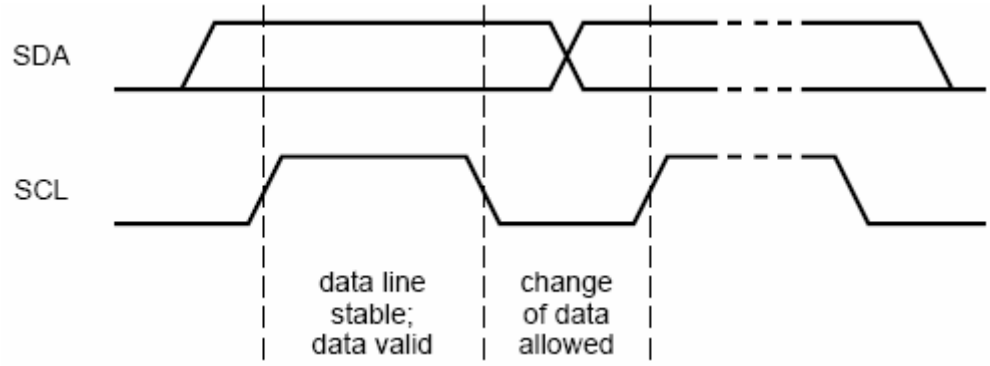
I²C basics application note

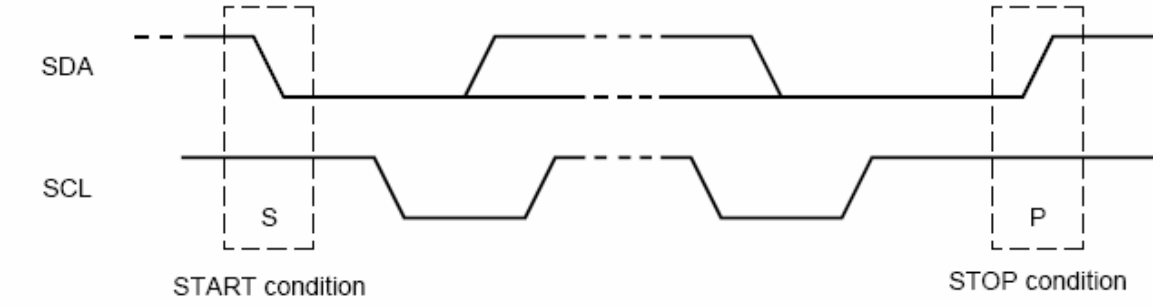
I²C (Inter-Integrated Circuit bus) is a protocol for providing communications links between integrated circuits. The I²C bus uses only two lines, SDA (serial data line) and SCL (serial clock line). These are bi-directional, open drain/collector lines, which are pulled up to +5V or +3.3V via a resistor or current source. Outputs from each I²C compatible device are connected together in a wired AND configuration, so devices communicate by pulling the bus line low. If both SDA and SCL lines are high, this indicates an idle condition and in this situation, any device is free to initiate communications. This is a true multi-master bus which uses collision detection and arbitration. A master is the device which requests and initiates communications with a slave, which is another device known by unique addresses. Data transfer is synchronized by the SCL line which is a clock generated by the master. Data transfers are 8-bit orientated with the MSB transmitted first and occur serially over the SDA line at a rate of 100kb/s, 400kb/s or 3.4Mb/s depending on the mode of operation.

Protocol for the bus is outlined below:

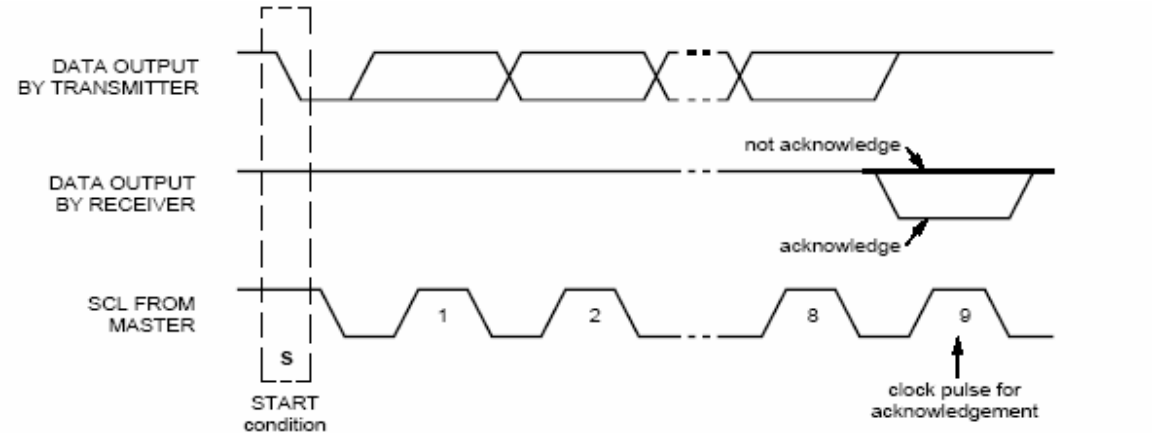
- For data on the SDA line to be considered valid, it must be stable during the **High** period of the clock, thus data may only change when the clock (SCL) line is **Low**.



- The protocol defines unique start and stop conditions:
  - A start condition is signified by a **High** to **Low** transition on the SDA line while SCL is **High**.
  - A stop condition is signified by a **Low to High** transition on the SDA line while SCL is **High**.



In order to initiate communications, the master transmits a start condition followed by the 8-bit address of the slave device. When a peripheral recognizes the address it transmits an acknowledgment by pulling the SDA line low. Transmitting a byte of data or an address takes eight clock cycles, but the master also generates a ninth cycle, during which it releases the SDA line (which returns high). This is an acknowledge clock pulse during which the receiver must pull the SDA line low and keep it low during the High period of the clock in order to acknowledge the previous byte or address. If the slave is not ready to receive more data it may produce a no acknowledge by leaving the line high. This will cause the master to either abort the transfer by producing a stop condition or suspend the transfer by producing a repeated start. A repeated start condition indicates that the bus is busy and the master retains possession.



The predefined procedures for the read and write operations, as used with the ADV740xA/ADV718xB decoders, are given below:

Write Sequence (see fig. 1. below)

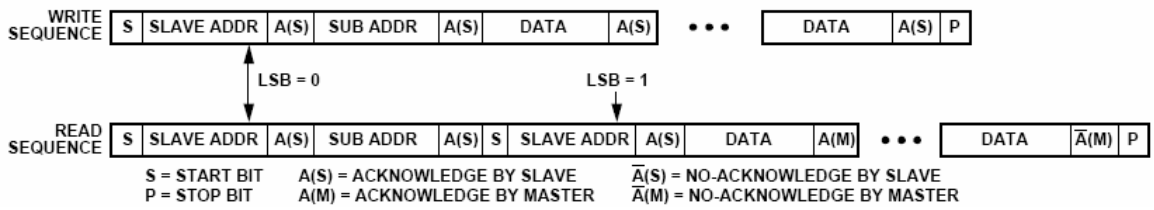
1) Send the start condition
2) Send the ADV740xA/ADV718xB slave address (0x40, ALSB = 0) / (0x42,ALSB = 1)
3) check for the acknowledge from ADV740xA/ADV718xB
4) Send the sub-address to be written to.
5) check for the acknowledge from ADV740xA/ADV718xB
8) Send the data to write to specified subaddress
9) check for the acknowledge from ADV740xA/ADV718xB
10) If No-acknowledge send the stop condition
13) Send a stop condition

Read Sequence (see fig. 1. below)

1) Send the start condition
2) Send the ADV740xA/ADV718xB slave address (0x40, ALSB = 0) / (0x42,ALSB = 1)
3) check for the acknowledge from ADV740xA/ADV718xB
4) Send the sub-address to be read from
5) check for the acknowledge from ADV740xA/ADV718xB
7) Send the start condition
8) Send the slave address = 0x41 / 0x43 for a read operation (LSB = 1)
9) check for the acknowledge from ADV740xA/ADV718xB
10) If No-acknowledge send the stop condition
11) If acknowledged read the data from specified sub-address
12) Send a No-Acknowledge
13) Send a stop condition

The sequences above are depicted graphically as shown below:

Fig.1. Read and write sequences



It should be noted that the decoders above have two possible slave addresses for both read and write operations, depending on the logic level of the ALSB, which is bit 1 of the slave address. The LSB (bit 0) sets either a read or write operation. Table 1 below summarizes this information:

Table 1

ALSB	R/W	Slave Address
0	0	0x40
0	1	0x41
1	0	0x42
1	1	0x43

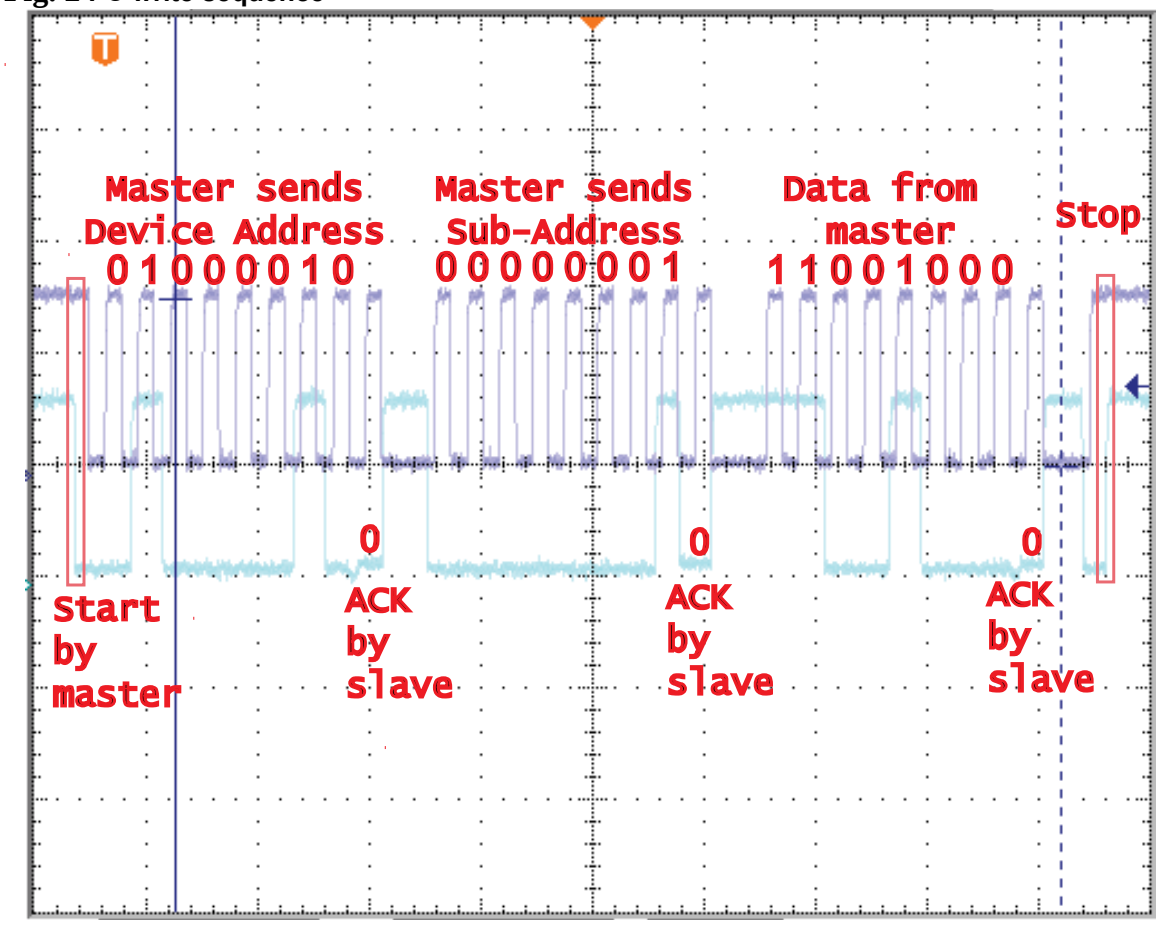
We should now be able to predict the waveforms that would be produced if we were to hook up an oscilloscope to the SDA and SCL lines during a read or write operation. Let us arbitrarily choose a value of 0xC8 to be written to sub address 1 of a slave device with address 0x42. A sub address is a location within the memory map of the addressed slave device. For this write operation, we would expect the following:

- The SDA and SCL lines should be initially High
- A start condition is issued

- During the high period of the next 8 clock pulses the master should transmit 01000010 (device address 0x42) over the SDA line
- SDA should be pulled low by the slave during the high period of the ninth clock pulse as an acknowledgement of its address
- During the high period of the next 8 clock pulses the master should transmit 00000001 (sub address 0x01) over the SDA line
- SDA should be pulled low by the slave during the high period of the ninth clock pulse as an acknowledgement of this address
- During the high period of the next 8 clock pulses the master should transmit 11001000 (Data to be written 0xC8) over the SDA line
- SDA should be pulled low by the slave during the high period of the ninth clock pulse as an acknowledgement of this data
- Stop condition is issued
- SDA and SCL lines should return High

Carrying out this write operation yielded the waveform in fig. 2 below. SCL on top and SDA below are shown overlapping for ease of timing comparison:

Fig. 2 I²C write sequence



We should now be able to read the value 0xC8 back from sub-address 01. For this read operation, we would expect the following:

- The SDA and SCL lines should be initially High
- A start condition is issued
- During the high period of the next 8 clock pulses the master should transmit 01000010 (device address 0x42) over the SDA line (See fig. (a) below)
- SDA should be pulled low by the slave during the high period of the ninth clock pulse as an acknowledgement of its address
- During the high period of the next 8 clock pulses the master should transmit 00000001 (sub address 0x01) over the SDA line (See fig. (b) below)
- Repeated start condition is issued so that the next byte will be interpreted as an address and not data to be written (See fig. (c) below).
- The LSB of the device address is set to indicate a read operation, 0x43 is thus transmitted to the slave
- During the high period of the next 8 clock pulses the slave should transmit 11001000 (Data to be read 0xC8) over the SDA line (See fig. (d) below)
- No acknowledge from the master indicates the last byte required has been read
- Stop condition is issued
- SDA and SCL lines should return High

Fig. (a)

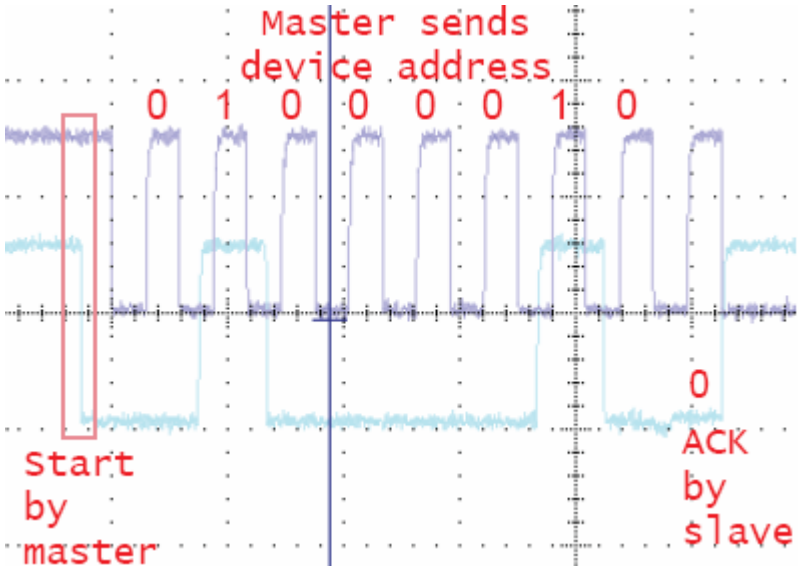


Fig. (b)

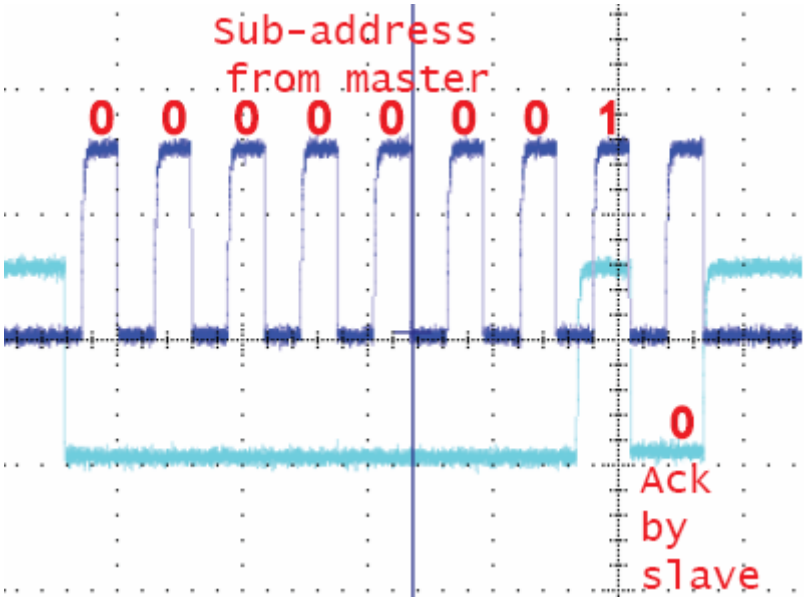

Sub-address
from master
0 0 0 0 0 0 0 1
0
Ack
by
slave

Fig. (c)

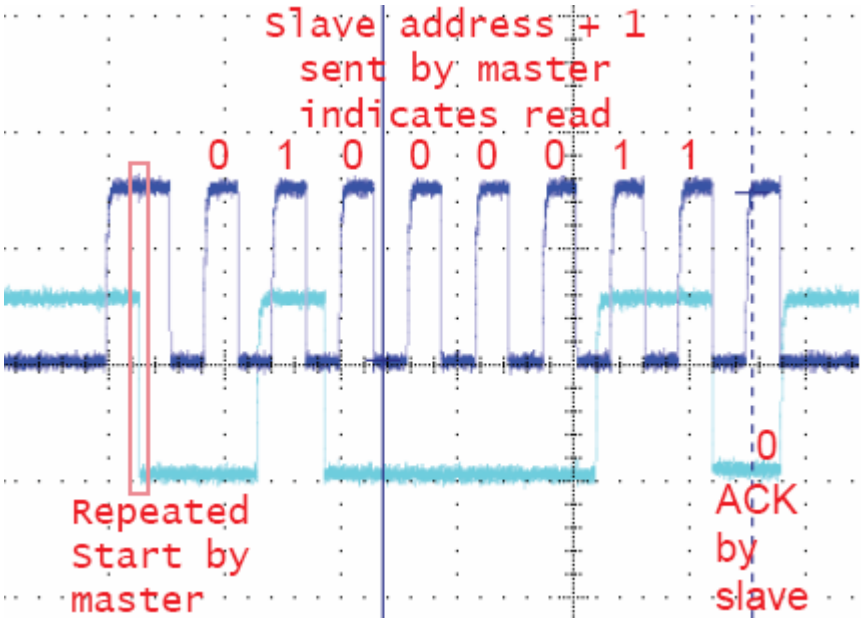
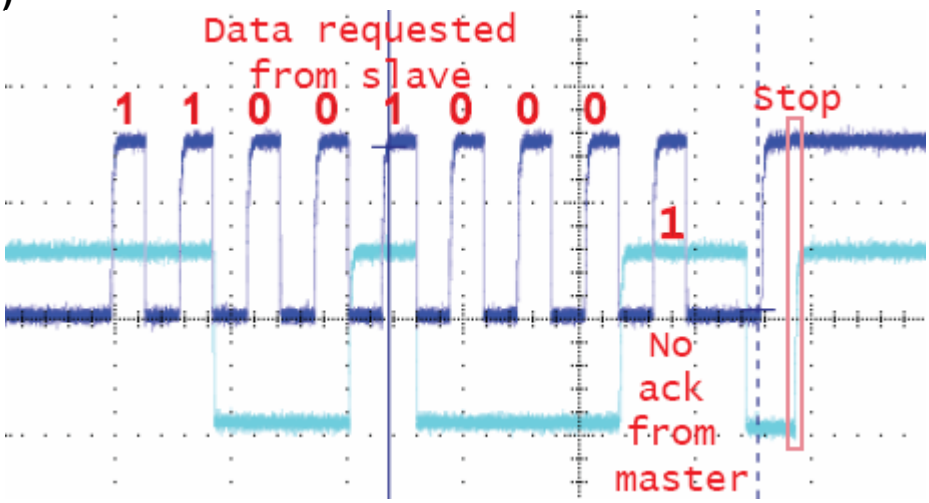

Slave address + 1
sent by master
indicates read
0 1 0 0 0 0 1 1
Repeated
Start by
master
0
ACK
by
slave

Fig. (d)


Data requested
from slave
1 1 0 0 1 0 0 0
Stop
1
No
ack
from
master