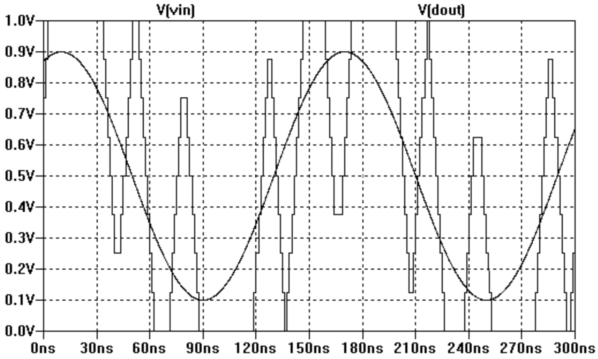

**Figure 9.22** Showing limit-cycle oscillations with a sinewave input signal.

## 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

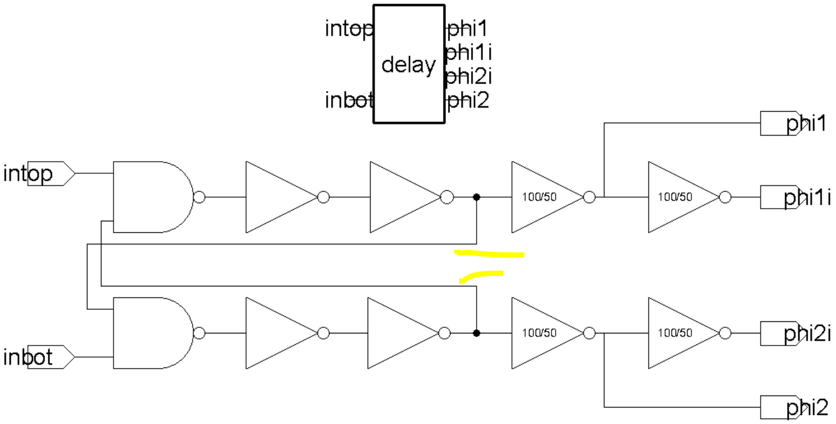

**Figure 9.23** Delay stage used in the ring oscillator, schematic and icon.

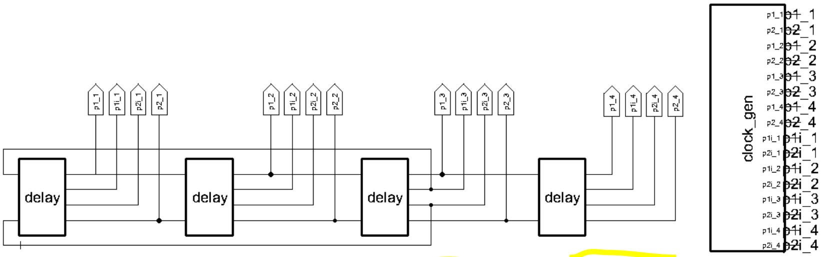

**Figure 9.24** Ring oscillator, schematic and icon, for use with the data converter.

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.

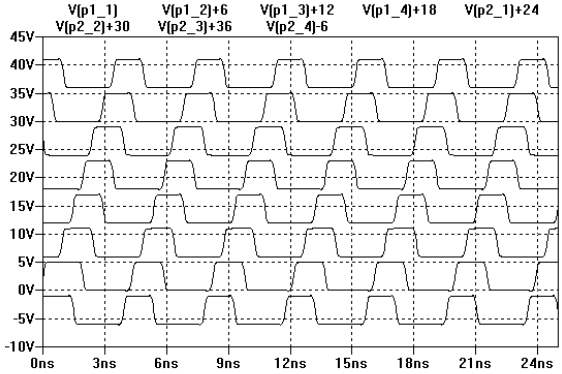

**Figure 9.25** Simulating the oscillator in a 500 nm process.

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).