble or to disassemble the channelized signal set. In a receiver, these processes are an input commutator to effect spectral folding or aliasing due to a reduction in sample rate, a polyphase M-path filter to time align the partitioned and resampled time series in each path, and a discrete Fourier transform to phase align and separate the multiple baseband aliases. In a transmitter, these same processes operate in a related manner to alias baseband signals to high order Nyquist zones while increasing the sample rate with the output commutator.**

This paper presents a sequence of simple modifications to sampled data structures based on analog prototype systems to obtain the basic polyphase structure. We further discuss ways to incorporate small modifications in the operation of the polyphase system to accommodate secondary performance requirements. MATLAB simulations of a 10-, 40-, and 50-channel resampling receiver are included in the electronic version of this paper. An animated version of the ten-channel resampling receiver illustrates the time and frequency response of the filter bank when driven by a slowly varying linear FM sweep.

***Index Terms*—Digital channelizers, digital radio, digital receivers, equivalency theorem, multichannel digital receiver, noble identity, polyphase filter bank.**

MOTIVATION

RADIO receivers and transmitters perform a sequence of invertible signal transformations in order to communicate through imperfect bandlimited channels. The transformations applied to waveforms are associated with disjoint frequency spans classically called baseband, IF, and RF. Early radios performed the desired transformations using appropriate linear and nonlinear lumped and distributed analog circuit elements.

The confluence of three technology areas has had profound effect on the way we manipulate baseband and low IF signals. Two of these areas, enabled by the transistor and later by integrated circuits (ICs), are the analog-to-digital converter (ADC) and digital-to-analog converter (DAC) and the programmable microprocessor. The third technology area is algorithm development by the digital signal processing (DSP) community. These technologies coupled with an educated and motivated work force led inexorably to insertion of DSP in the signal-processing path of radio receiver and transmitter systems.

Intel's former CEO G. Moore [4] observed that the cost of performing a specified processing task on an IC drops by a factor of two every 18 months or, equivalently, the amount of processing that can be performed at a fixed cost doubles every 18 months. This relationship, known as Moore's law, appears to be unique to the semiconductor industry. A similar cost-performance curve does not exist for general circuit components A consequence of Moore's law is the migration from designs that assemble and integrate sub-system to designs that are full systems on a chip (SOC).

An important participant in the semiconductor arena is the field programmable gate array (FPGA) [5]. The FPGA consists of a vast array of configurable logic tiles, multipliers, and memory. This technology provides the signal-processing engineer with the ability to construct a custom data path that is tailored to the application at hand. FPGAs offer the flexibility of instruction set digital signal processors while providing the processing power and flexibility of an application-specific integrated circuit (ASIC). The FPGA enables significant design cycle compression and time-to-market advantages, an important consideration in an economic climate with ever-decreasing market windows and short product life cycles.

DSP-based processing of baseband and low IF signals offer cost and performance advantages related to manufacturability, insensitivity to environment, ability to absorb design changes, and ease of feature insertion for product evolution and differentiability. The DSP segment of a radio enhances the radio while reducing its cost, thus enabling larger market penetration, as well as new market formation. DSP and RF and microwave communication systems are tightly coupled.

The authors have written this paper to help the RF and microwave engineer acquire an understanding of the key work performed by their DSP partners in pursuit of their common goal, the design and production of competitive high-quality RF communication and RF monitoring systems. We begin this paper with a review of a standard architecture for analog transmitters

Manuscript received May 2, 2002.

F. J. Harris is with the Department of Electrical and Computer Engineering, College of Engineering, San Diego State University, San Diego, CA 92182-1309 USA (e-mail: fred.harris@sdsu.edu).

C. Dick is with Xilinx, San Jose, CA 95124 USA (e-mail: chris.dick@xilinx.com).

M. Rice is with the Department of Electrical Engineering, Brigham Young University, Provo, UT 84601 USA.

This paper has supplementary downloadable material available at http://ieee-explore.ieee.org, provided by the authors. This includes six MATLAB script files. These are standard m-files that will run under a MATLAB program. The Signal Processing Toolbox is the only MATLAB Toolbox required to run these script files. This material is 3.4 MB in size.

Digital Object Identifier 10.1109/TMTT.2003.809176

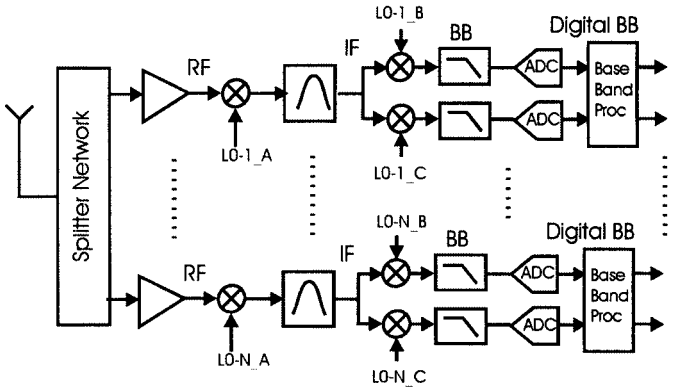

Fig. 1. First-generation RF architecture of N-channel receiver.

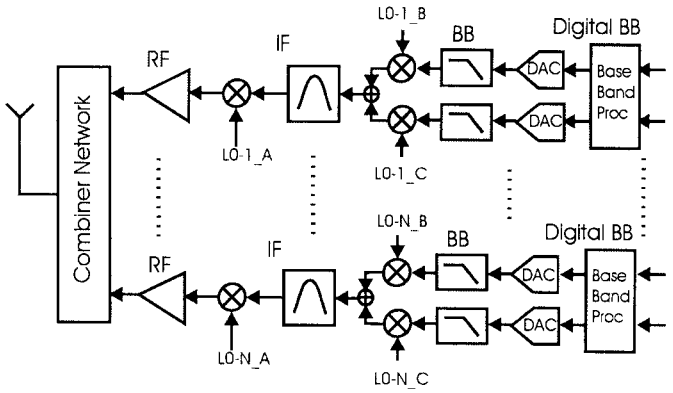

Fig. 2. First-generation RF architecture of N-channel transmitter.

and receivers. Here, the interface between continuous and sampled data is located at the end of the signal-processing path and operates at the highest signal-to-noise ratio with the lowest sample rate. We then present variants of the standard architectures in which the operating conditions change to process signals at higher dynamic range and at higher sample rates. A common variant of the high sample rate option is IF sampling.

High sample rate converters offer the option in a receiver to acquire large segments of input bandwidth and absorb much of the signal-processing tasks and functions in DSP algorithms. The dual task of assembling large segments of bandwidth in a transmitter is implied, but is not addressed here. The receiver processing includes partitioning, filtering, translation, and demodulated. The remainder of this paper is restricted to a description of various techniques to accomplish single or multiple channel extraction of signal bands from the bandwidth collected by high-bandwidth converters.

I. Introduction

Base stations for cellular mobile communication systems [6] offer an example of a radio receiver that must down-convert and demodulate multiple simultaneous narrow-band RF channels. The traditional architecture of a radio receiver that performs this task is shown in Fig. 1. This architecture contains N sets of dual-conversion sub-receivers. Each receiver amplifies and down-converts a selected RF channel to an IF filter that performs initial bandwidth limiting.

The output of each IF filter is again down converted to baseband by matched quadrature mixers that are followed by matched baseband filters that perform final bandwidth control. Each quadrature down-converted signal is then converted to their digital representation by a pair of matched ADCs. The output of the ADCs is processed by DSP engines that perform the required synchronization, equalization, demodulation, detection, and channel decoding.

Fig. 2 shows a base-station companion radio transmitter formed by N sets of dual conversion sub-transmitters that modulate and up-convert multiple simultaneous narrow-band RF channels. Note that the signal flow for the transmitter chain is simply a reversal of the signal flow of the receiver chain.

Gain and phase imbalance between the two paths containing the quadrature mixers, analog baseband filters, and ADC in an N-channel receiver or N-channel transmitter is the cause of

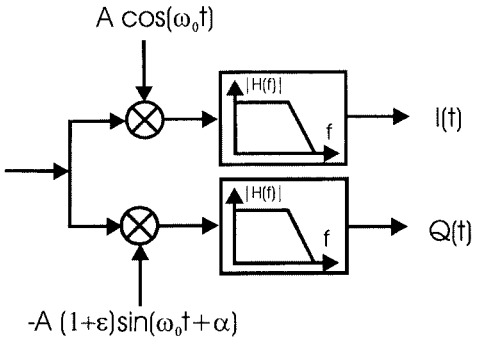

Fig. 3. Quadrature down converter with gain and phase imbalance.

crosstalk between the in-phase and quadrature (*I/Q*) components [7]. This, in turn, results in coupling between the many narrow-band channels sometimes called ghosts or images. This spectral coupling can be described compactly by examining the model shown in Fig. 3. Here, the composite *I/Q* gain and phase imbalances have been assigned to the quadrature term as the amplitude and phase shift of the sinusoid.

We can examine the unbalanced complex sinusoid presented to the mixer pair and compare its spectrum to that of the balanced spectrum. The complex sinusoid shown in (1) is expanded in (2) to explicitly show the positive and negative frequency components. Equation (3) uses the small-signal approximation to obtain a simple estimate of the effects of gain and phase imbalance on the positive and negative frequency components of the quadrature mixer signal. Fig. 4 presents a graphical visualization of these same spectral components.

$$g(t) = A\left[\cos(\omega_0 t) - j(1+\varepsilon)\sin(\omega_0 t + \alpha)\right] \tag{1}$$

$$\begin{aligned} g(t) = &\left\{\left[\frac{A}{2} - \frac{A}{2}(1+\varepsilon)\cos(\alpha)\right]\right. \\ &\left. - j\left[\frac{A}{2}(1+\varepsilon)\sin(\alpha)\right]\right\} e^{+j\omega_0 t} \\ &+ \left\{\left[\frac{A}{2} + \frac{A}{2}(1+\varepsilon)\cos(\alpha)\right]\right. \\ &\left. - j\left[\frac{A}{2}(1+\varepsilon)\sin(\alpha)\right]\right\} e^{-j\omega_0 t} \end{aligned} \tag{2}$$

$$g(t) \cong A\left[\frac{\varepsilon}{2} - j\frac{\alpha}{2}\right] e^{+j\omega_0 t} + A\left[\left(1+\frac{\varepsilon}{2}\right) - j\frac{\alpha}{2}\right] e^{-j\omega_0 t}. \tag{3}$$

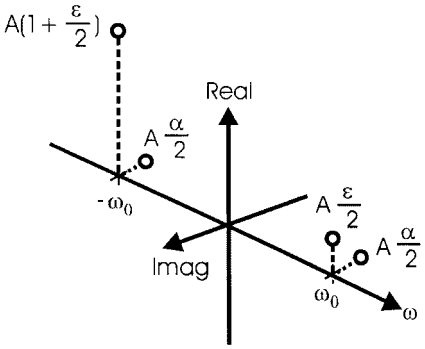

Fig. 4. Spectral components of unbalanced complex sinusoid.

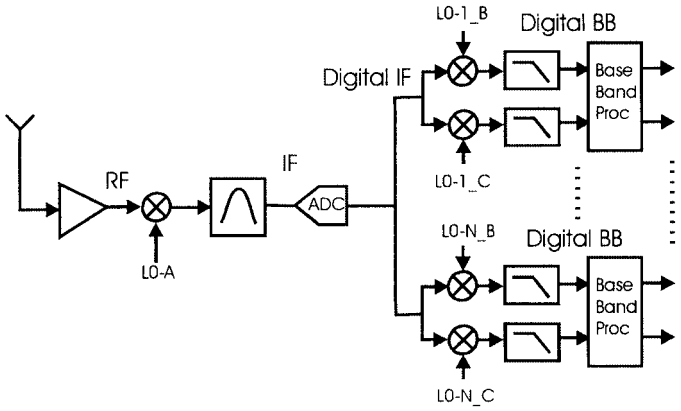

Fig. 5. Second-generation RF architecture of N-channel receiver.

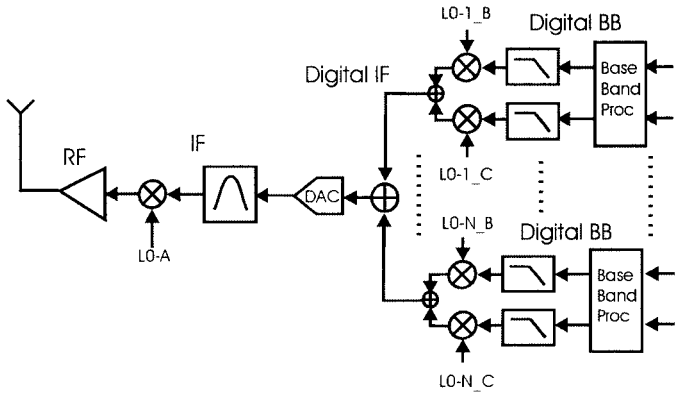

Fig. 6. Second-generation RF architecture of N-channel transmitter.

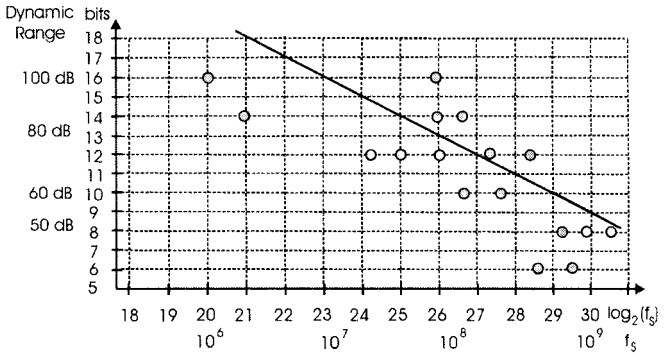

Fig. 7. Scatter diagram showing speed-precision performance of ADCs.

Besides the obvious coupling between the quadrature components at the same frequency due to phase imbalance, we see a coupling between positive and negative frequencies due to both amplitude and phase imbalance. To achieve an imbalance related spectral image 40 dB below the desired spectral term, each imbalance term must be less than 1% of the desired term. It is difficult to sustain, over time and temperature, gain and phase balance of analog components to better than 1%. Third-generation (3G) wireless systems impose severe requirements on level of *I/Q* balance. The need to achieve extreme levels of *I/Q* balance motivates us perform the complex conversion process in the DSP domain.

Figs. 5 and 6 present block diagrams of a second-generation multichannel receiver and transmitter in which the conversion from analog to digital (or digital to analog) occurs at IF rather than at baseband. Examining the receiver, we see that the down conversion of the separate channels is performed by a set of digital down converters and digital low-pass filters. The digital process can realize arbitrarily small levels of imbalance by controlling the number of bits involved in the arithmetic operations. Precision of coefficients used in the filtering process sets an upper bound to spectral artifact levels at -5 dB/bit so that 12-bit arithmetic can achieve image levels below -60 dB. Thus, the DSP-based complex down conversion does not introduce significant imbalance-related spectral terms. Similar comments apply to the DSP based up-conversions in the digital transmitter. The rule-of-thumb here is that the levels of spectral images are controlled to be below the quantizing noise floor of the ADC or DAC involved in the conversion process. A second advantage of the digital translation process is that the digital filters following or preceding the mixers are designed to have linear phase characteristics, a characteristic trivially simple to realize in digital nonrecursive filters [18].

The dynamic range and conversion speed of the ADC and DAC becomes the limiting factor in the application of the architectures shown in Figs. 5 and 6. The dynamic range of the converter is determined to first order by the number of bits in the converter with each bit contributing 6 dB [8]. The Nyquist criterion [9] establishes the minimum sample rate to obtain an alias-free representation of the sampled signal. The Nyquist criterion directs us to select the sample rate to exceed the two-sided bandwidth of the signal. Converters have the property that the product of sample rate and number of conversion levels is a constant [10]. This relationship is shown in (4), where b is the number of bits in the converter. Equation (5), a rearrangement of (4), shows how the number of bits varies inversely with the sample rate.

Fig. 7 is a graphical presentation of this relationship along with a scattering of data points showing the conversion speed versus precision performance exhibited by a number of current (mid-year 2002) ADCs. A useful rule-of-thumb is that a converter operating at 10-MHz sample rate can deliver 16-bit performance and that for every doubling of the sample rate results in a 1-bit (or 6 dB) reduction in conversion precision. The sloped line in Fig. 7 matches this rule. The intercept of this performance line is related to the aperture uncertainty of

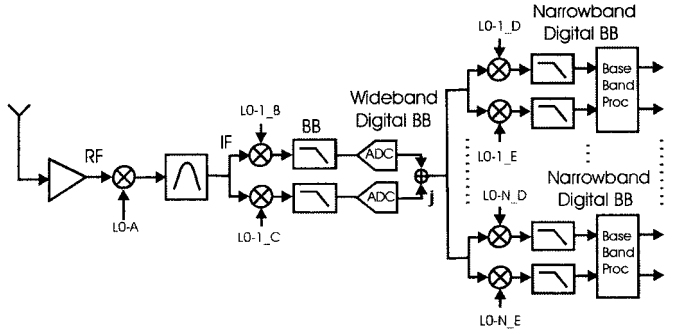

Fig. 8. Second-generation hybrid RF digital N-channel receiver.

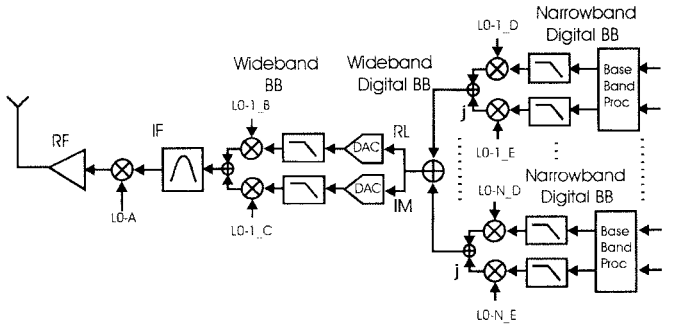

Fig. 9. Second-generation hybrid RF digital N-channel transmitter.

the conversion process, a parameter that improves slowly in response to advances in semiconductor technology.

$$\log_2\left(2^b f_{\mathrm{SAMPLE}}\right) = k \tag{4}$$

$$b = \log_2\{k\} - \log_2\left(f_{\mathrm{SAMPLE}}\right). \tag{5}$$

A final comment on ADCs is that the spurious terms generated by converter nonlinearities often exceed the quantizing noise levels described by the -6-dB per bit rule. The true performance measure of the ADC is the full bandwidth, full-scale spurious-free dynamic range (SPDR) [11].

The limited dynamic range available from high-speed ADCs restricts the range of applications for the architectures presented in Figs. 5 and 6 to IF center frequencies to the low to mid–100's of megahertz. To extend the application range of digital N-channel receivers and digital N-channels transmitters, we often use a hybrid scheme in which the initial complex down conversion is performed with analog *I/Q* mixers and the channelization is performed digitally after the ADC. The first conversion can be considered a block conversion to baseband that delivers the frequency band of interest to the DSP arena for subsequent channelization. The hybrid forms of the digital N-channel receiver and the digital N-channel transmitter are shown in Figs. 8 and 9, respectively. DSP techniques are applied to the digitized *I/Q* data to balance the gain and phase offsets in the analog ADC and DAC. DSP-based *I/Q* balance correction is a standard signal conditioning task in high-end, as well as consumer-based, receivers and transmitters.

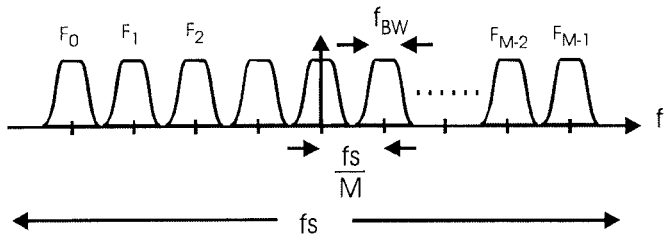

Fig. 10. Input spectrum of FDM signal to be channelized.

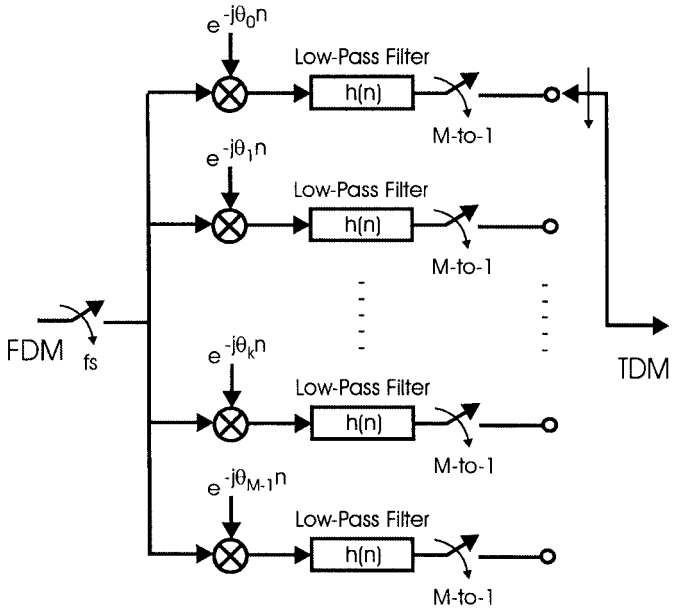

Fig. 11. Conventional channelizer as a replica of analog prototype: down converters, baseband filters, and resamplers.

II. Digital Down Conversion

In Section I, we described the process of sampling an analog IF signal or complex analog baseband signal containing the set of N frequency-division-multiplexed (FDM) channels to be further processed or channelized by DSP techniques. We consider the input signal to be composed of many equal-bandwidth equally spaced FDM channels, as shown in Fig. 10. These many channels are digitally down converted to baseband, bandwidth constrained by digital filters, and subjected to a sample rate reduction commensurate with the bandwidth reduction.

The signal-processing task can be performed as a replica of the analog prototype solution by a DSP-based set of independent down-conversion processes, as indicated in Fig. 11. For clarity of presentation, we describe how digital frequency denoted by the angle θ_k is derived from analog frequency f_k. This change of variables is shown in (6)–(8). Equation (6) presents a complex sinusoid of frequency $2\pi f_k$. We note that frequency is the time derivative of the time evolving phase angle $\theta(t)$ and has units of radians/second. The sampled data sinusoid is obtained by replacing the time variable "t" with the sampled time variable "nT," as shown in (7). Note that the units of the sample time variable are samples and seconds/sample, respectively. The angle formed by the product $2\pi f_k$ and T or by the equivalent

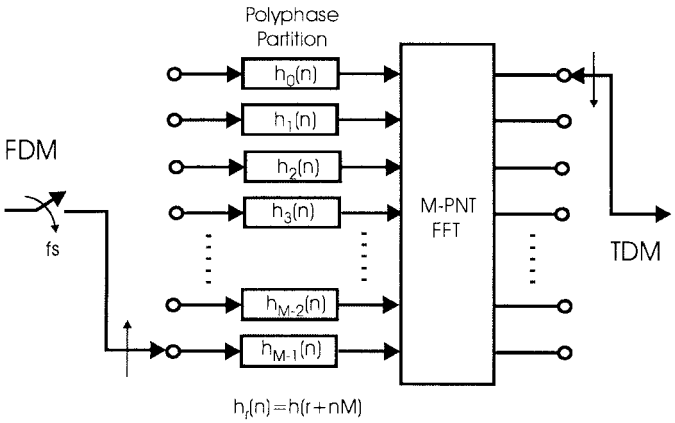

Fig. 12. Polyphase channelizer: resampler, all-pass partition, and FFT phase shifters.

term $2\pi f_k/f_S$, where $f_S = 1/T$, as shown in (8). Here, the product term $2\pi f_k T$, denoted by θ_k, has units of radians/second multiplied by seconds/sample or radians/sample.

$$g(t) = \exp(j2\pi f_k t) \tag{6}$$

$$g(n) = g(t)\big|_{t=nT} = \exp(j2\pi f_k nT) \tag{7}$$

$$g(n) = \exp\left(j2\pi \frac{f_k}{f_S} n\right) = \exp(j\theta_k n). \tag{8}$$

An alternate implementation performs the channelization as a single merged process called a polyphase N-path filter bank [12], as shown in Fig. 12. The polyphase filter bank partition offers a number of significant advantages relative to the set of individual down-conversion receivers. The primary advantage is reduced cost due to major reduction in system resources required to perform the multichannel processing.

The first sector in the communications community to make wide use of this form of the transmultiplxer was the Bell System network that used this structure in the early 1980s to modulate and demodulate analog single-sideband (SSB) FDM supergroups containing 60 4-kHz channels [13]. We now present a tutorial review to describe how the conventional channelizer is converted to the standard polyphase channelizers [14], [15]. This review contains simple equations and informative block diagrams representing the sequence of modifications that affect the transformation. We then extend the tutorial to incorporate a number of variations to perform secondary processing tasks along with the basic channelization task.

III. Transforming the Channelizer: First Step

The block diagram of a single channel of a conventional channelizer is shown in Fig. 13. This structure performs the standard operations of down conversion of the selected channel with a complex heterodyne low-pass filtering to reduce bandwidth to the channel bandwidth, and down sampling to a reduced rate commensurate with the reduced bandwidth. We mention that the down sampler is commonly referred to as a decimator, a term which means to destroy every tenth one. Since nothing is destroyed, and nothing happens in tenths, we prefer, and will continue to use the more descriptive name, down sampler.

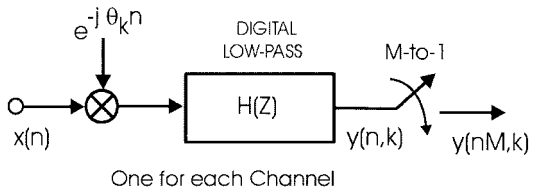

Fig. 13. kth channel of conventional channelizer.

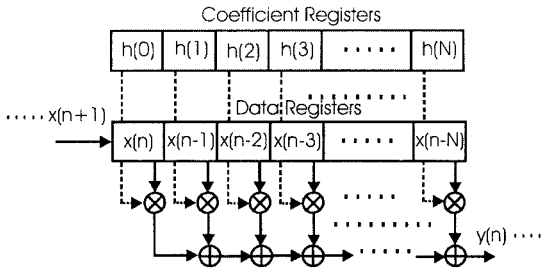

Fig. 14. Conceptual digital filter: coefficients and data registers, multipliers, and adders.

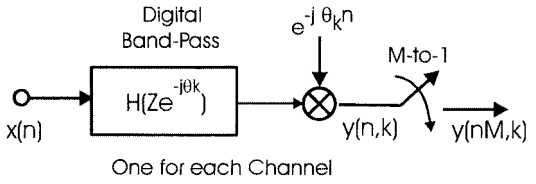

Fig. 15. Bandpass filter, kth channel of channelizer.

The expression for $y(n, k)$, the time series output from the kth channel, prior to resampling, is a simple convolution, as shown in the following:

$$\begin{aligned} y(n, k) &= \left[x(n)e^{-j\theta_k n}\right] * h(n) \\ &= \sum_{r=0}^{N-1} x(n-r)e^{-j\theta_k(n-r)}h(r). \end{aligned} \tag{9}$$

The output data from the complex mixer is complex and, hence, is represented by two time series $I(n)$ and $Q(n)$. The filter with real impulse response $h(n)$ is implemented as two identical filters, each processing one of the quadrature time series. The convolution process is performed by a simple digital filter that performs the multiply and add operations between data samples and filter coefficients extracted from two sets of addressed memory registers. One register set contains the data samples, while the other contains the coefficients that define the filter impulse response. This structure is shown in Fig. 14.

We can rearrange the summation of (9) to obtain a related summation reflecting the *equivalency theorem* [16]. The equivalency theorem states that the operations of down conversion followed by a low-pass filter are totally equivalent to the operations of a bandpass filter followed by a down conversion. The block diagram demonstrating this relationship is shown in Fig. 15, while the rearranged version of (9) is shown in (10). Note here, that the up-converted filter $h(n)\exp(j\theta_k n)$ is complex and, as such, its spectrum resides only on the positive frequency axis without a negative frequency image. This is not a common structure for an analog prototype because of the difficulty of forming a pair of analog quadrature filters exhibiting

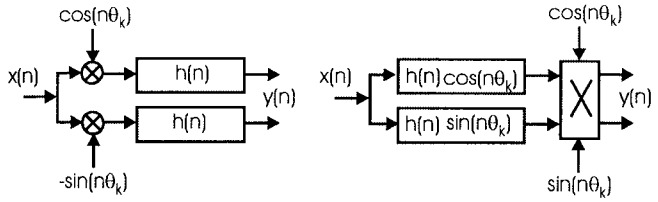

Fig. 16. Block diagrams illustrating equivalency between operations of heterodyne and baseband filter with bandpass filter and heterodyne.

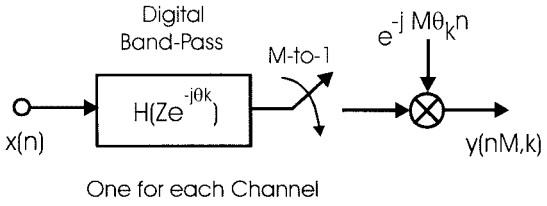

Fig. 17. Down-sampled down-converter bandpass kth channel.

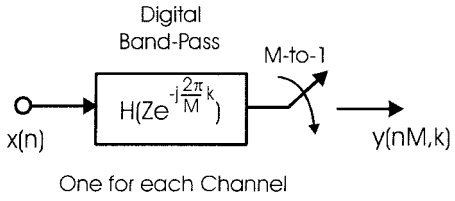

Fig. 18. Alias to baseband down-sampled down-converter bandpass kth channel.

a 90° phase difference across the filter bandwidth. The closest equivalent structure in the analog world is the filter pair used in image-reject mixers.

$$\begin{aligned} y(n,k) &= \sum_{r=0}^{N-1} x(n-r)e^{-j(n-r)\theta_k}h(r) \\ &= \sum_{r=0}^{N-1} x(n-r)e^{-jn\theta_k}h(r)e^{jr\theta_k} \\ &= e^{-jn\theta_k}\sum_{r=0}^{N-1} x(n-r)h(r)e^{jr\theta_k}. \end{aligned} \tag{10}$$

Applying the transformation suggested by the equivalency theorem to an analog prototype system does not make sense since it doubles the required hardware. We would have to replace a complex scalar heterodyne (two mixers) and a pair of low-pass filters with a pair of bandpass filters, containing twice the number of reactive components, and a full complex heterodyne (four mixers). If it makes no sense to use this relationship in the analog domain, why does it make sense in the digital world? The answer is found in the fact that we define a digital filter as a set of weights stored in coefficient memory. Thus, in the digital world, we incur no cost in replacing the pair of low-pass filters $h(n)$ required in the first option with the pair of bandpass filters $h(n)\cos(n\theta_k)$ and $h(n)\sin(n\theta_k)$ required for the second option. We accomplish this task by a simple download to the coefficient memory. The filter structures corresponding to the two sides of the equivalency theorem are shown in Fig. 16. Note the input signal interacts with the complex sinusoid as a product at the filter input or as a convolution in the filter weights.

We still have to address the matter of the full complex heterodyne required for the down conversion at the filter output rather than at the filter input. Examining Fig. 16, we note that following the output down conversion, we perform a sample rate reduction by retaining only one sample in every M samples. Recognizing that there is no need to down convert the samples we discard in the down sample operation, we choose to down sample only the retained samples. This is shown in Fig. 17.

We note in Fig. 17 that, when we bring the down converter to the low data-rate side of the resampler, we are, in fact, also down sampling the time series of the complex sinusoid. The rotation rate of the sampled complex sinusoid is θ_k and $M\theta_k$ radians per sample at the input and output, respectively, of the M-to-1 resampler.

This change in rotation rate is an aliasing affect, a sinusoid at one frequency or phase slope, appears at another phase slope when resampled. We now invoke a constraint on the sampled data center frequency of the down-converted channel. We choose center frequencies θ_k, which will alias to dc (zero frequency) as a result of the down sampling to $M\theta_k$. This condition is assured if $M\theta_k$ is congruent to 2π, which occurs when $M\theta_k = k2\pi$ or, more specifically, when $\theta_k = k2\pi/M$. The modification to Fig. 17 to reflect this provision is seen in Fig. 18. The constraint that the center frequencies be integer multiples of the output sample rate assures aliasing to baseband by the sample rate change. When a channel aliases to baseband by the resampling operation, the resampled related heterodyne defaults to a unity-valued scalar, which consequently is removed from the signal-processing path. Frequency offsets of the channel center frequencies, due to oscillator drift or Doppler effects, are removed after the down conversion by a baseband phase-locked loop (PLL)-controlled mixer. This baseband mixer operates at the output sample rate rather than at the input sample rate for a conventional down converter. We consider this required final mixing operation a post conversion task and allocate it to the next processing block.

The operations invoked by applying the equivalency theorem to the down-conversion process guided us to the following sequence of maneuvers:

1) slide the input heterodyne through the low-pass filters to their outputs;
2) doing so converts the low-pass filters to a complex bandpass filter;
3) slide the output heterodyne to the downside of the down sampler;
4) doing so aliases the center frequency of the oscillator;
5) restrict the center frequency of the bandpass to be a multiple of the output sample rate;
6) doing so assures alias of the selected passband to baseband by the resampling operation;
7) discard the now unnecessary heterodyne.

The spectral effect of these operations is shown in Fig. 19. The savings realized by this form of the down conversion is due to the fact we no longer require a quadrature oscillator, nor the pair of input mixers to effect the frequency translation.

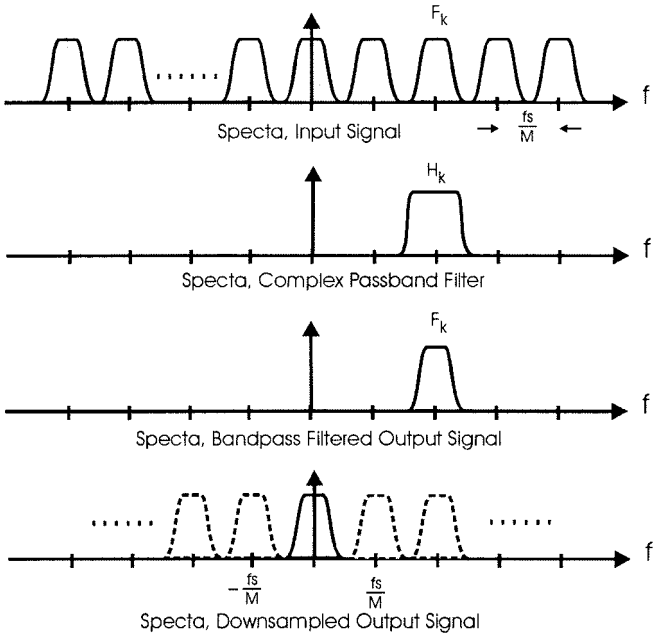

Fig. 19. Spectral description of down conversion realized by a complex bandpass filter at a multiple of output sample rate, aliased to baseband by output resampling.

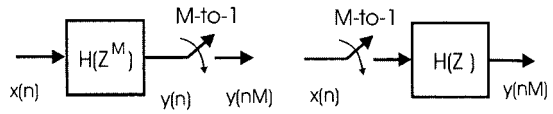

Fig. 20. *Noble identity*: a filter processing every Mth input sample followed by an output M-to-1 down sampler is the same as an input M-to-1 down sampler followed by a filter processing every Mth input sample.

IV. Transforming the Channelizer: Second Step

Examining Fig. 18, we note that the current configuration of the single-channel down converter involves a bandpass filtering operation followed by a down sampling of the filtered data to alias the output spectrum to baseband. Following the idea developed in Section III that led us to down convert only those samples retained by the down sampler, we similarly conclude that there is no need to compute the output samples from the passband filter that will be discarded by the down sampler. We now interchange the operations of filter and down sample with the operations of down sample and filter. The process that accomplishes this interchange is known as the *noble identity* [17], which we now review.

The noble identity is compactly presented in Fig. 20, which we describe with similar conciseness by "The output from a filter $H(Z^M)$ followed by an M-to-1 down sampler is identical to an M-to-1 down sampler followed by the filter $H(Z)$." The Z^M in the filter impulse response tell us that the coefficients in the filter are separated M-samples rather than the more conventional one sample delay between coefficients in the filter $H(Z)$. We must take care to properly interpret the operation of the M-to-1 down sampler. The interpretation is that the M-to–1 down-sampled time series from a filter processing every Mth input sample presents the same output by first

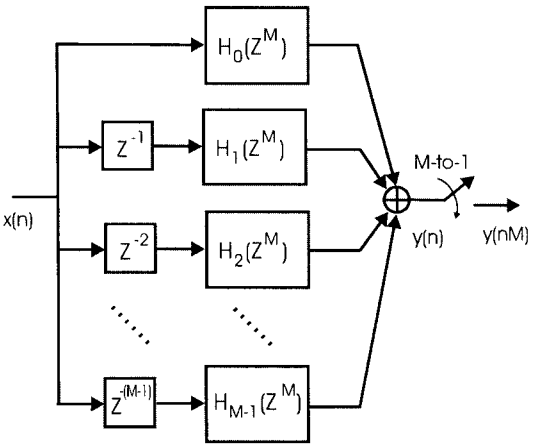

Fig. 21. M-path partition of prototype low-pass filter with output resampler.

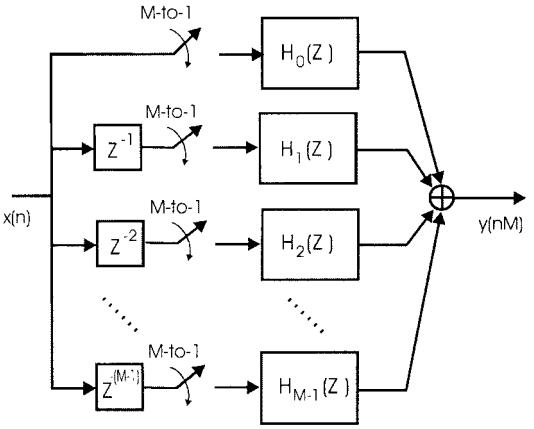

Fig. 22. M-path partition of prototype low-pass filter with input resamplers.

down sampling the input by M-to-1 to discard the samples not used by the filter to compute the retained output samples and then operating the filter on the retained input samples. The noble identity works because M-samples of delay at the input clock rate is the same interval as one-sample delay at the output clock rate.

We might ask, "Under what condition does a filter manage to operate on every Mth input sample?" We answer this query by rearranging the description of the filter to establish this condition so that we can invoke the noble identity. This rearrangement starts with an initial partition of the filter into M-parallel filter paths. The Z-transform description of this partition is presented in (11)–(14), which we interpret in Figs. 21–23. For ease of notation, we first examine the baseband version of the noble identity and then trivially extend it to the passband version.

$$\begin{aligned} H(Z) &= \sum_{n=0}^{N-1} h(n)Z^{-n} \\ &= h(0) + h(1)Z^{-1} + h(2)Z^{-2} \\ &\quad + h(3)Z^{-3} + \cdots + h(N-1)Z^{-(N-1)}. \end{aligned} \tag{11}$$

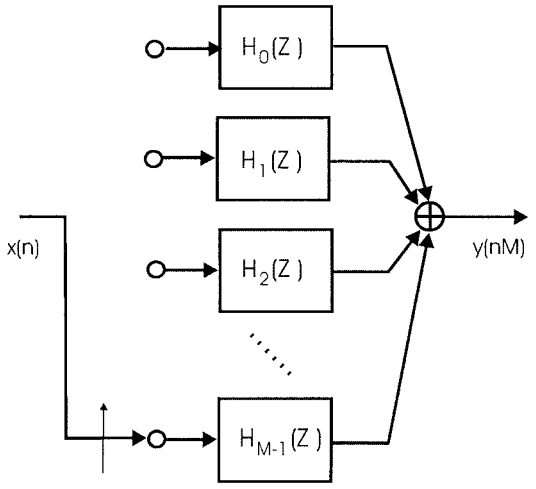

Fig. 23. M-path partition of prototype low-pass filter with input path delays and M-to-1 resamplers replaced by input commutator.

Anticipating the M-to-1 resampling, we partition the sum shown in (11) to a sum of sums, as shown in (12) at the bottom of this page. This partition maps a one-dimensional array of weights (and index markers Z^{-n}) to a two-dimensional array. This mapping is sometimes called lexicographic, for natural order, a mapping that occurs in the Cooley–Tukey fast Fourier transform (FFT). In this mapping, we load an array by columns, but process the array by rows. In our example, the partition forms columns of length M containing M successive terms in the original sum, and continues to form adjacent M-length columns until we account for all the elements of the original one-dimensional array shown in (12).

We note that the first row of the two-dimensional array is a polynomial in Z^M, which we will denote $H_0\,(Z^M)$, a notation to be interpreted as an addressing scheme to start at index 0 and increment in stride of length M. The second row of the same array, while not a polynomial in Z^M, is made into one by factoring the common Z^{-1} term and then identifying this row as $Z^{-1}\,H_1(Z^M)$. It is easy to see that each row of (12) can be described as $Z^{-r}H_r\,(Z^M)$ so that (12) can be rewritten in a compact form, as shown in the following:

$$H(Z) = H_0\big(Z^M\big) + Z^{-1}H_1\big(Z^M\big) + Z^{-2}H_2\big(Z^M\big) + \cdots + Z^{-(M-1)}H_{(M-1)}\big(Z^M\big). \quad (13)$$

We rewrite (13) in the traditional summation form, as shown in (14), which describes the original polynomial as a sum of delayed polynomials in Z^M.

$$H(Z) = \sum_{r=0}^{M-1} Z^{-r}H_r\big(Z^M\big) = \sum_{r=0}^{M-1} Z^{-r} \sum_{n=0}^{(N/M)-1} h(r+nM)Z^{-Mn} \quad (14)$$

The block diagram reflecting this M-path partition of a resampled digital filter is shown in Fig. 21. The output of the filter is the resampled sum of the output of the M separate filter stages along the M-paths. We pull the resampler through the output summation element and down sample the separate outputs, only performing the output sum for the retained output sample points. With the resamplers at the output of each filter, which operates on every Mth input sample, we are prepared to invoke the noble identity and pull the resampler to the input side of each filter stage. This is shown in Fig. 22. The input resamplers operate synchronously, all closing at the same clock cycle. When the switches close, the signal delivered to the filter on the top path is the current input sample. The signal delivered to the filter one path down is the content of the one-stage delay line, which, of course, is the previous input sample. Similarly, as we traverse the successive paths of the M-path partition, we find upon switch closure, that the kth path receives a data sample delivered k samples ago. We conclude that the interaction of the delay lines in each path with the set of synchronous switches can be likened to an input commutator that delivers successive samples to successive legs of the M-path filter. This interpretation is shown in Fig. 23.

We now complete the final steps of the transform that changes a standard mixer down converter to a resampling M-path down converter. We note and apply the frequency translation property of the Z-transform [18]. This property is illustrated and stated in (15). Interpreting the relationship presented in (15), we note that, if $h(n)$, the impulse response of a baseband filter, has a Z-transform $H(Z)$, then the sequence $h(n)e^{+j\theta n}$, the impulse response of a passband filter, has a Z-transform $H(Ze^{-j\theta n})$. Simply stated, we can convert a low-pass filter to a bandpass filter by associating the complex heterodyne terms of the mod-

$$H(Z) = \begin{array}{ccccccc} h(0) & + & h(M+0)Z^{-M} & + & h(2M+0)Z^{-(2M+0)} & + & \cdots \\ h(1)Z^{-1} & + & h(M+1)Z^{-(M+1)} & + & h(2M+1)Z^{-(2M+1)} & + & \cdots \\ h(2)Z^{-2} & + & h(M+2)Z^{-(M+2)} & + & h(2M+2)Z^{-(2M+2)} & + & \cdots \\ h(3)Z^{-3} & + & h(M+3)Z^{-(M+3)} & + & h(2M+3)Z^{-(2M+3)} & + & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \\ h(M-1)Z^{-(M-1)} & + & h(2M-1)Z^{-(2M-1)} & + & h(3M-1)Z^{-(3M-1)} & + & \cdots \end{array} \quad (12)$$

ulation process either with the filter weights or with the delay elements storing the filter weights.

If

$$\begin{aligned} H(Z) = {} & h(0) + h(1)Z^{-1} + h(2)Z^{-2} + \cdots \\ & + n(N-1)Z^{-(N-1)} \\ = {} & \sum_{n=0}^{N-1} h(n)Z^{-n} \end{aligned}$$

and

$$\begin{aligned} G(Z) = {} & h(0) + h(1)e^{j\theta}Z^{-1} + h(2)e^{j2\theta}Z^{-2} + \cdots \\ & + h(N-1)e^{j(N-1)\theta}Z^{-(N-1)} \\ = {} & h(0) + h(1)\left[e^{-j\theta}Z\right]^{-1} + h(2)\left[e^{-j\theta}Z\right]^{-2} + \cdots \\ & + h(N-1)\left[e^{-j\theta}Z\right]^{-(N-1)} \\ = {} & \sum_{n=0}^{N-1} h(n)\left[e^{-j\theta}Z\right]^{-n} \end{aligned} \tag{15}$$

then

$$G(Z) = H(Z)|_{Z=e^{-j\theta}Z} = H\left(e^{-j\theta}Z\right).$$

We now apply this relationship to (10) or, equivalently, to Fig. 23 by replacing each Z with $Ze^{-j\theta}$, or perhaps more clearly, replacing each Z^{-1} with $Z^{-1}e^{j\theta}$, with the phase term θ satisfying the congruency constraint of the previous section that $\theta = k(2\pi/M)$. Thus, Z^{-1} is replaced with $Z^{-1}e^{jk(2\pi/M)}$, and Z^{-M} is replaced with $Z^{-M}e^{jkM(2\pi/M)}$. By design, the kMth multiple of $2\pi/M$ is a multiple of 2π for which the complex phase rotator term defaults to unity, or in our interpretation, aliases to baseband (dc). The default to unity of the complex phase rotator occurs in each path of the M-path filter shown in Fig. 24. The nondefault complex phase angles are attached to the delay elements on each of the M paths. For these delays, the terms Z^{-r} are replaced by the terms $Z^{-r}e^{jkr(2\pi/M)}$. The complex scalar $e^{jkr(2\pi/M)}$ attached to each path of the M-path filter can be placed anywhere along the path and, in anticipation of the next step, we choose to place the complex scalar after the down-sampled path filter segments $H_r(Z)$. This is shown in Fig. 24.

The modification to the original partitioned Z-transform of (14) to reflect the added phase rotators of Fig. 24 is shown in the following:

$$H\left(Ze^{-j\,(2\pi/M)\,k}\right) = \sum_{r=0}^{M-1} Z^{-r}e^{j\,(2\pi/M)\,rk}H_r(Z). \tag{16}$$

The computation of the time series obtained from the output summation in Fig. 24 is shown in (17). Here, the argument nM reflects the down-sampling operation, which increments through the time index in stride of length M, delivering every Mth sample of the original output series. The variable $y_r(nM)$ is the nMth sample from the filter segment in the rth path, and $y(nM, k)$ is the nMth time sample of the time series from the kth center frequency. Remember that the down-converted center

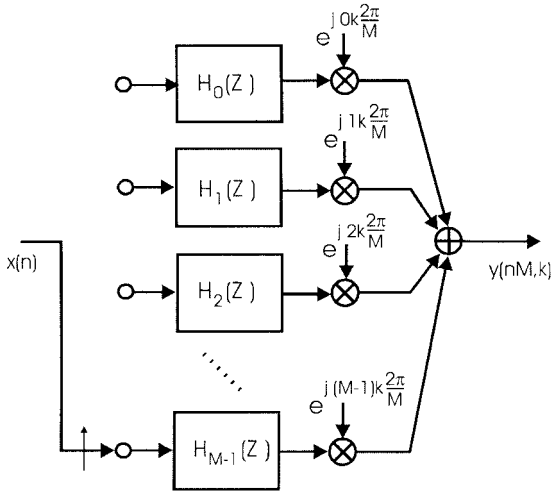

Fig. 24. Resampling M-path down converter.

frequencies located at integer multiples of the output sample frequency are the frequencies that alias to zero frequency under the resampling operation. Note the output $y(nM, k)$ is computed as a phase coherent summation of the M output series $y_r(nM)$. This phase coherent sum is, in fact, a discrete Fourier transform (DFT) of the M-path outputs, which can be likened to beam forming the output of the path filters.

$$y(nM, k) = \sum_{r=0}^{M-1} y_r(nM)e^{j\,(2\pi/M)\,rk}. \tag{17}$$

The beam-forming perspective offers an interesting insight to the operation of the resampled down-converter system we have just examined. The reasoning proceeds as follows: the commutator delivering consecutive samples to the M input ports of the M-path filter performs a down-sampling operation. Each port of the M-path filter receives data at one Mth of the input rate. The down sampling causes the M-to-1 spectral folding, effectively translating the M-multiples of the output sample rate to baseband. The alias terms in each path of the M-path filter exhibit unique phase profiles due to their distinct center frequencies and the time offsets of the different down-sampled time series delivered to each port. These time offset are, in fact, the input delays shown in Fig. 22 and (18). Each of the aliased center frequency experiences a phase shift shown in (18) equal to the product of its center frequency and the path time delay.

$$\phi(r, k) = \omega_k \Delta T_r = 2\pi \frac{f_s}{M} krT_s = 2\pi \frac{f_s}{M} kr \frac{1}{f_s} = \frac{2\pi}{M} kr. \tag{18}$$

The phase shifters of the DFT perform phase coherent summation, very much like that performed in narrow-band beam forming, extracting from the myriad of aliased time series, the alias with the particular matching phase profile. This phase-sensitive summation aligns contributions from the desired alias to realize the processing gain of the coherent sum while the remaining alias terms, which exhibit rotation rates corresponding to the M roots of unity, are destructively canceled in the summation.

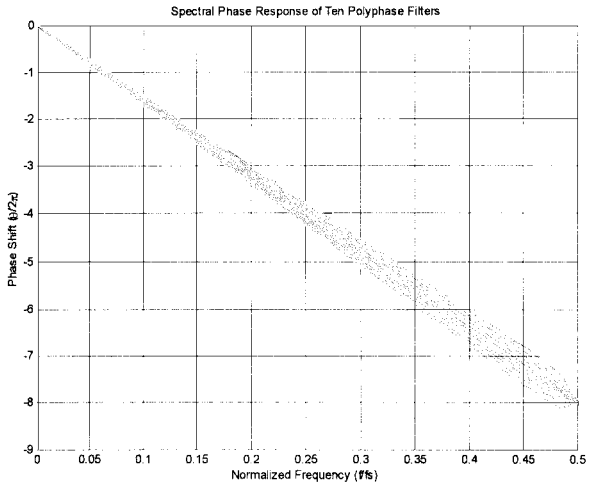

Fig. 25. Phase profiles for ten stages of ten-path polyphase partition.

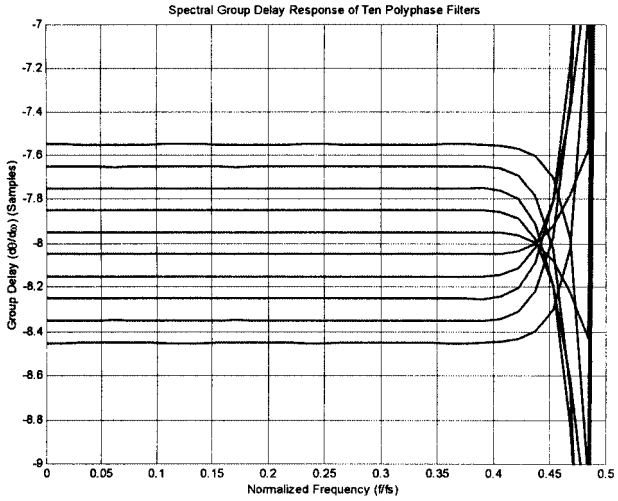

Fig. 26. Group delay profiles for ten stages of ten-path polyphase partition.

The inputs to the M-path filter are not narrow-band, and phase shift alone is insufficient to effect the destructive cancellation over the full bandwidth of the undesired spectral contributions. Continuing with our beam-forming perspective, to successfully separate wide-band signals with unique phase profiles due to the input commutator delays, we must perform the equivalent of time-delay beam forming. The M-path filters, obtained by M-to-1 down sampling of the prototype low-pass filter supply the required time delays. The M-path filters are approximations to all-pass filters, exhibiting, over the channel bandwidth, equal ripple approximation to unity gain and the set of linear phase shifts that provide the time delays required for the time-delay beam-forming task.

The filter achieves this property by virtue of the way we partitioned the low-pass prototype. Each of the M-path filters, filter $h_r(n)$, for instance, with weights $h(r+nM)$ is formed by starting with an initial offset of "r" samples and then incrementing by stride of M samples. The initial offsets, unique to each path, are the source of the different linear phase-shift profiles. It is for this reason, the different linear phase profiles, that the filter partition is known as *a polyphase* filter. The phase-shift and group-delay profiles for a ten-path filter are shown in Figs. 25 and 26. These figures are part of the output suite of figures formed by the MATLAB m-file **filter_ten** contained in Appendix 1 of the electronic version of this paper on the CDROM accompanying this issue. This file synthesizes a ten-stage polyphase channelizer and presents input and output time series and spectra.

A useful perspective is that the phase rotators following the filters perform phase alignment of the band center for each aliased spectral band while the polyphase filters perform the required differential phase shift across these same channel bandwidths. When the polyphase filter is used to down convert and down sample a single channel, the phase rotators are implemented as external complex products following each path filter. When a small number of channels are being down converted and down sampled, appropriate sets of phase rotators can be applied to the filter stage outputs and summed to form each channel output. We take a different approach when the number of channels becomes sufficiently large. Here, sufficiently large means on the order of $\log_2(N)$. Since the phase rotators following the polyphase filter stages are the same as the phase rotators of a DFT, we can use the DFT to simultaneously apply the phase shifters for all of the channels we wish to extract from the aliased signal set. This is reminiscent of phased-array beam forming. For computational efficiency, the FFT algorithm implements the DFT.

It is useful to once again examine Fig. 12 in which the polyphase filter and FFT was first presented as a channelized receiver. Think of the many input arms of the FFT as being coupled through a set of distributed phase rotators to each output port of the FFT with each output port accessed through a different vector of phase slopes. Readers may recognize that the phase shifts between input and output ports of the FFT are the same as those forming the Butler matrix used in phased-array beam forming [19], [20].

At this point, it is instructive to make a comparison of the conventional mixer down converter and the resampled polyphase down converter. The input to either process can be real or complex. In the mixer down-converter model, a separate mixer pair and filter pair must be assigned to each channel of the channelizer and these mixers must all operate at the high input data rate, prior to down sampling. By way of contrast, in the resampled polyphase, there is only one low-pass filter required to service all the channels of the channelizer, and this single filter accommodates all the channels as co-occupying alias contributors of the baseband bandwidth. This means that all the processing performed in the resampled polyphase channelizer occurs at the low-output sample rate. When the input signal is real, there is another significant difference between the two processes. In the mixer down-converter model, the signal is made complex by the input mixers as we enter the process, which means that the low-pass filtering task requires two filters, one for each of the quadrature components, while in the resampling channelizer, the signal is made complex by the phase rotators as we leave the process; consequently, we require only one partitioned low-pass filter to process the real input signal.

Before moving on to the next topic, let us summarize what we have accomplished to this point. The commutator performs an input sample rate reduction by commutating successive input samples to selected paths of the M-path filter. Sample rate reduction occurring prior to any signal processing causes spectral regions residing at multiples of the output sample rate to alias to baseband. This desired result allows us to replace the many down converters of a standard channelizer, implemented with dual mixers, quadrature oscillators, and bandwidth reducing filters, with a collection of trivial aliasing operations performed in a single partitioned and resampled filter.

The partitioned M-path filter performs the task of aligning the time origins of the offset sampled data sequences delivered by the input commutator to a single common output time origin. This is accomplished by the all-pass characteristics of the M-path filter sections that apply the required differential time delay to the individual input time series. The DFT performs the equivalent of a beam-forming operation; the coherent summation of the time-aligned signals at each output port with selected phase profiles. The phase coherent summation of the outputs of the M-path filters separate the various aliases residing in each path by constructively summing the selected aliased frequency components located in each path, while simultaneously destructively canceling the remaining aliased spectral components.

This section of the presentation emphasized the structure of an N-channel polyphase receiver. A similar exposition can be mounted for the N-channel polyphase transmitter. Rather than repeat the many steps that took us to the polyphase structure from the more conventional structure, we will merely comment that the transmitter is the dual process of the receiver. The dual process simply reverses all signal flow of the original process. In the dual structure, we enter the N-channel process at the FFT and leave the process by the polyphase commutator. Reversing the signal flow results in a process that up-samples and up-converts rather than one that down converts and down samples. The two processes are shown in Fig. 27.

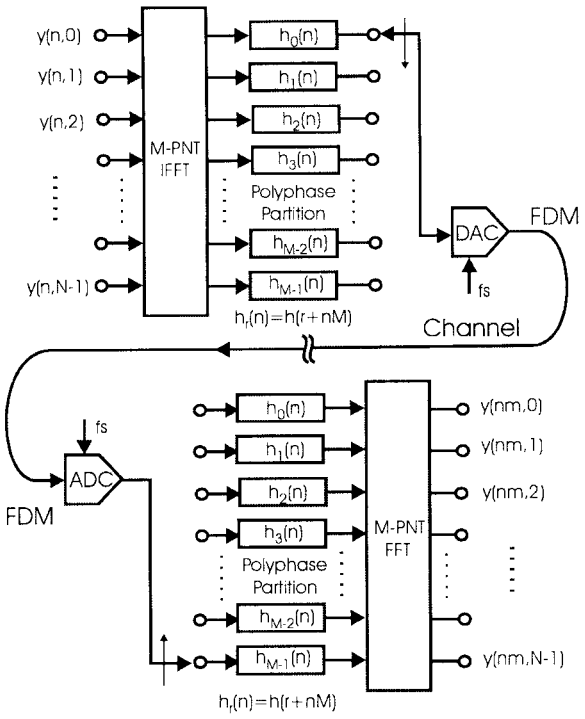

Fig. 27. N-channel transmitter and N-channel receiver. Dual circuits formed with polyphase filters, FFT, and commutator.

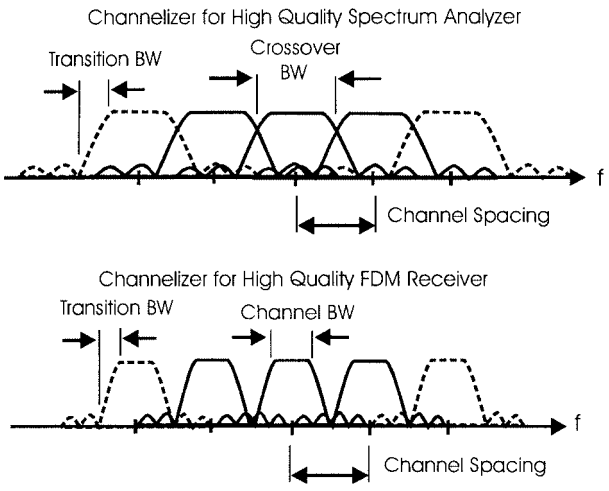

Fig. 28. Spectral characteristics of two channelizers with same channel spacing one for spectral analysis and one for FDM channel separation.

V. Arbitrary Bandwidth, Spectral Spacing, and Output Sample Rates

We now address the interaction and coupling, or lack of coupling, between the parameters that define the polyphase filter bank [21]. We observe that the DFT performs the task of separating the channels after the polyphase filter so it is natural to conclude that the transform size is locked to the number of channels and this is a correct assessment. We then note that the filter bandwidth is determined by the weights of the low-pass prototype and that this bandwidth and spectral shape is common to all the channels. We comment on filter length later when we address total computation complexity of the polyphase channelizer.

In standard channelizer designs, the bandwidth of the prototype is specified in accord with the end use of the channelizer outputs. For instance, when the channelizer is used as a spectral analyzer, the channels may be designed to have a specified passband attenuation such as -3, -1, or -0.1 dB at their crossover frequency and have a specified stopband attenuation at their adjacent center frequency. Overlap of adjacent channel responses permits a narrow-band input signal to straddle one or more output channels, which is a common occurrence in the spectral analysis of signals with arbitrary bandwidth and center frequencies. On the other hand, when a channelizer is used to separate adjacent communication channels, which are characterized by known center frequencies and known controlled nonoverlapping bandwidths, the channelizer must preserve separation of the channel outputs. Inadequate adjacent channel separation results in adjacent channel interference. Typical spectral responses for channel bandwidths corresponding to the two scenarios just described are shown in Fig. 28.

The polyphase filter channelizer uses the input M-to-1 resampling to alias the spectral terms residing at multiples of the output sample rate to baseband. This means that, for the standard polyphase channelizer, the output sample rate is the same as the channel spacing. For the case of the spectral analyzer application operating at this sample rate permits aliasing of the band edges into the down-sampled passband. When operated in this mode, the system is called a maximally decimated filter bank. For the case of the communication channelizer, operating at this rate satisfies the Nyquist criterion, permitting the separation of the channels with an output rate that avoids band-edge aliasing. An example of a spectrum that would require this mode of operation is the quadrature amplitude modulation (QAM) channels of a digital cable system. Here, the channels are separated by 6-MHz centers and operate with 20% excess bandwidth, square-root Nyquist shaped spectra, at symbol rates of 5.0 MHz. The sampled data rate from a cable channelizer would be 6.0 megasamples/s and would satisfy the Nyquist sample criterion.

Another example used on the upstream cable link (subscriber to head end) is a set of channels operating at 128-kHz symbol rate, with square-root Nyquist spectra having 50% excess bandwidth. The channels are separated by 192 kHz, which matches the two-sided bandwidth of the shaped channel. Here too, when operated at 192 kHz, the sample rate, matched to the channel spacing, satisfies the Nyquist sample-rate criterion. Systems that channelize and supply samples of the Nyquist-shaped spectra most often present the sampled data to an interpolator to resample the time series from the collected Nyquist rate to two times the symbol rate. For the two example cited earlier, the 6-Ms/s 5-Msymbol TV signal would have to be resampled by 5/3 to obtain 10 Ms/s, and the 192-ks/s 128 K-symbol reverse channel signal would have to be resampled by 4/3 to obtain 256 ks/s. These are not difficult tasks and it is done quite regularly in single-channel receivers. This represents significant computational burden if we are to perform this interpolative resampling for every channel.

The conventional way we use the M-path polyphase filter bank is to deliver M-input samples to the M-paths and then compute outputs from each channel at the rate f_s/M. The thought may occur to us, "Is it possible to operate the polyphase filter bank in a manner that the output rate is higher than one Mth of the input rate?" For instance, can we operate the bank so that we deliver $M/2$ inputs prior to computing an output sample rather than delivering M input samples before computing an output sample? Increasing the output sample rate of the polyphase channel bank by a factor of two makes subsequent interpolation tasks less expensive since the spectra of the output signals would already be oversampled by a factor of two with increased spectral separation. Operation in this mode would also permit channelization of overlapped channels without aliasing of the spectral transition bands. The alias-free partition is handy in applications requiring perfect reconstruction of the original time series from spectrally partitioned sub channels, a requirement sometimes imposed in receivers used for electronic warfare (EW) applications. For the record, a polyphase filter bank can be operated with an output sample rate any rational ratio times the input sample rate. With minor modifications, the filter can be operated with totally arbitrary ratios between input and output sample rates. This is true for the sample rate reduction imbedded in a polyphase receiver, as well as for the sample rate increase embedded in a polyphase transmitter.

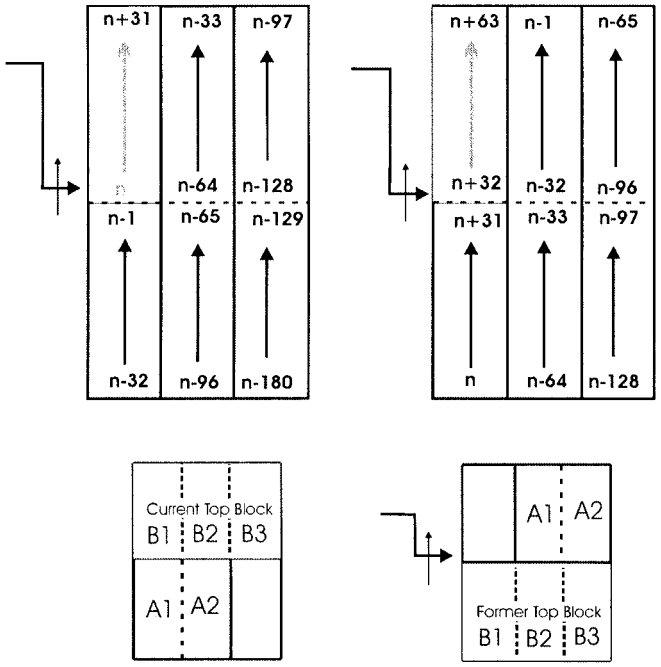

Fig. 29. Data memory loading for successive 32-point sequences in a 64-stage polyphase filter.

We first examine the task of increasing the output sample rate from the polyphase filter bank from f_s/M to 2 f_s/M. We accomplish this by controlling the commutator delivering input data samples to the polyphase stages. We normally deliver M inputs to the M-stage filter by delivering successive input samples starting at port $M-1$ progressing up the stack to port 0 and by doing so deliver M inputs per output for an M-to-1 down sampling. To obtain the desired $(M/2)$-to-1 down sampling, we deliver $M/2$ successive input samples starting at port $(M/2)-1$ progressing up the stack to port 0. The $M/2$ addresses to which the new $M/2$ input samples are delivered are first vacated by their former contents, the $M/2$ previous input samples. All the samples in the two-dimensional filter undergo a serpentine shift of $M/2$ samples with the $M/2$ samples in the bottom half of the first column sliding into the $M/2$ top addresses of the second column, while the $M/2$ samples in the top half of the second column slide into the $M/2$ addresses in the bottom half of the second column and so on. This is equivalent to performing a linear shift through the prototype one-dimensional filter prior to the polyphase partition. In reality, we do not perform the serpentine shift, but rather perform a swap of two memory banks, as shown in Fig. 29 for successive sequences of length 32 being delivered to a filter bank with 64 stages.

We continue this discussion with comments on the 64-stage example. After each 32-point data sequence is delivered to the partitioned 64-stage polyphase filter, the outputs of the 64 stages are computed and conditioned for delivery to the 64-point FFT. The data shifting into the polyphase filter stages causes a fre-

quency-dependent phase shift of the form shown in (19). The time delay due to shifting is nT, where n is the number of samples and T is the interval between samples. The frequencies of interest are integer multiple "k" of $1/M$th of the sample rate $2\pi/T$. Substituting these terms in (19) and canceling terms, we obtain the frequency-dependent phase shift shown in (20). Here, we see that, for time shifts "n" equal to multiples of M, the phase shift is a multiple of 2π and contributes no offset to the spectra observed at the output of the FFT. The M-sample time shift is the time shift applied to the data in the normal use of the polyphase filter. Now suppose that the time shift is $M/2$ time samples. When substituted in (20), we find a frequency-dependent phase shift of $k\pi$ from which we conclude that odd-indexed frequency terms experience a phase shift of π radians for each successive $N/2$ shift of input data.

$$\theta(\omega) = \Delta t \cdot \omega \tag{19}$$

$$\theta(\omega) = nT \cdot k\,\frac{1}{M}\,\frac{2\pi}{T} = \frac{nk}{M}\,2\pi. \tag{20}$$

This π radian phase shift is due to the fact that the odd indexed frequencies alias to the half sample rate when the input signal is down sampled by $M/2$. We can compensate for the alternating signs in successive output samples by applying the appropriate phase correction to the spectral data as we extract successive time samples from the odd-indexed frequency bins of the FFT. The phase correction here is trivial, but for other down-sampling ratios, the residual phase correction would require a complex multiply at each transform output port. Alternatively, we can cancel the frequency-dependent phase shift by applying a circular time shift of $N/2$ samples to the vector of samples prior to their presentation to the FFT. As in the case of the serpentine shift of the input data, the circular shift of the polyphase filter output data is implemented as a data swap. This data swap occurs on alternate input cycles and a simple two-state machine determines for which input cycle the output data swap is applied. This option is shown in Fig. 30.

We are now prepared to examine the process that permits resampling of the polyphase filter bank by any rational ratio. We first demonstrated the modification to the standard polyphase structure to support $N/2$ down sampling. The modifications involved a serpentine shift of input memory and a circular shift of output memory that are both implemented by data swaps.

There are relatively few papers in the open literature that describe arbitrary resampling embedded in the polyphase filter bank. Our group at San Diego State University (SDSU), San Diego, CA, had written an application note in 1989 [3] that demonstrated how to obtain arbitrary sample rates from the polyphase filter bank by use of the commutator mechanism with serpentine data shifts just described. An earlier contribution [22] presented the technique for absorbing the phase shifts at the output of the FFT as circular shifts of the data vector at the input to the FFT. Numerous papers have been written describing arbitrary resamplers for baseband interpolators. We believe this paper is the first open literature description of the two merged techniques for polyphase channelizers. The technique is based on the observation that the commutator is the component in the polyphase filter bank that effects and controls the resampling, not the spacing between the adjacent

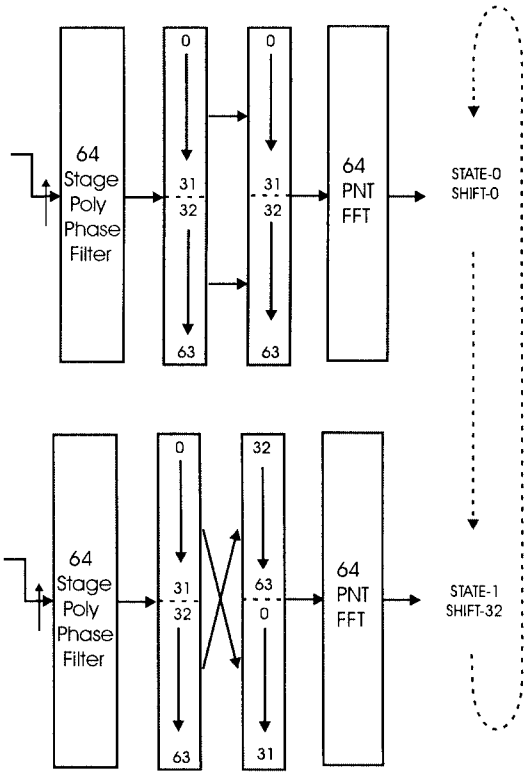

Fig. 30. Cyclic shift of input data to FFT to absorb phase shift due to linear time shift of data through polyphase filter.

channels. This is true even though it was the resampling process that first guided us to select the channel spacing so we could access the aliasing to baseband. As we just demonstrated, there are two modifications to the polyphase-resampling filter required to obtain arbitrary resampling. These modifications would normally lead to an exercise in time-varying reside index mapping of the two-dimensional input data array. If we limit the presentation to the index-mapping process, we would develop little insight into the process and further would be bored to tears. Instead, we derive and illustrate the modifications by examining a specific example and observe the process develop.

VI. Second Example Processing Task

Here, we describe a more general resampling channelizer and present the process that guides us to the solution. The problem is this: we have a signal containing 50 FDM channels separated by 192-kHz centers containing symbols modulated at 128 kHz by square-root Nyquist filters with 50% excess bandwidth. Our task is to baseband channelize all 50 channels and output data samples from each channel at 256 ks/s, which is two samples per symbol. We start by selecting a sample rate and transform size matched to the channel spacing. We select a 64-point FFT to span the 50 channels with the excess bandwidth allocated to the analog antialias filter. Thus, the sample rate for the collected spectra is 64 times the 192-kHz channel spacing or 12.288 MHz. These are complex samples formed from either a baseband block conversion or a digital down conversion and resampling from a digital IF, often centered at the quarter sample rate. The desired output sample rate is 2×128 or 256 kHz. The ratio between the input and output sample rates

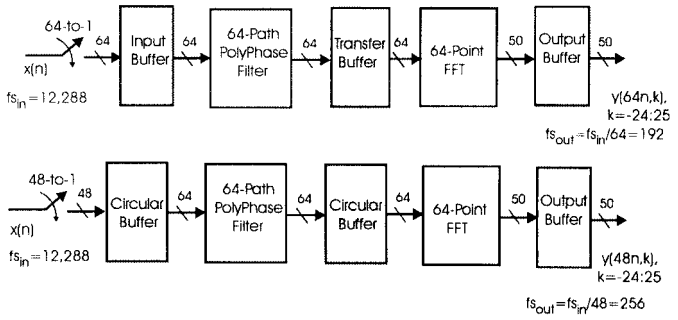

Fig. 31. Maximally decimated filter bank structure and modified two-sample-per-symbol filter bank structure.

is the resampling ratio, which is 12 288/256 or 48-to-1. Thus, our task is to use the 64-point DFT to separate and deliver 50 of the possible 64 channels spanned by the sample rate, but to deliver one output sample for every 48 input samples. Fig. 31 is a block diagram of the original maximally decimated version of the 64-stage polyphase channelizer and the modified form of the same channelizer. The difference in the two systems resides in the block inserted between the 64-stage polyphase filter and the 64-point FFT. Remarkably, the inserted block performs no computation, but rather only performs a set of scheduled circular buffers shifts. We are about to develop and describe the operation of the circular buffer stage and state machine scheduler.

Our first task is to modify the input commutator to support the 48-to-1 down sample rather than the standard 64-to-1 down sample. This is an almost trivial task. We arrange for the modified resampling by keeping the 64-path filter, but stripping 16 ports from the commutator. The commutator for the standard 64-point polyphase filter starts at port 63 and delivers 64 successive inputs to ports 64, 63, 62, and so on through 0, the modified commutator starts at port 47 and delivers 48 successive inputs to ports 47, 46, 45, and so on through 0. Input memory for the 64-path filter must be modified to support this shortened commutator input schedule. The mapping structure of the reindexing scheme is best seen in the original one-dimensional prototype filter shown in Fig. 32 and then transferred to the two-dimensional polyphase partition.

Fig. 32 presents the memory content for a sequence of successive 48-point input data blocks presented to the 64-point partitioned prototype filter. In this figure, we have indicated the interval of 64-tap boundaries that become the columns of the two-dimensional array, as well as the boundaries of successive 48-point input blocks that are presented to the input array. Successive input blocks start loading at address 47 and work up to address 0. The beginning and end of this interval are denoted by the tail and arrow, respectively, of the left-most input interval in the filter array. As each new 48-point input array is delivered, the earlier arrays must shift to the right-hand side. These shifting array blocks cross the 64-point column boundaries and, hence, move to adjacent columns in the equivalent two-dimensional partition. This crossing can be visualized as a serpentine shift of data in the two-dimensional array or, equivalently, as a circular row buffer down shift of 48 rows in the poly phase memory with a simultaneous column buffer right shift of the input data column. The operation of this circular buffer is illustrated in Fig. 33, which indicates the indexes of input data for two input cycles. Here, we see that, between two successive input cycles, the rows in the top one-fourth of memory translates to the bottom fourth, while the bottom three-fourths of rows translates up one-fourth of memory. We also see that the columns in the bottom three-fourths shift to the right-hand side on column during the circular row translations. The next input array is loaded in the left-most column of this group of addresses.

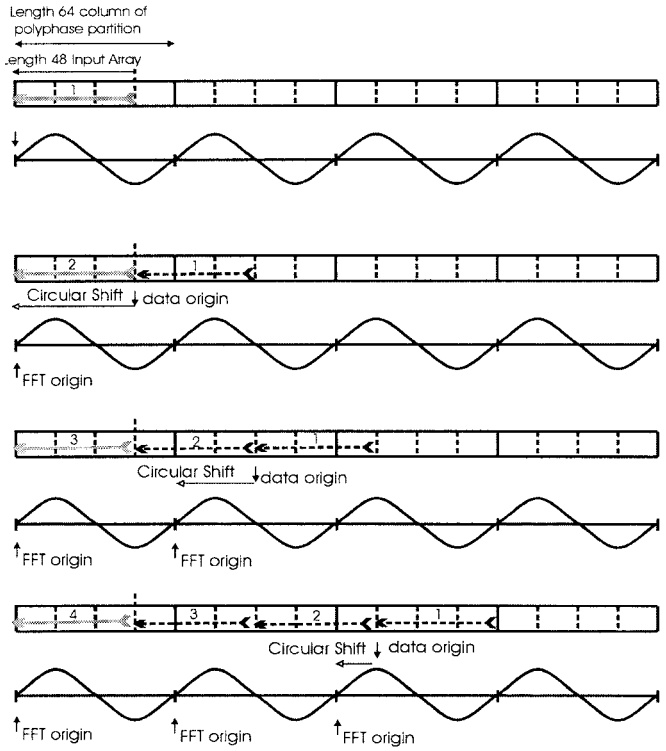

Fig. 32. Memory contents for successive 48-point input data blocks into a 64-point prototype pre-polyphase partitioned filter and FFT.

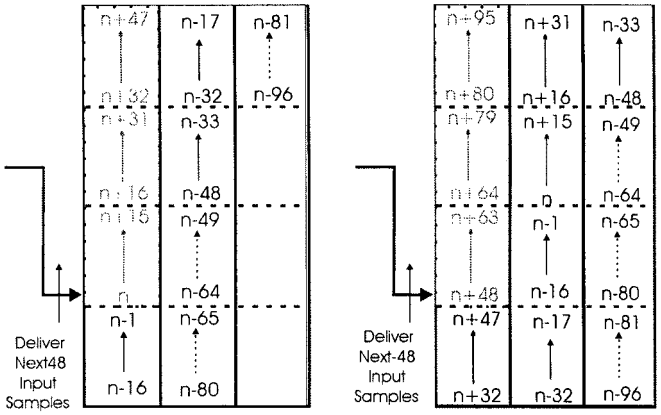

Fig. 33. Memory contents for successive 48-point input data blocks into a 64-point polyphase filter.

Returning to Fig. 32, the one-dimensional prototype, we note that every new data block shifts the input data origin to the right by 48 samples. The vector $\hat{y}(r, 48n)$ formed as the polyphase filter output from all 64 path filters is processed by the FFT to form the vector $\hat{Y}(k, 48n)$ of channelized (index k) output time series (index $48n$). On each successive call to the FFT, the origin

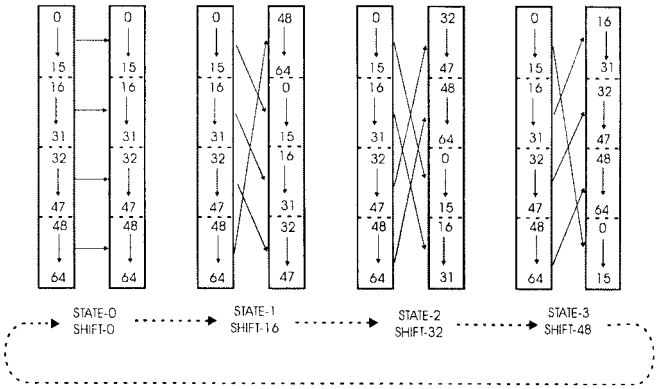

Fig. 34. Cyclic shift schedule for input array to FFT.

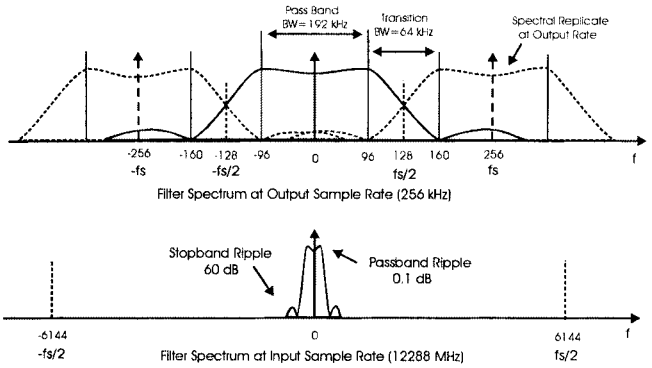

Fig. 35. Spectral characteristics of prototype filter at output and input sample rates.

of the sinusoids in the FFT is reset to the beginning of the input array. Since the origin of the input array shifts to the right on successive inputs while the origin of the FFT simultaneously resets to the beginning of the input array, a precessing offset exists between the origins of the polyphase filter and FFT. We align the origins, removing the offsets, by performing a circular shift of the vector $\hat{y}(r, 48n)$ prior to passing it to the FFT. Since the offset is periodic and is a known function of the input array index, the circular offset of the vector can be scheduled and controlled by a simple state machine. Fig. 32 shows the location of the two origins for four successive 48-point input arrays and the amount of circular offset required to align the two prior to the FFT. Note that the offset schedule repeats in four cycles, four being the number of input intervals of length 48 that is a multiple of 64.

The cyclic shift for schedule for the array $\hat{y}(r, 48n)$ prior to the FFT is shown in Fig. 34, Appendix II in the electronic version of this paper contain a MATLAB m-file that implements the 50-stage polyphase channelizer described in this section. The program plots the impulse response of the prototype low-pass filter and the frequency response at the input and output sample rates. It then plots the spectrum and the input signal formed as a sum of sinusoids distributed over the span of frequencies matching the bandwidth of the channelizer. Finally, a series of plots are generated showing the time series from 60 of the channelizer outputs. Here, we observe the transient of the filtering process in the occupied and unoccupied channels, as well as the effect of filtering and aliasing spectral translation of the occupied channels. The code is well annotated and the reader is invited to modify the input time series and probe the performance of the channelizer with a variety of test signals.

VII. Polyphase Computational Complexity

This section compares the computational workload required to implement a channelizer as a bank of conventional down converters with that required to implement the polyphase resampling approach. Here, we call on the example of the 50-channel channelizer to supply actual numbers. We first determine the length of the finite impulse response (FIR) prototype filter required to satisfy the filter specifications. We note that the filter designed to operate at its input rate (12.288 MHz) has its specifications controlled by its output rate (256 kHz). This is because the filter must satisfy the Nyquist sampling criterion after spectral folding as a result of the down-sample operation. The length of any FIR filter is controlled by the ratio of input sample rate to filter transition bandwidth and the required out-of band attenuation, as well as level of in-band ripple. The specifications of the filter used in this paper are listed in Fig. 35.

Standard design rules determine the filter length from the filter specification and for those indicated in Fig. 35, the filter length was found to be 512 samples and the Remez algorithm [23],[1] was used to supply the filter impulse response. A side comment is called for here. Filters designed by the standard Remez algorithm exhibit constant level out-of-band sidelobe levels. The spectra corresponding to these sidelobes folds back into the filter passband under the resampling operation resulting in integrated sidelobe levels considerably above the designed attenuation level. Modifying the penalty function of the Remez algorithm so that the sidelobes fall off at approximately 6-dB per octave reduces the level of integrated side lobes. See, for instance, the filter design package QED-2000 available from Momentum Data Systems. An alternate scheme to control spectral sidelobe decay rate involves modifying the end points of the filter impulse response to suppress the outlier samples responsible for the constant level spectral side lobes. See, for instance, the MATLAB examples in the appendixes of the electronic version of this paper.

An important consideration and perspective for filters that have different input and output sample rates is the ratio of filter length (with units of operations/output) to resample ratio (with units of inputs/output) to obtain the filter workload (with units of operations/input). A useful comparison of two processes is the number of multiplies and adds per input point. We count a multiply and add with their requisite data and coefficient fetch cycles as a single processor operation and use the shorthand notation of "ops" per input.

A single channel of a standard down-converter channelizer requires one complex multiply per input point for the input heterodyne and computes one complex output from the pair of 512 tap filters after collecting 48 inputs from the heterodyne. The four real ops per input for the mixer and the two $(512/48) = 22$ ops per input for the filter result in a per channel workload of 26 ops per input, which occur at the input sample rate.

The polyphase version of the down converter collects 48 input samples from the input commutator, performs 1024 ops in the

[1]MATLAB help menu for remez.m

pair of 512 tap filters, and then performs a 64-point FFT with its upper bound workload of $2N \log_2(N)$ real ops. The total workload of 1024 ops for the filter and 768 ops for the FFT results in 1792 ops performed once per 48 inputs for an input workload of 38 real ops/input. The higher workload per input is the consequence of forming 64 output channels in the FFT, but preserving only 50 of them.

Now we must be careful: the workload per input sample for the standard channelizer was found to be 26 ops, and for the polyphase channelizer was found to be 38 ops: where is the promised advantage? The advantage is that the polyphase 38 ops per input built all 50 channels, and the standard down converter's 26 ops per input built only one channel and has to be repeated 50 times. Isn't that impressive? Comparing numbers, we see that we should use the polyphase form even if we are forming just a few output channels because the polyphase down converter requires less computations than even two standard down converters.

On a final note, when we compare hardware resources, we observe that the standard channelizer must build and apply 50 complex sinusoids as input heterodynes to the input data at the high input sample rate and further must store the 50 sets of down converted data for the filtering operations. On the other hand, the polyphase filter bank only stores one set of input data because the complex phase rotators are applied after the filter rather than before and the phase rotators are applied at the filter output rate, as opposed to the filter input rate.

VIII. Applications

The polyphase filter structure we have just reviewed and demonstrated can absorb a number of different system specifications. We now discuss some of the options. When the FDM signal is a collection of independent signals, as might occur in a multiple access application, the many signals do not share a common time reference or a common carrier frequency reference. The channels likely differ in small carrier and timing offsets that must be resolved in subsequent modem processing. For this situation it is appropriate for the channelizer to only perform the standard down conversion, bandwidth reduction, and sample rate reduction. For this application, the filter passes the selected channel bandwidths without further spectral modification. This form of multiple channelization is also appropriate for analog modulation such as the 30-kHz narrow-band FM channels found in analog wireless advanced mobile phone service (AMPS), or the old standby SSB super group.

In another application, the multiple channels are tightly coupled with negligible frequency offset and only differ in local time base for the modulation. For this case, we can require the polyphase filter bank to perform the functions of the matched filter. The filter passes the selected channel bandwidths while performing the spectral shaping required of the matched filter. Since the data is formed at two samples per symbol, subsequent processing only has to align the phase of each channel and interpolate to the correct timing phase of the complex envelope. We note that the timing interpolation process requires significantly fewer resources than does the simultaneous interpolation matched filter and timing task.

In yet a third application, the multiple channels share a common clock and carrier reference and, hence, are fully synchronous and exhibit negligible frequency offset. For this case, we can ask the polyphase filter bank to perform the functions of the matched filter and to participate in timing recovery [24], [25]. Since, in the example cited, we are down sampling by a factor of 48-to-1, there are 48 different contenders for the arbitrary origin of the process task. We can have the timing recovery loop advance or retard the start of the vector load of the polyphase filter. Using the processing origin to shift the sample time relative to the underlying modulation epochs offers fine grain time offsets equal to 1% of symbol interval without the need for additional interpolation.

The example we used to demonstrate the polyphase resampling channelizer implemented a 1/48 resampling in a 64-stage channelizer. Any ratio of small integers can be implemented using variations of the technique we have presented. The number of polyphase filter stages in a multichannel receiver also does not have to be large to warrant the application of the process described here. For instance, we have designed a number of polyphase resampling structures for 3G [26] wireless applications that employ from 3 to 11 channels, using 5- and 15-point transforms. These applications have required sample rate changes from 3.84 to 6.144 MHz and to 15.36 MHz requiring ratios of 3-to-5 and 2-to-5.

In a fashion similar to the process we have presented here to design down-sampling polyphase receiver channelizers, we can also design polyphase up-sampling channelized transmitters. In fact, when we design polyphase receivers, we are often obliged to design polyphase transmitters to test the receivers. An unexpected result discovered when we designed the transmitters for a given receiver is that they are not each other's duals because they are, in fact, not performing the inverse operators. For instance, in the receiver example we developed here, we formed a 48-to-1 down sampler in a 64-point DFT, which is a ratio of 3/4. The modulator on the other hand forms channels at a 128-kHz symbol rate at a common sample rate of 64 times 192 kHz, an up sample of 96 in the 64-point DFT, which is a ratio of 2/3.

Finally, we note that there are filter structures [27] that permit the substitution of recursive polyphase filters for the nonrecursive filter we have examined in this paper. The recursive filter options offer a reduction in workload by a factor of 3–6 and are available with both nonuniform phases and an equal-ripple approximation to linear phase. A minor drawback here is that the recursion in the filter prohibits computation pipeline delay, which limits the maximum output sample rate to the range of 200–400 MHz.

IX. Conclusions

We have presented a description of the process by which a multichannel polyphase filter bank can simultaneously perform the uncoupled tasks of down conversion, bandwidth limiting, and sample rate change. We included a tutorial derivation of the polyphase filter bank as a sequence of transformations that rearrange the operations of mixing, filtering, and resampling to obtain remarkably efficient processing structures. The sequence of transformations included application of the equivalency theorem, alias-based spectral translation, sometimes referred to as

IF sampling, and of the noble identity. We also demonstrated that the ratio of input sample rate to output sample rate could differ from the conventional resampling ratio, the number of stages in the polyphase partition. The modification to the conventional polyphase channelizer required the insertion of circular buffers between the input commutator and polyphase filter and the insertion of a second circular buffer between the output of the polyphase filter and the FFT phase rotator. A number of excellent tutorials [1], [28]–[31] are available in the literature that present aspects of some of the material presented here from a number of different perspectives. Readers may find value in examining other author's perspectives after being exposed to ours. The reader may also benefit from the background and tutorial material found in [1] and [2]

We examined a specific example of a polyphase resampling channelizer to better illustrate the processes required to obtain arbitrary resampling in the filter bank. Comments on variations to the modified polyphase structure were included to give the reader a sense of the wide range of applicability of this process. Finally, we compared the workload of a standard mixer-based down-converter filter bank with that of the polyphase resampling form. We invite readers to e-mail requests to the author for the MATLAB code that implements the 10- and 50-channel channelizer described in this paper. The electronic version of this paper has the MATLAB files attached as appendixes. A MATLAB script for an animated version of the ten-channel channelizer is also included in the appendexes as is a 40-channel modulator that performs a 1-to-56 up sampling with a companion 40-channel demodulator that performs a 28-to-1 down sampling embedded in the channelization processes.

REFERENCES

[1] N. J. Fliege, "Polyphase FFT filter bank for QAM data transmission," in *Proc. IEEE ISCAS'90*, pp. 654–657.

[2] R. E. Chrochier and L. R. Rabiner, *Multirate Digital Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1981.

[3] F. J. Harris, "On the relationship between multirate polyphase FIR filters and windowed, overlapped FFT processing," presented at the 23rd Annu. Asilomar Conf. on Signals and Computers, 1989.

[4] G. Moore. (1965, Apr.) Cramming more components onto integrated circuits. *Electronics* [Online] Available: http://www.intel.com/research/silicon/mooreslaw.htm

[5] C. Dick, "The platform FPGA: Enabling the software radio," presented at the Software Defined Radio Technical Conf., San Diego, CA, Nov. 11–12, 2002.

[6] A. M. Badda and M. Donati, "The software defined radio technique applied to the RF front-end for cellular mobile systems," in *Software Radio Technologies and Services*, E. Del Re, Ed. Berlin, Germany: Springer-Verlog, 2001.

[7] F. J. Harris, "On measuring the gain and phase imbalance and DC offsets of quadrature A-to-D converters with an adaptive canceling filter," presented at the 21st Annu. Asilomar Conf. on Signals, Systems, and Computers, Pacific Grove, CA, Nov. 2–4, 1987.

[8] B. Sklar, *Digital Communications: Fundamentals and Applications*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 2002, sec. 12.2.

[9] J. G. Proakis and D. G. Manolokis, *Digital Signal Processing: Principles, Algorithms, and Applications*, 3rd ed. Upper Saddle River, NY: Prentice-Hall, 1996, sec. 1.4.

[10] D. Steinbrecher, "Broadband high dynamic range A/D conversion limitations," presented at the Analogue to Digital and Digital to Analogue Conversion Int. Conf., Swansea, U.K., Sept. 17–19, 1991.

[11] (Feb. 2001 and Jan. 2001) Defining and testing dynamic parameters in high speed ADC: Part I and Part II. *MAXIM Application Notes* [Online]Available: http://www.maxim-ic.com/appnotes.cfm/appnote_number/728; http://www.maxim-ic.com/appnotes.cfm/appnote_number/729

[12] M. Ribey, "Exploration of transmultiplexers in telecommunication networks," *IEEE Trans. Commun.*, vol. COM-30, pp. 1493–1497, July 1982.

[13] M. Bellanger and J. Daguet, "TDM-FDM transmultiplxer: Digital polyphase and FFT," *IEEE Trans. Commun.*, vol. COM-22, pp. 1199–1204, Sept. 1974.

[14] F. J. Harris, "A fresh view of digital signal processing for software defined radios: Part I," presented at the Int. Telemetry Conf., San Diego, CA, Oct. 21–24, 2002.

[15] ——, "A fresh view of digital signal processing for software defined radios, Part II," presented at the Int. Telemetry Conf., San Diego, CA, Oct. 21–24, 2002.

[16] J. Wozencraft and I. M. Jacobs, *Principles of Communication Engineering*. New York: Wiley, 1967, sec. 7.2.

[17] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[18] S. K. Mitra, *Digital Signal Processing: A Computer Based Approach*, 2nd ed. New York: McGraw-Hill, 2001.

[19] J. L. Butler, *Microwave Scanning Antennas*, R. M. Foster, Ed. New York: Academic, 1964, vol. III, ch. 3.

[20] B. Pattan, *Robust Modulation Methods & Smart Antennas in Wireless Communications*. Englewood Cliffs, NJ: Prentice-Hall, 2000, ch. 9.

[21] F. J. Harris and C. Dick, "Performing simultaneous arbitrary spectral translation and sample rate change in polyphase interpolating and decimating filters in transmitters and receivers," presented at the Software Defined Radio Technical Conf., San Diego, CA, Nov. 11–12, 2002.

[22] D. Elliot, Ed., "Handbook of digital signal processing: Engineering applications," in *Time Domain Signal Processing With the DFT*. New York: Academic, 1987, ch. 8, pp. 639–666.

[23] J. H. McClellan, T. W. Parks, and L. B. Rabiner, "A computer program for designing optimum FIR linear phase digital filters," *IEEE Trans. Audio Electroacoust.*, vol. AU-21, no. 6, pp. 506–526, Dec. 1973.

[24] F. J. Harris and M. Rice, "Multirate digital filters for symbol timing synchronization in software defined radios," *IEEE J. Select. Areas Commun.*, vol. 19, pp. 2346–2357, Dec. 2001.

[25] M. Rice and F. J. Harris, "Polyphase filter banks for symbol synchronization in sampled data receivers," presented at the MILCOM, Anaheim, CA, Oct. 7–10, 2002.

[26] D. Metri, "Reconfigurable receivers for 3-G applications," M.S. thesis, Dept. Elect. Comput. Eng., San Diego State Univ., San Diego, CA, 2002.

[27] M. Renfors and T. Saramaki, "Recursive Nth band digital filters: Parts I an II," *IEEE Trans. Circuits Syst.*, vol. CAS-34, pp. 24–51, Jan. 1987.

[28] P. P. Vaidyanathan, "Multirate digital filters, filter banks, polyphase networks and applications: A Tutorial," *Proc. IEEE*, vol. 78, pp. 56–93, Jan. 1990.

[29] R. D. Koilpillai, T. Q. Nguyen, and P. P. Vaidyanathan, "Some results in the theory of crosstalk free transmultiplexer," *IEEE Trans. Signal Processing*, vol. 39, pp. 2174–2183, Oct. 1991.

[30] H. Scheuermann and H. Göckler, "A comprehensive survey of digital transmultiplexing methods," *Proc. IEEE*, vol. 69, pp. 1419–1450, Nov. 1981.

[31] K. C. Zangi and R. D. Koilpillai, "Software radio issues in cellular base stations," *IEEE J. Select. Areas Commun.*, vol. 17, pp. 561–573, Apr. 1999.

Fredric J. Harris (S'67–M'70–SM'78–F'03) received the B.S. degree from the Polytechnic Institute of Brooklyn, Brooklyn, NY, in 1961, the M.S. degree from San Diego State University, San Diego, CA, in 1967, and the course work required for the Ph.D. degree from the University of California at San Diego, La Jolla, in 1975, all in electrical engineering.

He holds the CUBIC Signal Processing Chair of the Communication Systems and Signal Processing Institute, San Diego State University, where he has taught since 1967 in areas related to DSP and communication systems. He holds a number of patents on digital receivers and related DSP technology. He lectures throughout the world on DSP applications to communication systems, as well as consults for organizations requiring high-performance DSP systems. He has authored and coauthored numerous papers and has contributed to a number of books on DSP and communication systems.

Mr. Harris was the technical chair and then general chair of the 1990 and 1991 Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, CA.

Chris Dick (M'89) received the B.S. and Ph.D. degrees in computer science and electronic engineering from Latrobe University, Melbourne, Australia, in 1986 and 1996, respectively.

In 1997, he joined Xilinx, San Jose, CA, where he is currently the Director of Signal Processing Engineering and the DSP Chief Architect. Prior to joining Xilinx, he was with La Trobe University, Melbourne Australia, where he was a Professor for 13 years and consulted for industry and the military sector. He has authored or coauthored over 70 journal and conference publications and has been an invited speaker at many international DSP and communications symposia. He has also authored or coauthored many papers in the fields of signal processing, parallel computing, inverse synthetic array radar (ISAR) imaging, and the use of FPGAs for building computing platforms for signal processing applications. His research interests are in the areas of fast algorithms for signal processing, digital communication, software-defined radios, hardware architectures for real-time signal processing, parallel computing, interconnection networks for parallel processors, and the use of FPGAs for custom computing machines and signal processing.

Michael Rice (M'82–SM'98) received the B.S.E.E. degree from Louisiana Tech. University, Ruston, in 1987, and the Ph.D. degree from the Georgia Institute of Technology, Atlanta, in 1991.

He was with Digital Transmission Systems Inc., Atlanta, GA, until joining the faculty at Brigham Young University, Provo, UT, in 1991, where he is currently an Associate Professor in the Department of Electrical and Computer Engineering. He was a NASA/ASEE Summer Faculty Fellow with the Jet Propulsion Laboratory from 1994 to 1995, during which time he was involved with land mobile satellite systems. During the 1999–2000 academic year, he was a Visiting Scholar with the Communication Systems and Signal Processing Institute, San Diego State University. His research interests include digital communication theory and error-control coding with a special interest in applications to telemetering and software radio design. He has been a consultant to both government and industry on telemetry-related issues.

Dr. Rice was chair of the Utah Section of the IEEE from 1997 to 1999. He is currently chair of the Signal Processing and Communications Society Chapter of the Utah Section and the Region-6 Student Activities coordinator.