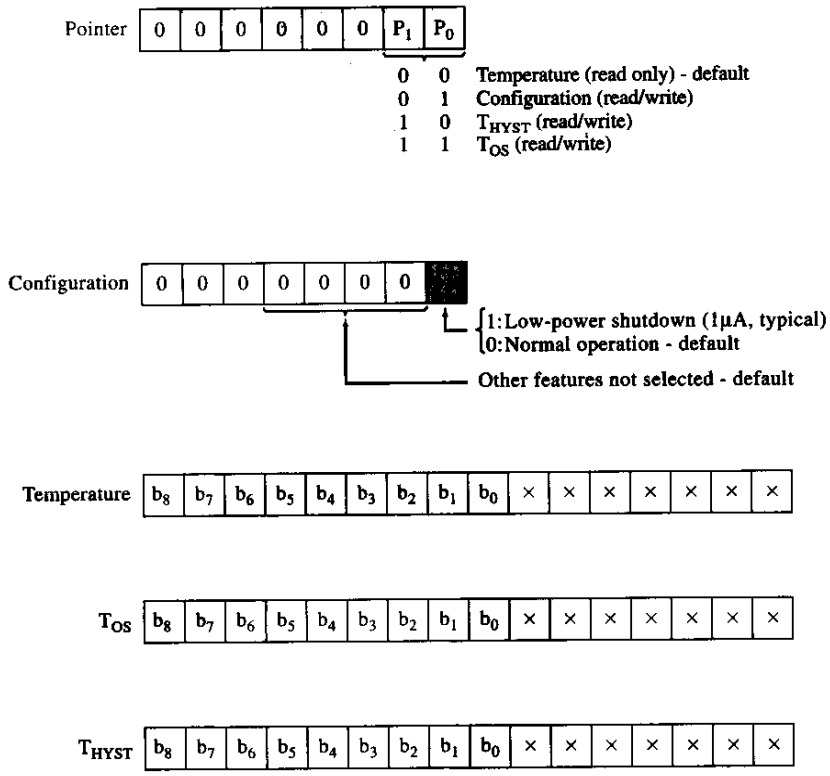

Figure 9-13 LM75 internal registers.

The EEPROM makes use of an internal address pointer that is set during the second byte of a "write" message string, as shown in Figures 9-16a and 9-16b. If further bytes are transmitted before the STOP condition, as in Figure 9-16b, they will be accepted as the data to be written into the selected internal addresses. The reception of the STOP condition triggers the programming of these bytes into the selected addresses.

While the EEPROM is doing its autonomous programming operation, it will not acknowledge another write command. Because of this, the acknowledge bit can be used as a flag to determine when the programming operation has been completed. Simply send out the slave address with the write bit low and check whether the ACK bit is pulled low by the EEPROM. Until it does get pulled low in acknowledgment, the START condition followed by the same byte can be sent repeatedly and the ACK bit tested. With a typical programming time of 2 ms, programming of many bytes can take place as rapidly as possible, faster than simply allowing the 10-ms worst-case write time to expire.

This EEPROM includes a *page-write buffer* for writing up to 8 bytes simultaneously with the single write message string shown in Figure 9-16b. Within 10 ms after the STOP condition is received by the EEPROM, all of the transmitted bytes will be programmed. However, all eight addresses are constrained to have the same upper 5 bits. That is, only the lower 3 bits of the EEPROM's internal address counter are incremented when more than one data byte is included in a write command sequence. For example, if the EEPROM address sent in the second byte of the write command is B'00010110' and

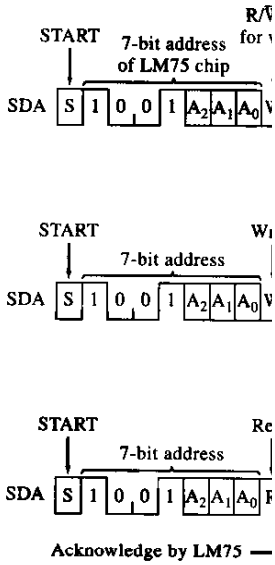

Figure 9-14 LM75 messa

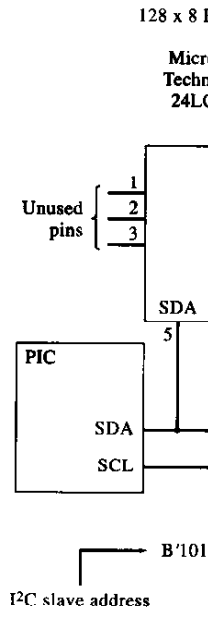

Figure 9-15 I²C bu