L or SDA

or SDA

d if this PIC will ever
the SCL line.
ent by the transmitter,
hen the PIC sends out
ı receiver. During the
line, pulling it low in
address.
ripheral chip, it is the
ı the PIC. In this case,
ceiver but also as bus
nal addresses back to
h byte and signals a
any number of bytes
heral to stop any fur-

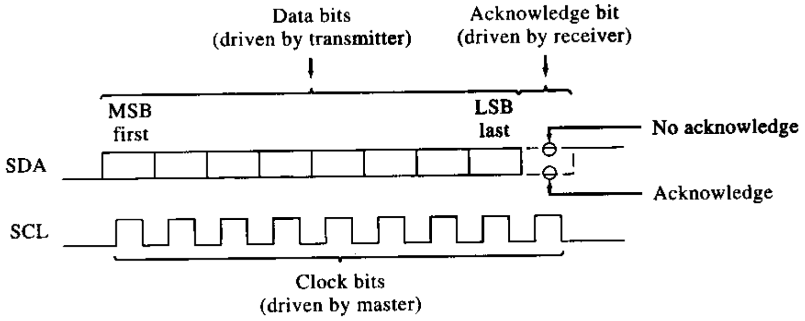

Figure 9-2 Byte transfer plus acknowledge.

Figure 9-2 also illustrates that the data bits on the SDA line must be stable during the high period of the clock. When the slave peripheral is driving the SDA line, either as transmitter or acknowledger, it initiates the new bit in response to the falling edge of SCL, after a specified hold time. It maintains that bit on the SDA line until the next falling edge of SCL, again after a specified hold time.

When the PIC master is driving the SDA line, it must meet the same hold-time specification when it changes SDA after driving SCL low. In addition, it must meet several other timing specifications as it changes SCL and SDA. These will be identified shortly.

I^{2}C bus transfers consist of a number of byte transfers *framed* between a START condition and either another START condition or a STOP condition. When bus transfers are *not* taking place, both the SDA and the SCL lines are released by all drivers and float high. The PIC (I^{2}C bus controller) initiates a transfer with the START condition. It first pulls SDA low and then it pulls SCL low, as shown in Figure 9-3a. Likewise, the PIC terminates a multiple-byte transfer with the STOP condition. With both SDA and SCL initially low, it first releases SCL and then SDA, as shown in Figure 9-3b. Both of these occurrences are easily recognized by the I^{2}C hardware in each peripheral chip since they both consist of a change in the SDA line while SCL is high, a condition that never happens in the middle of a byte transfer.

The PIC I^{2}C bus master generates the first byte after the START condition. It consists of a 7-bit *slave address* followed by an R/W bit, as shown in Figure 9-4. If the R/W bit is low, subsequent bytes transmitted on the bus will be *written* by the PIC to the selected peripheral. If the R/W bit is high, subsequent bytes will be sent by the selected peripheral and *read* by the PIC.

The I^{2}C bus standard was augmented in 1995 with the definition of 10-bit addresses that begin with what looks like a nonstandard 7-bit address, B'11110xx'. The last two bits of this 7-bit address plus a second 8-bit address byte form the 10-bit address in the augmented standard. Since the three peripheral chips to be discussed later in this chapter all use 7-bit addresses (as do most commodity chips having an I^{2}C interface), the several ramifications that attend the use of 10-bit addressing will not be discussed here.

The functions of the bytes that follow the first, or control, byte are defined by the needs of the peripheral chip. For a peripheral chip that contains more than one internal register or memory address, the PIC will typically write a second byte to the chip to set a pointer to the selected internal register or address. Subsequent bytes in the message string will typically be written to that address and then to the consecutive addresses that follow it. This is illustrated in Figure 9-5a.

A message string for reading internal peripheral registers or addresses is shown in Figure 9-5b. It begins with a 2-byte message string that selects the internal address of the selected peripheral chip