

Two wires control SPI high-speed ADC

Dan Meeks, Texas Instruments Inc, Austin, TX



Most current microprocessors, DSPs, and field-programmable gate arrays integrate hardware and software resources that support either or both of two common interface standards—SPI (serial-peripheral interface) and I²C (inter-IC)/SMBus. Both two-wire-interface standards suffer from a few crucial disadvantages. For

example, I²C's throughput rates are 100 kbps, 400 kbps, or 3.4 Mbps in standard-, fast-, and high-speed modes, respectively, and can thus restrain a fast peripheral data converter's sample rate. Excluding framing and overhead bits, a 100k-sample/sec, 12-bit ADC must transfer at least 1.2 Mbps over the interface, a rate that only I²C's

high-speed mode supports. Many processors and controllers currently offer no I²C high-speed mode and thus would be unable to support a fast data converter.

One of I²C's major benefits reduces the number of host-to-target interconnections. Using only two wires plus ground, the host controller can address the target device and exchange data, whereas SPI requires three wires—data, clock, and chip-selection—plus ground. Multiple SPI-target devices can share data and clock lines, but each device requires its own

dedicated chip-selection line.

Given the perpetual demand for higher sample rates and resolution, μ C's limited speed may restrict its use in some applications and instead force designers to select SPI. However, SPI requires an additional I/O pin on the host controller. In situations in which extra pins are unavailable but the application requires a fast SPI-bus converter, you can apply the technique in **Figure 1**.

For example, Texas Instruments' ADS7816 comprises a 200k-sample/sec, 12-bit-sampling ADC that requires a bit rate of 3M samples/sec to sample continuously at a 200k-sample/sec rate (**Reference 1**). Selecting the ADS7816's active-low \overline{CS} (chip-select) pin initiates a conversion cycle. After toggling and holding \overline{CS} low during the data transfer, \overline{CS} returns high after transferring the data completes the process.

When the clock line initially goes low, it also asserts \overline{CS} to a low state. The time constant of the peak detector comprising D_1 , R_1 , and C_1 ensures that \overline{CS} does not go high until the clock line remains high for more than one clock cycle (**Figure 2**). Although the clock line toggles and retrieves data from IC_2 , \overline{CS} remains asserted low, and upon completion of retrieval, the clock line goes high, and \overline{CS} follows, readying the circuit for another conversion cycle.

Because C_1 must discharge at the end

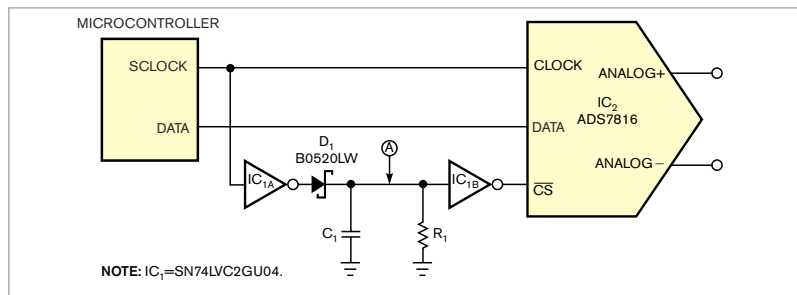
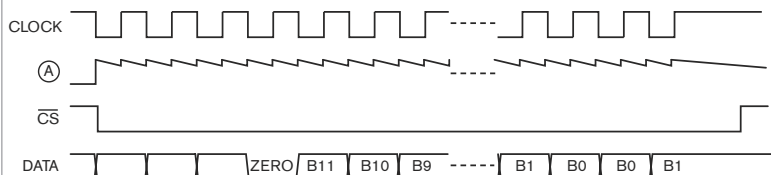


Figure 1 Two inverters and a few components can substitute for an SPI ADC's chip-select line.



NOTE: THE ADS7816 RETRANSMITS THE DATA IN REVERSE-BIT FORMAT AFTER ALL DATA HAS BEEN SENT.

Figure 2 An SPI-clock waveform (top trace) evokes data (bottom trace), and peak detection of the clock (the waveform at Point A in Figure 1) yields a signal (next-to-bottom trace) that mimics the chip-selection line's behavior.

of a conversion cycle, the controller should delay the start of the next conversion cycle until C_1 fully discharges. Careful choice of R_1 and C_1 minimizes the delay to a minimum of three clock cycles. In addition, the voltage across C_1 must not fall below inverter IC_{1B} 's input threshold before the next clock pulse arrives to refresh the capacitor's voltage. Inverter IC_{1A} 's output voltage and current capabilities affect C_1 's recharge

time, and R_1 and IC_{1B} 's input impedance affect the discharge time. To ensure a robust design, include components' tolerances and temperature coefficients along with variations of logic-input and -output thresholds. **EDN**

REFERENCE

1 ADS7816 data sheet, <http://focus.ti.com/docs/prod/folders/print/ads7816.html>, Texas Instruments.