2-WIRE SERIAL CONTROL PORT

A 2-wire serial interface control interface is provided in the AD9889B. Up to two AD9889B devices can be connected to the 2-wire serial interface, with a unique address for each device.

The 2-wire serial interface comprises a clock (SCL) and a bidirectional data (SDA) pin. The analog flat panel interface acts as a slave for receiving and transmitting data over the serial interface. When the serial interface is not active, the logic levels on SCL and SDA are pulled high by external pull-up resistors.

Data received or transmitted on the SDA line must be stable for the duration of the positive-going SCL pulse. Data on SDA must change only when SCL is low. If SDA changes state while SCL is high, the serial interface interprets that action as a start or stop sequence.

There are six components to serial bus operation:

- Start signal
- Slave address byte
- Base register address byte
- Data byte to read or write
- Stop signal
- Acknowledge (Ack)

When the serial interface is inactive (SCL and SDA are high) communications are initiated by sending a start signal. The start signal is a high-to-low transition on SDA while SCL is high. This signal alerts all slaved devices that a data transfer sequence is coming.

The first eight bits of data transferred after a start signal comprise a seven bit slave address (the first seven bits) and a single R/W\ bit (the eighth bit). The R/W\ bit indicates the direction of data transfer, read from (1) or write to (0) the slave device. If the transmitted slave address matches the address of the device (set by the state of the SA0 input pin as shown in Table 97), the AD9889B acknowledges by bringing SDA low on the 9th SCL pulse. If the addresses do not match, the AD9889B does not acknowledge.

Table. Serial Port Addresses

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
A_6 (MSB)	A_5	A_4	A_3	A_2	A_1	A_0
0	1	1	1	1	0	1
0	1	1	1	1	0	0

Data Transfer via Serial Interface

For each byte of data read or written, the MSB is the first bit of the sequence.

If the AD9889B does not acknowledge the master device during a write sequence, the SDA remains high so the master can gener-ate a stop signal. If the master device does not acknowledge the AD9889B during a read sequence, the AD9889B interprets this as end of data. The SDA remains high, so the master can generate a stop signal.

Writing data to specific control registers of the AD9889B requires that the 8-bit address of the control register of interest be writ-ten after the slave address has been established. This control register address is the base address for subsequent write opera-tions. The base address auto-increments by one for each byte of data written after the data byte intended for the base address. If more bytes are transferred than there are available addresses, the address does not increment and remains at its maximum value. Any base address higher than the maximum value does not produce an acknowledge signal.

Data are read from the control registers of the AD9889B in a similar manner. Reading requires two data transfer operations:

The base address must be written with the R/W bit of the slave address byte low to set up a sequential read operation.

Reading (the R/$\overline{W}$ bit of the slave address byte high) begins at the previously established base address. The address of the read register auto-increments after each byte is transferred.

To terminate a read/write sequence to the AD9889B, a stop signal must be sent. A stop signal comprises a low-to-high transition of SDA while SCL is high.

A repeated start signal occurs when the master device driving the serial interface generates a start signal without first genera-ting a stop signal to terminate the current communication. This is used to change the mode of communication (read, write) between the slave and master without releasing the serial interface lines.

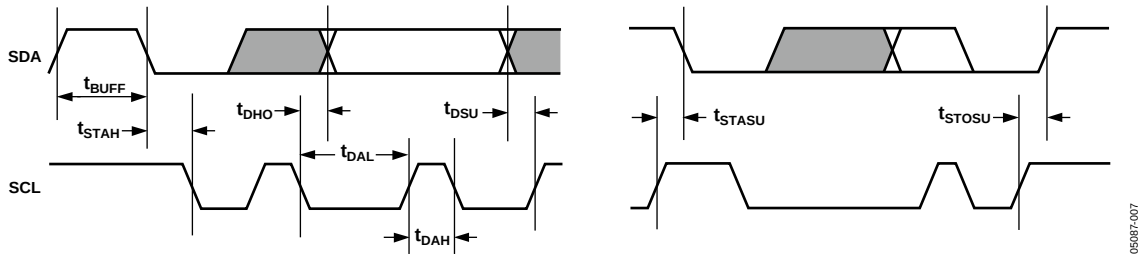

Figure xx Serial Port Read/Write Timing

Serial Interface Read/Write Examples

Write to one control register:

- Start signal
- Slave address byte (R/W\ bit = low)
- Base address byte
- Data byte to base address
- Stop signal

Write to four consecutive control registers:

- Start signal
- Slave address byte (R/W\ bit = LOW)
- Base address byte
- Data byte to base address
- Data byte to (base address + 1)
- Data byte to (base address + 2)
- Data byte to (base address + 3)
- Stop signal

Read from one control register:

- Start signal
- Slave address byte (R/W\ bit = low)
- Base address byte
- Start signal
- Slave address byte (R/W\ bit = high)
- Data byte from base address
- Stop signal

Read from four consecutive control registers:

- Start signal
- Slave address byte (R/W\ bit = low)
- Base address byte
- Start signal
- Slave address byte (R/W\ bit = high)
- Data byte from base address
- Data byte from (base address + 1)
- Data byte from (base address + 2)
- Data byte from (base address + 3)
- Stop signal

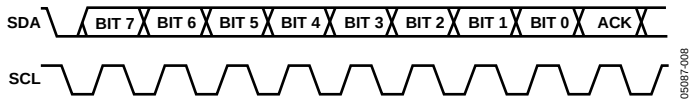

Figure 1. Serial Interface—Typical Byte Transfer