

# ***TMS470R1x Class II Serial Interface A (C2S1a) Reference Guide***

Literature Number: SPNU218B  
November 2004



## IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

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

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI 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 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. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

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

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

### Products

Amplifiers	<a href="http://amplifier.ti.com">amplifier.ti.com</a>
Data Converters	<a href="http://dataconverter.ti.com">dataconverter.ti.com</a>
DSP	<a href="http://dsp.ti.com">dsp.ti.com</a>
Interface	<a href="http://interface.ti.com">interface.ti.com</a>
Logic	<a href="http://logic.ti.com">logic.ti.com</a>
Power Mgmt	<a href="http://power.ti.com">power.ti.com</a>
Microcontrollers	<a href="http://microcontroller.ti.com">microcontroller.ti.com</a>

### Applications

Audio	<a href="http://www.ti.com/audio">www.ti.com/audio</a>
Automotive	<a href="http://www.ti.com/automotive">www.ti.com/automotive</a>
Broadband	<a href="http://www.ti.com/broadband">www.ti.com/broadband</a>
Digital Control	<a href="http://www.ti.com/digitalcontrol">www.ti.com/digitalcontrol</a>
Military	<a href="http://www.ti.com/military">www.ti.com/military</a>
Optical Networking	<a href="http://www.ti.com/opticalnetwork">www.ti.com/opticalnetwork</a>
Security	<a href="http://www.ti.com/security">www.ti.com/security</a>
Telephony	<a href="http://www.ti.com/telephony">www.ti.com/telephony</a>
Video & Imaging	<a href="http://www.ti.com/video">www.ti.com/video</a>
Wireless	<a href="http://www.ti.com/wireless">www.ti.com/wireless</a>

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

### REVISION HISTORY

REVISION	DATE	NOTES
B	11/04	Removed author names; released to mass market
A	9/02	Converted to a stand-alone book
*	1/02	<i>Pre-dates revision history</i>



# Contents

---

---

---

<b>1</b>	<b>Overview</b> .....	<b>2</b>
<b>2</b>	<b>Functional Description of the C2Sla</b> .....	<b>3</b>
2.1	C2Sla Internal Registers .....	4
<b>3</b>	<b>Data Format</b> .....	<b>6</b>
<b>4</b>	<b>Transmitting C2Sla Messages</b> .....	<b>11</b>
4.1	Transmitting Non-IFR Messages .....	13
4.2	Transmitting In-Frame Response (IFR) Messages .....	14
4.3	Transmitting BREAK Messages .....	15
4.4	Transmission Arbitration .....	15
4.5	Byte Boundary Loss of Arbitration when Transmitting .....	16
4.6	Bus Error Conditions .....	16
4.6.1	Short-to-Voltage .....	17
4.6.2	Short-to-Ground .....	17
4.6.3	Open Bus .....	18
4.6.4	Other Bus Errors Not Discussed .....	18
<b>5</b>	<b>Receiving C2Sla Messages</b> .....	<b>19</b>
5.1	Receiving Normal Messages .....	19
5.2	Receiving In-Frame Response (IFR) Messages .....	19
5.3	Receiving BREAK Messages .....	20
5.4	Receiving Digital Filters .....	20
<b>6</b>	<b>Interrupts</b> .....	<b>21</b>
6.1	Proper Handling of Events .....	21
<b>7</b>	<b>General Purpose I/O</b> .....	<b>23</b>
<b>8</b>	<b>DMA Interface</b> .....	<b>24</b>
8.1	DMA Transactions .....	24
8.2	Non-DMA Transactions .....	24
8.3	4X Mode .....	25
8.4	Low Power Mode .....	25
8.5	Emulation Mode .....	26
8.6	Calibration Mode .....	27
<b>9</b>	<b>C2Sla Internal Registers</b> .....	<b>29</b>
9.1	C2Sla Interrupt Status Register (C2SIAISR) .....	29
9.2	C2Sla Interrupt Control Register (C2SIAICR) .....	33
9.3	C2Sla Global Status Register (C2SIAGSR) .....	36
9.4	C2Sla Global Control Register (C2SIAGCR) .....	37

9.5	C2Sla Transmit Data Buffer Register (C2SIATDB).....	40
9.6	C2Sla Completion Code Status Register (C2SIACCSR) .....	41
9.7	C2Sla Peripheral Control Register (C2SIACTR).....	45
9.8	Interface Clock Register (C2SIACLK) .....	47
9.9	C2Sla Transmit Byte Counter (C2SIATBC) .....	48
9.10	C2Sla Pin Control Register 1 (C2SIAPC1) .....	48
9.11	C2Sla Pin Control Register 2 (C2SIAPC2) .....	51
9.12	C2Sla Pin Control Register 3 (C2SIAPC3) .....	51
9.13	C2Sla Receive Data Emulation Buffer Register(C2SIAEMU).....	52
9.14	C2Sla Receive Data Buffer Register (C2SIARDB).....	52
9.15	C2Sla Peripheral Control Register (C2SIACAL).....	53
9.16	C2Sla Peripheral Control Register (C2SIABRK).....	54
<b>10</b>	<b>Timing Examples.....</b>	<b>55</b>
<b>11</b>	<b>History .....</b>	<b>59</b>

# Figures

---

---

---

---

1	C2Sla Pin Connection Diagram . . . . .	3
2	C2Sla Block Diagram . . . . .	4
3	Simplest 1 Byte Message . . . . .	7
4	Typical C2Sla Data Frame Format Without In-Frame Response . . . . .	7
5	Typical C2Sla Data Frame Format With In-Frame Response . . . . .	7
6	Byte Transmission . . . . .	12
7	Simple Transmit Example . . . . .	55
8	Send Message and Receive IFR Example . . . . .	56
9	Long Break Timing Example . . . . .	57
10	Loss of Arbitration Timing Example . . . . .	58

# Tables

---

---

---

---

- 1 C2Sla Internal Registers ..... 5
- 2 C2Sla Message Time Duration ..... 8
- 3 C2Sla Control Register File Used With the TMS470 CPU ..... 29
- 4 Completion Code Descriptions ..... 43

# Class II Serial Interface A (C2Sla)

---

---

---

The class II serial interface A (C2Sla) is a communication module used for transmitting and receiving data over a multi-master network. The C2Sla module is the interface from the digital logic of the TMS470R1x family of microcontrollers to an external, analog interface chip. Class II communications follow the J1850 Class B protocol established by the Society of Automotive Engineers (SAE).

Topic	Page
1 Overview .....	2
2 Functional Description of the C2Sla .....	3
3 Data Format .....	6
4 Transmitting C2Sla Messages .....	11
5 Receiving C2Sla Messages .....	19
6 Interrupts .....	21
7 General Purpose I/O .....	23
8 DMA Interface .....	24
9 C2Sla Internal Registers .....	29

## 1 Overview

The class II serial interface (C2Sla) is designed to handle all of the class II digital logic functions and operations between the host CPU and the class II bus interface (analog interface).

The C2Sla module has the following features:

- Three external device pins
  - C2SlaTX (C2Sla transmit data output)
  - C2SlaRX (C2Sla receive data input)
  - C2SlaLPN (C2Sla loop-back enable)
- Two selectable data rates
  - Normal mode: 10.4 Kips
  - 4X mode: 41.6 Kips
- Multiple error detection flags
  - Break detect error
  - Overrun error
  - Incomplete byte error
  - Bit timing error causing data to be corrupted
  - CRC error
  - Transmission errors
  - Short to ground errors
- Double-buffered receive and transmit functions
- Separate transmitter and receiver interrupts that can be interrupt driven or polled through the use of status flags
- Enable bits for interrupts
- Automatic CRC generation
- Low-power mode
- Automatic or manual calibration with external analog interface device.

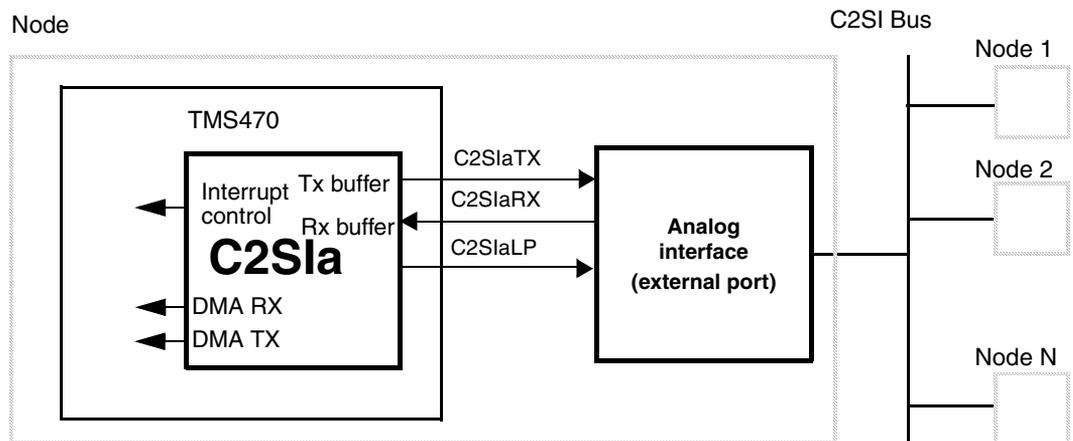
## 2 Functional Description of the C2Sla

The C2Sla is contained within the TMS470. The C2Sla connects externally to an analog interface chip which is outside of the TMS470 as shown in Figure 1. This analog interface chip is required and acts as a buffer between the digital signals of the C2Sla and analog signals recognized by the C2SI bus. The analog interface chip can be implemented with either a commercially available chip, such as the Harris HIP7020, or can be designed by the end-user.

The C2Sla is a serial interface that supports SAE J1850 class B protocol with selectable data transmissions at either normal or 4X operational mode. The C2Sla's receiver and transmitter are double buffered, and each has its own interrupt flags and bits.

Each individual C2Sla on the bus is referred to as a node as shown in Figure 1. For information on the flow of data through the C2Sla to bus interface pins and the related control pin registers see section 9.10, *C2Sla Pin Control Register 1 (C2SIAPC1)*, on page 48.

Figure 1. C2Sla Pin Connection Diagram



The C2SI bus is a serial data communications link. The nondestructive contention protocol of the Class II bus requires that there be an active (high) voltage and a passive (low) voltage state of the bus. The function of the analog interface is to actively drive the bus to a high voltage when signaled by the C2Sla, and passively let an RC network pull the bus down to a low voltage. It also monitors the class II data-bus state for received data that is transferred to the C2Sla. The bus is a wired OR arrangement.



Table 1. C2S1a Internal Registers

Address Offset †	Mnemonic	Name	Description	Page
0x00	C2S1AISR	C2S1a Interrupt Status Register	Contains transmit/receive interrupt status flags	29
0x04	C2S1AICR	C2S1a Interrupt Control Register	Contains transmit/receive interrupt enable control bits	33
0x08	C2S1AGSR	C2S1a Global Status Register	Contains bus status flags	36
0x0C	C2S1AGCR	C2S1a Global Control Register	Contains control bits for initiating and controlling transmissions	37
0x10	C2S1ATDB	C2S1a Transmit Data Buffer	Contains data bits to be transmitted out of the C2S1ATXD pin	40
0x14	C2S1ACCSR	C2S1a Completion Code Status Register	Contains read-clear transmit/receive completion status flags	41
0x18	C2S1ACTR	C2S1a Control Register	Contains read/write bits for enabling control functions	45
0x1C	C2S1ACLK	C2S1a Interface Clock Register	Set to the frequency of the interface clock	47
0x20	C2S1ATBC	C2S1a Transmit Byte Counter	Determines the number of bytes to be transmitted	48
0x24	C2S1APCI	C2S1a Pin Control Register 1	Determines if individual pins are used as general I/O or C2S1a function pins	48
0x28	C2S1APC2	C2S1a Pin Control Register 2	Determines the value of the general I/O output	51
0x2C	C2S1APC3	C2S1a Pin Control Register 3	Reflects the value on the pins	51
0x30	C2S1AEMU	C2S1a Emulation Buffer Register	Mirror of C2S1ARDB, but read does not clear interrupt	52
0x34	C2S1ARDB	C2S1a Receive Data Buffer	Contains the current data from the receiver shift register	52
0x38	C2S1ACAL	C2S1a Calibration Register	Contain the calibration constant	53
0x3C	C2S1ABRK	C2S1a Break status Register	Contains the current status of the break register	54

† The actual address of these registers is device specific and CPU specific. See the specific device data sheet to verify the C2S1a register addresses.

### 3 Data Format

The C2S1a receive and transmit data formats, shown in Figure 2 and Figure 3, consist of the components listed below (a detailed description of the individual message components follows after Table 2). For an explanation of the various forms and effects of an in-frame response, refer to SAE J2178/1.

- Start of frame (SOF) period
- Data
- Cyclic redundancy check (CRC) byte
- End of data (EOD)
- Normalization bit when an in-frame response (IFR) protocol is used
- IFR bytes initiated by a responder immediately following the normalization bit when an IFR protocol is used.
- End of data (EOD), end of frame (EOF), and IDLE period

Figure 3. Simplest 1 Byte Message

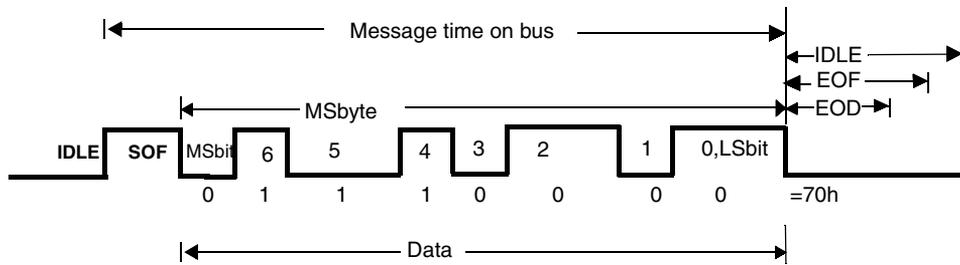


Figure 4. Typical C2S1a Data Frame Format Without In-Frame Response

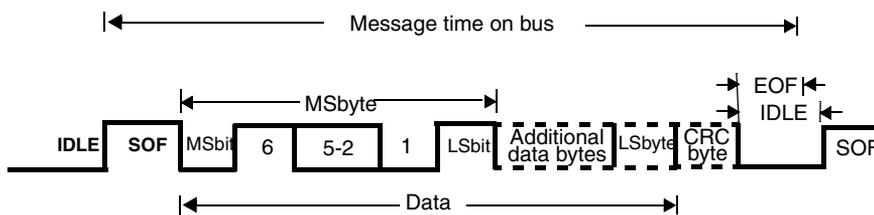


Figure 5. Typical C2S1a Data Frame Format With In-Frame Response

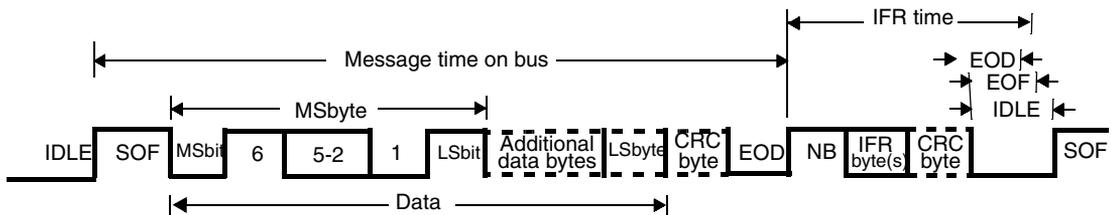


Table 2. C2SIa Message Time Duration

Message Components	Level	Normal Mode		4x Mode	
		TX ( $\mu$ s)	RX ( $\mu$ s)	TX ( $\mu$ s)	RX ( $\mu$ s)
SOF		192-208	163-239	48-52	41-60
Data Bits / CRC	0 	60-68	34-96	14-18	9-24
	1 	122-134	97-163	30-34	24-41
	0 	122-134	97-163	30-34	24-41
	1 	60-68	34-96	14-18	9-24
EOD		193-207	164-239	48-52	41-60
NB†		122-134	97-163	30-34	24-41
		60-68	34-96	14-18	9-24
EOF		271-289	240-320	67-73	60-80
Break		>290 short >758 long	>239	>290 short >758 long	>239

† There are two different conventions used for the Normalization bit. One type is an active long indicating that the in-frame response contains a CRC, and an active short indicating that it does not contain a CRC. The other type is vice versa: An active short indicating that the in-frame response contains a CRC, and an active long indicating it does not contain a CRC. The NBPOL bit (C2SIaGCR.4) is used to determine which type of convention is to be used. See the NBPOL bit description in section 9.4 on page 37.

The C2SIa transmits data bits via variable pulse width modulation (VPM) at either the normal mode or the 4x mode (this can be controlled by the 4XMODE bit (C2SIAGCR.6)). The following are descriptions of each of the message components:

#### Start Of Frame (SOF)

The start of every message is initiated when the transmitter drives the bus high for approximately 200  $\mu$ s (normal mode) or 50  $\mu$ s (4x mode), which is referred to as the start of frame (SOF).

#### Data Bits and IFR Bits (when an IFR is used)

Once the SOF duration has been established, the data bits are transmitted on alternating high-low levels. Whether a data bit is a '0' or '1' is designated by the time between two consecutive transitions. A pulse duration of approximately 64  $\mu$ s (normal mode) or 16  $\mu$ s (4x mode) represents a data 0 bit if the pulse is low, or a data 1 bit if the pulse is high. Likewise, a pulse duration of

approximately 128  $\mu\text{s}$  (normal mode) or 32  $\mu\text{s}$  (4x mode) represents a 0 bit if the pulse is high, or a 1 bit if the pulse is low. Refer to Table 2, *C2S1a Message Time Duration*.

### Cyclic Redundancy Check (CRC)

The CRC is optional. When this option is used, the C2S1a will automatically generate a CRC and append it to the end of the data bytes in a message, and to the end of an in-frame response.

There is no CRC for type I and II in-frame responses. Only type III in-frame responses include the CRC. For more information on the types of in-frame responses and CRC value calculations, see the SAE J2178/1 specification.

The generation of CRC in transmitted messages (normal messages and in-frame response messages) and the expectation of CRC in received messages is controlled by the CRCDIS bit (C2SIAGCR.7).

### End Of Data (EOD)

Once all data bits including CRC are sent, a falling edge occurs to generate a low level of approximately 200  $\mu\text{s}$  (normal mode) or 50  $\mu\text{s}$  (4x mode). This signifies the end of data (EOD).

An EOD will always appear after the last data byte in a message. Refer to Figure 2 and Figure 3.

### Normalization Bit

When there is an in-frame response from a responding device, the EOD duration ends when the responder sends its normalization bit prior to the start of the first in-frame response byte (refer to Figure 3).

The normalization bit is always an active high level. The duration of the normalization bit is the same as a high level data 0 or data 1 bit time. When the normalization bit is 1, the in-frame response message ends with a CRC byte. When the normalization bit is 0, the in-frame response message does not end with a CRC byte. The NBPOL bit (C2SIAGCR.4) will switch to meaning of 1 and 0 (1 = NO CRC) in order to conform to specific manufacturers conventions.

## End Of Frame (EOF)

An end of frame (EOF) signifies the end of a message and appears at the end of all messages. If there is no in-frame response from a responding device, then the low level end of data (EOD) duration at the end of the data bytes will eventually stretch into an end of frame (EOF). Refer to Fig. 1-2. If there is an in-frame response, then the EOF appears after the last in-frame response byte (or CRC byte, if the CRC is used). Refer to Fig. 1-3.

The EOF is a falling edge that lasts approximately 280  $\mu\text{s}$  (normal mode) or 70  $\mu\text{s}$  (4x mode). Once EOF reaches 320  $\mu\text{s}$  (normal mode) or 80  $\mu\text{s}$  (4x mode), the device that transmitted the previous message may begin transmitting a new start of frame (SOF) since no other nodes are trying to access the C2SI bus.

Other C2SIa's desiring bus access may try to arbitrate as early as between 280 $\mu\text{s}$  and 320  $\mu\text{s}$  (normal mode). When all other devices that desire bus access detect this rising edge on the bus, they send their start of frame (SOF) almost immediately. If a device loses arbitration (the high voltage level is dominant in arbitration), it removes itself from the bus and its transmission is stopped.

## Break

When a break signal is sent onto the bus, all nodes on the bus stop transmission immediately and go back into a reset condition. A break signal is initiated upon a rising edge and has a duration of at least 240  $\mu\text{s}$ . The C2SIa can transmit two different lengths depending on the LONGBRK bit. If this bit is a '1' then the break is 768  $\mu\text{s}$  else the break is 300  $\mu\text{s}$ .

## 4 Transmitting C2Sla Messages

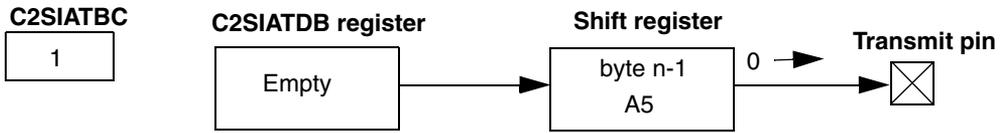
C2Sla transmission data must be transferred to the C2SIATDB transmit data buffer by using the CPU or the DMA controller. For DMA based operations, see section 8.1, *DMA Transactions*, on page 24 and the DMA controller specification.

All messages can be transmitted with or without a CRC appended, and is controlled by the CRCDIS bit (C2SIAGCR.7). See page 38 for the CRCDIS bit description.

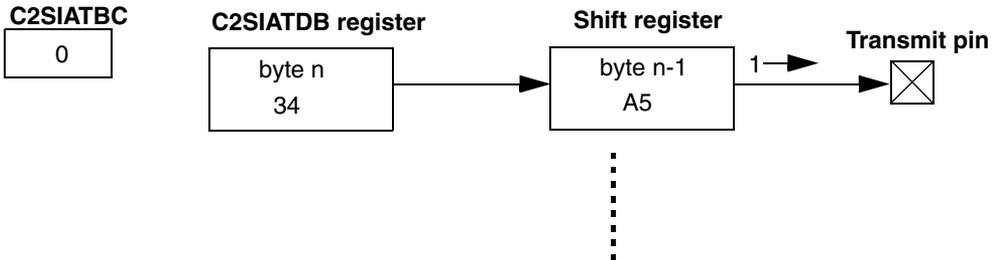
Internal to the C2Sla's transmitter is a shift register that holds the contents of a byte as it is physically shifted out MS bit first on the C2SIATXD pin. User software continues to supply data to the C2SIATDB register until the entire message has been transferred. Figure 5 shows the transmission of the last 2 bytes of an n byte message plus an appended CRC byte.

Figure 6. Byte Transmission

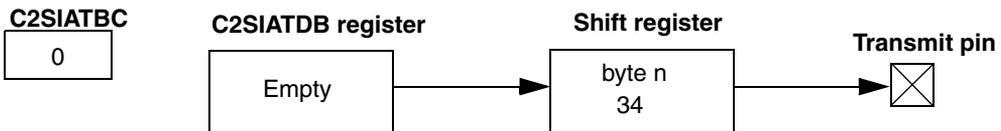
The C2SIATDB register is empty as the shift register shifts out byte n-1.



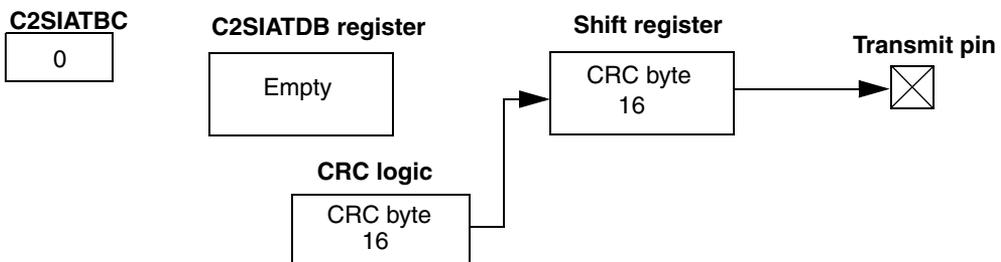
Write the last byte, n, into the C2SIATDB buffer. Note that C2SIATBC decrements to zero when C2SIATDB is written. Data shifts out MSB first.



As soon as the shift register has completely shifted out byte n-1, byte n is loaded from the C2SIATDB register into the shift register. The C2SIATDB is now empty.



Once byte n is completely shifted out of the shift register and the transmit byte counter (C2SIATBC) has counted down to zero, if CRCDIS = 0 (C2SIAGCR.7), then the CRC byte is calculated and loaded into the shift register.



During transmission, the C2SIa's receiver and transmitter monitor each bit that the transmitter delivers to the C2SI bus to detect loss of arbitration. Since high voltage is the dominant level, arbitration is lost when the transmitter attempts to deliver a low voltage to the C2SI bus and the receiver detects a high voltage on the data link. This event forces the transmitter to immediately enter the idle state. It also causes the arbitration lost bit (ARBIF), transmit idle bit

(TIDLIF), and transmit buffer empty bit (TBEIF) (C2SIAISR.7, 6, 5) to be set to 1 and the transmit DMA enable bit (TXDMAEN) (C2SIACR.3) to be cleared to 0.

A start bit (SOF) is neither a data '1' nor a data '0' and is not subject to arbitration. The C2Sla's transmitter will place the start bit on an idle bus or begin following another module's start bit already in progress.

---

**Note: Transmission with Lost Arbitration**

The C2Sla module does not attempt to retransmit a message that lost arbitration during a transmission attempt except in the case of a type II in-frame response. See the T2IFR bit (C2SIAGCR.5) description on page 1-39.

Detection of lost arbitration is handled by monitoring the ARBIF bit (C2SIAISR.7). If the ARBIE bit (C2SIAICR.7) = 1, a transmit interrupt request can be generated on this event.

While the ARBIF bit is set, the transmitter does not attempt to communicate on the link. User software must guarantee that ARBIF is cleared before attempting to transmit on the bus.

---

## 4.1 Transmitting Non-IFR Messages

Non-IFR messages are sent when the TIFR bit (C2SIAGCR.1) = 0. Before beginning a transmission, the following steps are recommended:

- Set the C2SIATBC register to a non-zero value.
- Read the C2SIACCSR register.
- Read the C2SIARDB register.
- Check that Arbitration bit (ARBIF) is cleared (C2SIAISR.7 = 0).
- Load the C2SIATDB register with either the CPU or with a DMA transfer. This last step starts the transmission.

The C2Sla monitors the C2SI bus until it has become IDLE. Transmission of an SOF sequence prefixes a normal message's packet of data bytes. Following the SOF, the shift register is loaded from the C2SIATDB register and the transmission of data bytes continues until the C2SIATDB register and the shift register are empty and C2SIATBC is 0. A CRC is appended to the end of the packet of data bytes if the CRCDIS bit is 0 (C2SIAGCR.7 = 0).

When you start a message, the first byte put into the C2SIATDB will quickly move to the empty shift register and the TBEIF bit will indicate that the C2SIATDB is ready for the second byte. Always wait for the TBEIF bit to be set before loading into the C2SIATDB.

Put a new byte into the C2SIATDB buffer after checking for errors and after checking if the TBEIF bit is set. If you clear the TBEIE bit after the writing the last byte you won't get an extraneous interrupt when the message finishes.

## 4.2 Transmitting In-Frame Response (IFR) Messages

In-frame response messages are sent when TIFR bit (C2SIAGCR.1) is 1. Before beginning a transmission, the following steps are recommended:

- Set the TIFR bit
- Check for message errors
- Set C2SIATBC register has to a non-zero value.
- Read the C2SIACCSR register.
- Read the C2SIARDB register.
- Check Arbitration lost interrupt bit (ARBIF) is cleared (C2SIAISR.7 = 0).
- Load C2SIATDB register with either the CPU or with a DMA transfer. This last step starts the transmission.

The responder of the message begins transmitting an in-frame response after the occurrence of an EOD. An NB bit, which reflects whether or not a CRC follows the in-frame response, prefixes the in-frame response packet of data bytes. Following the NB bit, the shift register is loaded by the C2SIATDB register and the transmission of data bytes continues until the C2SIATDB register and the shift register are empty. A CRC is appended to the end of the packet of the in-frame response bytes if CRCDIS bit is 0 (C2SIAGCR.7 = 0).

The TIFR bit (C2SIAGCR.1) controls whether or not the C2Sla is transmitting an in-frame response or a normal start-of-frame message. The TIFR bit (C2SIAGCR.1) is automatically cleared if the receiver detects any errors during reception. This guarantees that an in-frame response is not transmitted in response to a corrupt message. When a receiver detects any errors during reception, the C2Sla will:

- Clear the TIFR bit (C2SIAGCR.1)
- Reset the transmitter
- Clear the TXDMAEN bit (C2SIACTR.3)
- Set the TBEIF (C2SIAISR.5) and TIDLIF (C2SIAISR.6) bits.

The application is responsible for recognizing that the message currently being received is expecting an in-frame response to be transmitted as a response. This may require the application to monitor each incoming byte of a message in order to detect this condition. The application must insure that

the in-frame response is ready for transmission prior to the completion of the received packet's EOD sequence.

### 4.3 Transmitting BREAK Messages

The transmission of a BREAK sequence is forced by setting the TBRK bit (C2SIAGCR.2) is 1. This results in the current transmit/receive condition of the link being overridden by the BREAK sequence. The duration of the BREAK sequence is independent of the state of the 4XMODE bit (C2SIAGCR.6). If the LONGBRK bit (C2SIACTR.7) is set, the break will be ~768  $\mu$ s long else if it is cleared the break will be ~300  $\mu$ s long.

Transmitting/receiving a BREAK disables all transmissions/transmitters on the link. It also automatically clears the 4XMODE bit. When the BREAK occurs, all transmitters transmitting a message other than BREAK will be forced off the bus.

You must wait for the break to finish before writing to the transmit data buffer (C2SIATDB). See section 5.3, *Receiving BREAK Messages*, on page 20 for an explanation on how the receiving devices react to a BREAK.

### 4.4 Transmission Arbitration

When a message is transmitted by the C2Sla module, a copy of the bit stream is redirected back to the received section so it can monitor its arbitration progress. If the C2Sla's transmitter sends a passive data level and the C2Sla's receiver returns an active data level, then the transmitter has lost arbitration.

When arbitration is lost, the transmitter removes itself from the data link, and the C2Sla will:

- Clear the XMITOK bit (C2SIACCSR.3) in the Completion Code register.
- Set the ARBIF bit (C2SIAISR.7)
- Clear the TXDMAEN bit (C2SIACTR.3) and go into a transmitter idle state.
- Mark the internal shift register and the C2SIATDB register as empty, therefore, setting the TIDLIF and TBEIF bits (C2SIAISR.6-5).

#### Transmission with Lost Arbitration

The C2Sla module does not attempt to retransmit a message that lost arbitration during a transmission attempt (except during a type II in-frame response). Detection of loss arbitration is correctly handled by monitoring

the ARBIF bit (C2SIAISR.7). An interrupt request can be generated on this event if ARBIF = 1 and ARBIE bit = 1 (C2SIAICR.7).

While the ARBIF bit is set, the transmitter does not attempt to communicate on the link. User software must guarantee that ARBIF is 0 (C2SIAISR.7) before attempting to re-transmit on the link.

If arbitration is not lost, the transmitter naturally idles itself at the end of a message. The transmitter is idle when the TXDMAEN bit (C2SIACR.3) is cleared and when TIDLIF and TBEIF bits (C2SIAISR.6-5) go high. In addition, the C2Sla's receiver and transmitter collectively determine whether the XMITOK bit (C2SIACCSR.3) should be set. It is set if the transmitter wins arbitration and the receiver has not detected any errors during the reception of the transmitted message.

#### 4.5 Byte Boundary Loss of Arbitration when Transmitting

If arbitration is lost on the last bit of a byte being transmitted, the transmitter does not immediately remove itself from the data link. Instead it transmits two additional '1's. If arbitration is lost again on the first '1', the transmitter immediately stops transmitting.

If this loss of arbitration had been due to noise, the extra two '1's are intended to corrupt a potentially acceptable, but erroneously short message generated by the transmitter. If loss of arbitration was due to a higher priority message, then the '1's have no affect on that message.

#### 4.6 Bus Error Conditions

Conditions can occur in a system that will temporarily or permanently cause the C2SI bus to fail. Three of the more common failure modes are:

- Short-to-voltage
- Short-to-ground
- Bus open - broken wire

The commercially available bus interface devices will handle the physical strain of short to ground and short to power without damaging either device. The interface device however will not directly inform the C2Sla module of the fault condition. The C2Sla module must determine the fault through indirect means.

### 4.6.1 Short-to-Voltage

The following conditions will make the bus appear to be shorted to a voltage source.

- Bus shorted to voltage source
- TX pin stuck high
- RX pin stuck high
- TX signal between C2S1a module and interface device shorted to voltage
- RX signal between C2S1a module and interface device shorted to voltage

Each of these conditions will cause a BREAK condition on the C2S1a and set the break flag bit, BRKIF. If the break interrupt enable bit, BRKIE, is also set and C2S1a interrupts enabled then the BREAK will cause an interrupt. See section 5.3, "Receiving BREAK Messages" for break handling. The problem can be isolated to a bus problem if the C2S1a enables the loop back mode of the device. The C2S1a will still be able to send and receive in loop back mode if the bus shorts to voltage.

### 4.6.2 Short-to-Ground

The follow conditions will make the bus appear to be shorted to ground.

- 1) Bus shorted to ground
- 2) TX pin stuck low (may still receive from other devices)
- 3) RX pin stuck low
- 4) TX signal between C2S1a module and interface device shorted to ground (may still receive from other devices).
- 5) RX signal between C2S1a module and interface device shorted to ground

If the C2S1a transmitter begins a message under these conditions the TXD pin will be set high and the receive pin will wait to see this high signal. If the receiver does not see a high for a certain time the C2S1a will declare a bus short condition and stop the transmitter, set the SHORTGND bit and set the TXERROR bit. The short declaration comes at (80 - calibration register constant) in microseconds which gives about a 55-65  $\mu$ s trip point.

If the short comes after the SOF, the message will be corrupted and can be detected in two different ways.

- 1) If it is before the first good byte, the TIDLIF and TXERROR bits get set.
- 2) If it is after the first good byte, the RCCIF bit sets in addition to the previous bits and some combination of BITERR, BYTERR, or CRCERR will be set depending on the short location in the message.

The C2Sla module cannot detect short to ground condition unless it is transmitting. Higher levels of software must detect the absence of bus traffic and attempt to isolate the cause.

The problem can be isolated to a bus problem if the C2Sla enables the loop back mode of the device. The C2Sla will still be able to send and receive in loop back mode if the bus shorts to ground or voltage.

#### **4.6.3 Open Bus**

The following conditions will make the bus appear to be open.

- Bus open or
- Bus wire broken

In this instance, the transmitter appears to be working if no in frame responses are expected. The data sent out the TXD pin will return through the RXD pin in the normal manner. Lack of an expected IFR will point to this condition as a possible problem. (Problems with the device generating the IFR is another cause)

The C2Sla will receive no data from some or none of the devices during this time so bus protocols should be set up to identify this type of problem.

#### **4.6.4 Other Bus Errors Not Discussed**

Some other bus errors not discussed:

- The intermittent error. How long are the errors and how often do they occur?
- Open signals between C2Sla and interface device. May look like either short to ground, short to power or intermittent depending on the interface device.
- Bus with too much load or capacitance -- slow bus.
- Combination of several of the above faults.

## 5 Receiving C2Sla Messages

### 5.1 Receiving Normal Messages

The reception of a normal message is preceded by an SOF sequence. Data bytes are received through the receiver's shift register, and then transferred into the C2SIARDB register. Reception of a normal message continues until a non-data bit sequence is received (for example, EOD, EOF, BREAK, noise). When this occurs after receiving one good byte then a completion code for the message is updated in the C2SIACCSR register. Messages received with errors in the first byte are ignored and no interrupts generated.

The state of the CRCDIS bit (C2SIAGCR.7) controls the formation of a CRC for transmitted data, and whether the receiver should expect a CRC at the end of the incoming data. If a CRC is expected and there is a CRC error, the CRCERR bit (C2SIACCSR.4) in the completion code is set.

### 5.2 Receiving In-Frame Response (IFR) Messages

The reception of an in-frame response (IFR) follows after the C2Sla successfully transmitted a normal message which ended with an end of data (EOD). What sets it apart from the reception of a normal message is the lack of an end of frame (EOF).

Reception of an in-frame response continues until a *non-data bit* sequence is received (for example, EOD, EOF, BREAK, noise). When this occurs, a completion code for the message is updated in the C2SIACCSR register.

The reception of an in-frame response sets the IFR bit (C2SIACCSR.1) in the completion code register. The state of the NB bit is reflected in the completion code's IFRCRC bit (C2SIACCSR.0). The IFRCRC bit is set when an in-frame response with a CRC byte appended to the end of it has been received. The CRCERR bit (C2SIACCSR.4) in the completion code will be set if there are any errors in this appended CRC byte.

If the software wants to determine if a message is normal or IFR on the first byte, it may be best to use the RIFR bit (C2SIAGSR.4). Reading the IFR bit in C2SIACCSR register could clear the RCCIF flag causing a missed interrupt.

If you need to know the IFR status after receiving a byte (RBFIF) then use the RIFR bit. To get the IFR status after reading a completion code (RCCIF) use the IFR bit in the C2SIACCSR register.

Note that the receiver can receive an IFR without a CRC even if CRCDIS bit = 0 (C2SIAGCR.7).

### 5.3 Receiving BREAK Messages

A BREAK on the data link causes any messages on the link to be aborted. The BREAK's corruption of a message in process is detected when a symbol has been overridden. The receiver's reaction to a BREAK depends on conditions at the time of the BREAK.

The BRKIF flag (C2SIAISR.1) will set when the C2SI bus has been active for over the minimum break time of 240  $\mu$ s. This break resets the transmitter which sets the TIDLIF and TBEIF bits. It stops the receiver and clears the 4XMODE bit.

If a message was in process, a symbol could get corrupted and the receiver posts a completion code with errors (BREAK and possibly BYTERR, BITERR or CRCERR). This sets the RCCIF bit (C2SIAISR.2) and, if RCCIE bit is 1, a receive interrupt request is generated.

When the software sees the BRKIF bit set it should read the C2SIABRK register to determine the current status of the break and to clear the BRKIF bit. If the BREAKEND bit is set then the current break is over and the C2Sla can begin any new transmissions. If the INBREAK bit is set then the C2Sla must suspend C2Sla messages until either the break ends or software determines a short-to-voltage condition exists.

To help in determining short-to-voltage conditions, the BRKIF bit will set every 4096  $\mu$ s while a break continues. Software needs to count these breaks flags and determine the point between the two possible conditions.

When the break ends, two interrupts sources will occur. The first is the BRKIF bit caused by the BREAKEND bit setting in C2SIABRK register. The second interrupt source is the RCCIF bit setting with the BREAK bit set in the C2SIACCSR register. The RCCIF can interrupt with BREAK bit set at either the end of a break corrupted message or at the end of the break.

### 5.4 Receiving Digital Filters

A digital filter internal to the C2Sla is used to filter noise pulses from the class II receiver driver module that are smaller/shorter than the analog interface chip's filter time constant.

## 6 Interrupts

The C2Sla has two main interrupt-related registers--one is an interrupt status register (C2SIAISR), and the other is an interrupt control register (C2SIAICR). The C2Sla module generates one interrupt request back to the CIM. When an interrupt generating event occurs, its corresponding interrupt status register (C2SIAISR) bit is set (for example, ARBIF = 1). If the corresponding interrupt control register (C2SIAICR) bit is enabled (for example, ARBIE = 1), then an interrupt request is sent to the CIM.

If multiple interrupt conditions occur either at the same time or during the time the interrupt service routine is being executed, multiple interrupt status flags will be set but only one interrupt will be generated back to the CIM. User software must read the C2SIAISR register to determine which event caused the interrupt and to determine which C2Sla interrupt condition has priority; the C2Sla allows the user to determine priority.

If the interrupt is caused by a wake from sleep mode, however, the WAKE bit (C2SIAGSR.0) is set, and will generate an interrupt. The WAKE bit is listed in the C2SIAGSR and not the C2SIAISR register. As a result, the application software may need to poll both registers to find all occurrences of an interrupt condition.

### 6.1 Proper Handling of Events

Some interrupt generating events, such as the transmit buffer empty interrupt are persistent and can occur continuously unless their cause is handled. Each interrupt bit should be cleared by either reading its associated register or by writing a '0' to the bit.

Bit	Name	Associated register
0	RBFIF -	Read the C2SIARDB register.
1	BRKIF -	Read the C2SIABRK register.
2	RCCIF -	Read the C2SIACCSR register.
3	RXOIF -	Write '0' to clear
4	TXUOIF -	Write '0' to clear
5	TBEIF -	Write to the C2SIATDB register
6	TIDLIF -	Write to the C2SIATDB register
7	ARBIF -	Write '0' to clear.
	WAKE -	Read the C2SIAGSR register (also clears ENWAKE)

When writing zeros to the bits, make sure all other bits are '1'. Do not use C commands like "ARBIF.bit = 0". This usually results in a read of the register followed by an AND operation and finally writing the result back. If an interrupt occurs in the middle of this then the final write will clear the interrupt and it will

be missed. Instead use writes to the registers like “C2SIAISR= 0x7F” to clear the ARBIF bit.

All these bits can be cleared by writing zeros to the bits but it is strongly recommended to read the associated register instead. In addition, it is also strongly recommended to only read the interrupts flag's associated register only when the flag is set. If you read the associated register just after flag gets set then the flag will clear and an interrupt will be missed. This caveat would not apply while initializing after a reset.

## 7 General Purpose I/O

Each of the C2Sla pins may be programmed via the C2Sla pin control registers (C2SIAPC1, C2SIAPC2 and C2SIAPC3) to be a general-purpose I/O pin.

When the C2Sla module is not used, the C2Sla pins may be programmed to be either general input or general output pins. This function is controlled via the C2SIAPC1 register. The xxDIN bits in the C2SIAPC3 register always reads the unfiltered values on the pins. When turning a pin into an output, first write the data to the xxOUT bit and then write to the xxDIR bit to turn on the output driver.

## 8 DMA Interface

### 8.1 DMA Transactions

If handling the C2S1a message traffic on a byte-by-byte basis requires too much CPU overhead and if the particular device is equipped with the DMA controller, the C2S1a may use the DMA controller to receive or transmit data directly to memory. The C2S1a module contains two DMA request enable bits: a transmit DMA enable (TXDMAEN) and a receive DMA enable (RXDMAEN), both of which are located in the C2SIACR register.

When a byte is being transmitted or received, the C2S1a will signal the DMA via a DMA request signal. The DMA controller will then perform the needed data manipulation.

For DMA-based transmissions, all messages (other than a BREAK) are assembled in RAM, and DMA transfers move the message, byte-by-byte, from RAM into the C2SIATDB register. (See the DMA controller specification). All messages transferred via DMA contain only data received via the receive buffer; the contents of the completion code status register (C2SIACCSR) are not transferred to RAM. User software must read the completion code status register (C2SIACCSR). See section 9.6, *C2S1a Completion Code Status Register (C2SIACCSR)*, on page 41.

The application is responsible for programming the C2SIATBC to the desired number of bytes to be transferred. If the C2SIATBC contains a different value than the DMA byte count register, a transmit under-run or overrun condition can occur. For specific DMA features, refer to the DMA controller specification.

Writing to the TXDMAEN bit will automatically request a DMA action so it should be the last actions when initializing. After that, the setting of the TBEIF will trigger the DMA action. The receive DMA action is triggered by the setting of the RBFIF bit. Of course, these two bits should have their interrupts disabled via the TBEIE and RBFIE bit to take advantage of the DMA.

### 8.2 Non-DMA Transactions

When the application performs data transfers via the CPU, the application is responsible for guaranteeing that the C2SIATDB register is kept full and that the C2SIARDB register is read in accordance with class II data communication rates. Failure to do so results in truncated transmissions or overrun during reception.

### 8.3 4X Mode

The C2SIa module has the ability to function in the normal mode or 4X clock mode. In the 4X clock mode, timing constants for the generation/reception of signals on the data link are effectively divided by four.

Changing the state of the 4XMODE bit (C2SIAGCR.6) resets the class II state-machine status, thereby aborting any transmissions or reception in progress. The receipt of a BREAK sequence automatically clears the 4XMODE bit and resets the class II state machine.

If the state of the 4XMODE bit is changed during the reception of a message, the C2SIACCSR register is updated to mark the end of the aborted message.

The C2SIa should calibrate before transmitting in 4X mode. See section 8.6, *Calibration Mode*, on page 27.

### 8.4 Low Power Mode

The C2SIa module has two means to be placed in a low-power mode: a global low-power mode from the system and a local low-power mode via the LPM bit (C2SIACTR.0). The net effect on the C2SIa is the same, independent of the source.

A low-power mode in effect shuts down all the clocks to the module. During a global low-power mode, nothing will be written to any register. A local low-power mode has the same effect, with the exception that the LPM bit may be written to, and hence able to place the module into a functional mode.

Since entering a low-power mode has the effect of suspending all state-machine activities, care must be taken when entering such modes to insure that a valid state is entered when low-power mode is active. For example, if a low power mode is entered during a transmission on the Class II bus before the message is complete, a completion code will never be sent out, and hence the integrity of the bus is corrupted. As a result, application software must insure that a low power mode is not entered during a transmission.

Low-power mode may be used in conjunction with the ENWAKE bit (C2SIAGCR.0) to allow C2SI bus activity to wake the device and exit the low-power mode.

To enter low-power mode and wake up on any C2SI bus activity:

- Wait for idle bus by checking the IDLE bit (C2SIAGSR.3)
- Service all interrupt sources or disable bits in the C2SIAICR register
- Set the ENWAKE bit (C2SIAGCR.0)
- Set the LPM bit (C2SIACR.0). This is now local low-power mode.

To enter global low-power mode, set the global LPM bits (CLKCNTL1:0). The global C2SIa interrupt does not have to be enabled to wake up the device.

To enter low-power mode and ignore all C2SI bus activity:

- Wait for idle bus by checking the IDLE bit (C2SIAGSR.3)
- Clear the ENWAKE bit (C2SIAGCR.0). Clear C2SIAICR.
- Set the LPM bit (C2SIACR.0). This is now local low-power mode.
- Unlike some other modules, you do not have to be in a privilege mode to set the LPM bit.

To enter global low-power mode, set the global LPM bits (CLKCNTL1:0).

To exit the low-power mode, you must first clear the LPM bit. No other C2SIa bit is writable until the LPM bit is cleared. The internal C2SIa counters and state machine will not start until the LPM bit is cleared. This means the time from the LPM cleared to the end of the OSF starting bit must meet the minimum SOF time in order to receive a good message. In many instances the wake-up message will be lost because this time could not be met.

The global peripheral power-down override bit (PPWNOVR< CLKCNTL.7) has no effect on the C2SIa.

## 8.5 Emulation Mode

The C2SIa module may be placed in a suspend mode by the TMS470 system. This is usually when the TMS470 is being used as an emulator or being debugged via the test access port (TAP). When being used by a monitor program, the receive data buffer (C2SIARDB) has a mirror register called C2SIAEMU. This register contains the same contents as the C2SIARDB, but a read of this register will not cause the receive buffer full interrupt flag (RBFIF) to clear. This allows the user to keep a memory window open for the receive buffer, without having the monitor program clear the interrupt automatically.

The software has the choice via the ESPEN bit (C2SIACR.1) as to the state-machine action taken during a suspend mode. If the ESPEN bit (C2SIACR.1) is active, the C2Sla will immediately suspend its activity. Once again, the user's software must ensure that suspending the transmission or reception of data will not corrupt the class II bus. If ESPEN is inactive, the C2Sla will continue operating normally.

## 8.6 Calibration Mode

Calibration allows the C2Sla module to know the expected time delay between sending a bit out of the TXD pin and receiving the same bit back into the RXD pin. Due to the variations among different analog designs, the C2Sla module should be calibrated to the actual bus load. An improperly calibrated C2Sla will have transmit bits times outside the ideal values and may not arbitrate correctly when the C2SI bus is already operating at very marginal levels. The J1850 protocol allows for variation but this tolerance is better used solving C2SI bus degradation.

The calibration constants do not affect the C2Sla's reception of data from other devices. The value is used mainly for transmission and arbitration.

On reset, the C2Sla will place a default value of 23 into the C2SIACAL register. Changing to 4X mode with the 4XMODE bit will place a default value of 7 in the register. Changing back to normal mode will again put 23 in the calibration register.

The C2Sla module calibration constant can be set automatically or manually. Use the formulas below to manually set the calibration value in the C2SIACAL register.

$$\begin{aligned} \text{cal register} &= \text{TX/RX delay } \mu\text{s} + 4 \text{ ICLKS} + 7 \mu\text{s} ; \text{ normal mode} \\ \text{cal register} &= \text{TX/RX delay } \mu\text{s} + 4 \text{ ICLKS} \quad ; \text{ 4X mode} \end{aligned}$$

Where ICLK is the peripheral bus clock frequency. At higher speeds, this term becomes insignificant. Digitizing errors ranging from 0 to 1  $\mu\text{s}$  apply to both calculations and will affect output values.

To find the TX/RX delay measure the time from the TXD changes to the time the change is observed on the RXD pin. This is a function of the external interface chip and to a lesser extent on the bus loading. Measure across the operating temperature range. The normal start bit measured at the TXD pin should ideally be 200  $\mu\text{s}$ .

To use the automatic mode the software must transmit at least one byte of code with the CALEN bit (C2SIACR.6) set. After sending the message the CALEN bit will clear and the measured value will be in the C2SIACAL register.

A counter in the C2S1a will measure the delay and adjust the transmit timings automatically according to the results of the calibration cycle.

If possible, it is recommended that you first calibrate using the loop-back mode to get close to the final value and then send a calibration message on the loaded bus. It is also possible to calibrate using ONLY the loop-back mode on some external transceivers, but this will need to be determined on a system-by-system basis.

If you calibrate on a busy bus, then you may lose arbitration to another device. If this happens you must repeat the loop-back cycle before attempting another calibration cycle on the bus. Keep trying until the C2S1a returns a XMITOK status after a bus calibration.

## 9 C2Sla Internal Registers

Table 3. C2Sla Control Register File Used With the TMS470 CPU

Address Offset†	REGISTER NAME	7	6	5	4	3	2	1	0	Register
0x00	<b>C2SIAISR</b>	ARBIF	TIDLIF	TBEIF	TXUOIF	RXOIF	RCCIF	BRKIF	RBFIF	Interrupt status register
0x04	<b>C2SIAICR</b>	ARBIE	TIDLIE	TBEIE	TXUOIE	RXOIE	RCCIE	BRKIE	RBFIE	Interrupt control register
0x08	<b>C2SIAGSR</b>	Rsrvd	Rsrvd	TXERROR	RIFR	IDLE	SHORT-GND	NOISE	WAKE	Global status register
0x0C	<b>C2SIAGCR</b>	CRCDIS	4XMODE	T2IFR	NBPOL	TXRESET	TBRK	TIFR	ENWAKE	Global control register
0x10	<b>C2SIATDB</b>	TDDATA.7:0								Transmit data buffer register
0x14	<b>C2SIACCSR</b>	ROVR	BITERR	BYTERR	CRCERR	XMITOK	BREAK	IFR	IFRCRC	Completion code status register
0x18	<b>C2SIACTR</b>	LONGBRK	CALEN	LPEN	SOFX	TXDMAEN	RXDMAEN	ESPEEN	LPM	Peripheral control register
0x1C	<b>C2SIACLK</b>	Reserved			ICKLFR					Interface clock register
0x20	<b>C2SIATBC</b>	TBCOUNT.7:0								Transmit byte counter
0x24	<b>C2SIAPC1</b>	TXPOL	RXPOL	TXFUN	RXFUN	LPFUN	TXDIR	RXDIR	LPDIR	Pin control register 1
0x28	<b>C2SIAPC2</b>	Reserved					TXDOUT	RXD-OUT	LPDOUT	Pin control register 2
0x2C	<b>C2SIAPC3</b>	Reserved					TXDIN	RXDIN	LPDIN	Pin control register 3
0x30	<b>C2SIAEMU</b>	REDATA.7:0								Receive emulation register
0x34	<b>C2SIARDB</b>	RDDATA.7:0								Receive data buffer register
0x38	<b>C2SIACAL</b>	ARBTYPE	0	CAL 5:0						Calibration control register
0x3C	<b>C2SIABRK</b>	Reserved						BREAK-END	INBREAK	break status register

† The actual address of these registers is device specific and CPU specific. See the specific device data sheet to verify the C2Sla register addresses.

### 9.1 C2Sla Interrupt Status Register (C2SIAISR)

The C2SIAISR register consists of interrupt flags. Each interrupt flag represents the occurrence of a different interrupt generating event; the individual flags are set when their respective interrupt generating event occurs. Interrupt flags in the C2SIAISR register are purely status flags and do

not initiate interrupt requests alone. Only when used in conjunction with the interrupt enable bits in the interrupt control register (C2SIAICR) will any requests be made to handle the interrupt.

Any bit in this register that is set along with a corresponding '1' in the C2SIAICR register will generate a continuous interrupt until either the enable or the flag bit is cleared. For most all instances, it is recommended that the code write zeros only to the ARBIF, TXUOIF, and RXOIF bits. The rest of the flag bits are best cleared by reading or writing the appropriate registers.

hex	7	6	5	4	3	2	1	0
0x00	ARBIF	TIDLIF	TBEIF	TXUOIF	RXOIF	RCCIF	BRKIF	RBFIF
	RC-0	RC-1	RC-1	RC-0	RC-0	RC-0	RC-0	RC-0

RC = Read/Clear, -n = Value after reset

### Bit 7 **ARBIF** Transmit Arbitration Lost Interrupt Flag.

This bit is set when the C2Sla determines that it has lost its transmit arbitration attempt for the data link. This can occur when a message of higher priority is being transmitted by another device. Also, a BREAK asserted on the data link could cause arbitration to be lost in certain cases.

When arbitration is lost and ARBIE = 1 (C2SIAICR.7), the C2Sla module generates a transmit interrupt request.

The ARBIF bit and the type II in-frame response control bit, T2IFR (C2SIAGCR.5), need to be monitored together in order to determine whether a transmission will be resent or not when a transmit arbitration lost interrupt occurs. See also T2IFR, page 1-39. The following two conditions can happen when the C2Sla loses arbitration while transmitting:

- If ARBIF = 1 and the T2IFR = 0 (meaning a type II in-frame response is not in progress):
  - The TXDMAEN bit (C2SIACTR.3) is automatically cleared
  - The TIDLIF and TBEIF bits (C2SIAISR.6,5) are set.

The transmitter is disabled from continuing its transmission and does not attempt to get on the data link. ARBIF bit must be cleared before any transmission can take place.

- If ARBIF = 1 and T2IFR = 1, the transmitter remains active and automatically re-sends the contents of the shift register.

For more details on the type II in-frame response see the SAE J1850 specification.

If transmit retry is required for the next frame, the latency requirement for interrupt response is:

- Normal operation: ~790  $\mu$ sec (a CRC byte shift time and an EOF in normal mode)
- 4X mode: ~70  $\mu$ sec (an EOF in 4X mode)

Clear this bit by writing a '0' to this bit. Write '1's to the other bits.

- 0 = Arbitration not lost.
- 1 = Transmitter lost during data link arbitration attempt.

#### **Bit 6** **TIDLIF** Transmit Idle Interrupt Flag.

Active high indicates that the transmitter is idle and available for reloading. This occurs whenever the following events occurs:

- The transmitter loses arbitration.
- Break
- Some transmit errors
- The transmit shift register became empty after emptying the C2SIATDB register while TXDMAEN = 0 (C2SIACR.3 = 0).

When one of these occur and TIDLIE = 1, C2Sla generates a transmit interrupt request.

TIDLIF remains set as long as both the C2SIATDB register and the transmit shift register are empty. It can be cleared by a CPU write to the C2SIATDB register or a DMA transfer from system RAM to the C2SIATDB register.

- 0 = Transmitter is in use.
- 1 = Transmitter is idle.

#### **Bit 5** **TBEIF** Transmit Buffer Empty Interrupt Flag.

This bit is set whenever the C2SIATDB register is emptied by the transfer of its contents into the internal shift register. TBEIF is set when the C2SIATDB register is empty or when bits are posted to the completion code register (C2SIACCSR). It can be cleared by a CPU write to the C2SIATDB register or a DMA transfer from system RAM to the C2SIATDB register

- 0 = Transmit data buffer contains a character.
- 1 = Transmit data buffer register is empty.

**Bit 4 TXUOIF** Transmit Under-run, Over-run Interrupt Flag.

A transmit under-run condition occurs when the transmitter has finished transmitting a byte and is ready to receive the next byte, but the DMA or CPU has not provided the next byte to be transmitted. However, if C2SIATBC = 0, then the transmit under-run flag is not set, as that indicates the end of transmission condition.

A transmit over-run condition can occur if the CPU or DMA sends a byte to be transmitted when C2SIATBC = 0.

Clear this bit by writing a '0' to this bit. Write '1's to the other bits.

- 0 = Transmit under-run, over-run has not occurred.
- 1 = Transmit under-run, over-run has occurred.

**Bit 3 RXOIF** Receive Over-run Interrupt Flag.

A receive over-run condition occurs when the receiver has received the next byte, transferred it to the receive buffer (C2SIARDB), but the previous buffer contents have not yet been read, either by the CPU or the DMA. The previous byte is overwritten (and hence lost), and RXOIF is set.

Clear this bit by writing a '0' to this bit. Write '1's to the other bits.

- 0 = Receive over-run has not occurred.
- 1 = Receive over-run has occurred.

**Bit 2 RCCIF** Receiver Completion Code Interrupt Flag.

The RCCIF bit is set:

- After an end of data (EOD) time at the end of a normal and in-frame response message,
- After the end of a BREAK
- After an error if at least one good byte was received.

This bit is continuously set until the C2SIACCSR register is read by the CPU. Reading the C2SIACCSR will clear this bit.

The interrupt latency requirement for interrupt response:

- Normal operation:~700  $\mu$ sec (an SOF pulse one byte shift times in normal mode)
  - 4X mode:~175  $\mu$ sec (an SOF and one byte shift time in 4X mode)
- 0 = All bits in C2SIACCSR register hold an intermediate completion code.
- 1 = Completion code C2SIACCSR register bits are valid.

### Bit 1 **BRKIF** Received Break Interrupt Flag.

This bit is set in three different ways:

- A new break symbol has just been detected.
- A break symbol has just ended.
- The break symbol has continued for 4096  $\mu$ s since this bit was last set.

The bit should be cleared by reading the C2SIABRK register. You can also clear this bit by writing '0' but it is not recommended. Writes to the C2SIATXD are ignored if this bit is set or when a break is in progress.

- 0 = Break conditions have not occurred since this bit was last cleared
- 1 = Break conditions have occurred since this bit was last cleared.

### Bit 0 **RBFIF** Receiver Buffer Full Interrupt Flag.

This bit is set when the receiver posts a received data byte in the C2SIARDB register. When this occurs and RBFIE = 1, the C2Sla generates a receive interrupt request.

RBFIF bit remains set as long as the C2SIARDB register is full and can be cleared by a CPU read of the C2SIARDB register or by a DMA transfer from the C2SIARDB register to system RAM.

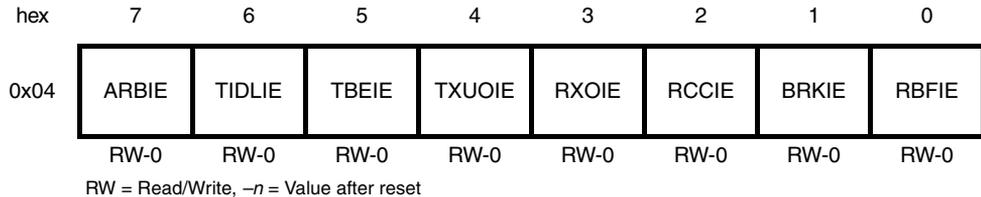
The latency requirement for interrupt response:

- Normal operation:~500  $\mu$ sec (one byte shift time in normal mode)
  - 4X mode:~125  $\mu$ sec (one byte shift time in 4X mode)
- 0 = No new data bytes have been received.
- 1 = A data byte has been received.

## 9.2 C2Sla Interrupt Control Register (C2SIAICR)

The C2SIAICR register consists of interrupt enable control bits. Interrupt flags in the interrupt status register (C2SIAISR) are purely status flags and do not

initiate interrupt requests alone. Each interrupt flag in the C2SIAISR register has a corresponding interrupt enable bit in the C2SIAICR register. Any interrupt generating events that cause the interrupt flags to be set, occurring while the interrupt enable is 0 are ignored and lost. Only events that occur after an interrupt enable is 1 initiate an interrupt request.



**Bit 7**      **ARBIE** Transmit Arbitration Lost Interrupt Enable.

When set to 1, the event (s) which causes ARBIF to be set to 1 also generates a transmit interrupt request.

When cleared to 0, no interrupt request is possible due to the setting of ARBIF; but this does not prevent the ARBIF flag from being set.

0      =    ARBIF interrupt disabled.  
 1      =    ARBIF interrupt enabled.

**Bit 6**      **TIDLIE** Transmitter Idle Interrupt Enable.

When set to 1, the event(s) which causes TIDLIF to be set to 1 also generates a transmit interrupt request.

When cleared to 0, no interrupt request is possible due to the setting of TIDLIF; but this does not prevent the TIDLIF flag from being set.

0      =    TIDLIF interrupt disabled.  
 1      =    TIDLIF interrupt enabled.

**Bit 5**      **TBEIE** Transmit Buffer Empty Interrupt Enable.

When set to 1, the event which causes TBEIF to be set to 1 also generates a transmit interrupt request.

When cleared to 0, no interrupt request is possible due to the setting of TBEIF; but this does not prevent the TBEIF flag from being set.

0      =    TBEIF interrupt disabled.  
 1      =    TBEIF interrupt enabled.

**Bit 4**      **TXUOIE** Transmit Under-run, Over-run Interrupt Enable

When set to 1, the event which causes TXUOIF to be set to 1 also generates a transmit interrupt request.

When cleared to 0, no interrupt request is possible due to the setting of TXUOIF; but this does not prevent the TXUOIF flag from being set.

0 = TXUOIF interrupt disabled.  
1 = TXUOIF interrupt enabled.

**Bit 3** **RXOIE** Receive Over-run Interrupt Enable

When set to 1, the event which causes RXOIF to be set to 1 also generates a receive interrupt request.

When cleared to 0, no interrupt request is possible due to the setting of RXOIF; but this does not prevent the RXOIF flag from being set.

0 = RXOIF interrupt disabled.  
1 = RXOIF interrupt enabled.

**Bit 2** **RCCIE** Receiver Completion Code Interrupt Enable.

When set to 1, the event which causes RCCIF to be set to 1 also generates a receive interrupt request.

When cleared to 0, no interrupt request is possible due to the setting of RCCIF; but this does not prevent the RCCIF flag from being set.

0 = RCCIF interrupt disabled.  
1 = RCCIF interrupt enabled.

**Bit 1** **BRKIE** Received Break Interrupt Enable.

When set to 1, the event which causes BRKIF to be set to 1 also generates a receive interrupt request.

When cleared to 0, no interrupt request is possible due to the setting of BRKIF; but this does not prevent the BRKIF flag from being set.

0 = BRKIF interrupt disabled.  
1 = BRKIF interrupt enabled.

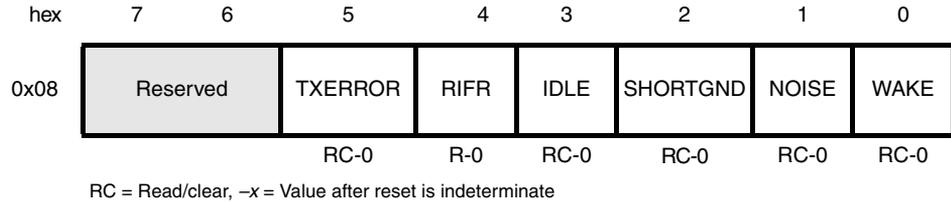
**Bit 0** **RBFIE** Receive Buffer Full Interrupt Enable.

When set to 1, the event which causes RBFIF to be set to 1 also generates a receive interrupt request.

When cleared to 0, no interrupt request is possible due to the setting of RBFIF; but this does not prevent the RBFIF flag from being set.

- 0 = RBFIF interrupt disabled.  
 1 = RBFIF interrupt enabled.

### 9.3 C2Sla Global Status Register (C2SIAGSR)



**Bits 7-6** **Reserved** Reads are undefined and writes have no effect.

**Bit 5** **TXERROR** Transmission error found.

An error was found during transmission. A loss of arbitration is not considered an error but if the C2Sla expects an active and sees a passive then this is considered an error. (See ARBTYPE bit for exceptions). This bit will be the only reliable indicator for transmission errors happening in the first byte. After the first byte the RCCIF bit will trigger an interrupt. This bit will not cause an interrupt directly but can be checked after receiving a TIDLIF interrupt.

This bit is cleared by writing a zero to this bit.

- 0 = A transmission error was not detected since last clearing this bit.  
 1 = A transmission error was detected since last clearing this bit.

**Bit 4** **RIFR** In Frame response.

This read only bit is set on the rising edge of a normalization bit and remains set during the IFR. Reading the completion code or receiving the first good non-IFR byte will reset this bit. This bit should be used to determine the IFR status after receiving a byte. Use the IFR bit in conjunction with the completion code.

- 0 = This message is not an IFR.  
 1 = This message is an IFR.

**Bit 3** **IDLE** Data Link Idle Flag.

An idle data link has a lack of activity for more than 280  $\mu$ sec. This bit reflects the current status of the data link and is set according to current activity.

- 0 = C2Sla data link is busy.

1 = C2Sla data link is idle.

**Bit 2 SHORTGND** Short to ground detected.

This bit indicates that the C2Sla transmitter attempted to send a message but it could not detect the active edge of the SOF after it was sent. This short condition resets the transmitter and sets this bit. The trip point time is (80 - calibration register) in microseconds.

0 = Short to ground not detected.

1 = Short to ground has been detected.

**Bit 1 NOISE** Noise detected on C2Sla Data Link Flag.

Noise is a pulse of duration less than a normal bit time. This bit is set and remains set after the detection of noise and therefore records the detection of past noise. The digital noise filter will eliminate the shorter noise pulses before triggering this bit.

0 = Noise has not been detected.

1 = Noise has been detected by the receiver on the data link.

**Bit 0 WAKE** Wake up from Low Power Mode Status Flag.

This bit tracks the state of the C2Sla wake up output. If ENWAKE = 1 (C2SIAGCR.0 = 1) and activity was detected on the data link, WAKE is set. A read of this register clears both the WAKE and the ENWAKE bits. See also the bit description for ENWAKE (C2SIAGCR.0)

If the C2Sla module had been sleeping, the TMS470 device wakes up when data link activity is detected. When a wake condition occurs, an interrupt is generated back to the CPU, and WAKE is set.

0 = No data on bus or wake up mode not enabled.

1 = Data detected on C2SI bus and the wake up mode is enabled.

## 9.4 C2Sla Global Control Register (C2SIAGCR)

hex	7	6	5	4	3	2	1	0
0x0C	CRCDIS	4XMODE	T2IFR	NBPOL	TXRESET	TBRK	TIFR	ENWAKE
	RW-0	RW-0	RW-0	RW-0	W-0	RW-0	RW-0	RW-0

RW = Read/Write, -n = Value after reset

**Bit 7**            **CRCDIS** Generation Disabled Control Bit.

When set, this bit prevents the C2Sla module from appending a CRC to the end of transmitted messages including IFRs. It also disables CRC checking during the receipt of a message. Therefore if this bit is set and the incoming message has a CRC appended to it, the C2Sla will see the CRC byte at the end of the message as a data byte.

When this bit is cleared, CRC transmission/reception is enabled. A CRC will be appended to the end of messages transmitted, and CRC checking is enabled for the reception of messages. If there is a CRC error during the receipt of a message, then the CRCERR bit (C2SIACCSR.4) will be set.

- 0        =    All data packets and transmitted IFRs include a CRC.
- 1        =    No CRC is appended/expected with messages.

**Bit 6**            **4XMODE** 4X Mode Control Bit.

When set, this bit puts the C2Sla in 4X mode which quadruples the transmit/receive bit rate. This bit is reset whenever a BREAK message is received. Toggling this bit causes the C2Sla internal state machine to be reset and forces TIFR (C2SIAGCR.1) and TXDMAEN (C2SIACTR.3) to be cleared. Changing this bit forces the default value for the mode into the C2SIACAL calibration register so software should update the C2SIACAL register if necessary. For additional information, see section 8.3, *4X Mode*, on page 25.

- 0        =    Operate in normal mode.
- 1        =    Operate in 4X mode.

**Bit 5**            **T2IFR** Type 2 In-Frame Response Control Bit.

This bit puts the C2Sla into a special type 2 in-frame response mode whereby one-byte In-Frame Responses may be arbitrated. When the T2IFR bit is set, the C2Sla automatically requests the one-byte in-frame response previously loaded and tries to transmit it again at the next byte boundary. Once the transmitter wins arbitration, the byte is gone and the T2IFR bit is automatically cleared by the C2Sla. The CRC check needs to be disabled, i.e., CRCDIS = 1 when transmitting Type II IFRs.

The program must set the T2IFR before the response byte is loaded.

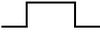
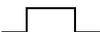
The CRCDIS bit and the TIFR bits must also be set when T2IFR = 1.

- 0        =    Normal mode.
- 1        =    Enter type II IFR mode. The next transmission by this node will be a type II IFR. The CRCDIS and TIFR bits should be set at the same time or before this bit.

**Bit 4**      **NBPOL** Normalization/Bit Polarity Control Bit.

This bit determines which required convention for the normalization bit is to be used.

The NBPOL bit may be changed at any time, but is intended to be initialized once along with any other setup tasks at the beginning of the user program.

NBPOL	NB		CRC appended to end of IFR?
0		0	NO
0		1	YES
1		0	YES
1		1	NO

When NBPOL = 0, the NB bit is decoded as follows:

- 0      =    No CRC follows the in-frame response.
- 1      =    CRC follows the in-frame response.

When NBPOL = 1, the NB bit is decoded as follows:

- 0      =    CRC follows the in-frame response.
- 1      =    No CRC follows the in-frame response.

**Bit 3**      **TXRESET.** Reset the transmitter.

Writing a '1' to this bit immediately resets the transmitter. This sets the TIDLIF and TBEIF bits, clears the C2SIATBC and causes the TXD pin to go passive. This bit always reads zero. The receiver may later find errors commensurate with an aborted transmission and it may set the TXERROR bit.

- 0      =    Writing a zero has no effect.
- 1      =    Writing a one will reset the transmitter.

**Bit 2**      **TBRK** Transmit BREAK Sequence Control Bit.

When set, this bit forces the transmitter to send a BREAK sequence. Once sent, this bit automatically clears itself and causes TIFR, 4XMODE (C2SIAGCR.1,6) and TXDMAEN (C2SIACTR.3) to be cleared also. After the

BREAK sequence is transmitted, a completion code is formed in the C2SIACCSR register and the RCCIF and BRKIF bits (C2SIAISR.2,1) are set. You must wait for the break to finish before writing to the transmit data buffer register (C2SIATDB). The LONGBRK bit determines the length of the break symbol.

- 0 = No BREAK sent.
- 1 = Send a BREAK now.

**Bit 1 TIFR** Transmit In-Frame Response Control Bit.

When set, this bit informs the transmitter to begin transmitting the in-frame response after an EOD. When clear, the transmitter waits for an EOF or an idle data link to transmit a normal message. The transmitter must be suitably enabled by having placed data into the internal shift register (either by DMA or CPU writes).

This bit is automatically cleared if the receiver detects any error, receives a BREAK, or the 4XMODE bit (C2SIAGCR.6) is manually reset. The program must clear this bit before the next regular transmitted message.

- 0 = Start transmission after an EOF or on an IDLE data link.
- 1 = Start transmission after an EOD.

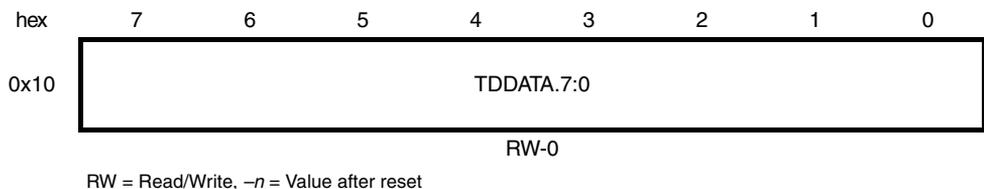
**Bit 0 ENWAKE** Wake up Enable.

When set, this bit enables the C2Sla module to generate a wake up interrupt whenever activity is detected on the data link. (This bit must be cleared to prevent the activation of the wake up signal)

Once a wakeup condition has occurred, ENWAKE is cleared.

- 0 = Operate in normal mode.
- 1 = Enable the wake up circuit.

**9.5 C2Sla Transmit Data Buffer Register (C2SIATDB)**



**Bits 7-0 TDDATA** Transmit Data Buffer Register.

After reset, both the C2SIATDB buffer and shifter are considered *empty* since nothing has been written to them. After a write to C2SIATDB, the buffer is considered *full*. The buffer will transfer its data to the shifter when the shifter is empty. After transferring, the C2SIATDB buffer will be empty. This means that the software can load two bytes at the start of message to fill both buffer and shifter. The user does not have direct access to the shifter.

Write to the C2SIATDB register only when TBEIF = 1.  
The emptying of C2SIATDB causes two events to occur:

- The setting of TBEIF = 1.
- The generation of a DMA request by the transmitter.

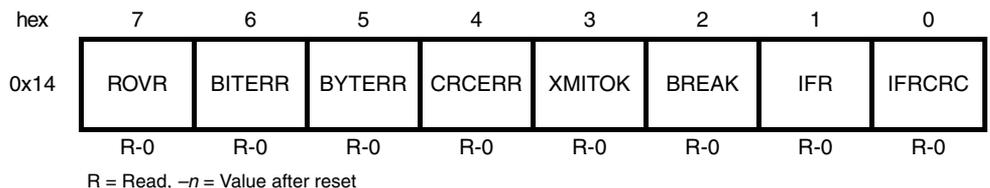
A write to the C2SIATDB register either by the CPU or transmit DMA causes the TBEIF and TIDLIF bits (C2SIAISR.5,6) to be cleared.

## 9.6 C2Sla Completion Code Status Register (C2SIACCSR)

The C2SIACCSR register reflects the status of a message being received. Reading this register will not alter the contents of the register. The new completion codes are described in Table 4 on page 43. This registers content cannot be altered by software. It is hardware set and cleared. It is set, by hardware, when the conditions are met and is cleared after receiving the next good byte.

Read this register only after seeing the RCCIF bit set. Since reading this register clears the RCCIF reading it at other times could cause the code to miss a completion code.

The BITERR, BYTERR, CRCERR and XMITOK bits are buffered and will change only when the RCCIF bit would be set. This register is cleared after the first good received byte so there at least a one byte window between the setting of the RCCIF and the bits clearing.



**Bit 7**      **ROVR** Receive Completion Code Over-Run Flag.

The ROVR bit indicates that the completion code has not been read but the C2SIACCSR has changed. This means that the C2SIACCSR could be invalid for the previous message. This bit sets when the C2SIACCSR has not been read but the C2Sla either clears the C2SIACCSR after the first good byte of a message is received, attempts to set the RCCIF, or is at the end of a break. The ROVR bit clears after the C2SIACCSR is read.

0 = No over-run  
1 = The receiver overran the C2SIACCSR register during the last message; completion code data was lost.

**Bit 6 BITERR** Received an Improperly Timed Bit Error Flag.

0 = No bit timing errors in received message  
1 = The received message is corrupted because of a bit timing error.

**Bit 5 BYTERR** Received an Incomplete Byte Error Flag.

This bit is set when the receiver detects that each byte does not contain exactly 8 bits.

0 = No byte errors in received message  
1 = The received message is corrupted because of an incomplete byte error.

**Bit 4 CRCERR** Received Message with a CRC error Flag.

This bit is set when the CRC, sent by the transmitter along with the data, does not correspond to the data read by the receiver.

0 = CRC correct for received message.  
1 = The received message is corrupted as indicated by a CRC error.

**Bit 3 XMITOK** Received Transmitted Message and Transmit was OK Flag.

This bit is set after receiving a message and the following conditions are satisfied:

- This C2Sla module was the transmitter of the message.
- The message won arbitration.
- No reception errors occurred.

This bit is always cleared when a BREAK is received.

0 = Not contending for data link, arbitration was lost, or transmission errors occurred.

	1	=	Transmitted message was sent successfully.
<b>Bit 2</b>	<b>BREAK</b>		Received a BREAK Sequence Flag.
			This bit sets when a break symbol is detected. The RCCIF flag sets when a message was corrupted by a break or if the break symbol just ended. Both of these conditions will show the BREAK bit set. Use the BRKEND and INBREAK bits in the C2SIABRK register for more control.
	0	=	No BREAK received.
	1	=	Received a BREAK sequence.
<b>Bit 1</b>	<b>IFR</b>		Received an In-Frame Response Flag.
			This bit is set after the first good byte of an IFR message is received. Since the NB bit indicates that an in-frame response follows, the setting of this bit is an indication that the incoming message is an in-frame response.
	0	=	An in-frame response has not been received.
	1	=	An in-frame response has been received.
<b>Bit 0</b>	<b>IFRCRC</b>		Received an In-Frame Response with a CRC Flag.
			This bit is set when an in-frame response message with a CRC appended to it has been received.
	0	=	An in-frame response with a CRC has not been received.
	1	=	An in-frame response with a CRC has been received.

Table 4. Completion Code Descriptions

Completion Code	Description
00h	Received at least 1 byte of a normal message.
02h	Received at least 1 byte of an in-frame response with no CRC transmission.
03h	Received at least 1 byte of an in-frame response with CRC and no transmission.
08h	Transmitted a message with no errors or lost of arbitration.
0Ah	Transmitted an IFR message with no errors or loss of arbitration.
0Bh	Transmitted an IFR message with no errors or lost of arbitration with a CRC byte.
04h	Received a BREAK message or the break message finished.

All bytes transmitted by the C2Sla's transmitter are re-directed back into the receive data buffer register (C2SIARDB). Therefore, the C2SIACCSR register is updated to reflect the status of not only receiving a message from another C2Sla but also transmitting a message to another C2Sla, since bytes that are transmitted are re-directed back and received in the C2Sla's own C2SIARDB register.

The resulting completion code at the end of a normal or in-frame response message will remain unchanged until the first byte of the next message (normal message or in-frame response message) is received and this byte is error free. If this first byte is error free then the completion code register resets. If this first byte is not error free, the C2SIACCSR register does not reset and the RCCIF bit will not get set, and a DMA transfer will not occur. The code must read the C2SIACCSR before the first byte of the next message.

When a break is detected on the bus, the BREAK bit in the completion code is set, and, depending on the situation, the BYTERR bit, BITERR bit, CRCERR, etc., may be set also. If a transmission of a message was in process when a break is detected, the RCCIF bit is set soon after the detection of the break because a message was aborted. When detecting release of break, the BRKIF will be set again, as well as RCCIF. For additional information, see section 5.3, *Receiving BREAK Messages*, on page 20.

If the C2Sla module is transmitting and its receiver overruns the C2SIARDB register, the transmitter continues unaffected. If the C2Sla module was waiting to transmit when its receiver overran and the C2SIACCSR is not read, the transmitter is held off until the C2SIACCSR register is either read by the CPU or DMA.

The RXDMAEN bit (C2SIACCTR.2) is cleared each time the RCCIF flag is set. Therefore, the receive DMA bit needs to be re-enabled (RXDMAEN = 1) or the C2SIARDB register needs to be read by the CPU before the next incoming byte is received in order to prevent overrun. Overrun occurs when a data byte is received and placed in the C2SIARDB register while the register is full. This destroys the first byte since it was over written by the second.

The contents of the C2SIACCSR register are only valid and meaningful when read after the setting of RCCIF (receive completion code interrupt) (C2SIAISR.0).

## 9.7 C2Sla Peripheral Control Register (C2SIACR)

hex	7	6	5	4	3	2	1	0
0x18	LONGBRK	CALEN	LPEN	SOFX	TXDMAEN	RXDMAEN	ESPEN	LPM
	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0

RW = Read/Write, -n = Value after reset

**Bit 7**      **LONGBRK.** Long break symbol.

This bit determines the length of the break symbol when a break is sent out via the TBRK bit. It does not affect the reception of the break symbol.

- 0      =    The break symbol is 300  $\mu$ s long
- 1      =    The break symbol is 768  $\mu$ s long.

**Bit 6**      **CALEN** Auto Calibration Enable.

Allows the C2Sla to calibrate itself with delays from the analog section. When set, the internal calibration logic will adjust itself to compensate for variations within the analog section. When calibrated, the CALEN bits will automatically clear. You must calibrate before transmitting in the 4X mode either automatically or manually via the C2SIACAL register. For additional information, see section 8.6, *Calibration Mode*, on page 27.

- 0      =    C2Sla is calibrated.
- 1      =    C2Sla is still calibrating itself.

**Bit 5**      **LPEN** Loopback Enable.

This bit enables the C2SIALPN output to an active low. This places the external analog circuit into a loopback mode for diagnostics. However, if calibration is required, the CALEN bit also needs to be set.

- 0      =    C2Sla is in normal operating mode.
- 1      =    The C2Sla directs the external transceiver to tie TX directly to RX.

**Bit 4**      **SOFX.** SOF behavior

This bit determines if the SOF will be driven if the C2Sla sees another transmitter during the EOF to the end of SOF period.

- 0      =    C2Sla will drive SOF even when another transmitter is sending an SOF.
- 1      =    C2Sla will not drive SOF when another transmitter is sending an SOF.

**Bit 3 TXDMAEN** Transmit DMA Enable.

Enables the transmit DMA request signal to be generated. For additional information, see section 8, *DMA Interface*, on page 24.

0 = Transmit DMA is not used.  
1 = Transmit DMA is used.

**Bit 2 RXDMAEN** Receive DMA Enable.

Enables the receive DMA request signal to be generated. For additional information, see section 8, *DMA Interface*, on page 24.

0 = Receive DMA is not used.  
1 = Receive DMA is used.

**Bit 1 ESPEN** Emulator Suspend.

Suspends the transmission and reception of data if the CPU is in debug mode. When inactive, the C2S1a will continue normal operation during emulation mode. For additional information, see section 8.5, *Emulation Mode*, on page 26.

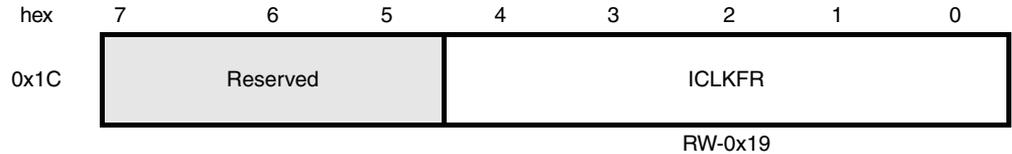
0 = Data is not suspended during emulation mode.  
1 = Data is suspended immediately upon entering emulation mode.

**Bit 0 LPM** Low Power Mode.

When active, the C2S1a enters a power down state and disables the C2S1as internal clocks. This bit is the last C2S1a bit set going into low-power mode and the first bit cleared coming out of low-power mode. Writes to all other bits are disabled when this bit is set. An enabled wake-up condition will generate a interrupt but the code must clear this bit before the C2S1a will function normally. For additional information, see section 8.4, *Low Power Mode*, on page 25.

0 = C2S1a is not in low power mode.  
1 = C2S1a is in low power mode.

## 9.8 Interface Clock Register (C2SIACLK)



RW = Read/Write, -n = Value after reset

**Bits 7-5** **Reserved.** Reads are undefined and writes have no effect.

**Bits 4-0** **ICLKFR** Interface Clock Frequency.

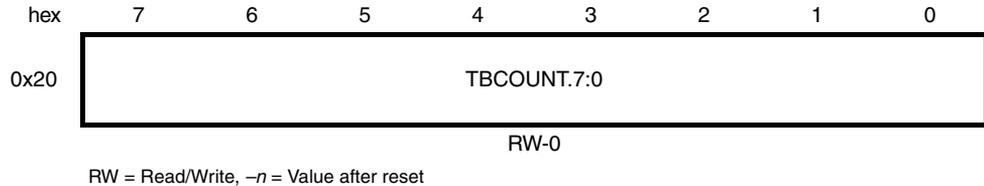
These bits must be set equal to the current C2Sla system clock frequency (ICLK). The following table shows the clock frequencies and their corresponding values for this register.

ICLK (MHz)	ICLKFR (4:0)	ICLK (MHz)	ICLKFR (4:0)
25	0x19	13	0x0D
24	0x18	12	0x0C
23	0x17	11	0x0B
22	0x16	10	0x0A
21	0x15	9	0x09
20	0x14	8	0x08
19	0x13	7	0x07
18	0x12	6	0x06
17	0x11	5	0x05
16	0x10	4	0x04
15	0x0F	3	0x03
14	0x0E	2	0x02
		1	0x01

**Note: Note: Non-Specified ICLK Values**

If an illegal combination is used (such as 0x1F), ICLK is assumed to be 25 MHz.

## 9.9 C2Sla Transmit Byte Counter (C2SIATBC)



### Bits 7-0 **TBCOUNT** Transmit Byte Counter.

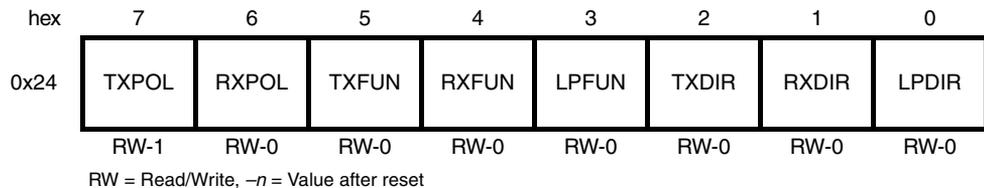
This register contains the number of data bytes to be transmitted. The CRC byte is not included in this number. The transmit byte counter decrements each time the transmit data buffer register (C2SIATDB) is loaded by CPU or DMA.

When the C2SIATBC attempts to decrement past 0 (either from DMA or user software), the C2Sla module will automatically append the required CRC byte, if CRC is used (CRCDIS = 0), to the end of the message.

Since C2SIATBC is an 8-bit register it allows, a maximum of 255 bytes to be sent out in a single message. See the SAE J1850, Class B Data Communications Network Interface specification for details of the J1850 message protocol and maximum J1850 message length.

Transmissions errors will clear the C2SIATBC counter. Any further writes to the C2SIATDB data buffer will generate a transmission overrun and set the TXUOIF bit.

## 9.10 C2Sla Pin Control Register 1 (C2SIAPC1)



A block diagram of the I/O pins controlled by these registers is shown in Figure 2, *C2Sla Block Diagram*, page 4.

### Bit 7 **TXPOL** Transmit Pin Polarity

Determines the polarity of the transmit pin when the transmit pin is configured as a C2Sla pin. An active low pin will place 0 volts to represent the SOF, while an active high pin will place Vcc volts to represent SOF.

- 0 = Transmit pin is active low
- 1 = Transmit pin is active high

**Bit 6 RXPOL Receive Pin Polarity**

Determines the polarity of the receive pin when the receive pin is configured as a C2Sla pin. An active low pin will expect 0 volts to represent a data 1, while an active high pin will expect Vcc volts to represent a SOF.

- 0 = Receive pin is active low
- 1 = Receive pin is active high

**Bit 5 TXFUN Transmit Pin Function**

Determines whether the transmit pin is to be used as a general-purpose I/O pin or as a C2Sla transmit pin. When this pin is inactive, the transmit pin (C2SIATXD pin) may be used as an input or output pin, depending on the value of the TXDIR bit. If in C2Sla mode, the pin characteristics are determined by the C2Sla function bits.

- 0 = Transmit pin is I/O
- 1 = Transmit pin is C2Sla pin

**Bit 4. RXFUN** Receive Pin Function

Determines whether the receive pin is to be used as a general-purpose I/O pin or as a C2S1a receive pin. When this pin is inactive, the receive pin (C2SIARXD pin) may be used as an input or output pin, depending on the value of the RXDIR bit. If in C2S1a mode, the pin characteristics are determined by the C2S1a function bits.

0 = Receive pin is I/O  
1 = Receive pin is C2S1a pin

**Bit 3 LPFUN** Loop-back Pin Function

Determines whether the loop-back pin is to be used as a general-purpose I/O pin or as a C2S1a loop-back pin. When this pin is inactive, the loop-back pin (C2SIALPN pin) may be used as an input or output pin, depending on the value of the LPDIR bit. If in C2S1a mode, the pin characteristics are determined by the C2S1a function bits.

0 = Loop-back pin is I/O  
1 = Loop-back pin is C2S1a pin

**Bit 2 TXDIR** Transmit Direction

Controls the direction of the transmit pin when it is used as a general-purpose I/O pin (TXFUN = 0). If the transmit pin is used as an C2S1a pin (TXFUN = 1), the TXDIR bit has no effect.

0 = Input  
1 = Output

**Bit 1 RXDIR** Receive Direction

Controls the direction of the receive pin when it is used as a general-purpose I/O pin (RXFUN = 0). If the receive pin is used as an C2S1a pin (RXFUN = 1), the RXDIR bit has no effect.

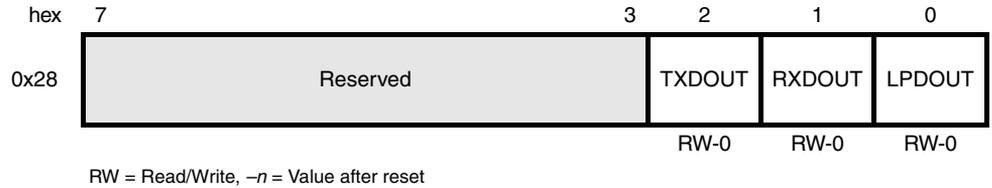
0 = Input  
1 = Output

**Bit 0 LPDIR** Loop-back Direction

Controls the direction of the loop-back pin when it is used as a general-purpose I/O pin (LPFUN = 0). If the loop-back pin is used as an C2S1a pin (LPFUN = 1), the LPDIR bit has no effect.

0 = Input  
1 = Output

## 9.11 C2Sla Pin Control Register 2 (C2SIAPC2)



**Bits 7-3** **Reserved.** Reads are undefined and writes have no effect.

**Bit 2** **TXDOUT** Transmit Data Out.

Only active if the transmit pin is configured as a general-purpose I/O pin (TXFUN = 0) and configured as an output (TXDIR = 1). The value of this bit indicates the value sent to the transmit pin (C2SIATXD pin).

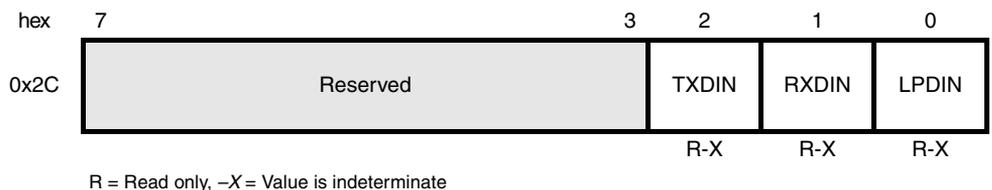
**Bit 1** **RXDOUT** Receive Data Out.

Only active if the receive pin is configured as a general-purpose I/O pin (RXFUN = 0) and configured as an output (RXDIR = 1). The value of this bit indicates the value sent to the receive pin (C2SIARXD pin).

**Bit 0** **LPDOUT** Loop-back Data Out.

Only active if the loop-back pin is configured as a general-purpose I/O pin (LPFUN = 0) and configured as an output (LPDIR = 1). The value of this bit indicates the value sent to the loop-back pin (C2SIALPN pin).

## 9.12 C2Sla Pin Control Register 3 (C2SIAPC3)



**Bits 7-3** **Reserved.** Reads are undefined and writes have no effect.

**Bit 2** **TXDIN** Transmit Data In.

The value of this bit reflects the value on the transmit pin (C2SIATXD pin).

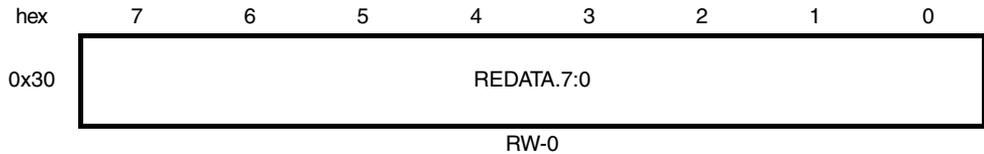
**Bit 1**            **RXDIN** Receive Data In.

The value of this bit reflects the value on the receive pin (C2SIARXD pin).

**Bit 0**            **LPDIN** Loop-back Data In.

The value of this bit reflects the value on the loop-back pin (C2SIALPN pin).

### 9.13 C2Sla Receive Data Emulation Buffer Register(C2SIAEMU)

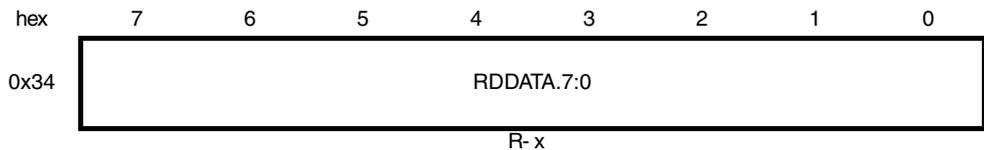


RW = Read/Write, -n = Value after reset

**Bits 7-0**        **REDATA** Receive Data Emulation Buffer Register.

This register is a mirror image of the receive data buffer register (C2SIARDB). Both the C2SIAEMU and the C2SIARDB contain identical values. The only difference between these two registers is that a read from C2SIAEMU will not automatically clear the RBFIF interrupt.

### 9.14 C2Sla Receive Data Buffer Register (C2SIARDB)



RW = Read/Write, -n = Value after reset

**Bits 7-0**        **RDDATA** Receive Data Buffer Register.

This register is a buffer following the receiver’s shift register. Once the shift register has been filled from receiving a byte, its contents are transferred into the C2SIARDB register and the RBFIF flag is set each time the C2SIARDB register is filled.

All data bytes transmitted out of the transmit data buffer register (C2SIATDB) will also be redirected back into the receive data buffer register (C2SIARDB) and will therefore set the RBFIF flag. This may cause confusion, and user software must determine whether the data read from this register is from a transmit or a receive.

Receive DMA transfers use this register or the C2SIACCSR register as their source.

A receive interrupt request will be generated if enabled by setting the RBFIE bit = 1. A read of this register by the CPU will clear RBFIF.

## 9.15 C2Sla Peripheral Control Register (C2SIACAL)



### Bit 7 **ARBTYPE.** Arbitration type.

This bit determines how the C2Sla defines an arbitration error. For all cases if the C2Sla sends a passive and receives an active then arbitration is lost and it is not considered an error. This bit determines what happens when the C2Sla sends an active but receives a passive. This should not happen under normal bus conditions but this bit determines how this special case is handled.

- 0 = A TX error occurs when the C2Sla is sending an active but receiving a passive. TXERR bit is set and ARBIF is not set
- 1 = A loss of arbitration occurs when the C2Sla is sending an active but receiving a passive. ARBIF is set and TXERR bit is not set.

### Bit 6 **0.** Calibration value zero

This bit always reads zero and is not affected by writes.

### Bit 5-0 **CAL.** Calibration value

This 6 bit register contains the expected delay time from the TX to RX of the external transceiver. This compensation time is used when transmitting and in arbitration.

When changing to 4X mode or back to normal mode this register is loaded with a default value. The software can change this value if required by the external transceiver parameters. The default calibration count for the normal mode is 23 and for the 4X mode it is 7.

To determine the correct value for this register, first determine the delay of the external transceiver in the system between inputting a value on the TX pin

and its appearance on the receiver RX pin. Use the formula below to calculate the register value for the desired clock speed.

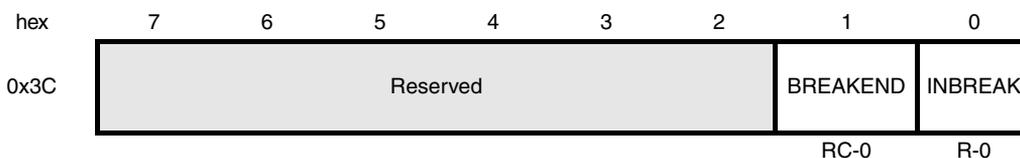
cal register = TX/RX delay  $\mu$ s + 4 iclks + 7  $\mu$ s ; normal mode

cal register = TX/RX delay  $\mu$ s + 4 iclks ; 4X mode

Where iclk is the peripheral bus clock frequency. At higher speeds, this term becomes insignificant. Digitizing errors ranging from 0 to 1  $\mu$ s apply to both calculations and will affect output values.

## 9.16 C2S1a Peripheral Control Register (C2SIABRK)

This register contains the current status of the break symbol and should only be read after seeing the BRKIF bit set. Reading this register clears the BRKIF so reading at other times may cause the code to miss a BRKIF interrupt.



RW = Read/Write, -n = Value after reset

**Bits 7-2** **Reserved.** Reads as undefined and writes have no effect.

**Bit 1** **BREAKEND.** End of Break found.

This bit sets when the receiver has found the end of a break symbol and indicates that the break is over. This bit is cleared by writing a zero or by the reception of a new break symbol.

0 = A break is not ended  
1 = A break has ended since it was last cleared.

**Bit 0** **INBREAK.** In a break symbol now.

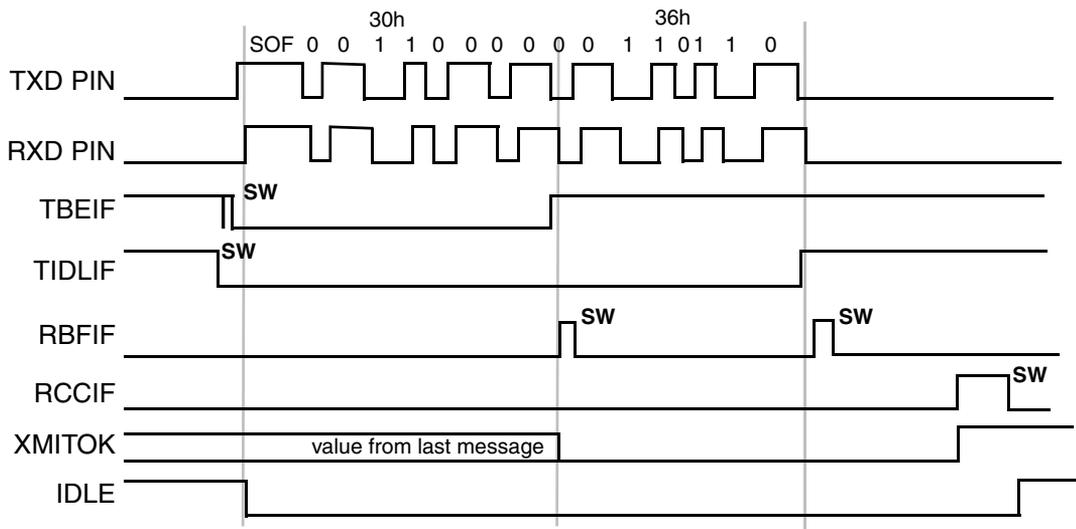
This bit indicates a break currently in progress. This bit goes low just after the ENDBRK bit is set so if both BREAKEND and INBREAK are set, the BREAKEND presides.

0 = A break is not happening now  
1 = A break is currently in progress.

## 10 Timing Examples

Some examples of the flags and signals are presented below that show how various bits react in different situations.

Figure 7. Simple Transmit Example



SW -- Edge cleared by software in interrupt service routine

In this example, the CPU loads two byte at the start of the message and this forces TBEIF to pulse low on the first write and stay low after the second byte is written to C2SIATDB. TIDLIF will go low when the CPU writes to the C2SIATDB and IDLE will go low when the C2SIa detects the RXD pin active.

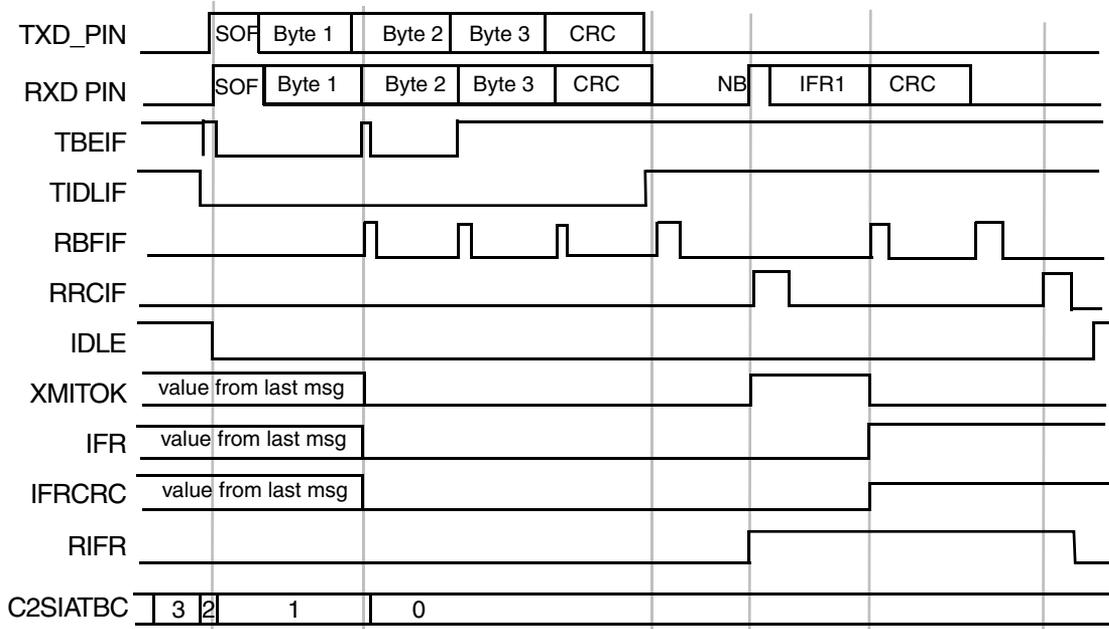
After the first byte is transmitted the TBEIF flag is set and since there is no more data in this example it remains high. The TBEIF interrupt should be disabled after writing the last byte to the C2SIATDB. The RBFIF receive buffer full flag sets when the first byte is received. This is usually around 16  $\mu$ s after the TBEIF sets due to the TX-RX delay in the external transceiver. It goes low when the CPU reads the C2SIARDB.

After the last bit is sent the TIDLIF bit sets. After an end-of-data (EOD) time the completion code is ready and the RCCIF bit set. The CPU read the C2SIACCSR register to clear this bit. After an end-of-frame (EOF) time the IDLE bits sets again.

Also notice that whatever is sent out on the TXD pin is receive back on the RXD pin after a short delay. This delay is dictated by the external J1850 interface device.

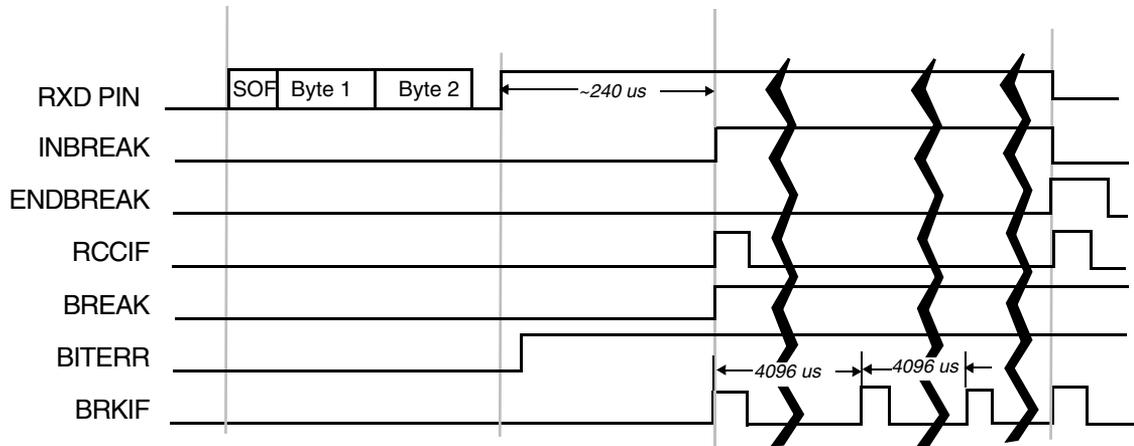
The CRC is disabled in this example.

Figure 8. Send Message and Receive IFR Example



In this example the CPU sends a 3 byte message and expects an IFR from another device. The C2SIACCSR bits from the previous message get cleared after receiving the first good byte of a normal or IFR message. This message has two completion code flags, one at the end of the normal message and one at the end of the IFR message. Although some C2SIACCSR bits set before the RRCIF flag sets, you should only read the completion code when RRCIF flag sets.

Figure 9. Long Break Timing Example

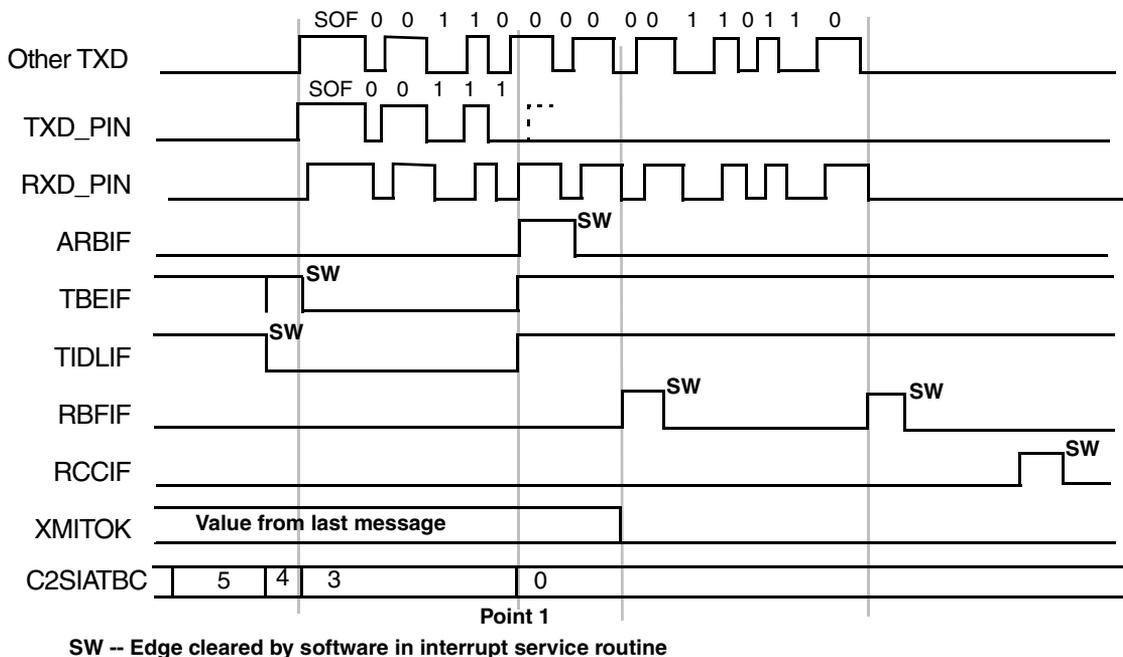


This example shows the signals changing during a long break. The C2S1a will declare a break symbol about 240  $\mu\text{s}$  after the RXD pin transitions from passive to active. The BRKIF will always set at this point. If the C2S1a was in the middle of receiving a message and it was corrupted by the break then the C2S1a will issue a completion code to indicate a corrupted message. In this example the BITERR in the C2SIACCSR was set but the BITERR or CRCERR could also have been set if the break started at a different point. The RCCIF bit sets to indicate the completion code valid. Notice that the BREAK bit in the C2SIACCSR will also have been set at this point.

The BRKIF flag will reset when the C2S1a reads the C2SIABRK register. The BRKIF will set again every 4096  $\mu\text{s}$  until the break symbol ends.

When the break symbol ends, the C2S1a will set the RCCIF to indicate the completion code with the BREAK bit set is ready. The BREAKEND bit will set in the C2SIABRK register. Because of the possible confusion with the BREAK bit in the C2SIACCSR register it will probably be more advantageous to use the C2SIABRK register to determine the break status.

Figure 10. Loss of Arbitration Timing Example



In this example two transmitters attempt to use the bus at the same time. The C2S1a tries to send a passive 1 and is overridden by the other transmitter ending its passive 0 bit and sending the next active bit. At this point the C2S1a declares a loss of arbitration and turns off the transmitter (Point 1). The C2S1a's receiver still gets the incoming byte and processes it like a normal message. The XMITOK bit in the C2SIACCSR shows the value from the last message at this loss of arbitration point but since the RCCIF has not been set, the C2SIACCSR should not be read.

The three interrupt flags ARBIF, TIDLIF and TBEIF will all set at the same time so the user must prioritize between the three. The transmit byte counter clears when arbitration is lost.

## 11 History

The following major changes have been added starting with version 4.0.0 of the C2S1a module.

- 1) Buffered Completion code to prevent mis-labeling good messages as bad.
- 2) Drives during start of frame even when other transmitters are active.
- 3) Detect and identify shorts to ground to prevent locking up the bus.
- 4) Periodic interrupt during Short to power to avoid using other resources.
- 5) End of Break bit added allows simpler software for breaks.
- 6) Calibration register now read/writable so you don't need a special cycle.
- 7) Clear TX counter during any TX error prevents trash on the bus.
- 8) NOISE bit cleared by the proper bit now.
- 9) TX buffer writes disabled during breaks.
- 10) Switching to 4X mode during a character will no longer hang the transmitter.
- 11) IFR status bit added to allow normal/IFR identification of each byte.
- 12) Included C2S1A\_TEST\_EN signal in ASWAKE terms to help with DFT.
- 13) When you drive an active level but see a passive this is now considered a TX error.
- 14) A TXERR bit was added to show TX errors during the first byte.
- 15) Added a bit to give either a GM long or a J1850 short break symbol.
- 16) Added a bit to reset the transmitter without affecting the receiver.
- 17) Removed requirement to do a calibration cycle after changing the 4XMODE bit.
- 18) New bits: TXERROR, RIFR, SHORTGND, TXRESET, LONGBRK, CAL5:0, BREAKEND, INBREAK, SOFX, ARBTYPE.
- 19) Changed bits: (new behavior): BRKIF; (buffered): BITERR, BYTERR, CRCERR, XMITOK.
- 20) ROVR bit changed to set when CCSR changes (cleared or set) not just when it sets

