Chapter
9

# A High-Speed Data Converter

The majority of the data converter topologies we've discussed up to this point have traded off time for resolution (so the signal bandwidth is generally much smaller than the clocking frequency). In other words, to get higher signal-to-noise ratios, SNRs, (wider output word size, $N$) we've averaged (filtered) the output of a noise-shaping modulator to remove the modulation noise. The averaging (again, filtering) has the undesired effect of reducing the noise-shaping modulator's allowable signal bandwidth. In this chapter we turn our attention towards high-speed topologies. Our approach is based on the material covered in Sec. 6.2.3 where $K$-paths are used but with switched-capacitor, SC, circuits used for the required feedback subtraction from the input (delta) and integration (sigma).

## 9.1 The Topology

Our high-speed topology, Fig. 6.24, consists of a single integrator (or resonator for a bandpass topology) that sums the difference between the input signal and the signals from the $K$ feedback paths. Let's start out this section by discussing the clock signals.

### 9.1.1 Clock Signals

Examine the clock signals seen in Fig. 9.1. Notice that the last four clock signals in this figure can be generated by simply inverting the first four clock signals (so for $K$-paths we need $K/2$ phases of a clock signal at $f_s$). By sampling an input waveform on the rising (or falling) edges of these clock signals we get an effective sampling frequency of

$$f_{s,new} = K_{path} \cdot f_s \tag{9.1}$$

*The key thing to note is that the frequency of the clock signal, $f_s$ , doesn't set the new sampling frequency.* Rather the delay between rising edges sets the effective sampling frequency, $f_{s,new}$. Why is this important? If we generate these clock signals using a ring oscillator, for example, made with inverters having delays of 10 ps then the new, or effective, sampling frequency is 100 GHz! If our desired signal bandwidth is 100 MHz then we are oversampling 1000 to 1. The frequency $f_s$ is selected to allow the circuits in each path to have enough time to respond (settle). Using the ring oscillator example, this consideration (path settling time) tells us how many inverters we need, the number of paths in the converter, and thus the ring oscillator's oscillation frequency $f_s$.

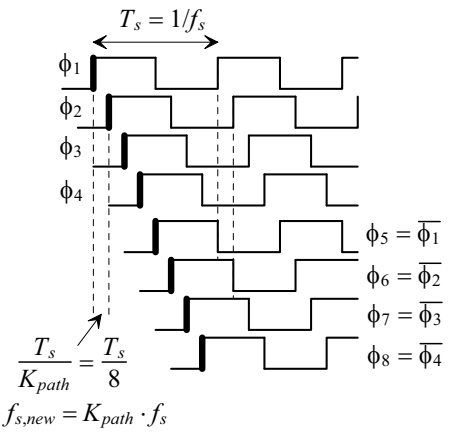



**Figure 9.1** Showing the clock signals used for time-interleaved sampling and the high-speed topologies discussed in this chapter.

*Path Settling Time*

Figure 2.37 details the clock signals used in traditional multi-rate signal processing. These signals should be compared to the clock signals seen in Fig. 9.1. When processing digital signals that are clocked, for example, on the rising edge of the clock signal, the signals in Figs. 2.37 and 9.1 are equivalent. In an analog circuit, however, the signals seen in Fig. 9.1 allow the capacitors more time to be charged or discharged. This is illustrated in Fig. 9.2. *We need to be careful here.* When sampling the input signal we must ensure that the input signal can charge the sampling capacitors, using either clock signals, in $T_s/K_{path}$. If this isn't the case then we effectively filter our input signal reducing its amplitude (the benefit of this inherent filtering is a built-in anti-aliasing filter, AAF). The reward for using the wider clock signals is that we can adjust the feedback signal's timing to help keep the topology stable. It's also critically important for a precision feedback signal from the DAC. The fact that the rate the capacitors charge/discharge is highest right after the switches close can be used to ensure that a single path pushes the summing circuit in the right direction. However, if the "push" isn't significant enough (doesn't occur quickly enough before clocking the next path) the topology will oscillate. The summing circuit's signal, say the voltage across *C* in Fig. 6.24, will move around confused because of the varying, or conflicting, information fed back from each path in the converter (see Sec. 9.1.5 for further discussions).

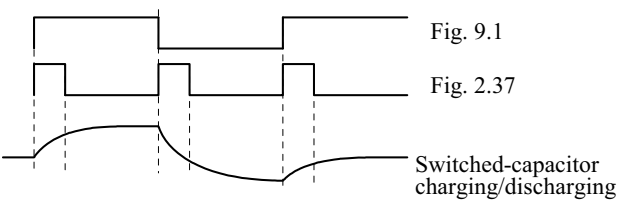



**Figure 9.2** Charging discharging a switched-capacitor.

### 9.1.2 Implementation

Figure 9.4 shows how we would implement a high-speed analog-to-digital topology, *a single integrator,* 1-*sigma, with K feedback paths, K-deltas*, (see Fig. 6.24) using switched-capacitors, Fig. 7.2, building blocks. The amplifier used in the active integrator has to respond very quickly, as described in the last chapter. Further note that the integration capacitor, $4C_i$, is selected based on an integrator gain of 0.25, again, discussed in detail in the last chapter. Also seen in this figure are the clock signals and the way the outputs from each path can be combined together. The outputs of the *K* paths can be summed to generate an output code ranging from 0000 (all outputs are low) to 1000 (all outputs are high). Summing the outputs together results in Sinc filtering with a transfer function of

$$H(z) = \frac{1 - z^{-8}}{1 - z^{-1}} \text{ with } f_{s,new} = K_{path} \cdot f_s \tag{9.2}$$

as seen in Fig. 9.3. Note that this is exactly the same response that we get when using the sample-and-hold, S/H, Fig. 2.17, clocked at $f_s$. While there are other ways to combine the path outputs let's do some examples.

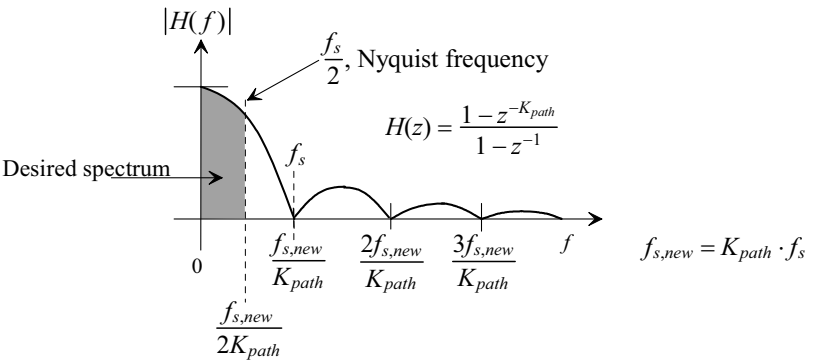



**Figure 9.3** Frequency response of the summing circuit (path filter) seen in Fig. 9.4.

**Example 9.1**
Simulate the operation of the data converter seen in Fig. 9.4 if $C_i$ = 100 fF and $f_s$ = 100 MHz. Show that an input frequency of 50 MHz results in, as seen in Figs. 9.3 and 2.17, attenuating the input by 3.9 dB (= 0.64).

Figure 9.5 shows the simulation results when the input is a 7 MHz sinusoid centered around 500 mV with an amplitude of 400 mV. Notice that the rate at which the data comes out of the summing circuit is 800 MHz (the output changes every 1.25 ns). Also notice that the change in the output is the reference voltage, here 1 V, divided by 8 or 125 mV.

Figure 9.6 shows the input and output if the frequency is increased to 50 MHz (the Nyquist rate if we were to decimate down to a clocking frequency of 100 MHz). We expect the amplitude of the output to be $0.64 \cdot 0.4 + 0.5$ or 756 mV. The simulation shows an output of 750 mV (remember the quantization noise). ■

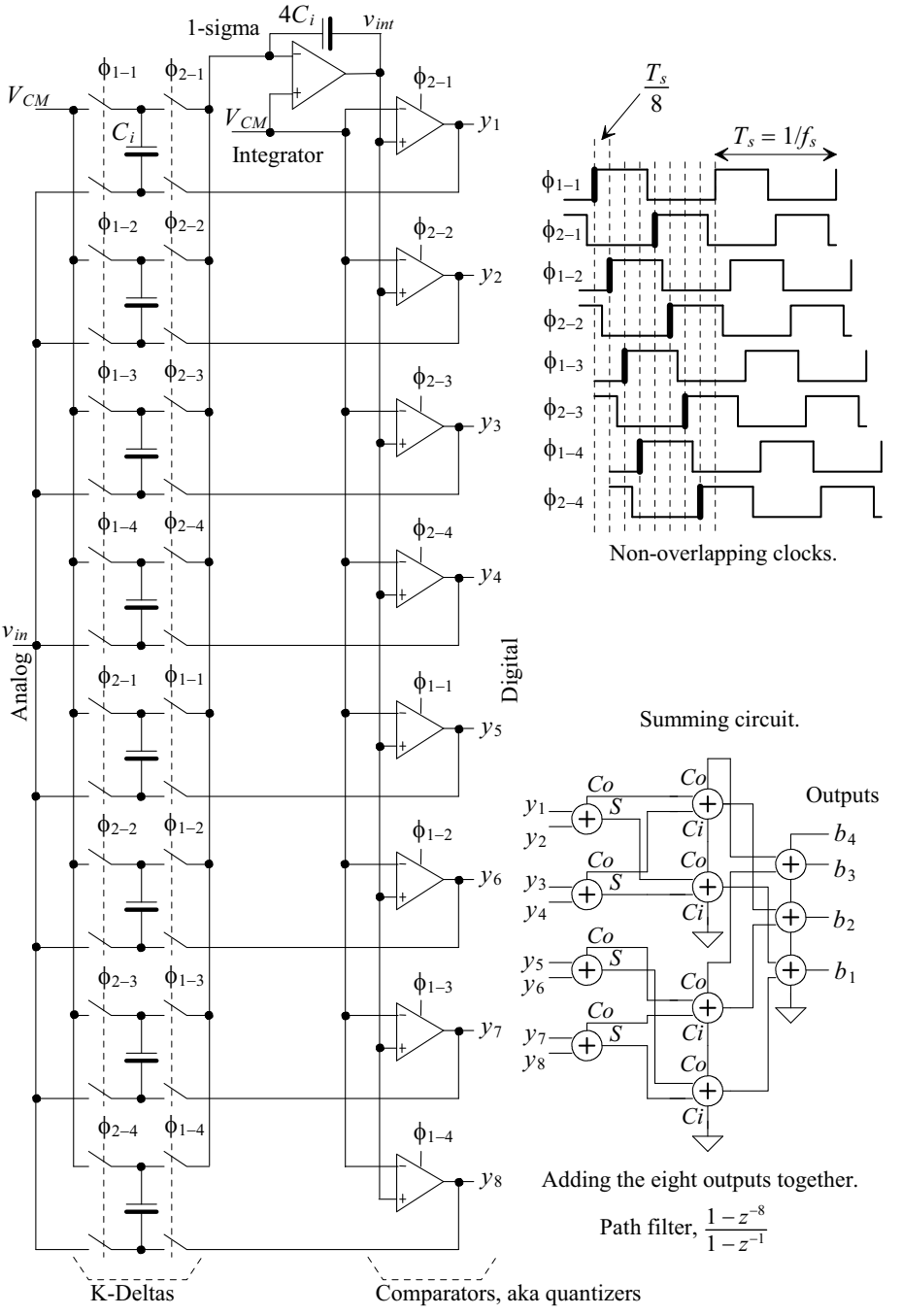



**Figure 9.4** A topology for high-speed data conversion using mixed-signal techniques, the K-delta-1-sigma topology.

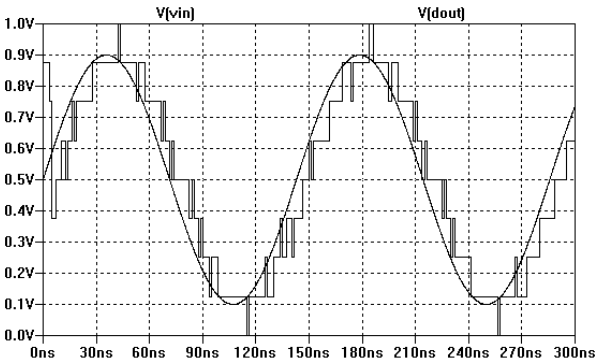



**Figure 9.5** Simulating the operation of the data converter in Fig. 9.4 with a 7 MHz input signal.

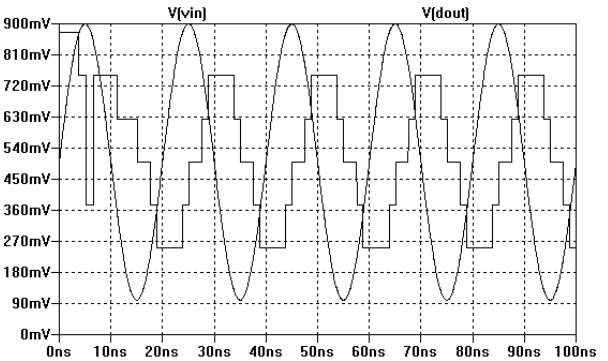
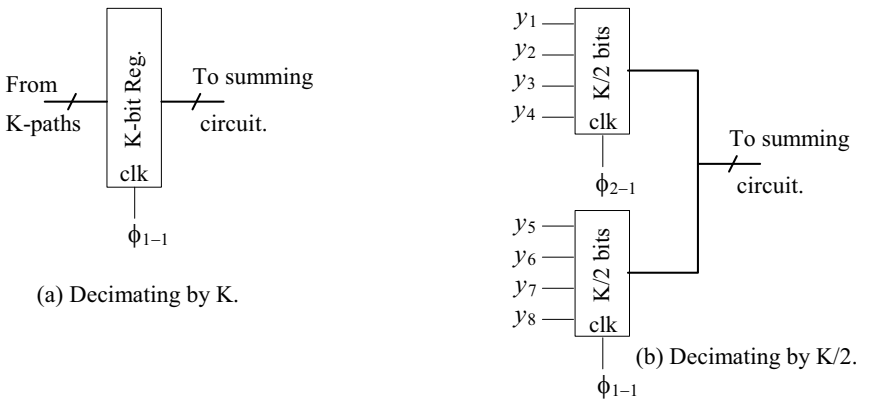



**Figure 9.6** Again, simulating the operation of the data converter in Fig. 9.4 but now with a 50 MHz input signal.

**Example 9.2**
Repeat Ex. 9.1 if the output is decimated down to the 100 MHz clock rate.

Figure 9.7a shows that adding a register in series with the $K$ outputs can be used to decimate the word rate down to $f_s$. A $K$-bit register can be placed in front of the summation circuit or a $\log_2 K$ wide register can be placed on the output of the summing circuit (not shown in Fig. 9.7). Timing mistakes in this digital part of the circuit are an important practical concern. The digital signals are moving quickly so eliminating the skew introduced by the summing circuit, by placing the register on the input side (Fig. 9.7a), is the approach we take here.

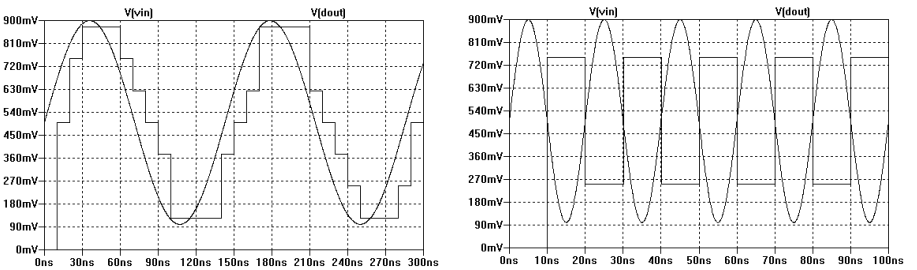
The simulation results are seen in Fig. 9.8. Notice that, when the input signal frequency is at the Nyquist rate in Fig. 9.8b, that the sampling instances repeat. Avoiding decimation (down-sampling), as can be seen when comparing Figs. 9.5, 9.6, and 9.8, and discussed in detail earlier in the book, helps keep the original signal intact.




(a) Decimating by K.

(b) Decimating by K/2.

**Figure 9.7** Adding a register to decimate the outputs of the high-speed modulator.




(a) 7 MHz input

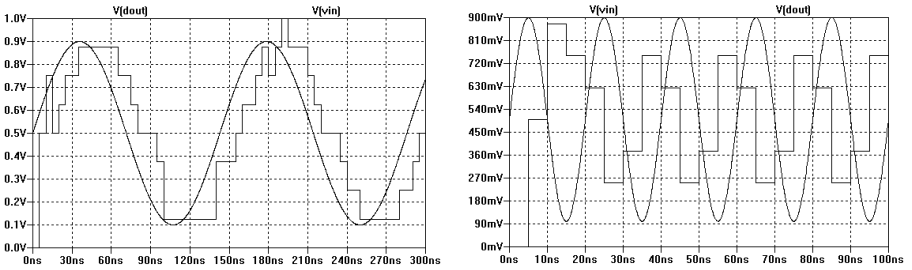
(b) 50 MHz input, note how the sampling times are repeating.

**Figure 9.8** Repeating Ex. 9.1 but with a decimation of 8.

In order to move towards reducing the unwanted effects of decimation (aliasing) and timing errors, examine the topology seen in Fig. 9.7b. Here we've split the word up into two paths (though we could also use 4 paths with the cost of tighter timing concerns). Summing the words together, we still get the filtering specified by Eq. (9.2). However, our decimated word rate is now $2f_s$. Figure 9.9 shows the simulation results using these techniques. The benefits should be obvious. ■

### 9.1.3 Filtering

Thinking about what we've done in this section we might come to the conclusion that the topology seen in Fig. 9.4 is nothing more than a flash ADC implemented with 8-comparators. We are only getting, effectively, 3-bits out of the topology (see previous examples) so why not simply use a 3-bit flash ADC? The simple answer to this is that averaging (read filtering) has little, Fig. 5.31, to no (for a DC input) effect on the resolution of a Nyquist-rate ADC while averaging can be used in a noise-shaping topology to increase the resolution, Fig. 7.7. We are repeating material presented earlier in the book so let's just do some examples to provide additional discussions.




(a) 7 MHz input (b) 50 MHz input

**Figure 9.9** Repeating Ex. 9.1 but with a decimation of 4.

*Example*s

Let's say that we want to design an ADC that is clocked with a 100 MHz clock and that outputs 8-bit words at least at a 100 MWord/s rate. In other words, we want to use mixed-signal circuit techniques to design the equivalent of a Nyquist-rate ADC (e.g., pipeline or flash) replacement (so the signal bandwidth ranges from DC to 50 MHz).

Let's use the topology seen in Fig. 9.4 as the basis for discussing this design. Generating the phases of the clocks is something we'll discuss later. Further we assume that the reader understands that selecting the size of the capacitors used in the topology is based on $kT/C$ noise, Table 2.1, considerations. We'll continue to use 100 fF capacitors in our topology with eight paths for an effective capacitance, for calculating input-referred noise, of 800 fF.

As mentioned at the beginning of the section, we must be able to charge the 100 fF capacitors with the input signal in $T_s/K$ seconds. For the present example $T_s/K$ = 1.25 ns so the switch resistance should be much less than 12.5 kΩ (very easy to ensure).

What this design comes down to is selection of both the appropriate decimation rate on the output of the $K$-path topology and the digital filter. Examine the block diagram seen in Fig. 9.10. Here we use the 8-path topology seen in Fig. 9.4 clocked at 100 MHz. Directly on the output of the modulator, which changes at a rate of 800 MHz, we decimate by 8 back down to 100 MHz, with the register, and add the outputs together. As indicated by Eq. (9.2) this lowpass filters the data. If our input is a sinusoid at 6.25 MHz centered around 500 mV with a peak amplitude of 400 mV, then we can estimate the attenuation of our signal from this addition (filtering) using

$$\frac{\sin \pi \cdot \frac{6.25}{100}}{\pi \cdot \frac{6.25}{100}} = 0.993 \tag{9.3}$$

The output of the summing circuit is a 4-bit word that ranges from 0000 (0 in decimal) to 1000 (8 in decimal). We pass the 4-bit output through two Sinc filters which increase the word size by 3-bits each for a total output word size of 10-bits. However, the word is not centered around the common-mode voltage so we can shift the word left to do a multiply by 2 and center the word (as discussed earlier). This reduces the word size to 9-bits, still more than required. The issue is that we also reduce the signal bandwidth. For our 6.25

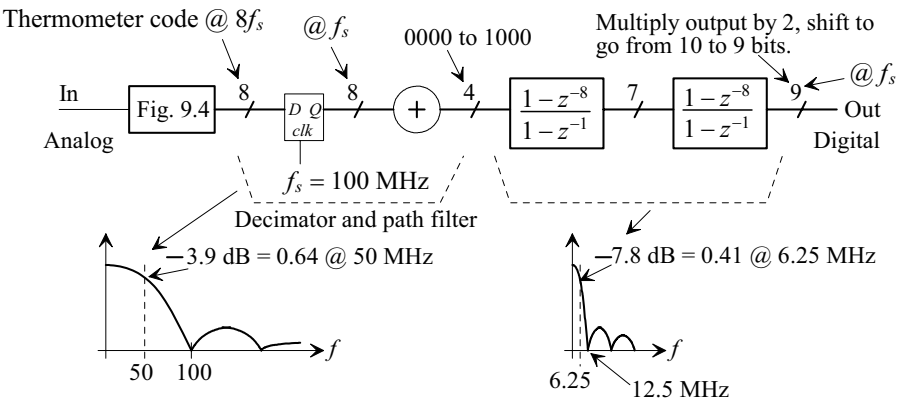



**Figure 9.10** Decimating and filtering the output of the K-path modulator, an approach that won't work for Nyquist-rate conversion.

MHz input signal, as seen in Fig. 9.10, this means that we attenuate the signal by 0.41. The total attenuation of our 6.25 MHz input signal is then 0.407 so the peak amplitude out of the filter is 663 mV (ideally it's 900 mV). Figure 9.11 shows simulation results verifying our hand calculations. Note the delay through the filter when starting.

Studying Fig. 9.10, we see that the decimation right at the beginning of the digital filter is limiting our bandwidth. Consider spreading the decimation out through the filter as seen in Fig. 9.12 (see also Sec. 4.2.5). Here we've put a register on the output of the summing circuit instead of on its input, Fig. 9.7a. For ideal signals we may not need this register; however, as mentioned earlier, eliminating skew and timing errors is important for proper data converter operation. Simulation results are seen in Fig. 9.13. The input signal undergoes less attenuation but the final resolution is 7-bits instead of 9-bits (indicating that there is less filtering in Fig. 9.12 than 9.10). In order to move towards our design requirements, an 8-bit converter with a signal bandwidth of 50 MHz that is clocked at 100 MHz, our final output clock rate will be increased to 200 MWords/s (instead of 100 MWords/s as in Figs. 9.10 and 9.12).

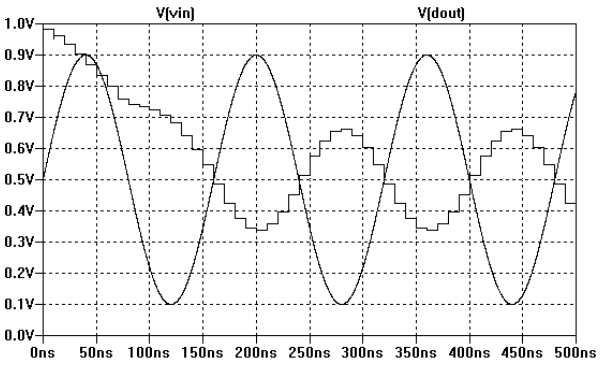



**Figure 9.11** Simulating Fig. 9.10 with a 6.25 MHz input signal.

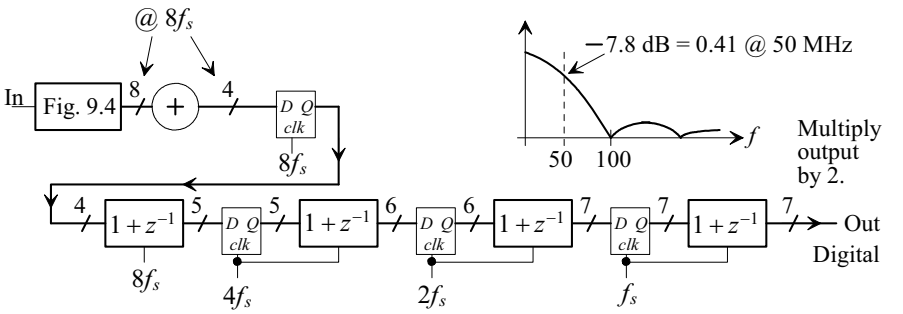



**Figure 9.12** Decimating through the filter, see Sec. 4.2.5.

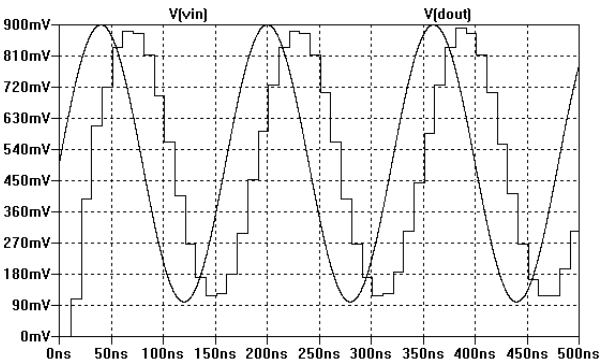



**Figure 9.13** Simulating Fig. 9.12 with a 6.25 MHz input signal.

Figure 9.14 shows our next attempt at filtering and decimation. The simulation results, with a 6.25 MHz input, are seen in Fig. 9.15. By adding together the 8 outputs in Fig. 9.4 we are still attenuating our input signal with a Sinc response. Note that the output of the adder is changing at 200 MWords/s. We don't employ decimation at any other place in the filter. The chain of averaging filters, on the adder's output, increases the word size by 1-bit per stage. The attenuation from this digital filter at 50 MHz is $(0.707)^5 = 0.177$, see Fig. 1.17. This, combined with the attenuation from the addition, results in an overall attenuation for a 50 MHz input signal of 0.113 (change the input signal frequency and simulate to verify).

Figure 9.16 shows yet another filter. This filter doesn't employ decimation so a 50 MHz input should see the −3.9 dB attenuation from adding the 8 outputs together, Fig. 9.10. In addition, as seen in Fig. 1.17 (but with frequencies on the x-axis scaled by 8 so the first zero point occurs at 400 MHz), the averaging stages will provide some additional, minor, attenuation, Fig. 9.17. The question we need to ask is "Does this filter remove the modulation noise in addition to increasing the word size?" If the answer is "No, the filter doesn't remove the modulation noise," then how do we meet our design goals? Further, how do we determine if the filter removes the modulation noise?

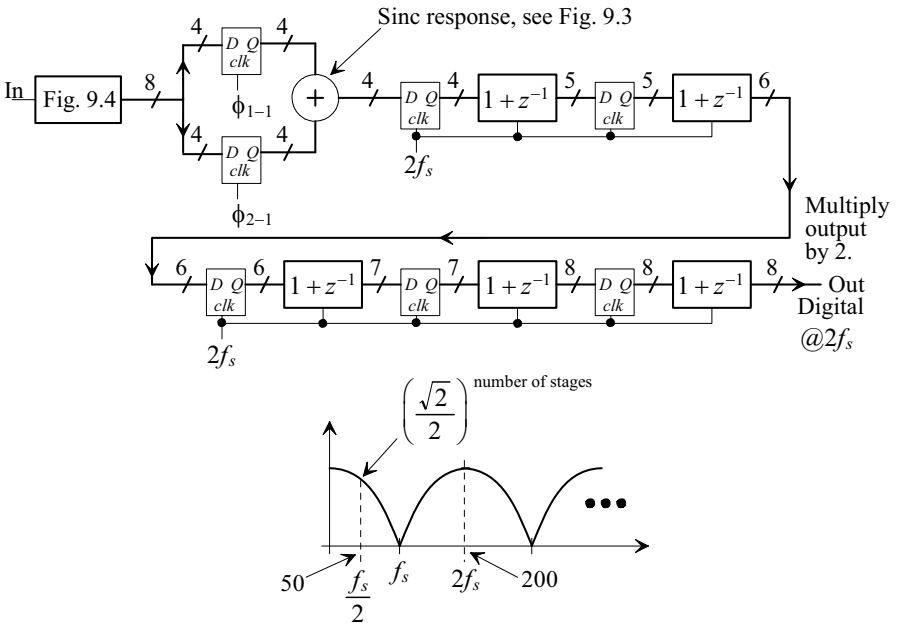



**Figure 9.14** Another filter; however, this topology has problems removing the modulation noise.

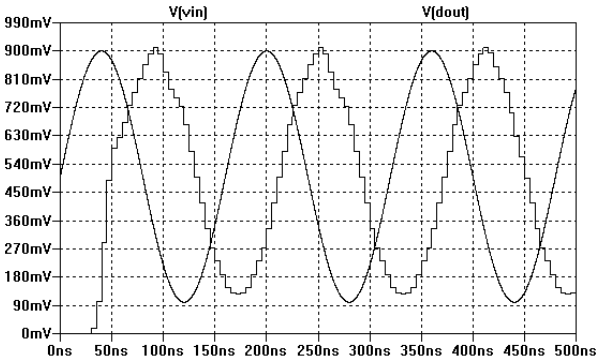



**Figure 9.15** Simulating Fig. 9.14 with a 6.25 MHz input signal. Note the the nonlinearity in the output.

In order to move towards answering these questions, notice the nonlinearity in both Figs. 9.15 and 9.17. We implemented the filter by focusing on only increasing the output word size to 8-bits. We didn't concern ourselves with removing the modulation noise and thus we see noise and nonlinearity in our output signal. Reviewing Sec. 7.1.3, we can remove the modulation noise by using the filter seen in Fig. 9.10 but without the decimation (noting that if $L = 2$ then only the path filter and one Sinc filter are used for a 7-bit output). This also helps us move closer towards the design goals. Figure 9.18 shows the simulation results when a 50 MHz input is applied to the topology seen in Fig. 9.10 without decimation. The input signal undergoes 3-Sinc filter responses $(0.64)^3 = 0.262$.

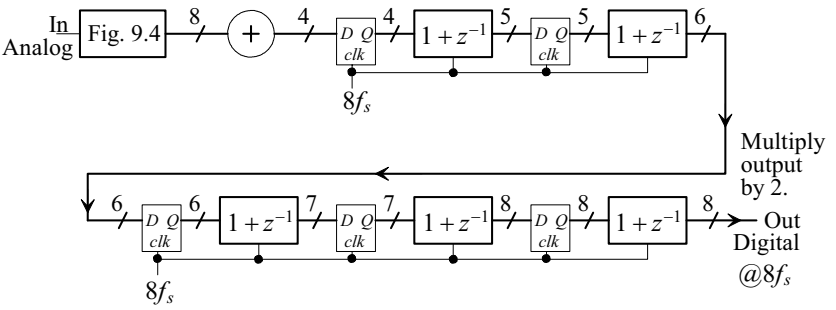



**Figure 9.16** Filtering without decimation. This filter won't do a good job removing the modulation noise but it does increase the output word size to 8-bits.

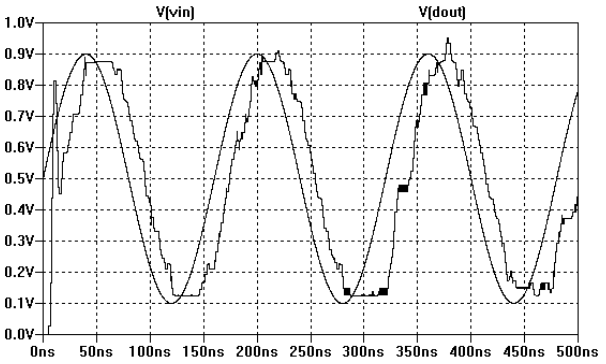



**Figure 9.17** Simulating Fig. 9.16 with a 6.25 MHz input signal. Again, note the the nonlinearity in the output.

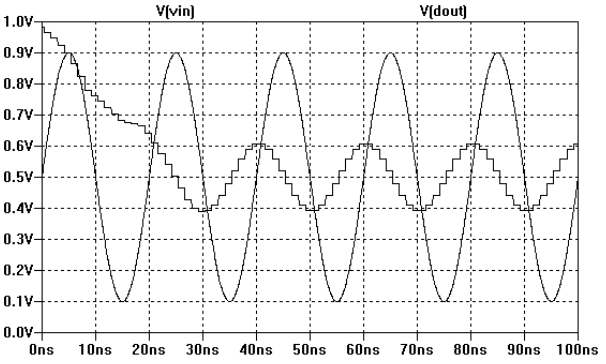



**Figure 9.18** Using the filter seen in Fig. 9.10 without decimation. Attenuation is due to 3-Sinc stages. The input signal frequency is 50 MHz.

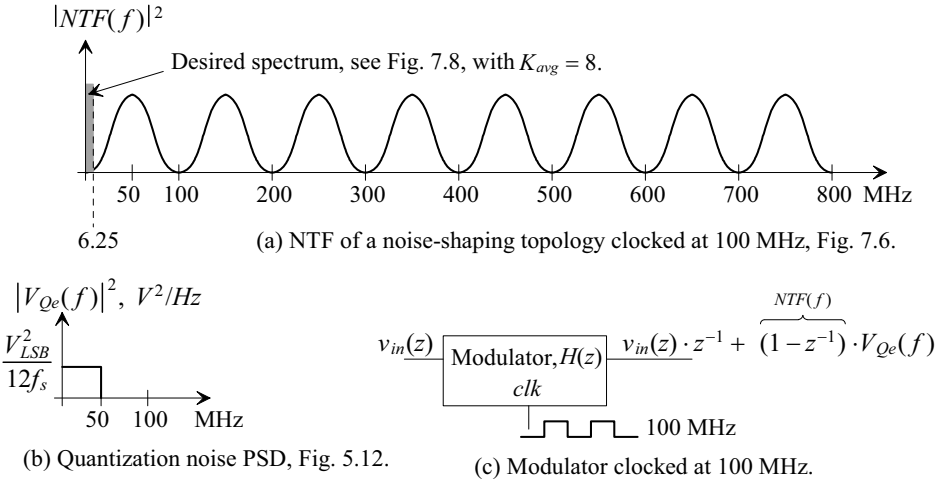
*Direction*

In order to move towards our design goals we can use the filter seen in Fig. 9.12 but with an additional averaging, $1+z^{-1}$, stage on the filter's output. This increases the word size to 8-bits. The penalty is an extra 3 dB attenuation at 50 MHz (so the total attenuation at 50 MHz is 10.8 dB or 0.288). The practical issue with this topology, for very high-speed operation, is avoiding timing errors in the stages clocked at $8f_s$. For a general design, we'll use the topology seen in Fig. 9.10. Timing errors are straightforward to minimize. The drawback is the reduction in signal bandwidth.

An area that should be investigated further is the use of biquad filters for removing the modulation noise, as discussed in Sec. 4.3.4.

### 9.1.4 Discussion

If the NS modulator seen in Fig. 7.2 is clocked at 100 MHz we get the noise transfer function, *NTF*, shape seen in Fig. 9.19a (out to 800 MHz or $8f_s$). While we are used to focusing on the spectral content between DC and $f_s$ (= 100 MHz here), we know that any discrete-time system's frequency response repeats with the sampling frequency, $f_s$. The signal bandwidth, as seen in the figure, is 6.25 MHz, assuming $K_{avg}$ = 8. Note that we are not showing the frequency response of the digital filter used for removing the modulation noise (which is where $K_{avg}$ is used, Eq. [6.47] and Fig. 7.8). Figure 9.19b shows the assumed spectrum of the quantization noise while Fig. 9.19c shows the single path clocked at 100 MHz (noting we are not showing the non-overlapping clock signals).



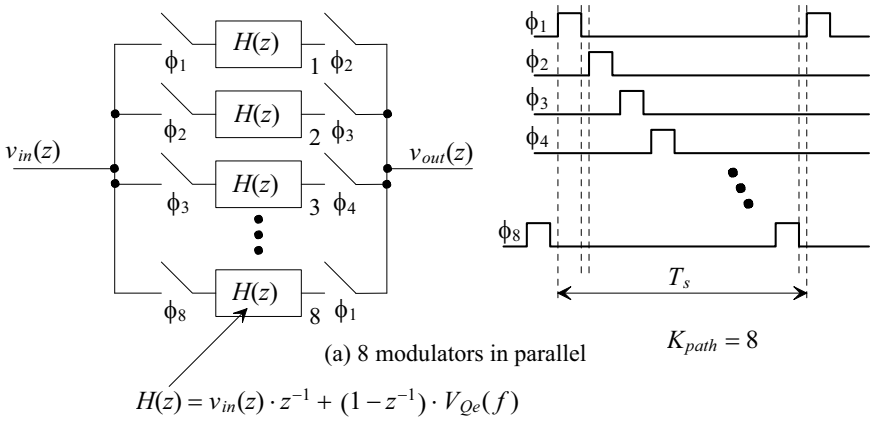

(a) NTF of a noise-shaping topology clocked at 100 MHz, Fig. 7.6.

(b) Quantization noise PSD, Fig. 5.12.

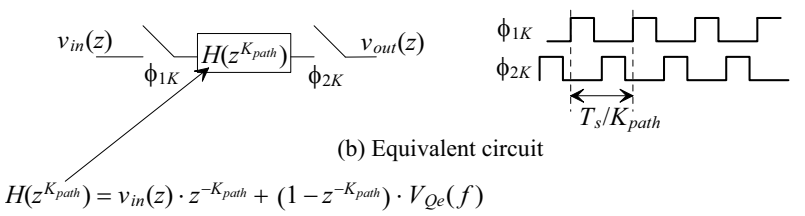
(c) Modulator clocked at 100 MHz.

**Figure 9.19** NTF and quantization noise spectrums of a first-order NS modulator clocked at 100 MHz with $K_{avg}$ = 8.

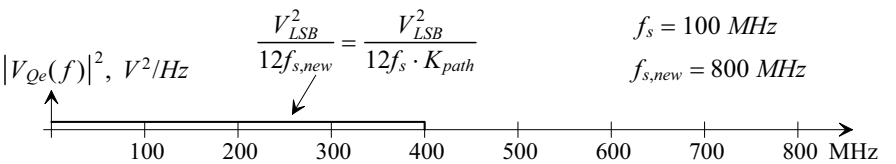
Next, examine the 8-path topology of modulators seen in Fig. 9.20 (modified from Fig. 2.37 for this discussion). In (a) of the figure each modulator is clocked at $f_s$, the same as in Fig. 9.19c, but with time-shifted clock signals. In (b) we show the equivalent circuit, see discussion on page 224 for details, noting the equivalent circuit's output changes with a frequency of $K_{path}\cdot f_s$. By using an effectively higher clock frequency we spread the



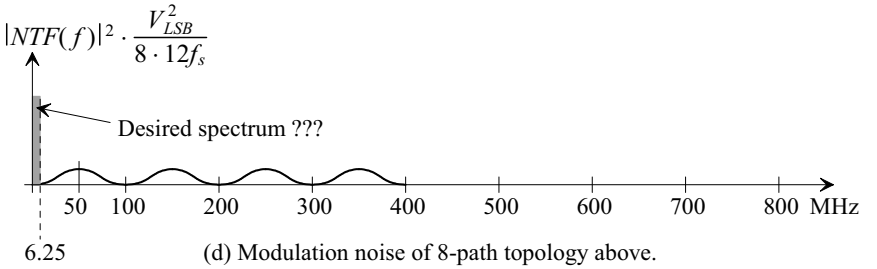

(a) 8 modulators in parallel




(b) Equivalent circuit




(c) Quantization noise for an 8-path topology clocked 100 MHz, compare to Fig. 9.19b.




(d) Modulation noise of 8-path topology above.

**Figure 9.20** Eight modulators in parallel (a time-interleaved topology).

quantization noise out over a wider range of frequencies, (c). This reduces the modulation noise added to our signal in the bandwidth from DC to 6.25 MHz as seen in Fig. 9.20d. If our desired signal bandwidth remains at 6.25 MHz then we will clearly get an improvement in SNR. Note that the output of the topology is 1-bit changing at a rate of 800 MHz.

Suppose we want to use this topology for high-speed data conversion where we use the fact that our effective output clock frequency is 800 MHz (and so the Nyquist frequency is 400 MHz). Is this possible or does it make sense? This is a good point to **remember the fundamentals**. As mentioned on page 201, using noise-shaping does not result in a reduction in quantization noise. Rather, noise-shaping topologies push the quantization noise to frequencies outside the region of interest. Reviewing Fig. 9.20d we see that (all of) our quantization noise is still present in the spectrum from DC to 400 MHz (see example below). It's impossible to reduce this noise with a digital filter without effecting the desired signal in this same frequency range.

**Example 9.3**

Estimate the RMS value of the quantization noise for the 8-path topology clocked at 100 MHz seen in Fig. 9.20d.

$$V_{Qe,RMS}^2 = \int_0^{4f_s} |NTF(f)|^2 \cdot \frac{V_{LSB}^2}{8 \cdot 12 f_s} \cdot df = \int_0^{4f_s} 4 \cdot \sin^2\left(\pi \cdot \frac{f}{f_s}\right) \cdot \frac{V_{LSB}^2}{8 \cdot 12 f_s} \cdot df$$

$$= \frac{V_{LSB}^2}{4 \cdot 12 f_s} \cdot \int_0^{4f_s} \left[1 - \cos\left(2\pi \cdot \frac{f}{f_s}\right)\right] \cdot df = \frac{V_{LSB}^2}{12}$$

or $V_{Qe,RMS} = V_{LSB}/\sqrt{12}$ , no reduction in quantization noise. While this result was derived for the case $K_{path} = 8$ the reader should see that the result is valid for any value of $K_{path}$. ■

**Example 9.4**

Estimate $SNR_{ideal}$ for the topology seen in Fig. 9.20a.

The procedure for calculating SNR was given in Sec. 5.2. Since we just calculated the quantization noise, all that's left is to calculate the desired output signal power. If we apply a sinusoid to the topology with a peak amplitude of $V_p$ , then adding the output signal powers from each path results in $K_{path} \cdot V_p^2/2$. We can then write

$$SNR_{ideal} = 20 \cdot \log \frac{\sqrt{K_{path}} \cdot V_p/\sqrt{2}}{V_{LSB}/\sqrt{12}}$$

or (see Fig. 5.31)

$$SNR_{ideal} = 6.02N + 1.76 + 10 \log K_{path} \tag{9.4}$$

For every doubling in the number of paths we get a 3-dB (0.5 bits) increase in the SNR. Such a modest increase in SNR is, generally, too insignificant for the hardware costs so this approach isn't used. ■

The virtues of the topology seen in Fig. 9.4 should be clearer. Using a single integrator (1-sigma) with $K$ feedback paths ($K$-deltas) we can attain conversion bandwidths approaching $f_s/2$ with reasonable resolutions. Since a single integrator is used, common information is applied to the input of each quantizer. This allows the quantization noise to be pushed to very high frequencies. In other terms, we get fast sampling by using $K$ feedback paths while we are able to push the quantization noise to higher frequencies using a single integrator common to all quantizers.

### 9.1.5 Understanding the Clock Signals

Notice that in Fig. 7.2 we used $\phi_1$ to clock both the input switches and the comparator. In our high-speed topology seen in Fig. 9.4, however, we used an earlier clock signal to strobe the comparator. For example, examining the top path in Fig. 9.4, the input switches are clocked with $\phi_{1-1}$ while the comparator is clocked with $\phi_{2-1}$. The input signal is captured on $C_i$ at the falling edge time of $\phi_{1-1}$. A very short time later, $T_s/K_{path}$, we clock the comparator. Since *near-ideal components* are used in our simulations the delay associated with transferring the charge on $C_i$ to the integrator's output is nearly zero. When the comparator makes a decision it is immediately fed back to the integrator, again with minimal delay, so that the next path's decision is directly dependent on the previous path's decision. The result is the topology behaves like a single, first-order, noise-shaping topology clocked at $K_{path} \cdot f_s$ with an input/output relationship given by Eq. (7.2).

In a practical implementation of the modulator, however, the delay through the integrator and comparator is finite and so the information fed back experiences a delay. This delay causes *a limit-cycle oscillation* (a self-sustained oscillation). Figure 9.21 shows an example of this oscillation. In this figure the input signal is the DC common-mode voltage, 500 mV here, and the clock signals are selected so that, as in Fig. 7.2, the same clock signal is used for both the input pair of switches and the comparator. The average value of the output signal does equal the input signal; however, we have the unwanted oscillation seen in the figure. Figure 9.22 shows another example but with an AC input signal. Since this oscillation always occurs at a relatively high frequency the digital filter on the output of the modulator reduces it.

The practical problem with using the clock phases seen in Fig. 9.4 with non-ideal elements is that the comparator delay varies making it appear as though the clock is jittering, see Secs. 5.2.1 and 5.2.2. The result is a reduction in *SNR* and *amplitude modulation* in the data converter's output, Fig. 5.22. Adding an amplifier in series with the output of the integrator to ensure that the comparators make a fast decision can help with this problem. In the design presented in the next section we use the same clock phase for both the input switches and the comparator. This reduces the requirements placed on the comparator allowing a decision to be made in nearly an entire (slow) clock cycle $T_s$. The drawback is that the quantization noise is no longer centered on $K \cdot f_s/2$ and so the *SNR* suffers.

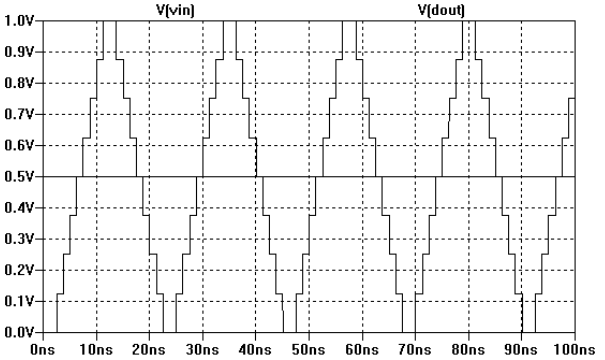



**Figure 9.21** Showing limit-cycle oscillations in the high-speed modulator.

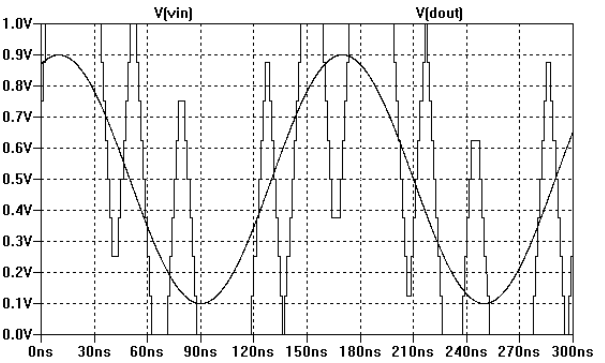



**Figure 9.22** Showing limit-cycle oscillations with a sinewave input signal.

## 9.2 Practical Implementation

In this section we'll discuss the practical implementation of the high-speed topology proposed in this book. The goal of this section is to provide discussions and provoke thought that should prove helpful when designing a converter using this topology. The goal is not to provide definitive solutions for specific applications. This endeavor is left for discovery by the engineers and researchers doing mixed-signal circuit design.

### 9.2.1 Generating the Clock Signals

Generating the 16 clock signals needed for the topology seen in Fig. 9.4 can be challenging. We could use a delay-locked loop (DLL) that takes an input clock signal and generates the 16 clock signals (but that adds complexity). Here we use a ring oscillator that runs asynchronously with an external clock signal. Figure 9.23 shows the basic delay stage schematic and icon used in the oscillator. Figure 9.24 shows the complete ring oscillator while Fig. 9.25 shows some simulation results in a 500 nm, 5-V, CMOS

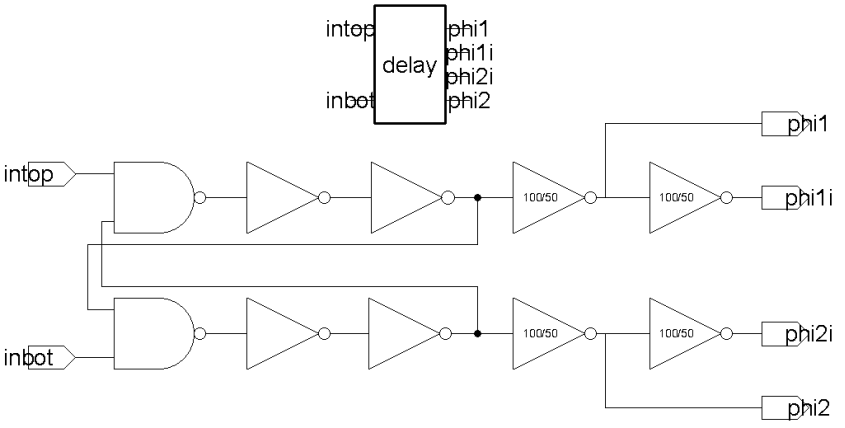



**Figure 9.23** Delay stage used in the ring oscillator, schematic and icon.

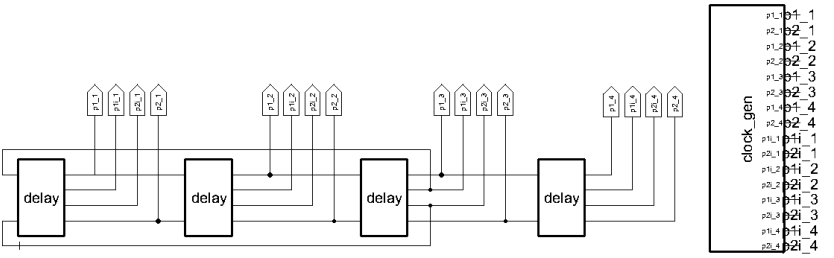



**Figure 9.24** Ring oscillator, schematic and icon, for use with the data converter.

process. Note that the frequency of the oscillator can be adjusted by adding or removing inverters in the delay stage. Also notice that the simulated oscillation frequency is around 200 MHz (about half of this for a simulation with layout parasitics). The analog input signal is sampled at a rate of $f_{s,new}$ or 1.6 GHz. The 8 path outputs are added together, Fig. 9.4, and then decimated, Fig. 9.7a.

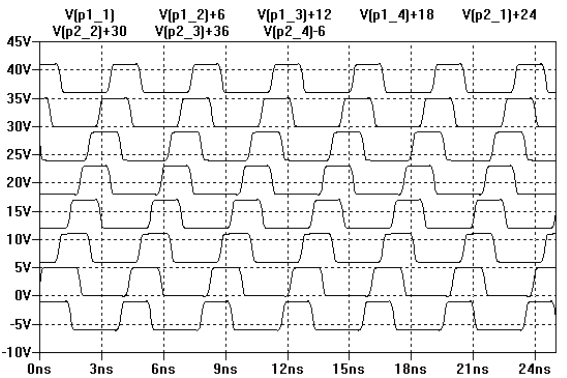



**Figure 9.25** Simulating the oscillator in a 500 nm process.

Figure 9.26 shows how the external (synchronous) and internal (asynchronous) clocks can be interfaced using a synchronizer. We are assuming that the internal clock is running faster than the external clock (so the output is decimated). Note that the synchronizer doesn't introduce aperture jitter as discussed in Sec. 5.2.1 since it only processes digital signals. After reviewing this section, however, we might wonder *if the ring oscillator is a practical choice for providing the clock signals to the data converter*. Ring oscillators are certainly not as stable as crystal-controlled oscillators. However, notice that by combining the $K_{path}$ outputs together we reduce the variance of the aperture jitter by $K_{path}$, see, for example, Eq. (5.30). Further filtering reduces the effects of a jittery clock signal. Slow variations in *VDD*, ground, or temperature will also have essentially no affect on the data converter's performance (via the ring oscillator) since these changes simply vary the sampling rate (the internal clock frequency). Note that for large differences in the internal and external clock frequencies aliasing concerns, when decimating, should be taken into consideration in the synchronizer (as should metastability concerns, e.g., the external clock (not) going low just after the internal clock goes high resulting in a glitch on the output of the AND gate in Fig. 9.26).

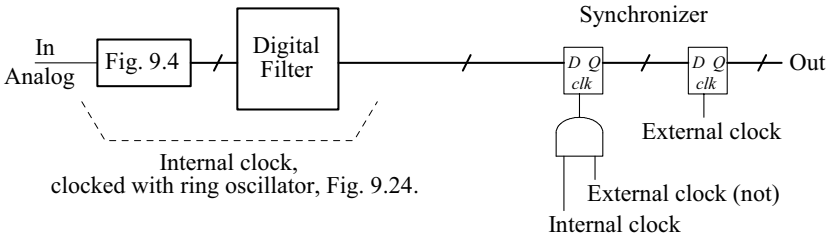



**Figure 9.26** Synchronizing the external and internal clock signals.

### 9.2.2 The Components

*The Switched-Capacitors*

Figure 9.27 shows the schematic and the icon for the switched-capacitors used in the modulator. The concerns, when selecting the widths of the MOSFETs, as discussed on pages 302 and 307, are that the capacitors can be charged/discharged within $T_s/K_{path}$. The effective resistances of each MOSFET in Fig. 9.27 is 500 Ω. Since two MOSFETs are in parallel with switches on each side of the 100 fF capacitor, the time constant is 50 ps. Finally, the size of the capacitors used in this circuit, as discussed on page 307, is set by thermal noise considerations.

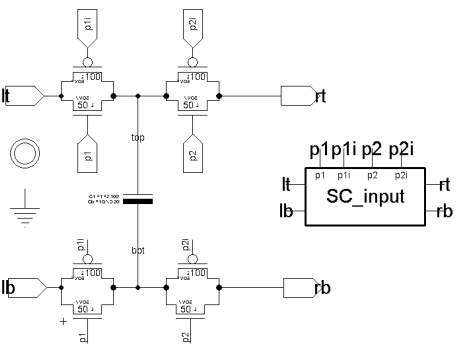



**Figure 9.27** Schematic and icon of the switched-capacitors used in the modulator.

*The Amplifier*

Figure 9.28 shows the schematic and icon for a self-biased amplifier. This topology was picked because it's simple and fast. Further, for the comparator discussed next, the outputs are complementary and generated from NMOS devices (which interface nicely with the comparator's inputs). Simulation results are seen in Fig. 9.29. The non-inverting amplifier input is held at 2.5 V while the inverting input is pulsed from 2.45 to 2.55 V (100 mV change). The output changes from roughly 2.8 to 0.8 (2 V change) so the gain is 20 (which should be larger than $K_{path}$ or 8 here). Both outputs of the amplifier are loaded, in the simulation, with 400 fF capacitors. Our concern, Sec. 7.1.8, is that we don't see slewing in the response. The amplifier sizes picked here are larger than required for our final design (so power is wasted).

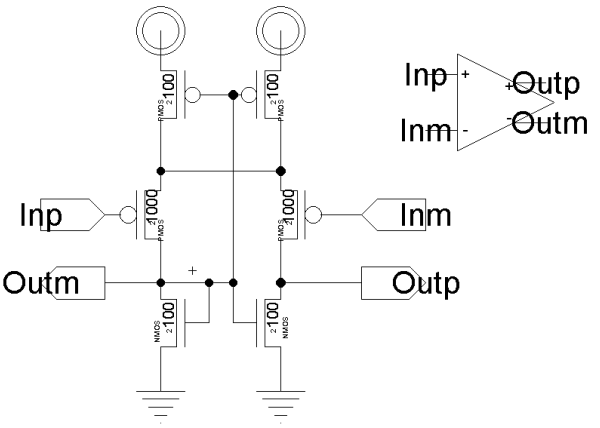



**Figure 9.28** Self-biased amplifier and icon.

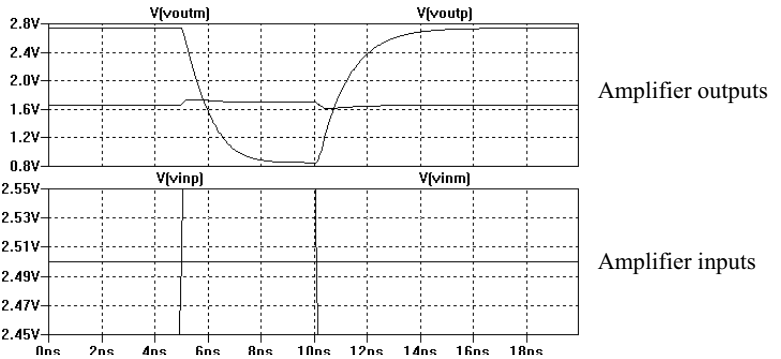



**Figure 9.29** Simulating the operation of the amplifier in Fig. 9.28 with 400 fF loads.

*The Clocked Comparator*

The schematic and icon of the clocked comparator used in the design is seen in Fig. 9.30. The input clock signal is sharpened and buffered up so that the input capacitance of the comparator is reduced and the edges in the comparator are fast (so that a good, reliable, decision can be made). The basic latch is formed with cross-coupled inverters (as usual). No DC current flows in the comparator. Current is pulled from *VDD* when the clock signal changes states. The NAND gates are used to ensure that the comparator output changes states (only) on the rising edge of the input clock signal. The outputs of the NAND gates are buffered up to ensure that they can drive the switched-capacitors. If these outputs don't fully charge the switched-capacitors or the *VDD*/ground connected to these gates droops or bounces, the feedback signal will be affected causing noise and nonlinearity. Finally, the inputs to the comparator should be above the threshold voltage of the NMOS device but not so large that the input devices are pushed into deep triode. Note, as discussed already, that we need to be careful to ensure minimal forward delay through the comparator (and, of course, the integrator).

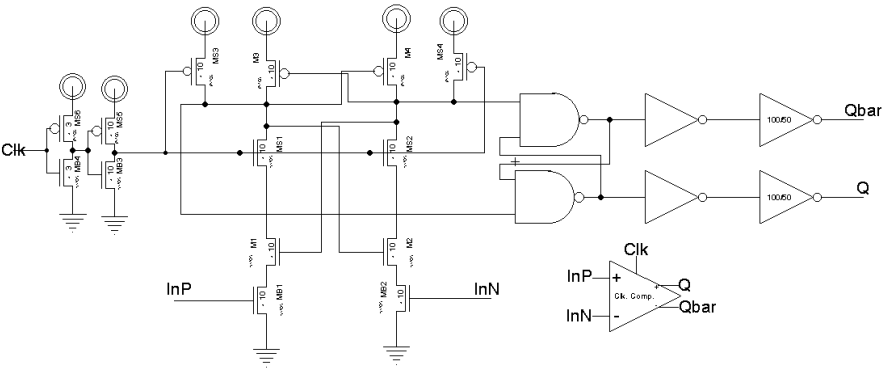



**Figure 9.30** Clocked comparator and icon.

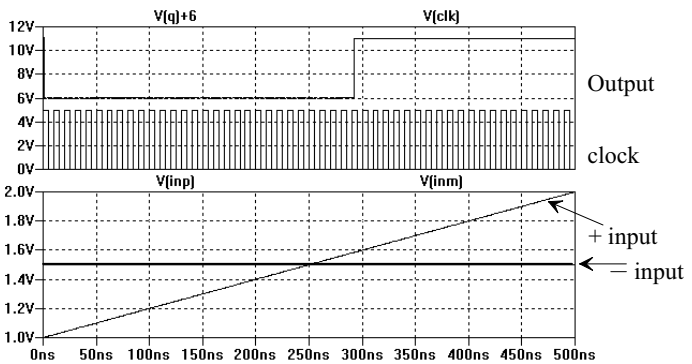



**Figure 9.31** Simulating the operation of the comparator in Fig. 9.30.

### 9.2.3 The ADC

Figure 9.32 shows the schematic of the high-speed ADC (minus the digital filter) we designed in this section using a 500 nm CMOS process with a *VDD* of 5 V. The resistors are used (for SPICE simulations) to provide a simple method of summing the digital outputs for ease of viewing. While they are useful for speeding up the simulation, they are not useful for looking at the performance of the converter. The finite transition times, glitches, etc. in the digital data limit their usefulness when viewing analog data. Figure 9.33 shows the results of simulating the ADC with a 10 MHz input signal.

At this point, in order to characterize the behavior of the ADC, we need to add the digital filter to the modulator's output. Ideally the *K*-delta-1-sigma topology seen in Fig. 9.4 behaves like a single, first-order, noise-shaping modulator clocked at $f_{s,new}$ ($=f_s \cdot K_{path}$ or 1.6 GHz here) with a 1-bit output (we selectively connect each path's output to the overall topology's output every $T_s/K_{path}$ seconds). Let's use the decimator and filter seen in Fig. 9.10, but with, as discussed in Sec. 7.1.3, $L = 2$ (the adder and only one Sinc filter). The final output word size is 7-bits, the effective 1-bit output by the modulator and then 3-bits (each) for the addition and the Sinc filter. We won't

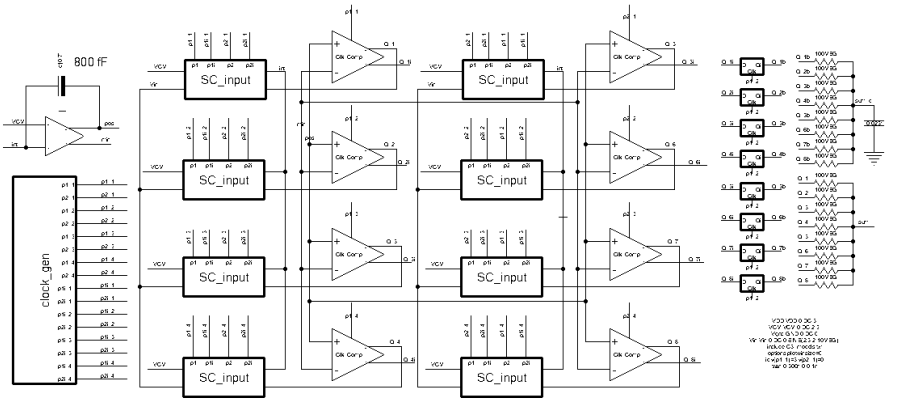



**Figure 9.32** K-delta-1-sigma modulator.

synchronize the output data to an external clock. This means that the output will change at the rate of the ring oscillator (roughly every 5 ns or 200 MHz). An input signal frequency of (roughly) 200 MHz/16 or 12.5 MHz will be attenuated by −3.9 dB. Figure 9.34 shows the simulation results for the ADC (modulator and filter) when the input signal is 10 MHz (same as Fig. 9.33 but showing the output of the digital filter). These conversion rates are comparable to any high-speed converter topology implemented in a 500 nm CMOS process (even faster if we use less decimation early in the digital filter).

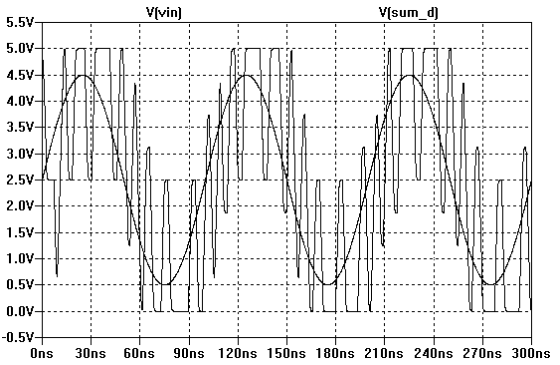



**Figure 9.33** Simulating the ADC in Fig. 9.32 with a 10 MHz input signal.

Figure 9.35 shows a simulation where the ADC's input is a ramp signal that transitions from 0 to *VDD* in 1 μs. The delay through the filter is 50 ns so the input signal should be shifted 50 ns later in time when comparing the ADC's input to its output. Of course, the ADC's output will then be valid only after this delay (so we neglect the start-up transient). We can estimate the minimum step size in the output of the ADC as $VDD/2^6$ or 78 mV. We use $2^6$ instead of $2^7$ (the exponent is the number of bits coming out of the filter) because in order for the output signal to swing rail-to-rail we have to

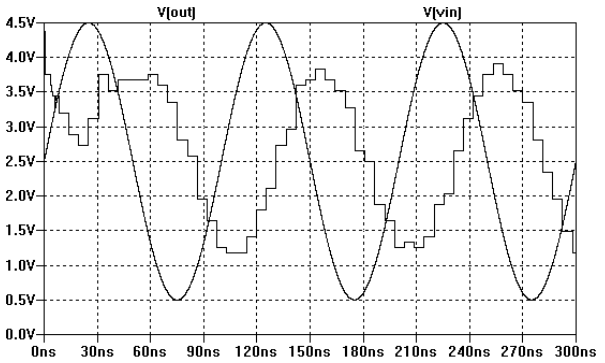



**Figure 9.34** Same as Fig. 9.33 (10 MHz input) but looking at the output of the digital filter.

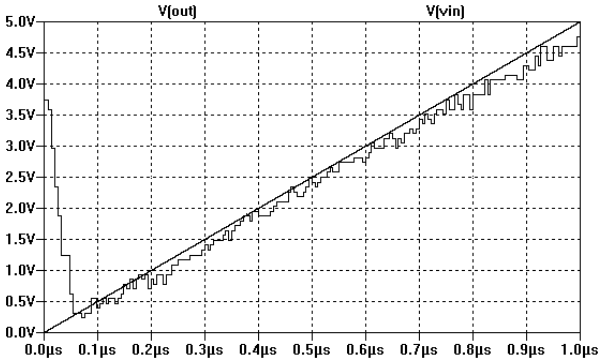



**Figure 9.35** Simulating the modulator with a ramp input signal. Shift input later in time by 50 ns to compare to output.

multiply it by two before applying it to the ideal DAC used to display the digital data as an analog output voltage (as discussed earlier).

Remember that increasing the output word size (increasing resolution) requires additional filtering. Also, again, recall that to increase the signal conversion bandwidth we need to reduce the amount of decimation used immediately following the modulator and prior to the adder that sums the $K_{path}$ outputs together (see Fig. 9.7b for the decimate by $K_{path}/2$ example).

## 9.3 Conclusion

This chapter has proposed a new topology for high-speed analog-to-digital conversion using the mixed-signal circuit design techniques presented in this book. The topology should prove useful when designing high-speed ADCs in nanometer CMOS. Future work can be focused in many areas including, but not limited to: fully-differential signal paths, the integrator (amplifier) design, higher-order (perhaps passive, Sec. 6.2.1) topologies, bandpass converters, segmenting feedback paths, digital calibration (e.g. for offset, Sec. 7.1.9) and the design of the digital filter (including the decimating stage).