

Introduction

The HI7190 Serial Interface is designed to be compatible with many industry standard synchronous transfer formats including the Motorola 6805/11 series SPI and Intel 8051 SSR protocols. The advantage of HI7190 Serial Interface is its flexibility. However, flexibility has its price - complexity. The complexity of the HI7190 Serial Interface may lead to confusion for first time users. This Technical Brief discusses general serial interface issues associated with using the HI7190. It is assumed the reader has read the HI7190 data sheet and understands the basic operational details of the device.

Serial Interface Functionality

Communication with the HI7190 occurs in what are called communication cycles and each communication cycle contains two phases. Figure 1 shows that the first phase of every communication cycle is the writing of an instruction byte. The first eight SCLK rising edges of each communication cycle are used to write the instruction byte into the Instruction Register (IR). The instruction byte provides the HI7190 Serial Interface with information regarding the data transfer in phase 2 of the communication cycle. The remaining SCLK edges are used for phase 2 of the communication cycle. Phase 2 is the actual data transfer between the HI7190 and the processor.

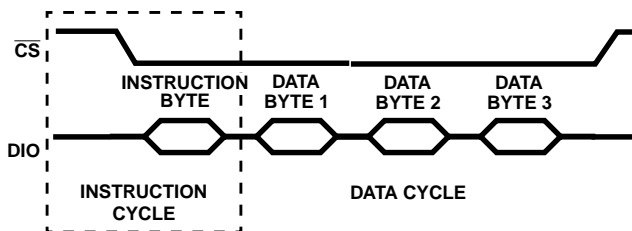


FIGURE 1. BYTE COMMUNICATION CYCLE

Phase 2 of the communication cycle is a transfer of 1, 2, 3, or 4 data bytes as determined by the instruction byte. This means that there are several ways to read or write a multi-byte register. For example, a 3 byte data transfer can be completed in one communication cycle by writing the IR to transfer 3 bytes of data. Or, the same three byte data transfer can be spread over three communication cycles by writing the IR to transfer 1 byte of data with three separate communication cycles. Finally, this same three byte data transfer can be accomplished over two communication cycles as well by writing the IR for a 2 byte transfer followed by writing the IR for a 1 byte transfer or vice versa! Flexibility is increased even further since the IR contains the starting byte address of the register from which data is transferred.

It is important to note that the instruction byte written during phase 1 of the communication cycle describes the number of bytes to be transferred during phase 2. This means that the number of bytes transferred as described in the instruction byte does not include the IR write of phase 1.

Normally using one communication cycle in a multi-byte transfer is the preferred method. However, single byte communication cycles are useful to reduce CPU overhead when a register access requires only one byte of data being transferred. For example, the HI7190 can be put into the sleep mode by a single byte write to the Control Register (CR).

The processor must maintain synchronism with the HI7190, or the internal communication controller will not be able to recognize further instructions. For example, if the processor sends an instruction byte for a two byte read and then pulses SCLK for a 3 byte read (24 falling edges), communication synchronization is lost. This is because the first 16 SCLK falling edges after the IR write caused data to be read out of the HI7190. But according to the instruction byte, that was the end of the communication cycle. So the HI7190 interpreted the last 8 SCLK rising edges as the next instruction byte. This would cause communication problems as the HI7190 believes it is in the middle of another communication cycle; but the processor believes a communication cycle has just ended. The only way to recover from this type of error is to reset the HI7190 and begin with a new communication cycle. Recall that resetting the HI7190 may change the entire circuit configuration and therefore, is not desired. Maintaining communication synchronization is critically important.

The complete set of IR bit specifications is as follows:

MSB	6	5	4	3	2	1	LSB
\bar{R}/W	MB1	MB0	FSC	A3	A2	A1	A0

R/W - Bit 7 of the Instruction Register determines whether phase 2 of the communication cycle will be a read or write operation. If \bar{R}/W is logic 1, a write transfer will occur in phase 2 of the communication cycle. If \bar{R}/W is logic 0, a read transfer will occur in phase 2 of the communication cycle.

MB1, MB0 - Bits 6 and 5 of the Instruction Register determine the number of bytes that will be transferred during phase 2 of the communication cycle. Any number of bytes from 1 to 4 is allowed. See the HI7190 data sheet for specific bit decodes.

FSC - Bit 4 is used to determine whether a Positive Full Scale Calibration Register I/O transfer (FSC = 0) or a Negative Full Scale Calibration Register I/O transfer (FSC = 1) is being performed if A3 and A2 are both set (11). If either A3 or A2 is reset (0), FSC is a don't care.

A3, A2, A1, A0 - Bits 3 and 2 (A3 and A2) of the Instruction Register determine which Internal Register will be accessed, while bits 1 and 0 (A1 and A0) determine which byte of that register will be accessed first. See the HI7190 data sheet for specific address requirements.

In addition to understanding the communication cycle concept it is important to note that the three least significant bits of the CR have a large impact on the serial interface. The least significant bit (CR<0>) determines whether one line or two line protocol is used. For one line protocol the SDIO pin is used for data input and output, while for 2 line protocol the SDIO pin is used for data input and the SDO pin is used for data output. The CR<2> and CR<1> bits are the Byte Direction (BD) and $\overline{\text{MSB}}$ bits respectively. The BD bit specifies whether data bytes will be accessed in ascending or descending order while the $\overline{\text{MSB}}$ bit specifies if each byte will be read MSB first or LSB first. Together, these bits allow the user to access data in MSB to LSB format or LSB to MSB format.

In review, each HI7190 Register can be accessed on a per byte basis or via multiple byte transfers depending on the instruction byte. Each communication cycle can be most significant bit to least significant bit order or vice versa depending on the configuration of the BD and $\overline{\text{MSB}}$ bits in the CR and the instruction byte. In addition, the HI7190 can be configured for either one line or two line protocol.

Data Transfer Format Programming

The HI7190 Serial Interface can be configured for data transfer in most significant to least significant bit position or least significant to most significant position. As previously mentioned the configuration is determined by the state of the BD bit and the $\overline{\text{MSB}}$ bit of the CR. Obviously, these bits can be written to any of four combinations, but only two of these combinations make sense. When MSB to LSB format is desired, the HI7190 must be configured for MSB first and descending byte order. When LSB to MSB format is desired, the HI7190 must be configured for LSB first and ascending byte order.

The following examples are useful in understanding the MSB to LSB data transfer.

Example 1: MSB to LSB Data Transfer - 3 Byte Read of the Data Register

The CR must be configured for descending byte direction and MSB first bit positioning. Therefore, the BD and $\overline{\text{MSB}}$ bits are programmed to logic zero.

The instruction byte required is 010x 0010 which specifies a 3 byte read of the Data Register starting at the most significant byte. The starting byte (address:10) data is driven out of the HI7190 on the falling edges of SCLK in most significant to least significant bit order. After this byte is transferred the HI7190 decrements the byte address to the next lesser significant value (address:01). This byte is transferred out on the next 8 falling SCLK edges. Again the HI7190 decrements the byte address to the next lesser significant value (address:00) and the last byte is driven out on the next 8 SCLK falling edges. The communication cycle is now com-

plete and the entire word was read MSB to LSB format in a single communication cycle.

Example 2: LSB to MSB Data Transfer - 3 Byte Read of the Data Register

The CR must be configured for ascending byte direction and LSB first bit positioning. Therefore, the BD and $\overline{\text{MSB}}$ bits are programmed to logic one.

The instruction byte required is 010x 0000 which specifies a 3 byte read of the Data Register starting at the least significant byte. The starting byte (address:00) data is driven out of the HI7190 on the falling edges of SCLK in least significant to most significant bit order. After this byte is transferred the HI7190 increments the byte address to the next greater significant value (address:01). This byte is transferred out, LSB to MSB on the next 8 falling SCLK edges. Again the HI7190 increments the byte address to the next greater significant value (address:10) and the last byte is driven out on the next 8 SCLK falling edges. The communication cycle is now complete and the entire word was read LSB to MSB format in a single communication cycle.

Example 3: Incorrect MSB to LSB Data Transfer

The CR is configured for ascending byte direction and MSB first bit positioning. That is, the BD bit is programmed to logic one and the $\overline{\text{MSB}}$ bit is programmed to logic zero.

With this configuration, multi-byte transfers cannot be completed in MSB to LSB format. The bits of each byte will be transferred in MSB to LSB format, but the data bytes will be in ascending order.

Assume a three byte Data Register read is invoked with the most significant byte being the starting byte. The starting byte (address:10) data is driven out of the HI7190 on the falling edges of SCLK in most significant to least significant bit position. After this byte is transferred the HI7190 increments the byte address to the next greater significant value (address:00). This byte is transferred out, MSB to LSB on the next 8 falling SCLK edges. Again the HI7190 increments the byte address to the next greater significant value (address:01) and the last byte is driven out on the next 8 SCLK falling edges completing the communication cycle. The data has been output as follows: DR<23:16>,DR<7:0>,DR<15:8>. Note that the MSB to LSB format is not maintained across the entire word.

Instruction Byte Quick Reference

Table 1 shows the required Instruction Register byte and Control Register bit requirements for commonly used IO functions.

TABLE 1. SERIAL INTERFACE QUICK REFERENCE TABLE

IO TRANSFER DESCRIPTION	CR<2:1>	IR (HEX)
24-Bit Data Register read, MSB to LSB.	00	42
16-Bit Data Register read, MSB to LSB (Note)	00	22
24-Bit Data Register read, LSB to MSB	11	40
16-Bit Data Register read, LSB to MSB (Note)	11	21
24-Bit Control Register write, MSB to LSB.	00	C6
24-Bit Control Register write, LSB to MSB.	11	C4
24-Bit Control Register read, MSB to LSB.	00	46
24-Bit Control Register read, LSB to MSB.	11	44
24-Bit Offset Calibration Register write, MSB to LSB.	00	CA
24-Bit Offset Calibration Register write, LSB to MSB.	11	C8
24-Bit Offset Calibration Register read, MSB to LSB.	00	4A
24-Bit Offset Calibration Register read, LSB to MSB.	11	48
24-Bit Positive Gain Calibration Register write, MSB to LSB.	00	CE
24-Bit Positive Gain Calibration Register write, LSB to MSB.	11	CC
24-Bit Positive Gain Calibration Register read, MSB to LSB.	00	4E
24-Bit Positive Gain Calibration Register read, LSB to MSB.	11	4C
24-Bit Negative Gain Calibration Register write, MSB to LSB.	00	DE
24-Bit Negative Gain Calibration Register write, LSB to MSB.	11	DC
24-Bit Negative Gain Calibration Register read, MSB to LSB.	00	5E
24-Bit Negative Gain Calibration Register read, LSB to MSB.	11	5C

NOTE: Some systems with less stringent requirements may need only 16-bit results. This IR is useful in those systems and reduces data read times.

Further Clarifications

1. When changing the IO configuration, which defaults to single data line in MSB to LSB bit order, the user must be aware that the configuration changes IMMEDIATELY after completing the write of that CR byte. This may occur in the middle of a communication cycle, depending on how the user writes the IR. It is recommended that the least significant byte of the CR be written last when reconfiguring the serial Interface of the HI7190. Another method would be to use a single byte write communication cycle when reconfiguring the serial interface.
2. It is important to realize that if a four byte transfer is requested the HI7190 requires that four bytes be transferred regardless of the number of bytes in the register. That is, if a 3 byte register is being accessed but a four byte transfer is requested, four bytes must be transferred. The first byte accessed will be transferred twice. The user should be careful not to start at an invalid byte address as Intersil has reserved the fourth byte as a test byte for some registers.

Example - The IR=11100110 (E6 hex) is a four byte write of the CR starting with the most significant byte. Assuming descending byte order and MSB first bit positioning the following transfer would occur.

1. Write CR<23:16>
2. Write CR<15:8>
3. Write CR<7:0>
4. Write CR<23:16>

The starting byte is accessed twice. Note that the internal byte address generator will NOT increment/decrement into an invalid byte address. Therefore, the only way a user can access an invalid byte is to make that byte the starting byte address. As previously stated, the user should be careful not to start at an invalid byte address as Intersil has reserved the fourth byte as a test byte for some registers.

All Intersil semiconductor products are manufactured, assembled and tested under **ISO9000** quality systems certification.

Intersil semiconductor products are sold by description only. Intersil Corporation reserves the right to make changes in circuit design and/or specifications at any time without notice. Accordingly, the reader is cautioned to verify that data sheets are current before placing orders. Information furnished by Intersil is believed to be accurate and reliable. However, no responsibility is assumed by Intersil or its subsidiaries for its use; nor for any infringements of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Intersil or its subsidiaries.

For information regarding Intersil Corporation and its products, see web site <http://www.intersil.com>