Author: Rahman Sobhan

I²C Fundamentals

Serial Interface

The name I²C is short hand for a standard Inter-IC (integrated circuit) bus. I²C is a simple protocol with low-bandwidth and short-distance for data transfer. Most I²C devices operate at speeds up to 400Kbps. The I²C is easy to use for linking multiple devices together since it has a built-in addressing scheme.

Intersil uses I²C for all optical sensor products. The I²C bus is a two wire serial bidirectional interface consisting of SCL (clock) and SDA (data). Both the wires are connected to the device supply via pull-up resistors.

The I²C protocol defines any device that sends data onto the bus as a transmitter and the receiving device as the receiver. The device controlling the transfer is a master and the device being controlled is the slave. The transmitting device pulls down the SDA line to transmit a “0” and releases it to transmit a “1”. The master always initiates the data transfer, only when the bus is not busy, and provides the clock for both transmit and receive operations. The optical sensors operate as a slave device in all applications. The serial communication over the I²C interface is conducted by sending the most significant bit (MSB) of each byte of data first.

Start Condition

During data transfer, the SDA line must remain stable while the SCL line is HIGH. All I²C interface operations must begin with a START condition, which is a HIGH to LOW transition of SDA while SCL is HIGH (see Figure 3). The optical sensors continuously monitors the SDA and SCL lines for the START condition and does not respond to any command until this condition is met (see Figure 3). A START condition is ignored during the power-up sequence.

Stop Condition

All I²C interface operations must be terminated by a STOP condition, which is a LOW to HIGH transition of SDA while SCL is HIGH (see Figure 3). A STOP condition at the end of a read/write operation places the device in its standby mode. If a stop is issued in the middle of a Data byte, or before 1 full Data byte + ACK is sent, then the serial communication of the optical sensors resets itself without performing the read/write. The contents of the array are not affected.

Acknowledge

An acknowledge (ACK) is a software convention used to indicate a successful data transfer. The transmitting device releases the SDA bus after transmitting 8-bits. During the ninth clock cycle, the receiver pulls the SDA line LOW to acknowledge the reception of the eight bits of data (see Figure 3). The optical sensors responds with an ACK after recognition of a START condition followed by a valid Identification Byte, and once again, after successful receipt of an Address Byte. The optical sensors also respond with an ACK after receiving a Data byte of a write operation. The master must respond with an ACK after receiving a Data byte of a read operation

Device Addressing

Following a START condition, the master must output a Device Address byte. The 7 MSBs of the Device Address byte are known as the device identifier. The device identifier bits of the optical sensors are internally hard-wired as “1000100”. The LSB of the Device Address byte IS defined as read or write (R/$\overline{W}$) bit. When this R/$\overline{W}$ bit is a “1”, a read operation is selected and when “0”, a write operation is selected (see Figure 1). The master generates a START condition followed by Device Address byte 1000100x (x as R/$\overline{W}$) and the optical sensors compares it with the internal device identifier. Upon a correct comparison, the device outputs an acknowledge (LOW) on the SDA line (see Figure 3).

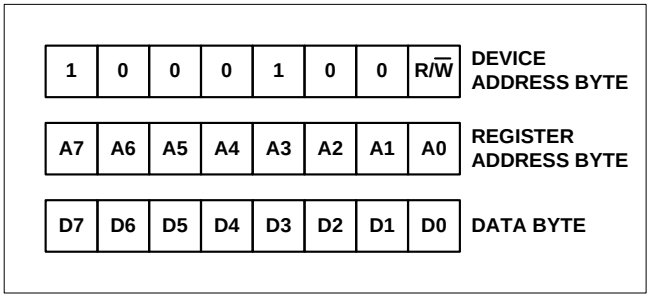

FIGURE 1. DEVICE ADDDRESS, REGISTER ADDRESS AND DATA BYTE

Write Operation

BYTE WRITE

In a byte write operation, optical sensors require the Device Address byte, Register Address byte, and the Data byte. The master starts the communication with a START condition. Upon receipt of the Device Address byte, Register Address byte, and the Data byte, the optical sensors respond with an acknowledge (ACK). Following the optical sensors data acknowledge response, the master terminates the transfer by generating a STOP condition. The optical sensors then begin an internal write cycle of the data to the volatile memory. During the internal write cycle, the device inputs are disabled and the SDA line is in a high impedance state, so the device will not respond to any requests from the master (see Figure 2). A basic write sequence for I²C follows the following order:

1. Send a start sequence
2. Send the I²C address of the slave with the R/W bit low
3. Send the register number to write to
4. Send the data byte
5. Send the stop sequence

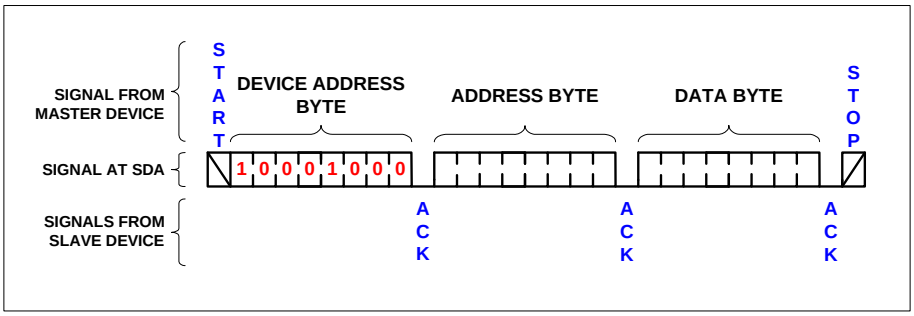

FIGURE 2. BYTE WRITE SEQUENCE

BURST WRITE

The optical sensors have a burst write operation, which allows the master to write multiple consecutive bytes from a specific address location. It is initiated in the same manner as the byte write operation, but instead of terminating the write cycle after the first Data byte is transferred, the master can write to the whole register array. After the receipt of each byte, the optical sensors responds with an acknowledge, and the address is internally incremented by one. The address pointer remains at the last address byte written. When the counter reaches the end of the register address list, it "rolls over" and goes back to the first Register Address.

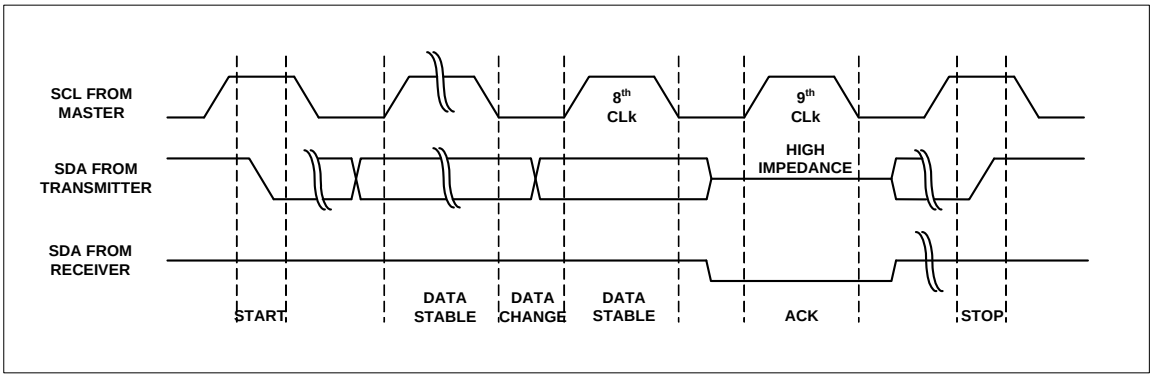

FIGURE 3. START, DATA STABLE, ACKNOWLEDGE, AND STOP CONDITION

Read Operation

Optical sensors have two basic read operations: Byte Read and Burst Read.

BYTE READ

Byte read operations allow the master to access any register location in the optical sensors. The Byte read operation is a two step process. The master issues the START condition and the Device Address byte with the R/$\overline{W}$ bit set to "0", receives an acknowledge, then issues the Register Address byte. After acknowledging receipt of the register address byte, the master immediately issues another START condition and the Device Address byte with the R/$\overline{W}$ bit set to "1". This is followed by an acknowledge from the device and then by the 8-bit data word. The master terminates the read operation by not responding with an acknowledge and then issuing a stop condition (see Figure 4). A basic read sequence for I^2C follows the following order:

1. Send a start sequence
2. Send I2C address of the slave with the R/W bit low
3. Master sends a Low for Acknowledge to verify data received
4. Send a start sequence again (repeated start)
5. Send I^2C address of the slave with the R/W bit high
6. Master sends a Low for Acknowledge to verify data received
7. Read data byte
8. Send the stop sequence

BURST READ

Burst read operation is identical to the Byte Read operation. After the first Data byte is transmitted, the master now responds with an acknowledge, indicating it requires additional data. The device continues to output data for each acknowledge received. The master terminates the read operation by not responding with an acknowledge but issuing a STOP condition (Refer to Figure 5).

For more information about the I^2C standard, please consult the Phillips™ I^2C specification documents.

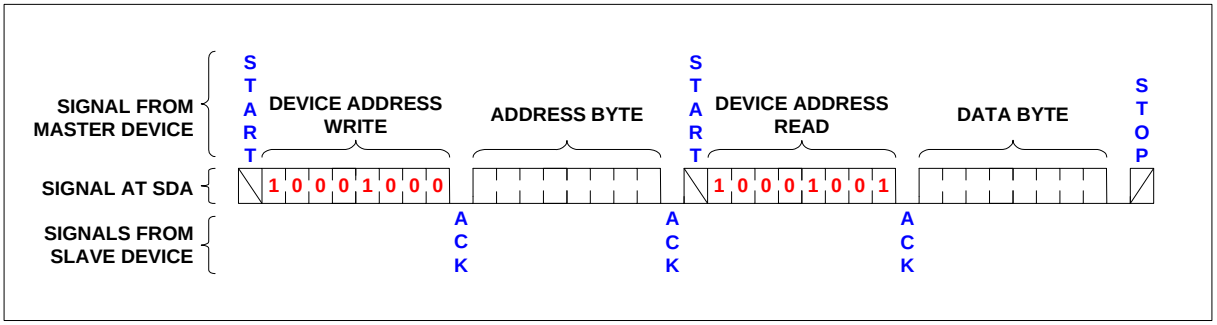

FIGURE 4. BYTE ADDRESS READ SEQUENCE

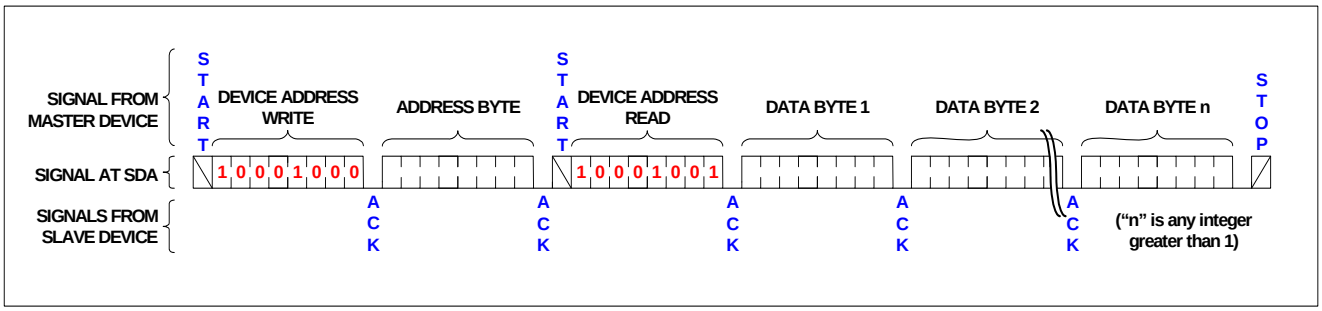

FIGURE 5. BURST READ SEQUENCE