

# 1 I2C Bus Interface

## 1.1 Overview

The I2C bus was created by the Phillips Corporation and is a serial bus with a two-pin interface. The SDA data pin is used for input and output functions and the SCL clock pin is used to control and reference the I2C bus. The I2C unit allows the processor to serve as a master and slave device that resides on the I2C bus. The I2C unit enables the processor to communicate with I2C peripherals and microcontrollers for system management functions. The I2C bus requires a minimum amount of hardware to relay status and reliability information concerning the processor subsystem to an external device. The I2C unit is a peripheral device that resides on the processor internal bus. Data is transmitted to and received from the I2C bus via a buffered interface. Control and status information is relayed through a set of memory-mapped registers. Refer to ***The I2C-Bus Specification*** for complete details on I2C bus operation.

The I2C has the following features:

- Supports only single master mode.
- Supports I2C standard-mode and F/S-mode up to 400 kHz.
- I2C receiver and transmitter are double-buffered.
- Supports burst reading or writing of data.
- Supports random writing access of data.
- Supports general call address and START byte format after START condition.
- Independent, programmable serial clock generator.
- Supports slave coping with fast master during data transfers by holding the SCL line on a bit level.
- The number of devices that you can connect to the same I2C-bus is limited only by the maximum bus capacitance of 400pF.

## 1.2 Pin Description

Table 1-1 Smart Card Controller Pins Description

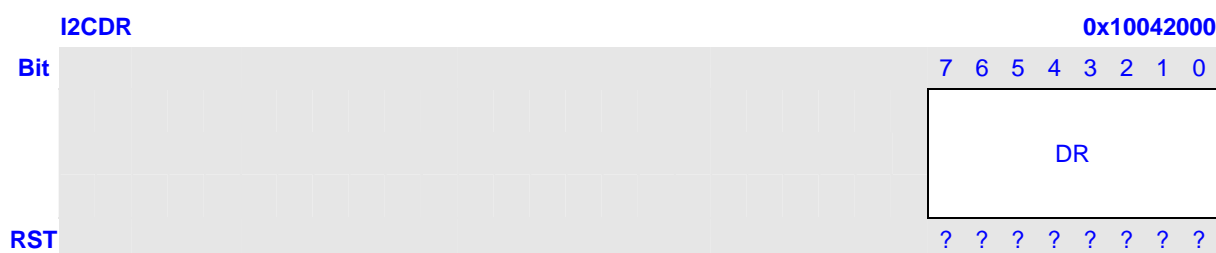
Name	I/O	Description
SDA	Input/Output	I2C Serial Clock Line signal.
SCL	Input/Output	I2C Serial Data/Address signal.

## 1.3 Register Description

Table 1-2 I2C Registers Description

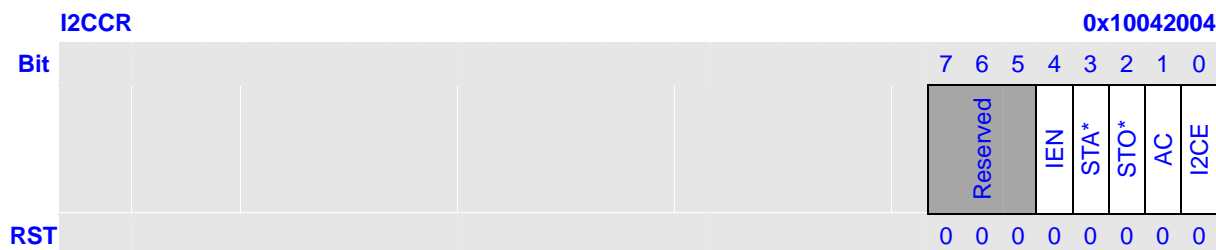
Name	RW	Reset Value	Address	Access Size
I2CDR	RW	0x??	0x10042000	8
I2CCR	RW	0x00	0x10042004	8
I2CSR	RW	0x04	0x10042008	8
I2CGR	RW	0x0000	0x1004200C	16

### 1.3.1 Data Register (I2CDR)



Bits	Name	Description	RW
7:0	DR	Data port of HW FIFO.	RW

### 1.3.2 Control Register (I2CCR)

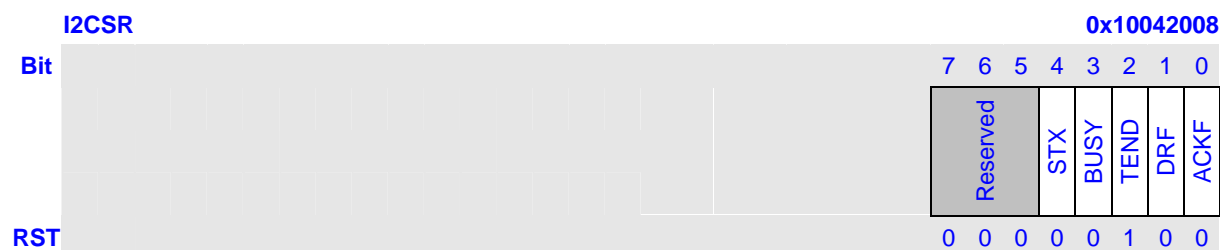


**\*Note:** STA and STO can only be written with 1.

Bits	Name	Description	RW
7:5	Reserved	These bits always read as 0. Write data to these bits are ignored.	R
4	IEN	I2C interrupt bit. 0 – Disable I2C interrupt. 1 – Enable I2C interrupt.	RW
3	STA	I2C START bit. 0 – START condition will not be sent to I <sup>2</sup> C bus. 1 – START condition will be sent to I <sup>2</sup> C bus.	RW
2	STO	I2C STOP bit. 0 – STOP condition won't be sent to I <sup>2</sup> C bus. 1 – STOP condition will be sent to I <sup>2</sup> C bus.	RW
1	AC	I2C Acknowledge Control Bit. 0 – will be sent to I <sup>2</sup> C bus as LOW level acknowledge signal. 1 – will be sent to I <sup>2</sup> C bus as HIGH level	RW

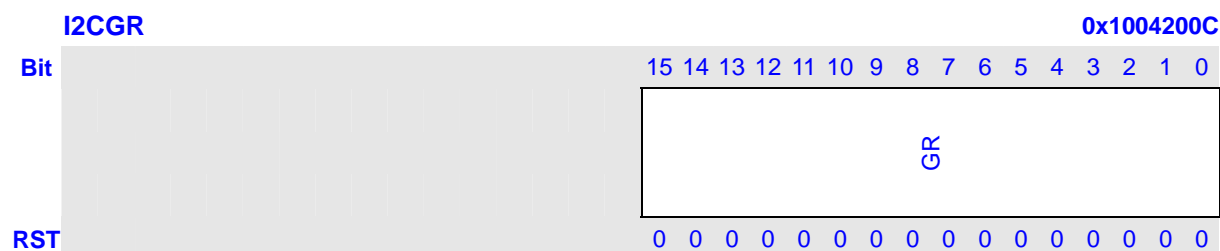
		acknowledge signal.	
0	I2CE	Enable of I2C. 0 – I2C module is disabled. 1 – I2C module is enabled.	RW

### 1.3.3 Status Register (I2CSR)



Bits	Name	Description	RW
7:5	Reserved	These bits always read as 0. Write data to these bits are ignored.	R
4	STX	STA/STO Command is On. 0 – STA/STO FIFO buffer is empty. 1 – STA/STO FIFO buffer is not empty.	R
3	BUSY	I2C Bus Busy. 0 – I2C bus is free. 1 – I2C bus is busy.	R
2	TEND	Transmission End Flag. 0 – Byte transmission or acknowledge bit for that byte has not completed. 1 – The I2C is in transmission idle state.	R
1	DRF	Data Register Valid Flag. 0 – Data in I2CDR is invalid. 1 – Data in I2CDR is valid.	RW
0	ACKF	Acknowledge Level Flag. 0 – The acknowledge signal from I <sup>2</sup> C-bus is “0”. 1 – The acknowledge signal from I <sup>2</sup> C-bus is “1”.	R

### 1.3.4 Clock Generator Register (I2CGR)



Bits	Name	Description	RW
15:01	GR	Sets the frequency of serial clock. The serial clocks frequency is calculated as follows: $[\text{Value of I2CGR}] = [\text{Frequency of Device\_clock}] / (16 * [\text{SCL clock rate}]) - 1$	RW

**Note:** To make the I2C operate normally, frequency of PCLK (APB-bus clock) should not lower than transfer 2 \* [byte rate].

## 1.4 I<sup>2</sup>C-Bus Protocol

### 1.4.1 Bit Transfer

Due to the variety of different technology devices (CMOS, NMOS, bipolar) which can be connected to the I<sup>2</sup>C-bus, the levels of the logical '0' (LOW) and '1' (HIGH) are not fixed and depend on the associated level of VDD. One clock pulse is generated for each data bit transferred.

### 1.4.2 Data Validity

The data on the SDA line must be stable during the HIGH period of the clock. The HIGH or LOW states of the data line can only change when the clock signal on the SCL line is LOW.

### 1.4.3 START and STOP Conditions

A HIGH to LOW transition on the SDA line while SCL is HIGH indicates a START condition. A LOW to HIGH transition on the SDA line while SCL is HIGH defines a STOP condition.

### 1.4.4 Byte Format

1. Every byte put on the SDA line must be 8-bits width
2. The number of bytes that can be transmitted/received per transfer is unrestricted.
3. Each byte has to be followed by an acknowledge (ack/nack) bit.
4. Data is transferred with the most significant bit (MSB) first.
5. Data transfer with an acknowledge signal (acknowledge or not-acknowledge) is obligatory.
6. The acknowledge\_ related clock pulse is generated by the master.
7. The transmitter releases the SDA line (HIGH) during the acknowledge clock pulse. The receiver must pull down the SDA line during the acknowledge clock pulse so that it remains stable LOW during the HIGH period of this clock pulse.
8. Slave can hold the SCL line LOW during the SCL in LOW level at any bit to force the master to proceed a lower speed of transfer.

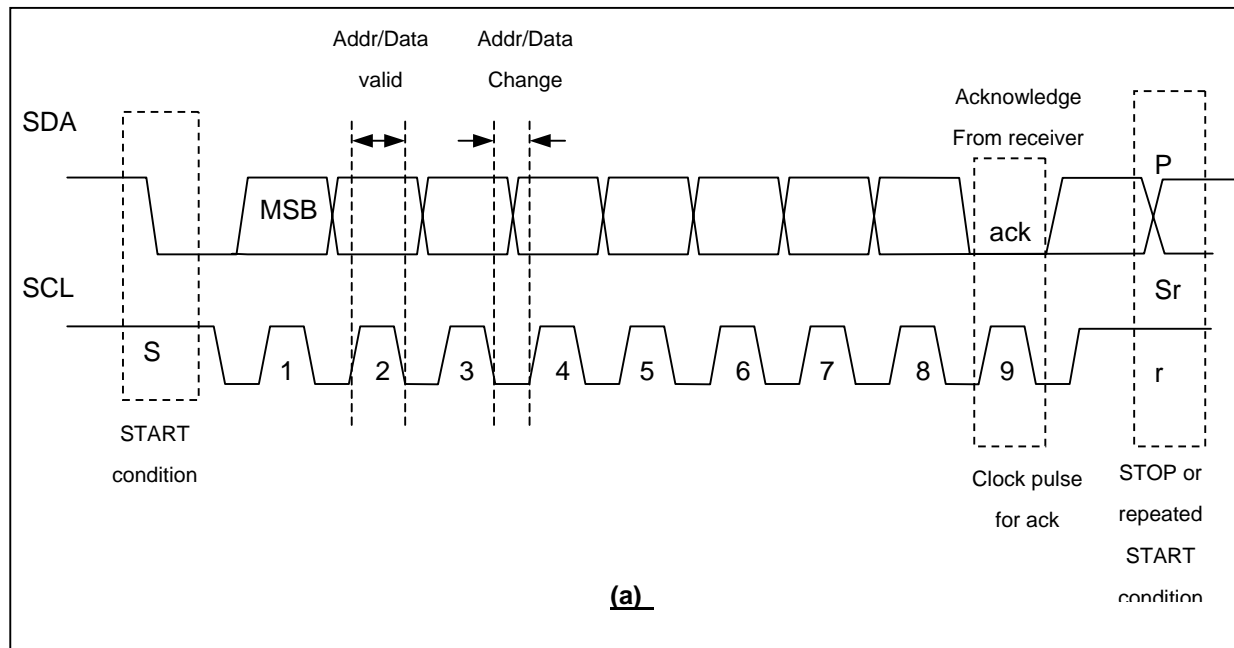
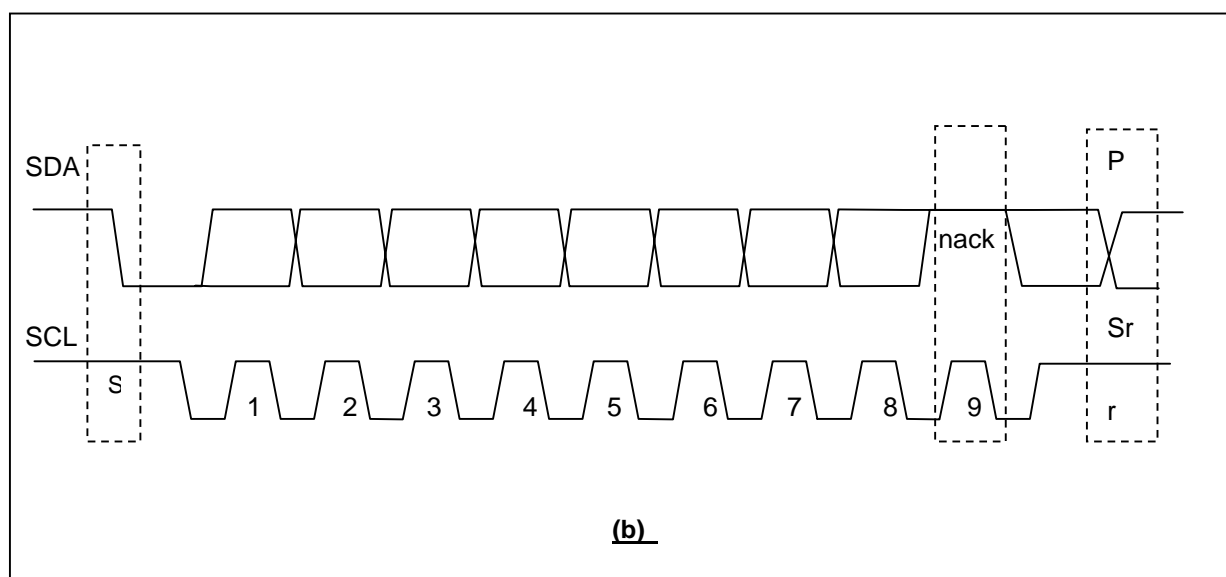


Figure 1-1 I2C-bus Protocol

Figure 1-2 I<sup>2</sup>C-bus Protocol (cont.)**Notes:**

1. Sr means repeated START condition. P means STOP condition.
2. In Fig (a), if the master does not generate Sr or P, the next data byte follows the ack.
3. In Fig (b), nack is received, the master generates Sr or P and the transfer terminates.

## 1.4.5 Data Transfer Format

### 1.4.5.1 First Byte

The first byte is a term indicates the address byte after START condition.

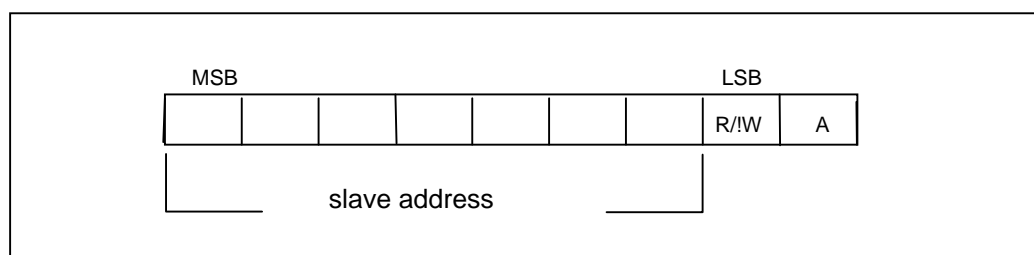
#### 1) Normal 7-bit Address:

After the START condition, the addressing procedure for the I<sup>2</sup>C-bus is such that the first byte usually determines which slave will be selected by the master.

The first seven bits of the first byte make up the slave address. The eighth bit is the LSB (least significant bit). It determines the direction of the message. A 'zero' in the least significant position of the first byte means that the master will write information to a selected slave. A 'one' in this position means that the master will read information from the slave.

When an address is sent, each device in a system compares the first seven bits after the START condition with its address. If they match, the device considers itself addressed by the master as a slave-receiver or slave-transmitter, depending on the R/W bit.

A slave address can be made-up of a fixed and a programmable part. Since it's likely that there will be several identical devices in a system, the programmable part of the slave address enables the maximum possible number of such devices to be connected to the I<sup>2</sup>C-bus. The number of programmable address bits of a device depends on the number of pins available. For example, if a device has 4 fixed and 3 programmable address bits, a total of 8 identical devices can be connected to the same bus.



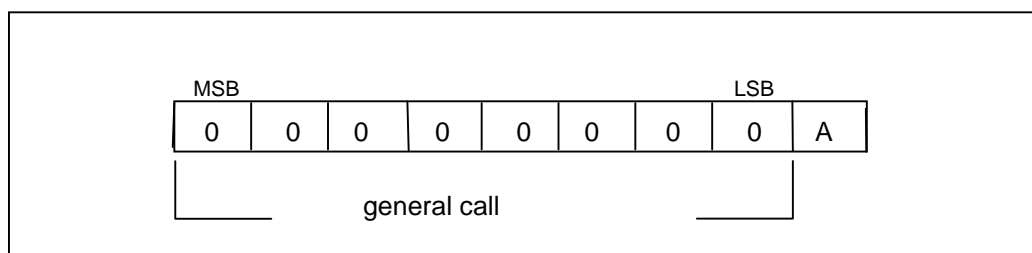
**Figure 1-3 Normal 7 Bit Address after START Condition**

#### 2) General Call Address:

Address byte with all bits are "0" is defined as "general call address". When this address is used, all devices should, in theory, respond with an acknowledge. However, if a device doesn't need any of the data supplied within the general call structure, it can ignore this address by not issuing an acknowledgment. If a device does require data from a general call address, it will acknowledge this address and behave as a slave- receiver. The second and following bytes will be acknowledged by every slave-receiver capable of handling this data. A slave that cannot process one of these bytes

must ignore it by not-acknowledging.

The second byte of the general call address then defines the action to be taken.



**Figure 1-4 General Call Address after START Condition**

### 3) START Byte Address:

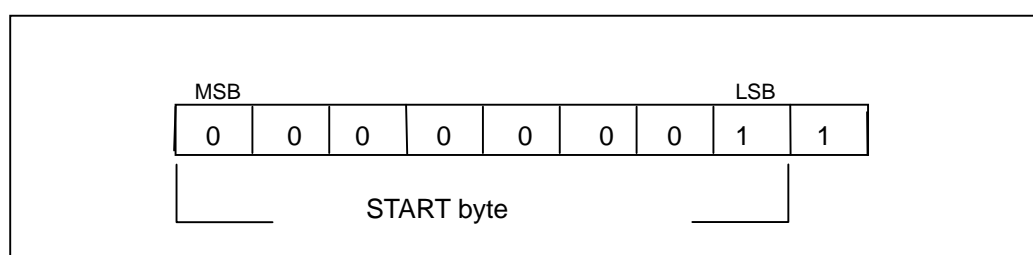
START Byte:

After the START condition S has been transmitted by the master, data transfer can be preceded by a start procedure which is much longer than normal. The start procedure consists of:

- A START condition (S)
- A START byte (00000001)
- An acknowledge clock pulse (ACK)\*
- A repeated START condition (Sr)

**Note:** An acknowledge-related clock pulse is generated after the START byte. This is present only to conform to the byte handling format used on the bus. No device is allowed to acknowledge the START byte.

When the START byte (00000001) is transmitted, another microcontroller (the slave) can therefore sample the SDA line at a low sampling rate (also determined by the I2CGR) until one of the seven zeros in the START byte is detected. After detection of this LOW level on the SDA line, the microcontroller can switch to a higher sampling rate to find the repeated START condition Sr which is then used for synchronization.



**Figure 1-5 START Byte after START Condition**



### 1.4.5.2 Transfer Format

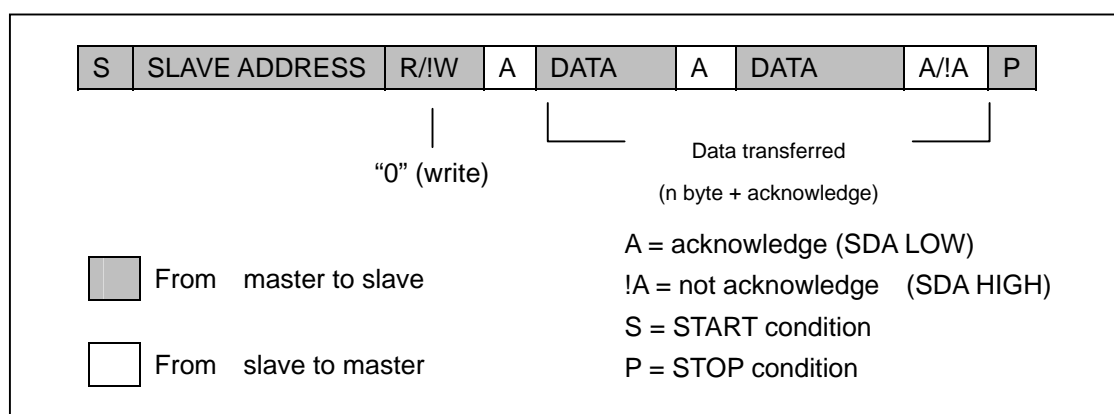
A data transfer is always terminated by a STOP condition (P) generated by the master. However, if a master still wishes to communicate on the bus, it can generate a repeated START condition (Sr) and address another slave without first generating a STOP condition. Various combinations of read/write formats are then possible within such a transfer.

**Possible data transfer formats are:**

- Master-transmitter transmits to slave-receiver. The transfer direction is not changed.
- Master reads slave immediately after first byte. At the moment of the first acknowledge, the master-transmitter becomes a master-receiver and the slave-receiver becomes a slave-transmitter.
- This first acknowledge is still generated by the slave. The STOP condition is generated by the master, which has previously sent a not-acknowledge.

**Notes:**

1. Combined formats can be used, for example, to control a serial memory. During the first data byte, the internal memory location has to be written. After the START condition and slave address is repeated, data can be transferred.
2. All decisions on auto-increment or decrement of previously accessed memory locations etc. are taken by the designer of the device.
3. Each byte is followed by an acknowledgment bit as indicated by the 'A' or '!A' blocks in the sequence.



**Figure 1-6 A Master-Transmitter Addresses a Slave Receiver with a 7-Bit Address**

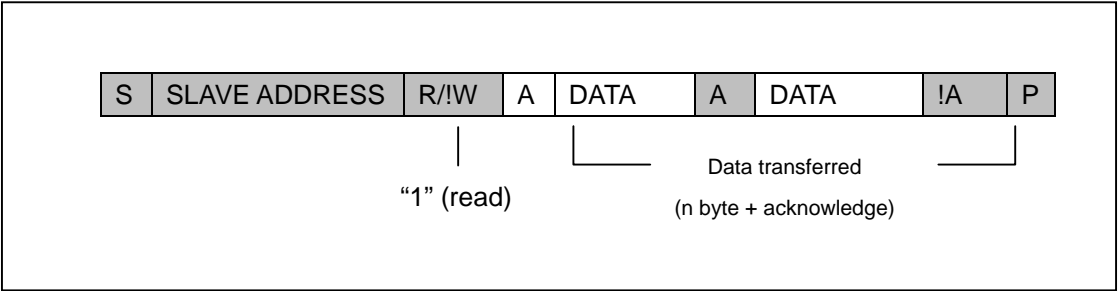
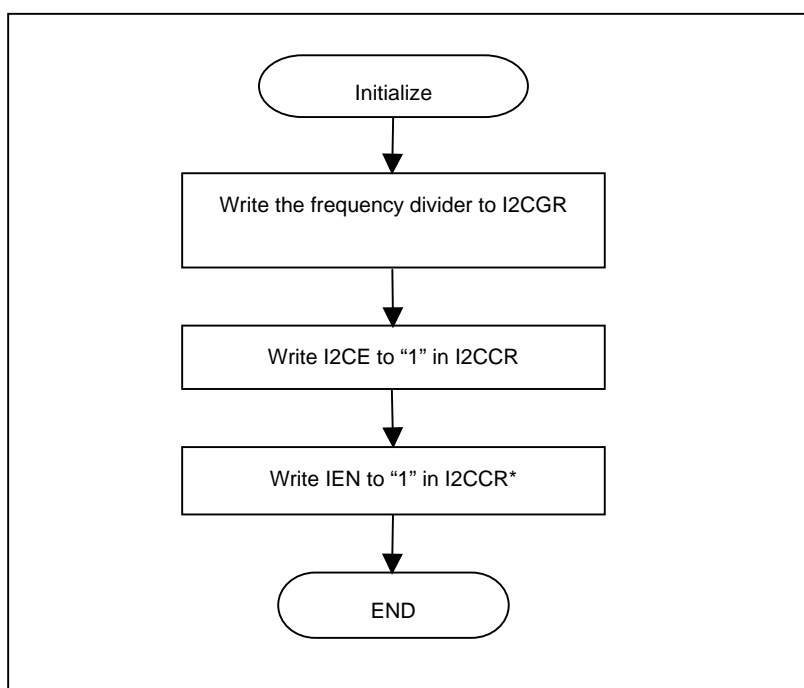


Figure 1-7 A Master Reads the Slave Immediately after the First Byte (Master-Receiver)

## 1.5 I2C Operation

### 1.5.1 I2C Initialization

Before transmitting and receiving data, set the I2CE bit in I2CCR to 1 to enable I2C operation and set I2CGR for proper serial clock frequency. Set the I2CE bit to 0 after transmitting or receiving data for low power dissipation.



**Figure 1-8 I2C Initialization**

**Note:** This step is selectable.

### 1.5.2 Write Operation

Following figure illustrates the flow of a write operation.

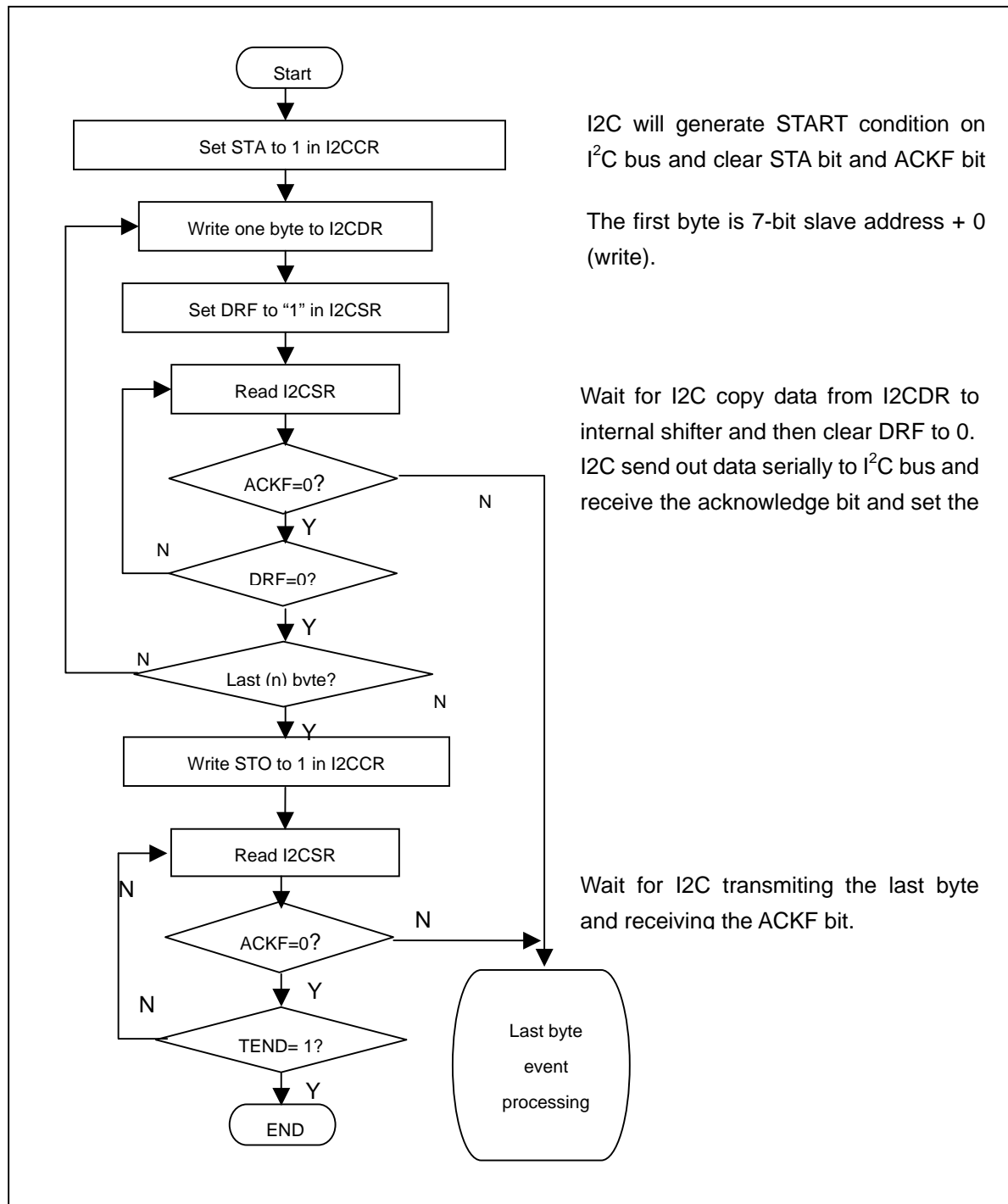


Figure 1-9 I2C Write Operation Flowchart

### 1.5.3 Read Operation

Following figure illustrates the flow of read operation.

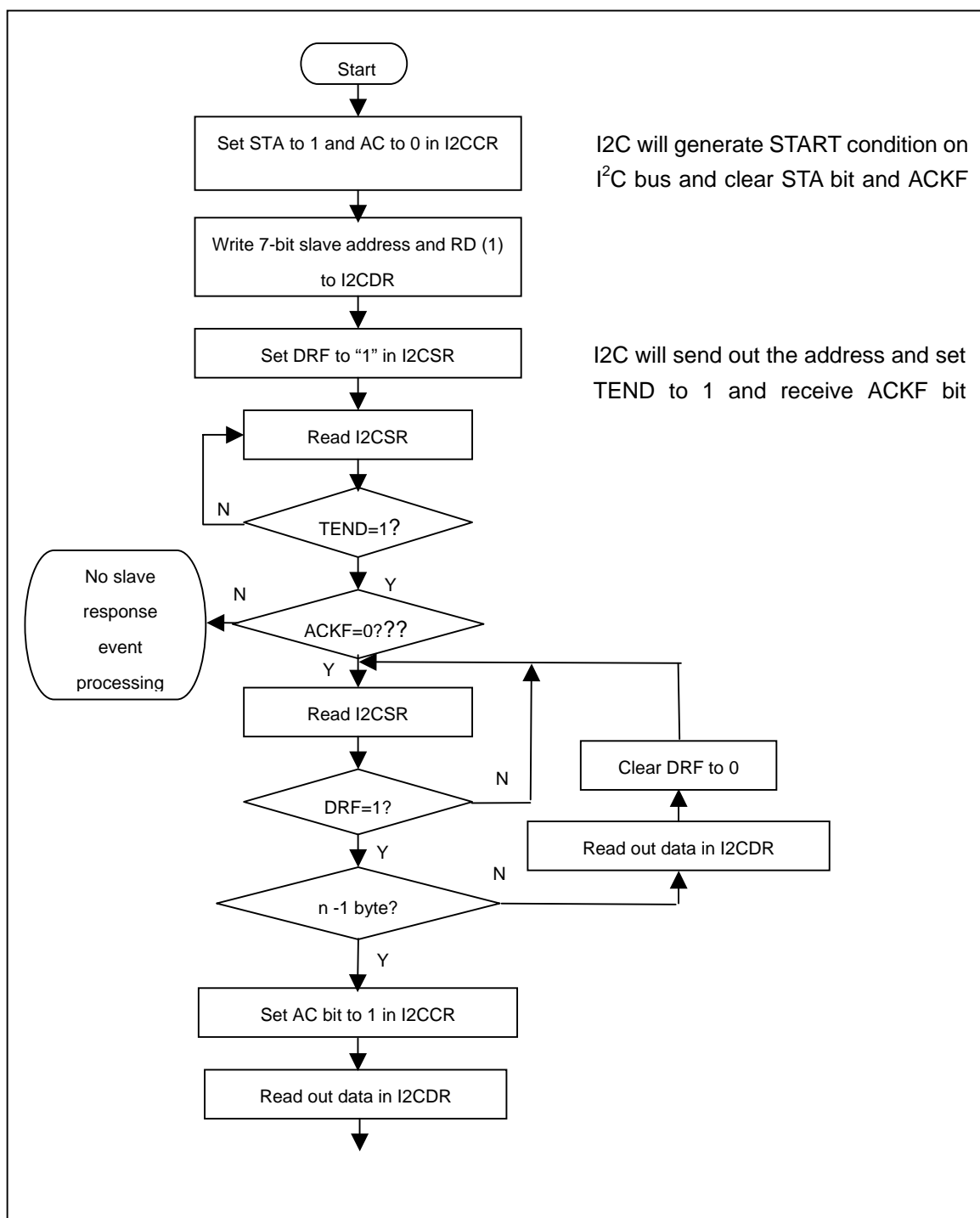


Figure 1-10 I2C Read Operation Flowchart

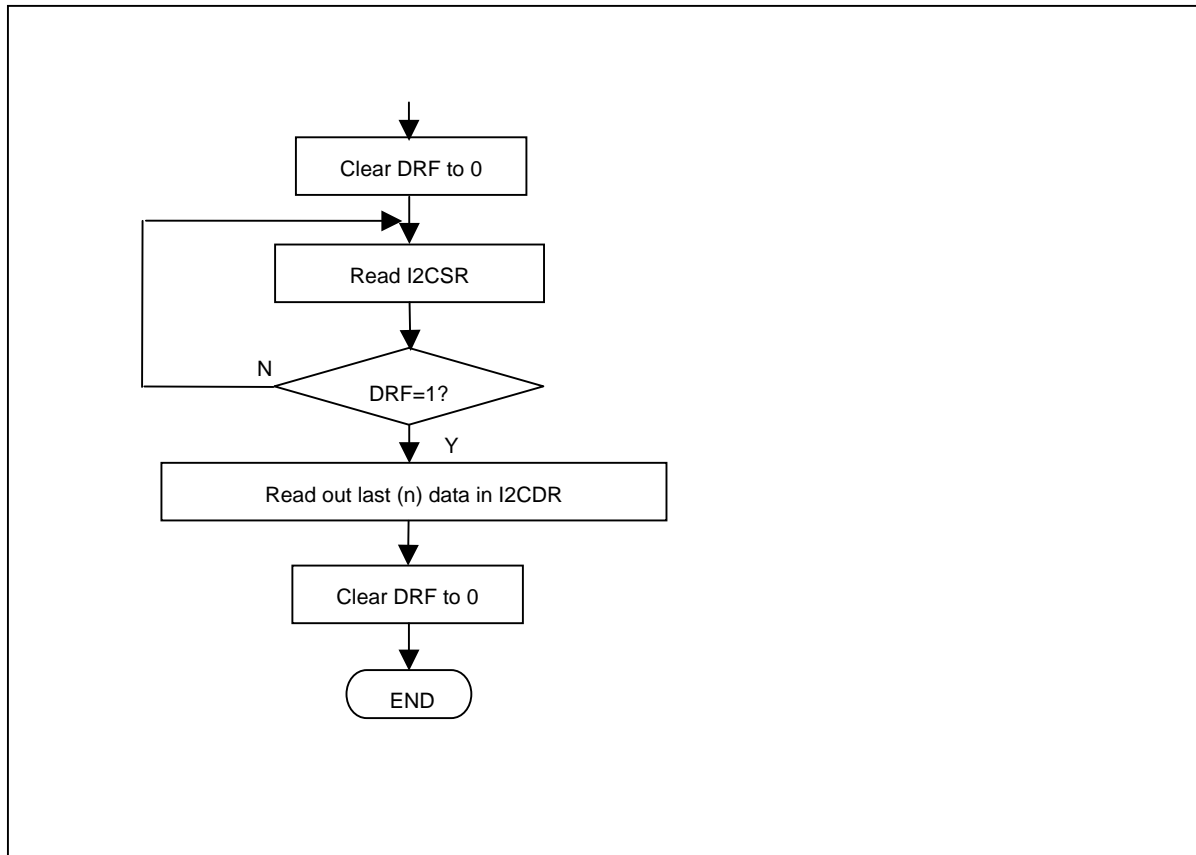


Figure 1-11 Read Operation Flowchart (cont.)