Two Wire Interface (TWI)

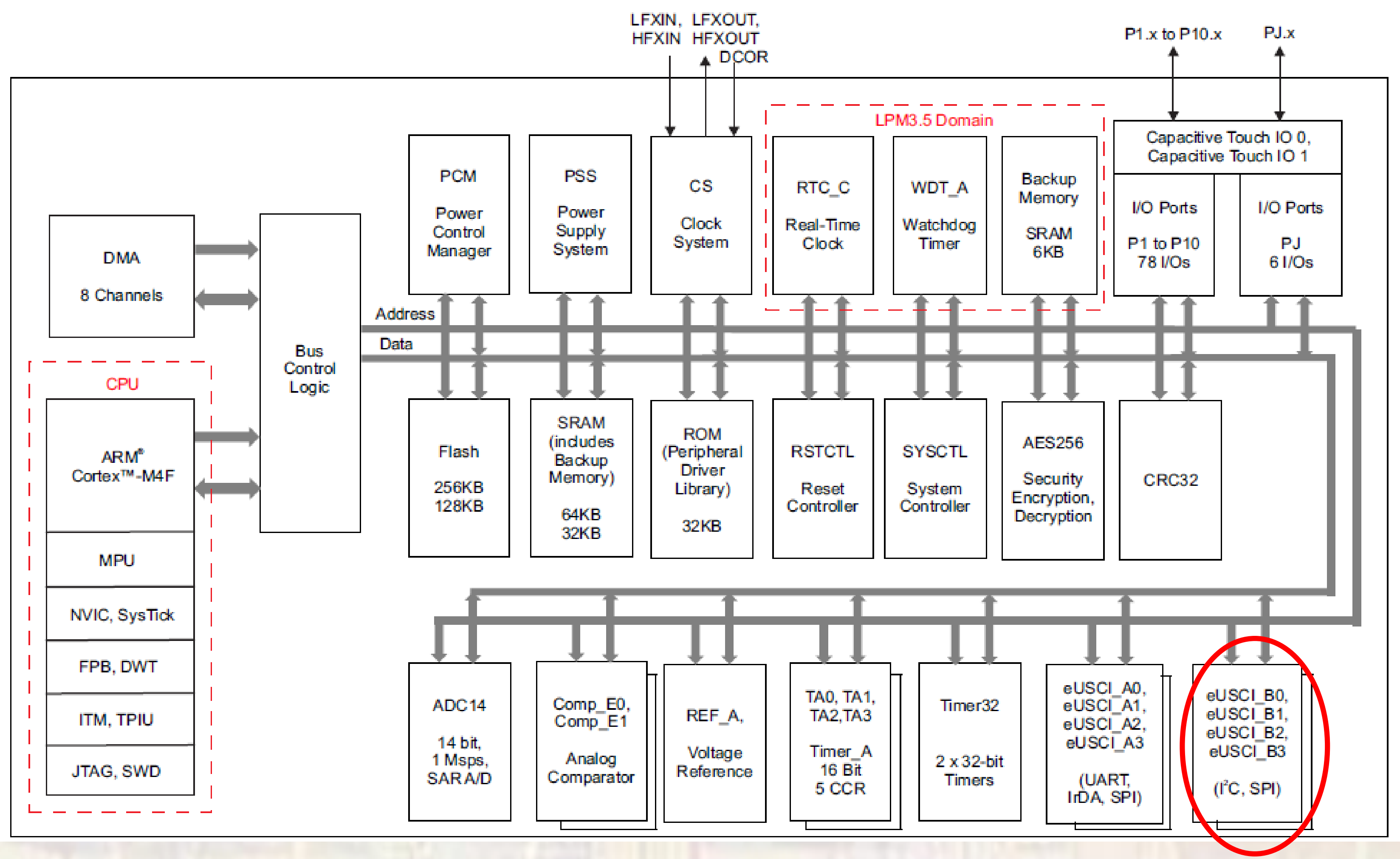
also commonly called

I2C

Last updated 6/17/19

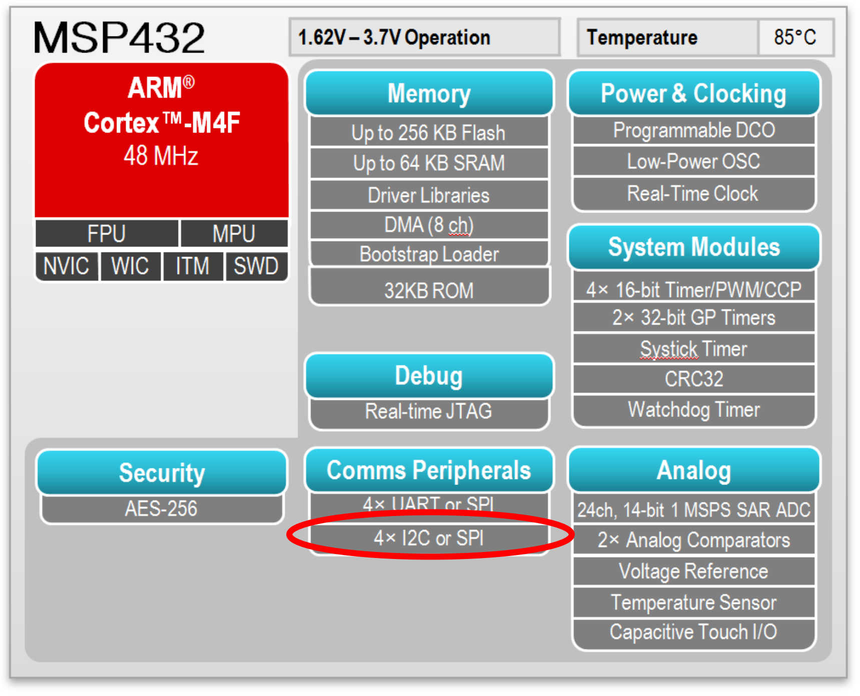
Two Wire Interface

- MSP432 I2C



Two Wire Interface

- MSP432 I2C
  - ARM (AMBA Compliant)
  - 8 bit transmission word
  - 7/10 bit addressing
  - Multi-master/slave modes
  - 4 slave addresses
  - 4 eUSCI-B modules



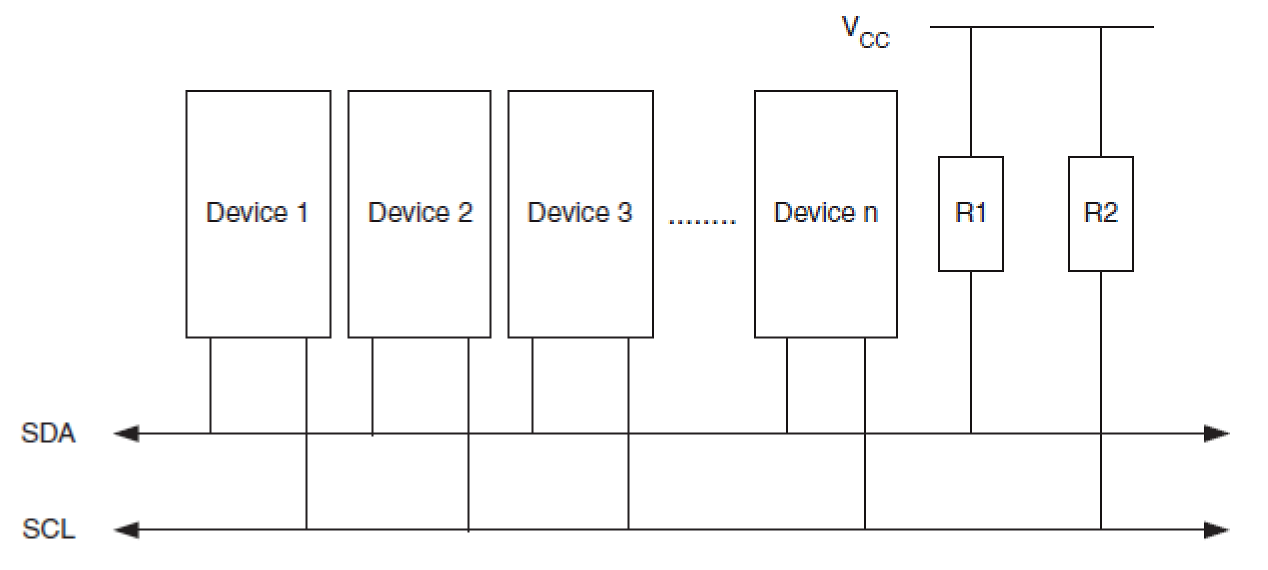
Two Wire Interface

- Overview
  - 8 bit synchronous shift register used to communicate externally
    - 9 bit total communication packet
    - uni-directional
  - Most often used to communicate with peripherals
    - displays, sensors, converters
  - Supports multiple masters and multiple slaves
  - 4 modes of operation
    - Master Receive
    - Master Transmit
    - Slave Receive
    - Slave Transmit

Two Wire Interface

- Overview

  - Open drain configuration
    - outputs only pull down
    - pull up resistors or current sources pull up

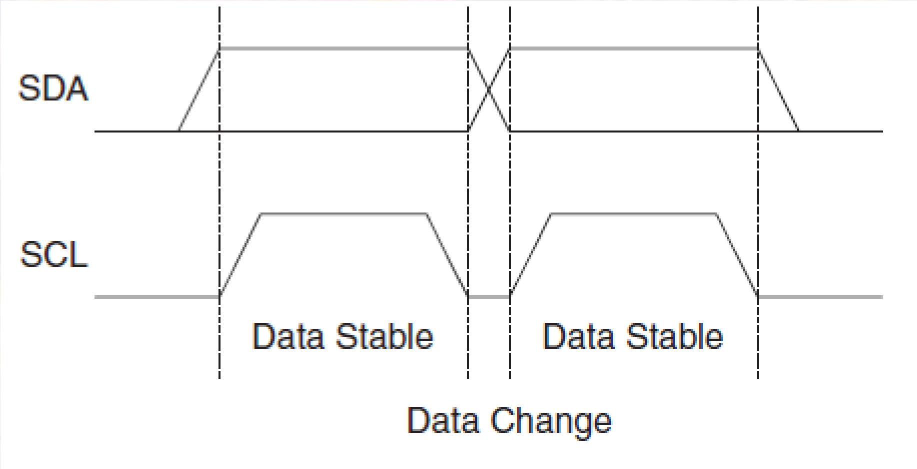


Two Wire Interface

- TWI Timing

  - SDA – data line
  - SCL – clock line

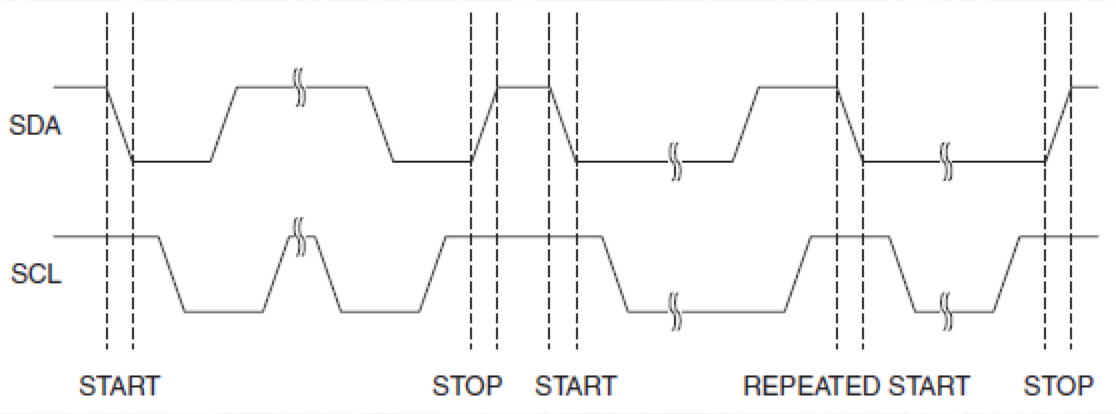
  - Data must be valid during the entire positive clock cycle time



Note: data changes occur during SCL low

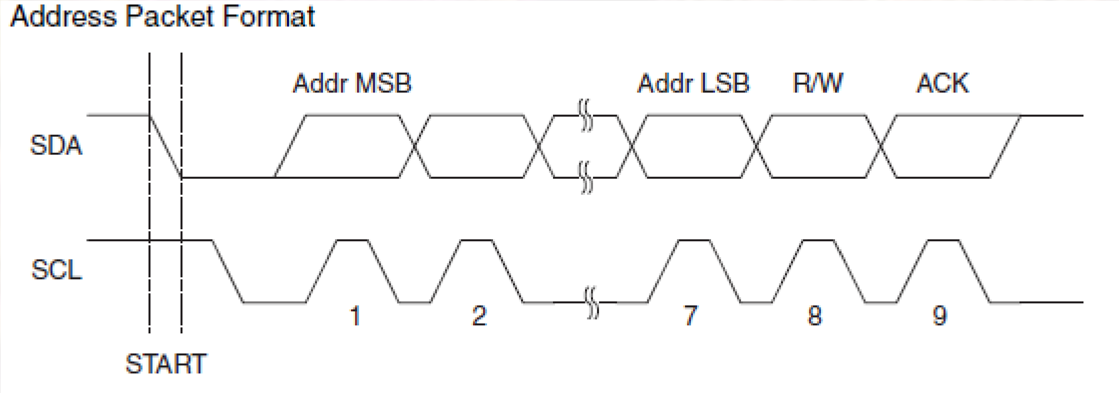
Two Wire Interface

- TWI Timing
  - Special timing requirements for
    - start transmission
    - stop transmission
    - repeated start transition
      - master does not relinquish the bus in this mode



Two Wire Interface

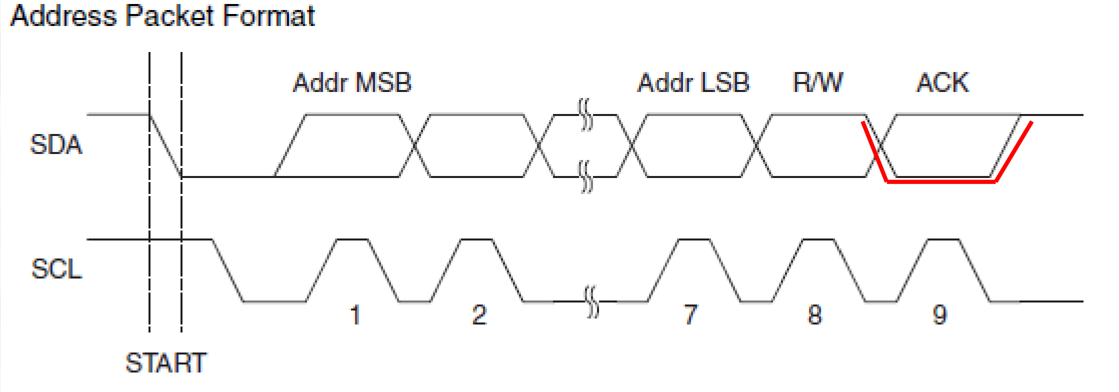
- TWI Timing
  - Addressing
    - Indicate which slave to transmit to or receive from by first transmitting the “address” of the desired device
    - Often this value is hardwired via external pins on the slave device
    - 7 bits for each address



Two Wire Interface

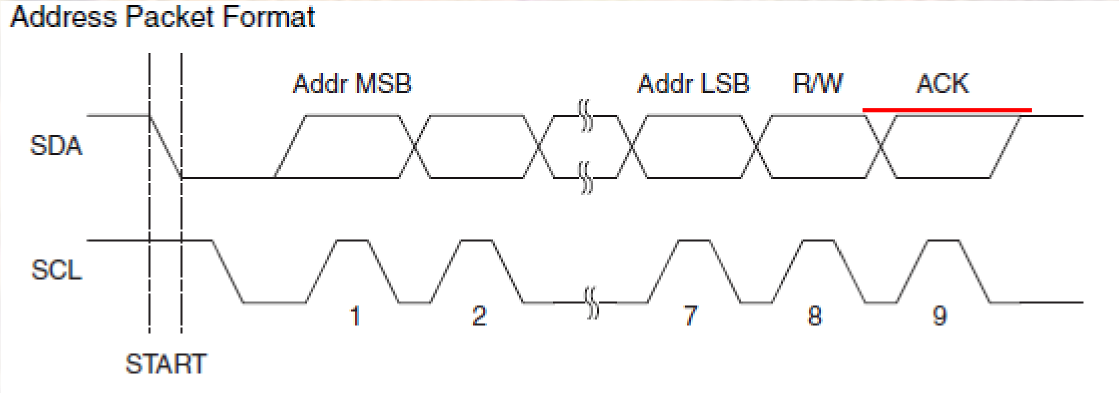
- TWI Timing

  - R/W bit indicates a read or write operation is to follow
    - Read is active high
  - ACK
    - The master drives the data bus from start through the R/W bit and then releases the bus
    - The slave then pulls down the bus in the last clock cycle to indicate a completed transmission



Two Wire Interface

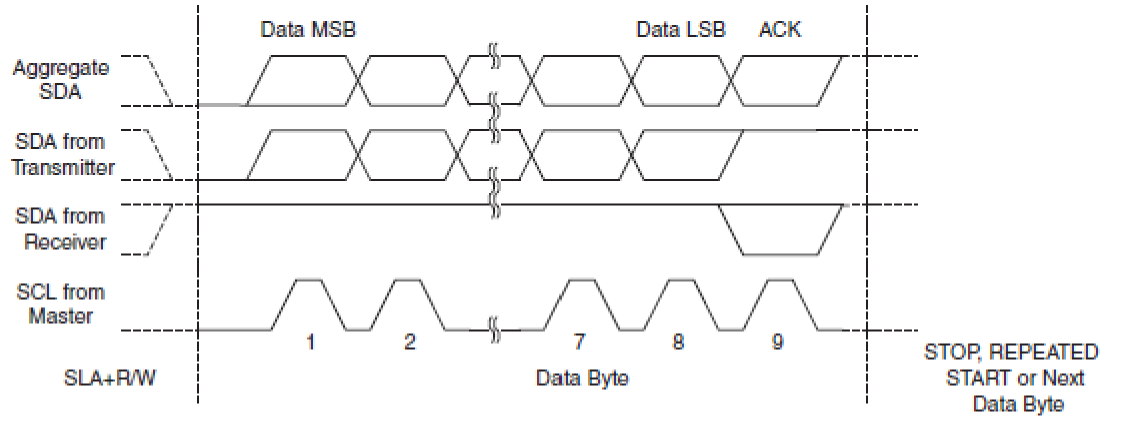
- TWI Timing
  - ACK – cont'd
    - If the master fails to see the slave pull down the bus in the 9th clock cycle (NACK)
      - Transmission failed
      - Some sort of error action is required



Two Wire Interface

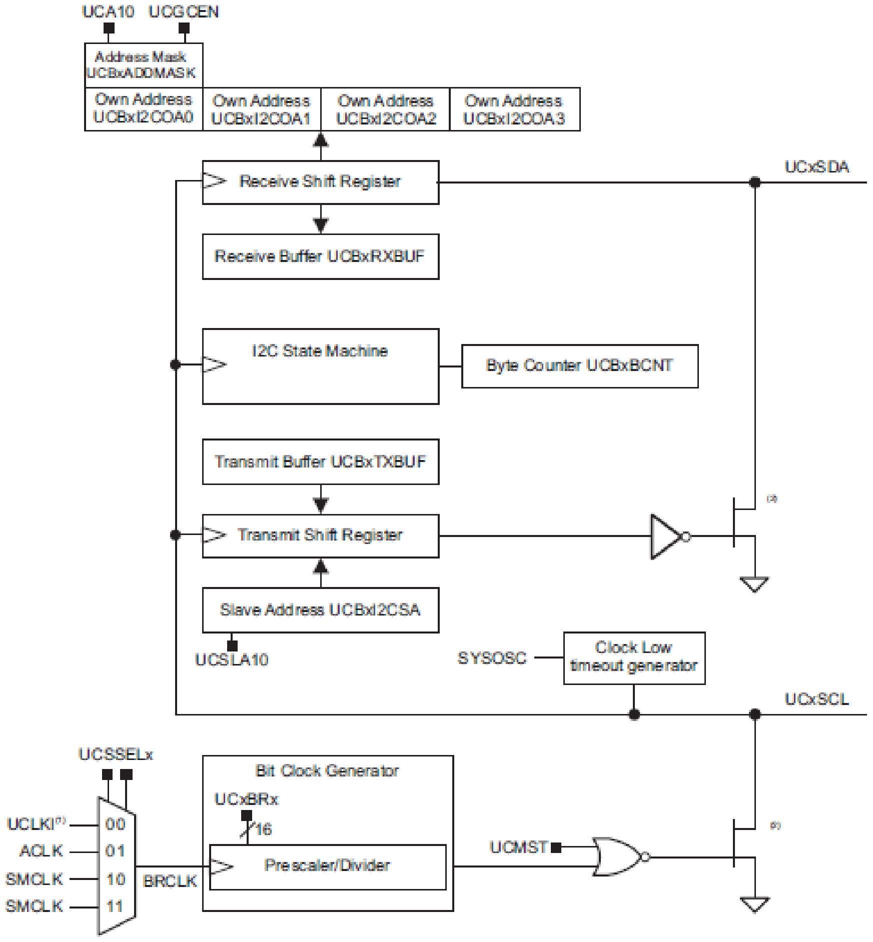
- TWI Timing
  - Data packet
    - After getting an ACK on the address – data can be sent
    - 8 bits of data
    - 1 bit for a data ACK
    - This can be repeated many times

Data Packet Format



Two Wire Interface

- Implementation
  - Separate RX/TX registers
  - Separate RX/TX buffers
  - State Machine
  - Clock divider
  - Control register
  - Status register

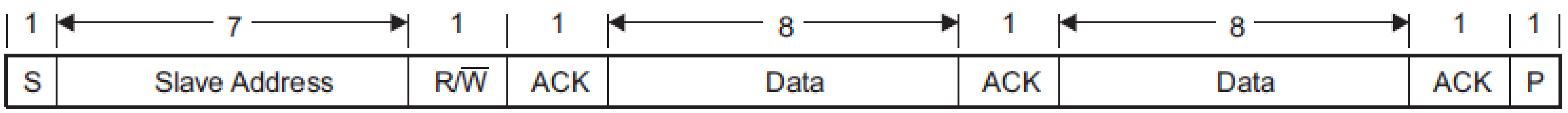


[1] externally provided clock on the eUSCI_B SPI clock input pin
[2] not the real implementation (transistor not located in eUSCI_B module)

Figure 24-1. eUSCI_B Block Diagram – I²C Mode

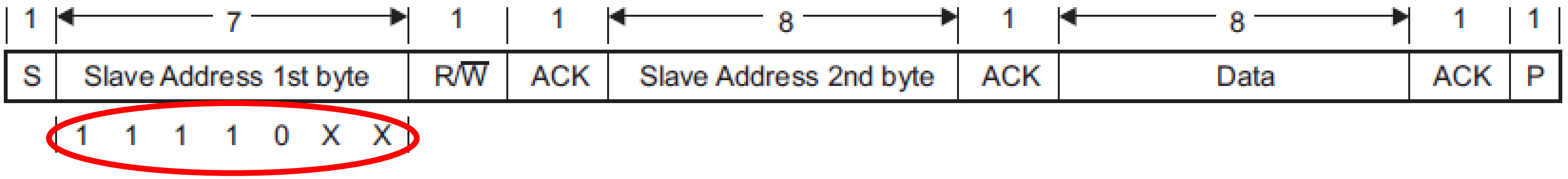
Two Wire Interface

- Addressing Modes



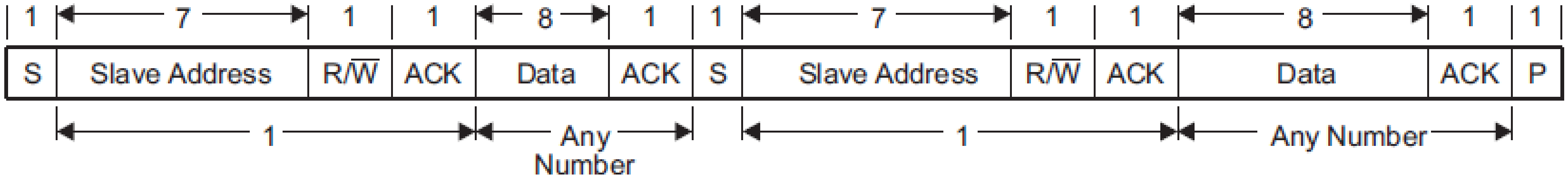
1	7	1	1	8	1	8	1	1
S	Slave Address	R/$\overline{W}$	ACK	Data	ACK	Data	ACK	P

Figure 24-5. I²C Module 7-Bit Addressing Format



1	7	1	1	8	1	8	1	1
S	Slave Address 1st byte	R/$\overline{W}$	ACK	Slave Address 2nd byte	ACK	Data	ACK	P
	1 1 1 1 0 X X							

Figure 24-6. I²C Module 10-Bit Addressing Format

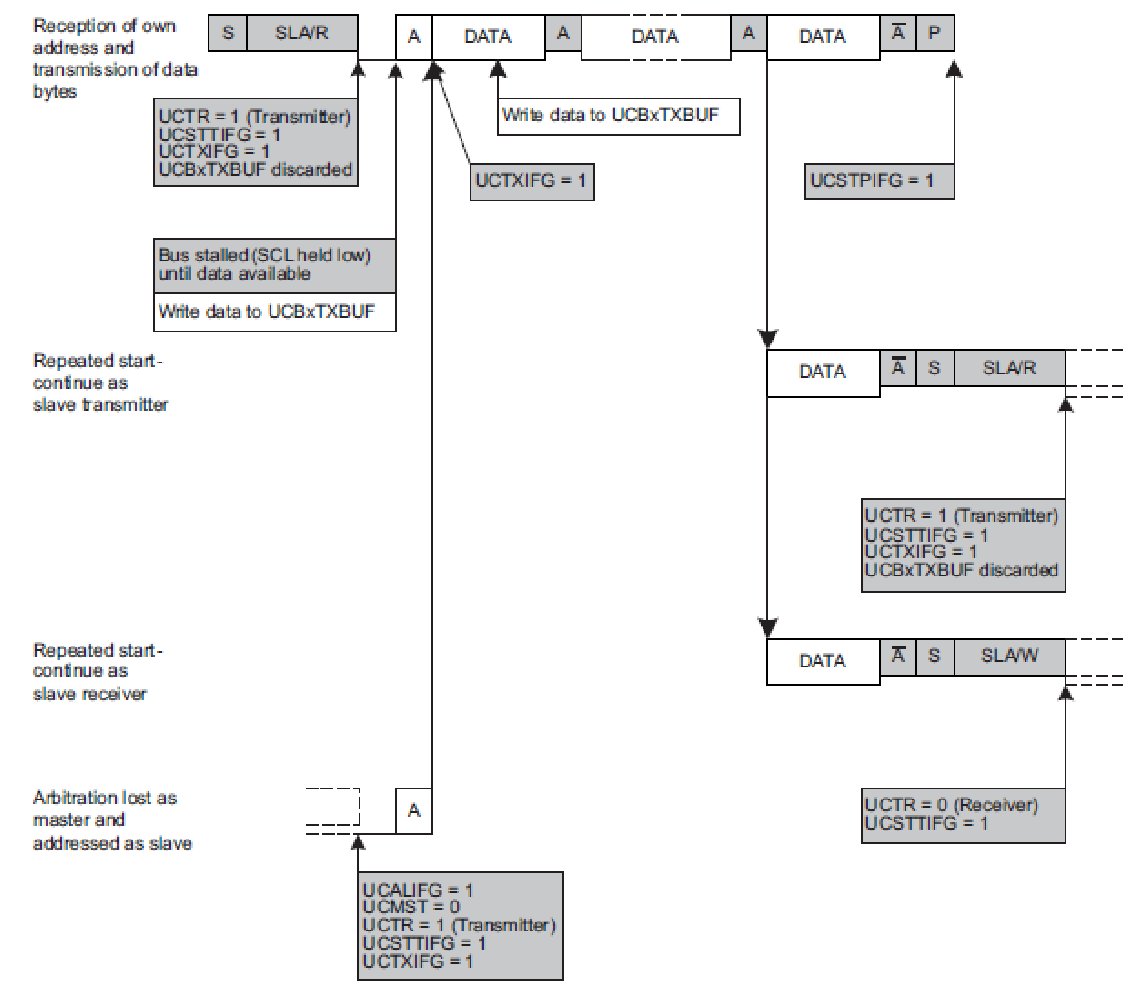


1	7	1	1	8	1	1	7	1	1	8	1	1
S	Slave Address	R/$\overline{W}$	ACK	Data	ACK	S	Slave Address	R/$\overline{W}$	ACK	Data	ACK	P
	1			Any Number			1			Any Number		

Figure 24-7. I²C Module Addressing Format With Repeated START Condition

Two Wire Interface

- Operating Modes



Reception of own address and transmission of data bytes

S | SLA/R | A | DATA | A | DATA | A | DATA | $\overline{A}$ | P

UCTR = 1 (Transmitter)
UCSTTIFG = 1
UCTXIFG = 1
UCBxTXBUF discarded

Write data to UCBxTXBUF

UCTXIFG = 1

UCSTPIFG = 1

Bus stalled (SCL held low) until data available

Write data to UCBxTXBUF

Repeated start - continue as slave transmitter

DATA | $\overline{A}$ | S | SLA/R

UCTR = 1 (Transmitter)
UCSTTIFG = 1
UCTXIFG = 1
UCBxTXBUF discarded

Repeated start - continue as slave receiver

DATA | $\overline{A}$ | S | SLA/W

UCTR = 0 (Receiver)
UCSTTIFG = 1

Arbitration lost as master and addressed as slave

A

UCALIFG = 1
UCMST = 0
UCTR = 1 (Transmitter)
UCSTTIFG = 1
UCTXIFG = 1

Figure 24-9. I²C Slave Transmitter Mode

Two Wire Interface

- Operating Modes

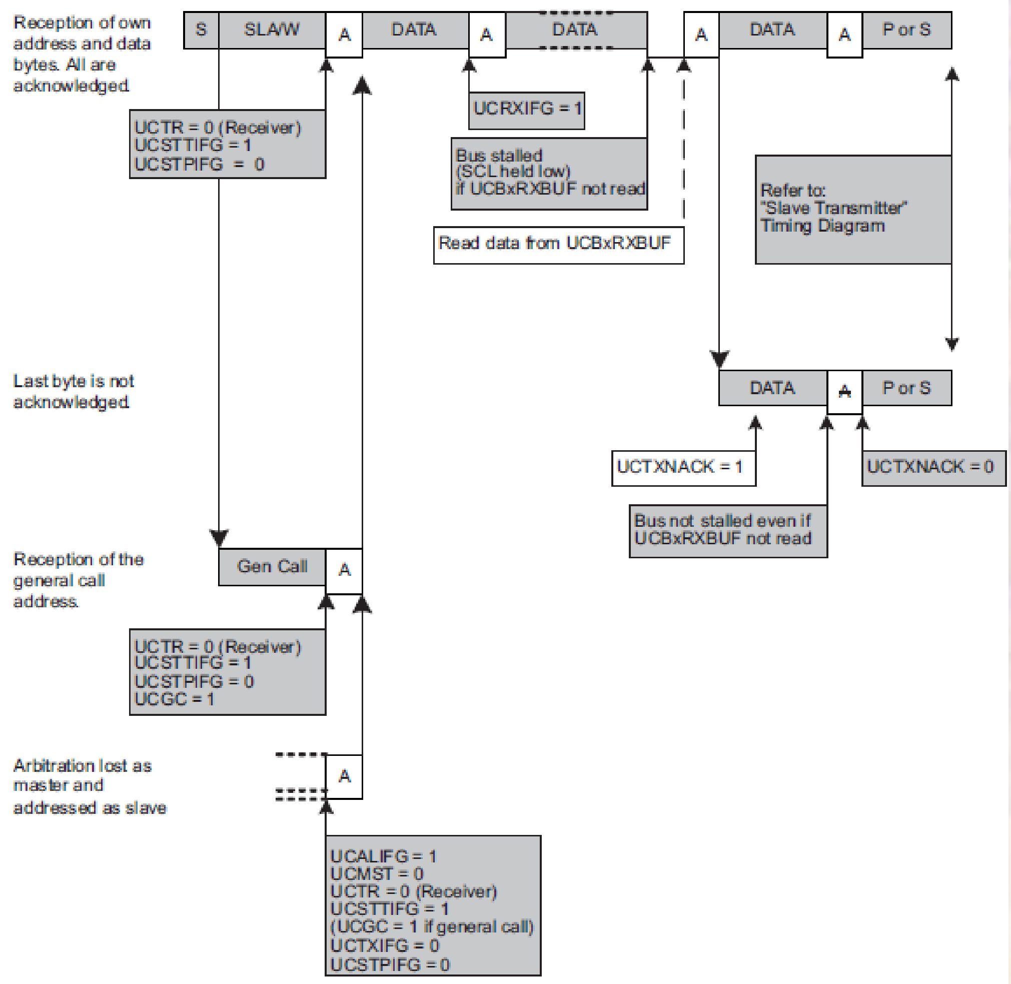

Reception of own address and data bytes. All are acknowledged.
S
SLA/W
A
DATA
A
DATA
A
DATA
A
P or S
UCTR = 0 (Receiver)
UCSTTIFG = 1
UCSTPIFG = 0
UCRXIFG = 1
Bus stalled (SCL held low) if UCBxRXBUF not read
Read data from UCBxRXBUF
Refer to: "Slave Transmitter" Timing Diagram
Last byte is not acknowledged.
DATA
A
P or S
UCTXNACK = 1
UCTXNACK = 0
Bus not stalled even if UCBxRXBUF not read
Reception of the general call address.
Gen Call
A
UCTR = 0 (Receiver)
UCSTTIFG = 1
UCSTPIFG = 0
UCGC = 1
Arbitration lost as master and addressed as slave
A
UCALIFG = 1
UCMST = 0
UCTR = 0 (Receiver)
UCSTTIFG = 1
(UCGC = 1 if general call)
UCTXIFG = 0
UCSTPIFG = 0

Figure 24-10. I²C Slave Receiver Mode

Two Wire Interface

- Operating Modes

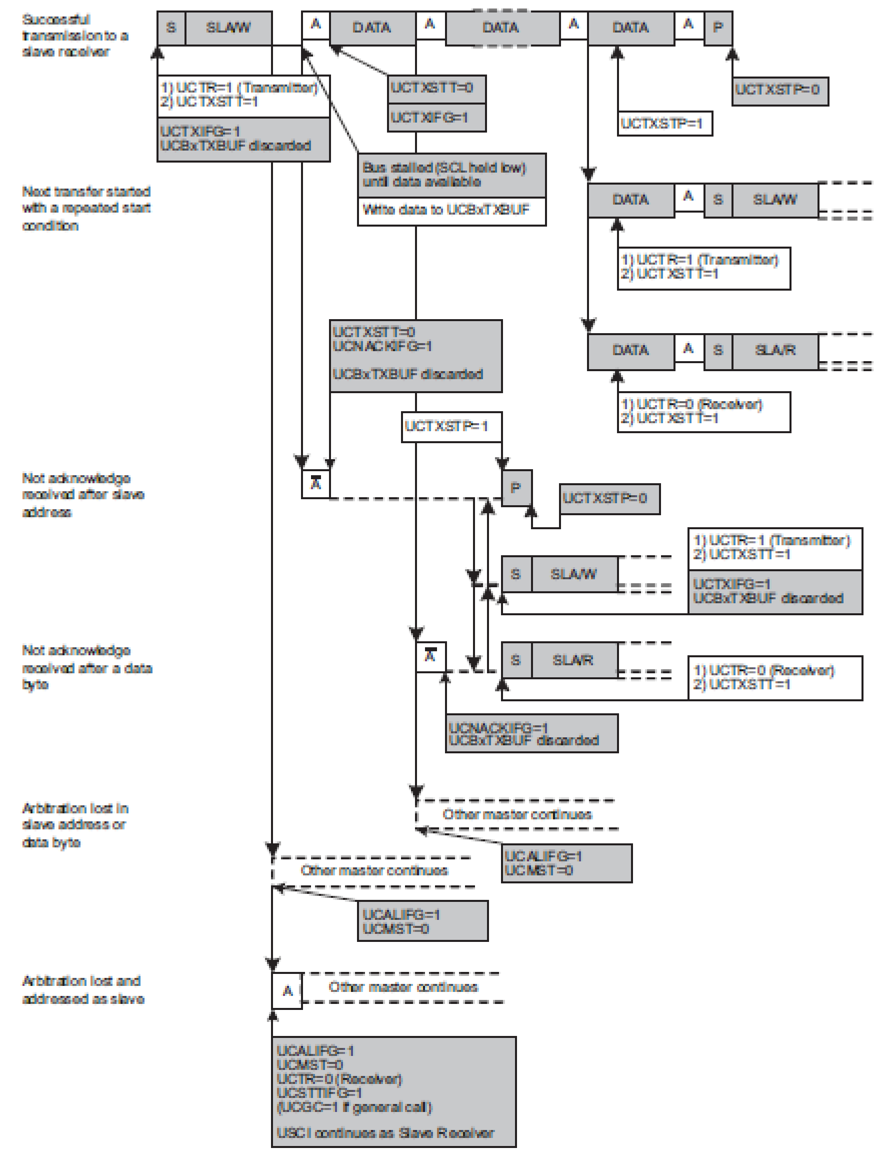


Figure 24-12. I²C Master Transmitter Mode

Two Wire Interface

- Operating Modes

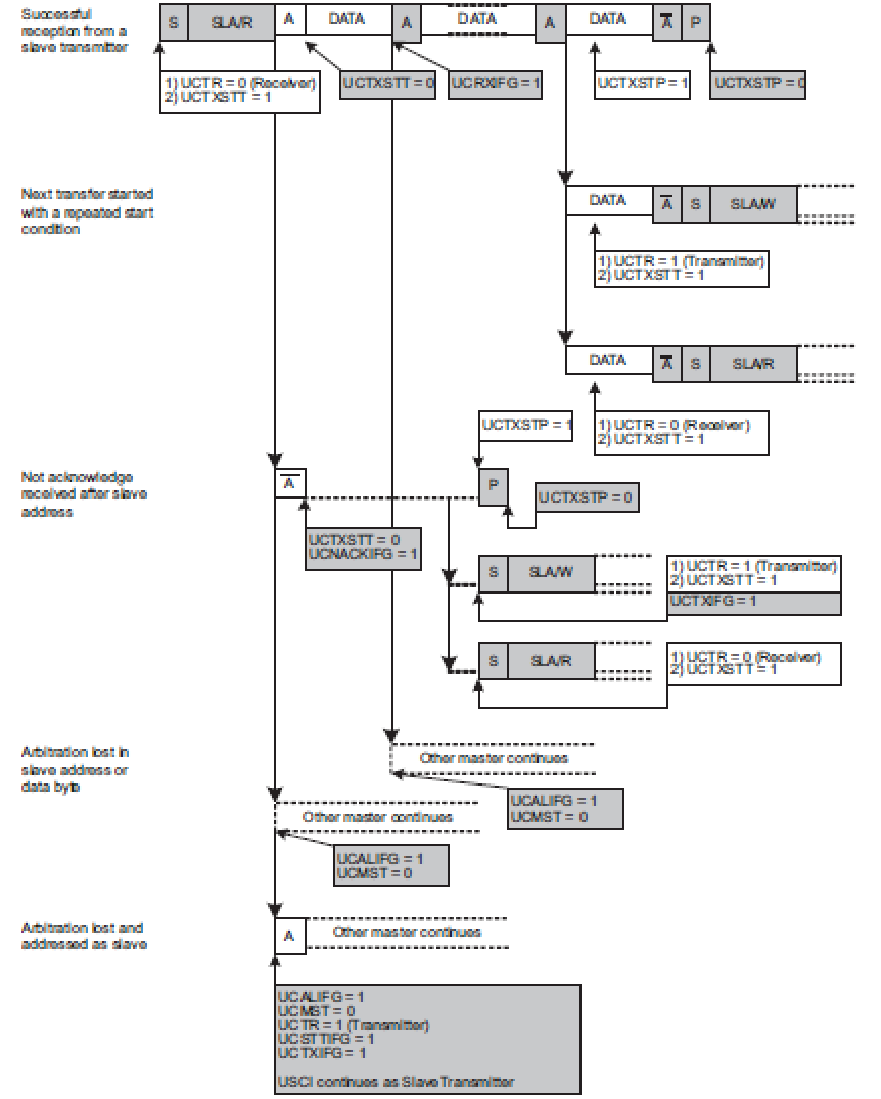
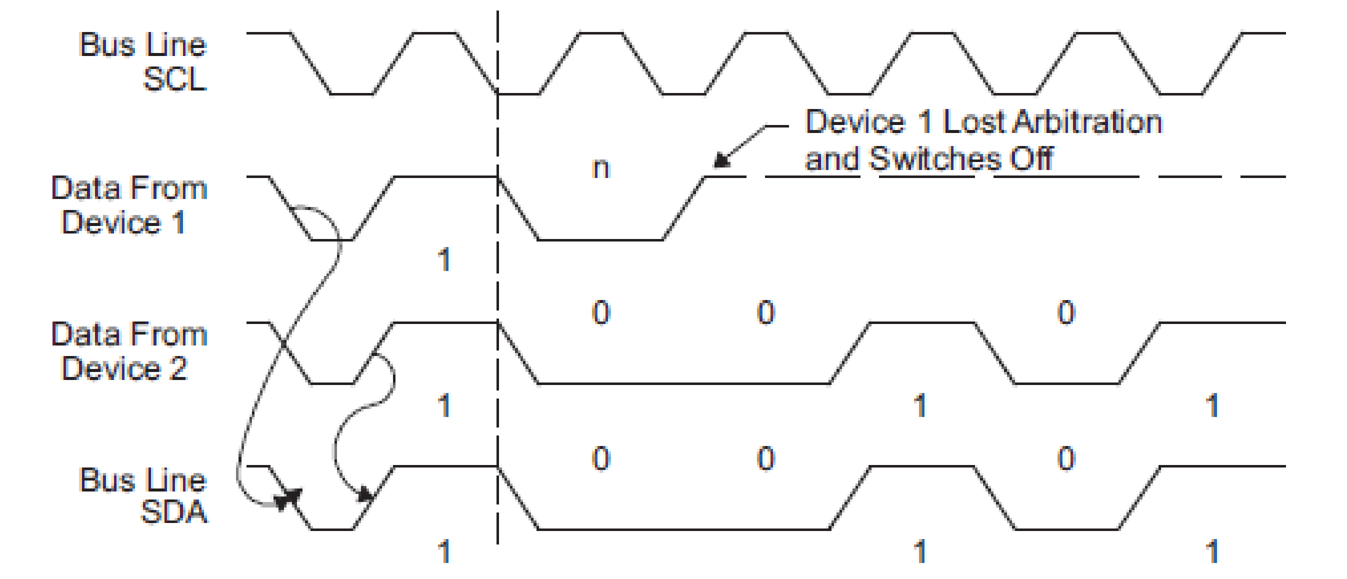


Figure 24-13. I²C Master Receiver Mode

Two Wire Interface

- Multi-Master Arbitration

  - First master that attempts to transmit a 1 when the other transmits a 0 – loses arbitration and shuts off



Two Wire Interface

- MSP432 I2C Registers

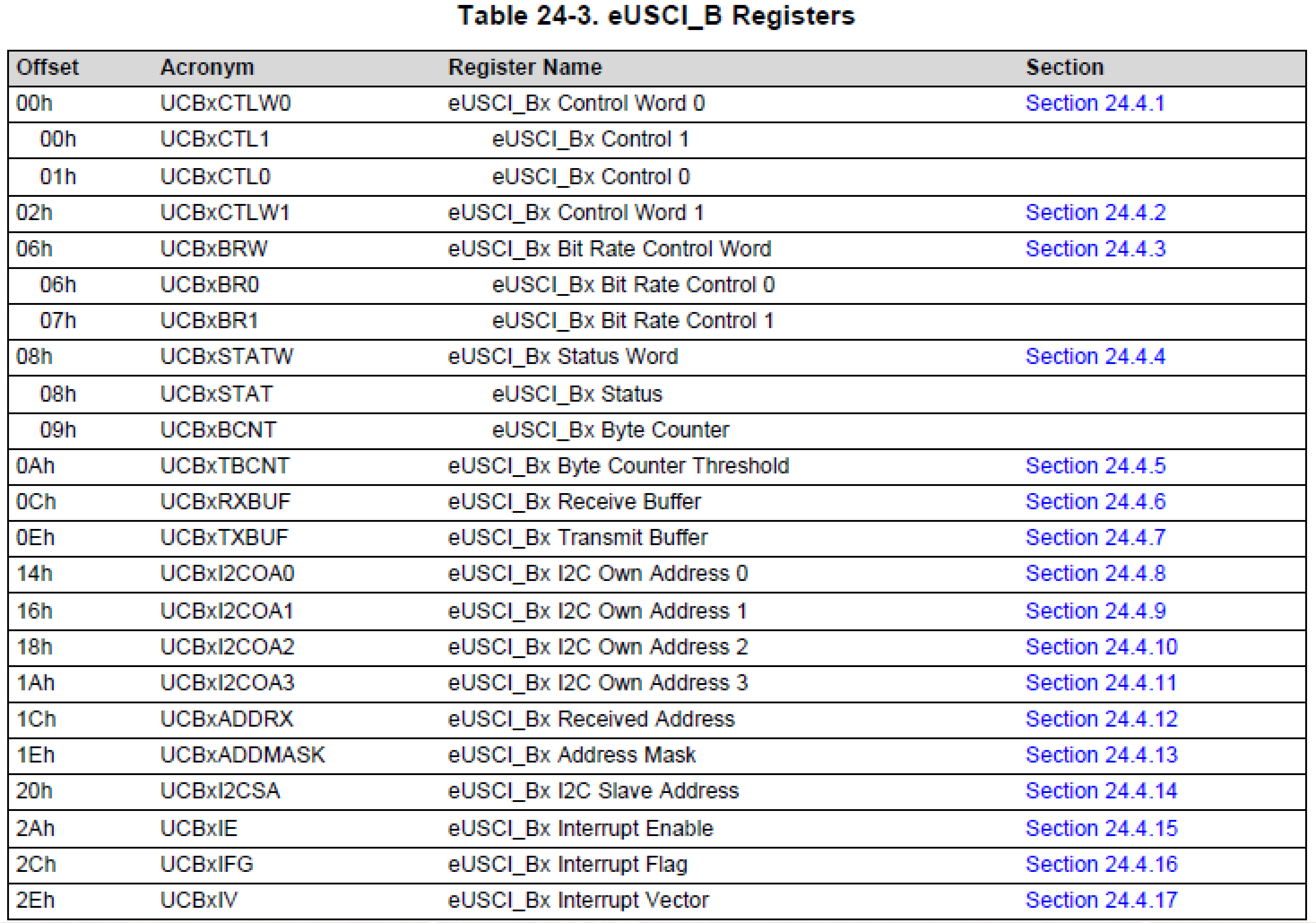


Table 24-3. eUSCI_B Registers

Offset	Acronym	Register Name	Section
00h	UCBxCTLW0	eUSCI_Bx Control Word 0	Section 24.4.1
00h	UCBxCTL1	eUSCI_Bx Control 1	
01h	UCBxCTL0	eUSCI_Bx Control 0	
02h	UCBxCTLW1	eUSCI_Bx Control Word 1	Section 24.4.2
06h	UCBxBRW	eUSCI_Bx Bit Rate Control Word	Section 24.4.3
06h	UCBxBR0	eUSCI_Bx Bit Rate Control 0	
07h	UCBxBR1	eUSCI_Bx Bit Rate Control 1	
08h	UCBxSTATW	eUSCI_Bx Status Word	Section 24.4.4
08h	UCBxSTAT	eUSCI_Bx Status	
09h	UCBxBCNT	eUSCI_Bx Byte Counter	
0Ah	UCBxTBCNT	eUSCI_Bx Byte Counter Threshold	Section 24.4.5
0Ch	UCBxRXBUF	eUSCI_Bx Receive Buffer	Section 24.4.6
0Eh	UCBxTXBUF	eUSCI_Bx Transmit Buffer	Section 24.4.7
14h	UCBxI2COA0	eUSCI_Bx I2C Own Address 0	Section 24.4.8
16h	UCBxI2COA1	eUSCI_Bx I2C Own Address 1	Section 24.4.9
18h	UCBxI2COA2	eUSCI_Bx I2C Own Address 2	Section 24.4.10
1Ah	UCBxI2COA3	eUSCI_Bx I2C Own Address 3	Section 24.4.11
1Ch	UCBxADDRX	eUSCI_Bx Received Address	Section 24.4.12
1Eh	UCBxADDMASK	eUSCI_Bx Address Mask	Section 24.4.13
20h	UCBxI2CSA	eUSCI_Bx I2C Slave Address	Section 24.4.14
2Ah	UCBxIE	eUSCI_Bx Interrupt Enable	Section 24.4.15
2Ch	UCBxIFG	eUSCI_Bx Interrupt Flag	Section 24.4.16
2Eh	UCBxIV	eUSCI_Bx Interrupt Vector	Section 24.4.17

Two Wire Interface

- MSP432 I2C Control Word Register 0

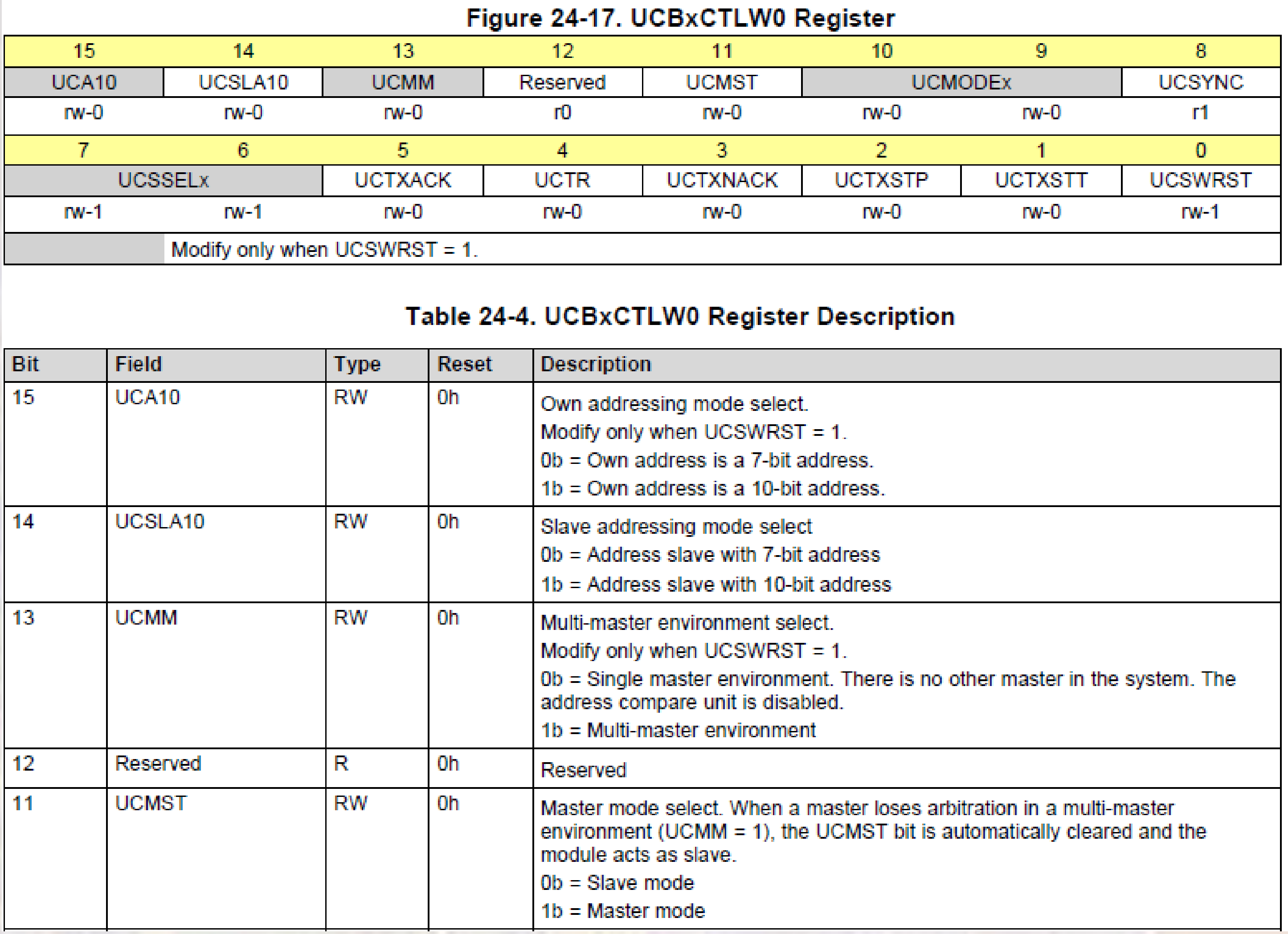


Figure 24-17. UCBxCTLW0 Register

15	14	13	12	11	10	9	8
UCA10	UCSLA10	UCMM	Reserved	UCMST	UCMODEx		UCSYNC
rw-0	rw-0	rw-0	r0	rw-0	rw-0	rw-0	r1
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
UCSSELx		UCTXACK	UCTR	UCTXNACK	UCTXSTP	UCTXSTT	UCSWRST
rw-1	rw-1	rw-0	rw-0	rw-0	rw-0	rw-0	rw-1

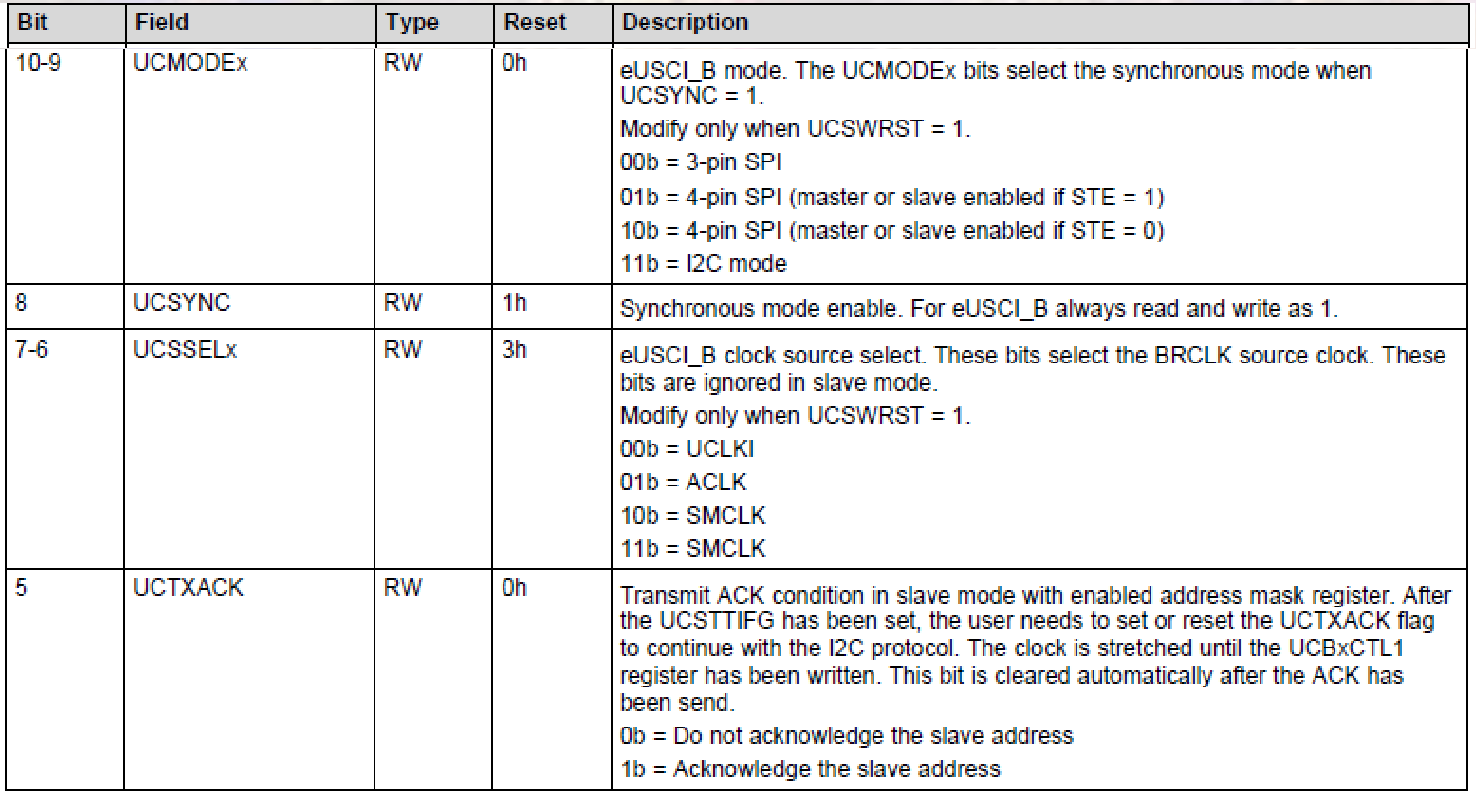
▨ Modify only when UCSWRST = 1.

Table 24-4. UCBxCTLW0 Register Description

Bit	Field	Type	Reset	Description
15	UCA10	RW	0h	Own addressing mode select. Modify only when UCSWRST = 1. 0b = Own address is a 7-bit address. 1b = Own address is a 10-bit address.
14	UCSLA10	RW	0h	Slave addressing mode select 0b = Address slave with 7-bit address 1b = Address slave with 10-bit address
13	UCMM	RW	0h	Multi-master environment select. Modify only when UCSWRST = 1. 0b = Single master environment. There is no other master in the system. The address compare unit is disabled. 1b = Multi-master environment
12	Reserved	R	0h	Reserved
11	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

Two Wire Interface

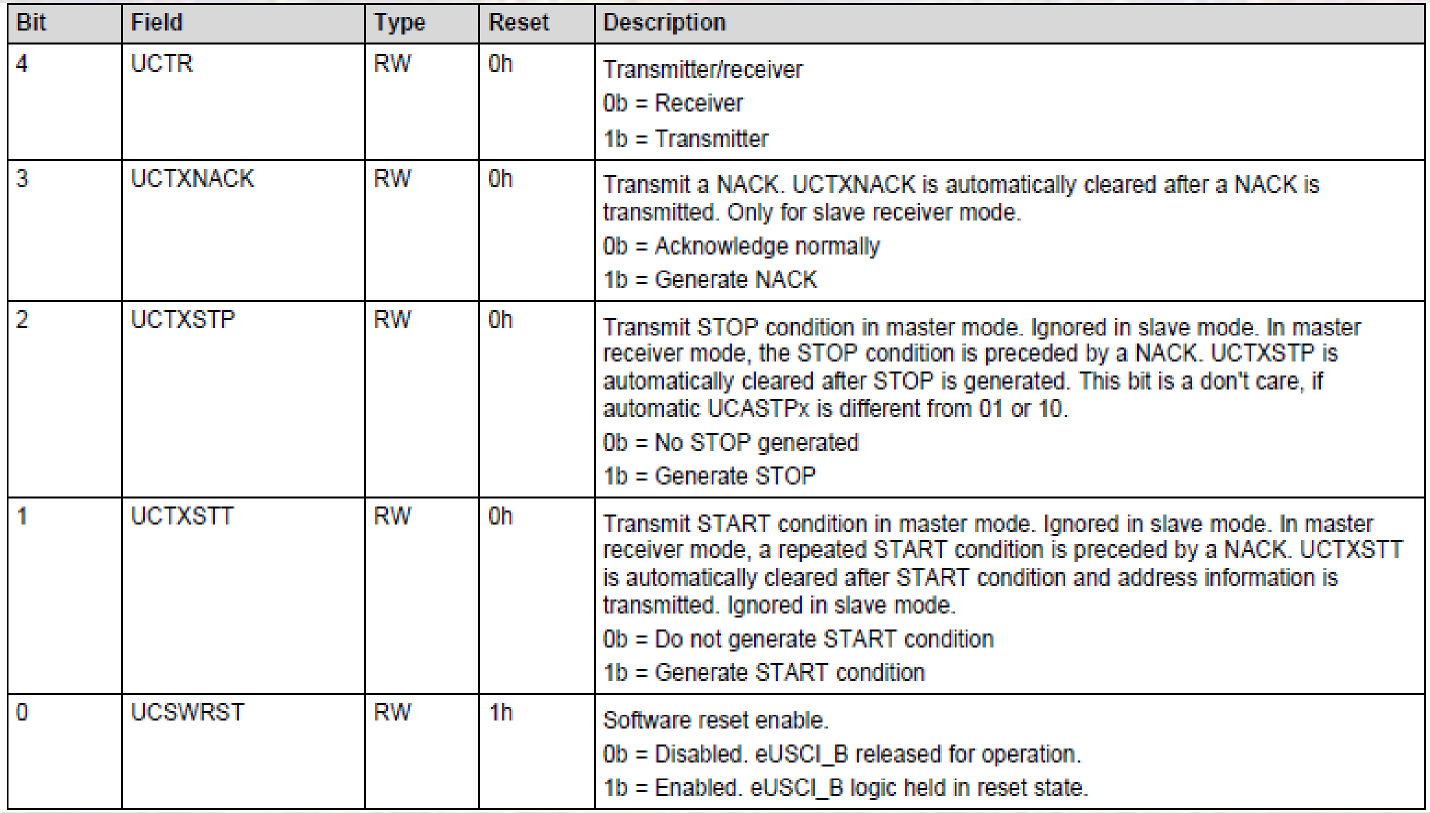
- MSP432 I2C Control Word Register 0



Bit	Field	Type	Reset	Description
10-9	UCMODEx	RW	0h	eUSCI_B mode. The UCMODEx bits select the synchronous mode when UCSYNC = 1. Modify only when UCSWRST = 1. 00b = 3-pin SPI 01b = 4-pin SPI (master or slave enabled if STE = 1) 10b = 4-pin SPI (master or slave enabled if STE = 0) 11b = I2C mode
8	UCSYNC	RW	1h	Synchronous mode enable. For eUSCI_B always read and write as 1.
7-6	UCSSELx	RW	3h	eUSCI_B clock source select. These bits select the BRCLK source clock. These bits are ignored in slave mode. Modify only when UCSWRST = 1. 00b = UCLKI 01b = ACLK 10b = SMCLK 11b = SMCLK
5	UCTXACK	RW	0h	Transmit ACK condition in slave mode with enabled address mask register. After the UCSTTIFG has been set, the user needs to set or reset the UCTXACK flag to continue with the I2C protocol. The clock is stretched until the UCBxCTL1 register has been written. This bit is cleared automatically after the ACK has been send. 0b = Do not acknowledge the slave address 1b = Acknowledge the slave address

Two Wire Interface

- MSP432 I2C Control Word Register 0



Bit	Field	Type	Reset	Description
4	UCTR	RW	0h	Transmitter/receiver 0b = Receiver 1b = Transmitter
3	UCTXNACK	RW	0h	Transmit a NACK. UCTXNACK is automatically cleared after a NACK is transmitted. Only for slave receiver mode. 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. This bit is a don't care, if automatic UCASTPx is different from 01 or 10. 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. eUSCI_B released for operation. 1b = Enabled. eUSCI_B logic held in reset state.

Two Wire Interface

- MSP432 I2C Control Word Register 1

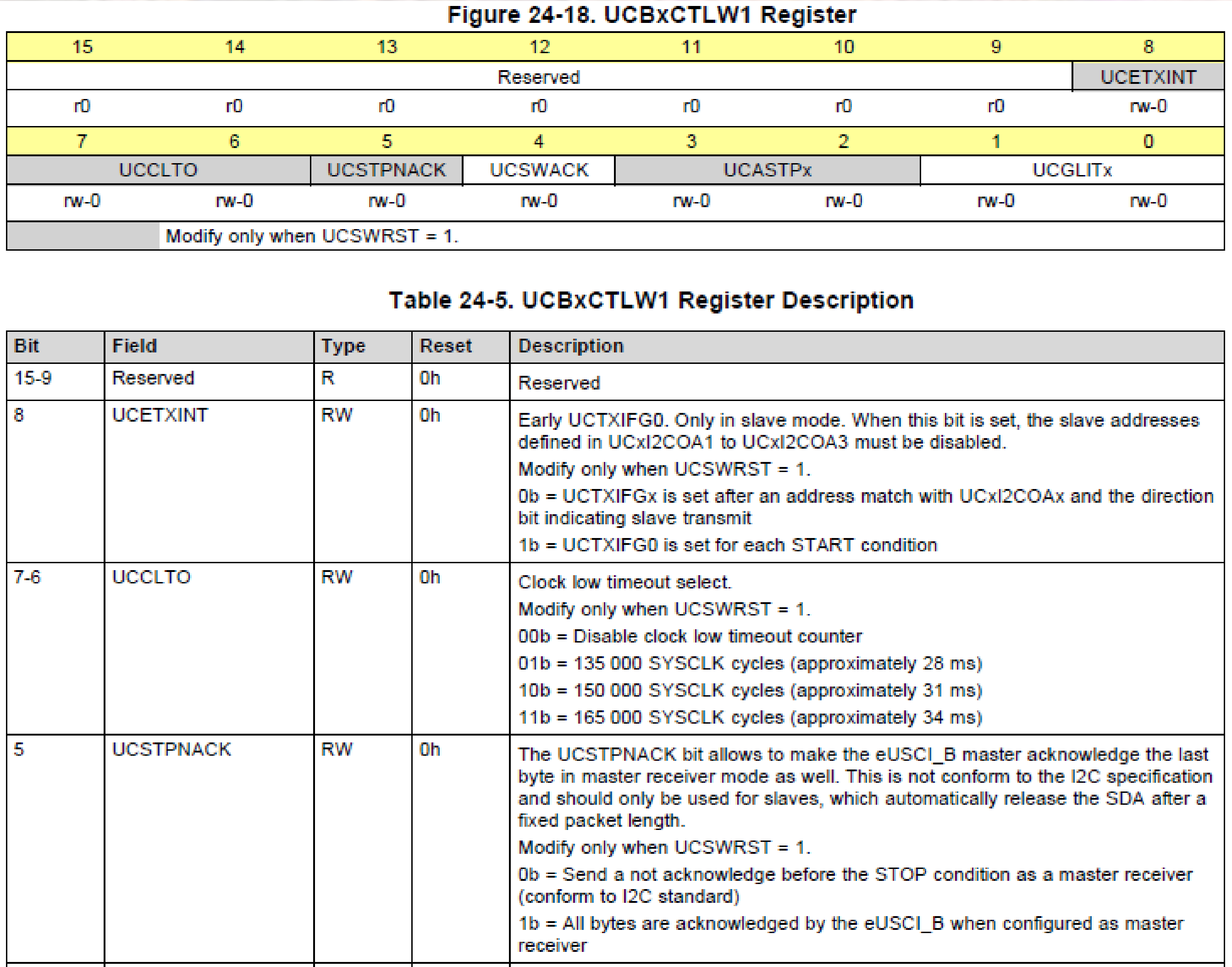


Figure 24-18. UCBxCTLW1 Register

15	14	13	12	11	10	9	8
Reserved							UCETXINT
r0	r0	r0	r0	r0	r0	r0	rw-0
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
UCCLTO		UCSTPNACK	UCSWACK	UCASTPx		UCGLITx	
rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0

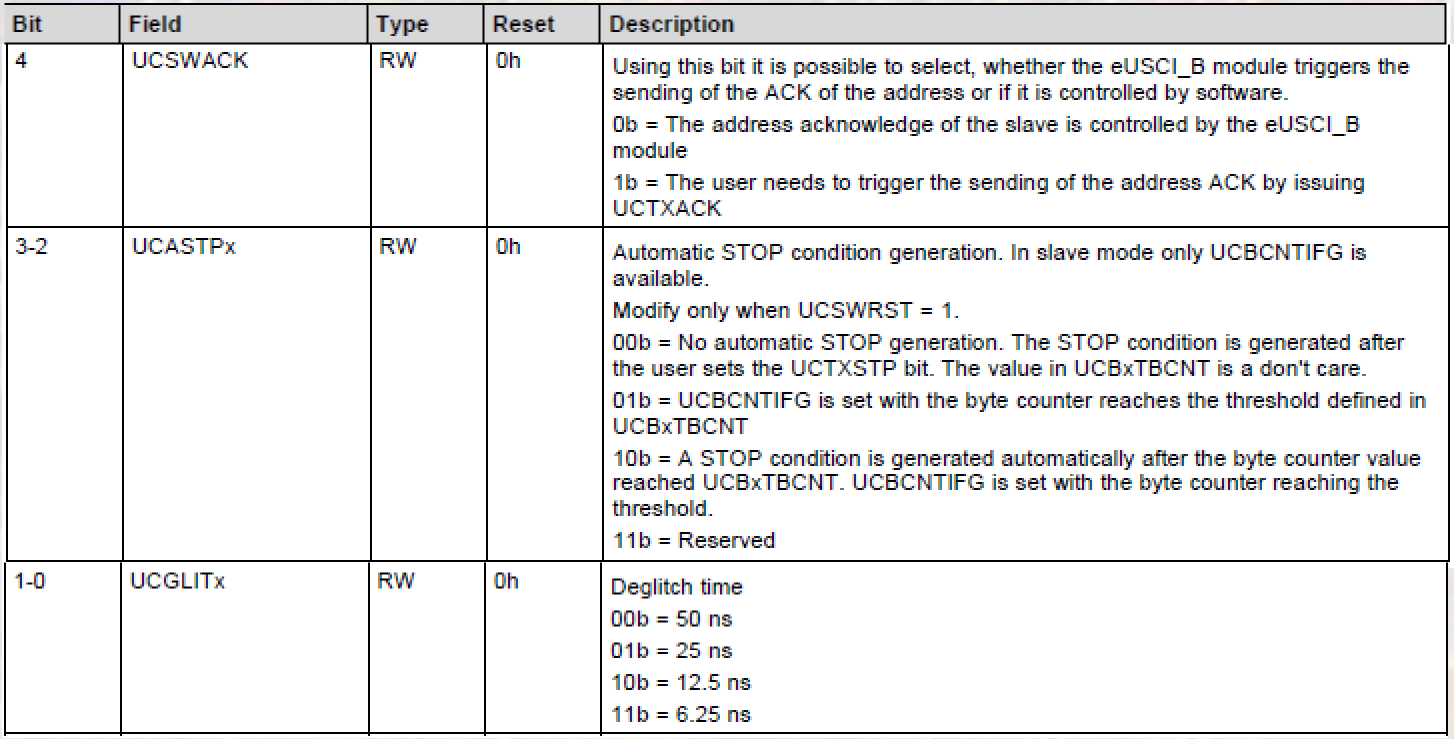
▨ Modify only when UCSWRST = 1.

Table 24-5. UCBxCTLW1 Register Description

Bit	Field	Type	Reset	Description
15-9	Reserved	R	0h	Reserved
8	UCETXINT	RW	0h	Early UCTXIFG0. Only in slave mode. When this bit is set, the slave addresses defined in UCxI2COA1 to UCxI2COA3 must be disabled. Modify only when UCSWRST = 1. 0b = UCTXIFGx is set after an address match with UCxI2COAx and the direction bit indicating slave transmit 1b = UCTXIFG0 is set for each START condition
7-6	UCCLTO	RW	0h	Clock low timeout select. Modify only when UCSWRST = 1. 00b = Disable clock low timeout counter 01b = 135 000 SYSCLK cycles (approximately 28 ms) 10b = 150 000 SYSCLK cycles (approximately 31 ms) 11b = 165 000 SYSCLK cycles (approximately 34 ms)
5	UCSTPNACK	RW	0h	The UCSTPNACK bit allows to make the eUSCI_B master acknowledge the last byte in master receiver mode as well. This is not conform to the I2C specification and should only be used for slaves, which automatically release the SDA after a fixed packet length. Modify only when UCSWRST = 1. 0b = Send a not acknowledge before the STOP condition as a master receiver (conform to I2C standard) 1b = All bytes are acknowledged by the eUSCI_B when configured as master receiver

Two Wire Interface

- MSP432 I2C Control Word Register 1



Bit	Field	Type	Reset	Description
4	UCSWACK	RW	0h	Using this bit it is possible to select, whether the eUSCI_B module triggers the sending of the ACK of the address or if it is controlled by software. 0b = The address acknowledge of the slave is controlled by the eUSCI_B module 1b = The user needs to trigger the sending of the address ACK by issuing UCTXACK
3-2	UCASTPx	RW	0h	Automatic STOP condition generation. In slave mode only UCBCNTIFG is available. Modify only when UCSWRST = 1. 00b = No automatic STOP generation. The STOP condition is generated after the user sets the UCTXSTP bit. The value in UCBxTBCNT is a don't care. 01b = UCBCNTIFG is set with the byte counter reaches the threshold defined in UCBxTBCNT 10b = A STOP condition is generated automatically after the byte counter value reached UCBxTBCNT. UCBCNTIFG is set with the byte counter reaching the threshold. 11b = Reserved
1-0	UCGLITx	RW	0h	Deglitch time 00b = 50 ns 01b = 25 ns 10b = 12.5 ns 11b = 6.25 ns

Two Wire Interface

- MSP432 I2C Bit Rate Control Word Register
  - TWI transfer bit rate: $f_{BitClock} = f_{BRCLK}/UCBRx$
    - BRCLK: ACLK or SMCLK

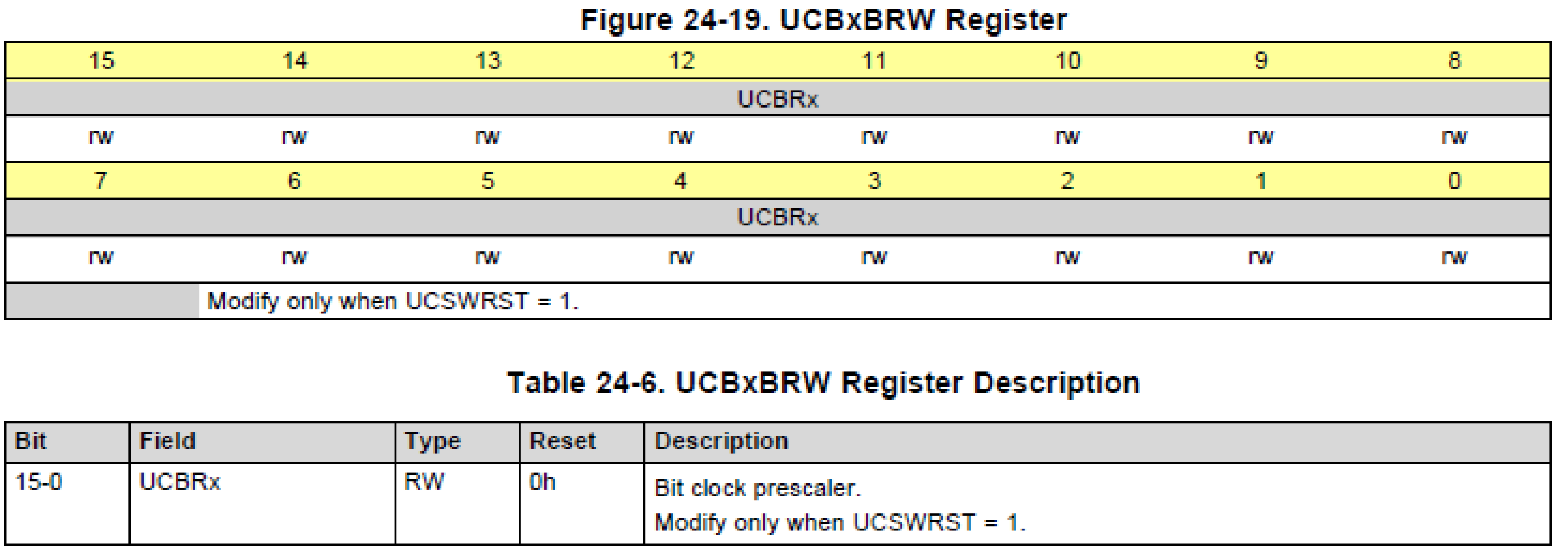


Figure 24-19. UCBxBRW Register

15	14	13	12	11	10	9	8
UCBRx							
rw	rw	rw	rw	rw	rw	rw	rw
7	6	5	4	3	2	1	0
UCBRx							
rw	rw	rw	rw	rw	rw	rw	rw

▨ Modify only when UCSWRST = 1.

Table 24-6. UCBxBRW Register Description

Bit	Field	Type	Reset	Description
15-0	UCBRx	RW	0h	Bit clock prescaler. Modify only when UCSWRST = 1.

Two Wire Interface

- MSP432 I2C Status Word Register

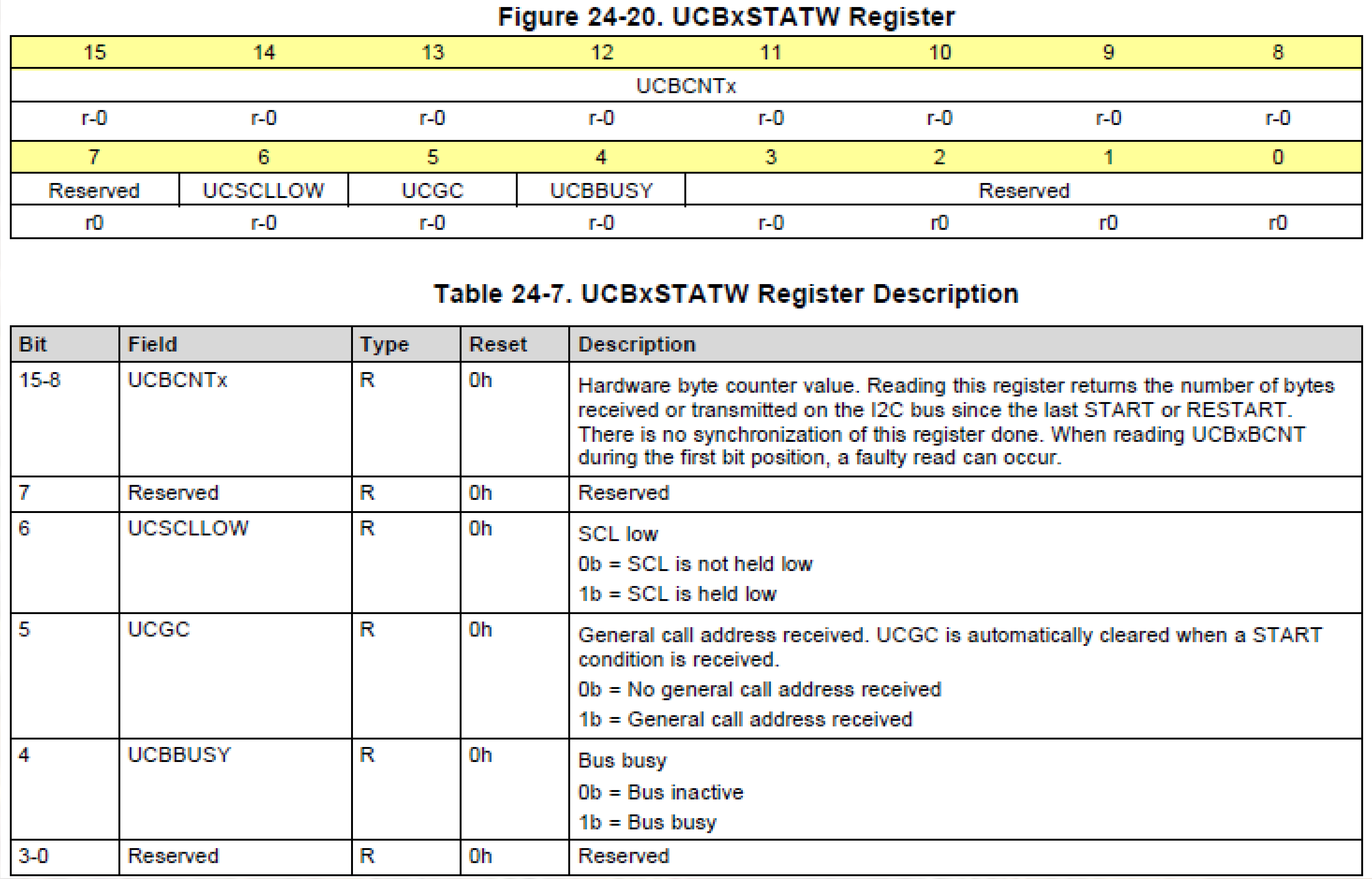


Figure 24-20. UCBxSTATW Register

15	14	13	12	11	10	9	8
UCBCNTx							
r-0	r-0	r-0	r-0	r-0	r-0	r-0	r-0

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

Table 24-7. UCBxSTATW Register Description

Bit	Field	Type	Reset	Description
15-8	UCBCNTx	R	0h	Hardware byte counter value. Reading this register returns the number of bytes received or transmitted on the I2C bus since the last START or RESTART. There is no synchronization of this register done. When reading UCBxBCNT during the first bit position, a faulty read can occur.
7	Reserved	R	0h	Reserved
6	UCSCLLOW	R	0h	SCL low 0b = SCL is not held low 1b = SCL is held low
5	UCGC	R	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

Two Wire Interface

- MSP432 I2C Byte Count Threshold Register

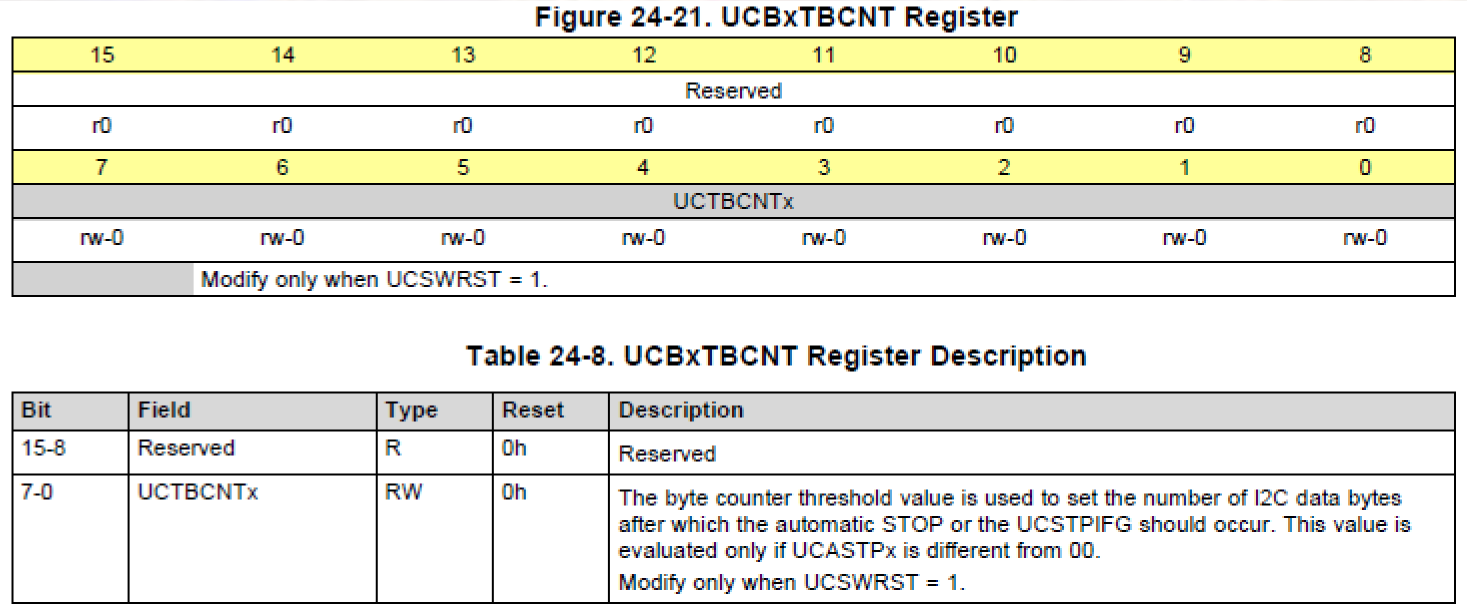


Figure 24-21. UCBxTBCNT Register

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

▨ Modify only when UCSWRST = 1.

Table 24-8. UCBxTBCNT Register Description

Bit	Field	Type	Reset	Description
15-8	Reserved	R	0h	Reserved
7-0	UCTBCNTx	RW	0h	The byte counter threshold value is used to set the number of I2C data bytes after which the automatic STOP or the UCSTPIFG should occur. This value is evaluated only if UCASTPx is different from 00. Modify only when UCSWRST = 1.

Two Wire Interface

- MSP432 I2C Receive Buffer Register

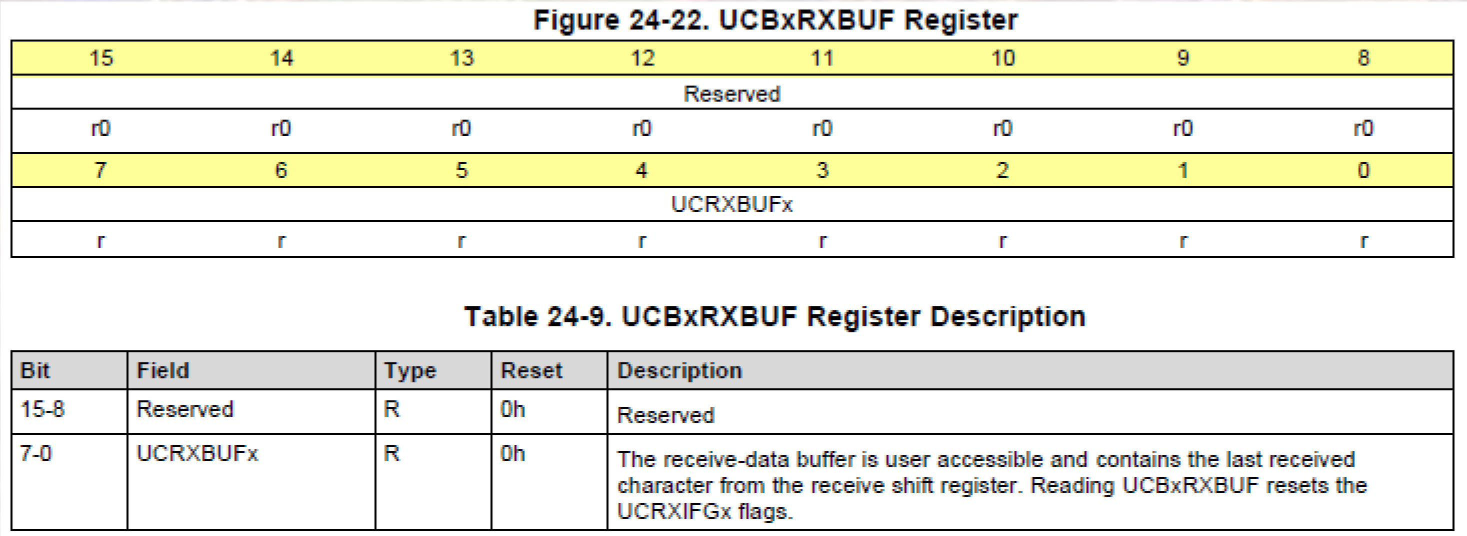


Figure 24-22. UCBxRXBUF Register

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

Table 24-9. UCBxRXBUF Register Description

Bit	Field	Type	Reset	Description
15-8	Reserved	R	0h	Reserved
7-0	UCRXBUFx	R	0h	The receive-data buffer is user accessible and contains the last received character from the receive shift register. Reading UCBxRXBUF resets the UCRXIFGx flags.

Two Wire Interface

- MSP432 I2C Transmit Buffer Register

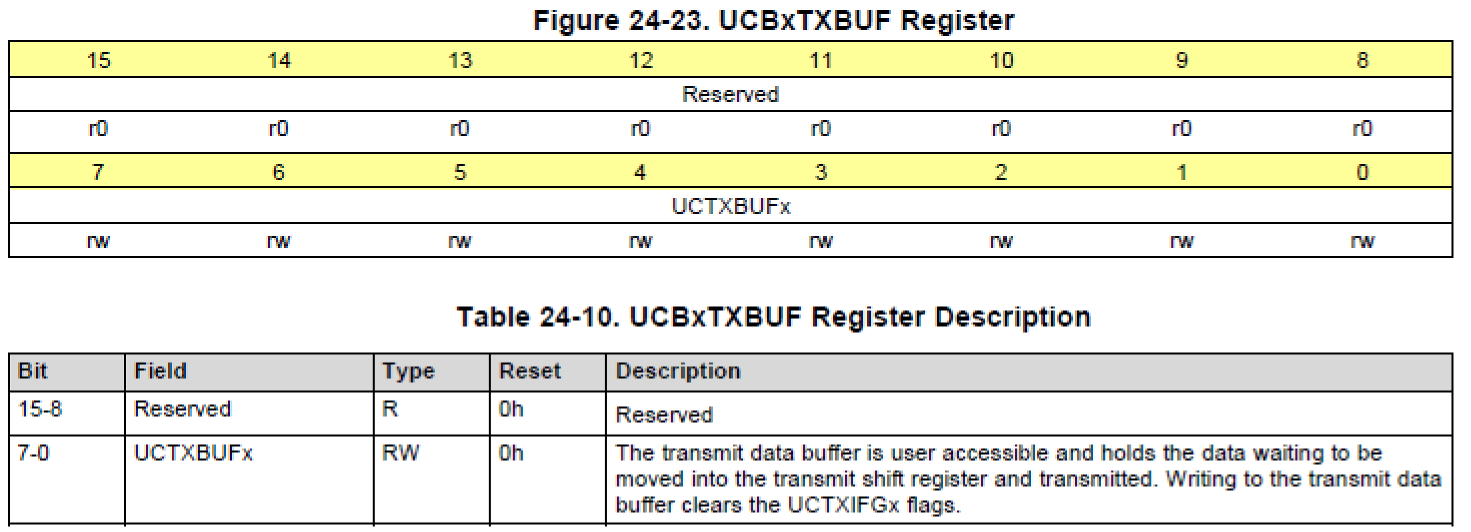


Figure 24-23. UCBxTXBUF Register

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

Table 24-10. UCBxTXBUF Register Description

Bit	Field	Type	Reset	Description
15-8	Reserved	R	0h	Reserved
7-0	UCTXBUFx	RW	0h	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 the UCTXIFGx flags.

Two Wire Interface

- MSP432 I2C Own Address Register 0

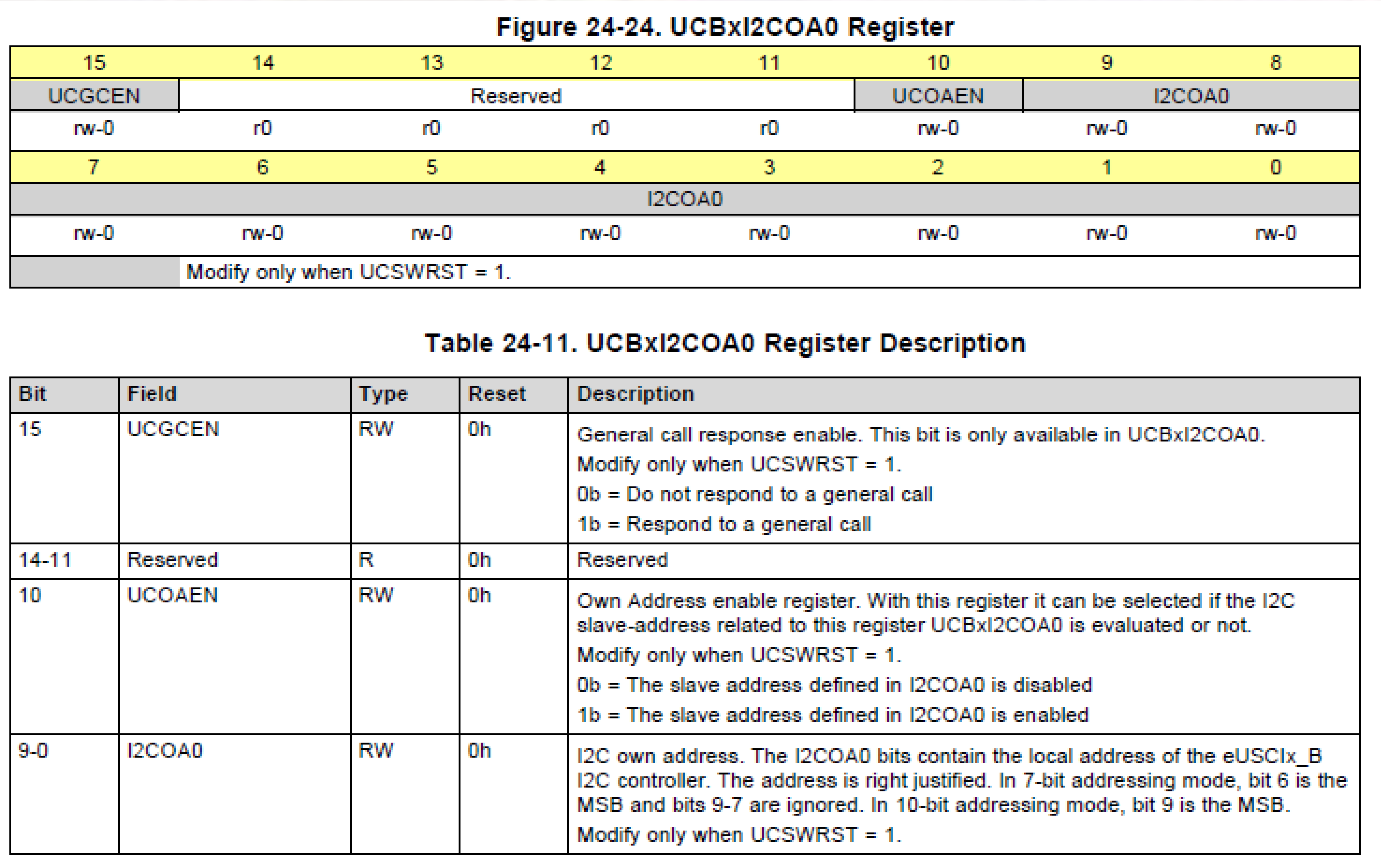


Figure 24-24. UCBxI2COA0 Register

15	14	13	12	11	10	9	8
UCGCEN	Reserved				UCOAEN	I2COA0	
rw-0	r0	r0	r0	r0	rw-0	rw-0	rw-0
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
I2COA0							
rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0
▨	Modify only when UCSWRST = 1.						

Table 24-11. UCBxI2COA0 Register Description

Bit	Field	Type	Reset	Description
15	UCGCEN	RW	0h	General call response enable. This bit is only available in UCBxI2COA0. Modify only when UCSWRST = 1. 0b = Do not respond to a general call 1b = Respond to a general call
14-11	Reserved	R	0h	Reserved
10	UCOAEN	RW	0h	Own Address enable register. With this register it can be selected if the I2C slave-address related to this register UCBxI2COA0 is evaluated or not. Modify only when UCSWRST = 1. 0b = The slave address defined in I2COA0 is disabled 1b = The slave address defined in I2COA0 is enabled
9-0	I2COA0	RW	0h	I2C own address. The I2COA0 bits contain the local address of the eUSCIx_B 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. Modify only when UCSWRST = 1.

Two Wire Interface

- MSP432 I2C Own Address Registers 1,2,3

1,2,3

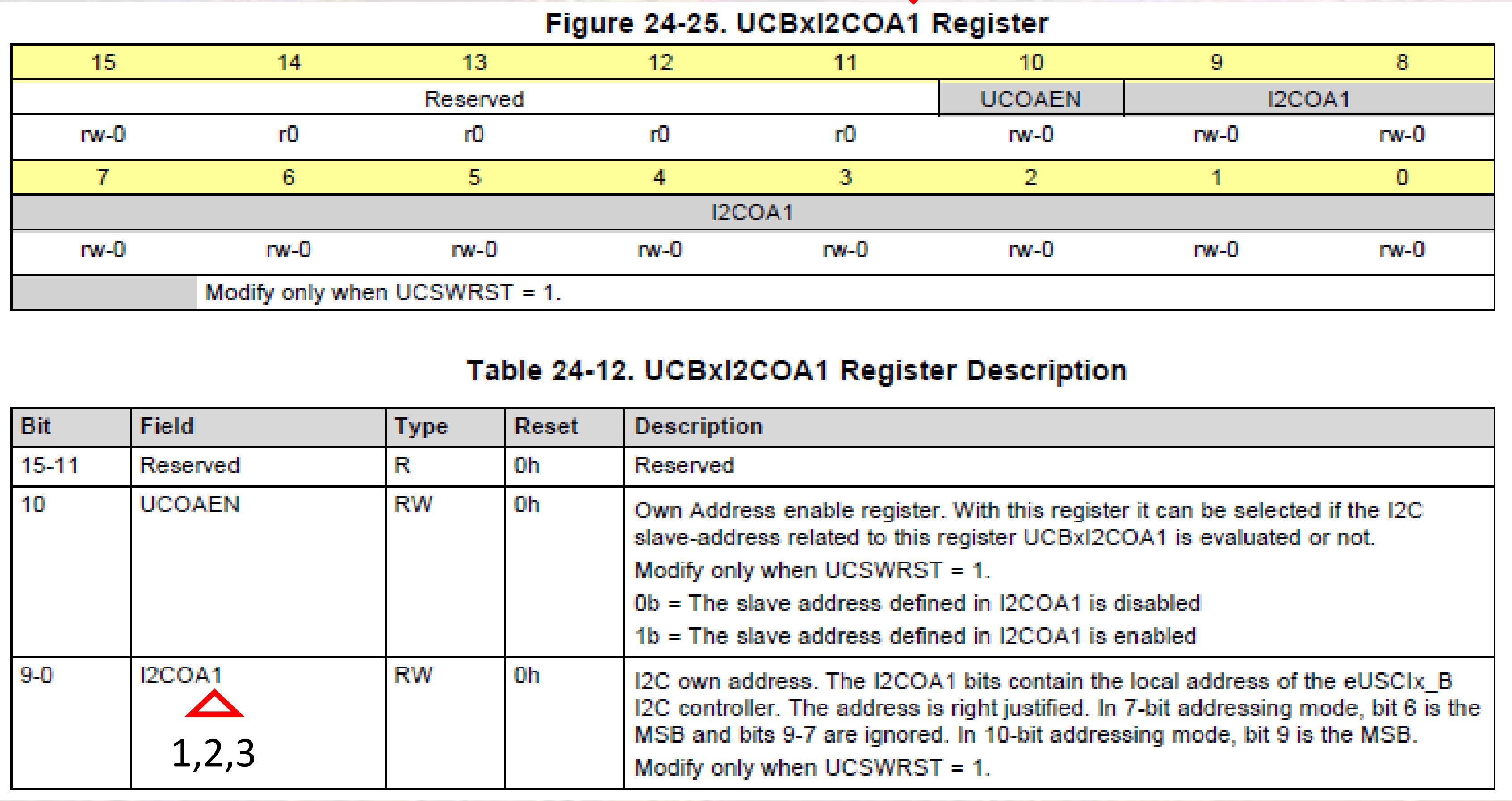


Figure 24-25. UCBxI2COA1 Register

15	14	13	12	11	10	9	8
Reserved					UCOAEN	I2COA1	
rw-0	r0	r0	r0	r0	rw-0	rw-0	rw-0
7	6	5	4	3	2	1	0
I2COA1							
rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0

▨ Modify only when UCSWRST = 1.

Table 24-12. UCBxI2COA1 Register Description

Bit	Field	Type	Reset	Description
15-11	Reserved	R	0h	Reserved
10	UCOAEN	RW	0h	Own Address enable register. With this register it can be selected if the I2C slave-address related to this register UCBxI2COA1 is evaluated or not. Modify only when UCSWRST = 1. 0b = The slave address defined in I2COA1 is disabled 1b = The slave address defined in I2COA1 is enabled
9-0	I2COA1 1,2,3	RW	0h	I2C own address. The I2COA1 bits contain the local address of the eUSCIx_B 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. Modify only when UCSWRST = 1.

Two Wire Interface

- MSP432 I2C Receive Address Register

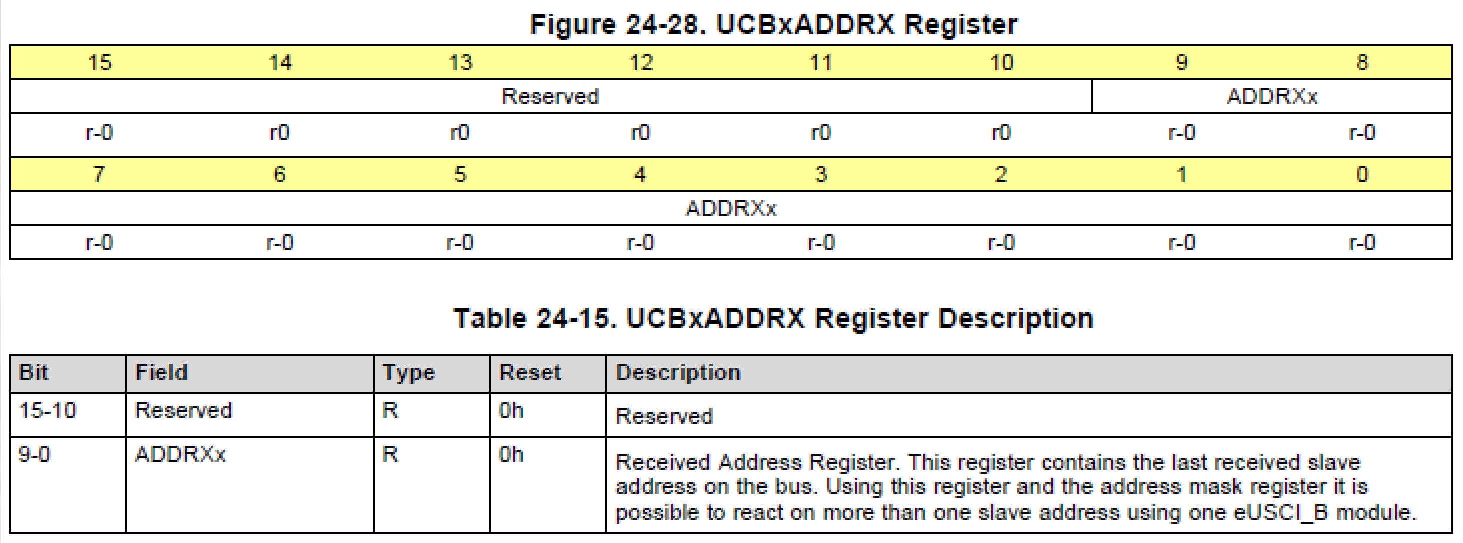


Figure 24-28. UCBxADDRX Register

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

Table 24-15. UCBxADDRX Register Description

Bit	Field	Type	Reset	Description
15-10	Reserved	R	0h	Reserved
9-0	ADDRXx	R	0h	Received Address Register. This register contains the last received slave address on the bus. Using this register and the address mask register it is possible to react on more than one slave address using one eUSCI_B module.

Two Wire Interface

- MSP432 I2C Address Mask Register

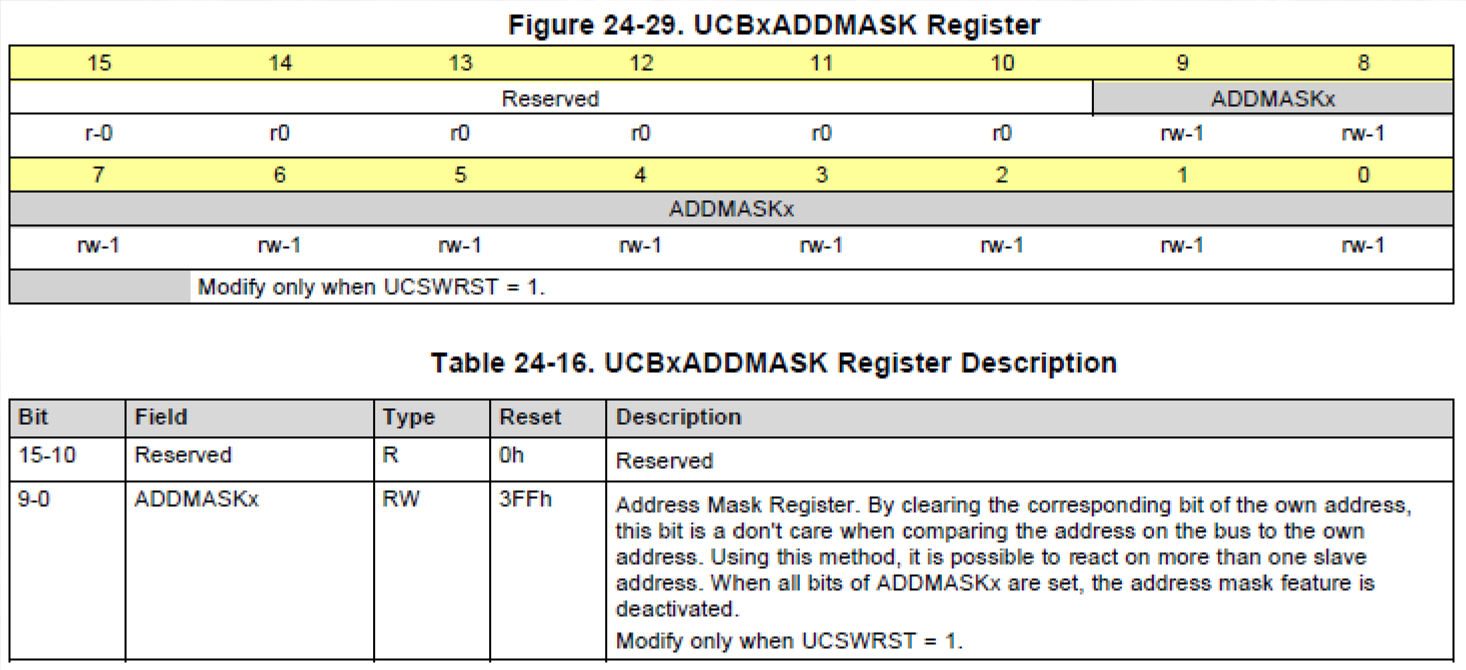


Figure 24-29. UCBxADDMASK Register

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

Modify only when UCSWRST = 1.

Table 24-16. UCBxADDMASK Register Description

Bit	Field	Type	Reset	Description
15-10	Reserved	R	0h	Reserved
9-0	ADDMASKx	RW	3FFh	Address Mask Register. By clearing the corresponding bit of the own address, this bit is a don't care when comparing the address on the bus to the own address. Using this method, it is possible to react on more than one slave address. When all bits of ADDMASKx are set, the address mask feature is deactivated. Modify only when UCSWRST = 1.

Two Wire Interface

- MSP432 I2C Slave Address Register

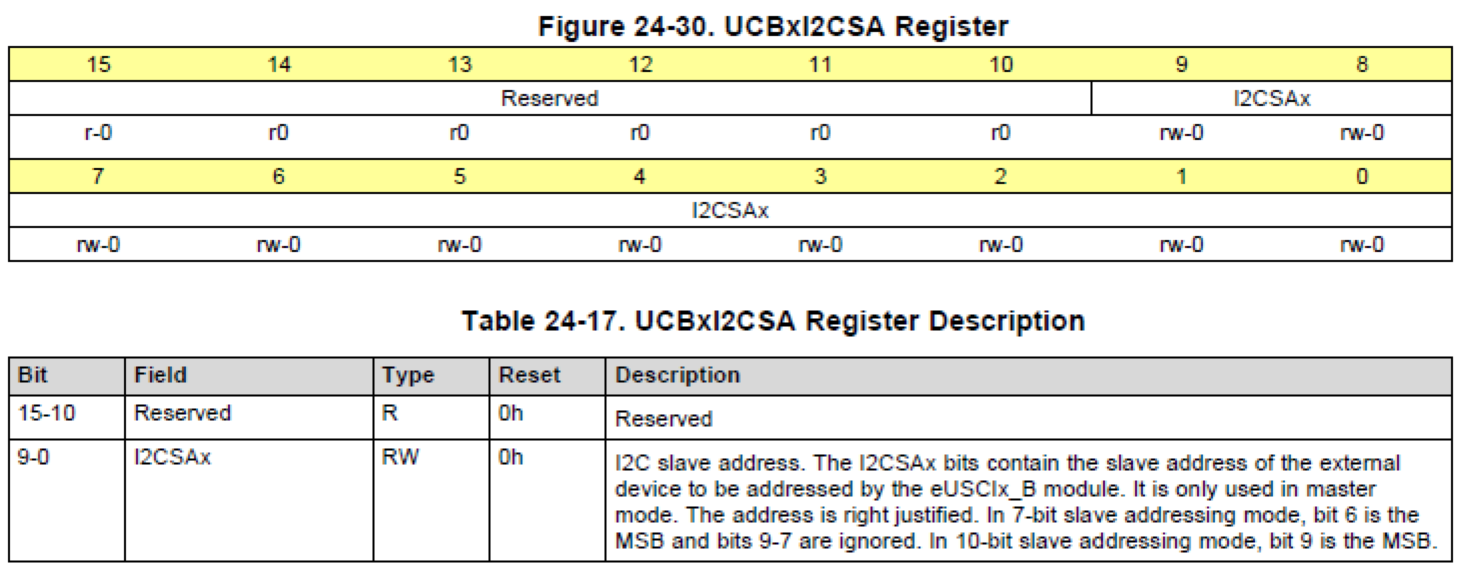


Figure 24-30. UCBxI2CSA Register

15	14	13	12	11	10	9	8
Reserved						I2CSAx	
r-0	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 24-17. UCBxI2CSA Register Description

Bit	Field	Type	Reset	Description
15-10	Reserved	R	0h	Reserved
9-0	I2CSAx	RW	0h	I2C slave address. The I2CSAx bits contain the slave address of the external device to be addressed by the eUSCIx_B 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.

Two Wire Interface

- MSP432 I2C Interrupt Enable Register

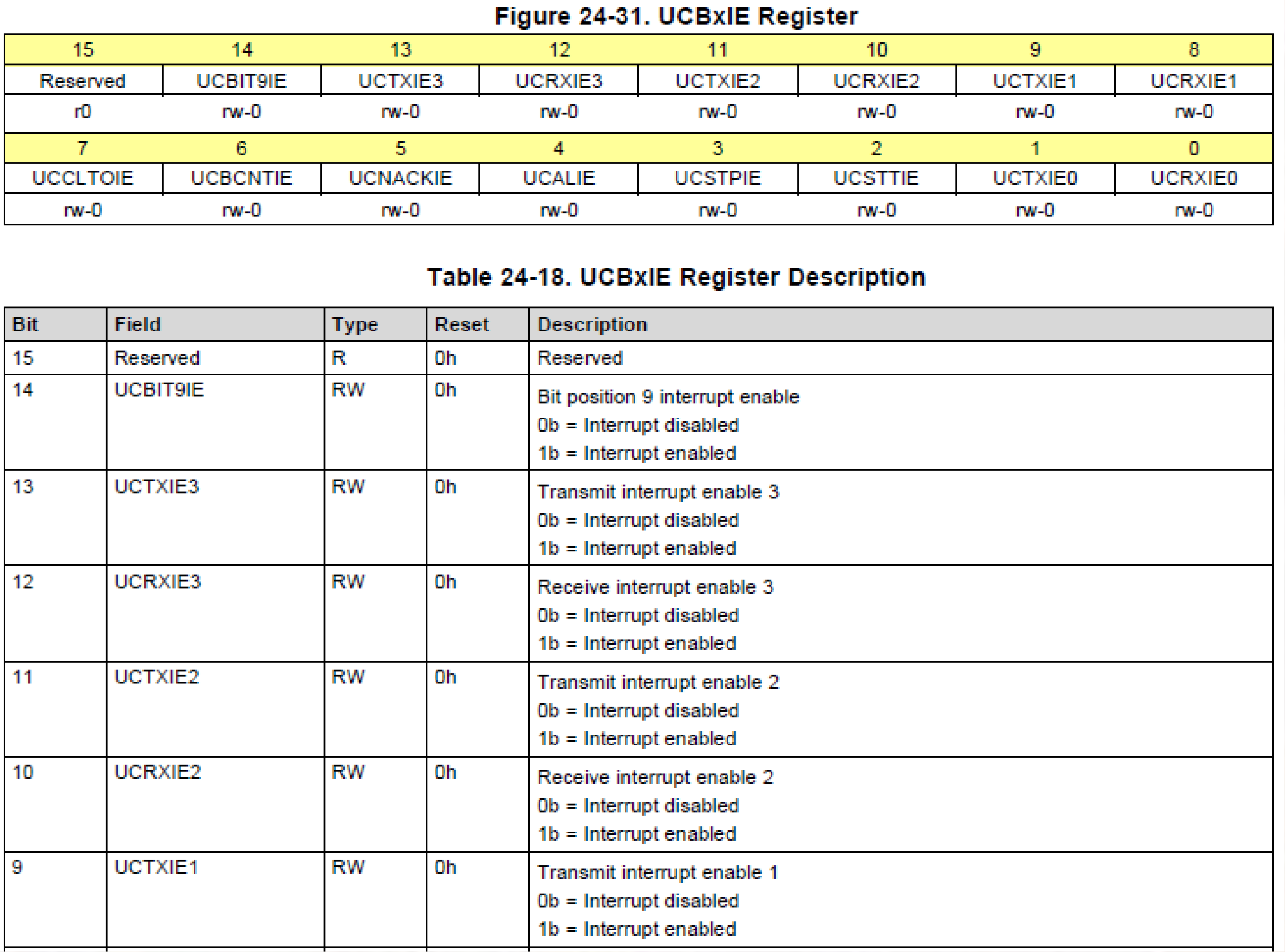


Figure 24-31. UCBxIE Register

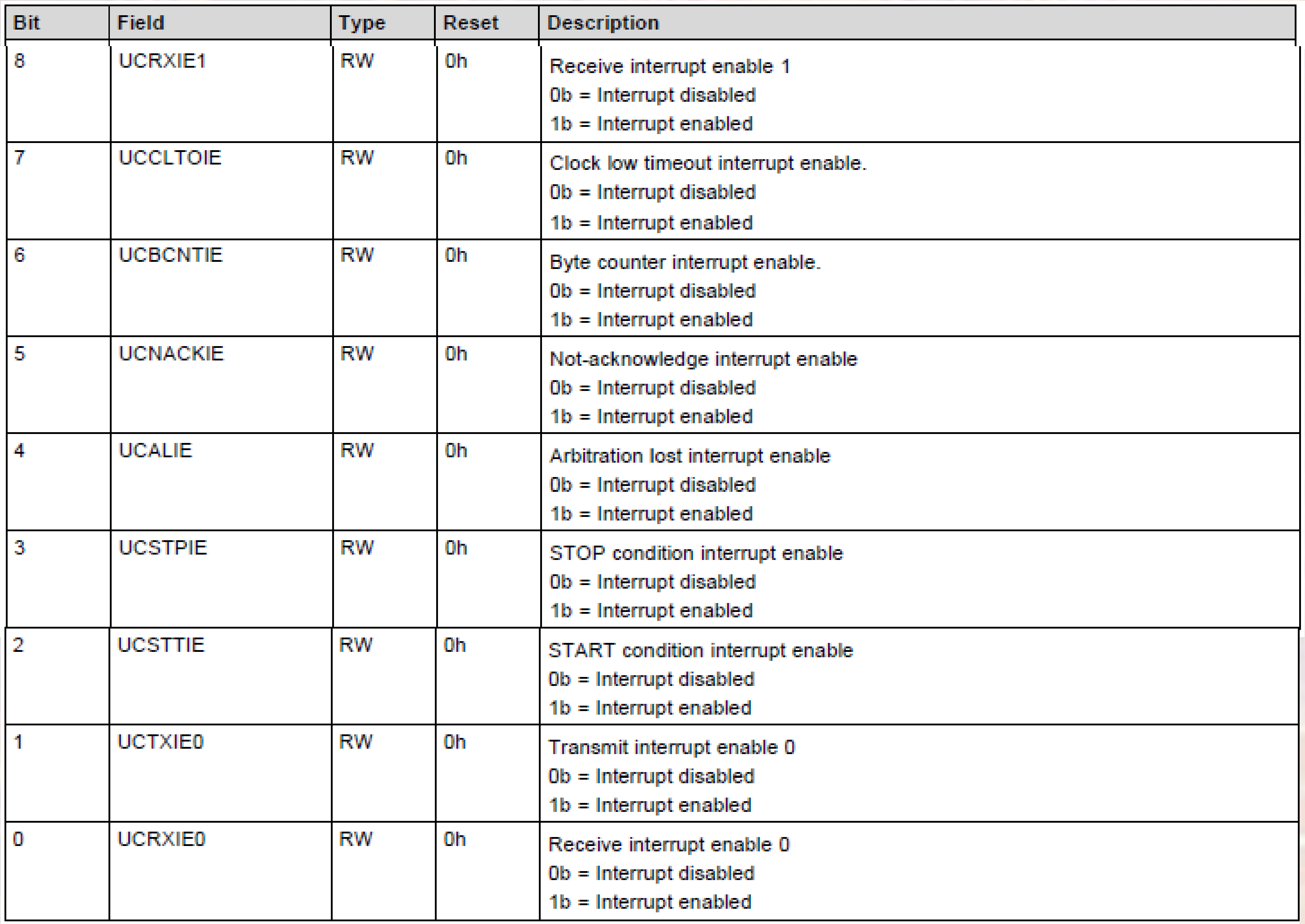
15	14	13	12	11	10	9	8
Reserved	UCBIT9IE	UCTXIE3	UCRXIE3	UCTXIE2	UCRXIE2	UCTXIE1	UCRXIE1
r0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0
7	6	5	4	3	2	1	0
UCCLTOIE	UCBCNTIE	UCNACKIE	UCALIE	UCSTPIE	UCSTTIE	UCTXIE0	UCRXIE0
rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0

Table 24-18. UCBxIE Register Description

Bit	Field	Type	Reset	Description
15	Reserved	R	0h	Reserved
14	UCBIT9IE	RW	0h	Bit position 9 interrupt enable 0b = Interrupt disabled 1b = Interrupt enabled
13	UCTXIE3	RW	0h	Transmit interrupt enable 3 0b = Interrupt disabled 1b = Interrupt enabled
12	UCRXIE3	RW	0h	Receive interrupt enable 3 0b = Interrupt disabled 1b = Interrupt enabled
11	UCTXIE2	RW	0h	Transmit interrupt enable 2 0b = Interrupt disabled 1b = Interrupt enabled
10	UCRXIE2	RW	0h	Receive interrupt enable 2 0b = Interrupt disabled 1b = Interrupt enabled
9	UCTXIE1	RW	0h	Transmit interrupt enable 1 0b = Interrupt disabled 1b = Interrupt enabled

Two Wire Interface

- MSP432 I2C Interrupt Enable Register



Bit	Field	Type	Reset	Description
8	UCRXIE1	RW	0h	Receive interrupt enable 1 0b = Interrupt disabled 1b = Interrupt enabled
7	UCCLTOIE	RW	0h	Clock low timeout interrupt enable. 0b = Interrupt disabled 1b = Interrupt enabled
6	UCBCNTIE	RW	0h	Byte counter interrupt enable. 0b = Interrupt disabled 1b = Interrupt enabled
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	UCTXIE0	RW	0h	Transmit interrupt enable 0 0b = Interrupt disabled 1b = Interrupt enabled
0	UCRXIE0	RW	0h	Receive interrupt enable 0 0b = Interrupt disabled 1b = Interrupt enabled

Two Wire Interface

- MSP432 I2C Interrupt Flag Register

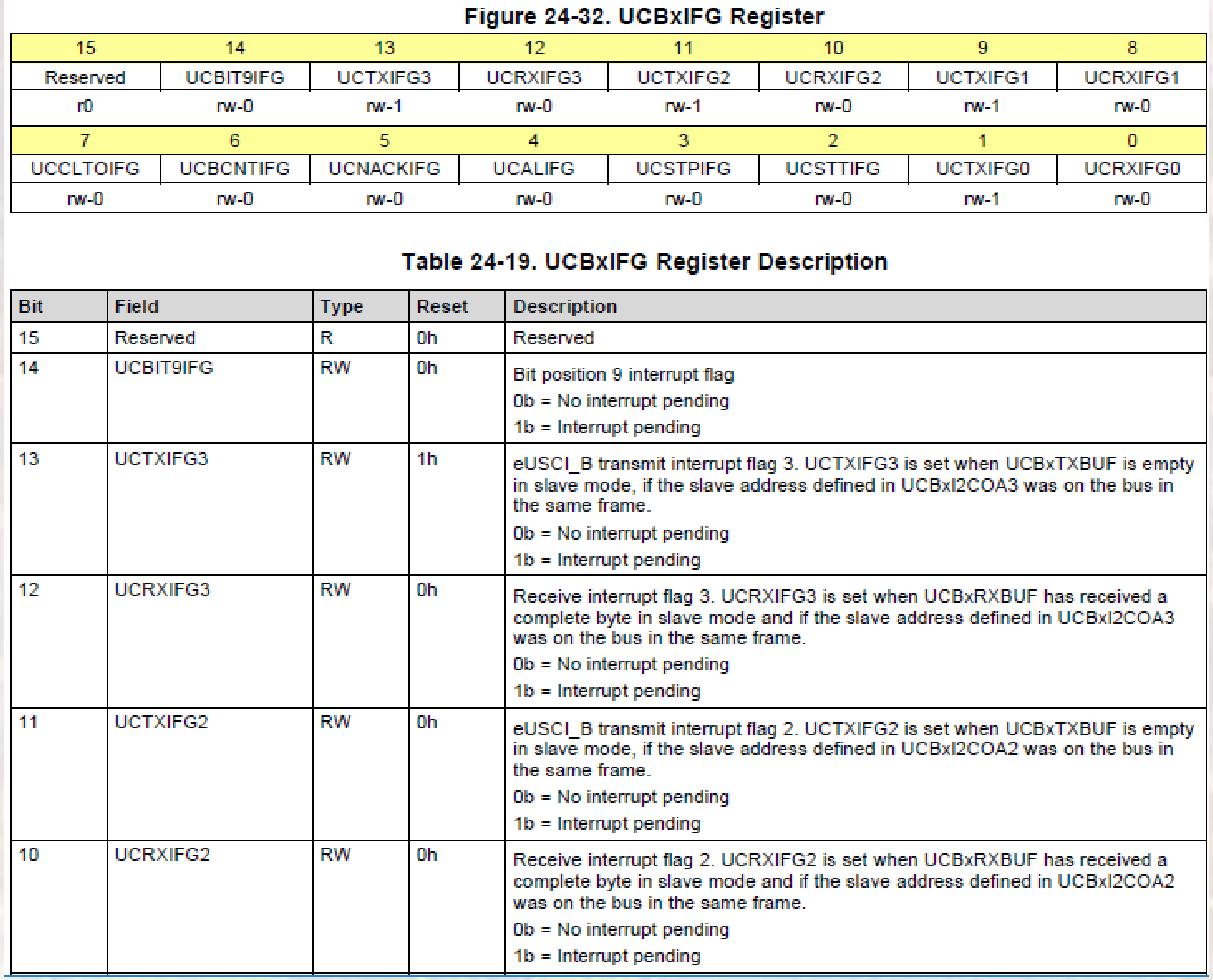


Figure 24-32. UCBxIFG Register

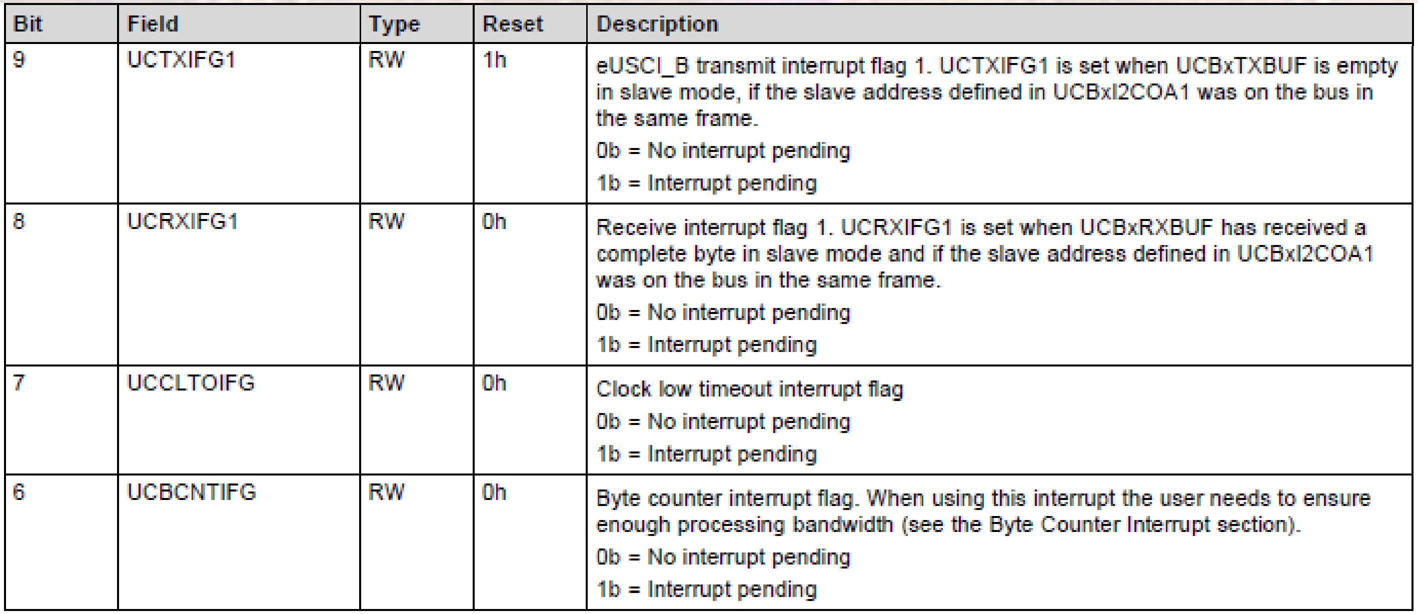
15	14	13	12	11	10	9	8
Reserved	UCBIT9IFG	UCTXIFG3	UCRXIFG3	UCTXIFG2	UCRXIFG2	UCTXIFG1	UCRXIFG1
r0	rw-0	rw-1	rw-0	rw-1	rw-0	rw-1	rw-0
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
UCCLTOIFG	UCBCNTIFG	UCNACKIFG	UCALIFG	UCSTPIFG	UCSTTIFG	UCTXIFG0	UCRXIFG0
rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-1	rw-0

Table 24-19. UCBxIFG Register Description

Bit	Field	Type	Reset	Description
15	Reserved	R	0h	Reserved
14	UCBIT9IFG	RW	0h	Bit position 9 interrupt flag 0b = No interrupt pending 1b = Interrupt pending
13	UCTXIFG3	RW	1h	eUSCI_B transmit interrupt flag 3. UCTXIFG3 is set when UCBxTXBUF is empty in slave mode, if the slave address defined in UCBxI2COA3 was on the bus in the same frame. 0b = No interrupt pending 1b = Interrupt pending
12	UCRXIFG3	RW	0h	Receive interrupt flag 3. UCRXIFG3 is set when UCBxRXBUF has received a complete byte in slave mode and if the slave address defined in UCBxI2COA3 was on the bus in the same frame. 0b = No interrupt pending 1b = Interrupt pending
11	UCTXIFG2	RW	0h	eUSCI_B transmit interrupt flag 2. UCTXIFG2 is set when UCBxTXBUF is empty in slave mode, if the slave address defined in UCBxI2COA2 was on the bus in the same frame. 0b = No interrupt pending 1b = Interrupt pending
10	UCRXIFG2	RW	0h	Receive interrupt flag 2. UCRXIFG2 is set when UCBxRXBUF has received a complete byte in slave mode and if the slave address defined in UCBxI2COA2 was on the bus in the same frame. 0b = No interrupt pending 1b = Interrupt pending

Two Wire Interface

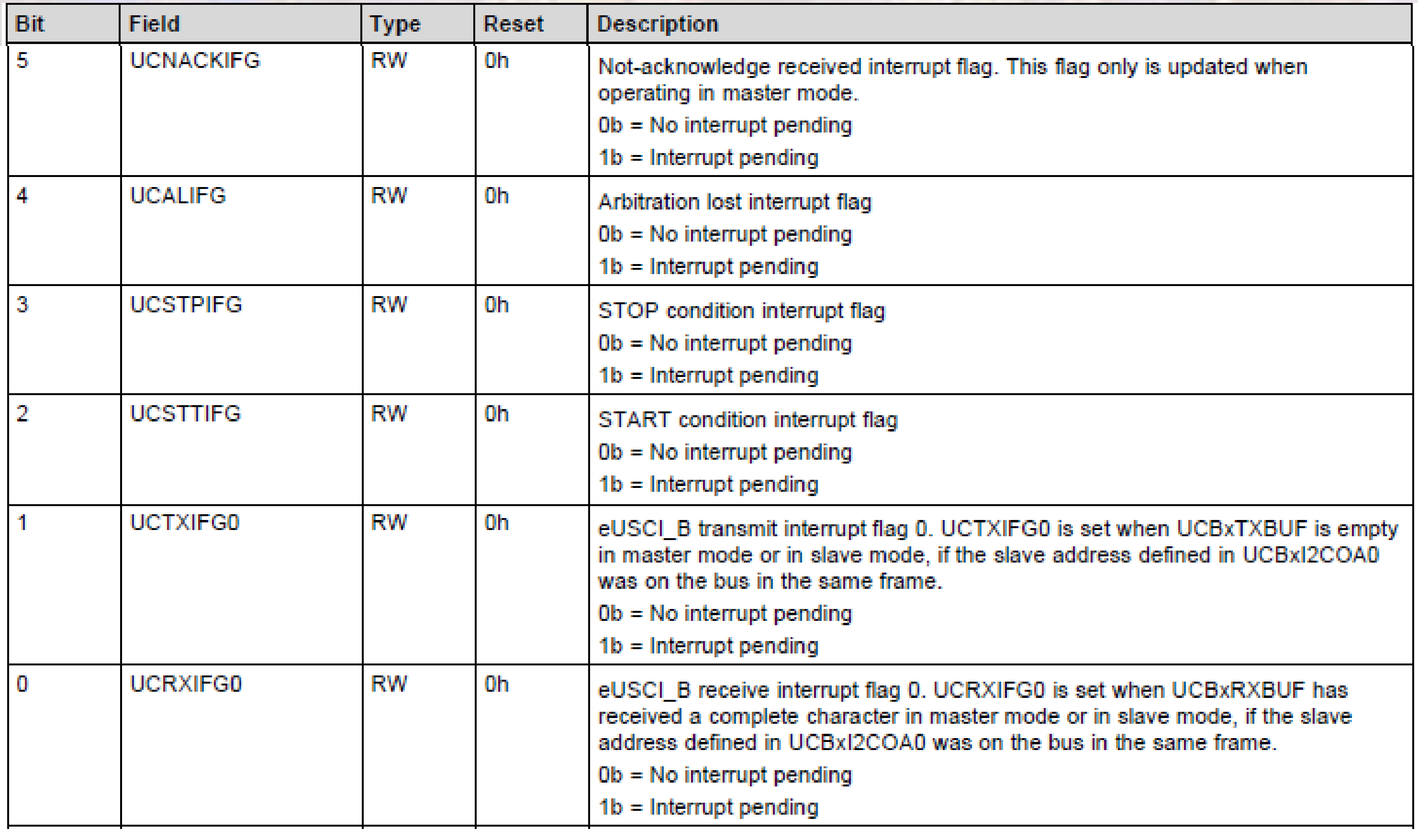
- MSP432 I2C Interrupt Flag Register



Bit	Field	Type	Reset	Description
9	UCTXIFG1	RW	1h	eUSCI_B transmit interrupt flag 1. UCTXIFG1 is set when UCBxTXBUF is empty in slave mode, if the slave address defined in UCBxI2COA1 was on the bus in the same frame. 0b = No interrupt pending 1b = Interrupt pending
8	UCRXIFG1	RW	0h	Receive interrupt flag 1. UCRXIFG1 is set when UCBxRXBUF has received a complete byte in slave mode and if the slave address defined in UCBxI2COA1 was on the bus in the same frame. 0b = No interrupt pending 1b = Interrupt pending
7	UCCLTOIFG	RW	0h	Clock low timeout interrupt flag 0b = No interrupt pending 1b = Interrupt pending
6	UCBCNTIFG	RW	0h	Byte counter interrupt flag. When using this interrupt the user needs to ensure enough processing bandwidth (see the Byte Counter Interrupt section). 0b = No interrupt pending 1b = Interrupt pending

Two Wire Interface

- MSP432 I2C Interrupt Flag Register



Bit	Field	Type	Reset	Description
5	UCNACKIFG	RW	0h	Not-acknowledge received interrupt flag. This flag only is updated when operating in master mode. 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 0b = No interrupt pending 1b = Interrupt pending
2	UCSTTIFG	RW	0h	START condition interrupt flag 0b = No interrupt pending 1b = Interrupt pending
1	UCTXIFG0	RW	0h	eUSCI_B transmit interrupt flag 0. UCTXIFG0 is set when UCBxTXBUF is empty in master mode or in slave mode, if the slave address defined in UCBxI2COA0 was on the bus in the same frame. 0b = No interrupt pending 1b = Interrupt pending
0	UCRXIFG0	RW	0h	eUSCI_B receive interrupt flag 0. UCRXIFG0 is set when UCBxRXBUF has received a complete character in master mode or in slave mode, if the slave address defined in UCBxI2COA0 was on the bus in the same frame. 0b = No interrupt pending 1b = Interrupt pending

Two Wire Interface

- MSP432 I2C Interrupt Vector Register

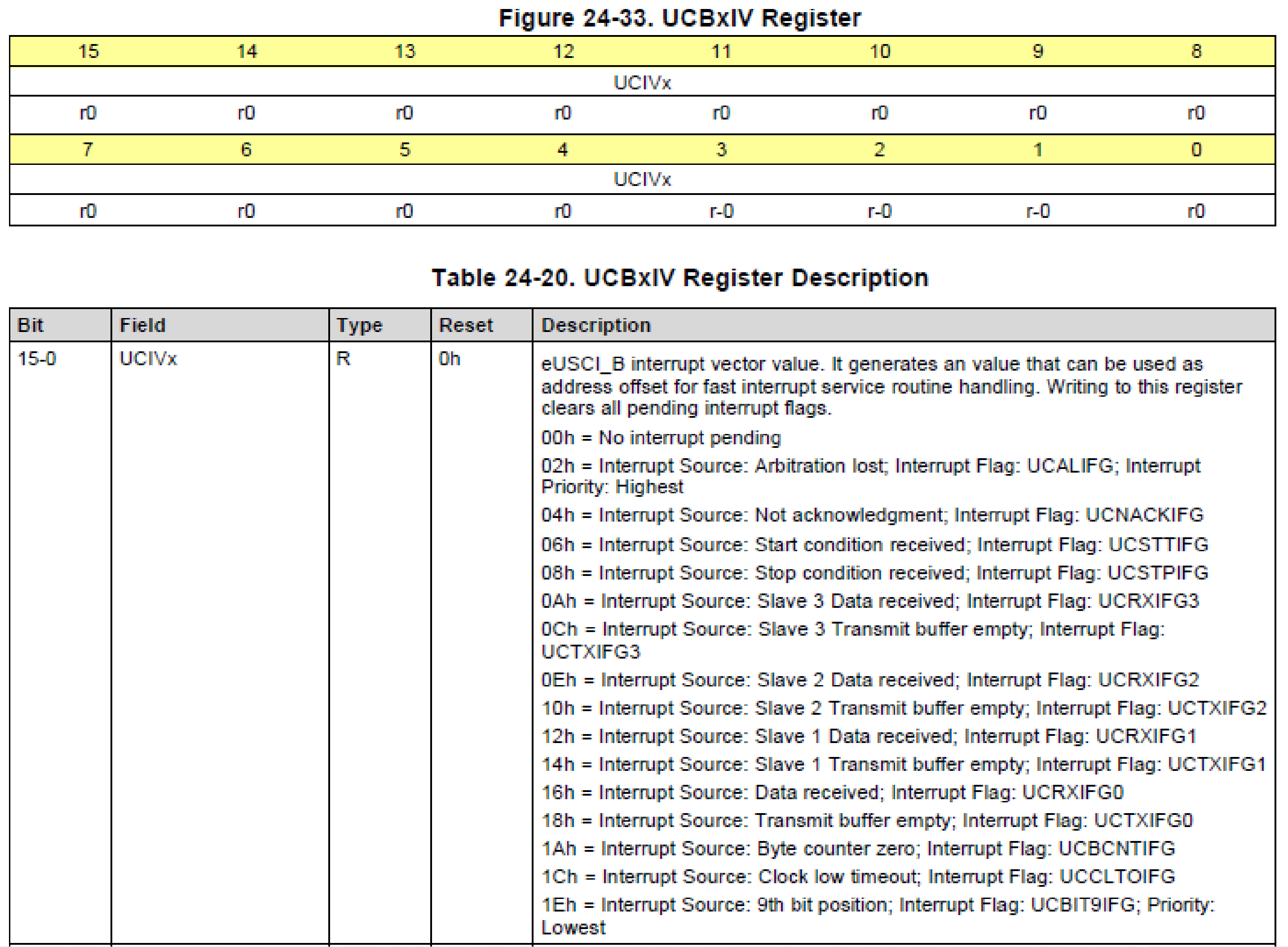


Figure 24-33. UCBxIV Register

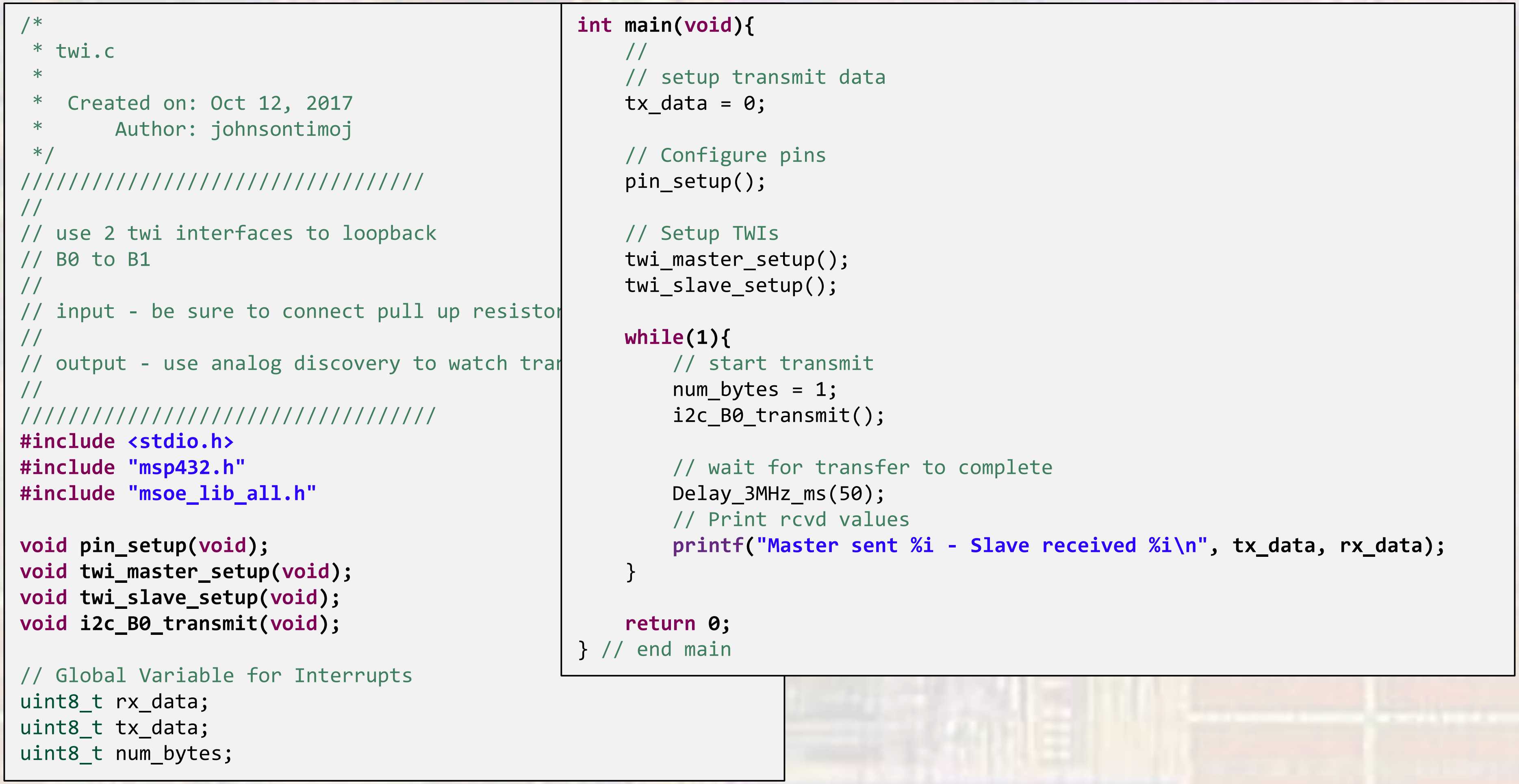
15	14	13	12	11	10	9	8
UCIVx							
r0	r0	r0	r0	r0	r0	r0	r0
7	6	5	4	3	2	1	0
UCIVx							
r0	r0	r0	r0	r-0	r-0	r-0	r0

Table 24-20. UCBxIV Register Description

Bit	Field	Type	Reset	Description
15-0	UCIVx	R	0h	eUSCI_B interrupt vector value. It generates an value that can be used as address offset for fast interrupt service routine handling. Writing to this register clears all pending interrupt flags. 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: Slave 3 Data received; Interrupt Flag: UCRXIFG3 0Ch = Interrupt Source: Slave 3 Transmit buffer empty; Interrupt Flag: UCTXIFG3 0Eh = Interrupt Source: Slave 2 Data received; Interrupt Flag: UCRXIFG2 10h = Interrupt Source: Slave 2 Transmit buffer empty; Interrupt Flag: UCTXIFG2 12h = Interrupt Source: Slave 1 Data received; Interrupt Flag: UCRXIFG1 14h = Interrupt Source: Slave 1 Transmit buffer empty; Interrupt Flag: UCTXIFG1 16h = Interrupt Source: Data received; Interrupt Flag: UCRXIFG0 18h = Interrupt Source: Transmit buffer empty; Interrupt Flag: UCTXIFG0 1Ah = Interrupt Source: Byte counter zero; Interrupt Flag: UCBCNTIFG 1Ch = Interrupt Source: Clock low timeout; Interrupt Flag: UCCLTOIFG 1Eh = Interrupt Source: 9th bit position; Interrupt Flag: UCBIT9IFG; Priority: Lowest

Two Wire Interface

- Loop Back Example
- B0 to B1



```
/*
 * twi.c
 *
 *  Created on: Oct 12, 2017
 *      Author: johnsontimoj
 */
//////////////////////////////////
//
// use 2 twi interfaces to loopback
// B0 to B1
//
// input - be sure to connect pull up resistor
//
// output - use analog discovery to watch trar
//
///////////////////////////////////
#include <stdio.h>
#include "msp432.h"
#include "msoe_lib_all.h"

void pin_setup(void);
void twi_master_setup(void);
void twi_slave_setup(void);
void i2c_B0_transmit(void);

// Global Variable for Interrupts
uint8_t rx_data;
uint8_t tx_data;
uint8_t num_bytes;
```

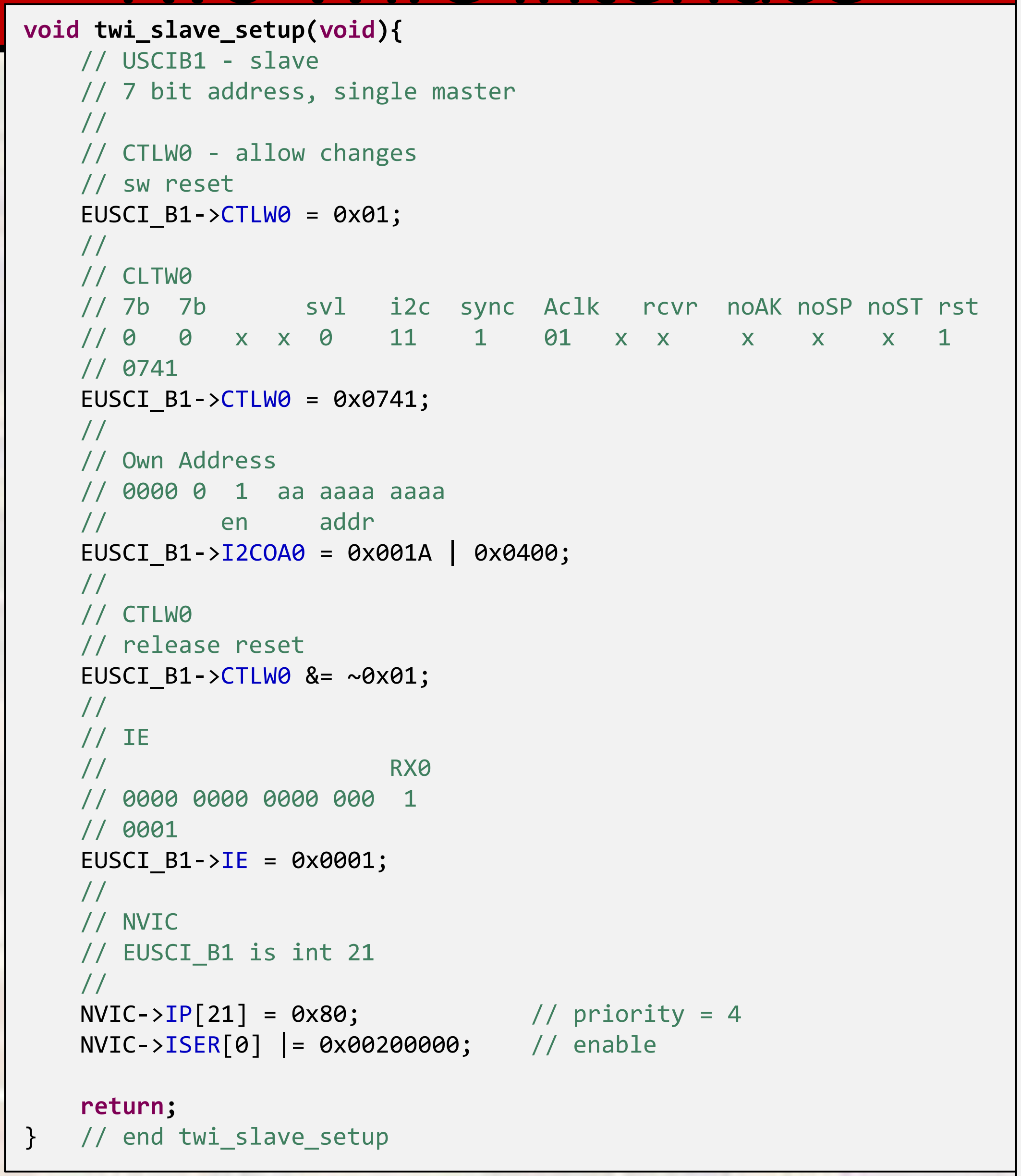
```
int main(void){
    //
    // setup transmit data
    tx_data = 0;

    // Configure pins
    pin_setup();

    // Setup TWIs
    twi_master_setup();
    twi_slave_setup();

    while(1){
        // start transmit
        num_bytes = 1;
        i2c_B0_transmit();

        // wait for transfer to complete
        Delay_3MHz_ms(50);
        // Print rcvd values
        printf("Master sent %i - Slave received %i\n", tx_data, rx_data);
    }

    return 0;
} // end main
```

Two Wire Interface

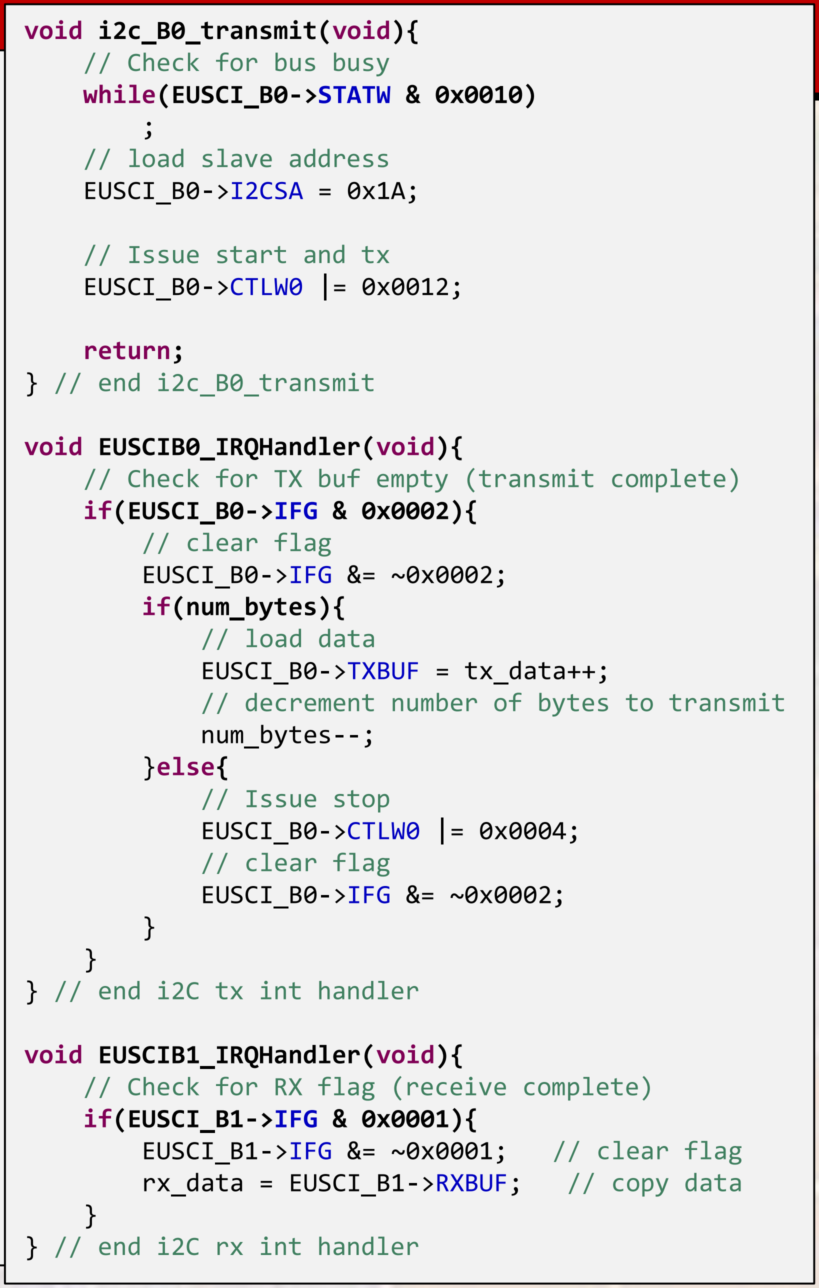
```
void pin_setup(void){
    // Setup USCIB0 as master
    // Setup USCIB1 as slave
    // Transfer data from Master to slave in a while loop
    //
    // USCIB0 --> P1.6-SDA, P1.7-SCK
    // USCIB1 --> P6.4-SDA, P6.5-SCK
    //
    P1->SEL0 |= 0xC0;   // Set to USC mode
    P1->SEL1 &= ~0xC0;  // 0-1 mode
    P6->SEL0 |= 0x30;
    P6->SEL1 &= ~0x30;

    // Nothing else needed - the module sets directions
    return;
} // end pin_setup
```

```
void twi_master_setup(void){
    // USCIB0 - master
    // 7 bit address, single master
    //
    // CTLW0 - allow changes
    // sw reset
    EUSCI_B0->CTLW0 = 0x0001;
    //
    // CLTW0
    //7b  7b  1M   mstr  i2c  sync  Aclk    txr  noAK noSP noST rst
    // 0   0   0  x  1    11    1    01   x  1     0    0    0   1
    // 0F41
    EUSCI_B0->CTLW0 = 0x0F51;
    //
    // BRW
    //  /64 - 0000 0000 0100 0000
    // 0040
    EUSCI_B0->BRW = 0x0040;
    //
    // CTLW0
    // release reset
    EUSCI_B0->CTLW0 &= ~0x01;
    //
    // IE
    // 0000 0000 00 1  0 00 1  0
    //              nk     TX0
    // 0022
    EUSCI_B0->IE = 0x0002;
    //
    // NVIC
    // EUSCI_B0 is int 20
    //
    NVIC->IP[20] = 0x60;            // priority = 3
    NVIC->ISER[0] |= 0x00100000;    // enable

    return;
}   // end twi_master_setup
```

Two Wire Interface



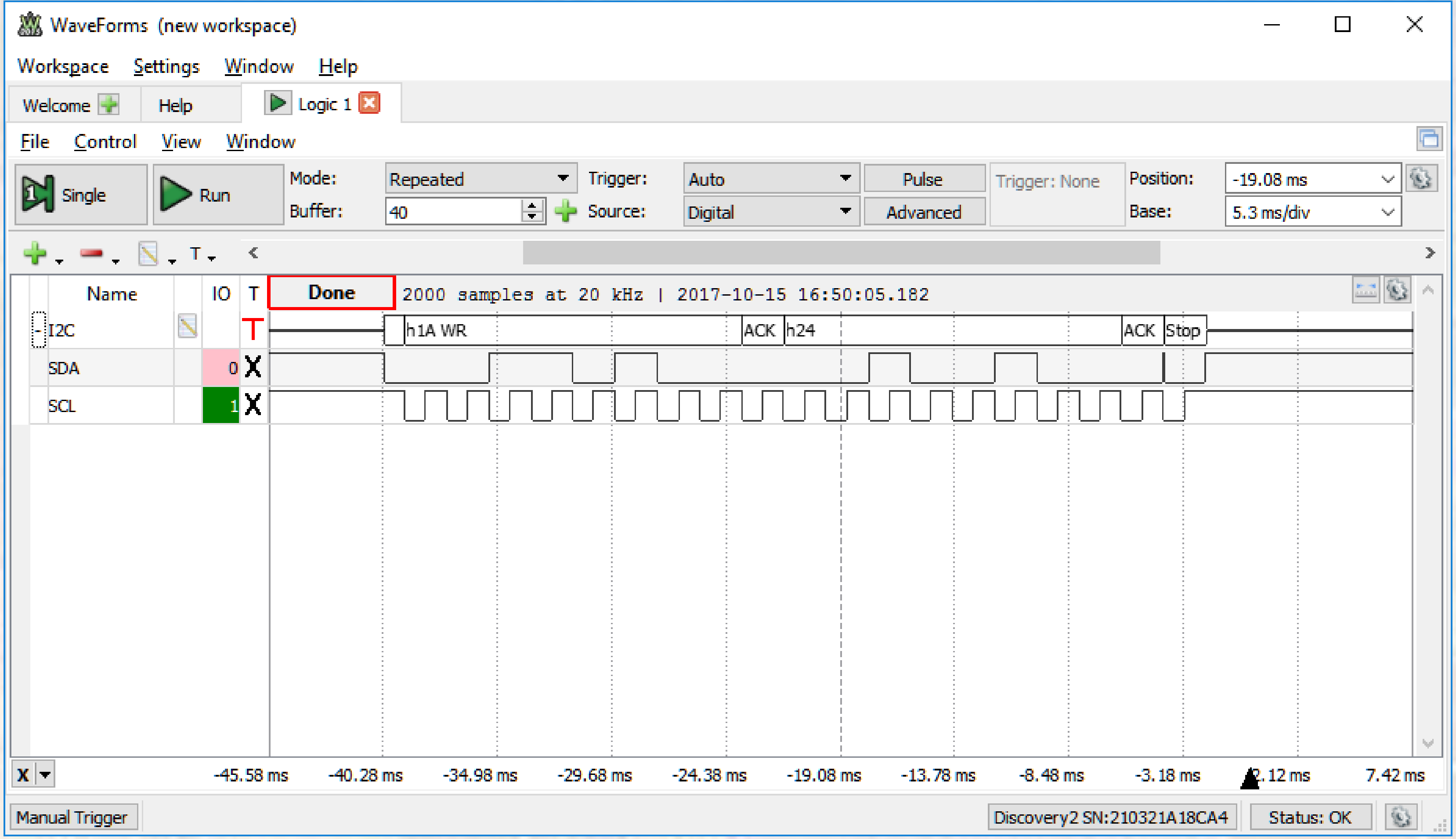
```c
void twi_slave_setup(void){
    // USCIB1 - slave
    // 7 bit address, single master
    //
    // CTLW0 - allow changes
    // sw reset
    EUSCI_B1->CTLW0 = 0x01;
    //
    // CLTW0
    // 7b  7b       svl   i2c  sync  Aclk   rcvr  noAK noSP noST rst
    // 0   0   x  x  0    11    1    01   x  x     x    x    x   1
    // 0741
    EUSCI_B1->CTLW0 = 0x0741;
    //
    // Own Address
    // 0000 0  1  aa aaaa aaaa
    //        en      addr
    EUSCI_B1->I2COA0 = 0x001A | 0x0400;
    //
    // CTLW0
    // release reset
    EUSCI_B1->CTLW0 &= ~0x01;
    //
    // IE
    //                    RX0
    // 0000 0000 0000 000  1
    // 0001
    EUSCI_B1->IE = 0x0001;
    //
    // NVIC
    // EUSCI_B1 is int 21
    //
    NVIC->IP[21] = 0x80;            // priority = 4
    NVIC->ISER[0] |= 0x00200000;    // enable

    return;
}   // end twi_slave_setup
```



```c
void i2c_B0_transmit(void){
    // Check for bus busy
    while(EUSCI_B0->STATW & 0x0010)
        ;
    // load slave address
    EUSCI_B0->I2CSA = 0x1A;

    // Issue start and tx
    EUSCI_B0->CTLW0 |= 0x0012;

    return;
} // end i2c_B0_transmit

void EUSCIB0_IRQHandler(void){
    // Check for TX buf empty (transmit complete)
    if(EUSCI_B0->IFG & 0x0002){
        // clear flag
        EUSCI_B0->IFG &= ~0x0002;
        if(num_bytes){
            // load data
            EUSCI_B0->TXBUF = tx_data++;
            // decrement number of bytes to transmit
            num_bytes--;
        }else{
            // Issue stop
            EUSCI_B0->CTLW0 |= 0x0004;
            // clear flag
            EUSCI_B0->IFG &= ~0x0002;
        }
    }
} // end i2C tx int handler

void EUSCIB1_IRQHandler(void){
    // Check for RX flag (receive complete)
    if(EUSCI_B1->IFG & 0x0001){
        EUSCI_B1->IFG &= ~0x0001;   // clear flag
        rx_data = EUSCI_B1->RXBUF;   // copy data
    }
} // end i2C rx int handler
```

Two Wire Interface

- MSP432 I2C
  - 1 byte transfer



Mode: Repeated

Buffer: 40

Trigger: Auto

Source: Digital

Trigger: None

Position: -19.08 ms

Base: 5.3 ms/div

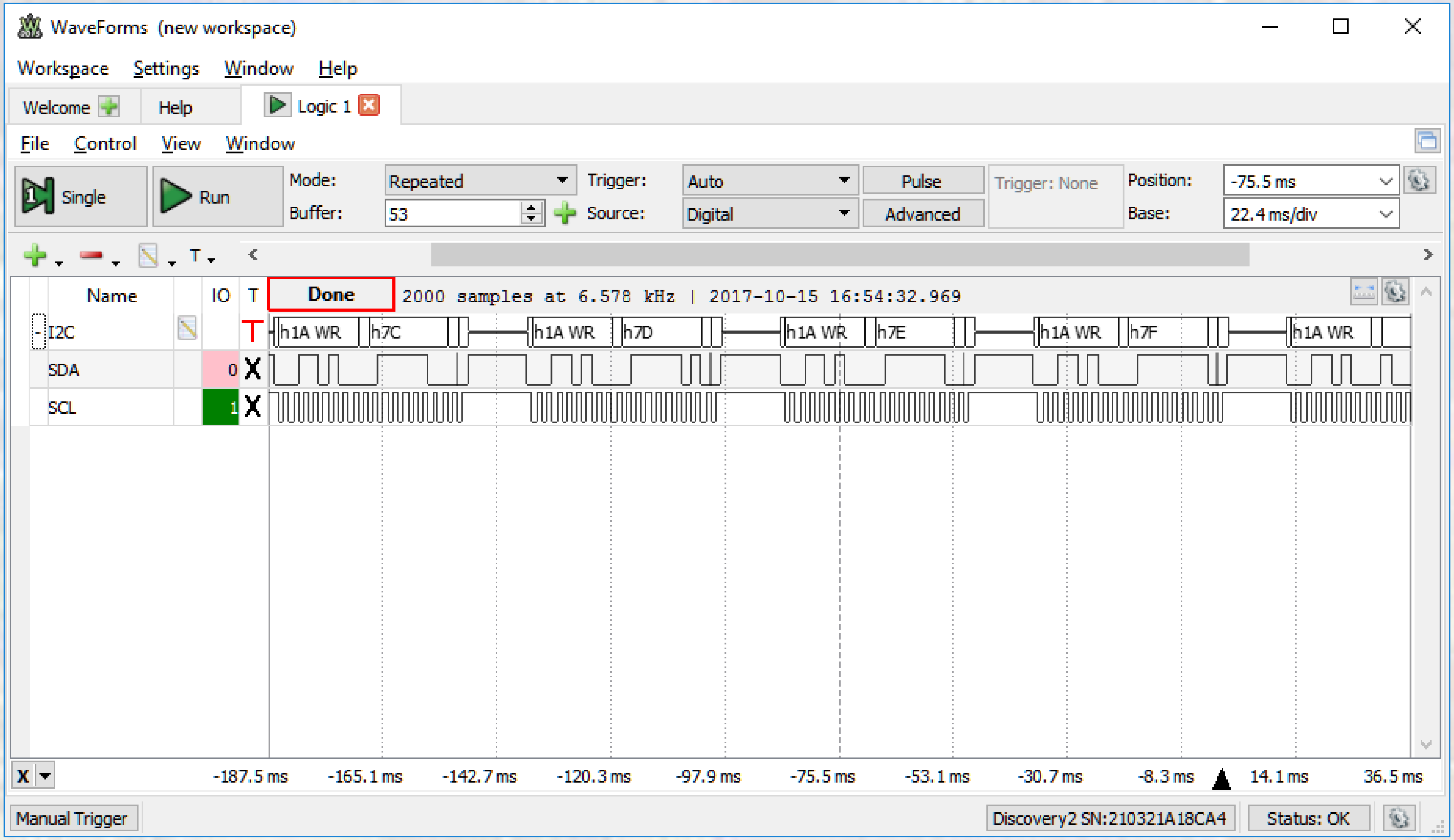
Done 2000 samples at 20 kHz | 2017-10-15 16:50:05.182

Discovery2 SN:210321A18CA4

Status: OK

Two Wire Interface

- MSP432 I2C
  - Multiple 1 byte transfers



Mode: Repeated
Buffer: 53
Trigger: Auto
Source: Digital
Trigger: None
Position: -75.5 ms
Base: 22.4 ms/div

Done 2000 samples at 6.578 kHz | 2017-10-15 16:54:32.969

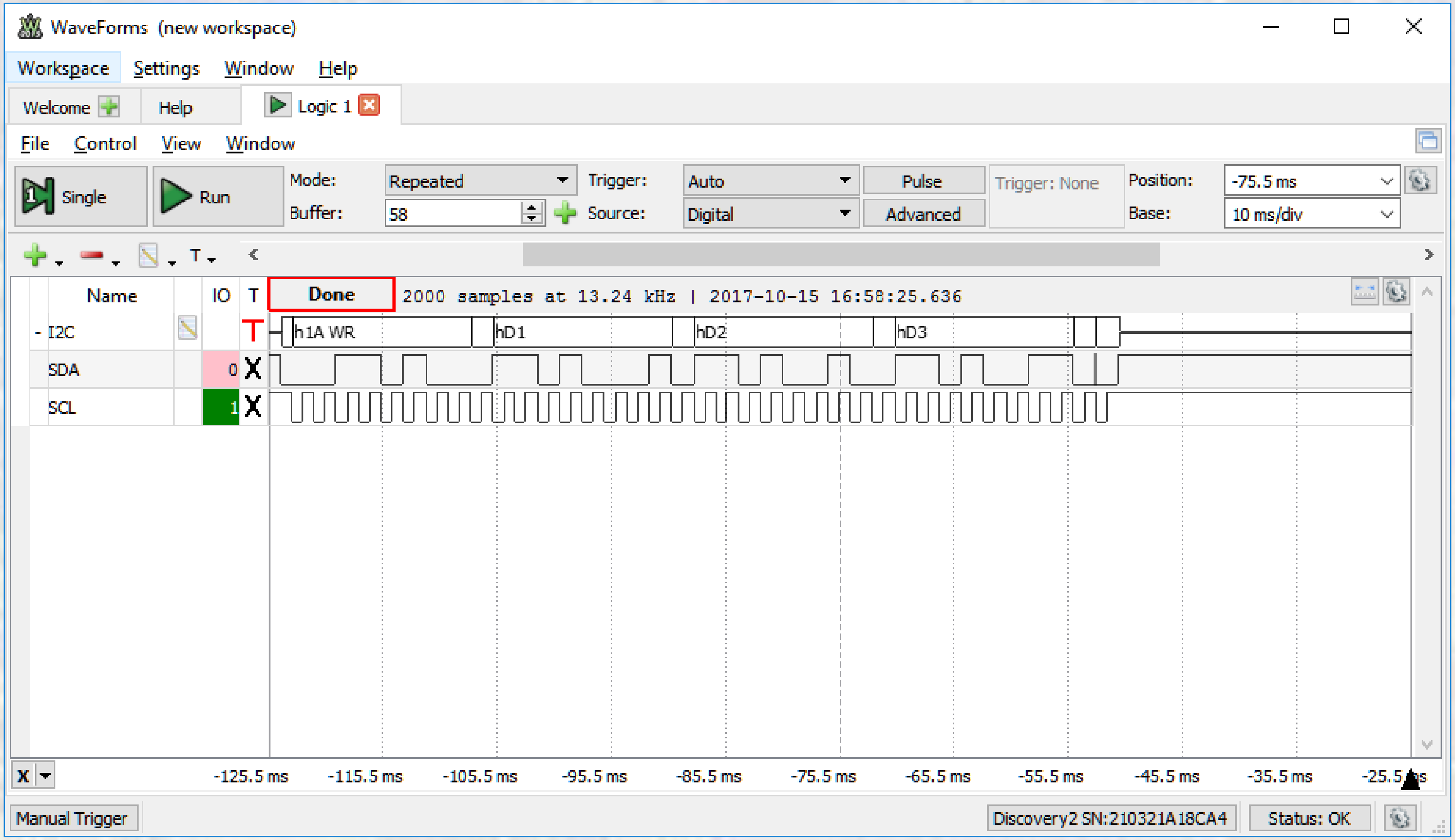
h1A WR h7C h1A WR h7D h1A WR h7E h1A WR h7F h1A WR

-187.5 ms -165.1 ms -142.7 ms -120.3 ms -97.9 ms -75.5 ms -53.1 ms -30.7 ms -8.3 ms 14.1 ms 36.5 ms

Discovery2 SN:210321A18CA4 Status: OK

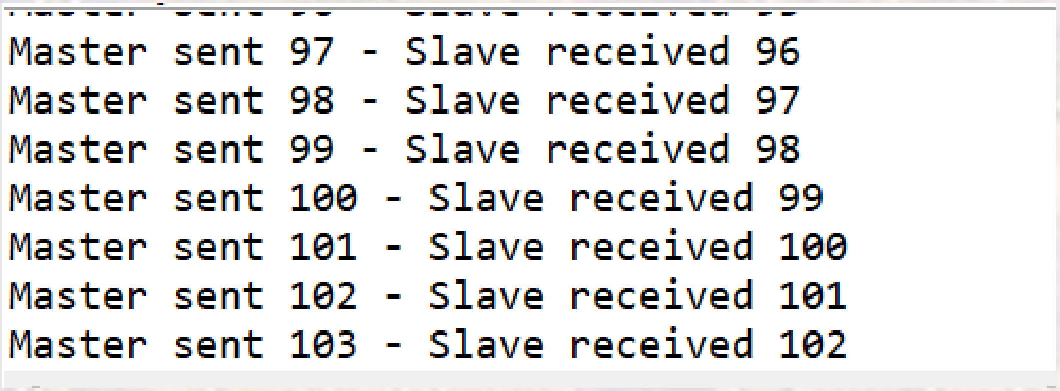
Two Wire Interface

- MSP432 I2C
  - 3 byte transfer



Two Wire Interface

- MSP432 I2C
  - 1 byte transfer



```
Master sent 97 - Slave received 96
Master sent 98 - Slave received 97
Master sent 99 - Slave received 98
Master sent 100 - Slave received 99
Master sent 101 - Slave received 100
Master sent 102 - Slave received 101
Master sent 103 - Slave received 102
```

This IS correct – so why are the values not the same????