Chapter 2

Continuous Wavelet Transform

2.1 Introduction to the 1-Dimensional Continuous Wavelet Transform

Wavelet transforms are time-frequency transforms which map the time-frequency plane in the manner of Figure 1.1 (c) using specific dilations and translations. Tilings of higher-frequency areas of the plane have larger bandwidth and thus, in accordance with the uncertainty principle, shorter timespan. The bandwidth is proportional to the position of the tile along the frequency axis. The relationship between frequency position and bandwidth (and inversely, timespan) is in keeping with the Nyquist sampling theory which states that in order to accurately capture the information about a signal (so as to be able, for instance, to reconstruct that signal), the signal must be sampled at a twice the value of the highest frequency in the signal. In other words, the timespan covered by the samples must be inversely proportional to the bandwidth of the signal. This makes wavelet tiling more "natural" in a sense, and while it cannot guarantee the most efficient representation of any arbitrary signal, it will tend to be a good representation for most natural signals.

It is important to explain what I mean by a "good representation." A voice-band recording sampled at 8 kHz can be called a "good representation" in the sense that it contains all the information about the signal necessary to reconstruct (*e.g.*, play back) the recording. On the other hand, a sampled signal (without any further transformations) is rarely used for speech recognition because the speech signal is encoded into the time-domain signal in complex ways that cannot be detected by a purely time-domain analysis (the Dirac basis). Equivalently, a Fourier transform (Fourier basis) of the signal is of limited use. A windowed Fourier transform does fairly well at capturing the essence of the speech recording, but if the window is of fixed length, then there will always be some events which are too short in time to be accurately represented. Two solutions to that problem

may come to mind: Either vary the size of the window according to the demands of the moment, or perform the analysis in parallel with windows of several different lengths. The first solution is adaptive, and while it is a very interesting solution, it is difficult to compute and requires information about the adaptation to be kept with the decomposed signal if the signal is to be analyzed or reconstructed. Advanced adaptive techniques are beyond the scope of this thesis. The other solution, parallel analysis at different scales, works well, but the result is redundantly computed, which increases the bandwidth of the decomposed signal n-fold for n parallel analyses. If we remove the redundant parts of the decomposition, the part that is left is a wavelet decomposition. Parts of the decomposition capture features of the signal which occur over short time (*transients*) and other parts capture features of the signal which have fine frequency structure (*formants*). Natural signals, particularly acoustic ones which are the primary topic of the thesis, require the accurate detection of both transient-like and formant-like features for automatic classification and interpretation of the signal.

Each tile of the time-frequency plane represents a single "wavelet coefficient" computed by applying a filtering function centered on that area and having the correct aspect ratio between time span and bandwidth. The so-called "mother function" describes a family of functions at different scales (a) and temporal offsets (b) which determine the position and aspect ratio of each tile covering the time-frequency plane:

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right). \tag{2.1}$$

It is desirable for the wavelet function to have *compact support*; that is, the function should be bounded or generally localized in time and frequency. In formal terms, the wavelet should be able to meet the criteria [14]

$$|\psi(t)| \leq c\,(1+|t|)^{-1-\epsilon} \tag{2.2}$$

$$|\Psi(\omega)| \leq c\,(1+|\omega|)^{-1-\epsilon} \tag{2.3}$$

for some $\epsilon > 0$. Existence of an inverse transform depends on the relationship

$$\int_{-\infty}^{\infty} \frac{|\psi(\omega)|^2}{|\omega|} d\omega < +\infty. \tag{2.4}$$

Equation (2.4) implies that $\Psi(\omega = 0) = 0$, which in turn implies an oscillatory function in time.

The Discrete Wavelet Transform (DWT) is a discrete-time function which derives from certain families of orthonormal basis functions which satisfy the conditions of (2.3) and (2.4). The

wavelet mother function is bound is space by satisfying the stronger relationship $\psi(t) = 0$ outside of a small region of compact support. Bounding the frequency is a matter of filtering. Because the system is discrete-time, the filtering functions are FIR filters and the bandpass characteristic results from sequential application of a lowpass and a highpass filter [10], a common practice in signal processing. An elegant "pyramid" algorithm [11] defines filtering over different frequency scales in a recursive manner in a way that allows information from higher frequencies to be pushed into lower frequencies as the recursion progresses.

Using well-developed discrete-time filter theory, the FIR highpass and lowpass filters can be transformed into an efficient pipelined butterfly filter (also known as a lattice filter) [12]. The FIR filter coefficients or the equivalent lattice filter coefficients are uniquely determined by the mother wavelet basis function. The length of the FIR or butterfly filter is the wavelet order. Higher-order wavelet functions typically yield better distribution of information among the wavelet dilations, leading to better data compression when the wavelet system is used for that purpose.

2.2 CWT *vs.* DWT

The Continuous Wavelet Transform (CWT) is an analog filtering function and is similar to what is known as the Gabor spectrogram [13]. Similarly to the Discrete Wavelet Transform, it requires operations of lowpass and highpass filtering at different scales. However, the filtering functions are performed on the input in parallel, as a filterbank, with the lowpass and highpass functions combined into a single bandpass function. The different scales are interpreted as adjacent bands in the frequency domain, with the bandwidth increasing proportionally with the center frequency of the band: thus the CWT is described by a constant-Q filterbank. That description is shown graphically in Fig. 2.1.

The DWT produces the equivalent result by starting with a block of discrete data and performing successive high- and lowpass digital filtering. The filtering is repeated on the lowpass output in order to bandpass the signal in a series of stages called "dilations." Each dilation divides the frequency space of the current interval in half while doubling the time span, thus keeping the time-frequency product constant. In contrast, the CWT divides a signal into a set of logarithmically-scaled frequency bands by passing it through a bank of constant-Q bandpass filters. Both the CWT and the DWT are filterbanks, although the CWT takes the more intuitive form of a physical filter with a transfer function in the frequency and time domains. DWT filters are carefully formulated mathematical constructs whose transfer functions are recursive and which can only be said to per-

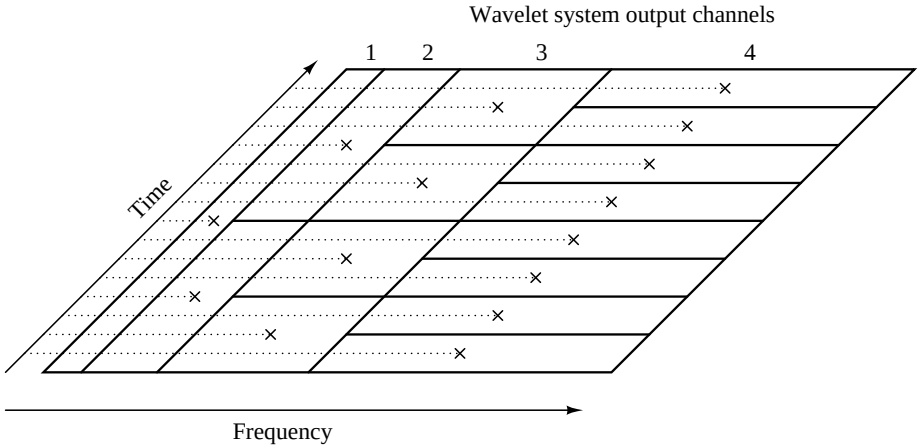

Figure 2.1: Output sampling. Points marked '$\times$' represent center of the time-frequency area covered by that sampled output.

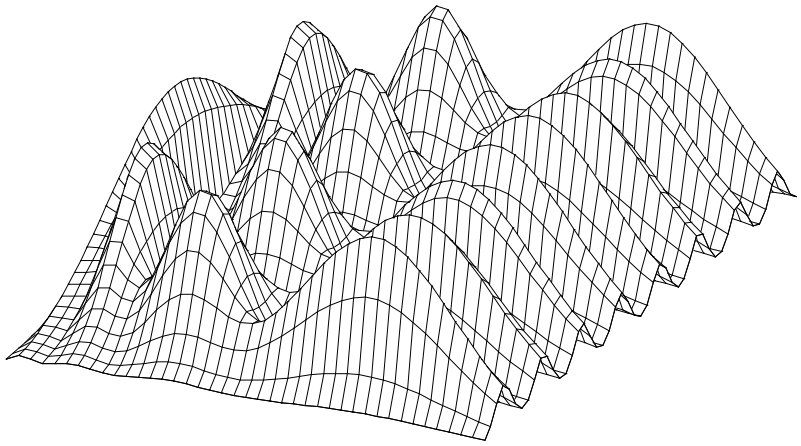

Figure 2.2: Frequency-time representation of the input as overlapping Gaussian filters.

form “lowpass” and “highpass” functions in a vaguely-defined sense. Due to the constraints of the Discrete Wavelet Transform specification, all DWT filters have compact support. The fact that the DWT can be confined absolutely in time and frequency is proof that it cannot be implemented by physical filters in a continuous time domain. By being physically realizable, the CWT cannot achieve perfect compact support. Instead, analogously to the way elliptical filters such as the Chebyshev and Butterworth are generated, CWT functions are constructed in order to achieve maximal compactness with respect to some criterion.

A major difference between the DWT and the CWT is that the CWT loses the concept of *compact support*. The continuous-time nature of the transform implies the use of causal filters, whose frequency response cannot be perfectly limited to a given bandwidth. Instead, the filters must overlap. The shape and complexity of the filter determines the amount of overlap and thus higher-order filter functions can be used to describe higher-order continuous wavelet transforms, analogously to the way that higher-order functions operate in the discrete wavelet transform.

2.3 Gabor Logons and Wavelets

The beginnings of the theory of the continuous wavelet transform begin with a seminal work by Gabor [8], a paper with the rather bold title “Theory of Communication” (1946) which outlines the physical basis behind limitations of time-domain and Fourier analysis and shows how both time and frequency can be incorporated into a function providing the optimal tradeoff in resolution between the two domains, and how the time-frequency plane can be effectively represented by tiling with these functions. Gabor calls the functions “logons,” those functions which have minimum $\Delta\omega\Delta t$. The simplest logon form (lowest order CWT filter) is the Gaussian function,

$$\Psi(\omega) = e^{-\omega^2/2\sigma^2}. \tag{2.5}$$

The laws of Fourier transforms dictate that a Gaussian in the time domain is also a Gaussian in the frequency domain, thus this function is smooth in all directions on the time-frequency plane. The properties of Fourier transforms also dictate a symmetry between time and frequency shifts of a signal and multiplication by a complex sinusoid in frequency and time, respectively:

$$\Psi(\omega) = e^{-j\omega t_a}\, e^{-(\omega-\omega_b)^2/2\eta\sigma_b^2} \tag{2.6}$$

$$\psi(t) = e^{-j\omega_a t}\, e^{-(t-t_b)^2/2\sigma_b^2} \tag{2.7}$$

The complex exponential in either transform can be split into real and imaginary parts, in which case the function appears as a sinusoid modulated by a Gaussian envelope, as shown in Figure 2.3. A family of curves can be generated on this basis, using the Gaussian as an envelope around a sinusoidal signal of differing frequencies. The meaning of these curves should be intuitively clear: The more cycles of a sinusoid fit into the Gaussian envelope, the better the frequency is defined, but the poorer the time is defined. A pure sinusoid of infinite duration represents one extreme, for which frequency is known exactly but time is not known at all. Likewise, the Gaussian itself represents the other extreme: the cosine function for which time is known exactly but for which frequency is entirely unknown, having no reference signal to which it can be compared. The "wavelet tiling" of the time-frequency plane (Figure 1.1 (c)) dictates the ratio of ω_a, the modulating frequency, to σ_b, the width of the Gaussian envelope. As one gets bigger, the other gets smaller.

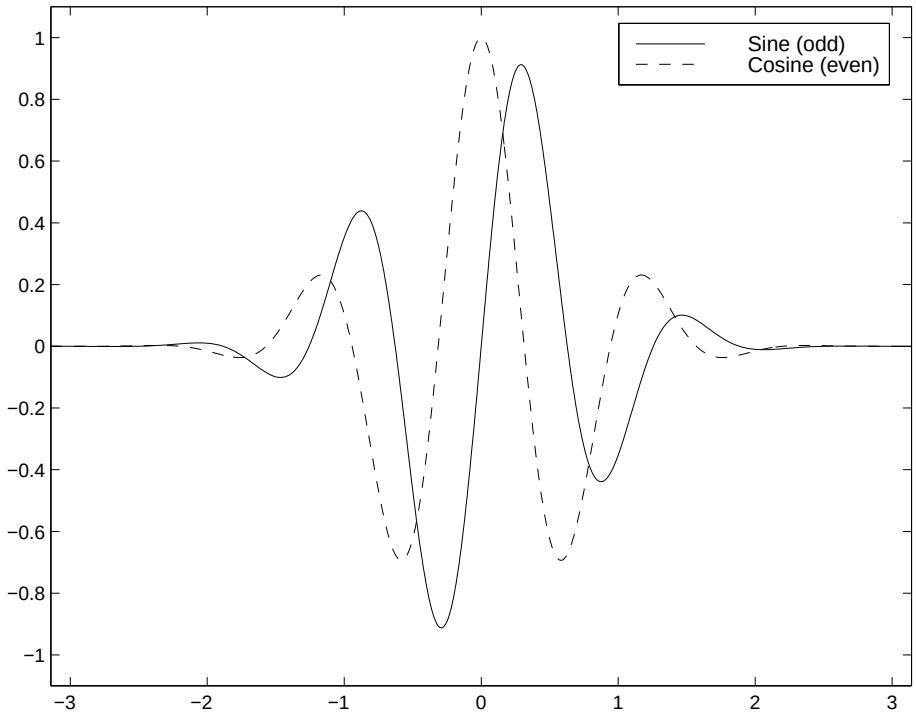

Figure 2.3: Gabor sine and cosine logons, or wavelets.

The Fourier transform property of time and frequency shifts yields an interesting insight about the connection to dilations and shifts of the discrete wavelet transform, since they are manifestations of the Fourier and inverse Fourier transforms. Unlike the discrete wavelet transform families, the Fourier transform pair of the Gabor logon family is beautifully symmetric.

Other useful continuous wavelet filter functions can be derived starting from different criteria for compactness. One such case is the chirplet [16], which allows rotation of the filter with respect to the time and frequency axes.

The frequency shift of the Gabor logon, which is a multiplication by a complex sinusoid in the time domain, amounts to a signal demodulation, where in order to maintain the symmetry between the time and frequency sides of the transform, the complex nature of the frequency shift must be maintained; that is, the modulation must be performed with both a sine and a cosine to represent the real and imaginary parts of the multiplication with the complex exponential.

2.4 Complex Demodulation

Continuous-time bandpass filter functions with arbitrarily high Q values are notoriously difficult to design with precision (a fact which will be discussed in detail in Chapter 3, Section 3.5). It is not especially difficult to build a second-order filter section which can be tuned externally to the desired specifications, but managing to get an entire filterbank of 16 or 32 independent sections to all match within a given tolerance can be difficult or impossible, depending on the strictness of the specifications. Adding a requirement of low power consumption compounds the problem. Generally, the only practical circuit solution is to leave the continuous-time domain and instead enter the discrete-time domain, with the use of switched capacitor (S-C) circuits. The result is accurate (for analog) computation with modest power consumption at the expense of die area, which tends to be high due to the use of large numbers of capacitors.

We did, in fact, resort to switched capacitor architectures for all of our wavelet filter functions. However, it was not necessary to stop optimizing at the architectural (circuit) level. On the algorithmic level, we were further able to make better bandpass filters using a method called *complex demodulation* [9]. Demodulation is a well-established method used widely in modems and radios, based on the principle that the frequency content of a signal can be shifted up or down by multiplying it by a pure sinusoidal "carrier" signal and then filtering appropriately. In those applications, the signal to be broadcast is first modulated on a carrier signal to push it into a high-frequency broadcast band, and then demodulated by the receiver to retrieve the original signal. The purpose of modulating is twofold: First, the high-frequency signal can be broadcast much further without significant attenuation, and second, a large number of signals can be transmitted at the same time by assigning each one a non-overlapping portion of the electromagnetic spectrum.

Demodulation of a signal by a carrier

An input time-varying signal $x(t)$ is multiplied by a sinusoidal reference signal $s(t) = \cos(\omega_0 t)$. As viewed in the frequency domain, the result splits the input around the reference to generate the sum (reference + input) and difference (reference − input) components. This result can be easily seen by considering a input consisting of a single sinusoidal function of arbitrary phase relationship to the reference, for instance $x(t) = \cos(\omega t + \phi)$. The modulation is then

$$x(t)s(t) = \cos(\omega t + \phi)\cos(\omega_0 t). \tag{2.8}$$

Now, adding the trigonometric identities

$$\cos(u+v) = \cos(u)\cos(v) - \sin(u)\sin(v) \tag{2.9}$$
$$\cos(u-v) = \cos(u)\cos(v) + \sin(u)\sin(v) \tag{2.10}$$

together gives

$$\cos(u+v) + \cos(u-v) = 2\cos(u)\cos(v) \tag{2.11}$$

such that, when used with Equation (2.8), we get

$$x(t)s(t) = \frac{1}{2}\left(\cos((\omega_0 + \omega)t + \phi) + \cos((\omega_0 - \omega)t - \phi)\right) \tag{2.12}$$

which contains one sinusoid describing the sum $\omega_0 + \omega$ of the carrier and signal frequencies, and one describing the difference $\omega_0 - \omega$. Extending the result to arbitrary inputs involves viewing the arbitrary input as a Fourier series of sine components; since the modulation multiplication is a linear function, superposition applies, and every Fourier component of the arbitrary input is split into sum and difference components across the carrier frequency. Filtering the resultant signal with a highpass filter which accepts the sum component but rejects the difference component is the process known as *modulation*. Filtering the same signal instead with a lowpass filter which accepts the difference component but rejects the sum component is the process known as *demodulation*.

A useful application of the principle of signal modulation involves performing the demodulation *first*. In such case, the signal to be encoded is multiplied by an in-band carrier frequency in order to shift the desired frequency band down to zero. The resulting signal is lowpass filtered to remove the component representing the sum of carrier and modulator. If desired, the signal can then be modulated back into its original band, at which point the result is a bandpassed signal. Not only is it bandlimited, but the band to which it is limited can be made arbitrarily small, since there is no limit (other than the practical limits of noise, jitter of the carrier frequency, *etc*.) to the cutoff frequency of the lowpass function. The tight band limit translates to an arbitrarily large effective value of Q. The position of the band is placed at the carrier frequency, so it can be made very accurate (particularly since in our architecture the carrier frequency is driven by a quartz oscillator).

The method just described is also a well-established method, used in subband coders and to make "lock-in" amplifiers able to analyze miniscule frequency bands for signal and noise analysis.

There is one small mathematical catch to this method. When a signal is demodulated to zero frequency, parts of the signal extend into the negative frequency domain, which in terms of the actual measured signal means that these frequencies are folded back (aliased) into the positive frequency domain. There they would be inseparable except for the "complex" part of complex demodulation. The method requires two carrier signals, one of which is offset from the other by 90 degrees phase. Thus if one carrier describes sin(ωt), then the other describes cos(ωt). Two demodulations are performed in parallel, each with one of the two carriers. The two resulting signals both have frequencies aliased into apparent mush, but between the two signals is all the information necessary to separate out the aliased parts after subsequent modulation. In fact, it can be shown that this requires nothing more than separately modulating the two results, again with the modulation carrier signals offset in phase, and adding the two modulation results together. No filtering is necessary for the modulation step, which is another consequence of the math.

Complex Demodulation and Reconstruction of a signal by a carrier

Returning to our previous example: An input time-varying signal $x(t)$ is multiplied by two sinusoidal reference signals $s_1(t) = \cos(\omega_0 t)$ and $s_2(t) = \sin(\omega_0 t)$. Choosing for $x(t)$ the simple form of a sinusoid of arbitrary phase ϕ:

$$x(t)s_1(t) = \cos(\omega t + \phi)\cos(\omega_0 t) \tag{2.13}$$

$$x(t)s_2(t) = \cos(\omega t + \phi)\sin(\omega_0 t) \tag{2.14}$$

Applying the following trigonometric identities—

$$\cos(u+v) + \cos(u-v) = 2\cos(u)\cos(v) \tag{2.15}$$

$$\sin(u+v) + \sin(u-v) = 2\sin(u)\cos(v) \tag{2.16}$$

yields the following expressions:

$$x(t)s_1(t) = \frac{1}{2}\left(\cos((\omega_0+\omega)t+\phi) + \cos((\omega_0-\omega)t-\phi)\right) \tag{2.17}$$

$$x(t)s_2(t) = \frac{1}{2}\left(\sin((\omega_0+\omega)t+\phi) + \sin((\omega_0-\omega)t-\phi)\right). \tag{2.18}$$

Now perform a lowpass filter by assuming an ideal filtering function, $h(t)$, having a Fourier transform $H(\omega)$, which perfectly rejects the sum of frequencies while perfectly passing the difference of frequencies (see Figure 2.4):

$$h(t) * x(t)s_1(t) = \frac{1}{2}\cos((\omega_0-\omega)t-\phi) \tag{2.19}$$

$$h(t) * x(t)s_2(t) = \frac{1}{2}\sin((\omega_0-\omega)t-\phi). \tag{2.20}$$

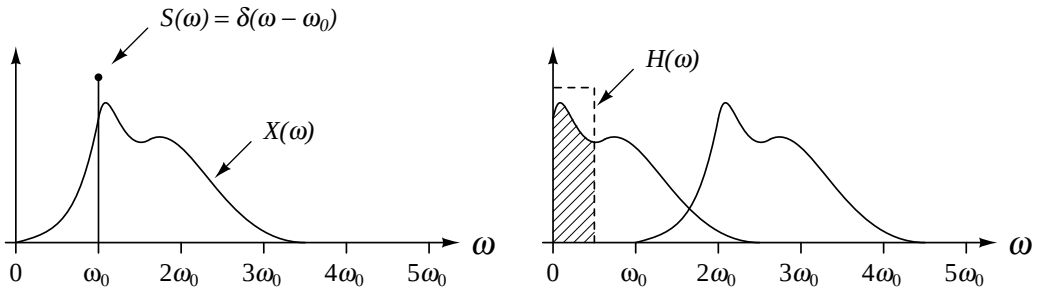

Figure 2.4: Demodulation of an input X in the frequency domain with a perfect sine wave S and ideal modulation filter H.

Reconstruction involves first multiplying each function by the same sine and cosine carriers:

$$h(t) * (x(t)s_1(t))s_1(t) = \cos(\omega_0 t)\frac{1}{2}\cos((\omega_0 - \omega)t - \phi) \quad (2.21)$$

$$h(t) * (x(t)s_2(t))s_2(t) = \sin(\omega_0 t)\frac{1}{2}\sin((\omega_0 - \omega)t - \phi). \quad (2.22)$$

Finally, without filtering, these two parts are added together to produce the reconstruction (in this case, exact reconstruction, due to the use of perfect filters):

$$x'(t) = h(t) * (x(t)s_1(t))s_1(t) + h(t) * (x(t)s_2(t))s_2(t) \quad (2.23)$$

$$= \cos(\omega_0 t)\frac{1}{2}\cos((\omega_0 - \omega)t - \phi) + \sin(\omega_0 t)\frac{1}{2}\sin((\omega_0 - \omega)t - \phi). \quad (2.24)$$

The trigonometric identity (2.10) applies directly, giving the final result:

$$x'(t) = \frac{1}{2}\cos((\omega_0 - \omega_0 + \omega)t + \phi) \quad (2.25)$$

$$= \frac{1}{2}\cos(\omega_0 t + \phi) \quad (2.26)$$

$$= \frac{1}{2}x(t) \quad (2.27)$$

showing that the reconstruction is exact except for the required application of a gain of two. Again, this example can be extended to arbitrary inputs by the application of Fourier series and superposition.

2.5 Complex Demodulation in the Continuous Wavelet Processor

The architecture used for complex demodulation in the wavelet processor is depicted in Figures 2.5 and 2.6 and described below.

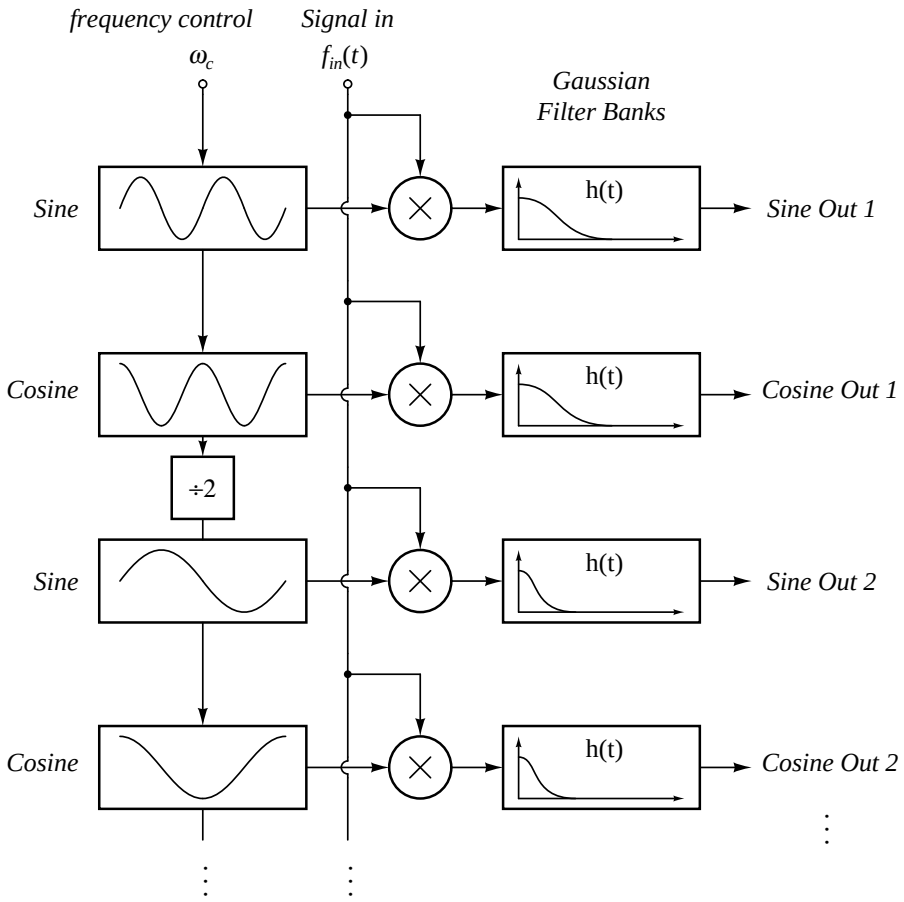

Figure 2.5: Complex demodulation (2 channels shown).

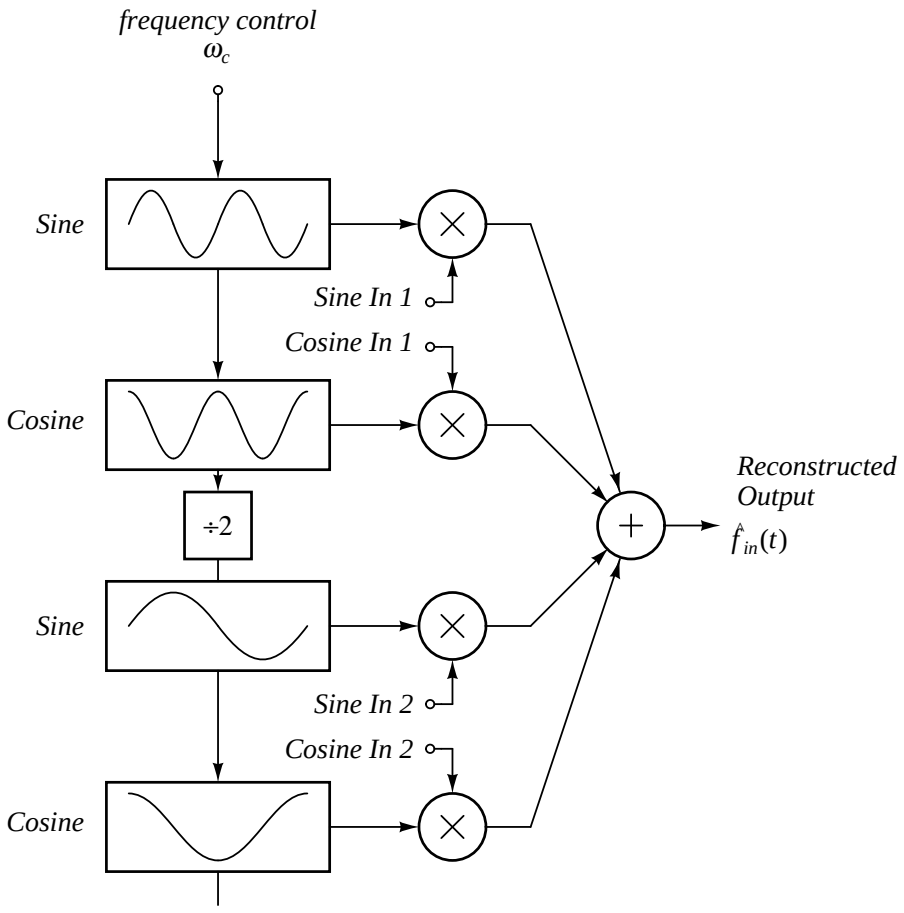

Figure 2.6: Complex modulation reconstruction (2 channels shown).

The input function is designated by $f_{\text{in}}(t)$. In order to demodulate it with respect to some given frequency ω_c (one of the center frequencies of the N channels of the wavelet decomposition), we use the following multiplication:

$$f_{\text{out}}(t) = 2h(t) * (f_{\text{in}}(t)(\cos\omega_c t + j\sin\omega_c t)) \tag{2.28}$$

where the lowpass filter time-domain transfer function $h(t)$ is shown without reference to a specific equation and is assumed for the sake of argument to be a filter which passes all frequencies below its cutoff unattenuated, and blocks all frequencies above its cutoff. Since the lowpass filter is assumed to be real, and the input $f_{\text{in}}(t)$ is real, then the output $f_{\text{out}}(t)$ of Equation (2.28) must necessarily be complex-valued, and can be represented by separating it into real and imaginary parts:

$$f_{\text{out}}(t) \equiv f_{\text{real}}(t) + j f_{\text{imag}}(t). \tag{2.29}$$

Therefore,

$$f_{\text{real}}(t) = 2h(t) * (f_{\text{in}}(t)\cos\omega_c t) \tag{2.30}$$

and

$$f_{\text{imag}}(t) = 2h(t) * (f_{\text{in}}(t)\sin\omega_c t). \tag{2.31}$$

Note that Equations (2.30) and (2.31) are themselves both real-valued. Both results are obtained easily by multiplying the input by two sinusoids which are 90° out of phase with each other.

Each part (sine and cosine) of the demodulation process produces a new signal which contains the sum and the difference of the original and "carrier" frequencies. Since we are demodulating the carrier frequency down to zero, we are interested only in the difference, so we use the lowpass filter $h(t)$ to get rid of the part containing the sum of the two signal frequencies. From the remaining difference, one cannot separate signals on one side of ω_c from those on the other since ω_c is now at zero and negative frequencies have no physical meaning. In the real-valued signal, negative frequencies are flipped over the frequency axis and alias into the positive frequency spectrum. However, the information necessary to separate the positive from the negative frequency components is preserved in the phase relationship between $f_{\text{real}}(t)$ and $f_{\text{imag}}(t)$, as shown by the exact reconstruction below.

In order to remodulate the signal back to its original frequency, we perform the following multiplication:

$$\hat{f}_{\text{in}}(t) = f_{\text{out}}(t)(\cos\omega_c t - j\sin\omega_c t). \tag{2.32}$$

This is the same function as the demodulation (2.28) except for the change in sign and the lack of a lowpass filter function. We multiply out the real and imaginary parts of this equation to get a purely real result:

$$\hat{f}_{\text{in}}(t) = f_{\text{real}}(t) \cos \omega_c t + f_{\text{imag}}(t) \sin \omega_c t. \tag{2.33}$$

This step separates the negative from the positive frequency components as it pushes the center frequency up from zero to its original value. The result is the exact reconstruction of the original input (within the limits of a physical lowpass filter to approximate the ideal one used here).

The signs have worked out such that the remodulating sinusoids have exactly the same phase relation as the demodulating sinusoids. This fact suggests that an efficient architecture should make use of the same hardware to perform both the demodulation and the remodulation. In other words, a single chip can be configured either as the function decomposer or as the function reconstructor. In the instance of the continuous wavelet transform, the lowpass filter for the demodulation can be combined with the Gaussian filter of the transformation such that no additional filter is required.

2.6 Post-processing

Subsequent to complex demodulation and Gaussian filtering the system outputs are in a form useful for signal processing: for instance, analog methods of compression wherein signal bands with energy less than a critical threshold can be eliminated to save bandwidth prior to transmission and reconstruction. As described thus far, the Gaussian CWT is a band equalizer, but with all the outputs occupying frequency space around zero frequency. Thus the outputs are not in a form suitable for efficient transmission, as they all overlap in the frequency domain. There are two ways to arrange the outputs for transmission to the reconstruction system:

1. Modulate the signals into nonoverlapping frequency bands
2. Sample the system and time-multiplex the samples into an efficient representation.

The first method is the method of reconstruction, although the separated channels can be modulated to any desired transmission frequency and ordered in any manner or compressed by the scheme mentioned above to reduce the total transmission bandwidth. The second method allows more flexibility in the representation by interleaving samples. It is also more faithful to the idea of *tiling* the time-frequency plane: the bandpass filterbank quantizes the frequency domain; a sampler

quantizes the time domain. A true wavelet transform needs to quantize both. A bandpass filterbank whose outputs are not sampled is not, strictly speaking, a wavelet transformer.

Since the bandwidth of each channel is proportional to the center frequency as required in a constant-Q system, and the contents of each band have been shifted down in frequency until the band is centered around zero frequency, the most efficient description in terms of the time-frequency uncertainty relation in a sampled-data representation requires that each frequency band be sampled at a rate proportional to its bandwidth. The result of the sampling is that each rectangle in the time-frequency plane shown in Figure 2.1 has an equal area representing the effective time and frequency bandwidths of the Gaussian filter (with some overlap). The overlap of filter functions is depicted in Figure 2.2. If the filter channels are sampled in the binary-tree fashion shown when the channels are centered on a $\log_2$ scale, the samples can be easily time-division multiplexed into a single output stream [17].

2.7 An Analog CWT Processor

The first attempt to build a Continuous Wavelet Transforming processor consisted primarily of a continuous-time, subthreshold analog design fabricated in a standard CMOS process. The analog circuits were based on the analog VLSI techniques described by Carver Mead in *Analog VLSI and Neural Systems* [1]. The underlying idea was to generate a set of exponentially-spaced clock (square) waveforms which would then be shaped (via filtering) into sine and cosine pairs, and multiplied directly with the continuous-time, continuous-valued input using a translinear (analog) multiplier.

This chip was designed in subthreshold analog as an alternative to using a computer or DSP system to perform the same transform. The advantages of this approach are reduced size and power consumption. The resulting implementation is inflexible in terms of ability to reprogram the type or order of the wavelet function, and requires dealing with the problems of temperature sensitivity, nonlinearity of analog computation, variable process parameters, and noise injection throughout the circuit (particularly that caused by the digital circuits generating the square wave). As will be seen presently, not all of these problems could be overcome sufficiently, resulting in a move toward a more mixed-mode architecture (Chapter 2 Section 2.15).

2.8 Generating carrier sinusoids

This chip, as mentioned above, used square waves (clock signals) as its basis for generating sinusoids at specific frequency and phase. Specifically, the architecture called for each channel center frequency to be half the value of the neighboring channel. Additionally, the architecture was made such that each wavelet processor core would generate center frequencies for six channels. The master clock is applied to the system most conveniently as a two-phase, non-overlapping signal which becomes the first input to a cascade of toggle flip-flops (T-FFs) made in the standard way from two transparent latches in series. The two-phase flip-flops conveniently give outputs from the first and second latches which are 90° out of phase with each other (this is necessarily true for the input of the second stage and beyond because each of the two-phase outputs of the previous stage is forced to be 25% duty cycle, and the two phases are exactly 180° apart. The same can be ensured for the first stage by doubling the master clock frequency and preceding the first stage with another toggle flip-flop). The flip-flop configuration is shown in Figure 2.7. Signals *cos(in)* and *sin(in)* are the pulse trains corresponding to the frequency and phase of the sine and cosine components of the preceding channel. Signals *cos(out)* and *sin(out)* are the pulse trains which are shaped by filtering and become the modulating signals for the current channel. The reset login ensures that the sine and cosine parts have the correct phase relative to each other upon initialization of the circuit.

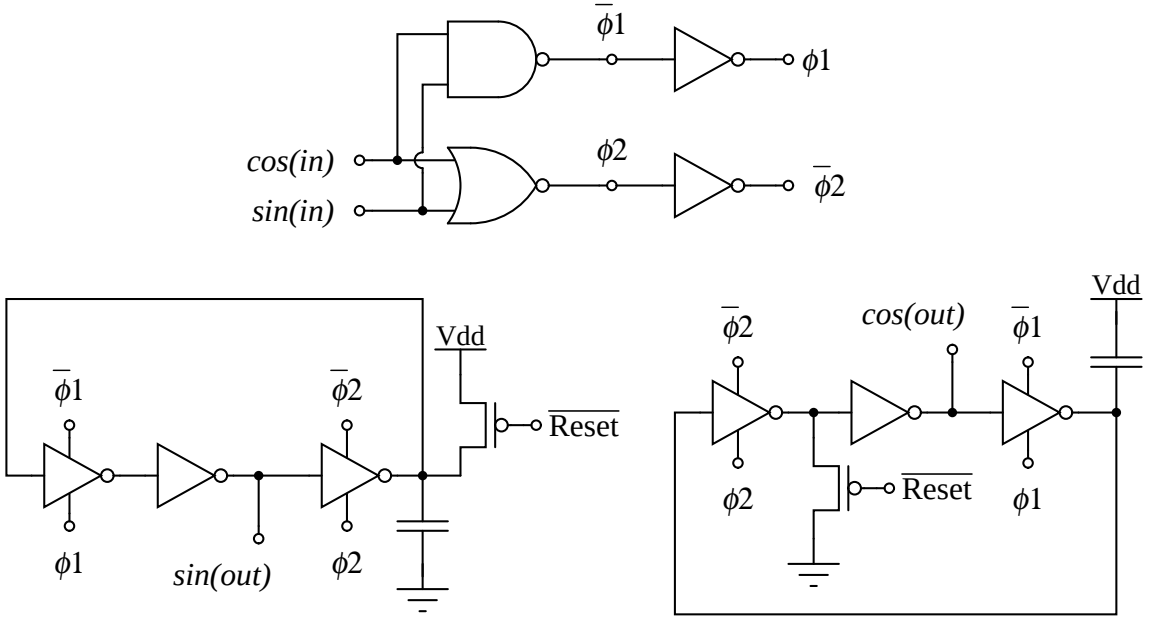

Figure 2.7: Circuit diagram of the frequency-division toggle flip-flops.

2.9 Analog multiplication

Purely analog multiplication is very problematic due to the limited range of linearity in such circuits, especially when the circuits are realized in subthreshold CMOS technology. A standard circuit for analog multiplication of voltages is the Gilbert multiplier (Figure 2.8). The Gilbert multiplier is based on the translinear principle and in terms of linearity and noise performance is best implemented using bipolar transistors rather than MOSFETs for the critical components of the translinear loop (for a full discussion of translinear circuits and the tradeoffs between MOSFET and bipolar devices, see Chapter 3).

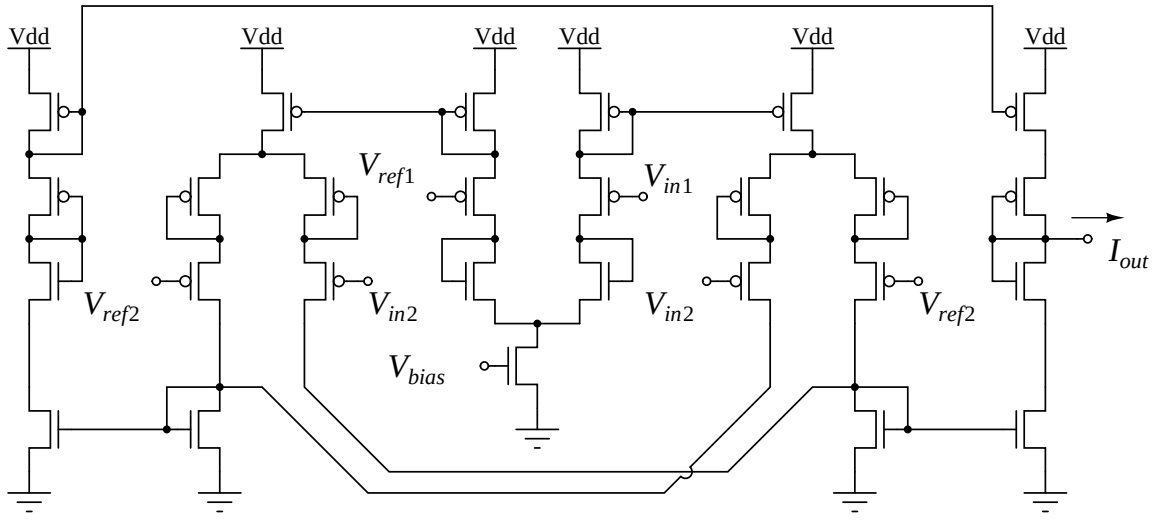

Figure 2.8: Gilbert multiplier with cascodes.

The Gilbert multiplier circuit executes a nonlinear function which, for input signals close to the reference, can be described by the linear approximation $I_{out} \propto (V_{in1} - V_{ref1}) \cdot (V_{in2} - V_{ref2})$. As with most subthreshold MOS translinear circuits, the input voltage swing is limited to a few kT/q, or about 50 mV. and is difficult to extend by more than a factor of two or so through the use of increasingly complicated linearization techniques [5]. A simple derivation of this limiting voltage can be found in Appendix A.

2.10 Wavelet Gaussian Function

In addition to allowing simple sampling of each output channel, the use of complex demodulation simplifies the process of designing bandpass filters that maintain a Gaussian shape while

allowing variable center frequency and width. Because the signals are first demodulated until the center of the frequency band of interest is placed at zero frequency, it is only necessary to create a Gaussian lowpass filter, which is one half of a Gaussian function placed at zero frequency. Two identical filters are required for the sine and cosine parts of the complex demodulation. The signals are remodulated to their respective center frequencies during reconstruction using the same method (and preferably the same hardware, if feasible). The resulting output will be reflected symmetrically across the modulation frequency, behaving as if it had been passed directly through a bandpass filter with Gaussian characteristics. For demodulation used in the context of the continuous wavelet transform, the lowpass filter for the demodulation can be combined with the Gaussian filter required by the wavelet transformation. Note that, according to Equation 2.33, reconstructing the signal does not require any filtering subsequent to remodulation.

We based the design of the circuit which approximates the half-Gaussian lowpass filter function (as described by Grossman [13]) on a probability argument. First, I present Equation (2.34) which describes the filter transfer function of the circuit:

$$H(s) = \frac{V_{\text{out}}}{V_{\text{in}}} = \left(\frac{1}{\tau s + 1}\right)^n. \tag{2.34}$$

This filter function describes a cascade of n first-order lowpass filter sections in series. Although Equation (2.46) converges to a delta function in the limit as $n \to \infty$ for constant τ, it can be shown that when τ is replaced by an expression which maintains constant bandwidth, the transfer function approaches the Gaussian function (2.5) as $n \to \infty$. A proof of this equation can be found in [31] which shows that the Gaussian shape is an example of the central limit theorem of probability: In other words, it is a result of the use of cascaded stages and is relatively independent of the shape of the filter.

For n sections in cascade, the relationship between τ and the Gaussian bandwidth σ from the wavelet mother function, Equation (2.5), is

$$\sigma = \frac{1}{\tau\sqrt{n\ln(2)}}. \tag{2.35}$$

The circuit is shown in part in Fig. 2.11. Although not an architectural necessity, we chose to implement each first-order filter as a transconductance-C filter using transconductance amplifiers operating in the subthreshold region. The transconductance amplifiers are connected as voltage followers, which in conjunction with the capacitor at the output is a configuration called a "follower-integrator." Due to considerations of signal-to-noise ratio and the ability to generate a

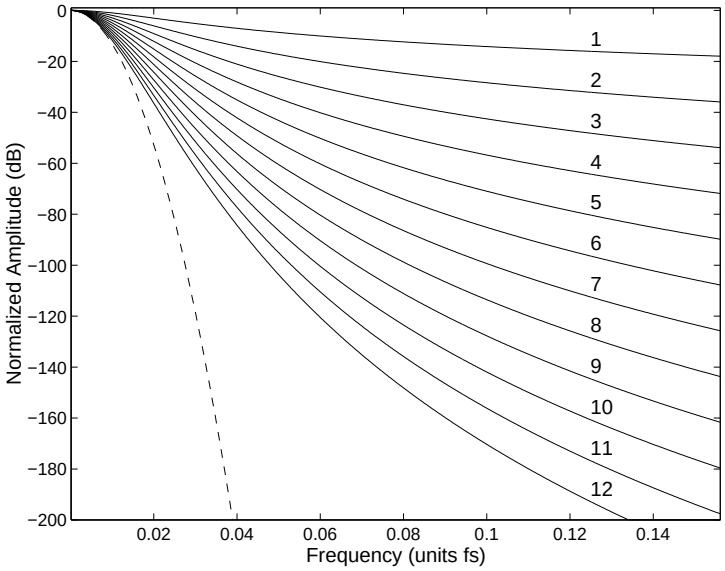

Figure 2.9: Frequency-domain transfer function of the final output of n cascaded lowpass filters as a function of the number of stages n.

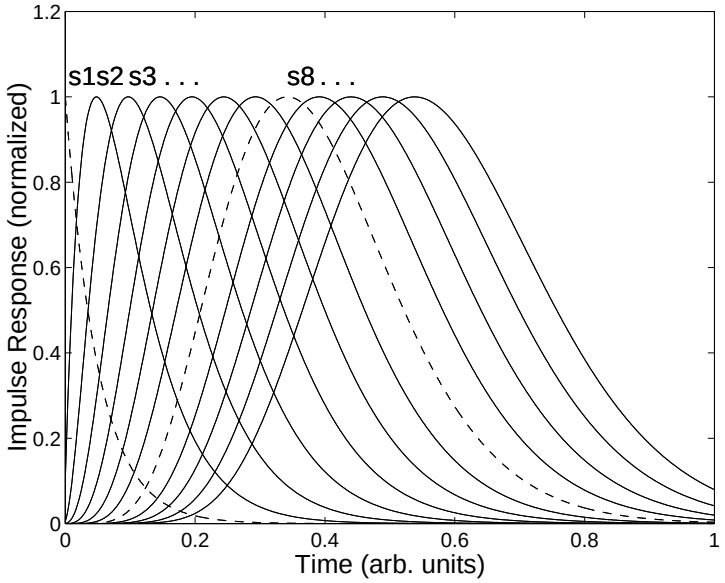

Figure 2.10: Impulse response of the final output of n cascaded lowpass filters as a function of the number of stages n.

given constant τ from a voltage applied to the follower-integrators, we chose a cascade order of five sections.

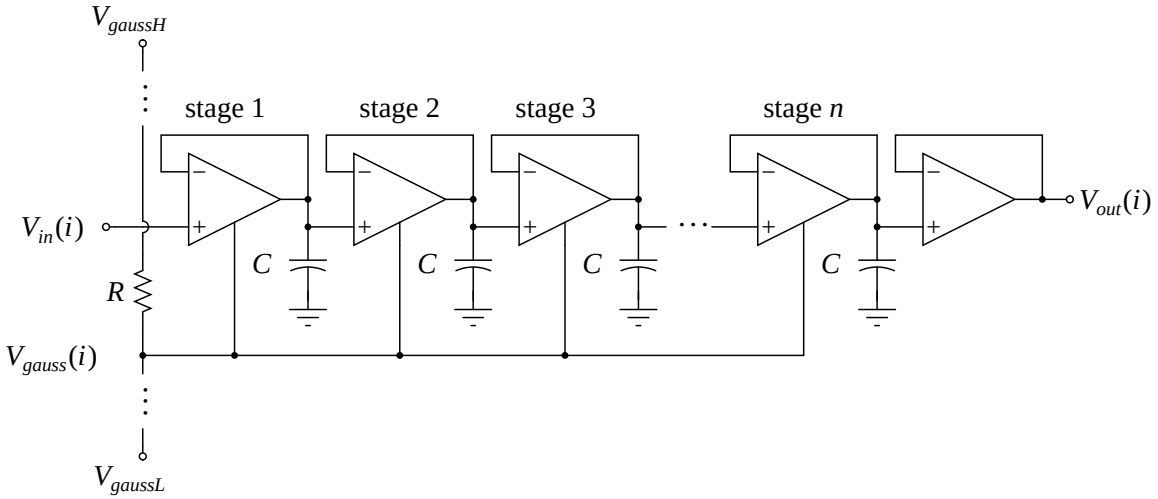

Figure 2.11: n cascaded stages of a filter approximating a half-Gaussian function, using continuous-time transconductance-C filters.

The center frequencies of the filters in the filter bank are spaced on a $\log_2$ scale. Assuming speech-quality bandwidth for the input signal, we decided that six outputs would be sufficient, giving typical center frequencies of 9kHz, 4.5kHz, 2.25kHz, 1.125kHz, 562.5Hz, and 281.25Hz (unless more than one system is interleaved in the manner described in Section 2.22). The center frequency of the highest-frequency filter is determined by an oscillator which can be generated either on-chip for a voltage-controlled frequency, or off-chip for a stable frequency. The center frequencies of the rest of the filters are determined by dividing down the oscillator appropriately.

The bandwidth of each Gaussian filter is set automatically with respect to the others with the exception of the first and last filters, which have widths adjustable using two control voltages V_{gaussH} and V_{gaussL}. In terms of the transfer function (2.46) for the filter, the parameters τ of the highest- and lowest-frequency filters are fixed by these control inputs. τ should be calculated to assure that the width of each Gaussian is proportional to the value of the center frequency, as it is not determined automatically from the other system parameters.

More recently, this idea has been expanded upon by Harris *et al.* [22] for the creation of so-called "gamma-tone" filters and filter structures which use carefully calculated weighted feedback and feedforward connections to cause the filter transfer function to reach a given accuracy of approximation of the Gaussian shape in significantly fewer stages.

2.11 Wavelet chip slice

The parallel nature of the analog wavelet computation allows the VLSI layout to be generated easily by abutting slices of circuitry. The filtering functions are identical for the sine and cosine parts of the transform, so slices are logically grouped in pairs, with each slice containing one transparent latch such that the pair of slices forms the flip-flop for dividing down the clock input to that section. Thus the path of the frequency stepping runs vertically across all channels while the signal path runs from left to right. In addition to generating the divide-by-two oscillator, each slice is responsible for shaping (filtering) the oscillator signal to produce a smooth sine (or cosine) wave, multiplying this modulating signal with the input, and filtering the result through a half-Gaussian-shaped lowpass function. Finally, depending on whether the section is configured for decomposition or reconstruction, the circuit samples the filter outputs and multiplexes them into a stream (decomposition), or else aggregates all the outputs together to produce the final result (reconstruction).

A block diagram of decomposition and reconstruction for a wavelet transform processor sine/cosine pair (single channel, or "slice") is shown in Figures 2.12 and 2.13, respectively. The entire analog Wavelet Transform chip is shown in Fig. 2.14.

2.12 Chip Specifications

- Power Supply: +5 V DC $\pm$5%
- Input mean value: 2.5 V $\pm$0.5 V
- Input p-p amplitude: 0.4 V
- Input frequency range: 80 Hz to 10 kHz
- Number of output channels: 12 (6 pairs)
- Silicon area: 1.96 $\times 10^6$ μm^2 (in a 2.0 micron CMOS process)
- Chip package: 68-pin PLCC

The technologies used to fabricate the several versions of the analog Continuous Wavelet Transform processor are as follows:

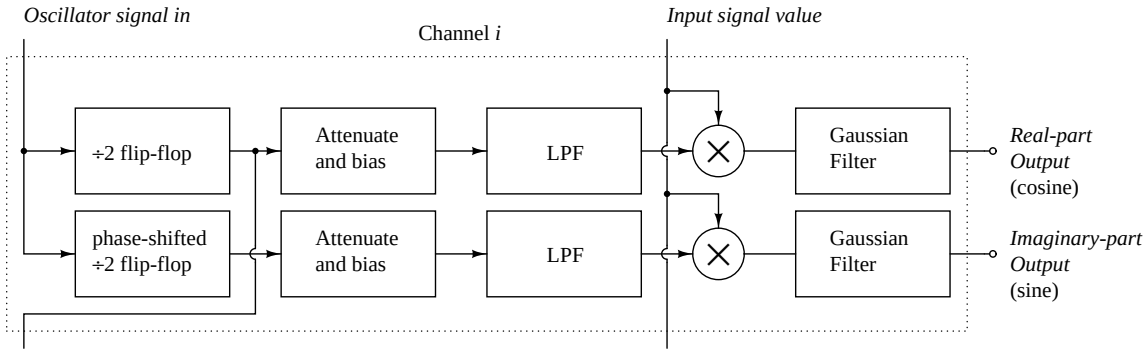

Figure 2.12: Wavelet decomposition block diagram for a single sine/cosine pair.

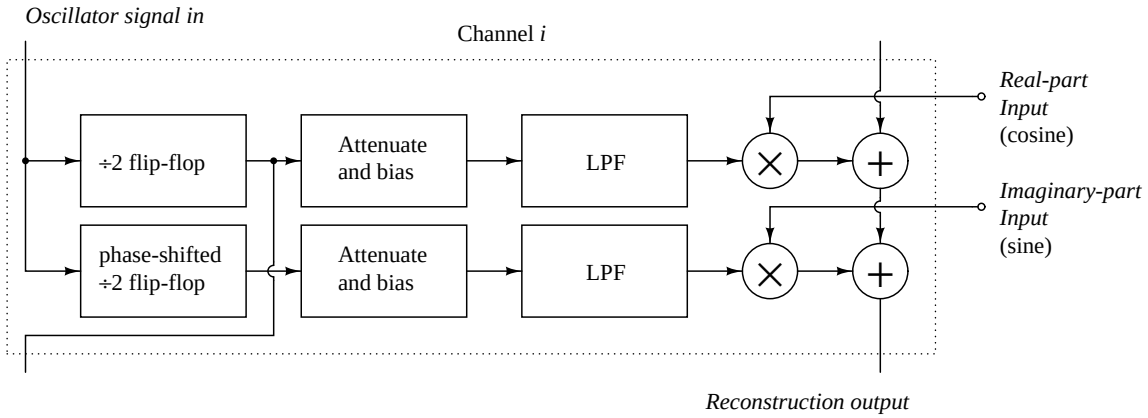

Figure 2.13: Wavelet reconstruction block diagram for a single sine/cosine pair.

Table 2.1: Technologies used for the Wavelet processors.

Chip name	*Foundry*	*Min. feature size*	*Well type*	*Poly*
WaveChip2	Zilog	2.0 micron	twin-well	single
Z89c55ba	Zilog	1.2 micron	twin-well	single
WaveChip5a	Orbit	2.0 micron	N-well	double
WaveChip5b	Orbit	2.0 micron	P-well	double

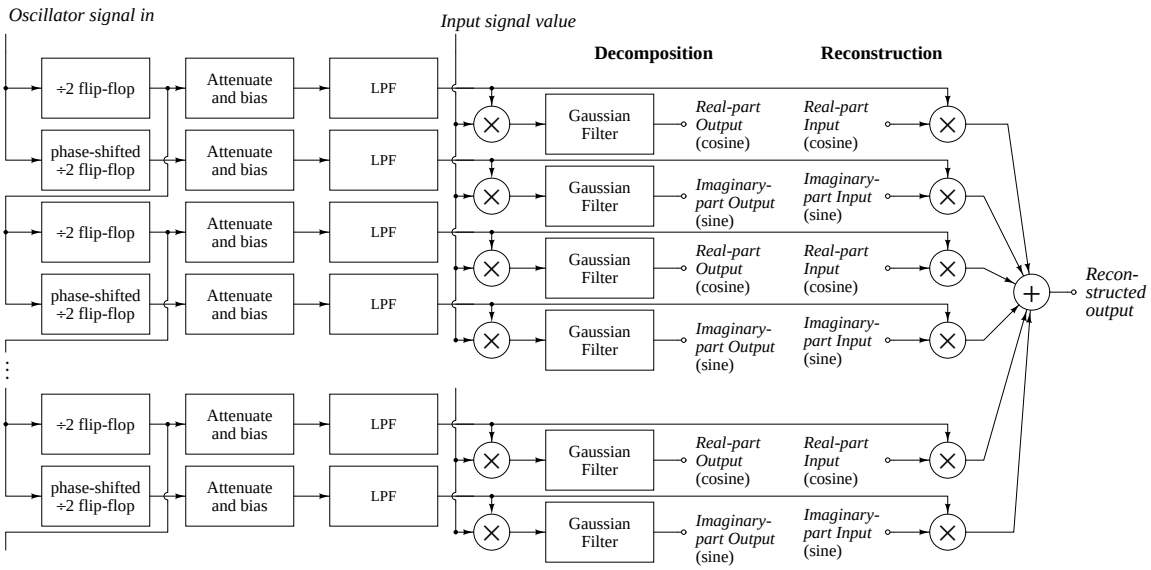

Figure 2.14: Wavelet Transform Chip block diagram.

2.13 Limitations of the Architecture

The major drawback of this analog architecture involves the use of filtered square wave signals as the modulating sinusoids. The use of a square wave as a modulation carrier is well-documented (see, for example, Horowitz & Hill, 2nd ed., p. 437) [28]. However, its use is restricted to modulation in the usual sense wherein a low-frequency signal is pushed up into a high-frequency band. The Fourier series describing a square wave contains an infinite series of odd harmonics beginning with the fundamental, which attenuates only as $1/n$, where n is the harmonic number (1, 3, 5, ...). The modulation performed by multiplying the square wave by the input signal can be broken down by superposition into the multiplication of each of the square wave harmonics with the input. If the destination band (carrier frequency) is large enough, then all of the intermodulation products (multiplication of the input by all harmonics greater than the fundamental) are very far out of band and can be easily attenuated with a simple filter. Unfortunately, if the function performed is demodulation instead of modulation, then by definition the carrier frequency is in the band of the signal and so any of its harmonics may be also. Consequently, intermodulation products will be in-band and cannot be filtered out. Shaping a square wave into a sinusoid by filtering out the higher harmonics is a difficult prospect at best involving complicated high-order elliptic filters; the filter cutoff must be prohibitively sharp to pass the fundamental frequency but attenuate the third harmonic to reasonable levels for clean signal processing (at least 40 to 50 dB for most audio

applications; 60 to 70 dB for high fidelity). Complicated filter architectures were not addressed in this series of wavelet processors, and so results did not reach acceptable levels of performance.

2.14 Variations on an Architecture

One interesting variation on this design was created and developed by Moreira-Tamayo *et al.* [23, 24] at Texas A&M University. Using primarily the same complex demodulation technique based on our architecture, the authors translated the entire problem into the time domain. In place of the wavelet Gaussian function, they used a rectified cosine (also known as a *Hanning window*, a function which is reasonably easy to generate in the time domain and which is reasonably close to the Gaussian in its time and frequency support. Its time-frequency product is approximately 0.513, or 2.6% larger than the minimum area (1/2) of the Gaussian. The rectified cosine itself becomes the envelope function; the wavelet function itself is generated by multiplying the rectified cosine by a sinusoid of higher frequency. Effectively, this is a double demodulation, and simplifies the implementation somewhat by making use of the same hardware architecture for generating both the Gaussian (or in this case, Hanning) envelope and the wavelet modulating function. Their wavelet family can be described by the function:

$$\psi(t) = Ae^{-j\omega_c t}\left(1 + m\cos\left(\omega_p t\right)\right), \qquad -\pi < \omega_p t < \pi \tag{2.36}$$

where this function is repeated (chained) in time, and the system repeated over dilations of the frequency. The wavelet computation on an input function $f(t)$ for frequency scale a and time shift b is written

$$CWT(f,a,b) = \frac{1}{\sqrt{a}}\int_{\ell_i}^{\ell_f} f(t)\cdot\left(g\left(\frac{t-b}{a}\right)\, v\left(\frac{t-b}{a}\right)\right)dt \tag{2.37}$$

where $g(t) = \exp\left(-j\omega_c t\right)$ is implemented as a complex demodulation by separating the function into $\sin\left(\omega_c t\right)$ and $\cos\left(\omega_c t\right)$, and $v(t) = 0.5(1 + m\cos\left(\omega_p t\right))$. The architecture of Moreira-Tamayo *et al.* also follows our architecture in its use of a master clock divided down by flip-flops for generation of multiple frequency square wave signals subsequently filtered to generate the sinusoid modulator. However, the system is built at the component level rather than as a VLSI processor. The analog multiplications are implemented with MC1494 analog multiplier integrated circuits. The integration operation in Equation (2.37) is implemented by a transconductance-C circuit. A slightly simplified block diagram of one channel of the system is shown in Figure 2.15. Only the wavelet decomposition was reported in [23]. Compare Figure 2.15 to the block diagram of the decomposition half of our wavelet decomposition architecture, Figure 2.12.

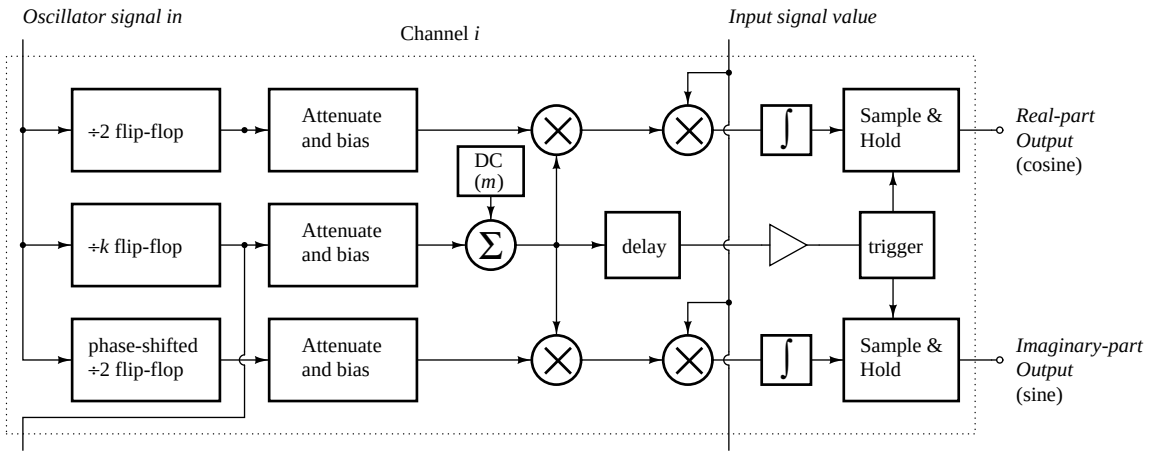

Figure 2.15: Architecture of Moreira-Tamayo *et al.*

The Moreira-Tamayo architecture quite nicely demonstrates the duality of the wavelet transform: Because it incorporates both the time domain and the frequency domain, efficient architectures can be realized in either one. The response of both systems to the same input is the same.

2.15 A Mixed-Mode Wavelet Processor

Several hardware implementations of the Continuous Wavelet Transform and related bandpass filter bank architectures have been reported in recent years [17, 19, 23], including the analog architecture discussed at the beginning of this chapter. The remaining sections of the chapter highlight a novel architecture we developed for the continuous wavelet transform processor which is unique in its encoding of the decomposition output and use of oversampling techniques.

The goal of the new architecture was to overcome the obvious drawbacks of the analog subthreshold MOS circuits, namely the quality of the carrier sinusoid signal and the linearity of the modulation multiplication. The new architecture relies more on digital processing and as such is more appropriately considered a "mixed-mode" design.

Other than the novel circuit methods used, the new design retains the essential characteristics of the original wavelet architecture. To recap: The processor performs a demodulation of the audio-frequency input signal in parallel across N channels, where the channels are adjacent with minimal overlap, cover the audio frequency band of interest, and are centered on a logarithmic scale. The process of demodulation shifts the signal frequencies from each channel center frequency

down to zero. Each demodulation result is then filtered by a lowpass version of the wavelet function, which also serves as the postmultiplication filter for the demodulator. Signal information on both sides of the channel center frequency is preserved by using complex demodulation, in which each channel is split into two parts, with the modulating sinusoid of one being 90° out of phase with that of the other. The CWT is invertible: The reconstruction process consists of modulating the channel outputs back to their respective center frequencies, and summing them all together. The decomposition method is shown in Figure 2.5 and the reconstruction method in Figure 2.6.

The wavelet nature of the output allows the channel outputs to be efficiently encoded. The outputs of the decomposition are time sampled at the Nyquist rate of each channel, and all the sampled channel outputs are time-multiplexed into a single stream. The reconstruction processor decodes the data stream prior to reconstruction.

As we developed the oversampling architecture described in the previous section, it became clear that, as the carrier signal was both binary and discrete-time based on a well-defined synchronous clock, the most efficient architecture would maintain the discrete-time nature of the carrier, and retain a digital mode of processing throughout as much of the architecture as possible. Following this line of reasoning leads to an elegant and extremely efficient design for a complex demodulation multiplier and Gaussian filter.

In the introduction to this chapter, I mentioned that by using complex demodulation, very accurate center frequencies for the bandpass function derive directly from the carrier signal frequency, which itself is derived from a quartz oscillator. In the previous section, the wavelet transform architecture was developed, but the circuits using that architecture could not achieve acceptable performance due to intermodulation products caused by poorly attenuated harmonics of the carrier signal. Analog multipliers built from analog circuits by nature have a limited range of operation due to nonlinearities in the circuit function. It remains to be explained how to get from the frequency-accurate, digital-domain square wave clock signal to an accurate and repeatable sinusoidal signal. That is the purpose of this section.

We developed a method for sinusoidal modulation of analog signals which does not require an explicit multiplication, and hinges on generation of accurate analog sine wave using the mixed-mode technique of oversampling.

As demonstrated by repeated failure of performance of previously fabricated versions of the wavelet processor, obtaining a sinusoidal signal by lowpass filtering a square wave introduces too much distortion to make the system usable. The modulation function is sensitive to distortion harmonics, which get multiplied by the input signal and act like many separate modulations about

many different frequency axes. Each produces its own sum and difference components, and some are bound to end up in the signal band at the system output. In order to get predictable system performance, it is necessary to have a sinusoid carrier signal with a fixed and controllable amount of harmonic distortion. To this end we considered the possibility of generating an oversampled version of a sinusoid using a delta-sigma modulator.

The delta-sigma circuit of Lu *et al.* [26], an excellent example of the method, begins with a standard resonator system consisting of a simple loop of two integrators. Implemented in the digital (z) domain, each integrator requires a delay and accumulate operator, and one multiplication by a constant coefficient. The delta-sigma method eliminates the multi-bit multiplications by inserting a delta-sigma modulator into the loop which renders part of the resonator circuit a single-bit value, where the single bit represents an oversampled sinusoidal oscillation. Figure 2.16 shows the architecture, where no actual multiplications are required: One multiplication is reduced to a multiplexing operation, and the other in the digital domain is a bit shift operation. The remainder of the system still requires precision digital delay and accumulate operations, as does the implementation of the second-order delta-sigma modulator.

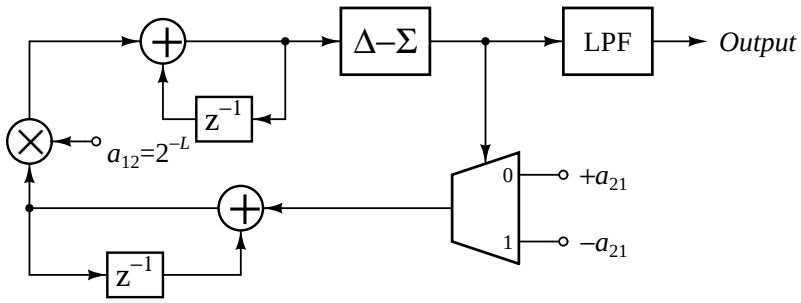

Figure 2.16: The analog oscillator architecture of Lu *et al.* [26] containing a 2nd-order delta-sigma modulator.

Eventually it became obvious to us that our system is much more specific than the general delta-sigma method described above. More to the point, the delta-sigma method is rather a bit of overkill for the purpose of generating a sine wave, particularly when only a finite set of frequencies is required rather than a full range of arbitrary values. We need to generate a number of sinusoids of known fixed frequency and amplitude. Presumably there exists a fixed sequence of bits which describes the optimal n-times oversampled sinusoid in the context of our system for a given value of n. If the optimal sequence can be determined, then it has a fixed amount of harmonic distortion which should decrease with increasing n as the oversampled sequence contains increasing amount

of information about the encoded signal.

The above phrase "in the context of our system" is critical in that it is not to be assumed that the same sequence is optimal for every possible implementation. The frequency-domain representation of the original (unfiltered) single-bit oversampled sequence can be determined by converting the bit sequence directly to a sequence of voltages in the time domain according to the following rules:

- bits '1' are represented by some fixed positive voltage
- bits '0' are represented by some fixed negative voltage
- each voltage is applied for a fixed length of time t such that the entire sequence of n bits has length $T = nt$.

We then compute the FFT of this sequence. A sequence which accurately encodes a sinusoid should show a large FFT component at the frequency of the sinusoid (which without loss of generality we consider to be the inverse of the sequence length, $f_0 = 1/T$), show a highly attenuated response at all frequencies close to the fundamental, with the harmonic distortion increasing at large frequencies (an inherent property of oversampled signals). The quality of the sinusoid in the context of the system cannot be determined from this FFT result. The oversampled sequence must be lowpass filtered to retrieve the encoded signal, and so the quality of the resulting signal depends both on the binary sequence used and the properties of the lowpass function used to retrieve the analog signal from its oversampled representation. Furthermore, the input signal must be prefiltered with a lowpass to attenuate components which would mix with the oversampled sequence and add to the resulting harmonic distortion. The point is that the harmonic distortion is determined by the process of the modulation itself—specifically, the way that it causes frequencies of the input to mix with harmonic distortion components of the carrier to become error components at the output—and so the binary sequence required for optimal performance is intimately tied to the system itself.

One fallout of this consideration is that the bit sequence used to encode the sinusoid cannot be determined until the system is known in detail. On the other hand, it is not possible to know the exact requirements for the filtering system (for instance, size requirement of the VLSI layout) until we know what the sequence is. Some iterations can be expected before a solution is agreed upon. This fact makes it necessary to have a method which can fairly rapidly determine the optimal sequence.

It should be clear at once that a full search of the n-bit space of the sequence is an "np-hard" problem, and furthermore that each search involves a large amount of computation for simulating the antialiasing filter and output filter, computing an FFT, and measuring harmonic distortion. For the purpose of finding the optimal sequence, we define harmonic distortion as the ratio of the amplitude of the largest distortion product to the amplitude of the fundamental frequency.

On the other hand, it should also be clear that there are a number of factors in our favor:

1. Inverted sequences have the same response as their non-inverted forms, and do not need to be investigated. This is also true of reversed sequences.

2. Sinusoids have quarter-wave symmetry, and therefore the sequences should also (symmetric signals can be closely approximated by non-symmetric sequences, as is generally the case in a 2nd-order delta-sigma system, but there is no reason to explore this space). This cuts the search space by a factor of four.

3. Only a small subset of 2^n sequences look anything like a sinusoid, allowing the possibility of heuristic approaches.

The first consideration involves generating a quarter sequence, and then generating the other three quarters by (in turn) reversing the sequence, inverting and reversing, and reversing once again. A reasonable heuristic approach is to start with a known sequence which looks something like a sinusoid, namely a square wave (for which the first quarter sequence is all '1' bits). From that point we need an iterative method which will explore the local space of nearby bit sequences ('nearby' meaning in a Hamming distance sense) which will (hopefully) converge quickly to a solution without encountering local minima along the way.

As is often true for these sorts of methods, we do not present (nor have we even attempted) a proof of convergence. Our method successfully found bit sequences which exceeded the system constraints and which met with our approval (*i.e.*, the sequence was short enough that it was conceivable to design a simple sequence generator to fit the wavelet system on a 2mm $\times$ 2mm layout).

2.16 Details of the Bit-Sequence-Finding Algorithm

It is possible to ascertain some properties of the solution before running the algorithm. For instance, the average of ones and zeros in the bit sequence must match the integration under the

sinusoid, and the difference between the two gives a rough indication of the minimum distortion possible using a length-n sequence.

The way we have defined the system in Section 2.15, a DC value of zero is represented exactly by a bit sequence of alternating ones and zeros ($\pm$ voltage). A DC value of one is represented exactly by a bit sequence of all ones. Given this consideration, the expected ratio of ones to zeros in a bit sequence in the first case is 1/2, and in the second case 1. Following this line of reasoning, the expected ratio of ones to zeros in the first quarter of the sine sequence is

$$\int_0^{\frac{\pi}{2}} (1 + 0.5 \sin x) dx. \tag{2.38}$$

The total integration divided by the interval comes to $1/2 + 1/\pi$, or 0.8183. This multiplied by 64 bits, for example, yields an expected value of between 52 and 53 'one' bits, leaving 12 or 11 'zero' bits. This is what we can expect the makeup of near-optimal sequences to be.

Comparing the integration under the curve to the instantaneous average of the bit sequence at each step and then determining the next bit of the sequence accordingly is exactly the method of sigma-delta modulation. The bit sequence produced is always changing from cycle to cycle due to the residual error between the filtered bit sequence and the integral under the sine curve, which is never zero.

A sequence of n bits which is repeated exactly on every cycle can never achieve the accuracy of an n-bit delta-sigma modulator. On the other hand, as n increases, so does the accuracy, so that if the system requirement is a maximum fixed value of total harmonic distortion, there exists a fixed bit sequence of some minimum length n which will meet that requirement.

The problem in finding the sequence is this: A sigma-delta system is deterministic: The next bit in the output sequence is an exact function of the current state of the system. But a sigma-delta-like method cannot be used to find an optimal fixed sequence, because changing any bit in the fixed sequence changes the past, present, and future state of the system. Instead, it is necessary to search the space of 2^n bit sequences to find one which achieves the desired accuracy.

Knowing the expected number of 'one' and 'zero' bits in the sequence still doesn't help much in finding the optimal sequence: The set of all combinations of m 'zero' bits in a sequence of length n remains computationally intractable for reasonable values of n (such as 64).

The modulation system for the CWT, when using the oversampled sequence method, is described by

$$y(t) = [(x(t) * g(t)) \cdot s'(t)] * h(t), \tag{2.39}$$

where $g(t)$ the impulse response of the lowpass prefilter and $h(t)$ is the postmultiplication filter as before (Section 2.4). The binary sequence $s'(t)$ is assumed periodic with quarter-wave symmetry, and therefore can be described in the frequency domain by a Fourier series containing only harmonics of odd orders in ω_0:

$$S'(\omega) = \sum_{k=0}^{\infty} {S'}_k(\omega) \,, \tag{2.40}$$

$${S'}_k(\omega) = jc_k \left[\delta\left(\omega - (2k+1)\omega_0\right) - \delta\left(\omega + (2k+1)\omega_0\right)\right]. \tag{2.41}$$

The fundamental component ${S'}_0(\omega)$ corresponds to the desired sinusoidal signal $s(t)$.

Figure 2.17 shows how the two systems operate under the assumption that the lowpass filter functions $H(\omega)$ and $G(\omega)$ are ideal, *i.e.*, flat in the passband and with infinite rolloff at the cutoff frequency. Under this assumption, it can be seen from the figures that an arbitrary input spectrum $X(\omega)$ corresponding to the time-domain input $x(t)$ produces the same output $y(t)$ for both systems if and only if the prefilter bandwidth $\mathrm{BW}(G(\omega))$ is constrained by

$$\omega_0 + \mathrm{BW}(H(\omega)) < \mathrm{BW}(G(\omega)) < 3\omega_0 - \mathrm{BW}(H(\omega)) \,. \tag{2.42}$$

if the last inequality is not satisfied, convolution products of $X(\omega) * {S'}_k(\omega)$ for $k > 0$ will be aliased into the output.

In reality, the filters $H(\omega)$ and $G(\omega)$ have finite rolloffs, and the equivalence between the systems in Figure 2.17 is only approximate. The quality of the approximation depends on the harmonic coefficients c_k corresponding to the binary sequence, which can be optimized for minimum distortion. There is a trade-off between the complexity of the sequence and that of the filters G and H, as illustrated in Section 2.18.

2.17 Sequence generation

The output spectrum generated by the modulation scheme contains intermodulation products between the prefiltered input $G(\omega) \cdot X(\omega)$ and the harmonics of ${S'}_k(\omega)$, with terms of the form

$$jc_k \; G\left(\omega \pm (2k+1)\omega_0\right) \cdot X\left(\omega \pm (2k+1)\omega_0\right) \cdot H(\omega). \tag{2.43}$$

Only the fundamental term $k = 0$ is desired, and distortion arises from the higher-order intermodulation products, $k > 0$. To reduce distortion, the coefficients c_k need to be small for $k > 0$,

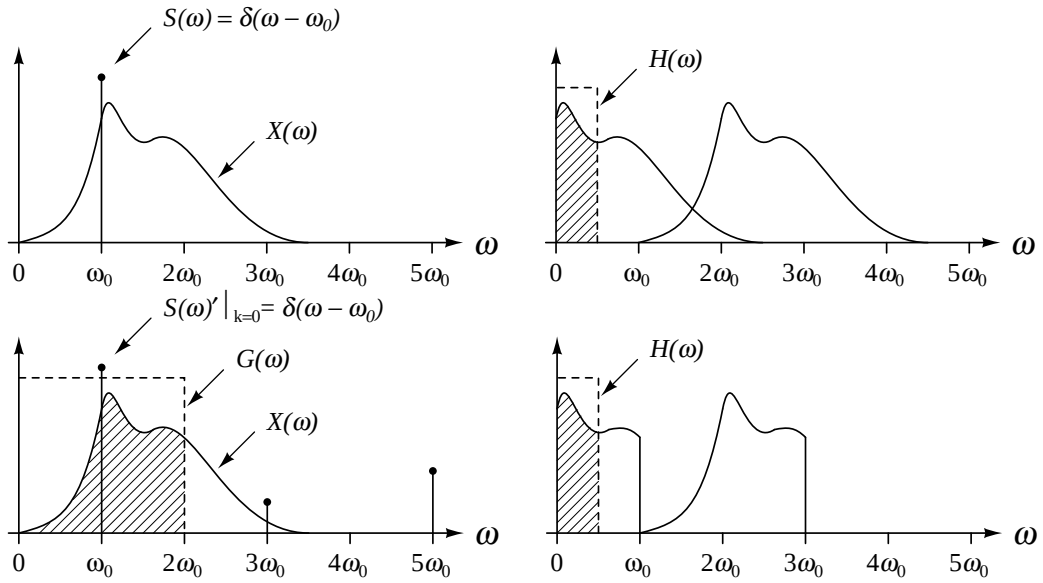

Figure 2.17: *Top*: Demodulation of an input X in the frequency domain with a perfect sine wave S and ideal modulation filter H. *Bottom*: Demodulation of an input X in the frequency domain with an oversampled sine wave S' using an ideal smoothing filter G and modulation filter H.

except for large values of k for which the terms of (2.43) are reasonably small due to the attenuation by the prefilter G. In other words, the low-frequency components of the binary sequence $s'(t)$ need to approximate the sine wave $s(t)$ as closely as possible. Qualitatively, this corresponds to an oversampled noise-shaped sine wave, as produced by the delta-sigma modulator method [26].

Techniques for deriving periodic sequences with several zero or small harmonic components of c_k are presented in [33]. We formulate the problem of finding an optimal binary sequence directly from a minimum distortion criterion on the intermodulation components (2.43).

In general, the amount of distortion is input dependent, and assumptions need to be made on $X(\omega)$ to formulate an optimization criterion. Our criterion is to maximize the ratio of energy in the fundamental harmonic modulation component ($k = 0$) to the combined energy of the distortion components ($k > 0$). Assuming a narrow bandwidth of $H(\omega)$ and an input spectrum $X(\omega)$ which in the worst case is flat in amplitude, the criterion becomes:

$$\text{Maximize}: \quad \frac{c_0^2 \left|G(\omega_0)\right|^2}{\sum_{k=1}^{\infty} c_k^2 \left|G\left((2k+1)\,\omega_0\right)\right|^2}. \tag{2.44}$$

which is equivalent to minimizing the harmonic distortion of the sequence $s'(t)$, filtered with the same prefilter $G(\omega)$. The criterion can be applied, in principle, to select the optimal bit sequence

$s'(nT)$, for $n = 1 \ldots N$. With quarter-wave symmetry, only $N/4$ bits (one quadrant) need to be determined. Still, this problem has a combinatorial complexity, and becomes intractable for large N. We obtain approximate solutions using a technique of iterative block optimization, where a full search is repeatedly conducted over randomly selected blocks of consecutive bits in the sequence.

Appendix B lists a short Matlab program which performs the block-iterative sequence search. The procedure for determining the harmonic distortion is as follows: A quarter-wave bit sequence is expanded by inverting and reversal into a full wave and described in terms of values $+1$ and -1. An FFT is applied to this sequence and its magnitude computed. The resulting spectrum is multiplied by the frequency-domain transfer function of the lowpass filter (described as an attenuation coefficient per FFT bin). Then we can directly compute total harmonic distortion as the magnitude of the second FFT bin (the signal) divided by the sum of the magnitudes of all the other FFT bins (the distortion).

In lieu of computing every one of the 2^n possible sequences, we combine exhaustive search with a random perturbative method. The exhaustive search is performed over a tractable subspace of m bits, where $m < n$ (generally, $m < 16$ to compute in reasonable time), where the remaining $n - m$ bits are held fixed. On each iteration, the starting point of the subsequence to exhaustively search is chosen at random, the search space of 2^m sequences computed for a minimum, then the minimizing sequence is chosen as the new sequence and the process is repeated. There is no guarantee of success, and in practice the error surface is rife with local minima. Usually, however, the algorithm produces acceptable results, meaning total harmonic distortion of -60 to -70 dB, which suffices for most applications. The formulation of the algorithm leaves open the possibility of variants based on genetic algorithms: at every iteration of the algorithm, the best z solutions are kept rather than than the single best solution, allowing population statistics to determine the course of the optimization. By searching a broader solution space at each step, the system is less likely to become trapped in local minima, and the convergence time is significantly reduced.

2.18 Results and implementation

We demonstrate the principle with the following example and describe a simple and elegant implementation. The filter $G(\omega)$ is third order, implemented as a cascade of three single-pole filter stages, each pole located at $z = 15/16$. The total sequence length is $N = 256$. Figure 2.19 shows the 64 bits of the first quarter of the sequence obtained using the iterative block optimization method with block size 8. Figure 2.20 shows the FFT results for the filtered and unfiltered binary

sequences. All harmonics of the filtered sequence are more than 60 dB below the fundamental. The magnitude of the prime harmonic of the sequence is approximately 1.02, which is within 2% of unity. This indicates that if the binary sequence is made of voltage levels $\pm V_{seq}$, then the resulting sine wave will have a zero-to-peak amplitude within 2% of V_{seq}.

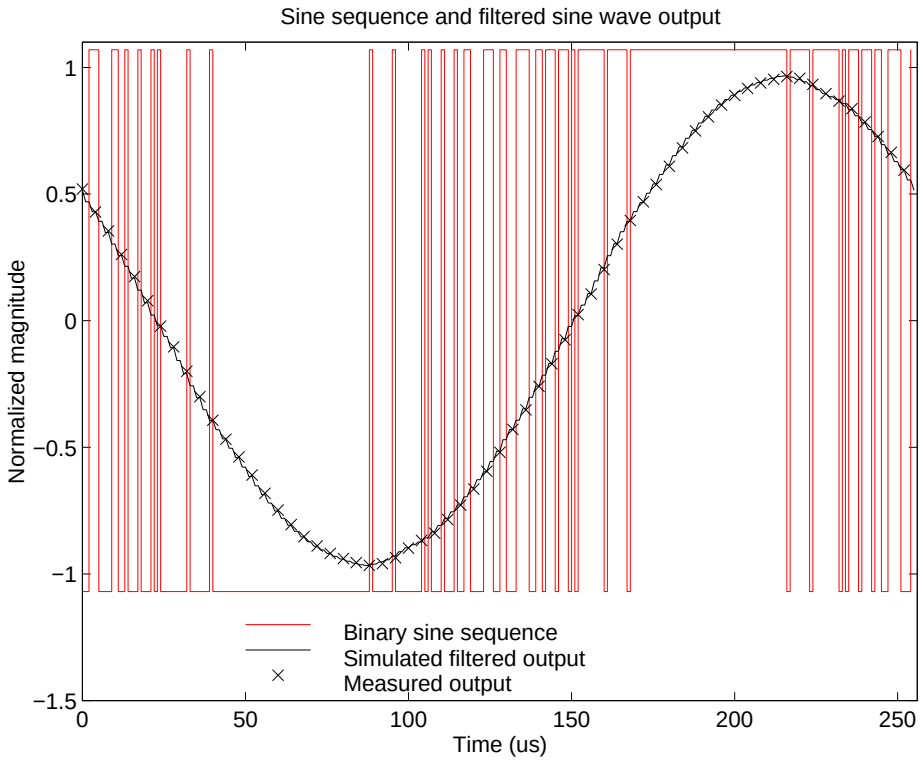

Figure 2.18: The oversampled sine sequence.

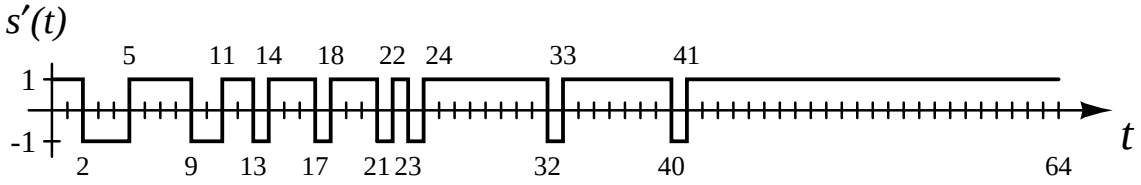

Figure 2.19: Optimized 64-bit oversampled sine sequence, first quadrant.

The advantages of using an oversampled modulation sequence rather than a simpler binary sequence can be appreciated by following comparison. To obtain the same 60 dB linearity performance with, say, a square wave modulator, a premultiplication filter G with much sharper rolloff such as a fourth-order Chebyshev or a 6th-order Butterworth would be required to compen-

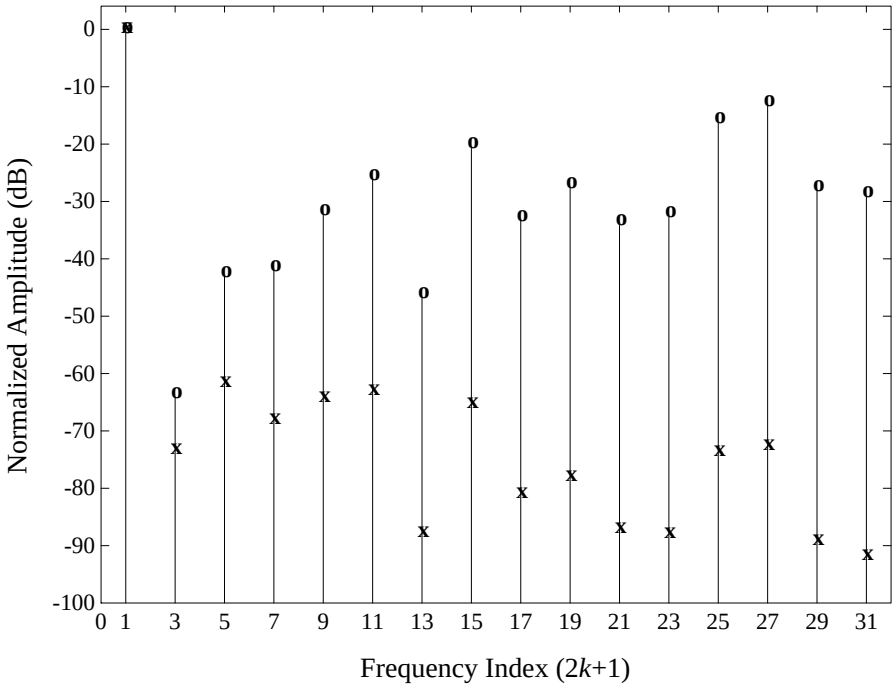

Figure 2.20: Frequency domain properties of the raw (**o**) and filtered (**x**) bit sequences.

sate for the sizable harmonics in the square wave. Such a filter would be more complicated to build than the simple cascade of single-pole filters in the above example, and would be more sensitive to mismatches in the implementation. On the other hand, the oversampled sequence is fairly easy to generate, as outlined below.

The first-quadrant sequence of Figure 2.19 consists of 11 zeros and 53 ones, as expected from the integral calculation in Equation 2.38. The asymmetry allows a simple implementation using a sparse address decoder. A binary counter counts from 0 to $r - 1$, where r is the length of one quarter of the full binary sequence. The address decoder, a wired-'or' implementation of nMOS transistors and a pMOS pullup device with a small layout footprint, generates the 11 'zero' bits at the proper points of the count. The inversion and reversing operations needed to obtain the rest of the full sequence can be elegantly realized by using a gray-code count rather than a binary count. In an n-bit gray code, as illustrated in Figure 2.21, the lower $n - 2$ bits describe a sequence which counts out forwards and then backwards. The inversion of the sequence is determined through an exclusive-or operation of the sequence bit with the n-th bit of the gray-code counter. It is also quite straightforward to generate the addresses of a sequence 90° out of phase with the original, for

complex modulation with both sine and cosine components, by performing inversion as above but using bit $n - 1$ of the counter. One technique is to use a sigma-delta modulator to transform an arbitrary signal into binary form [25, 26]. However, in the case of modulation the (carrier) signal is known *a priori*, so a sigma-delta modulator is unnecessary and can be replaced by a simple digital circuit which provides the multiplexer with a predetermined fixed sequence.

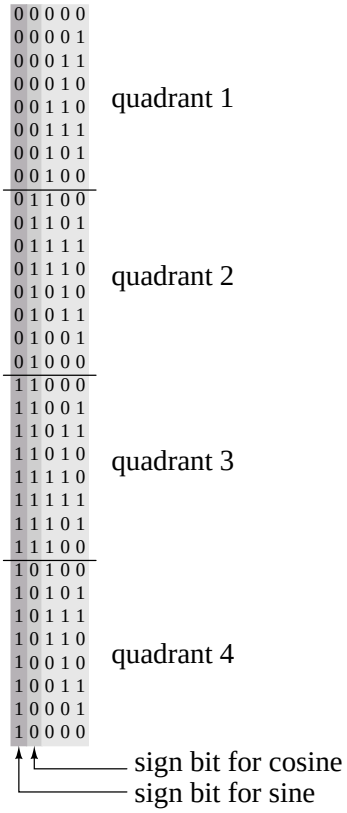

Figure 2.21: Use of Gray code to generate sine and cosine sequences.

2.19 Modulation Multiplier

The previous section described a method by which an accurate sine wave can be generated by filtering an oversampled binary sequence. Utilizing this technique immediately alleviates one of the two major problems of the analog wavelet processor. This section addresses the other problem, that of producing a linear multiplication of the input with the sinusoidal carrier signal.

Because implementation of the demodulation multiplication requires multiplication of an

arbitrary continuous-valued input with a *known* periodic function (the sine wave), we are able to use the oversampled sine sequence directly to achieve a highly linear analog multiplier.

Simple but practical modulation schemes make use of multiplication of a continuous-valued signal and a binary-valued signal. An exact multiplication of an arbitrary input by a binary-valued (± 1) function can be realized as a multiplexer which is controlled by the function and alternates between the input and its inverse. This is shown in Figure 2.22. In the figure, the multiplexer is

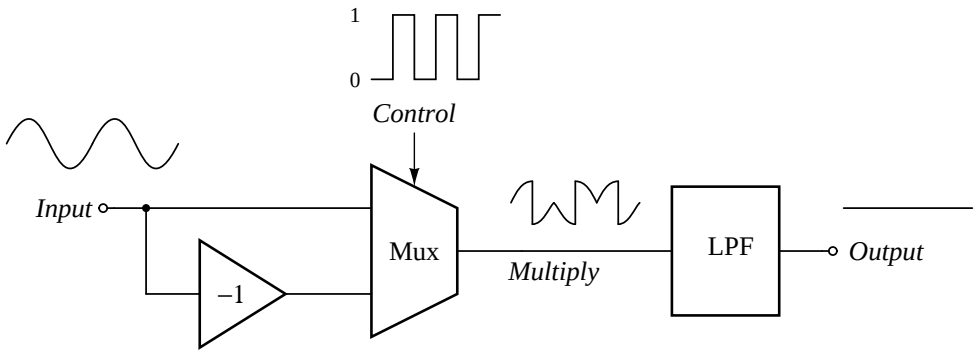

Figure 2.22: Multiplexing *vs.* Multiplying.

controlled by a square wave [28]. The square wave is used as an approximation to a sine wave with all harmonics other than the fundamental considered to be "error" terms. As mentioned in the previous section, this works as long as the intermodulation products of the input signal and these "error" harmonics fall well outside of the passband of the final result after the usual lowpass filtering.

In the same manner as the square wave example, any arbitrary binary sequence can be used as a control, insofar as any unwanted intermodulation products fall outside the passband of the postmultiplication filter. Since the purpose of oversampled representations of waveforms is to push unwanted harmonics as far away as possible from the harmonics of the desired waveform, binary-valued oversampled representations work extremely well in place of the square wave in Figure 2.22. It is assumed that the oversampling noise is negligible in the frequency band of interest, and can therefore be filtered from the multiplication output.

2.20 Switch-Cap Wavelet Gaussian Function

The circuit which approximates a Gaussian filter is based on the same architecture as described in Section 2.10 (shown in Figure 2.11), which is in turn based on the Central Limit Theorem of statistics: A half-Gaussian profile is produced by an infinite cascade of lowpass filters.

A finite cascade of simple filters will approximate the Gaussian transfer function to the degree of accuracy required, provided that there is enough space in the VLSI layout to accommodate the number of filters required.

Once it was determined that the output of the modulation multiplication would be a synchronous discrete-time signal, the decision was made to change the continuous time filters into switched capacitor filters. This is done for three reasons:

- The discrete-time system scales correctly to any clock speed, so the system can be run accurately either with real-time inputs or with non-real-time inputs, such as test inputs generated by a computer. Non-real-time inputs do not even need to be strictly synchronous.
- The switched capacitor architecture has much larger linear input and output ranges than the transconductance-C filters used in the analog architecture.
- The bandwidth is controlled digitally and so can be computed directly from the system clock driving the center-frequency oscillators, as opposed to being an independently-controlled variable.

As with the continuous-time filter architecture, it is only necessary to create one half of a Gaussian centered around zero. The same argument relating to the Central Limit Theorem applies, so the switched capacitor filter retains the architecture of a set of cascaded first-order lowpass sections. Figure (2.23) shows a section of the filter we designed. This circuit is an RC lowpass filter using a simple switched capacitor simulated resistor [27]. The filter design is a discrete-time circuit and directly implements the (z-transformed) lowpass function through distribution of charge:

$$\beta = \left(\frac{1}{1+\alpha} \right), \tag{2.45}$$

$$V_{out} = \beta V_{in} z^{-\frac{1}{2}} + (1-\beta)\, V_{out} z^{-1} \tag{2.46}$$

where in the figure ϕ_1 and ϕ_2 are nonoverlapping clock signals of period T which maps to the z-domain unit delay. This circuit maps to an equivalent (s-domain) RC lowpass filter by $RC = \alpha T$.

The Gaussian lowpass filter consists of a cascade of n sections, which have a combined transfer function which converges to a Gaussian in the limit $n \to \infty$. The choice of the number of sections is a tradeoff between the accuracy of the filter with respect to a true Gaussian and the delay (and also noise) incurred by a signal passing through the cascade. We chose a cascade of eight

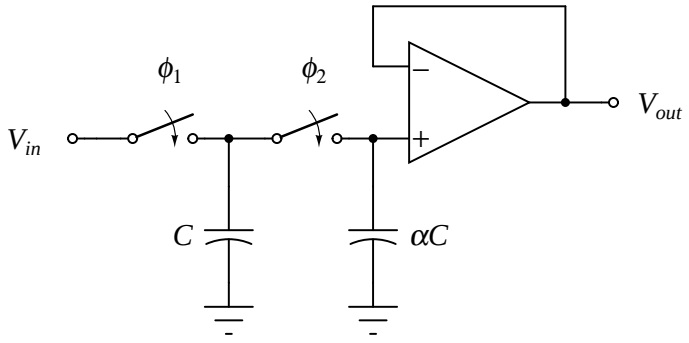

Figure 2.23: Single discrete-time lowpass filter section.

sections due to a primary constraint of accuracy. For eight stages, the resulting transfer function matches a true Gaussian to about −60 dB.

The center frequencies of the effective bandpass functions are determined by the frequency of the modulator sine waves (described below), which are spaced on a $\log_2$ scale. The bandwidth of each wavelet filter is determined by the cutoff frequency of the switched capacitor lowpass filter, and therefore is proportional to the period of its driving two-phase clock. The frequency of this clock signal, like the clock which generates the oversampled carrier signal, is spaced on a $\log_2$ scale. The fixed relationship between the carrier and the lowpass cutoff gives the wavelet filterbank a constant-Q characteristic. The clock frequency can be derived from the clock driving the modulator sequence generator (in our implementation, this is accomplished by dividing down the master clock off-chip). The bandwidths typically are set so that adjacent channels overlap at the half-magnitude point of the Gaussian function.

2.21 Output Time Multiplexing

Outputs of the Gaussian filter sections should be sampled in a manner which generates the time-frequency distribution shown in Figure 2.1. The binary-tree time multiplexing shown is quite easy to generate from a Gray code. As shown in Figure 2.24, control signals for taking the output from each wavelet channel can be achieved by a counter which uses two-phase clocking and generates Gray code as output. This control signal is the trigger signal, ϕ_2, for each Gray code digit Q_i. It can be seen that if ϕ_{21} controls sampling of Channel 1, ϕ_{22} controls sampling of Channel 2, *etc.*, then Channel 1 is sampled twice as often as Channel 2, which is sampled twice as often as Channel 3, and so forth, and none of the signals overlap. One time slot per sequence of outputs is

unused, but could be used to encode the DC level of the input. Note that a Gray code counter is as simple to design as a binary counter. In the usual implementation of a binary counter as a cascade of toggle flip-flops where each toggle flip-flop is a pair of transparent latches in series, the output of the first latch in each pair encodes a Gray code while the output of the second latch in each pair encodes the binary sequence.

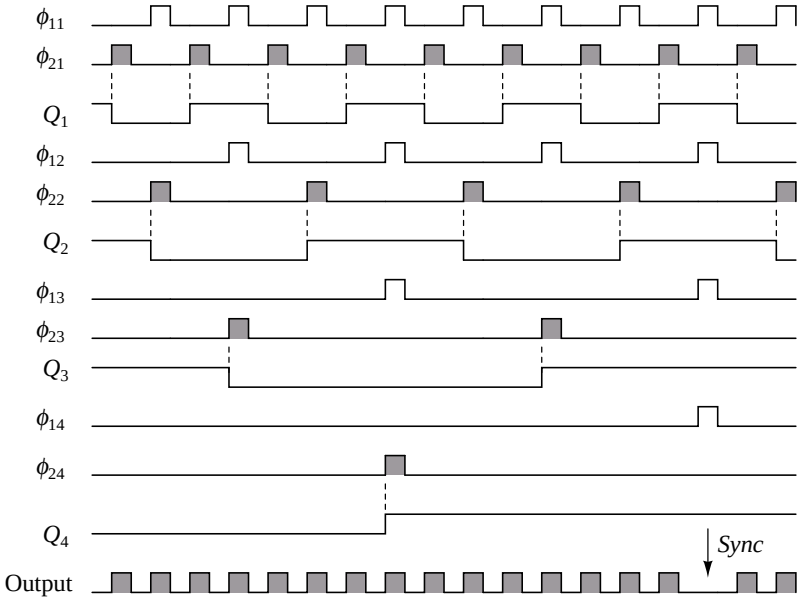

Figure 2.24: A scheme for controlling time-multiplexing of the outputs.

The same method is used throughout the chip to generate clocks which ensure that the modulator frequencies and filter bandwidths differ by a factor of 2 for each channel. An additional bonus of using Gray code is that digital noise is kept low because only one channel is clocked at a time, and power consumption is kept to a minimum due to the event-driven nature of the process.

2.22 Wavelet chip slice

The parallel nature of the analog wavelet computation makes the chip easy to create using abutting slices of circuitry. We built a bank of eight slices (channels), each containing the logic to divide down the incoming clock signals, generate the sine and cosine sequences, multiply these modulating signals with the chip inputs, filter the result with a Gaussian-shaped function, and time-multiplex the outputs on two buses (one for sine, one for cosine parts). I devised a simple way to expand the system from eight to sixteen channels covering the same total frequency span: In a

sixteen channel system, each channel has a center frequency value which is $\sqrt{2}$ of the neighboring channel, with the bandwidth of each channel narrowed by a factor of two to account for the fact that there are twice as many channels squeezed into the same total frequency span. The value of $\sqrt{2} = 1.41421\ldots$ is approximated reasonably well by the integer fraction $7/5 = 1.4$, which is only about 1.0% low. The sixteen-channel system is enabled by starting with a master clock which is first divided down by five and by seven, each result becoming the master clock for one eight-channel section. The outputs of each eight-channel section are interleaved, and the sampling scheme of Figure 2.1 must be modified so that the timesteps are halved so as to allow both sections to be sampled. The sixteen-channel architecture is shown in Figure 2.25. Wavelet decompositions of any multiple of eight channels can be realized in a similar manner.

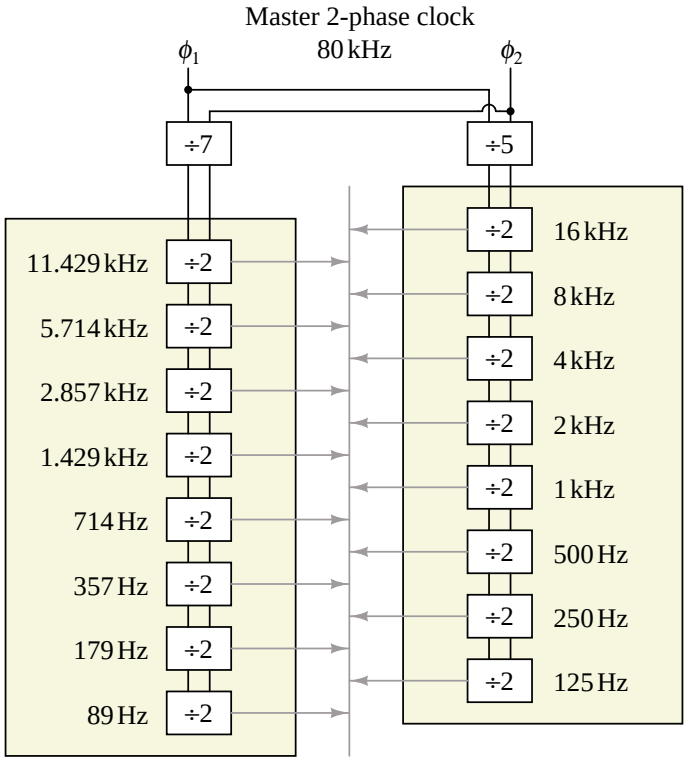

Figure 2.25: Sixteen-channel architecture using the 7-to-5 frequency ratio.

Decomposition and reconstruction functions are similar and therefore are able to share the same multiplier and filter circuitry, and the chip can be configured for either function. Reconstruction requires sample and hold circuitry on the front end to demultiplex the sine and cosine inputs, and a capacitive adder at the output. During reconstruction, the Gaussian filters do not shape the

signal but are used only to reject the high-frequency noise output of the multiplier. A block diagram showing two channels of the Wavelet Transform chip is shown in Figure 2.26.

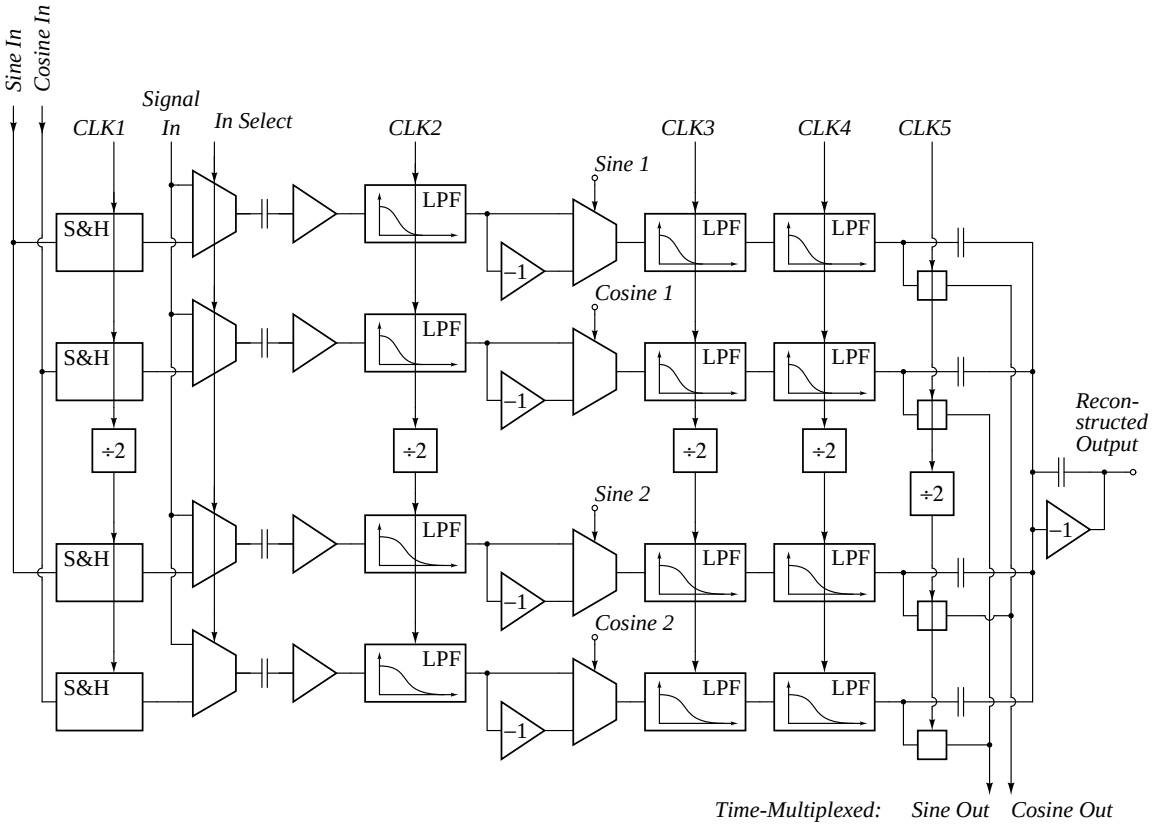

Figure 2.26: The wavelet chip, block diagram.

The circuit layout of the CWT processor fits two sections of eight channels each on a single 4 mm×6 mm die in a 2 µm CMOS p-well process, packaged in an 84-pin PGA. Figure 2.27 is a photomicrograph of the integrated circuit. Test results reported here are from this chip and a separate test chip containing a single channel.

2.23 Experimental Results

In a test of the modulation multiplication (including the postmultiplication filter), the sinusoid modulator is multiplied by a constant input. A binary sequence representing the oversampled sinusoid is produced at the multiplexer output, and is smoothed into a sine wave by the lowpass filter. Figure 2.20 shows the FFT of the test chip modulation output before ("**o**") and after ("**x**")

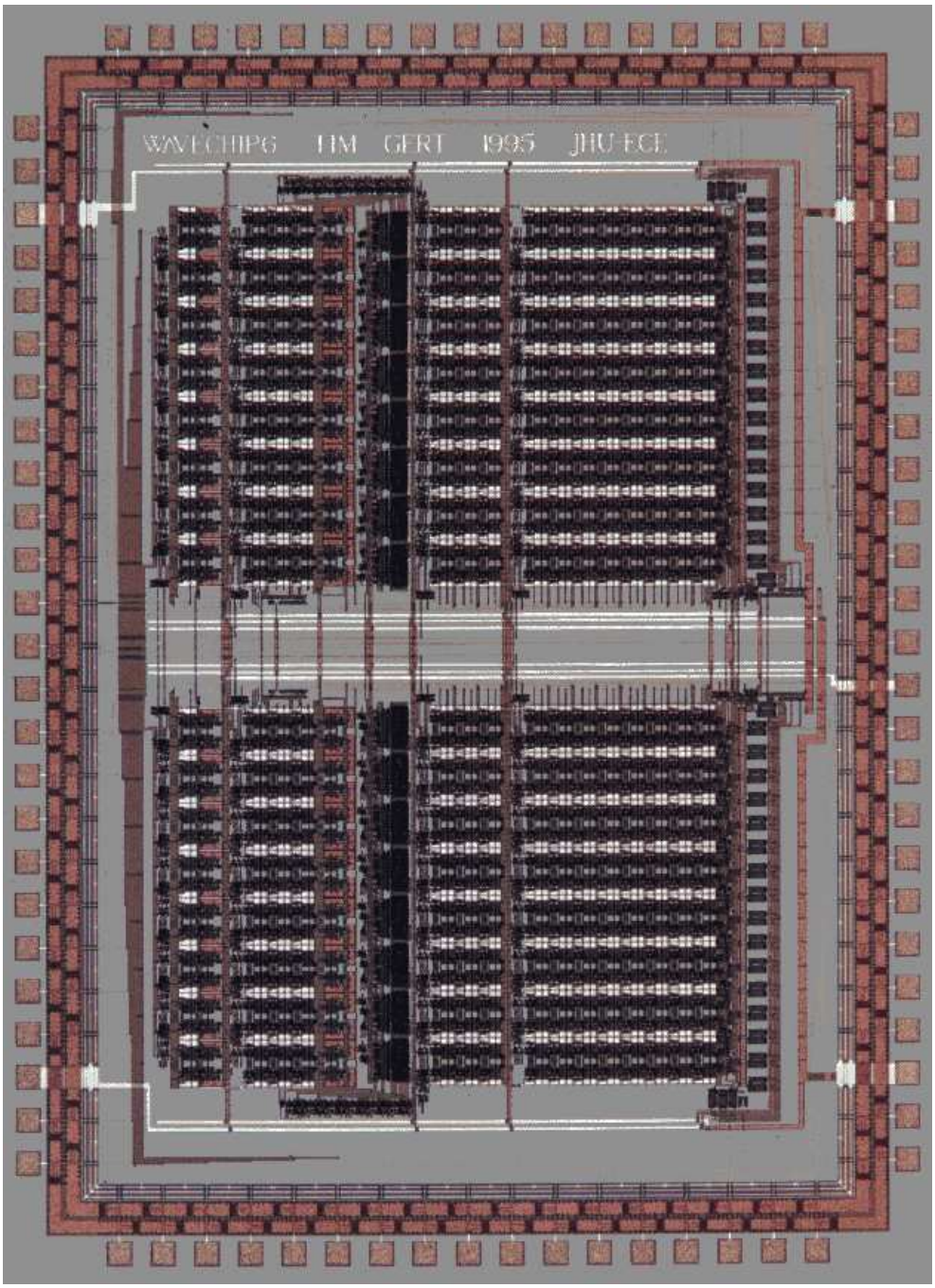

Figure 2.27: Photomicrograph of the mixed-mode continuous wavelet transform processor, a $4\,\text{mm} \times 6\,\text{mm}$ die size fabricated in a $2\,\mu\text{m}$ CMOS p-well process.

lowpass filtering. Distortion components of the binary sequence are attenuated by the filter to below -60 dB.

Figure 2.28 is an oscilloscope photograph showing the demodulating hardware in action. The top trace is the input signal, a sine wave generated by a function generator for the purposes of evaluating the system. The middle trace is the result of multiplying the input by the binary sequence. The function flips rapidly from positive to negative voltages, which is too fast to be captured in the oscilloscope photograph where it appears to be two overlapping sinusoids. The bottom trace is the output after filtering through the Gaussian filters, and is a sinusoid of a frequency which is the difference between the input and the carrier. The bottom trace is jagged due to the discrete-time nature of the switched capacitor hardware. Because the output is sampled at the same rate prior to time-division multiplexing into the output stream, there is no need at this point to apply a smoothing filter to the output.

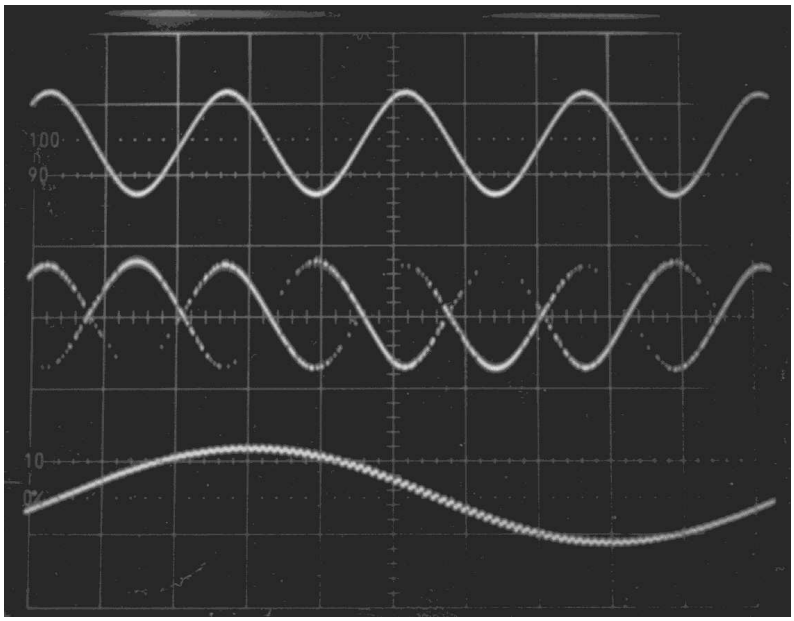

Figure 2.28: Signal demodulation using the wavelet chip. The top trace is the input signal. The middle trace is the input multiplied by the binary sequence. The bottom trace is the output after filtering.

Figures 2.29 and 2.30 show the measured transfer function (magnitude and phase) of the Gaussian filter, a cascade of eight single-pole switched capacitor filters, as compared to the predicted result for an ideal eight-stage lowpass filter cascade. Bandwidth is normalized to the clock frequency of the filter switches, showing that the shape of the filter is independent of the corner frequency.

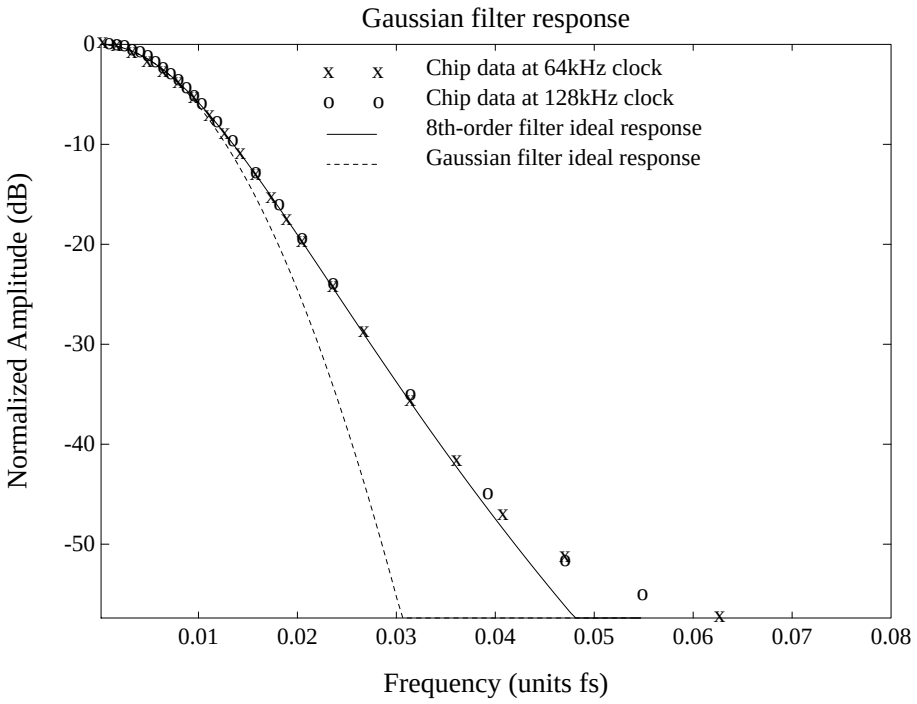

Figure 2.29: Gaussian filter magnitude response: predicted and measured.

The mismatch between channels of a sixteen-channel wavelet decomposition (the setup of which is described in the previous section) was tested by measuring the peak-to-peak magnitude of the output of each channel *vs.* the system input frequency. This measurement clearly shows the bandpass nature of the wavelet channels. The response of the first eight channels of a sixteen-channel wavelet output (channel frequencies spaced on a $\log\sqrt{2}$ scale) to a single sine wave input of variable frequency is plotted in Figure 2.31, along with the theoretical response (solid lines) in which the Gaussian function is approximated by cascaded lowpass filters as it is on the chip. The center frequency of each channel is exact, as it must be by design; the mismatch between filter bandwidths is negligible, and the mismatch in amplitude is acceptable.

2.24 Extensions of the Research

It is possible to apply the same complex demodulation scheme in a system designed to create an *arbitrary* mapping of the time-frequency plane, which unlike those depicted in Figure 1.1, has no regular structure, but a structure determined by the instantaneous information content of the

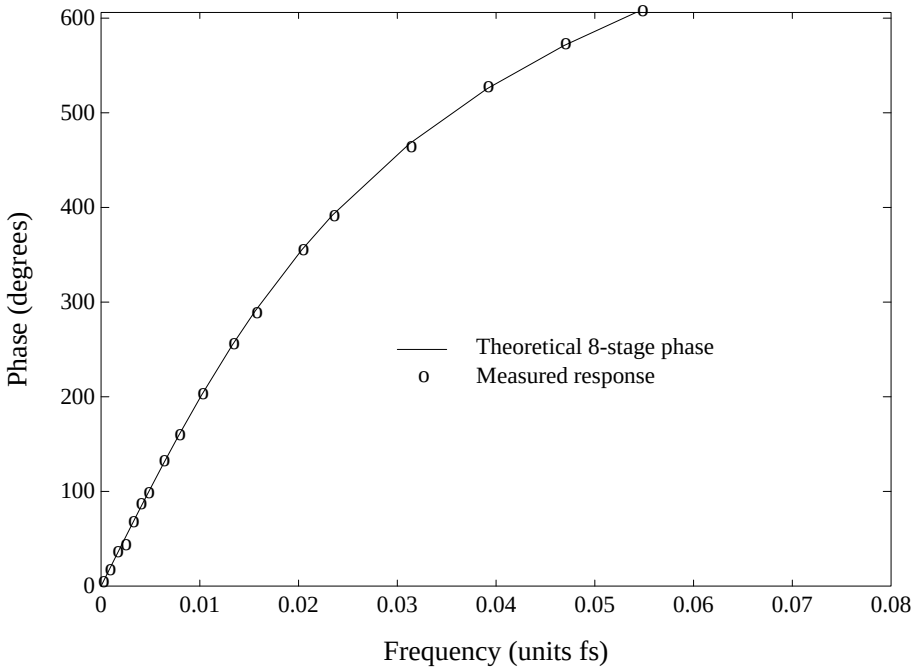

Figure 2.30: Gaussian filter phase response: predicted and measured.

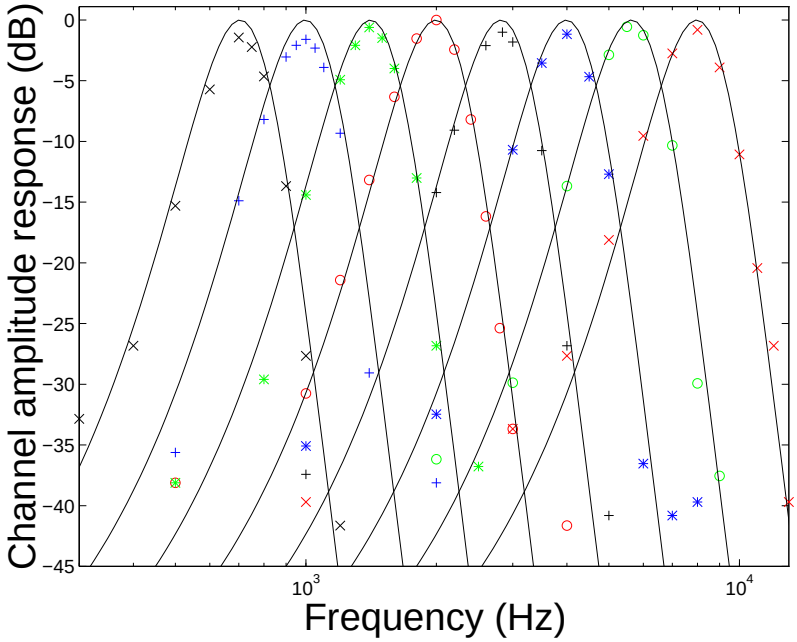

Figure 2.31: Wavelet responses to isolated sine wave input.

input signal itself. One such possible instantaneously computed map is depicted in Figure 2.32. The purpose of such a mapping is to optimally isolate the information content of the input signal in order to minimize the amount of bandwidth at the output. For example, if the input signal were a pure tone, then the output would be restricted to a single filter of minimum bandwidth and maximum timespan, sampled appropriately. Although algorithms to compute the optimal arbitrary coverage of the time-frequency plane are difficult to derive (formulas in [15] are nonintuitive and not recognizably amenable to hardware implementation), a system could potentially be designed based on heuristic methods.

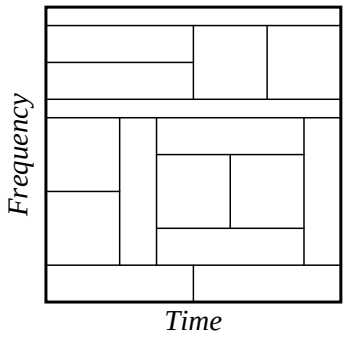

Figure 2.32: An arbitrary tiling of the time-frequency plane.

Another possible extension of the work consists of exploring algorithms which can make use of the demodulated outputs for acoustic pattern (speech/sound) recognition applications. One benefit of the wavelet processor *vs*. a standard filterbank approach normally used for speech recognition applications is that the wavelet output preserves phase information in the orthogonal sine and cosine outputs. Phase information does play an important role in some speech events and many other kinds of acoustic events (such as music and sonar), and may be used to the advantage of recognition systems. Another benefit, when the wavelet sampling scheme is used, is that the communication bandwidth required between the filterbank and the recognition system is greatly reduced, unlike standard filterbanks in which all channels are sampled at the same rate or are not demodulated to reduce the total required bandwidth for sampling. In any situation in which the frontend processor cannot be close to the recognition system (or other backend processor), this method constitutes an increase in efficiency and reduction in power in signal transmission.

2.25 Summary

There is an interest in the signal processing community for hardware to efficiently perform the Continuous Wavelet Transform for applications such as speech recognition, sound processing, and data compression. The Wavelet Chip described herein is able to compute both the CWT and its inverse (wavelet decomposition and reconstruction) in hardware, generating analog, discrete-time outputs. It is built using a mixed analog/digital architecture which is small and energy-efficient when compared to possible all-digital implementations. We show that our system computes a highly linear analog multiplication with a sine wave of low harmonic distortion by using oversampling techniques to implement the modulation multiplication. We utilize the Central Limit Theorem in order to generate a Gaussian-shaped filter from a cascade of lowpass filter sections. An additional benefit of the mixed analog/digital approach is precise control over the center frequencies and bandwidths of the channels. The output of the transform is realized by time-multiplexing the multiresolution outputs of the filters in an efficient manner. Experimental results on a sixteen-channel prototype have demonstrated the effectiveness of the new architecture.

The continuous wavelet processor architecture is perhaps given more significance by ignoring the "wavelet" aspect of it altogether. That is, there are uses for the architecture beyond the decomposition and reconstruction of signals for compression, frequency shifting, and so forth. The architecture represents a novel method of modulation and demodulation which can be used, to give an instance, for simple but effective lock-in amplification (extremely high-Q bandpass filtering). The oversampling method of sine wave generation can be used for precision function generation, or even a form of parameterized audio synthesis. The use of oversampling methods in this way to circumvent limitations of analog multiplication is a method that could bear considerably more scrutiny and has a wide range of potential applications.