09/05/01

Errata: CS5376 Rev. A

(Reference CS5376 Data Sheet revision DS256PP1 dated Jan '01)

Introduction

The CS5376 is a low power multi-channel digital filter with integrated system peripherals. It uses a coefficient programmable signal processing architecture to provide filtering for up to four ΔΣ modulators. Integrated peripherals simplify system design with a buffered high speed serial data output port, a test bit stream generator suitable for driving a test DAC, general purpose I/O pins for local hardware control, a secondary master mode SPI port for serial peripherals, and a JTAG port for boundary scan testing. In addition, a clock and synchronization block synchronizes the CS5376 to the host system, and a time break controller generates timing reference information in the output data stream.

Contents

1. SDTKI signal timing into the SD port

After filtering is completed, each 24-bit output sample is combined with an 8-bit status word. This 32-bit data word is written to an 8-deep FIFO buffer and then transmitted to the communications interface through the high speed serial data output port (SD port). The SD port can operate in two configurations, a continuous output configuration where data is output immediately when ready, or a requested output configuration where data is output only when polled by the communications controller.

Continuous Output SD Port Configuration

The continuous output SD port configuration requires the SDTKI pin to receive continuous rising edges. A simple method for generating rising edges on SDTKI is to connect it to a 4 MHz or slower system clock. Whenever output data is available from the decimation engine, a rising edge on SDTKI initiates an SD port transaction.

Once an SD port transaction starts, the SDTKI signal must be gated off to ensure no rising edges occur during the transaction. The easiest way to guarantee this is to gate the SDTKI input signal with the SDRDY output signal using an AND gate. When an SD port transaction starts, the SDRDY signal is driven low by the CS5376 which will gate off the SDTKI input. When the SD port transaction is complete, the SDRDY signal automatically returns high to re-enable the SDTKI input.

Requested Output SD Port Configuration

When using the SD port in a requested output configuration, a single pulse to the SDTKI pin initiates an SD port transaction. The pulse is generated by a controller that schedules the data into the communication network. Because data is requested only when the controller is ready to receive, local data buffering between the CS5376 and the communication network is not required.

When an SD port transaction is completed, the SDTKO output pin automatically generates a pulse that can trigger the SDTKI pin of another CS5376. In this way a pulse 'token' started by the communications controller can pass through a series of daisy-chained CS5376 devices, initiating SD port transactions in each. The final CS5376 SDTKO output can be returned to the controller to signal the end of a polling cycle.

1.1 Erratum Description

A timing constraint in the SD port was discovered during initial testing of CS5376 rev A. The signal that initiates an SD port transaction, SDTKI, is required to arrive with a certain timing relative to the internal decimation engine clock.

1.2 Erratum Work Around

Depending how the SD port is used in a design, either in the continuous output configuration or the requested output configuration, two simple methods are available to work around the SDTKI timing constraint. Both solutions use an inverted MCLK or MCLK/2 signal to establish valid timing for the SDTKI rising edge. The continuous output configuration work around is used on the CDB5372-76 rev A.0 evaluation board.

Continuous Output SD Port Configuration

To work around the SDTKI timing constraint when using the continuous output SD port configuration, an inverted MCLK or MCLK/2 clock signal is used as the rising edge source. For designs using an FPGA, the inverted MCLK or MCLK/2 signal is ANDed with the SDRDY signal and connected to the SDTKI input as shown in Figure 1. The inverter and AND gate are programmed into the FPGA, requiring the MCLK or MCLK/2 signal and the SDRDY signal to be supplied as inputs.

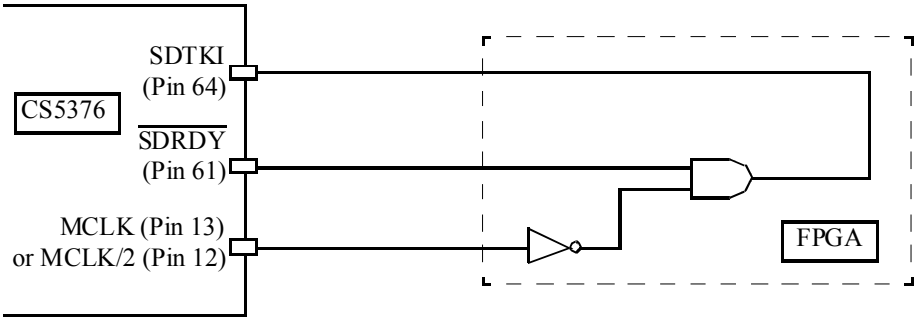

Figure 1. Continuous output configuration SDTKI work around using an FPGA

For designs that don't include an FPGA, discrete components are required. The number of required components is minimized by using the configuration shown in Figure 2. This configuration requires only a single 74HC00 quad 2-input NAND gate.

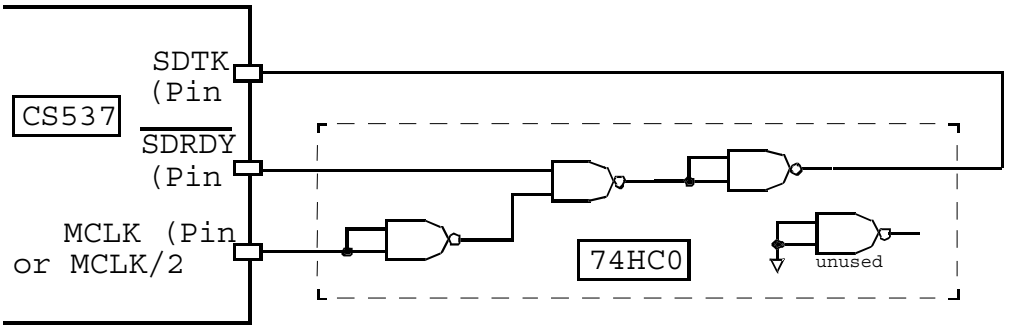

Figure 2. Continuous output configuration SDTKI work around using discrete logic

Requested Output SD Port Configuration

To work around the SDTKI timing constraint when using the requested output SD port configuration, an inverted MCLK or MCLK/2 signal is used to re-time the incoming SDTKI pulse. For designs using an FPGA, the inverted MCLK or MCLK/2 is used to clock a D-type flip-flop for the SDTKI signal as shown in Figure 3. The inverter and flip-flop are programmed into the FPGA, requiring the MCLK or MCLK/2 signal and the SDTKI signal to be supplied as inputs.

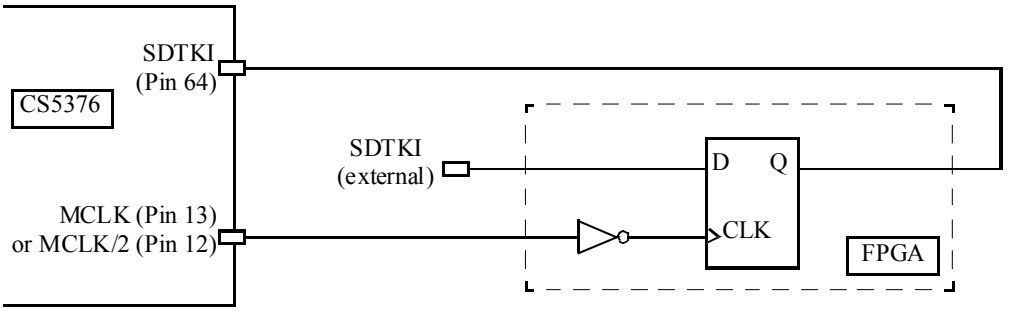

Figure 3. Requested output configuration SDTKI work around using an FPGA

For designs that don't include an FPGA, discrete components are required. The required board area is minimized by using Fairchild TinyLogic or similar components. The configuration shown in Figure 4 requires one TinyLogic NC7S14 inverter and one TinyLogic NC7SZ175 D-type flip-flop.

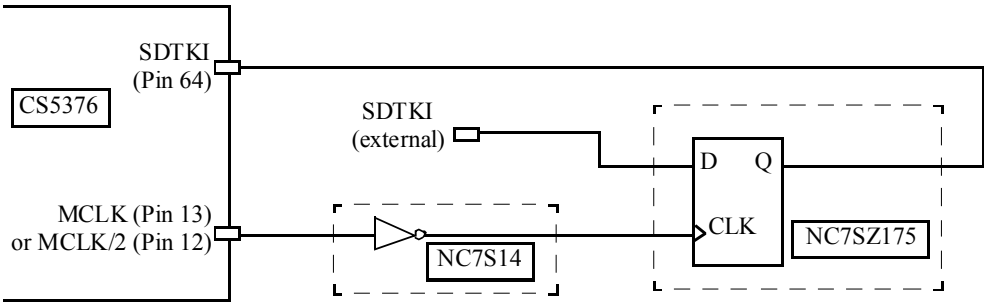

Figure 4. Requested output configuration SDTKI work around using discrete logic

Evaluation Board Modification

On the CDB5372-76 rev A.0 evaluation board, the continuous output SD port configuration is used. The Capture+ interface board includes an FPGA that uses the inverted MCLK/2 signal to generate the SDTKI signal as shown in Figure 1.

To connect the MCLK/2 signal to the Capture+ board, a wire should be installed on the back of the CDB5372-76 evaluation board between HDR36, pin 1 (MCLK/2) and J5, pin 18 (CAPTURE PLUS). The wiring change is shown in Figure 5.

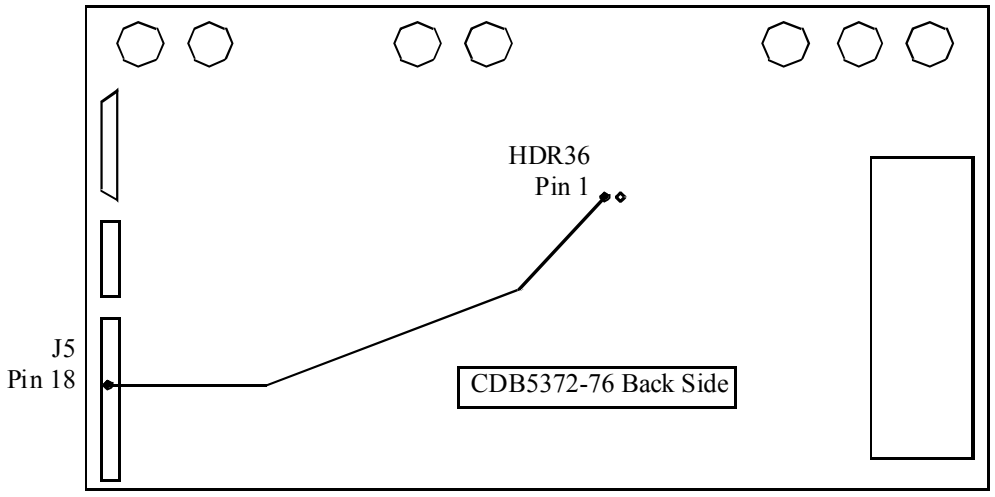

Figure 5. Continuous output configuration SDTKI work around for CDB5372-76 rev A.0

2. EEPROM configuration loader preamble

The CS5376 loads custom coefficients and other configuration information via the SPI 1 serial port from either a microcontroller or a configuration EEPROM. The BOOT pin input selects how the CS5376 loads, a logical low selects coprocessor boot mode using a microcontroller and a logical high selects stand-alone boot mode using a configuration EEPROM.

In stand-alone boot mode, configuration information is read from EEPROM and no microcontroller is required. The EEPROM contains all configuration information including filter coefficients, register settings, and test bit stream data. The CS5376 expects EEPROM programming to start at memory location 0x10, with the bytes from 0x00 to 0x0F defined for manufacturing header information.

2.1 Erratum Description

An error was discovered in the internal boot loader of CS5376 rev A when using an EEPROM to load configuration information.

2.2 Erratum Work Around

When creating a configuration EEPROM, the following data bytes must be written as the initial configuration data, starting at location 0x10 and ending at location 0x28. This sequence of data bytes activates an internal correction for the EEPROM loader error.

0C 00 10 CE 00 10 D3 00 00 01 00 00 00 00 10 00 FF FF FF 7F FF FF 80 00 00

Normal EEPROM command and data values will follow these data bytes in the loader EEPROM, as documented in the CS5376 data sheet, starting at EEPROM memory location 0x29.

If there are any questions concerning this information,
Please contact: (datacq@crystal.cirrus.com)