Lecture 9
USCI Module

Texas Instruments Incorporated
University of Beira Interior (PT)

Pedro Dinis Gaspar, António Espírito Santo, Bruno Ribeiro, Humberto Santos
University of Beira Interior, Electromechanical Engineering Department
www.msp430.ubi.pt

>> Contents

Contents

USCI module introduction (1/3)

- **Although supporting UART, SPI and I²C, the USCI (Universal Serial Communication Interface) module is a communications interface specially designed to interconnect with high-speed industrial protocols:**
  - LIN (Local interconnect Network), used for low-cost modules in cars e.g. door modules, alarms, rain-sensors;
  - IrDA (Infrared Data Association).

- **The USCI module is available in the following devices:**
  - MSP430F5xx;
  - MSP430F4xx and MSP430FG41xx;
  - MSP430F2xx.

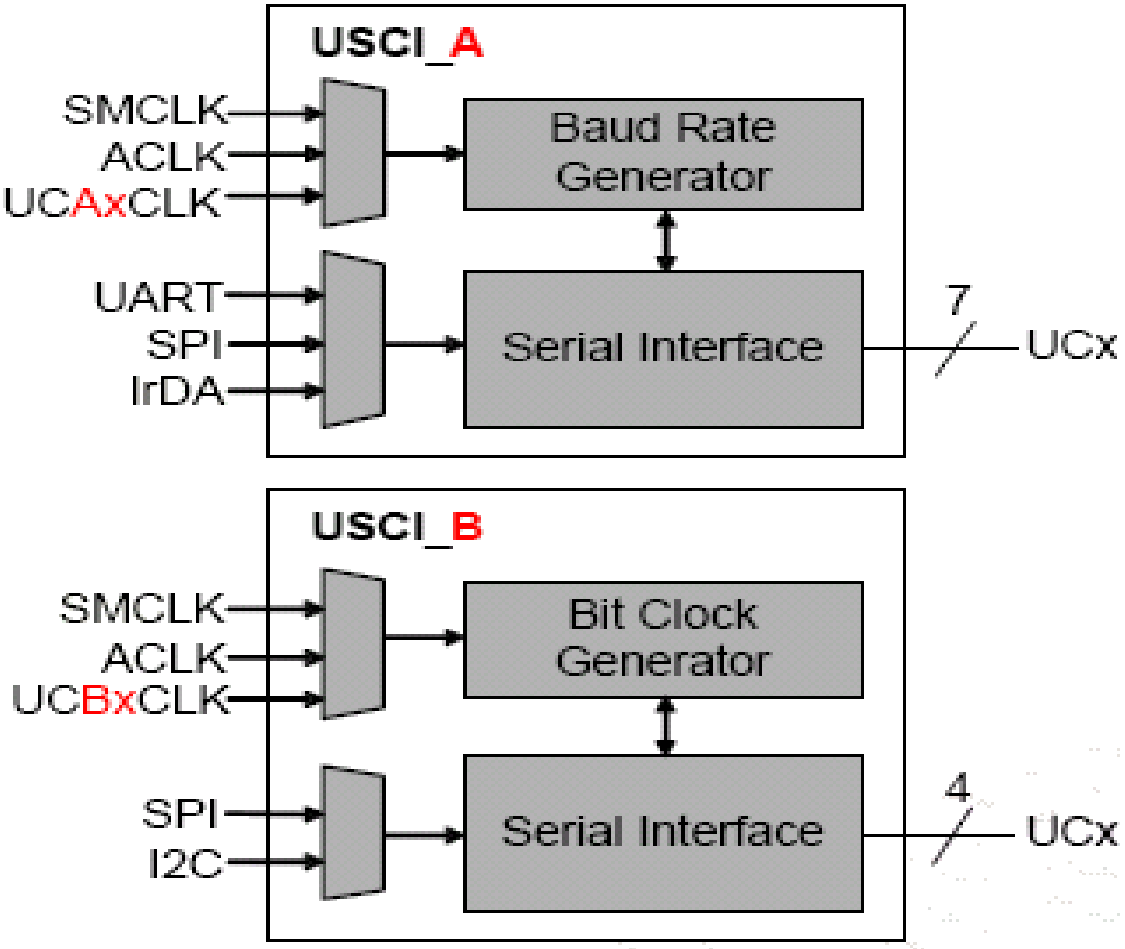
USCI module introduction (2/3)

The USCI module supports:

- Low power operating modes (with auto-start);
- Two individual blocks:
  - USCI_A: UART and SPI;
  - USCI_B: SPI and I^2C.
- Double buffered TX/RX;
- Baud rate/bit clock generator:
  - With auto-baud rate detect;
  - Flexible clock source.
- RX glitch suppression;
- DMA enabled;
- Error detection.

USCI module introduction (3/3)

❑ USCI block diagram:

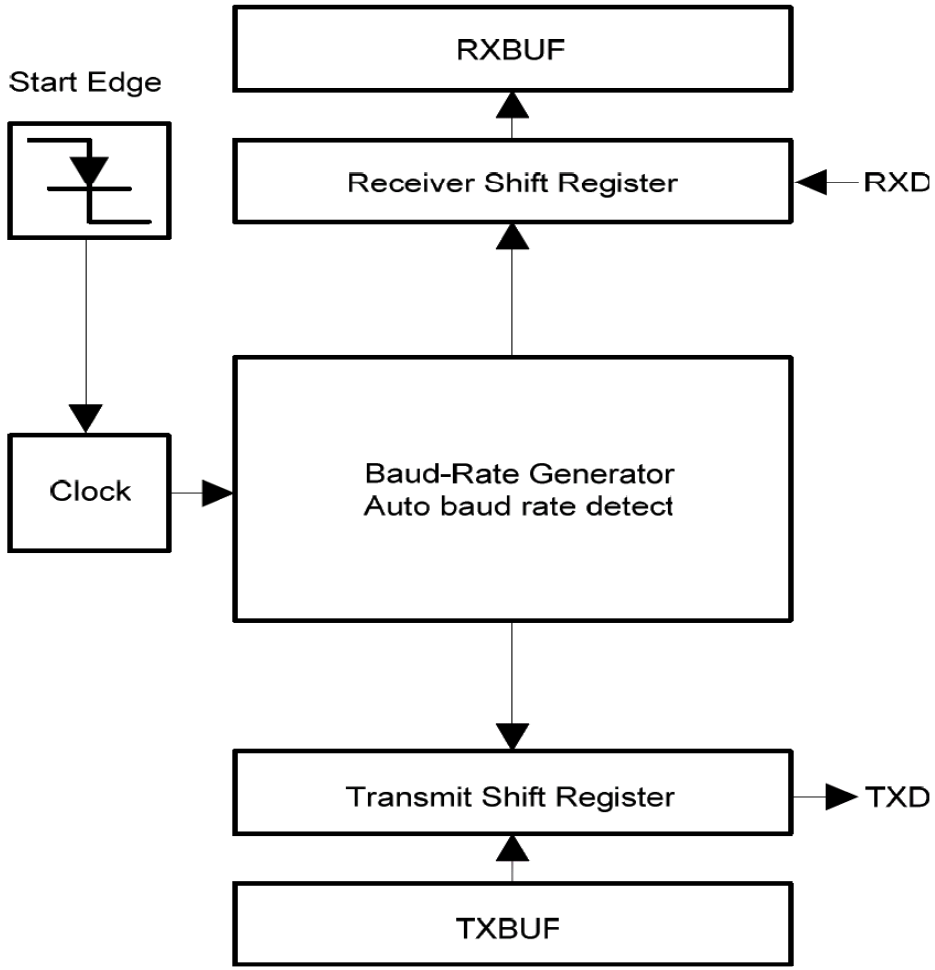


USCI operation: UART mode (1/17)

- In asynchronous mode, the USCI_Ax modules connect the MSP430 to an external system via two external pins, UCAxRXD and UCAxTXD;
- UART mode is selected when the UCSYNC bit is cleared;
- USCI transmits and receives characters asynchronously;
- Timing for each character is based on the selected baud rate of the USCI;
- Transmit and receive use the same clock frequency leading to the same baud rate;

USCI operation: UART mode (2/17)

❑ USCI operation in UART mode block diagram:



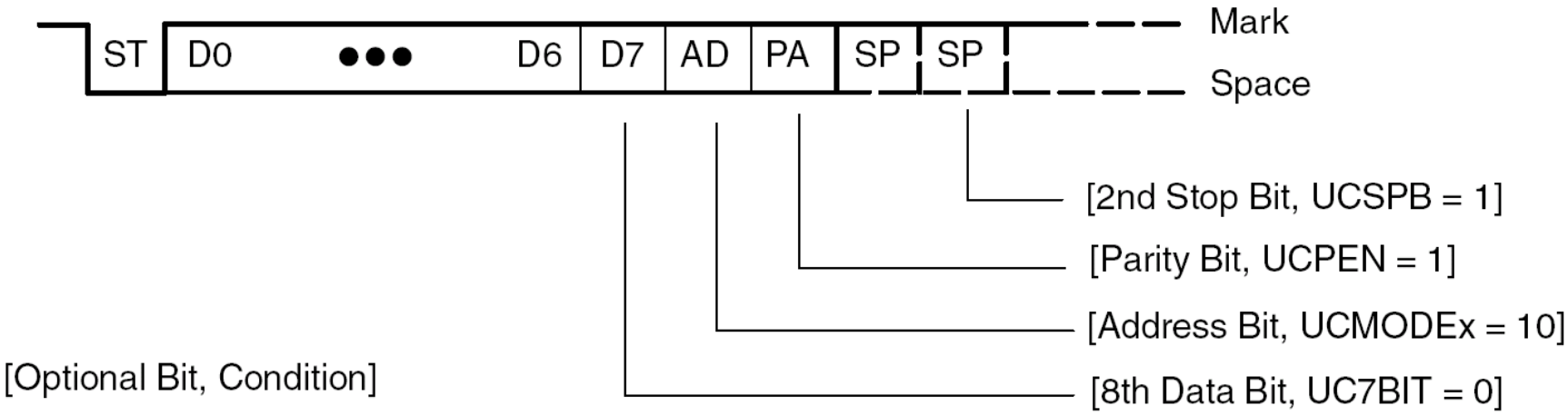
USCI operation: UART mode (3/17)

Recommended initialization/re-configuration process:

- Set UCSWRST (BIS.B #UCSWRST,&UCAxCTL1);
- Initialize all USCI registers with UCSWRST = 1 (including UCAxCTL1);
- Configure ports;
- Clear UCSWRST via software:
  (BIC.B #UCSWRST,&UCAxCTL1);
- Enable interrupts (optional) via UCAxRXIE and/or UCAxTXIE.

USCI operation: UART mode (4/17)

- **Character format specified as follows:**
  - Start bit;
  - Seven or eight data bits;
  - Even/odd/no parity bit;
  - Address bit (address-bit mode);
  - One or two stop bits.

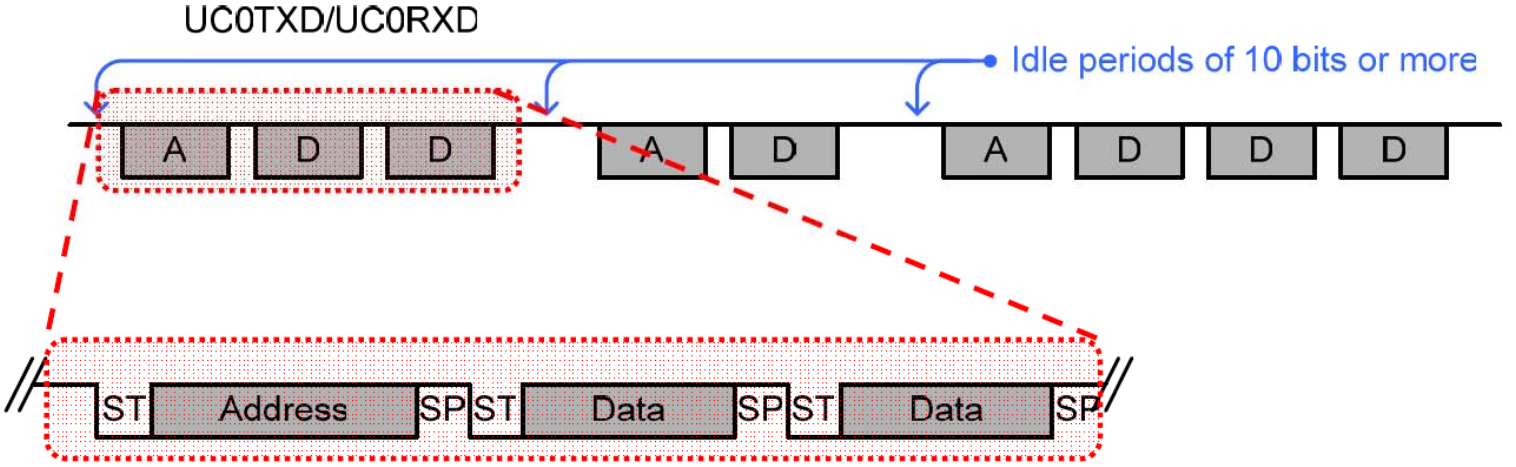


- The UCMSB bit controls the direction of the transfer and selects LSB (usual in UART communication) or MSB first.

USCI operation: UART mode (5/17)

❑ Asynchronous communication formats:

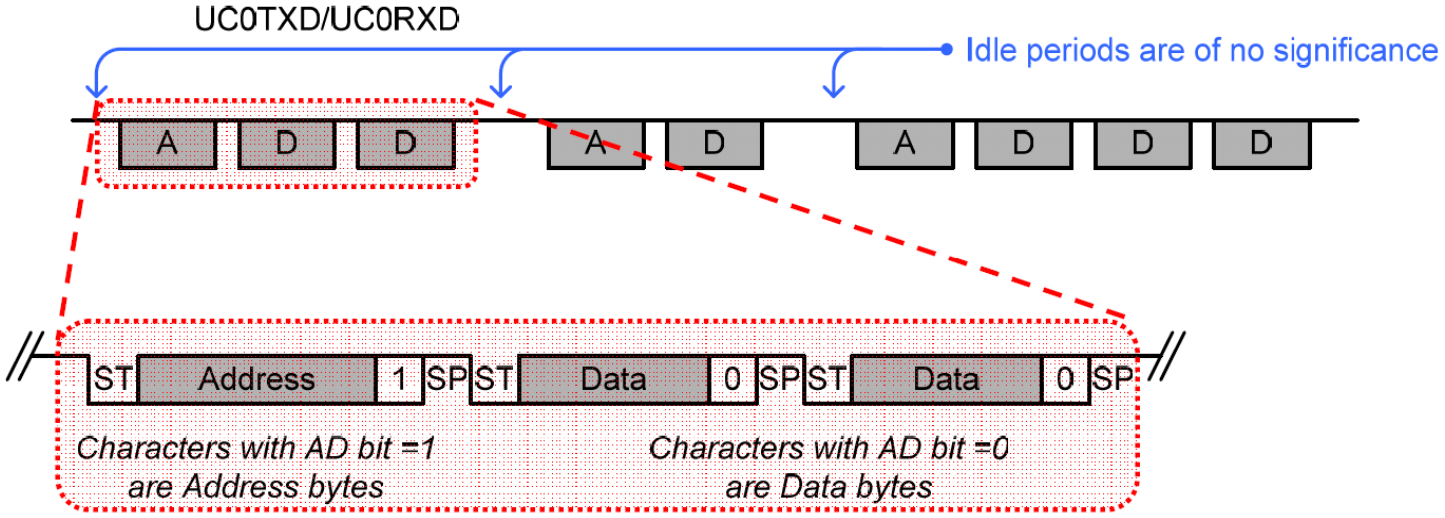
- **Idle-line multiprocessor communication protocol** (minimum of two devices):
  - IDLE is detected after > 10 periods of continuous marks after the stop bit;
  - The first character after IDLE is an address;
  - Can be programmed to receive only address characters.



USCI operation: UART mode (6/17)

❑ Asynchronous communication formats (continued):

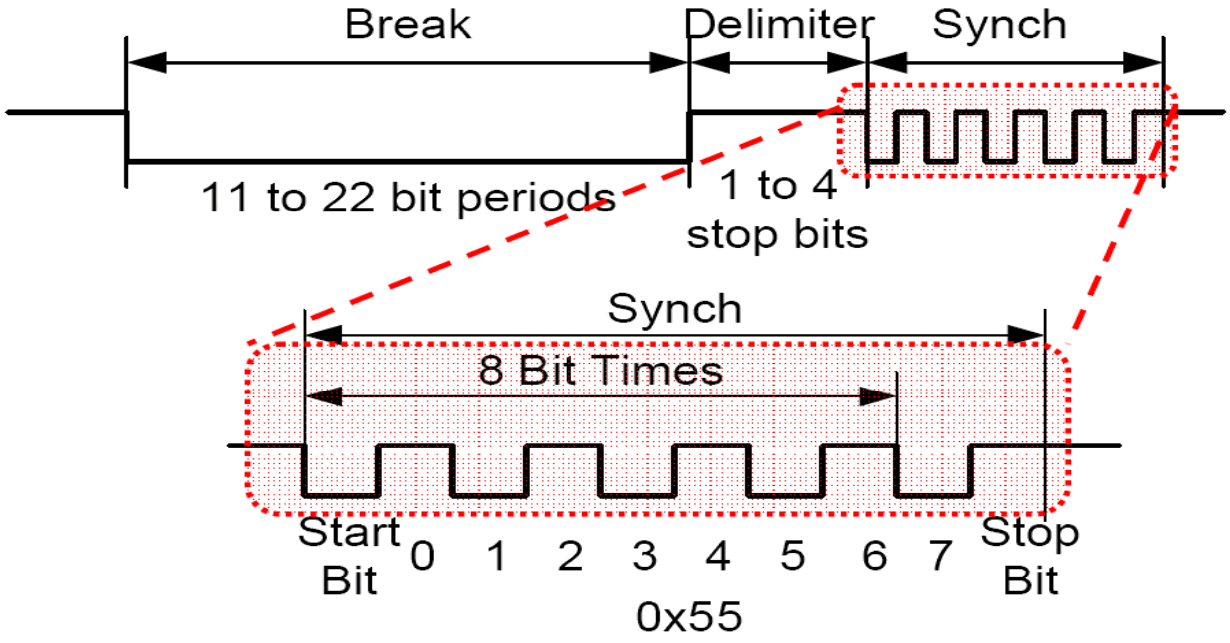
- **Address-bit multiprocessor communication protocol** (minimum of three devices):
  - An extra bit in the received character marks an address character;
  - UART can be programmed to receive only address characters.



USCI operation: UART mode (7/17)

❑ Automatic baud rate detection (UCMODEx = 11):

- Data frame is preceded by a synchronization sequence:
  - Break: Detected when 11 or more continuous zeros (spaces) are received;
  - Synch field: Data 055h inside a byte field.



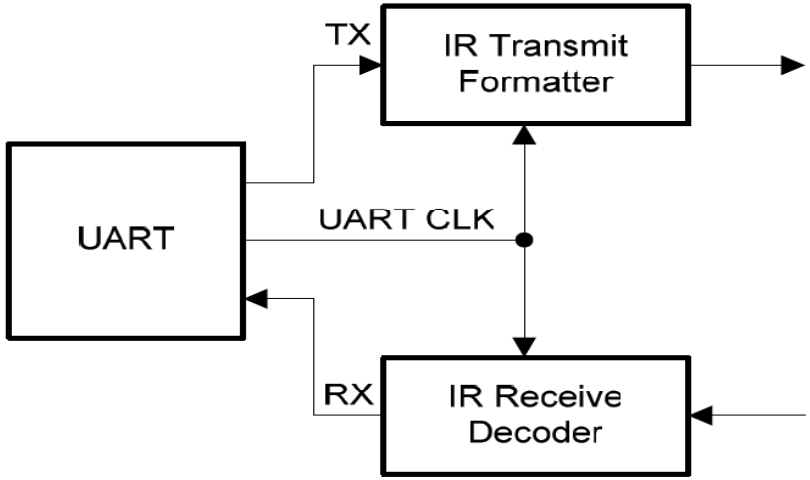
USCI operation: UART mode (8/17)

❑ Automatic baud rate detection (UCMODEx = 11):

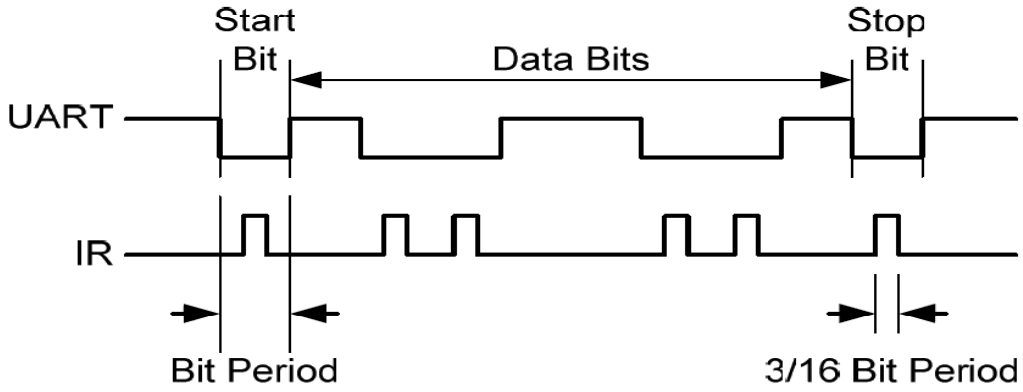
- The baud rate is calculated from a valid SYNC;
- Auto baud rate value stored in UxBR1, UxBR0 and UxMCTL (modulation pattern);
- BREAK time-out detect in hardware;
- Programmable delimiter time;

USCI operation: UART mode (9/17)

- **IrDA encoder and decoder (UCIREN = 1):**



UART frame vs IR frame



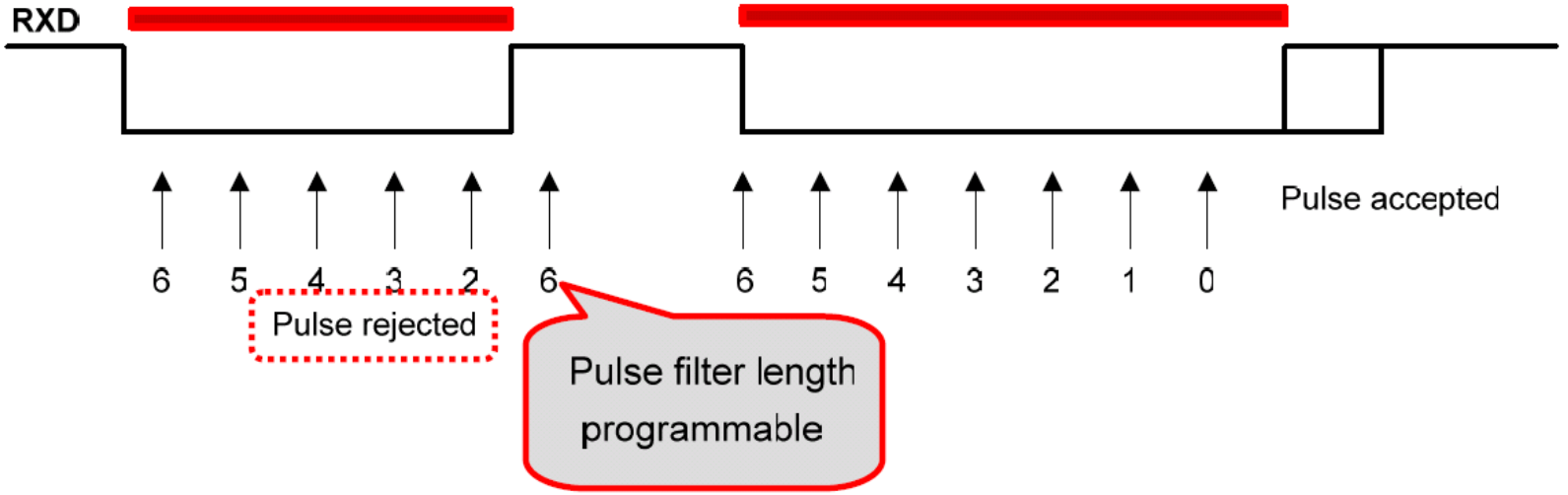
USCI operation: UART mode (10/17)

- **IrDA encoder and decoder (UCIREN = 1):**
  - **IrDA encoding:**
    - Encoder sends a pulse for every zero bit in the transmit bit stream coming from the UART;
    - Pulse duration (defined by UCIRTXPLx bits) specifies the number of half clock periods of the clock (UCIRTXCLK);
    - Oversampling baud rate generator allows selection of IrDA standard 3/16 bit length.

USCI operation: UART mode (11/17)

❑ IrDA encoder and decoder (UCIREN = 1):

- **IrDA decoding:**
- Programmable low or high pulse detection (UCIRRXPL) by the decoder;
- Programmable received pulse length filter adds noise filter capability as well as glitch detection.



USCI operation: UART mode (12/17)

- **Automatic error detection:**
  - Glitch suppression prevents the USCI from being accidentally started;
  - Any pulse on UCAxRXD shorter than the deglitch time (approximately 150 ns) will be ignored.
  - **Framing error UCFE**: Set if the stop bit is missing from a received frame;
  - **Parity error UCPE**: Set if there is a parity mismatch in a received frame;
  - **Receive overrun error UCOE**: Set if UCAxRXBUF is overwritten;
  - **Break condition UCBRK**:
    - Set if all bits in the received frame = 0;
    - Set the UCAxRXIFG if UCBRKIE bit is set.

USCI operation: UART mode (13/17)

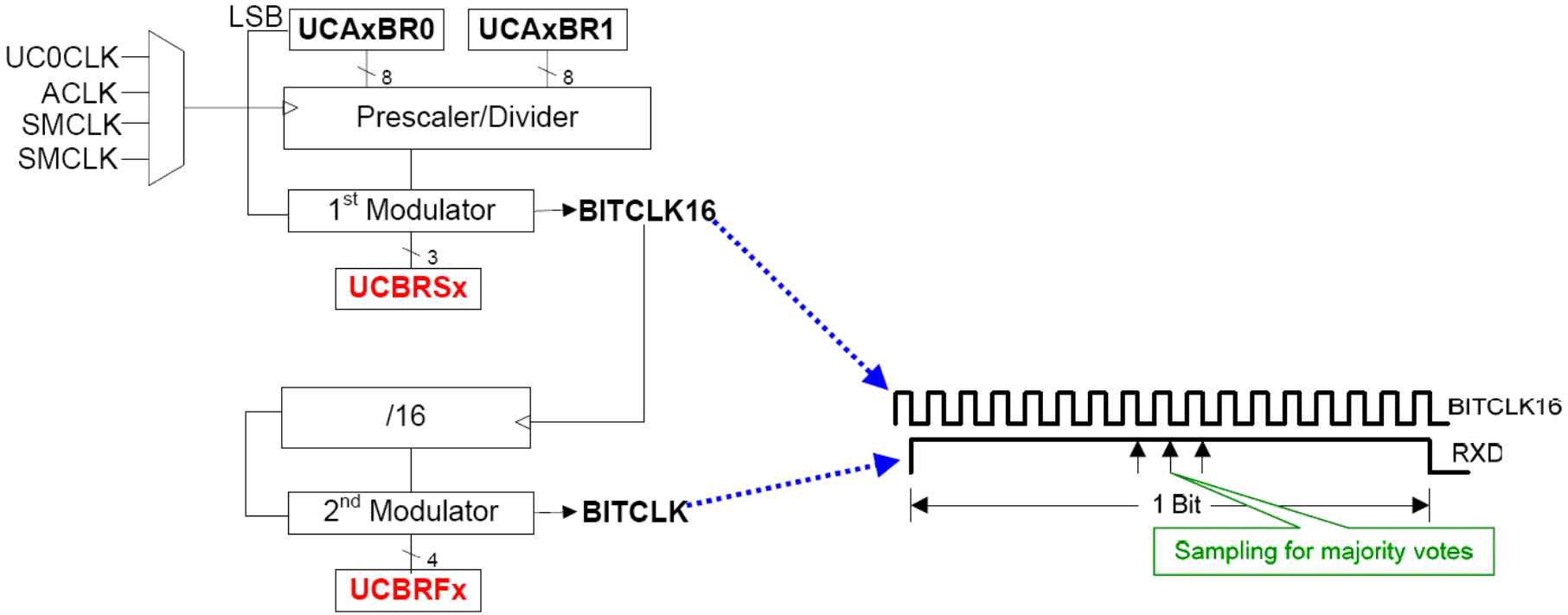
- **Enable the USCI receive enable bit URXEx:**
  - Clear UCSWRST;
  - The falling edge of the start bit enables the baud rate generator;
  - If a valid start bit is detected, a character will be received.

- **USCI transmit enable:**
  - Clear UCSWRST;
  - Transmission is initiated by writing data to UCAxTXBUF;
  - The baud rate generator is enabled;
  - The data value in UCAxTXBUF is moved to the transmit shift register on the next BITCLK after the transmit shift register is empty;
  - UCAxTXIFG is set when a new data value can be written into UCAxTXBUF.

USCI operation: UART mode (14/17)

❑ USCI baud rate generation:

- Standard baud rates from non-standard source frequencies;
- Two modes of operation (UCOS16 bit):
  - Low-frequency baud rate;
  - Oversampling baud rate.



USCI operation: UART mode (15/17)

- **Transmit bit timing:**
  - The timing for each character is the sum of the individual bit timings;
  - A modulation feature of the baud rate generator reduces the cumulative bit error.

- **Two error sources for receive bit timing:**
  - Bit-to-bit timing error;
  - Error between a start edge occurring and the start edge being accepted by the USCI module.

USCI operation: UART mode (16/17)

USCI interrupts:

- One interrupt vector for transmission and one interrupt vector for reception:

- **USCI transmit interrupt operation:**
  - UCAxTXIFG interrupt flag is set by the transmitter to indicate that UCAxTXBUF is ready to accept another character;

  - An interrupt request is generated if UCAxTXIE and GIE are also set;

  - UCAxTXIFG is automatically reset if a character is written to UCAxTXBUF.

USCI operation: UART mode (17/17)

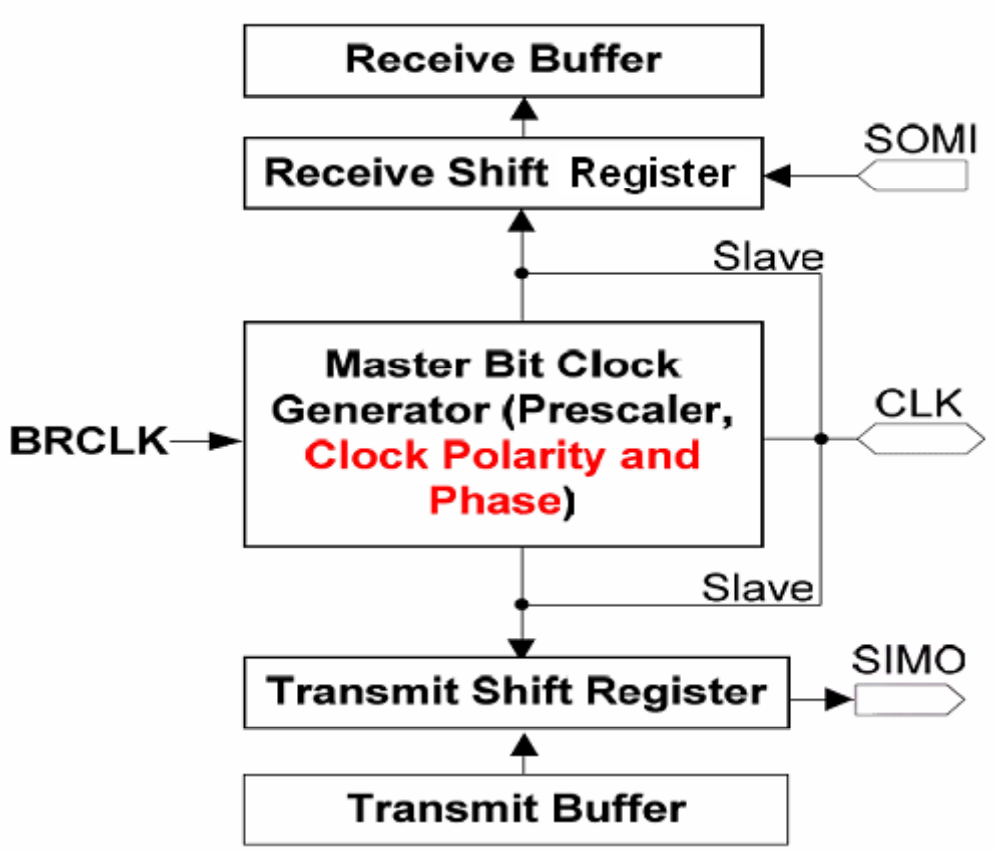
- **USCI interrupts (continued):**
  - **USCI receive interrupt operation:**
    - UCAxRXIFG interrupt flag is set each time a character is received and loaded into UCAxRXBUF;
    - An interrupt request is also generated if UCAxRXIE and GIE are set;
    - UCAxRXIFG and UCAxRXIE are reset by a system reset PUC signal or when UCSWRST = 1;
    - UCAxRXIFG is automatically reset when UCAxRXBUF is read.

USCI operation: SPI mode (1/9)

- **Flexible interface:**
  - 3- or 4-pin SPI;
  - 7- or 8-bit data length;
  - Master or slave;
  - LSB or MSB first.
- **S/W configurable clock phase and polarity;**
- **Programmable SPI master clock;**
- **Double buffered TX/RX;**
- **Interrupt driven TX/RX (USCI_A and USCI_B share TX and RX vector);**
- **Direct Memory Address (DMA) enabled;**
- **LPMx operation.**

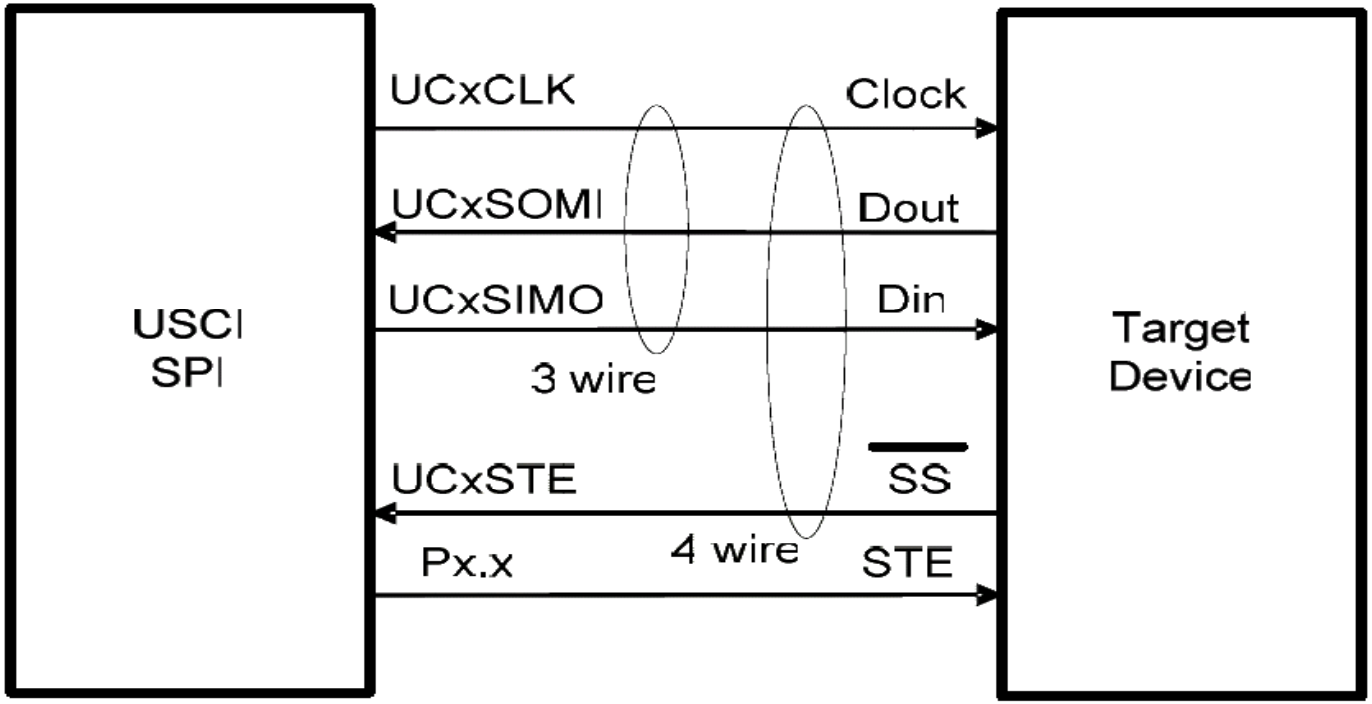
USCI operation: SPI mode (2/9)

❑ USCI module: SPI mode block diagram:



USCI operation: SPI mode (3/9)

❑ USCI module: SPI connections:



USCI operation: SPI mode (4/9)

- **Serial data transmitted and received by multiple devices using a shared clock provided by the master;**

- **Three or four signals are used for SPI data exchange:**
  - UCxSIMO: Slave in, master out;
  - UCxSOMI: Slave out, master in;
  - UCxCLK: USCI SPI clock;
  - UCxSTE: Slave transmit enable:
    - Enables a device to receive and transmit data and is controlled by the master;
    - 4 wire master, senses conflicts with other master(s);
    - In 4 wire slave, externally controls TX and RX.

USCI operation: SPI mode (5/9)

USCI initialization/re-configuration process:

- Set UCSWRST (BIS.B #UCSWRST,&UCAxCTL1);
- Initialize all USCI registers with UCSWRST = 1 (including UCxCTL1);
- Configure ports;
- Clear UCSWRST via software (BIC.B #UCSWRST,&UCxCTL1);
- Enable interrupts (optional) via UCxRXIE and/or UCxTXIE.

USCI operation: SPI mode (6/9)

- **Define the character format as presented earlier;**
- **Define mode: Master or Slave;**
- **Enable SPI transmit/receive clearing the UCSWRST bit;**
- **Define serial clock control:**
  - UCxCLK is provided by the master on the SPI bus;
  - Configure serial clock polarity and phase (UCCKPL and UCCKPH bits).

USCI operation: SPI mode (7/9)

❑ USCI interrupts:

- One interrupt vector for transmission and one interrupt vector for reception:

- **SPI transmit interrupt operation:**
  - UCxTXIFG interrupt flag is set by the transmitter to indicate that UCxTXBUF is ready to accept another character;

  - An interrupt request is generated if UCxTXIE and GIE are also set;

  - UCxTXIFG is automatically reset if the interrupt request is serviced or if a character is written to UCxTXBUF.

USCI operation: SPI mode (8/9)

❑ USCI interrupts (continued):

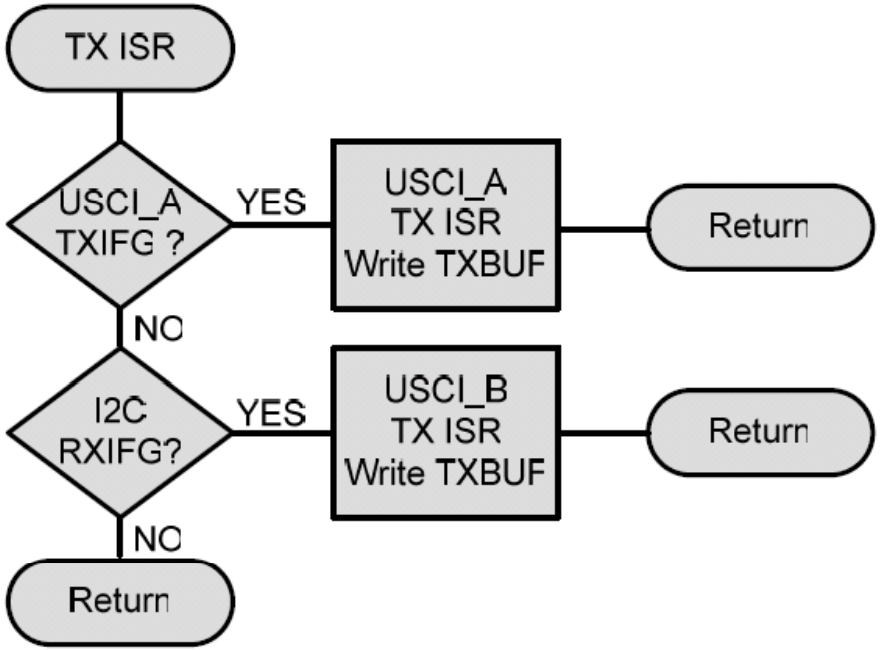
▪ USCI receive interrupt operation:

- UCxRXIFG interrupt flag is set each time a character is received and loaded into UCxRXBUF;
- An interrupt request is also generated if UCxRXIE and GIE are set;
- UCxRXIFG and UCxRXIE are reset by a system reset PUC signal or when SWRST = 1;
- UCxRXIFG is automatically reset if the pending interrupt is serviced (when UCSWRST = 1) or when UCxRXBUF is read.

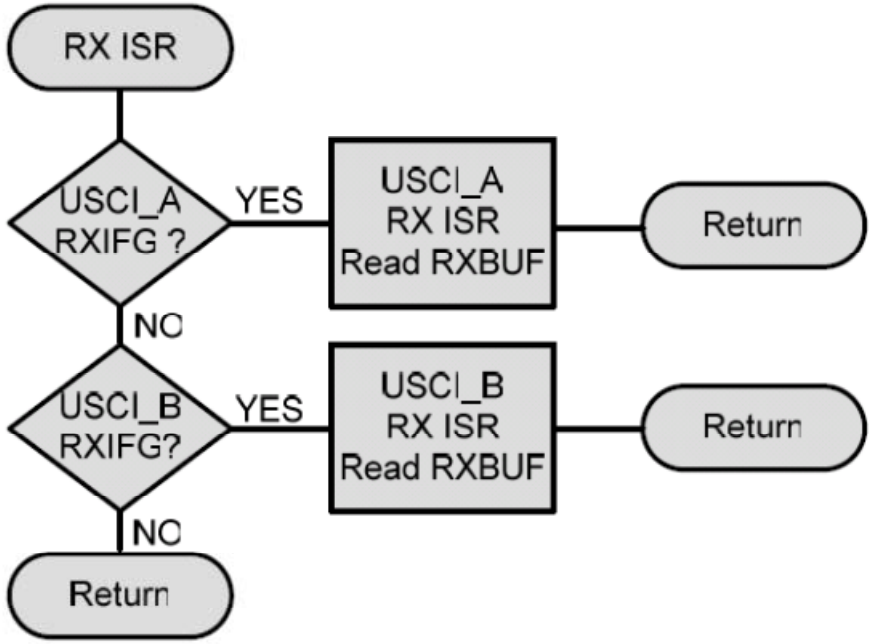
USCI operation: SPI mode (9/9)

❑ USCI interrupts (continued):

SPI TX interrupt:



SPI RX interrupt:

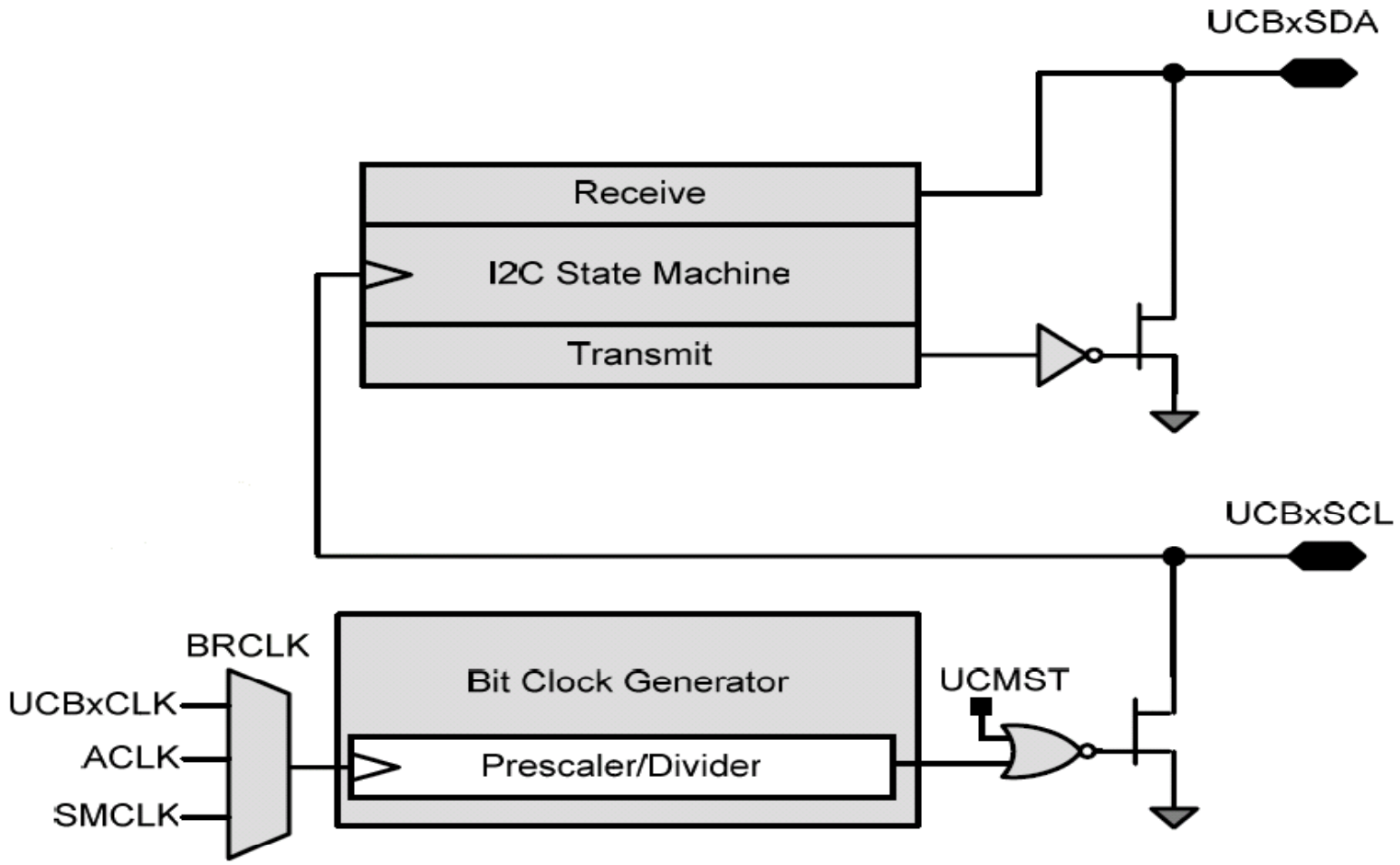


USCI operation: I²C mode (1/11)

- The I²C mode supports any master or slave I²C-compatible device (Specification v2.1);
- Each I²C device is recognized by a unique address and can operate as either a transmitter or a receiver, as well as either the master or the slave;
- A master initiates a data transfer and generates the clock signal SCL;
- Any device addressed by a master is considered a slave;
- Communication using the bi-directional serial data (SDA) and serial clock (SCL) pins;

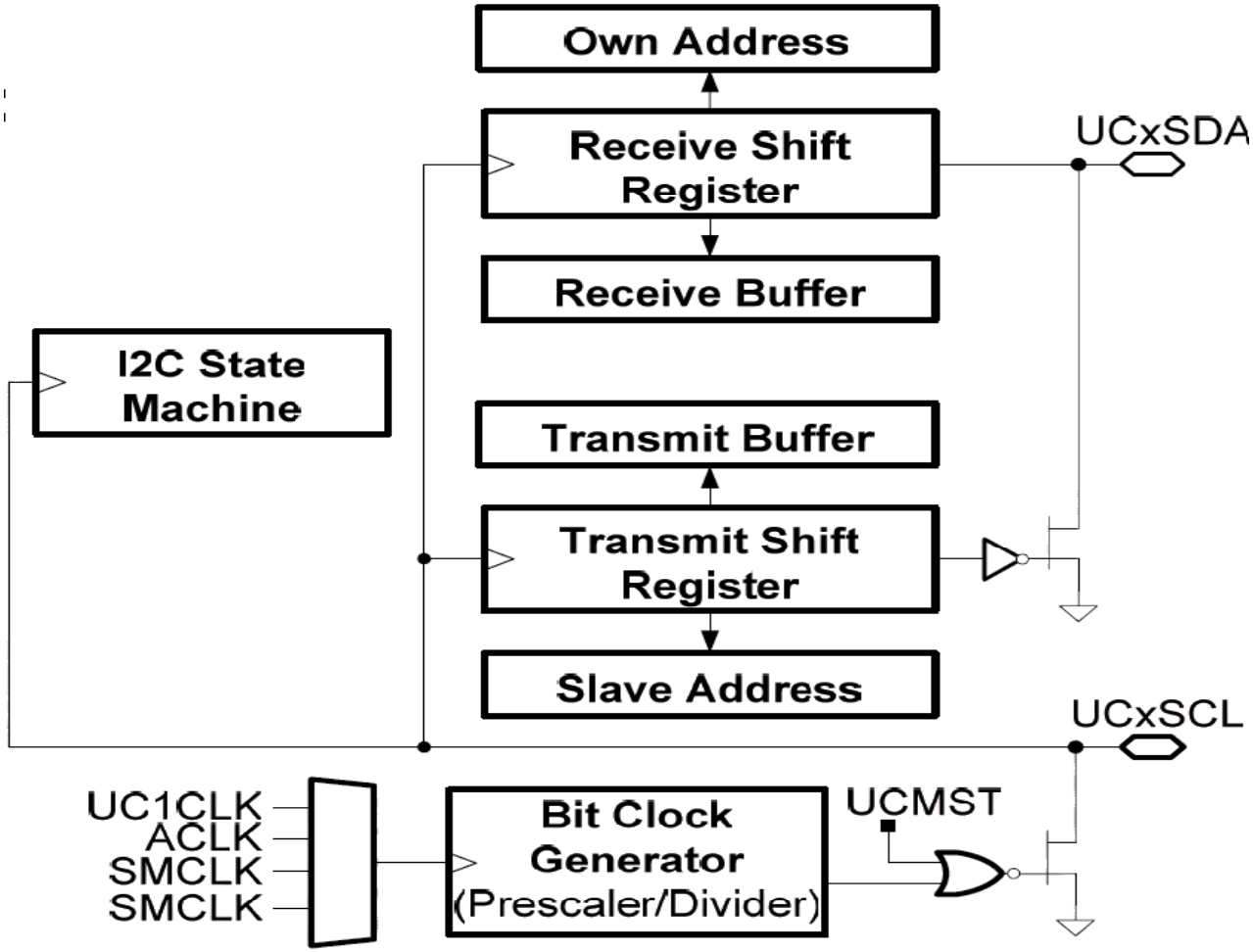
USCI operation: I²C mode (2/11)

I²C mode block diagram:



USCI operation: I²C mode (3/11)

I²C mode block diagram:



USCI operation: I²C mode (4/11)

- **Initialized using the sequence given earlier;**

- **I²C serial data:**
  - One clock pulse is generated by the master for each data bit transferred;
  - Operates with byte data (MSB transferred first);
  - The first byte after a START condition consists of a 7-bit slave address and the R/W bit:
    - R/W = 0: Master transmits data to a slave;
    - R/W = 1: Master receives data from a slave.
  - The ACK bit is sent from the receiver after each byte on the 9th SCL clock.

USCI operation: I²C mode (5/11)

- **I²C addressing modes (7-bit and 10-bit addressing modes);**

- **I²C module operating modes:**
  - Master transmitter;
  - Master receiver;
  - Slave transmitter;
  - Slave receiver.

- **Arbitration procedure is invoked if two or more master transmitters simultaneously start a transmission on the bus;**

USCI operation: I²C mode (6/11)

- **I²C Clock generation and synchronization:**
  - SCL is provided by the master on the I²C bus;
  - Master mode: BITCLK is provided by the USCI bit clock generator;
  - Slave mode: the bit clock generator is not used.

USCI operation: I²C mode (7/11)

I²C interrupts:

- One interrupt vector for transmission and one interrupt vector for reception;
- **I²C transmit interrupt operation:**
  - UCxTXIFG interrupt flag is set by the transmitter to indicate that UCBxTXBUF is ready to accept another character;
  - An interrupt request is also generated if UCxTXIE and GIE are set;
  - UCxTXIFG is automatically reset if a character is written to UCxTXBUF or a NACK is received.

USCI operation: I²C mode (8/11)

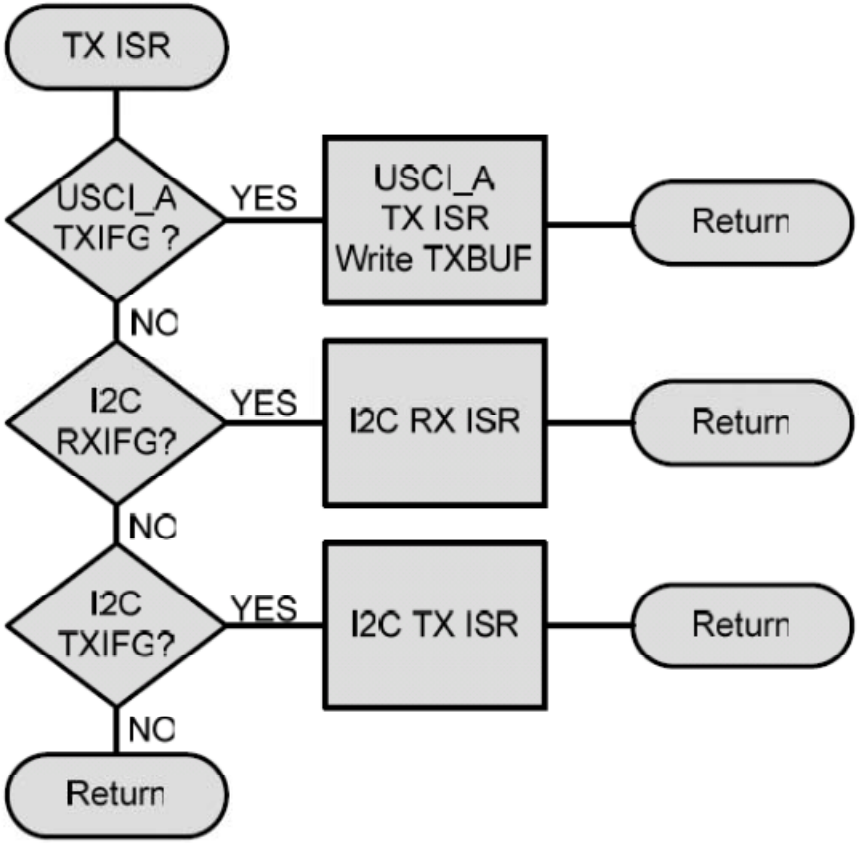
I²C interrupts (continued):

I²C receive interrupt operation:

- UCxRXIFG interrupt flag is set each time a character is received and loaded into UCxRXBUF;
- An interrupt request is also generated if UCxRXIE and GIE are set;
- UCxRXIFG and UCxRXIE are reset by a system reset PUC signal or when SWRST = 1;
- UCxRXIFG is automatically reset when UCxRXBUF is read.

USCI operation: I²C mode (9/11)

- I²C interrupts (continued):
  - I²C transmit/receive interrupt operation:



>> Contents

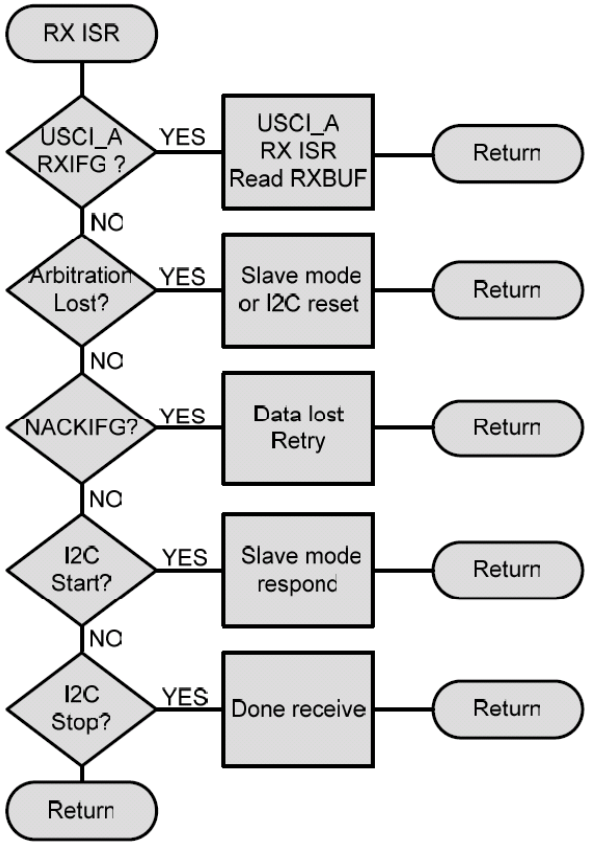
USCI operation: I²C mode (10/11)

- **I²C interrupts (continued):**
  - **I²C state change interrupt flags:**
    - **Arbitration-lost, UCALIFG**: Flag is set when two or more transmitters start a transmission simultaneously, or operates as master but is addressed as a slave by another master;
    - **Not-acknowledge interrupt, UCNACKIFG**: Flag set when an acknowledge is expected but is not received;
    - **Start condition detected interrupt, UCSTTIFG**: Flag set when the I²C module detects a START condition together with its own address while in slave mode;
    - **Stop condition detected interrupt, UCSTPIFG**: Flag set when the I²C module detects a STOP condition while in slave mode.

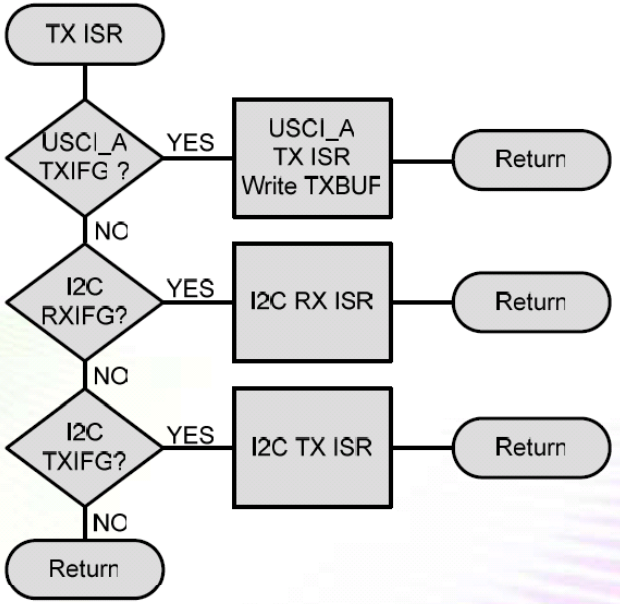
USCI operation: I²C mode (11/11)

- I²C interrupts (continued):

I²C TX interrupt:



I²C RX interrupt:



USCI registers (UART, SPI and I²C modes) (1/20)

- **UCAxCTL0, USCI_Ax Control Register 0 (UART, SPI)**
- **UCBxCTL0, USCI_Bx Control Register 0 (SPI, I²C)**

Mode	7	6	5	4	3	2	1	0
UART	UCPEN	UCPAR	UCMSB	UC7BIT	UCSPB	UCMODEx		UCSYNC=0
SPI	UCCKPH	UCCKPL	UCMSB	UC7BIT	UCMST	UCMODEx		UCSYNC=1
I2C	UCA10	UCSLA10	UCMM	Unused	UCMST	UCMODEx=11		UCSYNC=1

Bit		UART mode description		SPI mode description		I²C mode description
7	UCPEN	Parity enable when UCPEN = 1	UCCKPH	Clock phase select: UCCKPH = 0 ⇒ Data is changed on the 1st UCLK edge and captured on the next one. UCCKPH = 1 ⇒ Data is captured on the 1st UCLK edge and changed on the next one.	UCA10	Own addressing mode select: UCA10= 0 ⇒ 7-bit address UCA10= 1 ⇒ 10-bit address
6	UCPAR	Parity select: UCPAR = 0 ⇒ Odd parity UCPAR = 1 ⇒ Even parity	UCCKPL	Clock polarity select. UCCKPL = 0 ⇒ Inactive state: low. UCCKPL = 1 ⇒ Inactive state: high.	UCSLA10	Slave addressing mode select: UCSLA10= 0 ⇒ 7-bit address UCSLA10= 1 ⇒ 10-bit address
5	UCMSB	MSB first select: UCMSB = 0 ⇒ LSB first UCMSB = 1 ⇒ MSB first	UCMSB	As UART mode	UCMM	Multi-master environment select: UCMM= 0 ⇒ Single master UCMM= 1 ⇒ Multi master

USCI registers (UART, SPI and I²C modes) (2/20)

❑ UCAxCTL0, USCI_Ax Control Register 0 (UART, SPI)

❑ UCBxCTL0, USCI_Bx Control Register 0 (SPI, I²C)

Mode	7	6	5	4	3	2	1	0
UART	UCPEN	UCPAR	UCMSB	UC7BIT	UCSPB	UCMODEx		UCSYNC=0
SPI	UCCKPH	UCCKPL	UCMSB	UC7BIT	UCMST	UCMODEx		UCSYNC=1
I²C	UCA10	UCSLA10	UCMM	Unused	UCMST	UCMODEx=11		UCSYNC=1

Bit		UART mode description	SPI mode description		I²C mode description	
4	UC7BIT	Character length: = 0 ⇒ 8-bit data = 1 ⇒ 7-bit data	UC7BIT	As UART mode	Unused	
3	UCSPB	Stop bit select: = 0 ⇒ One stop bit = 1 ⇒ Two stop bits	UCMST	Master mode: = 0 ⇒ USART is slave = 1 ⇒ USART is master	UCMST	Master mode select. = 0 ⇒ Slave mode = 1 ⇒ Master mode
2-1	UCMODEx	USCI asynchronous mode: = 00 ⇒ UART = 01 ⇒ Idle-Line Multiproc. = 10 ⇒ Address-Bit Multiproc. = 11 ⇒ UART with ABR.	UCMODEx	USCI synchronous mode: = 00 ⇒ 3-Pin SPI = 01 ⇒ 4-Pin SPI (slave enabled when UCxSTE=1) = 10 ⇒ 4-Pin SPI (slave enabled when UCxSTE=0) = 11 ⇒ I²C	UCMODEx=11	USCI Mode: = 00 ⇒ 3-Pin SPI = 01 ⇒ 4-Pin SPI (master/slave enabled if STE = 1) = 10 ⇒ 4-Pin SPI (master/slave enabled if STE = 0) = 11 ⇒ I²C
0	UCSYNC=0	Synchronous mode enable: = 0 ⇒ Asynchronous = 1 ⇒ Synchronous	UCSYNC=1	As UART mode	UCSYNC=1	As UART mode

USCI registers (UART, SPI and I²C modes) (3/20)

- UCAxCTL1, USCI_Ax Control Register 1 (UART, SPI)
- UCBxCTL1, USCI_Bx Control Register 1 (SPI, I²C)

Mode	7	6	5	4	3	2	1	0
UART	UCSSELx		UCRXEIE	UCBRKIE	UCDORM	UCTXADDR	UCTXBRK	UCSWRST
SPI	UCSSELx		Unused	Unused	Unused	Unused	Unused	UCSWRST
I²C	UCSSELx		Unused	UCTR	UCTXNACK	UCTXSTP	UCTXSTT	UCSWRST

Bit		UART mode description		SPI mode description		I²C mode description
7-6	UCSSELx	BRCLK source clock: = 00 ⇒ UCLK = 01 ⇒ ACLK = 10 ⇒ SMCLK = 11 ⇒ SMCLK	UCSSELx	BRCLK source clock: = 00 ⇒ N/A = 01 ⇒ ACLK = 10 ⇒ SMCLK = 11 ⇒ SMCLK	UCSSELx	BRCLK source clock: = 00 ⇒ UCLKI = 01 ⇒ ACLK = 10 ⇒ SMCLK = 11 ⇒ SMCLK
5	UCRXEIE	Receive erroneous-character IE: = 0 ⇒ Rejected (UCAxRXIFG not set) = 1 ⇒ Received (UCAxRXIFG set)	Unused		Unused	Slave addressing mode select: UCSLA10= 0 ⇒ 7-bit address UCSLA10= 1 ⇒ 10-bit address
4	UCBRKIE	Receive break character IE: = 0 ⇒ Not set UCAxRXIFG. = 1 ⇒ Set UCAxRXIFG.	Unused		UCTR	Transmitter/Receiver select: = 0 ⇒ Receiver = 1 ⇒ Transmitter

USCI registers (UART, SPI and I²C modes) (4/20)

❑ UCAxCTL1, USCI_Ax Control Register 1 (UART, SPI)

❑ UCBxCTL1, USCI_Bx Control Register 1 (SPI, I²C)

Mode	7	6	5	4	3	2	1	0
UART	UCSSELx		UCRXEIE	UCBRKIE	UCDORM	UCTXADDR	UCTXBRK	UCSWRST
SPI	UCSSELx		Unused	Unused	Unused	Unused	Unused	UCSWRST
I²C	UCSSELx		Unused	UCTR	UCTXNACK	UCTXSTP	UCTXSTT	UCSWRST

Bit		UART mode description	SPI mode description		I²C mode description	
3	UCDORM	Dormant. Puts USCI into sleep mode: = 0 ⇒ Not dormant = 1 ⇒ Dormant	Unused		UCTXNACK	Transmit a NACK: = 0 ⇒ Acknowledge normally = 1 ⇒ Generate NACK
2	UCTXADDR	Transmit address: = 0 ⇒ Next frame transmitted is data = 1 ⇒ Next frame transmitted is address	Unused		UCTXSTP	Transmit STOP condition in master mode: = 0 ⇒ No STOP generated = 1 ⇒ Generate STOP
1	UCTXBRK	Transmit break: = 0 ⇒ Next frame transmitted is not a break = 1 ⇒ Next frame transmitted is a break or a break/synch	Unused		UCTXSTT	Transmit START condition in master mode: = 0 ⇒ No START generated = 1 ⇒ Generate START
0	UCSWRST	Software reset enable =0 ⇒ Disabled. USCI reset released for operation 1 ⇒ Enabled. USCI logic held in reset state	UCSWRST	As UART mode	UCSWRST	As UART mode

USCI registers (UART, SPI and I²C modes) (5/20)

- UCAxBR0, USCI_Ax Baud Rate Control Register 0 (UART, SPI)
- UCBxBR0, USCI_Bx Bit Rate Control Register 0 (SPI, I²C)

Mode	7	6	5	4	3	2	1	0
UART / SPI / I²C	UCBRx – low byte							

- UCAxBR1, USCI_Ax Baud Rate Control Register 1 (UART, SPI)
- UCBxBR1, USCI_Bx Bit Rate Control Register 1 (SPI, I²C)

Mode	7	6	5	4	3	2	1	0
UART / SPI / I²C	UCBRx – high byte							

Bit		UART mode description		SPI mode description		I²C mode description	
7-6	UCBRx	Clock prescaler setting of the baud rate generator: Prescaler value (16-bit value) = {UCAxBR0+UCAxBR1x256}	UCBRx	Bit clock prescaler setting: Prescaler value (16-bit value) = {UCAxBR0+UCAxBR1×256}	UCBRx	As SPI mode

USCI registers (UART, SPI and I²C modes) (6/20)

❑ UCAxSTAT, USCI_Ax Status Register (UART, SPI)

❑ UCBxSTAT, USCI_Bx Status Register (SPI, I²C)

Mode	7	6	5	4	3	2	1	0
UART	UCLISTEN	UCFE	UCOE	UCPE	UCBRK	UCRXERR	UCADDR UCIDLE	UCBUSY
SPI	UCLISTEN	UCFE	UCOE	Unused	Unused	Unused	Unused	UCBUSY
I²C	Unused	UCSCLLOW	UCGC	UCBBUSY	UCNACKIFG	UCSTPIFG	UCSTTIFG	UCALIFG

Bit		UART mode description		SPI mode description		I²C mode description	
7	UCLISTEN	Listen enable: = 0 ⇒ Disabled = 1 ⇒ UCAxTXD is internally fed back to receiver	UCLISTEN	Listen enable: = 0 ⇒ Disabled = 1 ⇒ The transmitter output is internally fed back to receiver	Unused	
6	UCFE	Framing error flag: = 0 ⇒ No error = 1 ⇒ Character with low stop bit	UCFE	Framing error flag: = 0 ⇒ No error = 1 ⇒ Bus conflict (4w master)	UCSCLLOW	SCL low: = 0 ⇒ SCL is not held low = 1 ⇒ SCL is held low
5	UCOE	Overrun error flag: = 0 ⇒ No error = 1 ⇒ Overrun error	UCOE	As UART mode	UCGC	General call address received: = 0 ⇒ No general call address = 1 ⇒ General call address

USCI registers (UART, SPI and I²C modes) (7/20)

❑ UCAxSTAT, USCI_Ax Status Register (UART, SPI)

❑ UCBxSTAT, USCI_Bx Status Register (SPI, I²C)

Mode	7	6	5	4	3	2	1	0
UART	UCLISTEN	UCFE	UCOE	UCPE	UCBRK	UCRXERR	UCADDR UCIDLE	UCBUSY
SPI	UCLISTEN	UCFE	UCOE	Unused	Unused	Unused	Unused	UCBUSY
I²C	Unused	UCSCLLOW	UCGC	UCBBUSY	UCNACKIFG	UCSTPIFG	UCSTTIFG	UCALIFG

Bit		UART mode description	SPI mode description		I²C mode description
4	UCPE	Parity error flag: = 0 ⇒ No error = 1 ⇒ Character with parity error	Unused	UCBBUSY	Bus busy: = 0 ⇒ Bus inactive = 1 ⇒ Bus busy
3	UCBRK	Break detect flag: = 0 ⇒ No break condition = 1 ⇒ Break condition occurred	Unused	UCNACKIFG	NACK received interrupt flag: = 0 ⇒ No interrupt pending = 1 ⇒ Interrupt pending
2	UCRXERR	Receive error flag. = 0 ⇒ No receive errors detected = 1 ⇒ Receive error detected	Unused	UCSTPIFG	Stop condition interrupt flag: = 0 ⇒ No interrupt pending = 1 ⇒ Interrupt pending
1	UCADDR UCIDLE	Address-bit multiproc. mode: = 0 ⇒ Received character is data = 1 ⇒ Received character is an address Idle-line multiproc. mode: = 0 ⇒ No idle line detected = 1 ⇒ Idle line detected	Unused	UCSTTIFG	Start condition interrupt flag: = 0 ⇒ No interrupt pending = 1 ⇒ Interrupt pending
0	UCBUSY	USCI busy: = 0 ⇒ USCI inactive = 1 ⇒ USCI transmit/receive	UCBUSY	UCALIFG	Arbitration lost interrupt flag: = 0 ⇒ No interrupt pending = 1 ⇒ Interrupt pending

>> Contents

USCI registers (UART, SPI and I²C modes) (8/20)

- UCAxRXBUF, USCI_Ax Receive Buffer Register (UART, SPI)
- UCBxRXBUF, USCI_Bx Receive Buffer Register (SPI, I²C)

Mode	7	6	5	4	3	2	1	0
UART / SPI / I²C	UCRXBUFx							

Bit		UART mode description			SPI mode description		I²C mode description	
7-0	UCRXBUFx	The receive-data buffer is user accessible and contains the last received character from the receive shift register. Reading UCxRXBUF resets receive-error bits, UCADDR/UCIDLE bit and UCAxRXIFG. In 7-bit data mode, UCAxRXBUF is LSB justified and the MSB is always cleared.	UCRXBUFx	As UART mode Reading UCxRXBUF resets the receive-error bits, and UCxRXIFG	UCRXBUFx	As SPI mode

USCI registers (UART, SPI and I²C modes) (9/20)

- UCAxTXBUF, USCI_Ax Transmit Buffer Register (UART, SPI)
- UCBxTXBUF, USCI_Bx Transmit Buffer Register (SPI, I²C)

Mode	7	6	5	4	3	2	1	0
UART / SPI / I²C	UCTXBUFx							

Bit		UART mode description		SPI mode description		I²C mode description	
7-0	UCTXBUFx	The transmit data buffer is user accessible and holds the data waiting to be moved into the transmit shift register and transmitted on UCAxTXD. Writing to the transmit data buffer clears UCAxTXIFG.	UCTXBUFx	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 UCxTXIFG.	UCTXBUFx	As SPI mode	

USCI registers (UART, SPI and I²C modes) (10/20)

❑ IE2, Interrupt Enable Register 2 (UART, SPI, I²C)

Mode	7	6	5	4	3	2	1	0
UART							UCA0TXIE	UCA0RXIE
SPI					UCB0TXIE	UCB0RXIE	UCA0TXIE	UCA0RXIE
I²C					UCB0TXIE	UCB0RXIE		

Bit		UART mode description		SPI mode description		I²C mode description	
3			UCB0TXIE	USCI_B0 transmit interrupt enable: = 0 ⇒ Disabled = 1 ⇒ Enabled	UCB0TXIE	As SPI mode
2			UCB0RXIE	USCI_B0 receive interrupt enable: = 0 ⇒ Disabled = 1 ⇒ Enabled	UCB0RXIE	As SPI mode
1	UCA0TXIE	USCI_A0 transmit interrupt enable: = 0 ⇒ Disabled = 1 ⇒ Enabled	UCA0TXIE	As UART mode		
0	UCA0RXIE	USCI_A0 receive interrupt enable: = 0 ⇒ Disabled = 1 ⇒ Enabled	UCA0RXIE	As UART mode		

USCI registers (UART, SPI and I²C modes) (11/20)

❑ IFG2, Interrupt Flag Register 2 (UART, SPI, I²C)

Mode	7	6	5	4	3	2	1	0
UART							UCA0TXIFG	UCA0RXIFG
SPI					UCB0TXIFG	UCB0RXIFG	UCA0TXIFG	UCA0RXIFG
I²C					UCB0TXIFG	UCB0RXIFG		

Bit	UART mode description		SPI mode description		I²C mode description	
3			UCB0TXIFG	USCI_B0 transmit interrupt flag: = 0 ⇒ No interrupt pending = 1 ⇒ Interrupt pending	UCB0TXIFG	As SPI mode
2			UCB0RXIFG	USCI_B0 receive interrupt flag: = 0 ⇒ No interrupt pending = 1 ⇒ Interrupt pending	UCB0RXIFG	As SPI mode
1	UCA0TXIFG	USCI_A0 transmit interrupt flag: = 0 ⇒ No interrupt pending = 1 ⇒ Interrupt pending	UCA0TXIFG	As UART mode		
0	UCA0RXIFG	USCI_A0 receive interrupt flag: = 0 ⇒ No interrupt pending = 1 ⇒ Interrupt pending	UCA0RXIFG	As UART mode		

USCI registers (UART, SPI and I²C modes) (12/20)

- UC1IE, USCI_A1 Interrupt Enable Register (UART, SPI)
- UC1IE, USCI_B1 Interrupt Enable Register (SPI, I²C)

Mode	7	6	5	4	3	2	1	0
UART	Unused	Unused	Unused	Unused			UCA1TXIE	UCA1RXIE
SPI	Unused	Unused	Unused	Unused	UCB1TXIE	UCB1RXIE	UCA1TXIE	UCA1RXIE
I²C	Unused	Unused	Unused	Unused	UCB1TXIE	UCB1RXIE		

Bit	UART mode description		SPI mode description		I²C mode description	
3			UCB1TXIE	USCI_B1 transmit interrupt enable: UTXIE1 = 0 ⇒ Disabled UTXIE1 = 1 ⇒ Enabled	UCB1TXIE	As SPI mode
2			UCB1RXIE	USCI_B1 receive interrupt enable: URXIE1 = 0 ⇒ Disabled URXIE1 = 1 ⇒ Enabled	UCB1RXIE	As SPI mode
1	UCA1TXIE	USCI_A1 transmit interrupt enable: UTXIE1 = 0 ⇒ Disabled UTXIE1 = 1 ⇒ Enabled	UCA1TXIE	As UART mode		
0	UCA1RXIE	USCI_A1 receive interrupt enable: URXIE1 = 0 ⇒ Disabled URXIE1 = 1 ⇒ Enabled	UCA1RXIE	As UART mode		

USCI registers (UART, SPI and I²C modes) (13/20)

- UC1IFG, USCI_A1 Interrupt Flag Register (UART, SPI)
- UC1IFG, USCI_B1 Interrupt Flag Register (SPI, I²C)

Mode	7	6	5	4	3	2	1	0
UART							UCA1TXIFG	UCA1RXIFG
SPI					UCB1TXIFG	UCB1RXIFG	UCA1TXIFG	UCA1RXIFG
I²C					UCB1TXIFG	UCB1RXIFG		

Bit	UART mode description		SPI mode description		I²C mode description	
3			UCB1TXIFG	USCI_B1 transmit interrupt flag: = 0 ⇒ No interrupt pending = 1 ⇒ Interrupt pending	UCB1TXIFG	As SPI mode
2			UCB1RXIFG	USCI_B1 receive interrupt flag: = 0 ⇒ No interrupt pending = 1 ⇒ Interrupt pending	UCB1RXIFG	As SPI mode
1	UCA1TXIFG	USCI_A1 transmit interrupt flag: = 0 ⇒ No interrupt pending = 1 ⇒ Interrupt pending	UCA1TXIFG	As UART mode		
0	UCA1RXIFG	USCI_A1 receive interrupt flag: = 0 ⇒ No interrupt pending = 1 ⇒ Interrupt pending	UCA1RXIFG	As UART mode		

USCI registers (UART, SPI and I²C modes) (14/20)

❑ UCAxMCTL, USCI_Ax Modulation Control Register (UART)

7	6	5	4	3	2	1	0
UCBRFx				UCBRSx			UCOS16

Bit		UART mode description
7-4	UCBRFx	First modulation pattern for BITCLK16 when UCOS16 = 1 (See *Table 19-3* of the MSP430x4xx User's Guide)
3-1	UCBRSx	Second modulation pattern for BITCLK (See *Table 19-2* of the MSP430x4xx User's Guide)
0	UCOS16	Oversampling mode enabled when UCOS16 = 1

USCI registers (UART, SPI and I²C modes) (15/20)

❑ UCAxIRTCTL, USCI_Ax IrDA Transmit Control Register (UART)

7	6	5	4	3	2	1	0
UCIRTXPLx						UCIRTXCLK	UCIREN

Bit		UART mode description
7-2	UCIRTXPLx	Transmit pulse length: t_{PULSE} = (UCIRTXPLx + 1) / (2 x $f_{IRTXCLK}$)
1	UCIRTXCLK	IrDA transmit pulse clock select: UCIRTXCLK = 0 ⇒ BRCLK UCIRTXCLK = 1 ⇒ BITCLK16, when UCOS16 = 1 ⇒ BRCLK, otherwise
0	UCIREN	IrDA encoder/decoder enable: UCIREN = 0 ⇒ IrDA encoder/decoder disabled UCIREN = 1 ⇒ IrDA encoder/decoder enabled

USCI registers (UART, SPI and I²C modes) (16/20)

❑ UCAxIRRCTL, USCI_Ax IrDA Receive Control Register (UART)

7	6	5	4	3	2	1	0
UCIRRXFLx						UCIRRXPL	UCIRRXFE

Bit		UART mode description
7-2	UCIRRXFLx	Receive filter length (minimum pulse length): $t_{MIN} = (UCIRRXFLx + 4) / (2 \times f_{IRTXCLK})$
1	UCIRRXPL	IrDA receive input UCAxRXD polarity. When a light pulse is seen: UCIRRXPL = 0 ⇒ IrDA transceiver delivers a high pulse UCIRRXPL = 1 ⇒ IrDA transceiver delivers a low pulse
0	UCIRRXFE	IrDA receive filter enabled: UCIRRXFE = 0 ⇒ Disabled UCIRRXFE = 1 ⇒ Enabled

USCI registers (UART, SPI and I²C modes) (17/20)

❑ UCAxABCTL, USCI_Ax Auto Baud Rate Control Register (UART)

7	6	5	4	3	2	1	0
Reserved		UCDELIMx		UCSTOE	UCBTOE	Reserved	UCABDEN

Bit		UART mode description
5-4	UCDELIMx	Break/synch delimiter length: UCDELIM1 UCDELIM0 = 00 ⇒ 1 bit time UCDELIM1 UCDELIM0 = 01 ⇒ 2 bit times UCDELIM1 UCDELIM0 = 10 ⇒ 3 bit times UCDELIM1 UCDELIM0 = 11 ⇒ 4 bit times
3	UCSTOE	Synch field time out error: UCSTOE = 0 ⇒ No error UCSTOE = 1 ⇒ Length of synch field exceeded measurable time
2	UCBTOE	Break time out error: UCBTOE = 0 ⇒ No error UCBTOE = 1 ⇒ Length of break field exceeded 22 bit times.
0	UCABDEN	Automatic baud rate detect enable: UCABDEN = 0 ⇒ Baud rate detection disabled UCABDEN = 1 ⇒ Baud rate detection enabled

USCI registers (UART, SPI and I²C modes) (18/20)

❑ UCBxI2COA, USCIBx I2C Own Address Register (I²C)

15	14	13	12	11	10	9	8
UCGCEN	0	0	0	0	0	I2COAx	

7	6	5	4	3	2	1	0
I2COAx							

Bit		UART mode description
15	UCGCEN	General call response enable: UCGCEN = 0 ⇒ Do not respond to a general call UCGCEN = 1 ⇒ Respond to a general call
9-0	I2COAx	I²C own address (local address of the USCI_Bx I²C controller) ⇒ Right-justified address ⇒ 7-bit address ⇒ Bit 6 is the MSB, Bits 9-7 are ignored. ⇒ 10-bit address ⇒ Bit 9 is the MSB.

USCI registers (UART, SPI and I²C modes) (19/20)

❑ UCBxI2CSA, USCI_Bx I²C Slave Address Register (I²C)

15	14	13	12	11	10	9	8
0	0	0	0	0	0	I2CSAx	

7	6	5	4	3	2	1	0
I2CSAx							

Bit		UART mode description
9-0	I2CSAx	I²C slave address (slave address of the external device to be addressed by the USCI_Bx module) ⇒ Only used in master mode ⇒ Right-justified address ⇒ 7-bit address ⇒ Bit 6 is the MSB, Bits 9-7 are ignored. ⇒ 10-bit address ⇒ Bit 9 is the MSB.

USCI registers (UART, SPI and I²C modes) (20/20)

❑ UCBxI2CIE, USCI_Bx I2C Interrupt Enable Register (I²C)

7	6	5	4	3	2	1	0
Reserved				UCNACKIE	UCSTPIE	UCSTTIE	UCALIE

Bit		UART mode description
3	UCNACKIE	Not-acknowledge interrupt enable: UCNACKIE = 0 ⇒ Interrupt disabled UCNACKIE = 1 ⇒ Interrupt enabled
2	UCSTPIE	Stop condition interrupt enable: UCSTPIE = 0 ⇒ Interrupt disabled UCSTPIE = 1 ⇒ Interrupt enabled
1	UCSTTIE	Start condition interrupt enable: UCSTTIE = 0 ⇒ Interrupt disabled UCSTTIE = 1 ⇒ Interrupt enabled
0	UCALIE	Arbitration lost interrupt enable: UCALIE = 0 ⇒ Interrupt disabled UCALIE = 1 ⇒ Interrupt enabled