

## Interfacing to High Speed ADCs via SPI

by the High Speed Converter Division

### INTRODUCTION

This application note describes how to use the serial peripheral interface (SPI) port on Analog Devices, Inc., high speed converters. In addition, this application note defines the electrical, timing, and procedural requirements for interfacing to these devices. The implementation is compatible with industry-standard SPI ports and employs, at minimum, a 2-wire mode and optional chip select.

### DEFINITION

The SPI port consists of three pins: the serial clock pin (SCLK), the serial data input/output pin (SDIO), and the chip select bar pin (CSB). Optionally, some chips may implement a serial data out pin (SDO), which is referred to as 3-wire mode. To minimize pin count, most chips omit this pin. However, if it is included, it is used only for reading data from the device.

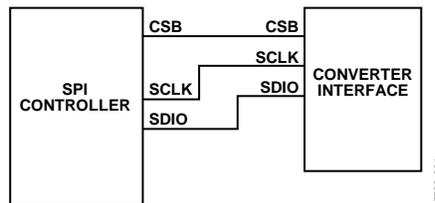


Figure 1. Single Device Control in 2-Wire Mode

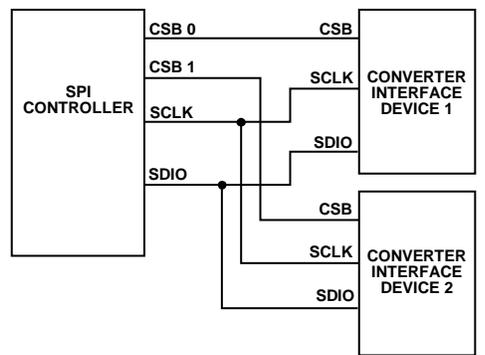


Figure 2. Multiple Device Control in 2-Wire Mode

## TABLE OF CONTENTS

Introduction.....	1	Address Bits.....	6
Definition.....	1	Detection of SPI Mode and Pin Mode.....	7
Revision History.....	2	Hardware Interfacing.....	7
SPI Port Pins.....	3	Chip Programming.....	8
Serial Clock (SCLK).....	3	Configuration Register (0x000).....	8
Serial Data Input/Output (SDIO).....	3	Transfer Register (Master-Slave Latching) (0x0FF).....	8
Chip Select Bar (CSB).....	3	Chip ID (0x001).....	9
Serial Data Out (SDO).....	4	Chip Grade (0x002).....	9
Format.....	5	Device Indexing (0x004 and 0x005).....	9
Instruction Phase.....	5	Program Registers.....	10
Read/Write.....	5	Programming Example.....	17
Word Length.....	5	Control Register.....	18
Streaming.....	6		

## REVISION HISTORY

### 4/2017—Rev. A to Rev. B

Change CSB 0 to CSB in Figure 1.....	1
Changes to Figure 2.....	1
Change to Serial Clock (SCLK) Section.....	3
Changes to Table 2 Caption.....	6
Changes to Table 3 Caption.....	8
Changes to Table 4 and Table 5.....	10
Changes to Table 6 and Table 7.....	11
Changes to Table 8.....	12
Changes to Table 9.....	13
Changes to Table 10 and Table 11.....	14
Changes to Table 12.....	15
Changes to Table 13.....	16
Change to Programming Example Section.....	17
Changes to Table 14.....	18

### 4/2007—Initial Version to Rev. A

Updated Format.....	Universal
Changes to Transfer Register Section.....	8
Changes to Figure 13.....	10
Added Table 6.....	11
Added PLL Control (0x00A) Section.....	11
Changes to Table 8.....	12

### 12/2005—Revision 0: Initial Version

## SPI PORT PINS

The following sections described the SPI port pins.

Caution: Refer to specific analog-to-digital converter (ADC) data sheets to determine the nominal and absolute maximum logic voltages.

### SERIAL CLOCK (SCLK)

The SCLK pin is the serial shift clock in pin. This pin is implemented with a Schmitt trigger, to minimize sensitivity to noise on the clock line, and it is pulled low by a nominal 50 k $\Omega$  resistor to ground. This pin may stall either high or low.

SCLK is used to synchronize serial interface reads and writes. Input data is registered on the rising edge of this clock and output data transmissions are registered on the falling edge.

Unless otherwise specified, the maximum clock speed of the ADC SPI port is 25 MHz. See the specific product data sheet for more information pertaining to SPI speeds supported for a particular device. The typical hold time ( $t_{DH}$ ) is 0 ns, and a minimum setup time ( $t_{DS}$ ) of 5 ns is required between SCLK and SDIO. (See the specific device data sheet to determine the exact interface timing requirements.) To optimize internal and external timing, the bus is capable of turning around the state of the SDIO line in half an SCLK cycle. This means that, after the address information is passed to the converter requesting a read, the SDIO line is transitioned from an input to an output within one half of a clock cycle. This ensures that by the time the falling edge of the next clock cycle occurs, data can be safely placed on this serial line for the controller to read. If the external controller is insufficiently fast to keep up with the ADC SPI port, the external device can stall the clock line to add additional time allowing for external timing issues.

### SERIAL DATA INPUT/OUTPUT (SDIO)

The SDIO pin is a dual-purpose pin. The typical role for this pin is as either an input or an output, depending on the instruction being sent (read or write) and the relative position (instruction or data phase) in the timing frame. During the first phase of a write or a read, this pin functions as an input that passes information to the internal state machine. If the command is determined to be a read command, the state machine changes this pin (SDIO) to an output, which then passes data back to the controller. (See  $t_{EN\_SDIO}$  and  $t_{DIS\_SDIO}$  in Table 1.) If the device includes an SDO pin and the configuration register is set to take advantage of it, the SDO becomes active instead of the SDIO pin changing to an output. At all other times, the SDO pin remains in a high impedance state. If the command is determined to be a write command, the SDIO pin remains an input for the duration of the instruction.

### CHIP SELECT BAR (CSB)

CSB is an active low control that gates the read and write cycles. There are several modes in which the CSB can be operated. For situations where the controller has a chip select output or other means of selecting multiple devices, this pin can be tied to the CSB line. When this line is low, the device is selected and information on the SCLK and SDIO lines is processed. If this pin is high, the device ignores any information on the SCLK and SDIO lines. In this manner, multiple devices can be connected to the SPI port. In cases where only one device is connected, the CSB line can be optionally tied low and the device is permanently enabled. (Tying the CSB line low excludes the possibility of resetting the device if an error occurs on the port.) The CSB line can also be tied high to enable secondary function of the SPI port. (See the Detection of SPI Mode and Pin Mode section for more details.) CSB is a high impedance line, pulled high by a nominal 50 k $\Omega$  resistor.

CSB may stall high, that is, remain high for multiple clock cycles (see Figure 5) in some configurations to allow for additional external timing. If three or fewer words (not counting instruction information) are being transmitted through the interface at a time, CSB may stall high between bytes, including the bytes of the instruction information. If CSB stalls high in the middle of a byte, the state machine is reset and the controller returns to the idle state, awaiting the transmission of a new instruction. This mechanism allows restoration after a fault has been detected. To detect the reset, at least one and no more than seven serial clocks must occur. Once the state machine has entered the idle state, the next falling edge of the CSB initiates a new transmission cycle.

Some devices implement secondary functions with the SPI pins. Typically, these functions include output data format, duty cycle stabilizer, or other common features. These pin functions are enabled by the CSB pin. If the CSB pin is tied high, the SPI functions are placed in a high impedance mode. In this mode, secondary functions are then turned on, allowing control of features on-chip, without requiring the SPI to operate. These features vary by device. Therefore, the individual device data sheet must be consulted to determine if this feature is supported and what it controls.

For applications to be controlled by the SPI port, the secondary function takes priority until the device has been accessed by the SPI port. By extension, any activity on the SCLK, SDIO, and SDO (if provided) is interpreted as a secondary function until the chip has been accessed by the SPI port. Therefore, the chip needs to be initialized as soon after power up as practical. (See the Detection of SPI Mode and Pin Mode section for more details.)

**SERIAL DATA OUT (SDO)**

To determine if a device supports the SDO pin, refer to the device data sheet. If SDO is present, it is in a high impedance state, unless data is actively being shifted on this pin to allow tying multiple devices together at the receiving end. Additionally, data is shifted out on the first falling edge of SCLK after the instruction phase is complete. When data is returned to the controller, the information is placed in the output shifters, within the time period between the last rising edge of SCLK associated with the instruction phase and the immediately next falling edge. This can be nominally 20 ns when operating at 25 MHz.

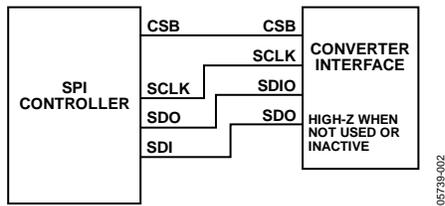


Figure 3. 3-Wire Control

Table 1. Serial Timing Specifications<sup>1</sup>

Symbol	Description
$t_{DS}$	Setup time between data and rising edge of SCLK.
$t_{DH}$	Hold time between data and rising edge of SCLK.
$t_{CLK}$	Period of the clock.
$t_S$	Setup time between CSB and SCLK.
$t_H$	Hold time between CSB and SCLK.
$t_{HI}$	Minimum period that SCLK needs to be in a logic high state.
$t_{LO}$	Minimum period that SCLK needs to be in a logic low state.
$t_{EN\_SDIO}$	Minimum time it takes the SDIO pin to switch between an input and an output relative to SCLK falling edge.
$t_{DIS\_SDIO}$	Minimum time it takes the SDIO pin to switch between an output and an input, relative to SCLK rising edge.

<sup>1</sup> See device data sheet for minimum and maximum ratings.

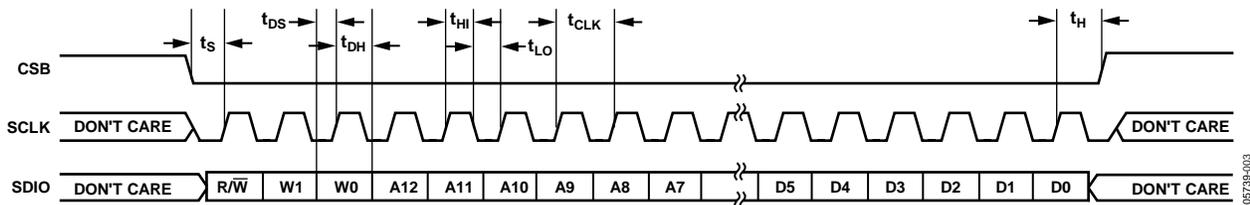


Figure 4. Setup and Hold Timing Measurements

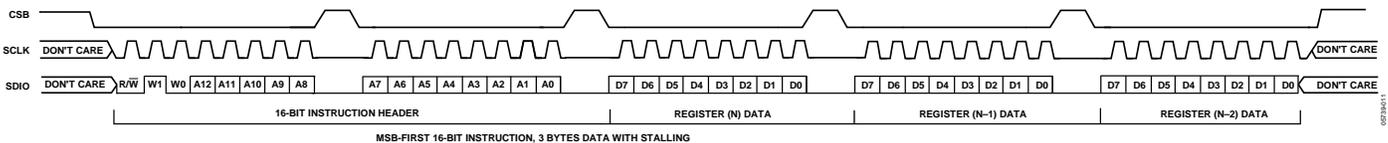


Figure 5. Most Significant Bit (MSB)-First Instruction and Data with Stalling

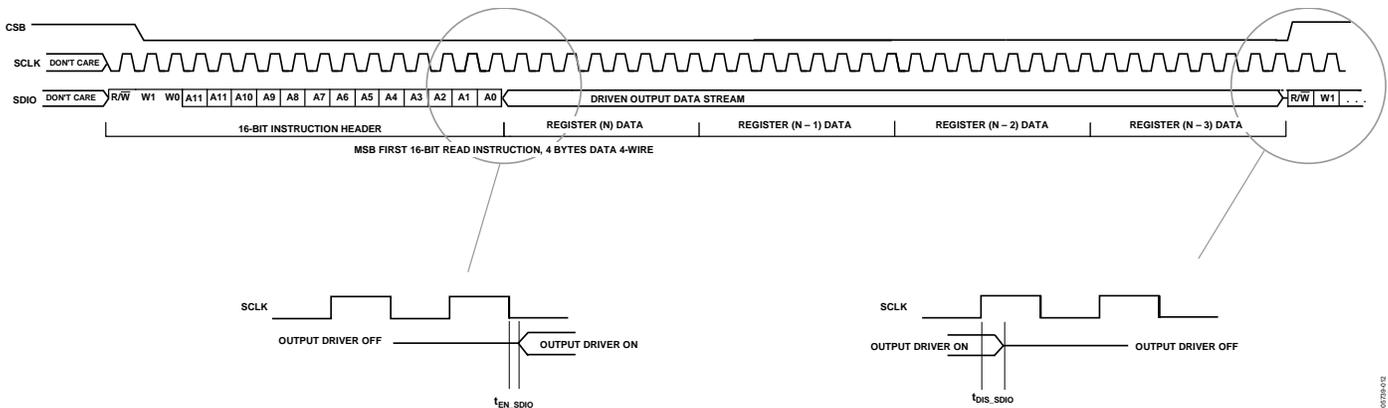


Figure 6. Typical SDIO Output Enable and Disable Timing

## FORMAT

The falling edge of CSB, in conjunction with the rising edge of SCLK, determines the start of framing. Once the beginning of the frame has been determined, timing is straightforward. The first phase of the transfer is the instruction phase, which consists of 16 bits followed by data that can be of variable lengths in multiples of 8 bits. If the device is configured with CSB tied low, framing begins with the first rising edge of SCLK.

### INSTRUCTION PHASE

The instruction phase is the first 16 bits transmitted. As shown in Figure 4 and Figure 7, the instruction phase is divided into a number of bit fields.

#### READ/WRITE

The first bit in the stream is the read/write indicator bit (R/W). When this bit is high, a read is being requested. At the completion of the instruction phase (the first 16 bits), the internal state machine uses the information provided to decode the internal address to be read. The direction of the SDIO line is changed from input to output, and the appropriate number of words defined by the word length are shifted out of the device (see the Word Length section). If the device is equipped with an SDO and the configuration register is appropriately set, the SDO line is taken out of high impedance and data is passed out the SDO pin instead of the SDIO pin. Once all data specified by the word length has been shifted out, the state machine returns to idle mode and awaits the next instruction phase.

When the first bit in the data stream is low, a write phase is entered. At the completion of the instruction phase, the internal state machine uses the information provided to decode the internal address to be written. All data after the instruction is shifted in the SDIO pin and sent to the target addresses. Once all data specified by the word length has been transferred, the state machine returns to idle mode and awaits the next instruction phase.

In either read or write mode, the process continues until the word length is reached or until the CSB line is lifted. If the end of memory is reached (either 0x000 or 0x0FF), the rollover occurs and the next address processed is 0x000, if the address is incrementing, or 0x0FE, if the address is decrementing.

#### WORD LENGTH

W1 and W0 represent the number of data bytes to transfer for either read or write. The value represented by  $W1:W0 + 1$  is the number of bytes to transfer. If the number of bytes to transfer is three or less (00, 01, or 10), CSB can stall high on byte boundaries. Stalling on a nonbyte boundary terminates the communications cycle. If these bits are 11, data can be transferred until CSB transitions high. CSB is not allowed to stall during the streaming process. Once streaming has begun (defined as beyond the third data byte), CSB is not allowed to return high until the operation is complete. When CSB does go high, streaming is terminated, and the next time CSB goes low, a new instruction cycle is initiated. If CSB goes high on a non-8-bit boundary, the communications cycle is terminated, and any incomplete bytes are lost. Completed data bytes, however, are properly handled.

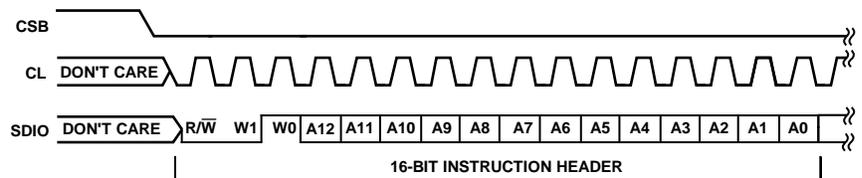


Figure 7. Instruction Phase Bit Field

Table 2. Word Length Settings

[W1:W0] Setting	Action	CSB Stalling
00	1 byte of data can be transferred.	Optional
01	2 bytes of data can be transferred.	Optional
10	3 bytes of data can be transferred.	Optional
11	4 or more bytes of data can be transferred. CSB must be held low for entire sequence; otherwise, the cycle is terminated, and an instruction cycle is anticipated when CSB returns low.	No

If the value represented by W0 and W1 is 0, one byte of data is transferred. If the value represented by W0 and W1 is 1, two bytes of data are transferred. If the value represented by W0 and W1 is 2, then three bytes of data are transferred. Following completion of the data transfer, the state machine returns to idle state, awaiting the next instruction phase.

**STREAMING**

If the value represented by W0 and W1 is 3, data is constantly streamed to the device. As long as CSB remains low, the part continues to accept new data, starting with the initial address and continuing to the next address with each new word received. It is recommended that streaming not be combined with the CSB line physically tied low, because streaming can only be terminated by lifting the CSB line high. If streaming is used with CSB tied low, the first instruction used is carried out indefinitely. This means that once a write (or read) cycle is entered, data may not be read (or written) from the device. Similarly, the starting address is continually and automatically incremented/decremented, according to the mode, with no chance to directly change the address of the state machine. (The address generator continues to wrap around the terminal addresses in a predictable manner.) This may not be a problem if the user only wants to program the device with no possibility of reading internal registers. It is recommended that users who tie

the CSB line low transfer data in 1-, 2-, or 3-byte blocks, unless they are certain that they do not wish to read data from the internal registers. Although it is not required, it is recommended that users maintain control over the CSB line so the streaming process can be interrupted and the state machine can be reset to the idle state.

**ADDRESS BITS**

The remaining 13 bits represent the starting address of the data sent. If more than one word is being sent, sequential addressing is used, starting with the one specified, and it either increments or decrements based on the mode setting.

**Data Phase**

Data follows the instruction phase. The amount of data sent is determined by the word length (Bit W0 and Bit W1). This can be one or more bytes of data. All data is composed of 8-bit words. If the state machine detects incomplete data being transmitted, the state machine resets and enters an idle state, awaiting a new instruction to be initiated by the next falling edge of the CSB line. If CSB is physically tied low, fault correction is not possible unless the device includes a chip reset function. (See the individual device data sheets for more detail.)

**Bit Order**

Data can be sent in either MSB-first mode or LSB-first mode (see the Configuration Register (0X000) section). On power up, MSB-first mode is the default. This can be changed by programming the configuration register. In MSB-first mode, the serial exchange starts with the highest-order bit and ends with the least significant bit (LSB). In LSB-first mode, the order is reversed. The instruction is 16 bits long, consisting of 2 bytes, as described earlier. In MSB-first mode, the bit order is highest-order bit to lowest-order bit. In LSB-first mode, the entire 16 bits are reversed, as shown in Figure 8.

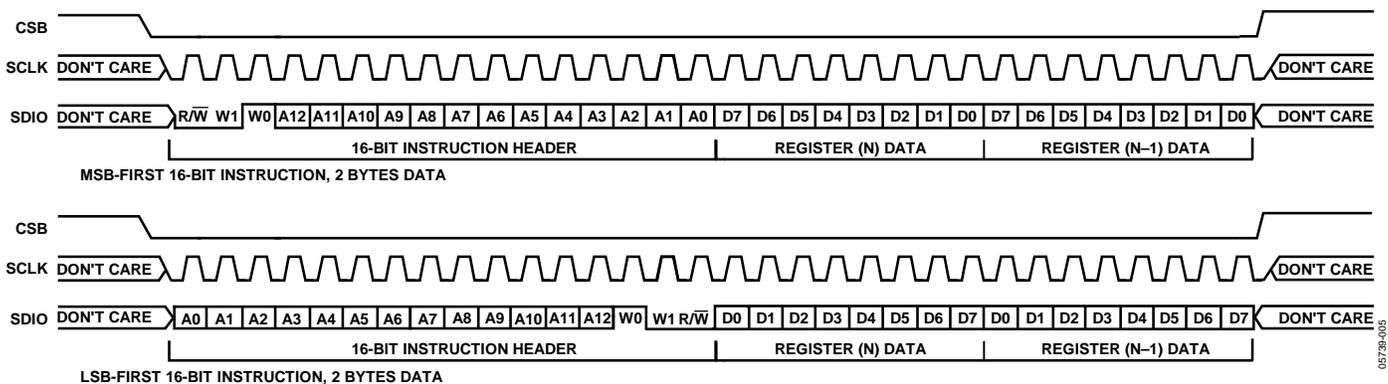


Figure 8. MSB-First and LSB-First Instruction and Data Phases

## DETECTION OF SPI MODE AND PIN MODE

Some users may choose not to use an SPI port to configure their device. Where possible, devices are designed to power up using typical settings. (For exact details, consult the appropriate device data sheet.) However, there may be cases where users want to change the basic features without inclusion of an SPI controller. Examples include controlling the duty cycle stabilizer or the format of the data output between twos complement and offset binary. For these types of options, the chip can be specified such that external controls can be used to change the options without having to program the device. To minimize the number of external pins, the SPI pins are reassigned to these alternate functions.

For devices that implement this option (see the device data sheet to determine if this option is supported), the user can choose to enable the pin control modes. To do so, the CSB line must be tied high. While this pin is high, the remaining SPI pins become alternate functions, and any setting on those pins takes effect as defined in the device data sheet. Once the user decides to enter SPI mode, pin mode cannot be re-entered unless the device is powered off first.

At power up, the device defaults to pin control mode as long as CSB is logic high. If the CSB line is wired high, the device always functions in pin control mode. Likewise, if the CSB line is wired low, the device powers up in SPI mode (see the Streaming section for limitations in this mode). In most cases, the CSB line is used to select the chip. Typically, in this mode, CSB is taken high usually at power up by the external SPI controller. Therefore, by default, the remaining SPI pins initially function in pin control mode. As soon as the CSB line is taken low to select the chip, the SPI function is enabled, which ignores the state of the other pins and places control strictly with the settings of the internal memory map.

When the CSB line is low, the state machine expects an SCLK to shift in data. After 9 clock cycles, representing the first byte (plus an extra cycle), the internal state machine no longer looks at the CSB pin to determine if pin mode or SPI mode is used. The logic used for this is shown in Figure 9. As long as the CSB line is high from power-up, an internal mux is used to select the alternate functions for the SPI pins. Once the CSB line is taken low, the mux is deselected from the input pins and begins interpreting these signals as SPI signals. After an SPI command is recognized, the mux stays in the SPI position, regardless of the state of the CSB line. Therefore, users cannot hop between SPI mode and pin mode.

Additionally, if the SPI port is being used for control, it is recommended that the device be configured as part of other start-up procedures to ensure that the device is ready in the desired state if the pin mode is not to be used. Performing a soft reset function does not cause the part to revert back to pin mode.

The only means to accomplish this are cycling the power on the device or asserting the device pin reset, if the part is so equipped. Note that not all parts include a pin reset. See the device data sheet for details.

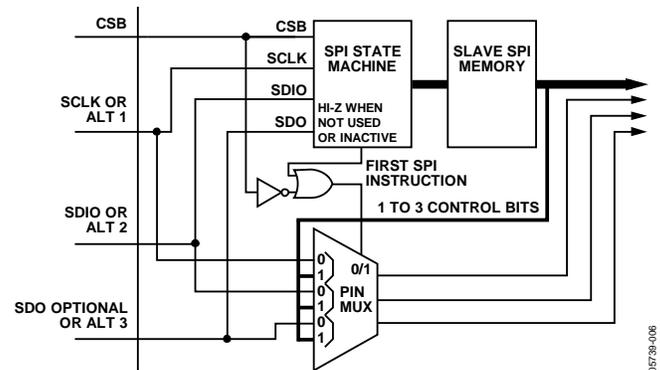


Figure 9. Hardware Interfacing

## HARDWARE INTERFACING

Although these devices are designed to be interfaced to SPI controllers, it is not necessary to always use an SPI controller to set up these devices. Pin mode provides one alternative, but in cases where more flexibility is desired, it is possible to use either serial PIC or PROMS microcontrollers, as shown in Figure 10 and Figure 11. For more details on programming these devices with PIC microcontrollers, see the [AN-812 Application Note, Microcontroller-Based Serial Port Interface \(SPI\) Boot Circuit](#).

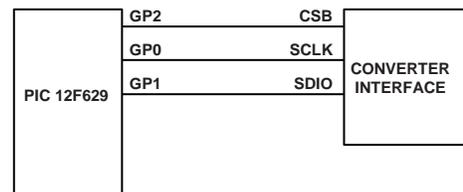


Figure 10. Programming with a Low Cost PIC Microcontroller

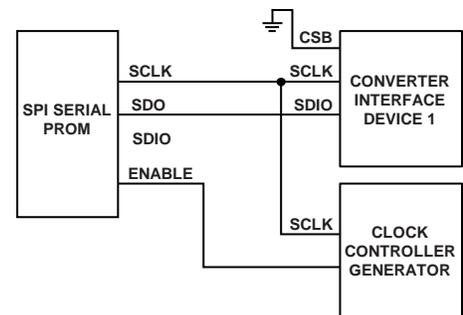


Figure 11. Programming with a Low Cost Serial PROM and External Clock Source

## CHIP PROGRAMMING

The SPI port is the mechanism for configuring the converter. In addition, a structured register space is defined for programming the device. This structure is divided into addresses pointed to by the address in the instruction phase of the data transfer. Each address is divided into 8-bit bytes. Each byte can be further divided down fields, which are documented in the following sections.

There are three types of registers: the configuration register, the transfer register, and the program register.

### CONFIGURATION REGISTER (0x000)

The configuration register is located at Address 0x000. This register is used to configure the serial interface, and it contains only four active bits in the upper nibble. The lower nibble is not connected and is held in reserve. Actively mirroring the data between the upper and lower nibble is recommended. By doing so, any loss of synchronization and directional information can be easily restored by writing to Address 0x000. Additionally, it enables the chip to be soft reset and configured in a known state, regardless of which direction data is currently being shifted. This ensures positive attention by the device if a fault condition occurs.

#### Bit 7—SDO Active

Bit 7 must be mirrored by the user in Bit 0. This bit is responsible for activating SDO on devices that include this pin. If the device does not include an SDO pin, setting this bit has no effect. If this bit is cleared, then SDO is inactive and read data is routed to the SDIO pin. If this bit is set, read data is placed on the SDO pin, if so equipped. The default for this bit is low, making SDO inactive.

#### Bit 6—LSB First

Bit 6 must be mirrored by the user in Bit 1. This bit is responsible for the order of information being sent and received. If this bit is clear, then data is processed MSB first. If this bit is set, then data is processed LSB first. In addition to the order of data shifting, Bit 6 controls the direction of auto-incrementing of the internal address pointer. If this bit is clear, that is, MSB-first mode, the internal address counter is decremented for each new datum processed. Contrarily, if this bit is set for LSB-first mode, the internal address counter is incremented for each new datum processed. The default for this bit is cleared, resulting in MSB-first operation.

#### Bit 5—Soft Reset Control

Bit 5 must be mirrored by the user in Bit 2. This bit is the soft reset control. The default for this bit is clear; however, when set high by the user, a chip soft reset is initiated. The soft reset returns all default values to the memory map registers except the configuration register (0x000). Values that have no defaults remain in the state last programmed by the user. Once the soft reset is processed, this bit is cleared, indicating that the reset process is complete.

#### Bit 4—Reserved

Bit 4 must be mirrored by the user in Bit 3. This bit defaults to 1 and cannot be changed.

**Table 3. Configuration Register 0x000**

Bit	Name	Description	Default
Bit 7	SDO active	When set, causes SDO to become active (if present). When clear, the SDO pin remains in tristate and all read data is routed to the SDIO pin.	Clear. SDIO is used for both input and output.
Bit 6	LSB first	When set, causes input and output data to be oriented as LSB first and addressing increments. When this bit is clear, data is oriented as MSB first and addressing decrements.	Clear. MSB first and decrementing addressing.
Bit 5	Soft reset	When set, the chip enters a soft reset mode, restoring any default values to internal registers. Registers with no default are not changed. Once this is complete, the state machine clears this bit.	Clear. On-chip power up, any register with a default is set.
Bit 4	Reserved	Default cannot be changed.	Set.

### TRANSFER REGISTER (MASTER-SLAVE LATCHING) (0x0FF)

It is desirable for many of the registers in the register map to be buffered with master and slave latches. Buffering enhances the ability to synchronize multiple devices in a system and aids in writing configurations that maybe dependent on values written in other parts of memory. Depending on the design, some registers may be buffered in this manner. Some registers are never buffered, such as 0x000, 0x004, 0x005, and 0x0FF, because they require immediate response for program and control purposes. (Consult the device data sheet to determine which registers are buffered.)

Regardless of buffering, the SPI port is responsible for placing information in the registers. However, for registers that are buffered, a transfer must be initiated to move the data to the slave registers. There are two defined mechanisms that cause the data to be transferred from the master register to the slave register. Unbuffered latches take effect immediately once received by the SPI state machine.

On some devices, the transfer bit may be located higher in memory if the device supports unique device-specific features. In these cases, the functionality of the transfer bit is the same; only the location is different. See the device data sheet for details.

**Bit 0—Software Transfer**

A software transfer is initiated by setting Bit 0 of this register as shown in Figure 10. When the state machine recognizes that this bit is set, it generates an internal transfer signal that moves data from the master register to the slave register. When complete, the state machine clears this bit, allowing the user to determine if the transfer has occurred. It is recommended that all other registers be configured as desired before initiating a transfer. Once the masters have been set, the last instruction should cause the data to be transferred. Data are maintained in the masters indefinitely, as long as power is applied. Therefore, it is possible to set up many chips independently and initiate a transfer to occur simultaneously across multiple chips by broadcasting the transfer command to all chips at the same time. Broadcasting can be achieved by bringing all CSB lines low at the same time, which sends the same data to all chips at once.

**Bit 7—Enable Hardware Transfer**

Not all devices support a hardware transfer mechanism (see the device data sheet to determine applicability). Bit 7 of this register is assigned the purpose of enabling hardware synchronization. If Bit 7 is clear, the default software synchronization is enabled. If this bit is set, transfer control is passed to the specified external pin (see Figure 13).

**CHIP ID (0x001)**

Register 0x001 is the chip ID register, a read-only register that returns the unique chip identifier that is coded during the design process, which typically indicates the child ID or grade of the device. This serves to identify which die is used in the package when multiple grades or options exist (see the device data sheet to determine the correct ID).

**CHIP GRADE (0x002)**

Register 0x002 is the chip grade register. This optional register may or may not contain end-user device information (see the device data sheet to determine if this register is supported and what the values mean).

**DEVICE INDEXING (0x004 AND 0x005)**

Register 0x004 and Register 0x005 are used for indexing individual converters on the same die. Register 0x005 references the lower-order devices ADC0 through ADC3, while 0x004 references the upper order devices ADC4 through ADC7. If there is only one ADC in the package, this register is not used. However, if there are several ADCs, this register must be used to indicate which device is being written to or read from. During a write process, more than one device at a time can be written by setting multiple bits in these registers that correspond to the

ADC channels to be written high. During a read process, only one bit at a time is recommended to be set high to prevent confusion over which ADC is currently placed on the read bus. Circuitry on-chip prevents bus contention, but the channel selected for readback is not known unless only one ADC at a time is enabled.

**Bit 7 to Bit 4—Auxiliary Devices**

The upper nibble is used to enable other devices that may be on-chip, such as clock generators or secondary converters.

**Bit 3 to Bit 0—Main Converters**

The lower nibble is used to enable up to four ADCs. Because there are two registers, a total of eight ADCs can be accessed.

**Writing**

Because the ADC enables are not decoded, it is possible to write to multiple devices at one time. To accomplish this, set Bit 0 through Bit 3 to enable writing to the selected device. It is possible to write to a subset of these registers by setting only those bits that correspond to the desired target converters. If both 0x004 and 0x005 are used, bit fields in both registers can be set to write to any or all of the ADCs (0 through 7) as well as any or all of the auxiliary devices.

**Reading**

When reading from devices, only one device at a time can be placed on the serial bus. No damage results if multiple devices are enabled, but the results may be indeterminate. Therefore, care must be taken to enable only one device at a time during readback operations.

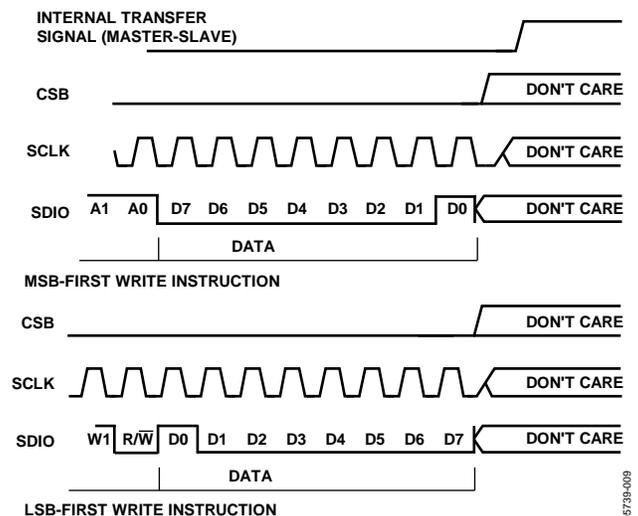


Figure 12. Internal Latching Sequence

05779-009



**PLL Control (0x00A)**

Register 0x00A is used to enable and control an on-chip PLL that may be used to generate a sample clock.

**Bit 7—PLL Locked**

This bit is controlled by the internal hardware and is set when the PLL is locked. If this bit is clear, the chip has not yet locked.

**Bit 6—PLL Auto**

When this bit is set, the PLL automatically chooses the most appropriate PLL setting for the specified divider.

**Bit 5 to Bit 0**

Set to the PLL divide ratio plus 1.

**Clock Divider (0x00B)**

Register 0x00B is used to divide the applied clock to a lower rate for the encode. If set to all 0s, the divider is bypassed. Otherwise, the divide ratio is the value in the register plus 1.

**Enhancement Modes (0x00C)**

Register 0x00C controls enhancement modes.

**Bit 7 to Bit 4—Reserved****Bit 3 to Bit 2—Chop Enable**

Chopping is used to improve noise performance at or near dc.

If Bit 3 to Bit 2 are set to:

- 00, internal chopping is disabled.
- 01, Chopping Mode 1 is enabled.
- 10, Chopping Mode 2 is enabled.
- 11, Chopping Mode 3 is enabled.

(See the device data sheet for details.)

**Table 6. Enhancement Modes, Register 0x00C, Bits[3:2]**

Bit 3 to Bit 2	Chopping Modes
00	No chopping
01	Enable Chopping Mode 1
10	Enable Chopping Mode 2
11	Enable Chopping Mode 3

**Bit 1 to Bit 0—Shuffle Mode**

Shuffling is used to improve the linearity of the ADC transfer function.

If Bit 1 to Bit 0 are set to:

- 00, internal shuffling is disabled.
- 01, Shuffling Mode 1 is enabled.
- 10, Shuffling Mode 2 is enabled.
- 11, Shuffling Mode 3 is enabled.

**Table 7. Enhancement Modes, Register 0x00C, Bits[1:0]**

Bit 1 to Bit 0	Shuffle Modes
00	No shuffling
01	Enable Shuffle Mode 1
10	Enable Shuffle Mode 2
11	Enable Shuffle Mode 3

**Output Test Modes (0x00D)**

Register 0x00D enables available test modes (see the device data sheet to determine what modes are supported). The default setting for this register is 0x00; however, when this register is set to one of the documented settings, the ADC data is replaced with test mode data. For Test Modes numbered 1, 2, 3, 5, and 6, the output format is determined by the setting of Register 0x014. All other output patterns provide logical output sequences and are not affected by the output format setting of Register 0x014.

**Bit 7 to Bit 6—Sequencing**

These bits are used in conjunction with Test Mode 8 defined by Bit 3 to Bit 0.

If these bits are set to:

- 00, then the test pattern stored in 0x019 and 0x01A is statically placed on the output.
- 01, the pattern alternates between the pattern stored in User Pattern 1 (0x019 and 0x01A) and User Pattern 2 (0x01B and 0x01C).
- 10, User Pattern 1 is placed on the output for one conversion cycle. Then the output is set to all 0s.
- 11, User Pattern 1 is placed on the output followed by User Pattern 2 on the next encode cycle. Further conversion cycles result in all 0s as determined by the output data format.

**Bit 5—PN23 Reset**

Bit 5 controls the reset long PN sequence (PN23). While this bit is set, the PN sequence is held in reset. When this bit is cleared, the PN sequence resumes from the seed value. The seed value is 0x003AFF.

**Bit 4—PN9 Reset**

Bit 4 controls the reset short PN sequence (PN9). While this bit is set, the PN sequence is held in reset. When this bit is cleared, the PN sequence resumes from the seed value. The seed value is 0x000092.

**Bit 3 to Bit 0—Test Modes**

When these bits are set to:

- 0000, the device functions as a normal ADC.
- 0001, the output is set to digital midscale.
- 0010, the output is set to +FS.
- 0011, the output is set to -FS.
- 0100, the output is set to an alternating checkerboard pattern.

- 0101, the output is set to a PN23 sequence, based on ITU 0.150 using the equation  $x^{23} + x^{18} + 1$ . The seed value is 0x003AFF. (See the device data sheet for applicable deviations.)
- 0110, the output is set to a PN9, based on ITU 0.150 using the equation  $x^9 + x^5 + 1$ . The seed value is 0x000092. (See the device data sheet for applicable deviations.)
- 0111, the output words toggle between all 1s and 0s.
- 1000, the output is set to the user mode, controlled by Bit 7 and Bit 6. If the output is in user mode 0x08 and Bit 7 and Bit 6 are set to 00, the pattern stored in the user pattern memory is statically placed on the output. If set to 01, the output toggles between User Pattern 1, stored in 0x019 and 0x01A, and User Pattern 2, stored in 0x01B and 0x01C. If set to 10, User Pattern 1 is placed on the output for one conversion cycle; then the output is set to all 0s. If set to 11,

User Pattern 1 and User Pattern 2, on the next encode cycle, are placed on the output. Further conversion cycles result in all 0s as determined by the output data format.

- 1001, the output is placed in a 1/0 bit toggle mode for serial output testing. This forces an alternating 1/0 transition on the serial output stream.
- 1010, the first half of the bits are set to 0 and the last half of the bits are set to 1. The cycle repeats for the next word frame (see Table 8 for details).
- 1011, the first bit of the serial word is set high and the following bits in the word are set low.
- 1100, the serial words shown in Table 8 are shifted.

Bit Mode 1101 and Bit Mode 1110 are reserved for future use.

Bit Mode 1111 is reserved for chip-specific test requirements.

**Table 8. Output Test Modes, Register 0x00D**

Output Test Mode <sup>1</sup>	Pattern	Word 1 <sup>2</sup>	Word 2 <sup>2</sup>	Subject to Data Format Select	Notes
0000	Off	Not applicable	Not applicable	Yes	
0001	Midscale short	1000000000000000	Not applicable	Yes	Offset binary code shown
0010	+FS short	1111111111111111	Not applicable	Yes	Offset binary code shown
0011	-FS short	0000000000000000	Not applicable	Yes	Offset binary code shown
0100	Checkerboard	1010101010101010	0101010101010101	No	
0101	PN sequence long	Not applicable	Not applicable	Yes	PN23 <sup>3</sup> ITU 0.150 $x^{23} + x^{18} + 1$
0110	PN sequence short	Not applicable	Not applicable	Yes	PN9 <sup>3</sup> ITU 0.150 $x^9 + x^5 + 1$
0111	1/0 word toggle	1111111111111111	0000000000000000	No	
1000	User input	Register 19 to Register 1A	Register 1B to Register 1C	No	
1001	1/0 bit toggle	1010101010101010	Not applicable	No	Useful in serial output mode
1010	1× sync	0000000011111111	Not applicable	No	Lower resolution truncates both a leading and a trailing digit
1011	1 bit high	1000000000000000	Not applicable	No	Useful in serial output mode
1100	Mixed-frequency	101000110011 (12 bit) 1001100011 (10 bit) 10100001100111 (14 bit) 10100011 (8 bit)	Not applicable	No	Useful in serial output mode
1101	Reserved	Not applicable	Not applicable	Not applicable	Not applicable
1110	Reserved	Not applicable	Not applicable	Not applicable	Not applicable
1111	Chip specific	Chip specific	Chip specific	Chip specific	Chip specific

<sup>1</sup> All devices may not support all modes. See the device data sheet for details.

<sup>2</sup> Truncated from the right for lower resolutions.

<sup>3</sup> See the device data sheet for applicable deviations.

**Built-In Self Test (0x00E)**

Register 0x00E configures and enables the built-in self test (BIST) functions. The BIST is a user feature that provides a high degree of confidence that the core process of the chip is performing as intended. BIST provides a simple means of determining, in a pass/fail manner, if the device is functioning. The results of the BIST are available in 0x024 and 0x025, the multiple input status register (MISR).

The BIST concept is a simple one. A PN sequence is fed to the digital block of the converter. The output of the digital block is added to an accumulator that was cleared at the start of the BIST cycle. The accumulated result consists of the sum of all PN sequences passed through the digital block. If the converter core is functioning properly, it responds exactly the same every time it is called. Therefore, the results should be consistent.

The results are placed in the MISR registers found at 0x024 and 0x025. The user can read these registers to determine if the digital section of the chip is functioning properly. This is done by comparing the values read with the values stored in the test code. Because the digital back end has many different programming options, there is no single value that represents a correct response. Instead, once the user has determined the configuration, the value from this register can be read on a working device to determine the correct response. All working devices in the specified configuration provide the same results. A different result indicates a fault.

**Bit 7 to Bit 3—Reserved****Bit 2—BIST Init**

Bit 2 is the BIST Init bit. If low, the MISR is not cleared before the BIST cycle is initiated. If this bit is high, the MISR is cleared prior to the BIST cycle. This allows several tests to be cascaded and the final results to be viewed rather than having to view each individual test.

**Bit 1 to Bit 0—BIST Mode**

If the bit pattern is:

- 00, BIST mode is disabled and the chip operates normally.
- 01, BIST mode 1 is enabled.

When BIST Mode 1 is set, the internal digital stream of the ADC is stimulated with a pseudorandom data stream and the output is accumulated in the MISR registers (24h and 25h). Any configuration settings that change data (offset or gain, for example) or reformat data (offset binary or twos complement, for example) affect the accumulation. Because the pseudorandom sequence is predictable, the accumulated value is always the same for any given configuration. This allows for a high degree of confidence that the digital back end is fully functional. The integration period is fixed at 256 encode cycles. After the BIST cycle is complete, this bit is cleared, unless Bit 2 is clear.

Note that 10 and 11 are reserved for future BIST modes.

**Analog Input (0x00F)**

Register 0x00F configures the analog input.

**Bit 7 to Bit 4—Bandwidth (Low-Pass)**

Bit 7 to Bit 4 determine the corner frequency or the on-chip low-pass filter. Note that 0000 is the default bandwidth, as specified in the device data sheet. Alternate bandwidths are defined with values 0001 through 1111. All options may not be available. See the device data sheet for options available.

**Table 9. Analog Input Bandwidth, Register 0x00F, Bits[7:4]**

Bit 7 to Bit 4	Bandwidth Mode
0000	Default bandwidth.
0001 through 1111	Alternate bandwidth choices.

**Bit 3—Reserved****Bit 2—Analog Disconnect**

Bit 2 is set to disconnect the analog input from the remainder of the ADC channel. When this bit is clear, the converter behaves normally. However, if this bit is set, the converter continues to operate, but with the analog input disconnected from the front end of the circuit. This enables the user to determine the amount of internal noise due to the converter, for applications that need this information.

**Bit 1—Common-Mode Input Enable**

Bit 1 enables any common-mode circuitry associated with the analog input of the ADC (see the device data sheet for additional details of application and functionality).

**Bit 0—Single Ended**

Bit 0 is set if the input is single ended, for a device that otherwise has a differential input, to enhance performance.

**Offset Adjust (0x010)**

Register 0x010 allows the offset of the device to be tweaked. The purpose of this register is to provide sufficient offset to move thermal noise off midscale. This is typically implemented as a digital offset, and the range for this adjustment is found in the device data sheet. The default of this register is 0x00 (midscale) with representation using twos complement notation 0x7F is the most positive offset adjustment, and 0x80 is the most negative offset adjustment. An offset of positive 1 is represented as 0x01, and a negative 1 is represented as 0xFF. The actual range of this register varies by part (see the device data sheet).

**Gain Adjust (0x011)**

Register 0x011 allows the gain of the device to be adjusted. The actual range and options vary by device (see the device data sheet for additional details).

**Output Mode (0x014)****Bit 7 to Bit 6—Logic Type**

Bit 7 to Bit 6 control the output logic type. The setting of these bits corresponds to the type of output logic selected. These are only specified as Level Option 0 through Level Option 3, and are defined in the device data sheet. Low voltage differential signaling (LVDS) type outputs, if used, can also work with 0x015 to determine output termination and driver current. Complementary metal-oxide semiconductor (CMOS) type outputs may also work with 0x015 to determine output drive strength.

**Table 10. Output Logic Type, Register 0x014 Bits[7:6]**

Bit 7 to Bit 6	Output Logic Levels
00	Option 0
01	Option 1
10	Option 2
11	Option 3

**Bit 5—Output Multiplexer**

If Bit 5 is set, the output is muxed between two different outputs or it interleaves two ADCs on the same output in a double data rate fashion.

**Bit 4—Output Enable**

Bit 4 is the output enable. If this bit is low, the output is enabled. For CMOS/TTL devices, this places the output in high impedance state. For other logic families, the output is placed in a mode defined by the device data sheet. If an external output enable exists, then the function of this bit is defeated. If an external pin is defined as an alternate function (see the Modes (0x008) section), then this bit controls the output.

**Bit 3—Double Data Rate Enable**

Bit 3 allows fewer output pins to be used to produce the same amount of data. When this bit is set, all data bits are sent using one-half of the output bits, but clocked at twice the sample rate. The remaining output bits are unused in this mode. When this bit is clear, the converter behaves in a normal manner with all output bits being used.

**Bit 2—Output Invert**

Bit 2 inverts the outputs when set.

**Bit 1 to Bit 0—Output Coding**

Bit 1 to Bit 0 determine the output coding.

If set to:

- 00, the output is offset binary.
- 01, the output is twos complement.
- 10, the output is gray code.

11 is reserved.

Only modes supported by the individual device are recognized. (See the device data sheet.) The default is 0x00.

**Table 11. Output Format, Register 0x014, Bits[1:0]**

Bit 1 to Bit 0	Output Data Format
00	Offset binary
01	Twos complement
10	Gray code
11	Reserved

**Output Settings (0x015)**

Register 0x015 works with CMOS and LVDS modes to set output termination and output driver current levels.

**Bit 7 to Bit 4—Output Termination**

Bit 7 to Bit 4 determine the output termination options for LVDS and other controlled impedance driver outputs (see the device data sheet for more details).

**Bit 3 to Bit 0—Output Drive Current**

Bit 3 to Bit 0 determine the output drive current for various CMOS and LVDS options (see the device data sheet for more details).

**Clock Divider Phase (0x016)**

Registers 0x016 determine which phase of the clock divider is used to latch data. This can be used in conjunction with either Register 0x00B or with a PLL divider output used to supply a serial clock. The default for this register is 0x00, selecting the first phase not inverted.

**Bit 7—Phase Invert**

Bit 7 inverts the internal phase.

**Bit 6 to Bit 4—Reserved****Bit 3 to Bit 0—Phase Select**

Bit 3 to Bit 0 determine which phase is selected to drive the serial clock.

**Output Delay Adjust (0x017)**

Register 0x017 sets the fine delay in the output latch relative to when the internal output registers are strobed. Internal timing is not altered by this setting. Only the output latch is changed to compensate for any external setup and hold time issues resulting from ADC timing issues. The range of this register is specified in the device data sheet.

**Bit 7—Enable**

For this feature, Bit 7 acts as an enable. If clear, default timing is selected providing reference timing.

**Bit 6—DLL Enable**

Setting Bit 6 enables an on-chip delay locked loop (DLL) that is used in the generation of the output latch. The DLL is used to maintain optimal timing between the output data eye and the latch for that data. This is useful in applications where timing is critical and data must be optimized. If this bit is clear, the DLL is off and the delay may be manually adjusted by Bit 5 to Bit 0 when enabled with Bit 7.

**Bit 5 to Bit 0—Delay**

Bit 5 to Bit 0 represent chip-specific offset timings, with 0x00 being the most negative adjustment and 3F being the most positive.

**Reference Adjust (0x018)**

Register 0x018 allows the internal reference voltage to be selected and/or adjusted.

**Bit 7 to Bit 6—VREF Select**

Bits[7:6] determine which  $V_{REF}$  is used.

If set to:

- 00, the primary  $V_{REF}$  is connected.
- 01, the secondary  $V_{REF}$  is selected.

1x is reserved for additional reference options.

**Bit 5 to Bit 0**

Bit 5 to Bit 0 allow the internal  $V_{REF}$  to be adjusted. The adjustment range is specified in the device data sheet.

**User Test Patterns (0x019 through 0x020)**

These registers are used with test mode configurations allowing the user to specify test patterns. These are paired registers with 0x019 paired with 0x01A, 0x01B with 0x01C, 0x01D with 0x01E, and 0x01F with 0x020. The low address is the least significant byte. (See the Output Test Modes (0x00D) section of this application note.)

**Serial Data Control Channel (0x021)**

Register 0x021 is the high speed serial data control channel. It may also be used in parallel output devices to control the number of output bits that are active (Bit 2 to Bit 0).

**Bit 7—LSB First**

When this bit is set, devices using a serial port for the converter data output, shift the data LSB first. If clear (default), the MSB is shifted first.

**Bit 6 to Bit 4—Reserved****Bit 3—PLL Optimize**

Bit 3 is used to optimize PLL operations for various frequency ranges (see the device data sheet for details).

**Bit 2 to Bit 0**

These bits are used to determine the number of bits shifted in the serial frame or parallel output. If set to 000, the native number of bits of the converter are shifted. This control allows for both truncation and padding of the bit stream. For example, a 12-bit converter can be forced to appear as an 8-bit converter by setting the lower 3 bits of this register to 001. Likewise, the same 12-bit converter can be forced to look like a 16-bit converter by padding the extra bits with zeroes (support for the full range of this setting is described in the device data sheet. Not all options may be present on all devices).

**Table 12. Output Frame Length, Register 0x021, Bits[2:0]**

Bit 2 to Bit 0	Serial Output Frame Length
000	Native bit length
001	Truncate/fill to 8 bits
010	Truncate/fill to 10 bits
011	Truncate/fill to 12 bits
100	Truncate/fill to 14 bits
101	Truncate/fill to 16 bits
110	Reserved
111	Reserved

**Serial Channel Power-Down (0x022)**

Serial channel power-down is used to control the state of each serial channel in a serial output converter.

**Bit 7 to Bit 2—Reserved****Bit 1—Channel Output Reset**

When Bit 1 (CH\_OUTPUT\_RESET) is selected for either a data channel or clock channel, everything is left powered up. However, the output flip-flop, prior to the LVDS driver associated with that channel is held in reset.

**Bit 0—Channel Power-Down**

When Bit 0 (CH\_POWER\_DOWN) is selected for a data channel, the associated ADC and LVDS driver are powered down while the associated digital circuitry is held in reset. When Bit 0 (CH\_POWER\_DOWN) is selected for a clock channel, the associated LVDS driver is powered down, and the associated digital circuitry is held in reset.

**MISR Registers (0x024 Through 0x025)**

Register 0x024 is the multiple input signature register (MISR) least significant byte. Register 0x025 is the MISR most significant byte. The MISR is a multiple input signature register. This register is used in conjunction with the BIST (0x00E). This register is a mirror of the core MISR and is read only.

**Features (0x02A)****Bit 7 to Bit 1—Reserved****Bit 0—Overrange Enable**

When Bit 0 is set, the overrange pin is disabled. When clear, the overrange operates normally.

**High-Pass (0x02B)**

Register 0x02B configures the high-pass filter.

**Bit 7, Bit 5 to Bit 3—Reserved****Bit 6—Tune**

Bit 6 is used to calibrate either the high-pass or the low-pass on-chip filters. Setting this bit initiates the bandwidth calibration process. Consult the device data sheet to determine which filters are calibrated and additional details.

**Bit 2 to Bit 0—Bandwidth (High-Pass)**

Bit 2 to Bit 0 determine the corner frequency or the on-chip high-pass filter. Note that 000 is the default bandwidth and corresponds to dc coupling. Alternate bandwidths are defined with values 001 through 111. Not all options may be available (see the device datasheet for options available).

**Table 13. High-Pass Filter Select, Register 0x02B, Bits[2:0]**

Bit 2 to Bit 0	Bandwidth Mode
000	Default bandwidth (dc)
001 through 111	Alternate high-pass choices

**Analog In (0x02C)****Bit 7 to Bit 1—Reserved****Bit 0—Input Impedance**

Bit 0 allows one of two input impedances to be selected (see the device data sheet for details).

**Cross Point Switch (0x02D)**

This function provides an analog cross point switch that may be used for connecting the analog input to the core ADC or to route various analog inputs to various auxiliary analog outputs as defined in the device data sheet.

## PROGRAMMING EXAMPLE

Programming tools are available to assist in the development of code for SPI devices. A user may wish to access the features available with SPI control, but not have access to a full featured SPI controller. If this is the case, consult the [AN-812 Application Note](#) for a low cost alternative to a full featured controller.

A software tool is also available (see [www.analog.com/FIFO](http://www.analog.com/FIFO)) for controlling the devices used on the corresponding evaluation board. This tool allows the registers to be configured to determine the optimal device configuration for the end application. In addition, once this process is complete, the software tool generates two files useful for programming the devices. The first file format is a pseudocode format.

```
write(0, 18); //configure serial interface for MSB first
write(5, 3); //set Devices-Index to program ADC Channels 0 and 1
write(18, 80); //set vref to option 2 and adjustment to all zeros
write(14, 10); //set output_mode to level option 0, disable output MUX, enable output and
               //offset binary
write(17, 83); //set output_delay to enable and set to delay value of 3
write(FF, 1); //write transfer bit (for configurations that require a manual transfer)

write(5, 2); //set Device-Index to program ADC Channel 1
write(10, 3); //set offset to 3 (for Channel 1 only)
write(FF, 1); //write transfer bit (for configurations that require a manual transfer)

Write(5, 4); //set Devices Index to program ADC Channel 2
write(10, 9); //set offset to 9 (for Channel 2 only)
write(FF, 1); //write transfer bit (for configurations that require a manual transfer)
```

This can be added to a C language project to set up the appropriate writes and reads to ensure the device is configured per the settings in the evaluation software. To use this pseudocode, the user need only supply the hardware-specific read and write functions associated with their SPI controller. The example code in this section outlines a sample program sequence for the devices.

The second file format is assembly code that can be used with the microcontroller described in the [AN-812 Application Note](#). (See [AN-812](#) for additional details on the usage of this output.)

For additional details on using these tools, see the [AN-878 Application Note](#), *High Speed ADC SPI Control Software*.

# CONTROL REGISTER

Table 14. Control Register Map

Address <sup>1</sup> , Register	Bit 7 (MSB)	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0 (LSB)	Default Value <sup>1</sup>	Comments	
00, CHIP_PORT_CONFIG	SDO active <sup>2</sup>	LSB first	Soft reset	Should be set. Do not clear.					18h	The nibbles should be mirrored by the user so that LSB-first or MSB-first mode registers correctly regardless of shift mode.	
01, CHIP_ID	8-bit chip ID; Bits[7:0]								Read only	Default is unique chip ID, different for each device. This is a read-only register. (See device data sheet for more details.)	
02, CHIP_GRADE	8-bit child ID								Read only	Read only. Child ID used to differentiate graded devices. (See device data sheet for more details.)	
04, DEVICE_INDEX_B	AUX 7	AUX 6	AUX 5	AUX 4	ADC 7	ADC 6	ADC 5	ADC 4	FFh	Bits are set to determine which device on-chip receives the next write command. The default will be all devices on-chip.	
05, DEVICE_INDEX_A	AUX 3	AUX 2	AUX 1	AUX 0	ADC 3	ADC 2	ADC 1	ADC 0	FFh	Bits are set to determine which device on-chip receives the next write command. The default is all devices on-chip.	
08, MODES	External power-down enable	External power-down mode 00: full power-down 01: standby 10: normal mode (output disabled) 11: normal mode (output enabled)			Function bypass	Internal power-down mode 0: chip run 1: full power-down 2: standby 3: reset 4: ADC power-down 5: analog front-end power-down 6: reserved 7: reserved			00h	Determines various generic modes of chip operation.	
09, CLOCK	Reserved for additional clock input support					PLL enable	Clock boost	Duty cycle stabilize	01h		
0A, PLL_CONTROL	PLL locked	PLL auto	PLL multiplier; Bits[5:0]						00h	Configures on-chip PLL by enabling and setting multiplier. MSB is set when the PLL is locked.	
0B, CLOCK_DIVIDE	Clock divider; Bits[7:0]								00h	The divide ratio is the value plus 1.	
0C, ENHANCE	Reserved	Reserved	Reserved	Chop enable 0: off 1: Mode 1 2: Reserved 3: Reserved		Shuffle mode 0: off 1: Mode 1 2: Reserved 3: Reserved				Shuffle mode determines how shuffling is performed. Chopping determines how the input is processed to improve noise near dc.	
0D, TEST_IO	User test mode 00: single 01: alternate 10: single once 11: alternate once		Reset PN long gen	Reset PN short gen	Output test mode 0: off 1: midscale short 2: +FS short 3: -FS short 4: checkerboard output 5: PN23 sequence 6: PN9 7: 1/0 word toggle 8: User input 9: 1/0 bit toggle 10: 1x sync 11: 1 bit high 12: mixed-bit frequency (format determined by OUTPUT_MODE)					00h	When set, the test data is placed on the output pins in place of normal data.
0E, TEST_BIST					BIST init	Reserved	BIST enable	00h	BIST mode configuration		
0F, ADC_INPUT	Low-pass filter bandwidth 0: default Bits[1:15]: alternate corner frequencies (See device data sheet for details)				Analog disconnect	Common-mode input enable	Single ended	00h			
10, OFFSET	8-bit device offset adjustment; Bits[7:0]								80h	Device offset trim	
11, GAIN	8-bit device gain adjustment; Bits[7:0]								00h	Device gain trim	

Address <sup>1</sup> , Register	Bit 7 (MSB)	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0 (LSB)	Default Value <sup>1</sup>	Comments
14, OUTPUT_MODE	0: Level Option 0 1: Level Option 1 2: Level Option 2 3: Level Option 3		Output mux enable (interleave)	Output enable	DDR enable	Output invert	0: Offset binary 1: Twos complement 2: Gray code 3: Reserved		Device specific	Configures the outputs and the format of the data.
15, OUTPUT_ADJUST	Output driver termination; Bits[7:4]				Output driver current; Bits[3:0]				Device specific	Determines LVDS or other output properties. Primarily functions to set the LVDS span and common-mode levels in place of an external resistor.
16, OUTPUT_PHASE	Output polarity				Output clock phase adjust; Bits[3:0]				00h	On devices that utilize clock divide, determines which phase of the divider output is used to supply the output clock. Internal latching is unaffected.
17, OUTPUT_DELAY	Enable	DLL enable	6-bit output delay; Bits[5:0]						00h	This sets the fine output delay of the output clock but does not change internal timing.
18, VREF	V <sub>REF</sub> select 0: primary (0) 1: secondary (1) 2: Option 2 3: Option 3		6-bit internal V <sub>REF</sub> adjustment; Bits[5:0]						20h	Select and/or adjust the V <sub>REF</sub> .
19, USER_PATT1_LSB	B7	B6	B5	B4	B3	B2	B1	B0	00h	User-Defined Pattern 1 LSB.
1A, USER_PATT1_MSB	B15	B14	B13	B12	B11	B10	B9	B8	00h	User-Defined Pattern 1 MSB.
1B, USER_PATT2_LSB	B7	B6	B5	B4	B3	B2	B1	B0	00h	User-Defined Pattern 2 LSB.
1C, USER_PATT2_MSB	B15	B14	B13	B12	B11	B10	B9	B8	00h	User-Defined Pattern 2 MSB.
1D, USER_PATT3_LSB	B7	B6	B5	B4	B3	B2	B1	B0	00h	User-Defined Pattern 3 LSB.
1E, USER_PATT3_MSB	B15	B14	B13	B12	B11	B10	B9	B8	00h	User-Defined Pattern 3 MSB.
1F, USER_PATT4_LSB	B7	B6	B5	B4	B3	B2	B1	B0	00h	User-Defined Pattern 4 LSB.
20, USER_PATT4_MSB	B15	B14	B13	B12	B11	B10	B9	B8	00h	User-Defined Pattern 4 MSB.
21, SERIAL_CONTROL	LSB first				PLL optimize	000: normal bit stream 001: 8 bits 010: 10 bits 011: 12 bits 100: 14 bits 101: 16 bits			00h	Serial stream control. Default causes MSB first and the native bit stream.
22, SERIAL_CH_STAT							Ch output reset	Ch power-down	00h	Used to power down individual sections of a converter (local).
24, MISR_LSB	B7	B6	B5	B4	B3	B2	B1	B0	00h	Least significant byte of MISR (read-only).
25, MISR_MSB	B15	B14	B13	B12	B11	B10	B9	B8	00h	Most significant byte of MISR (read-only).
2A, FEATURES							OVR alternate pin	OVR output enable	00h	Auxiliary feature set control.
2B, HIGH_PASS		Tune				Corner frequency 000: dc 000 through 111: Alternate corner frequencies			00h	High-pass filter control.
2C, AIN								Input impedance	00h	Analog input control.
2D, CROSS_POINT									00h	Analog input cross point switch.
FF, DEVICE_UPDATE	Enable HW transfer							SW transfer	00h	Synchronously transfers data from the master shift register to the slave.

<sup>1</sup> Hexadecimal.<sup>2</sup> Not supported on most devices.

**NOTES**