Chapter 1
SLAU412E–August 2012–Revised October 2016

Universal Serial Communication Interface – I²C Mode

NOTE: This chapter is an excerpt from the *MSP430x5xx and MSP430x6xx Family User's Guide*. The most recent version of the full user's guide is available here: http://www.ti.com/lit/pdf/slau208.

The universal serial communication interface (USCI) supports multiple serial communication modes with one hardware module. This chapter discusses the operation of the I²C mode.

Topic **Page**

1.1 Universal Serial Communication Interface (USCI) Overview

The USCI modules support multiple serial communication modes. Different USCI modules support different modes. Each different USCI module is named with a different letter. For example, USCI_A is different from USCI_B, etc. If more than one identical USCI module is implemented on one device, those modules are named with incrementing numbers. For example, if one device has two USCI_A modules, they are named USCI_A0 and USCI_A1. See the device-specific data sheet to determine which USCI modules, if any, are implemented on each device.

USCI_Ax modules support:

- UART mode
- Pulse shaping for IrDA communications
- Automatic baud-rate detection for LIN communications
- SPI mode

USCI_Bx modules support:

- I^2C mode
- SPI mode

1.2 USCI Introduction – I²C Mode

In I²C mode, the USCI module provides an interface between the device and I²C-compatible devices connected by the two-wire I²C serial bus. External components attached to the I²C bus serially transmit and/or receive serial data to/from the USCI module through the 2-wire I²C interface.

The I²C mode features include:

- Compliance to the Philips Semiconductor I²C specification v2.1
- 7-bit and 10-bit device addressing modes
- General call
- START/RESTART/STOP
- Multi-master transmitter/receiver mode
- Slave receiver/transmitter mode
- Standard mode up to 100 kbps and fast mode up to 400 kbps support
- Programmable UCxCLK frequency in master mode
- Designed for low power
- Slave receiver START detection for auto wake up from LPMx modes (wake up from LPMx.5 is not supported)
- Slave operation in LPM4

Figure 1-1 shows the USCI when configured in I²C mode.

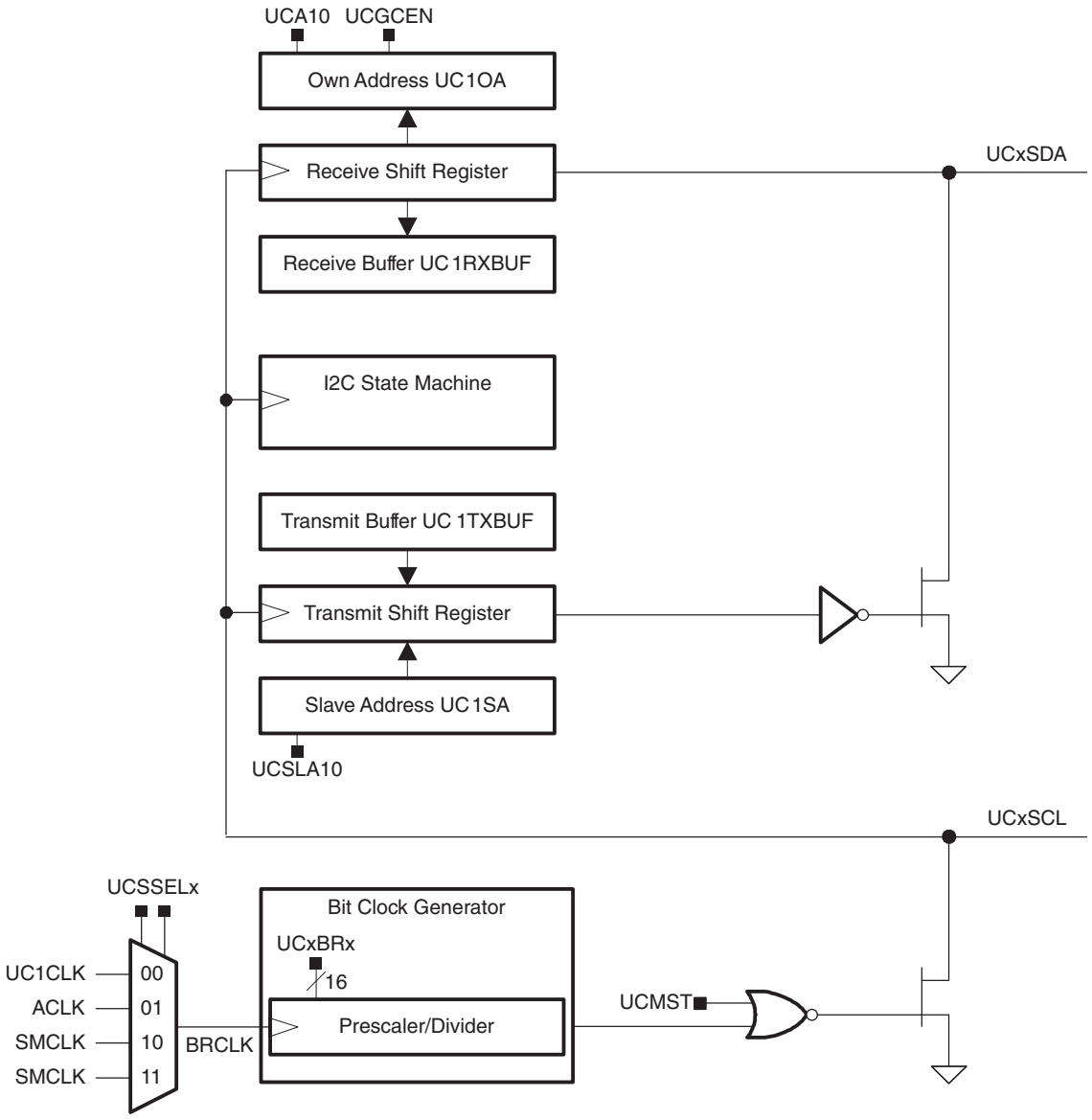

Figure 1-1. USCI Block Diagram – I²C Mode

1.3 USCI Operation – I²C Mode

The I²C mode supports any slave or master I²C-compatible device. Figure 1-2 shows an example of an I²C bus. Each I²C device is recognized by a unique address and can operate as either a transmitter or a receiver. A device connected to the I²C bus can be considered as the master or the slave when performing data transfers. A master initiates a data transfer and generates the clock signal SCL. Any device addressed by a master is considered a slave.

I²C data is communicated using the serial data (SDA) pin and the serial clock (SCL) pin. Both SDA and SCL are bidirectional and must be connected to a positive supply voltage using a pullup resistor.

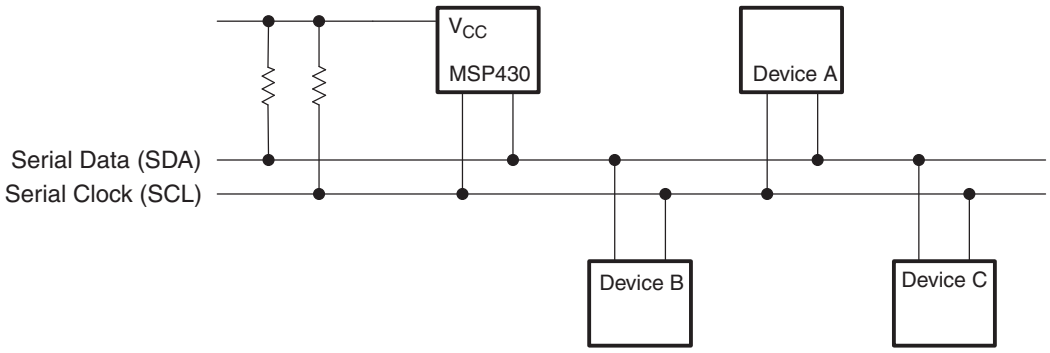

Figure 1-2. I²C Bus Connection Diagram

NOTE: SDA and SCL levels

The SDA and SCL pins must not be pulled up above the device V_{CC} level.

1.3.1 USCI Initialization and Reset

The USCI is reset by a PUC or by setting the UCSWRST bit. After a PUC, the UCSWRST bit is automatically set, keeping the USCI in a reset condition. To select I²C operation, the UCMODEx bits must be set to 11b. After module initialization, it is ready for transmit or receive operation. Clear UCSWRST to release the USCI for operation.

To avoid unpredictable behavior, configure or reconfigure the USCI module only when UCSWRST is set.

Setting UCSWRST in I²C mode has the following effects:

- I²C communication stops.
- SDA and SCL are high impedance.
- UCBxI2CSTAT, bits 6–0 are cleared.
- Registers UCBxIE and UCBxIFG are cleared.
- All other bits and register remain unchanged.

NOTE: Initializing or reconfiguring the USCI module

The recommended USCI initialization/reconfiguration process is:

1. Set UCSWRST (`BIS.B #UCSWRST,&UCxCTL1`).
2. Initialize all USCI registers with UCSWRST = 1.
3. Configure ports.
4. Clear UCSWRST through software (`BIC.B #UCSWRST,&UCxCTL1`).
5. Enable interrupts (optional).

1.3.2 I²C Serial Data

One clock pulse is generated by the master device for each data bit transferred. The I²C mode operates with byte data. Data is transferred MSB first as shown in Figure 1-3.

The first byte after a START condition consists of a 7-bit slave address and the R/$\overline{W}$ bit. When R/$\overline{W}$ = 0, the master transmits data to a slave. When R/$\overline{W}$ = 1, the master receives data from a slave. The ACK bit is sent from the receiver after each byte on the ninth SCL clock.

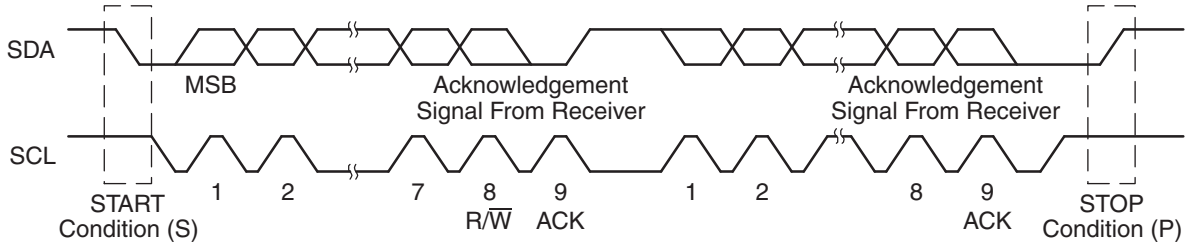

Figure 1-3. I²C Module Data Transfer

START and STOP conditions are generated by the master and are shown in Figure 1-3. A START condition is a high-to-low transition on the SDA line while SCL is high. A STOP condition is a low-to-high transition on the SDA line while SCL is high. The bus busy bit, UCBBUSY, is set after a START and cleared after a STOP.

Data on SDA must be stable during the high period of SCL (see Figure 1-4). The high and low state of SDA can only change when SCL is low, otherwise START or STOP conditions are generated.

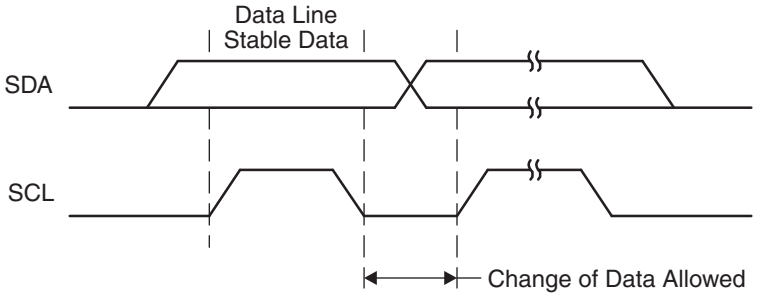

Figure 1-4. Bit Transfer on I²C Bus

1.3.3 I²C Addressing Modes

The I²C mode supports 7-bit and 10-bit addressing modes.

1.3.3.1 7-Bit Addressing

In the 7-bit addressing format (see Figure 1-5), the first byte is the 7-bit slave address and the R/$\overline{W}$ bit. The ACK bit is sent from the receiver after each byte.

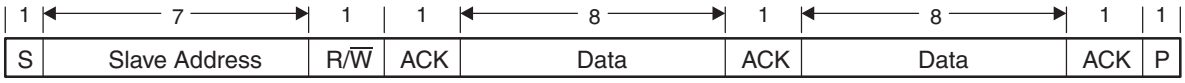

Figure 1-5. I²C Module 7-Bit Addressing Format

1.3.3.2 10-Bit Addressing

In the 10-bit addressing format (see Figure 1-6), the first byte is made up of 11110b plus the two MSBs of the 10-bit slave address and the R/$\overline{W}$ bit. The ACK bit is sent from the receiver after each byte. The next byte is the remaining eight bits of the 10-bit slave address, followed by the ACK bit and the 8-bit data. See I2C Slave 10-bit Addressing Mode and I2C Master 10-bit Addressing Mode for details how to use the 10-bit addressing mode with the USCI module.

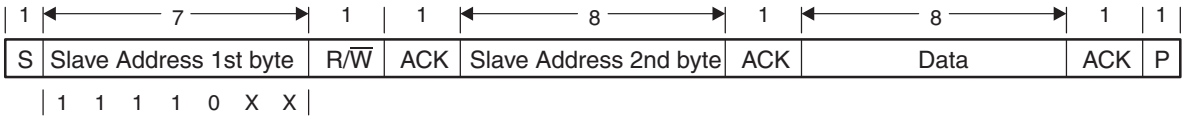

Figure 1-6. I²C Module 10-Bit Addressing Format

1.3.3.3 Repeated Start Conditions

The direction of data flow on SDA can be changed by the master, without first stopping a transfer, by issuing a repeated START condition. This is called a RESTART. After a RESTART is issued, the slave address is again sent out with the new data direction specified by the R/$\overline{W}$ bit. The RESTART condition is shown in Figure 1-7.

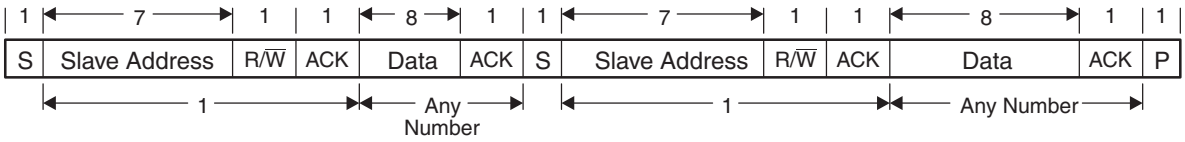

Figure 1-7. I²C Module Addressing Format With Repeated START Condition

1.3.4 I²C Module Operating Modes

In I²C mode, the USCI module can operate in master transmitter, master receiver, slave transmitter, or slave receiver mode. The modes are discussed in the following sections. Time lines are used to illustrate the modes.

Figure 1-8 shows how to interpret the time-line figures. Data transmitted by the master is represented by grey rectangles; data transmitted by the slave is represented by white rectangles. Data transmitted by the USCI module, either as master or slave, is shown by rectangles that are taller than the others.

Actions taken by the USCI module are shown in grey rectangles with an arrow indicating where in the data stream the action occurs. Actions that must be handled with software are indicated with white rectangles with an arrow pointing to where in the data stream the action must take place.

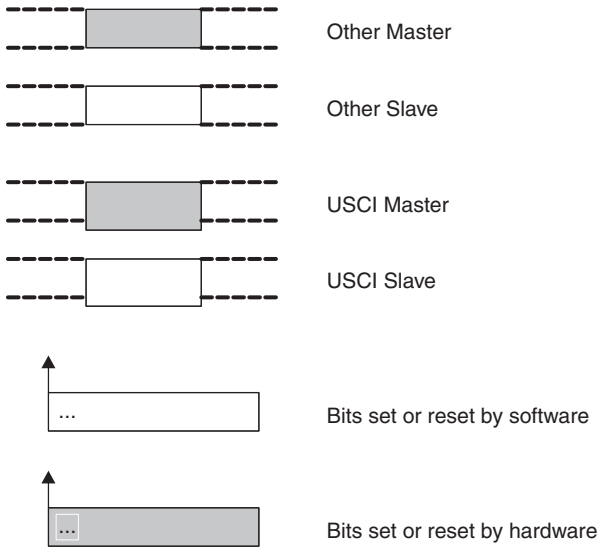

Figure 1-8. I²C Time-Line Legend

1.3.4.1 Slave Mode

The USCI module is configured as an I²C slave by selecting the I²C mode with UCMODEx = 11 and UCSYNC = 1 and clearing the UCMST bit.

Initially, the USCI module must to be configured in receiver mode by clearing the UCTR bit to receive the I²C address. Afterwards, transmit and receive operations are controlled automatically, depending on the R/$\overline{W}$ bit received together with the slave address.

The USCI slave address is programmed with the UCBxI2COA register. When UCA10 = 0, 7-bit addressing is selected. When UCA10 = 1, 10-bit addressing is selected. The UCGCEN bit selects if the slave responds to a general call.

When a START condition is detected on the bus, the USCI module receives the transmitted address and compare it against its own address stored in UCBxI2COA. The UCSTTIFG flag is set when address received matches the USCI slave address.

1.3.4.1.1 I²C Slave Transmitter Mode

Slave transmitter mode is entered when the slave address transmitted by the master is identical to its own address with a set R/$\overline{W}$ bit. The slave transmitter shifts the serial data out on SDA with the clock pulses that are generated by the master device. The slave device does not generate the clock, but it does hold SCL low while intervention of the CPU is required after a byte has been transmitted.

If the master requests data from the slave, the USCI module is automatically configured as a transmitter and UCTR and UCTXIFG become set. The SCL line is held low until the first data to be sent is written into the transmit buffer UCBxTXBUF. Then the address is acknowledged, the UCSTTIFG flag is cleared, and the data is transmitted. As soon as the data is transferred into the shift register, the UCTXIFG is set again. After the data is acknowledged by the master, the next data byte written into UCBxTXBUF is transmitted or, if the buffer is empty, the bus is stalled during the acknowledge cycle by holding SCL low until new data is written into UCBxTXBUF. If the master sends a NACK succeeded by a STOP condition, the UCSTPIFG flag is set. If the NACK is succeeded by a repeated START condition, the USCI I²C state machine returns to its address-reception state.

Figure 1-9 shows the slave transmitter operation.

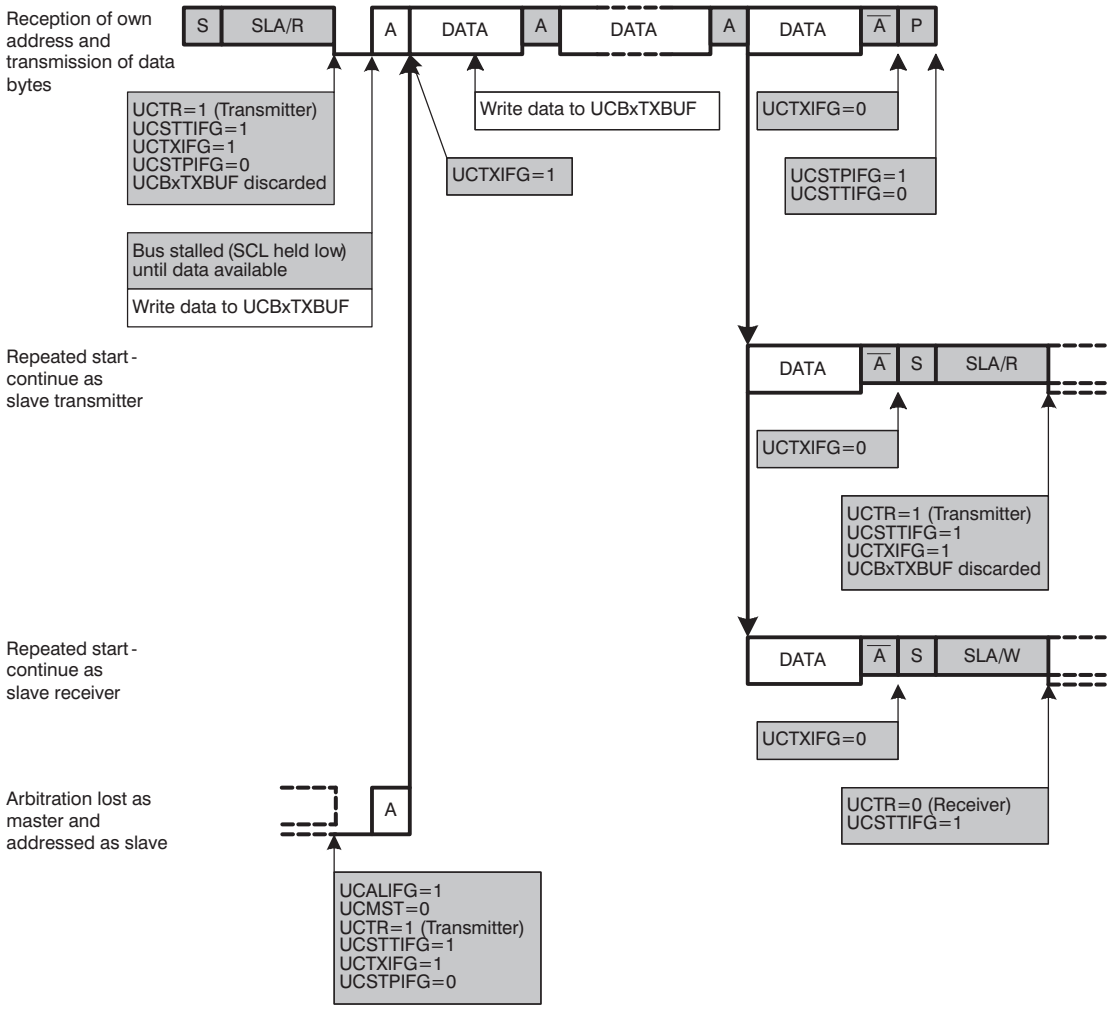

Figure 1-9. I²C Slave Transmitter Mode

1.3.4.1.2 I²C Slave Receiver Mode

Slave receiver mode is entered when the slave address transmitted by the master is identical to its own address and a cleared R/$\overline{W}$ bit is received. In slave receiver mode, serial data bits received on SDA are shifted in with the clock pulses that are generated by the master device. The slave device does not generate the clock, but it can hold SCL low if intervention of the CPU is required after a byte has been received.

If the slave should receive data from the master, the USCI module is automatically configured as a receiver and UCTR is cleared. After the first data byte is received, the receive interrupt flag UCRXIFG is set. The USCI module automatically acknowledges the received data and can receive the next data byte.

If the previous data was not read from the receive buffer UCBxRXBUF at the end of a reception, the bus is stalled by holding SCL low. As soon as UCBxRXBUF is read, the new data is transferred into UCBxRXBUF, an acknowledge is sent to the master, and the next data can be received.

Setting the UCTXNACK bit causes a NACK to be transmitted to the master during the next acknowledgment cycle. A NACK is sent even if UCBxRXBUF is not ready to receive the latest data. If the UCTXNACK bit is set while SCL is held low, the bus is released, a NACK is transmitted immediately, and UCBxRXBUF is loaded with the last received data. Because the previous data was not read, that data is lost. To avoid loss of data, the UCBxRXBUF must be read before UCTXNACK is set**.**

When the master generates a STOP condition, the UCSTPIFG flag is set.

If the master generates a repeated START condition, the USCI I²C state machine returns to its address reception state.

Figure 1-10 shows the I²C slave receiver operation.

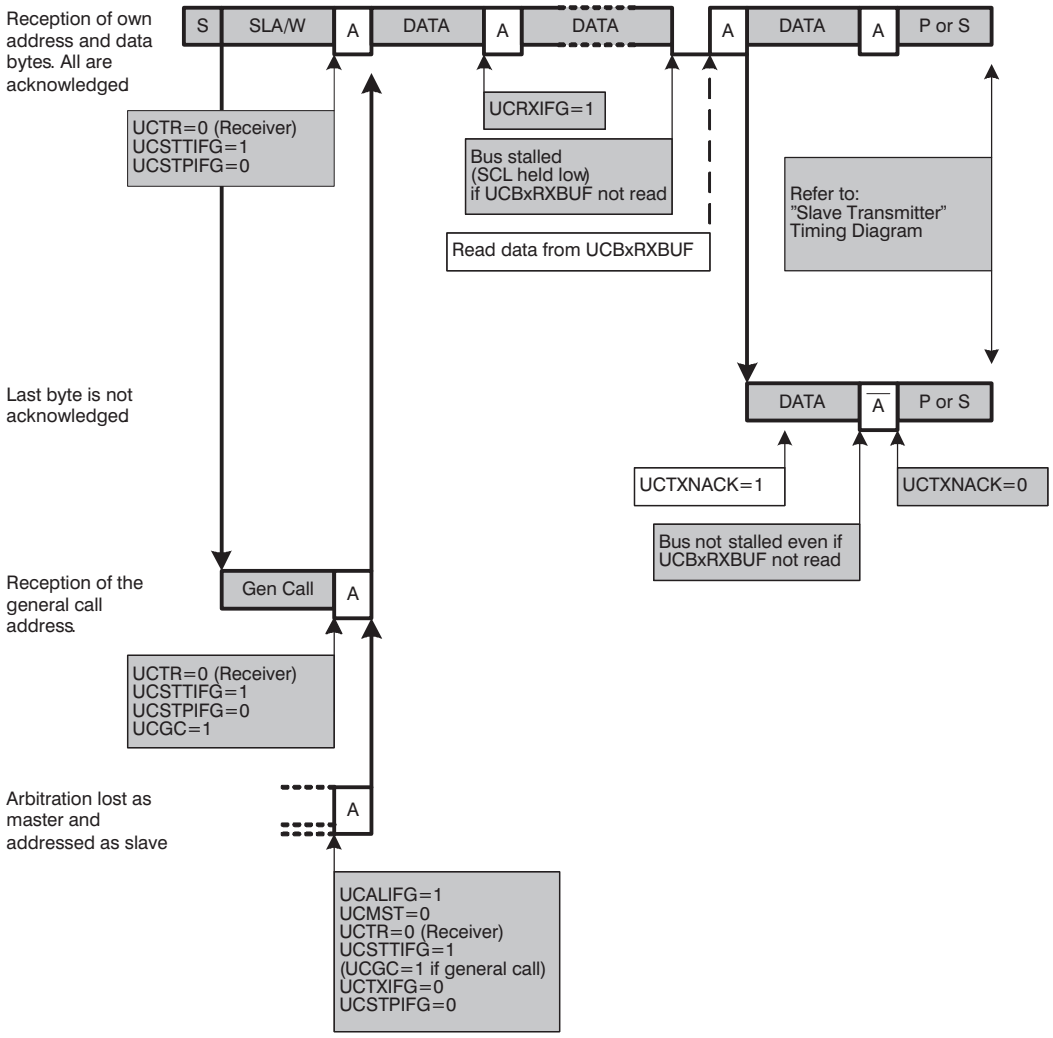

Figure 1-10. I²C Slave Receiver Mode

1.3.4.1.3 I²C Slave 10-Bit Addressing Mode

The 10-bit addressing mode is selected when UCA10 = 1 and is as shown in Figure 1-11. In 10-bit addressing mode, the slave is in receive mode after the full address is received. The USCI module indicates this by setting the UCSTTIFG flag while the UCTR bit is cleared. To switch the slave into transmitter mode, the master sends a repeated START condition together with the first byte of the address but with the R/$\overline{W}$ bit set. This sets the UCSTTIFG flag if it was previously cleared by software, and the USCI modules switches to transmitter mode with UCTR = 1.

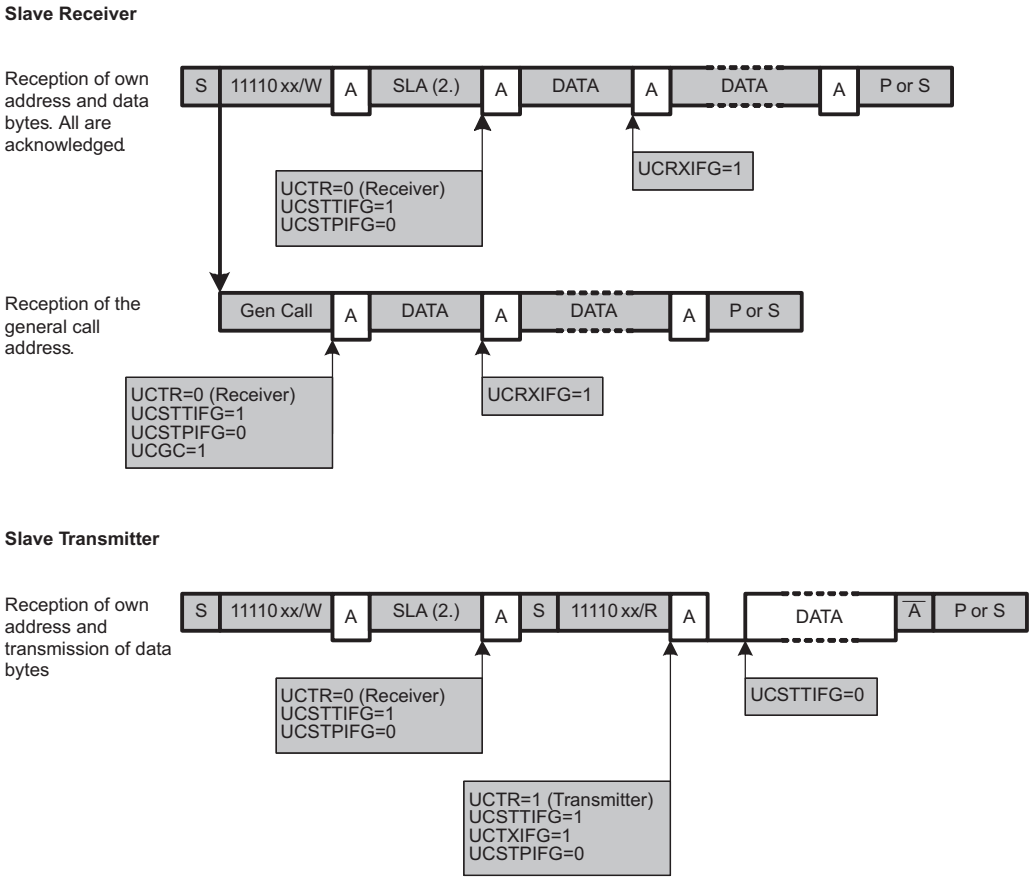

Figure 1-11. I²C Slave 10-Bit Addressing Mode

1.3.4.2 Master Mode

The USCI module is configured as an I²C master by selecting the I²C mode with UCMODEx = 11 and UCSYNC = 1 and setting the UCMST bit. When the master is part of a multi-master system, UCMM must be set and its own address must be programmed into the UCBxI2COA register. When UCA10 = 0, 7-bit addressing is selected. When UCA10 = 1, 10-bit addressing is selected. The UCGCEN bit selects if the USCI module responds to a general call.

1.3.4.2.1 I²C Master Transmitter Mode

After initialization, master transmitter mode is initiated by writing the desired slave address to the UCBxI2CSA register, selecting the size of the slave address with the UCSLA10 bit, setting UCTR for transmitter mode, and setting UCTXSTT to generate a START condition.

The USCI module checks if the bus is available, generates the START condition, and transmits the slave address. The UCTXIFG bit is set when the START condition is generated and the first data to be transmitted can be written into UCBxTXBUF. As soon as the slave acknowledges the address, the UCTXSTT bit is cleared.

NOTE: Handling of TXIFG in a multi-master system

In a multi-master system (UCMM =1), if the bus is unavailable, the USCI module waits and checks for bus release. Bus unavailability can occur even after the UCTXSTT bit has been set. While waiting for the bus to become available, the USCI may update the TXIFG based on SCL clock line activity. Checking the UCTXSTT bit to verify if the START condition has been sent ensures that the TXIFG is being serviced correctly.

The data written into UCBxTXBUF is transmitted if arbitration is not lost during transmission of the slave address. UCTXIFG is set again as soon as the data is transferred from the buffer into the shift register. If there is no data loaded to UCBxTXBUF before the acknowledge cycle, the bus is held during the acknowledge cycle with SCL low until data is written into UCBxTXBUF. Data is transmitted or the bus is held, as long as the UCTXSTP bit or UCTXSTT bit is not set.

Setting UCTXSTP generates a STOP condition after the next acknowledge from the slave. If UCTXSTP is set during the transmission of the slave's address or while the USCI module waits for data to be written into UCBxTXBUF, a STOP condition is generated, even if no data was transmitted to the slave. When transmitting a single byte of data, the UCTXSTP bit must be set while the byte is being transmitted or anytime after transmission begins, without writing new data into UCBxTXBUF. Otherwise, only the address is transmitted. When the data is transferred from the buffer to the shift register, UCTXIFG is set, indicating data transmission has begun, and the UCTXSTP bit may be set.

Setting UCTXSTT generates a repeated START condition. In this case, UCTR may be set or cleared to configure transmitter or receiver, and a different slave address may be written into UCBxI2CSA if desired.

If the slave does not acknowledge the transmitted data, the not-acknowledge interrupt flag UCNACKIFG is set. The master must react with either a STOP condition or a repeated START condition. If data was already written into UCBxTXBUF, it is discarded. If this data should be transmitted after a repeated START, it must be written into UCBxTXBUF again. Any set UCTXSTT is also discarded. To trigger a repeated START, UCTXSTT must be set again.

Figure 1-12 shows the I²C master transmitter operation.

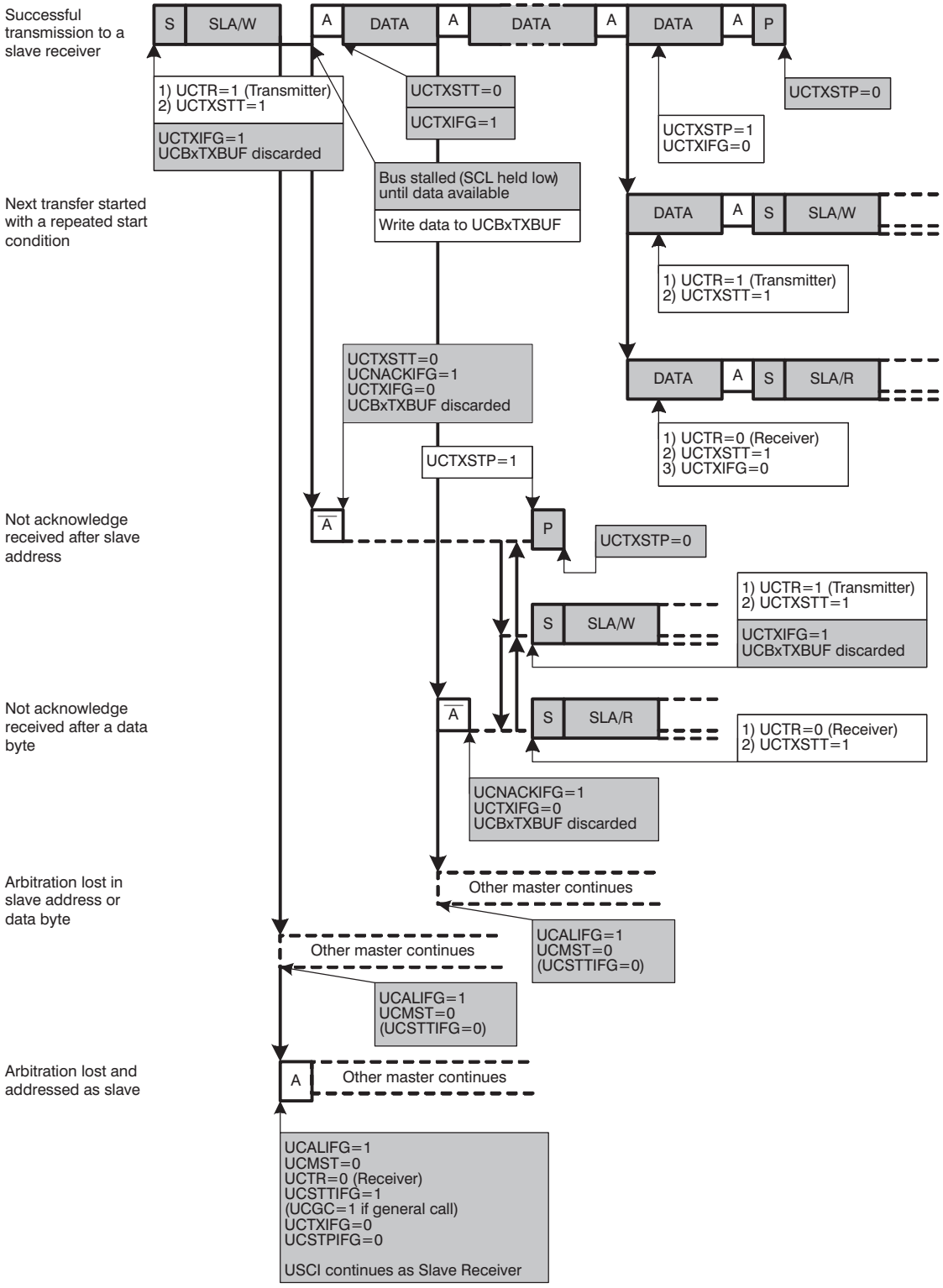

Figure 1-12. I²C Master Transmitter Mode

1.3.4.2.2 I²C Master Receiver Mode

After initialization, master receiver mode is initiated by writing the desired slave address to the UCBxI2CSA register, selecting the size of the slave address with the UCSLA10 bit, clearing UCTR for receiver mode, and setting UCTXSTT to generate a START condition.

The USCI module checks if the bus is available, generates the START condition, and transmits the slave address. As soon as the slave acknowledges the address, the UCTXSTT bit is cleared.

After the acknowledge of the address from the slave, the first data byte from the slave is received and acknowledged and the UCRXIFG flag is set. Data is received from the slave, as long as UCTXSTP or UCTXSTT is not set. If UCBxRXBUF is not read, the master holds the bus during reception of the last data bit and until the UCBxRXBUF is read.

If the slave does not acknowledge the transmitted address, the not-acknowledge interrupt flag UCNACKIFG is set. The master must react with either a STOP condition or a repeated START condition.

Setting the UCTXSTP bit generates a STOP condition. After setting UCTXSTP, a NACK followed by a STOP condition is generated after reception of the data from the slave, or immediately if the USCI module is currently waiting for UCBxRXBUF to be read.

If a master wants to receive a single byte only, the UCTXSTP bit must be set while the byte is being received. For this case, the UCTXSTT may be polled to determine when it is cleared:

```
            BIS.B   #UCTXSTT, &UCB0CTL1  ;Transmit START cond.
POLL_STT    BIT.B   #UCTXSTT, &UCB0CTL1  ;Poll UCTXSTT bit
            JC      POLL_STT             ;When cleared,
            BIS.B   #UCTXSTP, &UCB0CTL1  ;transmit STOP cond.
```

Setting UCTXSTT generates a repeated START condition. In this case, UCTR may be set or cleared to configure transmitter or receiver, and a different slave address may be written into UCBxI2CSA if desired.

NOTE: Repeated START

The UCTXSTT bit must be set before the last data byte is received; that is, immediately after the UCRXIFG is set and the UCRXBUF with the second to last byte is read, the UCTXSTT bit should be set.

NOTE: Consecutive master transactions without repeated START

When performing multiple consecutive I²C master transactions without the repeated START feature, the current transaction must be completed before the next one is initiated. This can be done by ensuring that the transmit STOP condition flag UCTXSTP is cleared before the next I²C transaction is initiated with setting UCTXSTT = 1. Otherwise, the current transaction might be affected.

Figure 1-13 shows the I²C master receiver operation.

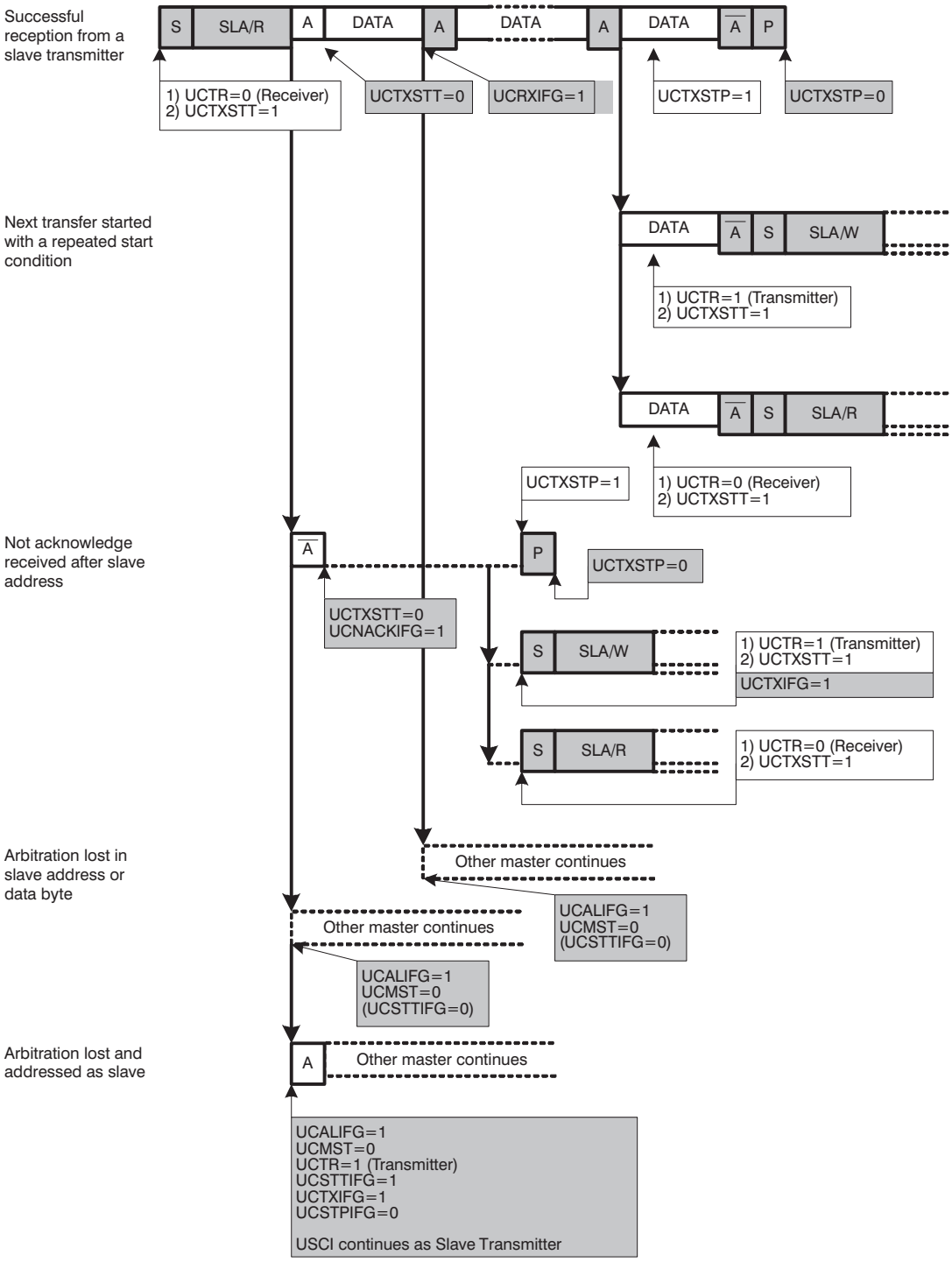

Figure 1-13. I²C Master Receiver Mode

1.3.4.2.3 I²C Master 10-Bit Addressing Mode

The 10-bit addressing mode is selected when UCSLA10 = 1 and is shown in Figure 1-14.

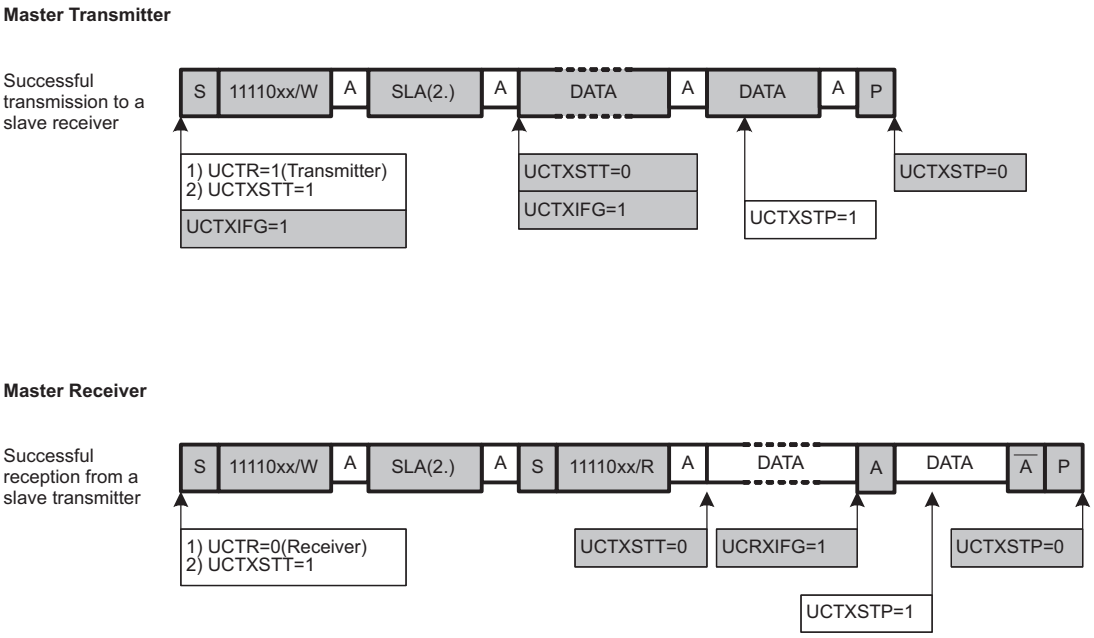

Figure 1-14. I²C Master 10-Bit Addressing Mode

1.3.4.3 Arbitration

If two or more master transmitters simultaneously start a transmission on the bus, an arbitration procedure is invoked. Figure 1-15 shows the arbitration procedure between two devices. The arbitration procedure uses the data presented on SDA by the competing transmitters. The first master transmitter that generates a logic high is overruled by the opposing master generating a logic low. The arbitration procedure gives priority to the device that transmits the serial data stream with the lowest binary value. The master transmitter that lost arbitration switches to the slave receiver mode and sets the arbitration lost flag UCALIFG. If two or more devices send identical first bytes, arbitration continues on the subsequent bytes.

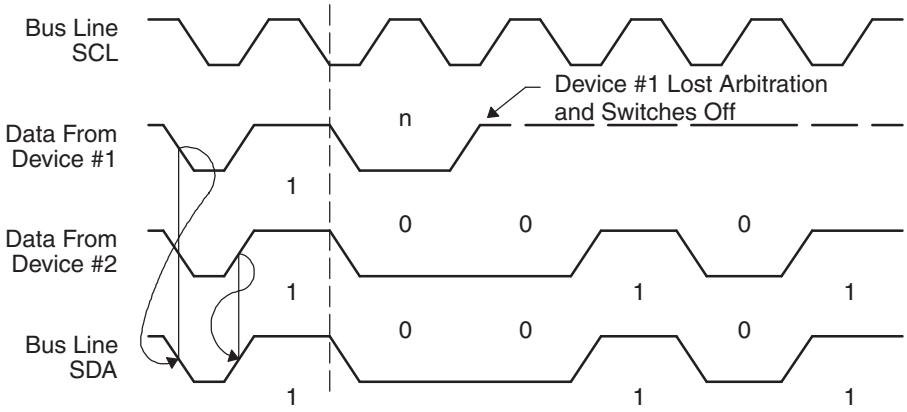

Figure 1-15. Arbitration Procedure Between Two Master Transmitters

If the arbitration procedure is in progress when a repeated START condition or STOP condition is transmitted on SDA, the master transmitters involved in arbitration must send the repeated START condition or STOP condition at the same position in the format frame. Arbitration is not allowed between:

- A repeated START condition and a data bit
- A STOP condition and a data bit
- A repeated START condition and a STOP condition

1.3.5 I²C Clock Generation and Synchronization

The I²C clock SCL is provided by the master on the I²C bus. When the USCI is in master mode, BITCLK is provided by the USCI bit clock generator and the clock source is selected with the UCSSELx bits. In slave mode, the bit clock generator is not used and the UCSSELx bits are don't care.

The 16-bit value of UCBRx in registers UCBxBR1 and UCBxBR0 is the division factor of the USCI clock source, BRCLK. The maximum bit clock that can be used in single master mode is $f_{BRCLK}/4$. In multi-master mode, the maximum bit clock is $f_{BRCLK}/8$. The BITCLK frequency is given by:

$f_{BitClock} = f_{BRCLK}/UCBRx$

The minimum high and low periods of the generated SCL are:

$t_{LOW,MIN} = t_{HIGH,MIN} = (UCBRx/2)/f_{BRCLK}$ when UCBRx is even

$t_{LOW,MIN} = t_{HIGH,MIN} = ((UCBRx – 1)/2)/f_{BRCLK}$ when UCBRx is odd

The USCI clock source frequency and the prescaler setting UCBRx must to be chosen such that the minimum low and high period times of the I²C specification are met.

During the arbitration procedure the clocks from the different masters must be synchronized. A device that first generates a low period on SCL overrules the other devices, forcing them to start their own low periods. SCL is then held low by the device with the longest low period. The other devices must wait for SCL to be released before starting their high periods. Figure 1-16 shows the clock synchronization. This allows a slow slave to slow down a fast master.

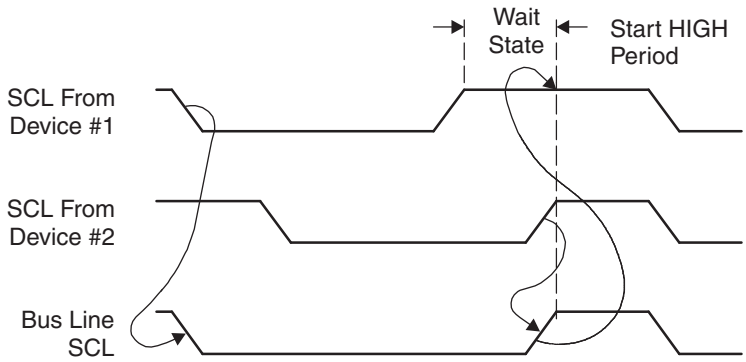

Figure 1-16. Synchronization of Two I²C Clock Generators During Arbitration

1.3.5.1 Clock Stretching

The USCI module supports clock stretching and also makes use of this feature as described in the Operation Mode sections.

The UCSCLLOW bit can be used to observe if another device pulls SCL low while the USCI module already released SCL due to the following conditions:

- USCI is acting as master and a connected slave drives SCL low.
- USCI is acting as master and another master drives SCL low during arbitration.

The UCSCLLOW bit is also active if the USCI holds SCL low because it is waiting as transmitter for data being written into UCBxTXBUF or as receiver for the data being read from UCBxRXBUF.

The UCSCLLOW bit might get set for a short time with each rising SCL edge because the logic observes the external SCL and compares it to the internally generated SCL.

1.3.6 Using the USCI Module in I²C Mode With Low-Power Modes

The USCI module provides automatic clock activation for use with low-power modes. When the USCI clock source is inactive because the device is in a low-power mode, the USCI module automatically activates it when needed, regardless of the control-bit settings for the clock source. The clock remains active until the USCI module returns to its idle condition. After the USCI module returns to the idle condition, control of the clock source reverts to the settings of its control bits.

In I²C slave mode, no internal clock source is required because the clock is provided by the external master. It is possible to operate the USCI in I²C slave mode while the device is in LPM4 and all internal clock sources are disabled. The receive or transmit interrupts can wake up the CPU from any low-power mode.

1.3.7 USCI Interrupts in I²C Mode

The USCI has only one interrupt vector that is shared for transmission, reception, and the state change. USCI_Ax and USC_Bx do not share the same interrupt vector.

Each interrupt flag has its own interrupt enable bit. When an interrupt is enabled and the GIE bit is set, the interrupt flag generates an interrupt request. DMA transfers are controlled by the UCTXIFG and UCRXIFG flags on devices with a DMA controller.

1.3.7.1 I²C Transmit Interrupt Operation

The UCTXIFG interrupt flag is set by the transmitter to indicate that UCBxTXBUF is ready to accept another character. An interrupt request is generated if UCTXIE and GIE are also set. UCTXIFG is automatically reset if a character is written to UCBxTXBUF or if a NACK is received. UCTXIFG is set when UCSWRST = 1 and the I²C mode is selected. UCTXIE is reset after a PUC or when UCSWRST = 1.

1.3.7.2 I²C Receive Interrupt Operation

The UCRXIFG interrupt flag is set when a character is received and loaded into UCBxRXBUF. An interrupt request is generated if UCRXIE and GIE are also set. UCRXIFG and UCRXIE are reset after a PUC signal or when UCSWRST = 1. UCRXIFG is automatically reset when UCxRXBUF is read.

1.3.7.3 I²C State Change Interrupt Operation

Table 1-1 describes the I²C state change interrupt flags.

Table 1-1. I²C State Change Interrupt Flags

Interrupt Flag	Interrupt Condition
UCALIFG	Arbitration-lost. Arbitration can be lost when two or more transmitters start a transmission simultaneously, or when the USCI operates as master but is addressed as a slave by another master in the system. The UCALIFG flag is set when arbitration is lost. When UCALIFG is set, the UCMST bit is cleared and the I²C controller becomes a slave.
UCNACKIFG	Not-acknowledge interrupt. This flag is set when an acknowledge is expected but is not received. UCNACKIFG is automatically cleared when a START condition is received.
UCSTTIFG	START condition detected interrupt. This flag is set when the I²C module detects a START condition together with its own address while in slave mode. UCSTTIFG is used in slave mode only and is automatically cleared when a STOP condition is received.
UCSTPIFG	STOP condition detected interrupt. This flag is set when the I²C module detects a STOP condition while in slave mode. UCSTPIFG is used in slave mode only and is automatically cleared when a START condition is received.

1.3.7.4 UCBxIV, Interrupt Vector Generator

The USCI interrupt flags are prioritized and combined to source a single interrupt vector. The interrupt vector register UCBxIV is used to determine which flag requested an interrupt. The highest-priority enabled interrupt generates a number in the UCBxIV register that can be evaluated or added to the PC to automatically enter the appropriate software routine. Disabled interrupts do not affect the UCBxIV value.

Any access, read or write, of the UCBxIV register automatically resets the highest-pending interrupt flag. If another interrupt flag is set, another interrupt is immediately generated after servicing the initial interrupt.

1.3.7.4.1 UCBxIV Software Example

The following software example shows the recommended use of UCBxIV. The UCBxIV value is added to the PC to automatically jump to the appropriate routine. The example is given for USCI_B0.

```
USCI_I2C_ISR
         ADD      &UCB0IV, PC  ; Add offset to jump table
         RETI                  ; Vector 0: No interrupt
         JMP      ALIFG_ISR    ; Vector 2: ALIFG
         JMP      NACKIFG_ISR  ; Vector 4: NACKIFG
         JMP      STTIFG_ISR   ; Vector 6: STTIFG
         JMP      STPIFG_ISR   ; Vector 8: STPIFG
         JMP      RXIFG_ISR    ; Vector 10: RXIFG
TXIFG_ISR                     ; Vector 12
         ...                  ; Task starts here
         RETI                 ; Return
ALIFG_ISR                     ; Vector 2
         ...                  ; Task starts here
         RETI                 ; Return
NACKIFG_ISR                   ; Vector 4
         ...                  ; Task starts here
         RETI                 ; Return
STTIFG_ISR                    ; Vector 6
         ...                  ; Task starts here
         RETI                 ; Return
STPIFG_ISR                    ; Vector 8
         ...                  ; Task starts here
         RETI                 ; Return
RXIFG_ISR                     ; Vector 10
         ...                  ; Task starts here
         RETI                 ; Return
```

1.4 USCI_B I2C Mode Registers

The USCI registers applicable in I2C mode are listed in Table 1-2. The base address can be found in the device-specific data sheet. The address offsets are listed in Table 1-2.

Table 1-2. USCI_B Registers

Offset	Acronym	Register Name	Type	Access	Reset	Section
00h	UCBxCTLW0	USCI_Bx Control Word 0	Read/write	Word	0101h	
00h	UCBxCTL1	USCI_Bx Control 1	Read/write	Byte	01h	Section 1.4.2
01h	UCBxCTL0	USCI_Bx Control 0	Read/write	Byte	01h	Section 1.4.1
06h	UCBxBRW	USCI_Bx Bit Rate Control Word	Read/write	Word	0000h	
06h	UCBxBR0	USCI_Bx Bit Rate Control 0	Read/write	Byte	00h	Section 1.4.3
07h	UCBxBR1	USCI_Bx Bit Rate Control 1	Read/write	Byte	00h	Section 1.4.4
0Ah	UCBxSTAT	USCI_Bx Status	Read/write	Byte	00h	Section 1.4.5
0Bh		Reserved - reads zero	Read	Byte	00h	
0Ch	UCBxRXBUF	USCI_Bx Receive Buffer	Read/write	Byte	00h	Section 1.4.6
0Dh		Reserved - reads zero	Read	Byte	00h	
0Eh	UCBxTXBUF	USCI_Bx Transmit Buffer	Read/write	Byte	00h	Section 1.4.7
0Fh		Reserved - reads zero	Read	Byte	00h	
10h	UCBxI2COA	USCI_Bx I2C Own Address	Read/write	Word	0000h	Section 1.4.8
12h	UCBxI2CSA	USCI_Bx I2C Slave Address	Read/write	Word	0000h	Section 1.4.9
1Ch	UCBxICTL	USCI_Bx Interrupt Control	Read/write	Word	0200h	
1Ch	UCBxIE	USCI_Bx Interrupt Enable	Read/write	Byte	00h	Section 1.4.10
1Dh	UCBxIFG	USCI_Bx Interrupt Flag	Read/write	Byte	02h	Section 1.4.11
1Eh	UCBxIV	USCI_Bx Interrupt Vector	Read	Word	0000h	Section 1.4.12

1.4.1 UCBxCTL0 Register

USCI_Bx Control Register 0

Figure 1-17. UCBxCTL0 Register

7	6	5	4	3	2	1	0
UCA10	UCSLA10	UCMM	Reserved	UCMST	UCMODEx		UCSYNC
rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	r-1

Can be modified only when UCSWRST = 1.

Table 1-3. UCBxCTL0 Register Description

Bit	Field	Type	Reset	Description
7	UCA10	RW	0h	Own addressing mode select 0b = Own address is a 7-bit address 1b = Own address is a 10-bit address
6	UCSLA10	RW	0h	Slave addressing mode select 0b = Address slave with 7-bit address 1b = Address slave with 10-bit address
5	UCMM	RW	0h	Multi-master environment select 0b = Single master environment. There is no other master in the system. The address compare unit is disabled. 1b = Multi-master environment
4	Reserved	R	0h	Reserved. Always reads as 0.
3	UCMST	RW	0h	Master mode select. When a master loses arbitration in a multi-master environment (UCMM = 1), the UCMST bit is automatically cleared and the module acts as slave. 0b = Slave mode 1b = Master mode
2-1	UCMODEx	RW	0h	USCI mode. The UCMODEx bits select the synchronous mode when UCSYNC = 1. 00b = 3-pin SPI 01b = 4-pin SPI (master/slave enabled if STE = 1) 10b = 4-pin SPI (master/slave enabled if STE = 0) 11b = I^2C mode
0	UCSYNC	R	1h	Synchronous mode enable 0b = Asynchronous mode 1b = Synchronous mode

1.4.2 UCBxCTL1 Register

USCI_Bx Control Register 1

Figure 1-18. UCBxCTL1 Register

7	6	5	4	3	2	1	0
UCSSELx		Reserved	UCTR	UCTXNACK	UCTXSTP	UCTXSTT	UCSWRST
rw-0	rw-0	r0	rw-0	rw-0	rw-0	rw-0	rw-1

Can be modified only when UCSWRST = 1.

Table 1-4. UCBxCTL1 Register Description

Bit	Field	Type	Reset	Description
7-6	UCSSELx	RW	0h	USCI clock source select. These bits select the BRCLK source clock. 00b = UCLKI 01b = ACLK 10b = SMCLK 11b = SMCLK
5	Reserved	RW	0h	Reserved. Always reads as 0.
4	UCTR	RW	0h	Transmitter or receiver 0b = Receiver 1b = Transmitter
3	UCTXNACK	RW	0h	Transmit a NACK. UCTXNACK is automatically cleared after a NACK is transmitted. 0b = Acknowledge normally 1b = Generate NACK
2	UCTXSTP	RW	0h	Transmit STOP condition in master mode. Ignored in slave mode. In master receiver mode, the STOP condition is preceded by a NACK. UCTXSTP is automatically cleared after STOP is generated. 0b = No STOP generated 1b = Generate STOP
1	UCTXSTT	RW	0h	Transmit START condition in master mode. Ignored in slave mode. In master receiver mode, a repeated START condition is preceded by a NACK. UCTXSTT is automatically cleared after START condition and address information is transmitted. Ignored in slave mode. 0b = Do not generate START condition 1b = Generate START condition
0	UCSWRST	RW	1h	Software reset enable 0b = Disabled. USCI reset released for operation. 1b = Enabled. USCI logic held in reset state.

1.4.3 UCBxBR0 Register

USCI_Bx Baud Rate Control Register 0

Figure 1-19. UCBxBR0 Register

7	6	5	4	3	2	1	0
UCBRx							
rw	rw	rw	rw	rw	rw	rw	rw

Can be modified only when UCSWRST = 1.

Table 1-5. UCBxBR0 Register Description

Bit	Field	Type	Reset	Description
7-0	UCBRx	RW	undefined	Bit clock prescaler low byte. The 16-bit value of (UCxxBR0 + UCxxBR1 × 256) forms the prescaler value UCBRx.

1.4.4 UCBxBR1 Register

USCI_Bx Baud Rate Control Register 1

Figure 1-20. UCBxBR1 Register

7	6	5	4	3	2	1	0
UCBRx							
rw	rw	rw	rw	rw	rw	rw	rw

Can be modified only when UCSWRST = 1.

Table 1-6. UCBxBR1 Register Description

Bit	Field	Type	Reset	Description
7-0	UCBRx	RW	undefined	Bit clock prescaler high byte. The 16-bit value of (UCxxBR0 + UCxxBR1 × 256) forms the prescaler value UCBRx.

1.4.5 UCBxSTAT Register

USCI_Bx Status Register

Figure 1-21. UCBxSTAT Register

7	6	5	4	3	2	1	0
Reserved	UCSCLLOW	UCGC	UCBBUSY	Reserved			
rw-0	r-0	rw-0	r-0	r0	r0	r0	r0

Table 1-7. UCBxSTAT Register Description

Bit	Field	Type	Reset	Description
7	Reserved	RW	0h	Reserved. Always reads as 0.
6	UCSCLLOW	R	0h	SCL low 0b = SCL is not held low. 1b = SCL is held low.
5	UCGC	RW	0h	General call address received. UCGC is automatically cleared when a START condition is received. 0b = No general call address received 1b = General call address received
4	UCBBUSY	R	0h	Bus busy 0b = Bus inactive 1b = Bus busy
3-0	Reserved	R	0h	Reserved. Always reads as 0.

1.4.6 UCBxRXBUF Register

USCI_Bx Receive Buffer Register

Figure 1-22. UCBxRXBUF Register

7	6	5	4	3	2	1	0
UCRXBUFx							
r	r	r	r	r	r	r	r

Table 1-8. UCBxRXBUF Register Description

Bit	Field	Type	Reset	Description
7-0	UCRXBUFx	R	undefined	The receive-data buffer is user accessible and contains the last received character from the receive shift register. Reading UCBxRXBUF resets UCRXIFG.

1.4.7 UCBxTXBUF Register

USCI_Bx Transmit Buffer Register

Figure 1-23. UCBxTXBUF Register

7	6	5	4	3	2	1	0
UCTXBUFx							
rw	rw	rw	rw	rw	rw	rw	rw

Table 1-9. UCBxTXBUF Register Description

Bit	Field	Type	Reset	Description
7-0	UCTXBUFx	RW	undefined	The transmit data buffer is user accessible and holds the data waiting to be moved into the transmit shift register and transmitted. Writing to the transmit data buffer clears UCTXIFG.

1.4.8 UCBxI2COA Register

USCIBx I2C Own Address Register

Figure 1-24. UCBxI2COA Register

15	14	13	12	11	10	9	8
UCGCEN	Reserved					I2COAx	
rw-0	r0	r0	r0	r0	r0	rw-0	rw-0
7	6	5	4	3	2	1	0
I2COAx							
rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0

Can be modified only when UCSWRST = 1.

Table 1-10. UCBxI2COA Register Description

Bit	Field	Type	Reset	Description
15	UCGCEN	RW	0h	General call response enable 0b = Do not respond to a general call 1b = Respond to a general call
14-10	Reserved	R	0h	Reserved. Always reads as 0.
9-0	I2COAx	RW	0h	I2C own address. The I2COAx bits contain the local address of the USCI_Bx I2C controller. The address is right justified. In 7-bit addressing mode, bit 6 is the MSB and bits 9-7 are ignored. In 10-bit addressing mode, bit 9 is the MSB.

1.4.9 UCBxI2CSA Register

USCI_Bx I2C Slave Address Register

Figure 1-25. UCBxI2CSA Register

15	14	13	12	11	10	9	8
Reserved						I2CSAx	
r0	r0	r0	r0	r0	r0	rw-0	rw-0
7	6	5	4	3	2	1	0
I2CSAx							
rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0

Table 1-11. UCBxI2CSA Register Description

Bit	Field	Type	Reset	Description
15-10	Reserved	R	0h	Reserved. Always reads as 0.
9-0	I2CSAx	RW	0h	I2C slave address. The I2CSAx bits contain the slave address of the external device to be addressed by the USCI_Bx module. It is only used in master mode. The address is right justified. In 7-bit slave addressing mode, bit 6 is the MSB and bits 9-7 are ignored. In 10-bit slave addressing mode, bit 9 is the MSB.

1.4.10 UCBxIE Register

USCI_Bx I2C Interrupt Enable Register

Figure 1-26. UCBxIE Register

7	6	5	4	3	2	1	0
Reserved		UCNACKIE	UCALIE	UCSTPIE	UCSTTIE	UCTXIE	UCRXIE
r-0	r-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0

Table 1-12. UCBxIE Register Description

Bit	Field	Type	Reset	Description
7-6	Reserved	R	0h	Reserved. Always reads as 0.
5	UCNACKIE	RW	0h	Not-acknowledge interrupt enable 0b = Interrupt disabled 1b = Interrupt enabled
4	UCALIE	RW	0h	Arbitration lost interrupt enable 0b = Interrupt disabled 1b = Interrupt enabled
3	UCSTPIE	RW	0h	STOP condition interrupt enable 0b = Interrupt disabled 1b = Interrupt enabled
2	UCSTTIE	RW	0h	START condition interrupt enable 0b = Interrupt disabled 1b = Interrupt enabled
1	UCTXIE	RW	0h	Transmit interrupt enable 0b = Interrupt disabled 1b = Interrupt enabled
0	UCRXIE	RW	0h	Receive interrupt enable 0b = Interrupt disabled 1b = Interrupt enabled

1.4.11 UCBxIFG Register

USCI_Bx I2C Interrupt Flag Register

Figure 1-27. UCBxIFG Register

7	6	5	4	3	2	1	0
Reserved		UCNACKIFG	UCALIFG	UCSTPIFG	UCSTTIFG	UCTXIFG	UCRXIFG
r-0	r-0	rw-0	rw-0	rw-0	rw-0	rw-1	rw-0

Table 1-13. UCBxIFG Register Description

Bit	Field	Type	Reset	Description
7-6	Reserved	R	0h	Reserved. Always reads as 0.
5	UCNACKIFG	RW	0h	Not-acknowledge received interrupt flag. UCNACKIFG is automatically cleared when a START condition is received. 0b = No interrupt pending 1b = Interrupt pending
4	UCALIFG	RW	0h	Arbitration lost interrupt flag 0b = No interrupt pending 1b = Interrupt pending
3	UCSTPIFG	RW	0h	STOP condition interrupt flag. UCSTPIFG is automatically cleared when a START condition is received. 0b = No interrupt pending 1b = Interrupt pending
2	UCSTTIFG	RW	0h	START condition interrupt flag. UCSTTIFG is automatically cleared if a STOP condition is received. 0b = No interrupt pending 1b = Interrupt pending
1	UCTXIFG	RW	0h	USCI transmit interrupt flag. UCTXIFG is set when UCBxTXBUF is empty. 0b = No interrupt pending 1b = Interrupt pending
0	UCRXIFG	RW	0h	USCI receive interrupt flag. UCRXIFG is set when UCBxRXBUF has received a complete character. 0b = No interrupt pending 1b = Interrupt pending

1.4.12 UCBxIV Register

USCI_Bx Interrupt Vector Register

Figure 1-28. UCBxIV Register

15	14	13	12	11	10	9	8
UCIVx							
r0	r0	r0	r0	r0	r0	r0	r0
7	6	5	4	3	2	1	0
UCIVx							
r0	r0	r0	r0	r-0	r-0	r-0	r0

Table 1-14. UCBxIV Register Description

Bit	Field	Type	Reset	Description
15-0	UCIVx	R	0h	USCI interrupt vector value 00h = No interrupt pending 02h = Interrupt Source: Arbitration lost; Interrupt Flag: UCALIFG; Interrupt Priority: Highest 04h = Interrupt Source: Not acknowledgment; Interrupt Flag: UCNACKIFG 06h = Interrupt Source: Start condition received; Interrupt Flag: UCSTTIFG 08h = Interrupt Source: Stop condition received; Interrupt Flag: UCSTPIFG 0Ah = Interrupt Source: Data received; Interrupt Flag: UCRXIFG 0Ch = Interrupt Source: Transmit buffer empty; Interrupt Flag: UCTXIFG; Interrupt Priority: Lowest

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products (also referred to herein as “components”) are sold subject to TI’s terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI’s terms and conditions of sale of semiconductor products. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

TI assumes no liability for applications assistance or the design of Buyers’ products. Buyers are responsible for their products and applications using TI components. To minimize the risks associated with Buyers’ products and applications, Buyers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI components or services are used. Information published by TI regarding third-party products or services does not constitute a license to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of significant portions of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI components or services with statements different from or beyond the parameters stated by TI for that component or service voids all express and any implied warranties for the associated TI component or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences, lessen the likelihood of failures that might cause harm and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI’s goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed a special agreement specifically governing such use.

Only those TI components which TI has specifically designated as military grade or “enhanced plastic” are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components which have ***not*** been so designated is solely at the Buyer's risk, and that Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components as meeting ISO/TS16949 requirements, mainly for automotive use. In any case of use of non-designated products, TI will not be responsible for any failure to meet ISO/TS16949.

Products		**Applications**	
Audio	www.ti.com/audio	Automotive and Transportation	www.ti.com/automotive
Amplifiers	amplifier.ti.com	Communications and Telecom	www.ti.com/communications
Data Converters	dataconverter.ti.com	Computers and Peripherals	www.ti.com/computers
DLP® Products	www.dlp.com	Consumer Electronics	www.ti.com/consumer-apps
DSP	dsp.ti.com	Energy and Lighting	www.ti.com/energy
Clocks and Timers	www.ti.com/clocks	Industrial	www.ti.com/industrial
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Security	www.ti.com/security
Power Mgmt	power.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video
RFID	www.ti-rfid.com		
OMAP Applications Processors	www.ti.com/omap	**TI E2E Community**	e2e.ti.com
Wireless Connectivity	www.ti.com/wirelessconnectivity		

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265