CIRRUS LOGIC

CL-CD2400/2401

Data Sheet

FEATURES

General

- **Four full-duplex channels**
- **All channels support async, bisync, HDLC, and X.21 protocols**
- **Bit rates to 64 kbit transmit and receive NRZ, NRZI, and Manchester data encoding supported**
- **Digital phase locked loop on each receiver**
- **Independent bit rate generators for transmit and receive**
- **Clock sources can be internal or external**
- **Transmit clock source can be receive DPLL output**
- **Full on-chip DMA controller**
- **Data transfer by DMA or interrupt mechanism selectable per channel per direction**
- **Vectored interrupts with Fair Share™ mechanism for cascading** *(cont. next page)*

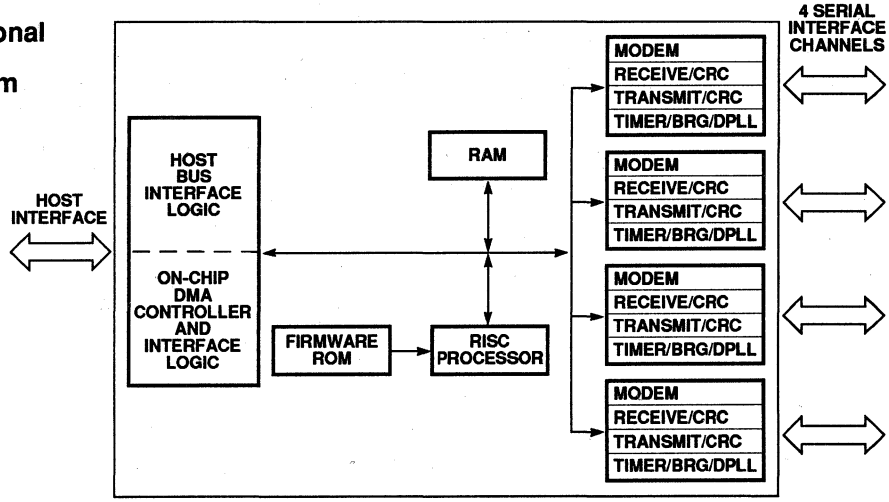
Four-Channel, Multi-Protocol Communications Controller

OVERVIEW

The CL-CD2400 is a synchronous/asynchronous communications controller featuring full on-chip DMA for each channel in each direction. Combining the DMA and protocol controllers enables a more efficient host interface to be provided for both synchronous and asynchronous modes. An efficient vectored interrupt scheme is also provided to minimize host processing overhead and share service fairly among channels and cascaded devices.

By providing the ability to simultaneously use more than one buffer area and to add data to existing buffers, efficient DMA control of data transfer can be achieved.

Functional Block Diagram



NOTE: The CL-CD2400 and CL-CD2401 are functional- and register-compatible. The CL-CD2400 comes in an 84-pin PLCC package and the CL-CD2401, a 100-pin QFP package. Because of the additional pins available to the CL-CD2401, the modem signals are de-multiplexed, making the controller compatible with more applications (e.g., X.21 bis). Unless otherwise stated, all subsequent descriptions for the CL-CD2400 apply to the CL-CD2401.

FEATURES

General (cont.)

- **CRC generation and validation**
- **32 bytes of data FIFO per channel (16-byte transmit FIFO, plus 16-byte receive FIFO)**

Async

- **DMA managed intelligently to minimize host interaction**
  - — transmit block mode buffer
  - — transmit append buffer
  - — receive buffer adjustments for exceptions
  - — receive timeout interrupts
- **Receive good data separated from exceptions**
- **In-band flow control via programmable characters**
- **Out-of-band flow control**
- **Line break detection and generation**
- **Special character recognition and transmission**
- **Receive and transmit timer supported**
- **UNIX® character processing**

Bisync

- **Programmable for ASCII or EBCDIC encoding**
- **Supports transparent bisync**
- **Recognition of all special characters enabling**
  - — block separation
  - — CRC generation/validation without user intervention
- **Chaining of long receive blocks into multiple buffers**
- **Unchaining of multiple buffers into long transmit blocks**
- **Two general timers per channel**

HDLC

- **Address matching of either 2 x 16-bit addresses or 4 x 8-bit addresses**
- **Idle-in flag or mark**
- **CRC generation and validation**
- **Chaining of a long receive frame into multiple buffers**
- **Unchaining of multiple buffers into a long transmit frame**
- **Option to send PAD characters before starting flags**
- **Two general timers per channel**

CONTENTS

* A page number cross-reference has been provided throughout this section for the location of each register.

List of Figures

List of Tables

1. PIN INFORMATION

1.1 Pin Diagram — CL-CD2400

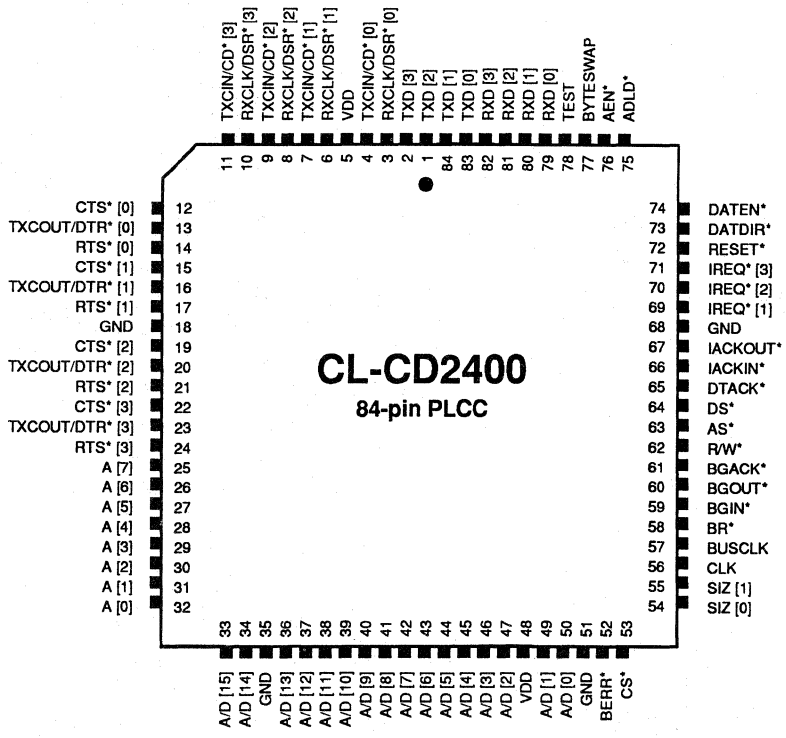


Figure 1–1. CL-CD2400 Pin Diagram

1. PIN INFORMATION *(cont.)*

1.2 Pin Diagram — CL-CD2401

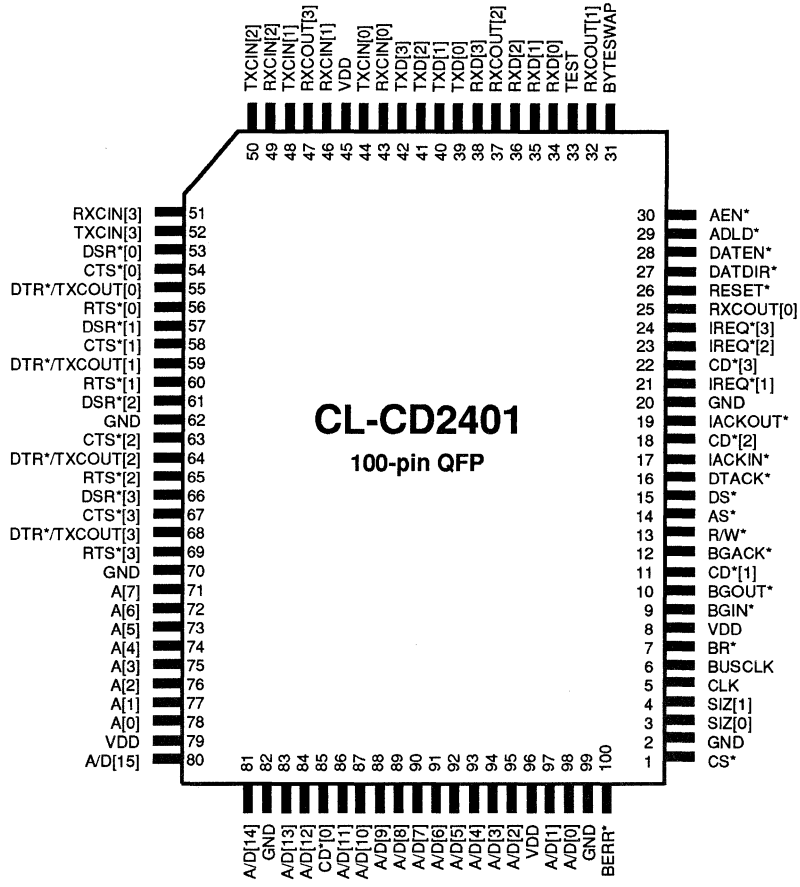


Figure 1–2. CL-CD2401 Pin Diagram

1.3 Pin Functions

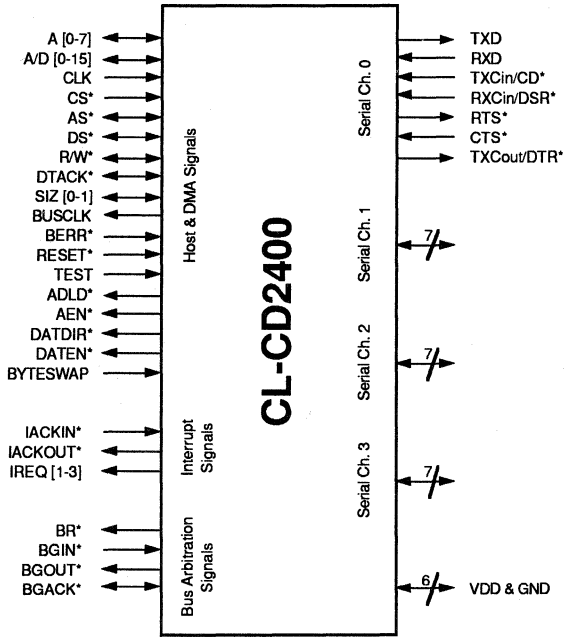


Figure 1–3. CL-CD2400 Pin Functions

1.4 Pin Descriptions

Symbol	Type	Description
CS*	Input	CHIP SELECT* – When low, the CL-CD2400 registers may be read or written by the host processor.
AS*	I/O (tristate)	ADDRESS STROBE* – When the CL-CD2400 is a bus master, this pin is an output which indicates that R/W*, A[0-7], and the externally latched A[8-31] are valid.
DS*	I/O (tristate)	DATA STROBE* – When the CL-CD2400 is not a bus master, this is an input used to strobe data into registers during write cycles and enable data onto the bus during read cycles. When the CL-CD2400 is a bus master, DS* is an output used to control data transfer to and from system memory.
R/W*	I/O (tristate)	READ/WRITE* – When the CL-CD2400 is not a bus master, this pin is an input which determines if a read or write operation is required when the CS* and DS* signals are active. When the CL-CD2400 is a bus master, R/W* is an output and indicates whether a read from or a write to system memory is being performed.
DTACK*	I/O (open drain)	DATA TRANSFER ACKNOWLEDGE* – When the CL-CD2400 is not a bus master, this is an output and indicates to the host when a read or write to the CL-CD2400 is complete. When BR* is driven low by the CL-CD2400, DTACK* is an input which indicates that the system bus is no longer in use. When the CL-CD2400 is a bus master, DTACK* is an input which indicates when system memory read and write cycles are complete.
SIZ[0-1]	I/O (tristate)	SIZE [0-1] – When not the active bus master, these are inputs which determine the size of the operand being read or written by the host. SIZ[1] SIZ[0] 0 1 - Byte * 1 0 - 16 Bit 0 0 - 32 Bit ** 1 1 - 3 Bytes** When the CL-CD2400 is a bus master, this is an output determining the size of the operand being transferred to or from system memory. SIZ[1] SIZ[0] 0 1 - Byte * 1 0 - 16 Bit * See BYTESWAP description ** The CL-CD2400 will drive DTACK* even though the device will not respond to such byte alignment.
IACKIN*	Input	INTERRUPT ACKNOWLEDGE IN* – This input qualified with DS*, and A[0-3] acknowledges CL-CD2400 interrupts.
IACKOUT*	Output	INTERRUPT ACKNOWLEDGE OUT* – This output is driven low during interrupt acknowledge cycles for which no internal interrupt is valid.

1.4 Pin Descriptions *(cont.)*

Symbol	Type	Description
IREQ*[1-3]	I/O (open drain)	INTERRUPT REQUEST* [1-3] – These outputs signal that the CL-CD2400 has a valid interrupt for modem-lead activity (IREQ*[1]), transmit activity (IREQ*[2]), or receive activity (IREQ*[3]).
BR*	Output	BUS REQUEST* – This output is used to signal to the (open drain) host processor or bus arbiter that bus mastership is required by the CL-CD2400.
BGIN*	Input	BUS GRANT IN* – This input indicates that the bus is available after the current bus master relinquishes the bus.
BGOUT*	Output	BUS GRANT OUT* – This output is asserted when BGIN* is low and no internal Bus Request has been made. A daisy-chain scheme of bus arbitration can be formed by connecting BGOUT* to BGIN* of the next device in the chain. If a priority scheme is preferred, Bus Requests must be prioritized externally and Bus Grant routed to the BGIN* of the appropriate device.
BGACK*	I/O (open drain)	BUS GRANT ACKNOWLEDGE* – As an input, this signal is used to determine if another alternate bus master is in control of the bus. As an output, it signals to other bus masters that this device is in control of the bus.
BERR*	Input	BUS ERROR* – If this input becomes active while the CL-CD2400 is a bus master, the current bus cycle will be terminated, the bus relinquished, and an interrupt generated to indicate the error to the host processor.
A[0-7]	I/O (tristate)	ADDRESS [0-7] – When the CL-CD2400 is not a bus master, these pins are inputs used to determine which registers are being accessed, or which interrupt is being acknowledged. When ADLD* is low, A[0-7] output address bits 8 through 15 for external latching. When the CL-CD2400 is a bus master, A[0-7] output the least significant byte of the transfer address.
A/D[0-15]	I/O (tristate)	ADDRESS/DATA [0-15] – When the CL-CD2400 is not a bus master, these pins provide the 16-bit data bus for reading and writing to the CL-CD2400 registers. When ADLD* is low, A/D[0-15] provide the upper address bits for external latching. When the CL-CD2400 is a bus master, A/D[0-15] provide a multiplexed address/data bus for reading and writing to system memory.
ADLD*	Output (tristate)	ADDRESS LOAD* – This is a strobe used to externally latch the upper portion of the system address bus A[8-31]. While ADLD* is low, address bits 16 through 31 are available on A/D[0:15], and address bits 8 through 15 on A[0-7].
AEN*	Output (tristate)	ADDRESS ENABLE* – This output is used to output enable the external address bus drivers during CL-CD2400 DMA cycles.

1.4 Pin Descriptions

Symbol	Type	Description
DATDIR*	Output (tristate)	DATA DIRECTION* – This output is active when either the CL-CD2400 is a bus master, or the CS* Pin is low. It is used to control the external data buffers; when low, the buffers should be enabled in the CL-CD2400 to system bus direction.
DATEN*	Output (tristate)	DATA ENABLE* – This output is active when either the CL-CD2400 is a bus master, or the CS* and DS* Pins are low. It is used to enable the external data bus buffers during host register read/write operations or during DMA operations. For operations on 32-bit buses, this signal needs to be gated with A[1] to select the correct half of the data bus.
CLK	Input	CLOCK – system clock.
BUSCLK	Output	BUS CLOCK – This is the system clock divided by 2 which is used internally to control certain bus operations. This pin is driven low during hardware reset.
RESET*	Input	RESET* – This signal should stay valid for a minimum of 20 ns. The reset state of the CL-CD2400 will be guaranteed at the rising edge of this signal. When RESET* is removed, the CL-CD2400 also performs a software initialization of its registers.
TEST	Input	TEST – In normal operation, this pin should be kept low. For board-level testing purposes, it provides a mechanism for forcing normal output pins to High Impedance Mode. When the TEST Pin is high, the following pins will be in High Impedance Mode: BUSCLK, BGOUT*, IACKOUT*, RXCout[0:3], RTS*[0:3}, DTR*[0:3] and TXD[0:3]. To ensure all CL-CD2400 outputs are high impedance, either of the following two conditions must be met: the RESET* Pin can be driven low, and the TEST Pin driven high; or, the CL-CD2400 is kept in the bus idle state (not accessed for read/write operations nor DMA active), and the TEST Pin is driven high.
RTS*[0-3]	Output	REQUEST TO SEND* [0-3] – This output can be controlled automatically by the CL-CD2400 to indicate that data is being sent on the TXD Pin.
TXCout/DTR* [0-3]	Output	TRANSMIT CLOCK OUT/DATA TERMINAL READY* [0-3] – This output can be controlled automatically by the CL-CD2400 to indicate a programmable threshold has been reached in the receive FIFO. It can also be programmed to output the transmit data clock. Following reset, this pin will be high and will stay high in Clock Mode until the transmit channel is enabled for the first time; after which, it remains active independent of the state of the transmit enable. In all modes, the clock transitions every bit time, even during idle fill in Asynchronous Mode. Data transitions are made on the negative going edge of TXCout.

1.4 Pin Descriptions *(cont.)*

Symbol	Type	Description
RXCout[0-3]	Output	RECEIVE CLOCK OUT [0-3] – This output provides a one-time bit rate clock for the receive data in all modes, except when an input (RXCin) one-time receive clock is used. After reset, this pin will be low until the channel is receive enabled for the first time, after which it remains active, independent of the state of receive enable. When in Asynchronous Mode, the output only transitions while receiving data and not during inter-character fill. The receive data is sampled on the positive-going edge of this clock.
CTS*[0-3]	Input	CLEAR TO SEND* [0-3] – This input can be programmed to control the flow of transmit data, for out-of-band flow control applications.
TXCIN/CD* [0-3]	Input	TRANSMIT CLOCK/CARRIER DETECT* [0-3] – When used as the transmit clock, the state of the pin is still available in the MSVR Register. This signal is demuxed on the CL-CD2400.
RXCIN/DSR* [0-3]	Input	RECEIVE CLOCK/DATA SET READY* [0-3] – When used as the receive clock, the state of the pin is still available in the MSVR Register. When used as DSR*, this input can be programmed to validate receive data.
TXD[0-3]	Output	TRANSMIT DATA [0-3] – Serial data output for each channel.
RXD[0-3]	Input	RECEIVE DATA [0-3] – Serial data input for each channel.
BYTESWAP	Input	This pin alters the byte ordering of data during certain 16-bit transfers and changes the half of the data bus on which byte transfers are made to comply with Intel® or Motorola® processor systems. BYTESWAP does not alter the bus handshake signals. When the BYTESWAP Pin is high, the byte on A/D[0:7] precedes that on A/D[8:15] in a string of transmit or receive bytes; when BYTESWAP is low, A/D[8:15] precedes A/D[0:7]. When the BYTESWAP Pin is high, bytes are transferred on A/D[0:7] when A[0] is low, and on A/D[8:15] when A[0] is high. When BYTESWAP is low, bytes are transferred on A/D[8:15] when A[0] is low, and A/D[0:7] when A[0] is high. A different register map is used, depending on the state of this pin.

BYTESWAP	BYTE ALIGNMENT
0	Motorola byte alignment
1	Intel byte alignment

2. REGISTER TABLE

Registers in the CL-CD2400 are either global or per-channel. The column 'Address Mode' in the Register Map on the following pages defines this attribute for each register. Only one set of global registers exists. The global registers are accessible by the host at any time. Four sets of per-channel registers exist; the set accessible at any one time is determined by the currently active channel number. The channel number is selected by the host in normal (non-interrupt) processing by writing to the Channel Access Register. The channel number in the Channel Access Register remains in force until changed by the host. The channel number is provided automatically by the CL-CD2400 during interrupt service routines and DMA transfers.

In the following list, some register locations appear twice. They have different names and functions for Asynchronous Operation and Synchronous Protocol Operation. See Section 6 of this document for detailed descriptions of all register functions.

2.1 CL-CD2400 Memory Map

2.1.1 Global Registers

		Addr. Mode[1]	INT[2]	MOT[3]	Size	Access	Page No.
Global Firmware Revision Code Register	{GFRCR}	G	82	81	B	R	57
Channel Access Register	{CAR}	G	EC	EE	B	R/W	57

2.1.2 Option Registers

Channel Mode Register	{CMR}	P	18	1B	B	R/W	58
Channel Option Register 1	{COR1}	P	13	10	B	R/W	59
Channel Option Register 2	{COR2}	P	14	17	B	R/W	61
Channel Option Register 3	{COR3}	P	15	16	B	R/W	65
Channel Option Register 4	{COR4}	P	16	15	B	R/W	69
Channel Option Register 5	{COR5}	P	17	14	B	R/W	70
Channel Option Register 6	{COR6}	P	1B	18	B	R/W	71
Channel Option Register 7	{COR7}	P	04	07	B	R/W	72
Special Character Register1	{SCHR1}	P	1C	1F	B	R/W Async	73
Special Character Register2	{SCHR2}	P	1D	1E	B	R/W Async	73
Special Character Register3	{SCHR3}	P	1E	1D	B	R/W Async	73
Special Character Register4	{SCHR4}	P	1F	1C	B	R/W Async	73
Special Character Range low	{SCRl}	P	20	23	B	R/W Async	74
Special Character Range high	{SCRh}	P	21	22	B	R/W Async	74
LNext Character	{LNXT}	P	2D	2E	B	R/W	74
Receive Frame Address Register1	{RFAR1}	P	1C	1F	B	R/W Sync	75
Receive Frame Address Register2	{RFAR2}	P	1D	1E	B	R/W Sync	75
Receive Frame Address Register3	{RFAR3}	P	1E	1D	B	R/W Sync	75
Receive Frame Address Register4	{RFAR4}	P	1F	1C	B	R/W Sync	75
CRC Polynomial Select Register	{CPSR}	P	D4	D6	B	R/W	75

NOTES:

The following are applicable for Sections 2.1.1 through 2.1.7:

1 Address Mode G: Global register – one set is always accessible.
Address Mode P: Per-channel register – four sets, one per channel, accessible via CAR or interrupt context.

2 INT = address for Intel-style processor.

3 MOT = address for Motorola-style processor.

2.1.3 Bit Rate and Clock Option Registers

		Addr. Mode[1]	INT[2]	MOT[3]	Size	Access	Page No.
Receive Baud Rate Period Register	{RBPR}	P	C9	CB	B	R/W	76
Receive Clock Option Register	{RCOR}	P	CA	C8	B	R/W	76
Transmit Baud Rate Period Register	{TBPR}	P	C1	C3	B	R/W	77
Transmit Clock Option Register	{TCOR}	P	C2	C0	B	R/W	77

2.1.4 Channel Command and Status Registers

Channel Command Register	{CCR}	P	10	13	B	R/W	78
Special Transmit Command Register	{STCR}	P	11	12	B	R/W Sync	80
Channel Status Register (Sync Modes)	{CSR}	P	19	1A	B	R	81
Modem Signal Value Registers	{MSVR-RTS}	P	DC	DE	B	R/W	84
	{MSVR-DTR}	P	DD	DF	B	R/W	84

2.1.5 Interrupt Registers

		Addr. Mode[1]	INT[2]	MOT[3]	Size	Access	Page No.
Local Interrupt Vector Register	{LIVR}	P	0A	09	B	R/W	85
Interrupt Enable Register	{IER}	P	12	11	B	R/W	86
Local Interrupting Channel Register	{LICR}	P	25	26	B	R/W	87
Stack Register	{STK}	G	E0	E2	B	R	88

2.1.5.1 Receive Interrupt Registers

Receive Priority Interrupt Level Register	{RPILR}	G	E3	E1	B	R/W	89
Receive Interrupt Register	{RIR}	G	EF	ED	B	R	89
Receive Interrupt Status Register	{RISR}	G	8A	88	W	R/W	90
Receive Interrupt Status Register low	{RISRl}	G	8A	89	B	R	90
Receive Interrupt Status Register high	{RISRh}	G	8B	88	B	R	93
Receive FIFO Output Count	{RFOC}	P	33	30	B	R	93
Receive Data Register	{RDR}	G	F8	F8	B	W	93
Receive End Of Interrupt Register	{REOIR}	G	87	84	B	W	94

2.1.5.2 Transmit Interrupt Registers

Transmit Priority Interrupt Level Register	{TPILR}	G	E2	E0	B	R/W	95
Transmit Interrupt Register	{TIR}	G	EE	EC	B	R	95
Transmit Interrupt Status Register	{TISR}	G	89	8A	B	R	96
Transmit FIFO Transfer Count	{TFTC}	G	83	80	B	R	97
Transmit Data Register	{TDR}	G	F8	F8	B	W	97
Transmit End Of Interrupt Register	{TEOIR}	G	86	85	B	W	98
Modem Priority Interrupt Level Register	{MPILR}	G	E1	E3	B	R/W	99
Modem Interrupt Register	{MIR}	G	ED	EF	B	R	99
Modem (/Timer) Interrupt Status Register	{MISR}	G	88	8B	B	R	100
Modem End Of Interrupt Register	{MEOIR}	G	85	86	B	W	100

2.1.6 DMA Registers

		Addr. Mode[1]	INT[2]	MOT[3]	Size	Access	Page No.
DMA Mode Register (write only)	{DMR}	G	F4	F6	B	W	101
Bus Error Retry Count	{BERCNT}	G	8D	8E	B	R/W	101
DMA Buffer Status	{DMABSTS}	P	1A	19	B	R	102

2.1.6.1 DMA Receive Registers

A Receive Buffer Address Lower	{ARBADRL}	P	40	42	W	R/W	103
A Receive Buffer Address Upper	{ARBADRU}	P	42	40	W	R/W	103
B Receive Buffer Address Lower	{BRBADRL}	P	44	46	W	R/W	103
B Receive Buffer Address Upper	{BRBADRU}	P	46	44	W	R/W	103
A Buffer Receive Byte Count	{ARBCNT}	P	48	4A	W	R	103
B Buffer Receive Byte Count	{BRBCNT}	P	4A	48	W	R	103
A Receive Buffer Status	{ARBSTS}	P	4C	4F	B	R	103
B Receive Buffer Status	{BRBSTS}	P	4D	4E	B	R	103
Receive Current Buffer Address Lower	{RCBADRL}	P	3C	3E	W	R	104
Receive Current Buffer Address Upper	{RCBADRU}	P	3E	3C	W	R	104

2.1.6.2 DMA Transmit Registers

A Transmit Buffer Address Lower	{ATBADRL}	P	50	52	W	R/W	104
A Transmit Buffer Address Upper	{ATBADRU}	P	52	50	W	R/W	104
B Transmit Buffer Address Lower	{BTBADRL}	P	54	56	W	R/W	104
B Transmit Buffer Address Upper	{BTBADRU}	P	56	54	W	R/W	104
A Buffer Transmit Byte Count	{ATBCNT}	P	58	5A	W	R/W	104
B Buffer Transmit Byte Count	{BTBCNT}	P	5A	58	W	R/W	104
A Transmit Buffer Status	{ATBSTS}	P	5C	5F	B	R/W	105
B Transmit Buffer Status	{BTBSTS}	P	5D	5E	B	R/W	105
Transmit Current Buffer Address Lower	{TCBADRL}	P	38	3A	W	R	106
Transmit Current Buffer Address Upper	{TCBADRU}	P	3A	38	W	R	106

2.1.7 Timer Registers

		Addr. Mode[1]	INT[2]	MOT[3]	Size	Access	Page No.
Timer Period Register	{TPR}	G	D8	DA	B	R/W	107
Receive Timeout Period Register	{RTPR}	P	26	24	W	R/W Async	108
Receive Timeout Period Register low	{RTPRl}	P	26	25	B	R/W Async	108
Receive Timeout Period Register high	{RTPRh}	P	27	24	B	R/W Async	108
General Timer 1	{GT1}	P	28	2A	W	R Sync	108
General Timer 1 low	{GT1l}	P	28	2B	B	R Sync	108
General Timer 1 high	{GT1h}	P	29	2A	B	R Sync	108
General Timer 2	{GT2}	P	2A	29	B	R Sync	108
Transmit Timer Register	{TTR}	P	2A	29	B	R Async	108

2.2 Register Definitions

2.2.1 General Global Registers

Global Firmware Revision Code Register {GFRCR} 82 81 B R

Firmware Revision code

Channel Access Register {CAR} EC EE B R/W

0	0	0	0	0	0	C1	C0

2.2.2 Option Registers

Channel Mode Register {CMR} 18 1B B R/W

RxMode	TxMode	0	0	0	chmd2	chmd1	chmd0

Channel Option Register 1 {COR1} 13 10 B R/W

HDLC

AFLO	ClrDet	AdMde1	AdMde0	Flags3	Flags2	Flags1	Flags0

Async/Bisync/X.21

Parity	ParM1	ParM0	Ignore	Chl3	Chl2	Chl1	Chl0

Channel Option Register 2 {COR2} 14 17 B R/W

HDLC

0	FCS	0	CRCNinv	0	RtsAO	CtsAE	DsrAE

Async

IXM	TxIBE	ETC	0	RLM	RtsAO	CtsAE	DsrAE

Bisync

LRC	BCC	EBCDIC	CRCNinv	0	0	0	0

X.21

0	0	0	ETC	0	0	0	0

Channel Option Register 3 {COR3} 15 16 B R/W

HDLC/Bisync

sndsync	Alt1	res	CRC	Idle	pad2	pad1	pad0

Async

ESCDE	RNGDE	FCT	SCDE	0	Stop2	Stop1	Stop0

X.21

0	SSDE	StrpSyn	SCDE	0	0	0	0

Channel Option Register 4 {COR4} 16 15 B R/W

DSRzd	CDzd	CTSzd	0	FIFO Threshold

Channel Option Register 5 {COR5} 17 14 B R/W

DSRod	CDod	CTSod	0	Rx Flow Control Threshold

2.2.2 Option Registers *(cont.)*

Channel Option Register 6 {COR6} 1B 18 B R/W
Async

IgnCR	ICRNL	INLCR	IgnBrk	NBrkInt	ParMrk	INPCK	ParInt

Channel Option Register 7 {COR7} 04 07 B R/W
Async

IStrip	LNE	FCErr	0	0	0	ONLCR	OCRNL

Special Character Register1	{SCHR1}	1C	1F	B	R/W Async
Special Character Register2	{SCHR2}	1D	1E	B	R/W Async
Special Character Register3	{SCHR3}	1E	1D	B	R/W Async
Special Character Register4	{SCHR4}	1F	1C	B	R/W Async
Special Character Range low	{SCRl}	20	23	B	R/W Async
Special Character Range high	{SCRh}	21	22	B	R/W Async
LNext Character	{LNXT}	2D	2E	B	R/W Async
Receive Frame Address Register1	{RFAR1}	1C	1F	B	R/W Sync
Receive Frame Address Register2	{RFAR2}	1D	1E	B	R/W Sync
Receive Frame Address Register3	{RFAR3}	1E	1D	B	R/W Sync
Receive Frame Address Register4	{RFAR4}	1F	1C	B	R/W Sync

CRC Polynomial Select Register {CPSR} D4 D6 B R/W

res	res	res	res	res	res	res	poly

2.2.3 Bit Rate and Clock Options Registers

Receive Baud Rate Period Register {RBPR} C9 CB B R/W

Receive Baud Rate Period (Divisor)

Receive Clock Option Register {RCOR} CA C8 B R/W

TLVal	res	dpllEn	Dpllmd1	Dpllmd0	ClkSel2	ClkSel1	ClkSel0

Transmit Baud Rate Period Register {TBPR} C1 C3 B R/W

Transmit Baud Rate Period (Divisor)

Transmit Clock Option Register {TCOR} C2 C0 B R/W

ClkSel2	ClkSel1	ClkSel0	res	Ext-1X	res	LLM	res

2.2.4 Channel Command and Status Registers

Channel Command Register {CCR} 10 13 B R/W

0	ClrCh	InitCh	RstAll	EnTx	DisTx	EnRx	DisRx
1	ClrT1	ClrT2	0	0	0	0	0

Special Transmit Command Register {STCR} 11 12 B R/W

0	AbortTx	AppdCmp	0	SndSpc	SSPC2	SSPC1	SSPC0

Channel Status Register {CSR} 19 1A B R

HDLC

RxEn	RxFlag	RxFrame	RxMark	TxEn	TxFlag	TxFrame	TxMark

Async

RxEn	RxFloff	RxFlon	0	TxEn	TxFloff	TxFlon	0

Bisync

RxEn	RxITB	RxFrme	0	TxEn	TxITB	TxFrme	0

X.21

RxEn	0	RxSpc	0	TxEn	0	TxSpc	0

Modem Signal Value Registers {MSVR} R/W

Modem Signal Value Register {MSVR-RTS} DC DE B R/W

{MSVR-DTR} DD DF B R/W

DSR/RxCk	CD/TxCk	CTS	DTRopt	res	PortID	DTR	RTS

2.2.5 Interrupt Registers

Local Interrupt Vector Register {LIVR} 0A 09 B R/W

X	X	X	X	X	X	IT1	IT0

Interrupt Enable Register {IER} 12 11 B R/W

Mdm	0	RET	0	RxD	TIMER	TxMpty	TxD

Local Interrupting Channel Register {LICR} 25 26 B R/W

X	X	X	X	C1	C0	X	X

Interrupt Stack Register {STK} E0 E2 B R

CLvl [1]	MLvl [1]	TLvl [1]	0	0	TLvl [0]	MLvl [0]	CLvl [0]

2.2.5.1 Receive Interrupt Registers

Receive Priority Interrupt Level Register {RPILR} E3 E1 B R/W
Receive Interrupt Register {RIR} EF ED B R

Ren	Ract	Reoi	0	Rvct [1]	Rvct [0]	Rcn [1]	Rcn [0]

Receive Interrupt Status Register {RISR} 8A 88 W R
Receive Interrupt Status Register low {RISRl} 8A 89 B R

HDLC

0	EOF	rxabt	CRC	OE	ResInc	0	ClrDct

Async

Timeout	SCdet2	SCdet1	SCdet0	OE	PE	FE	Break

Bisync

0	EOF	rxabt	CRC	OE	0	0	0

X.21

LVal	SCdet2	SCdet1	SCdet	OE	PE	0	LChg

Receive Interrupt Status Register high {RISRh} 8B 88 B R

Berr	EOF	EOB	0	BA/BB	0	0	0

Receive FIFO Output Count {RFOC} 33 30 B R

0	0	0	RxCt4	RxCt3	RxCt2	RxCt1	RxCt0

Receive Data Register {RDR} F8 F8 B R

D7	D6	D5	D4	D3	D2	D1	D0

Receive End Of Interrupt Register {REOIR} 87 84 B W

TermBuff	DiscExc	SetTm2	SetTm1	Notransf	Gap2	Gap1	Gap0

2.2.5.2 Transmit Interrupt Registers

Transmit Priority Interrupt Level Register {TPILR} E2 E0 B R/W
Transmit Interrupt Register {TIR} EE EC B R

Ten	Tact	Teoi	0	Tvct [1]	Tvct [0]	Tcn [1]	Tcn [0]

Transmit Interrupt Status Register {TISR} 89 8A B R

Berr	EOF	EOB	UE	BA/BB	0	TxEmpty	0

Transmit FIFO Transfer Count {TFTC} 83 80 B R

0	0	0	TxCt4	TxCt3	TxCt2	TxCt1	TxCt0

Transmit Data Register {TDR} F8 F8 B W

D7	D6	D5	D4	D3	D2	D1	D0

Transmit End Of Interrupt Register {TEOIR} 86 85 B W

TermBuff	EOF	SetTm2	SetTm1	Notransf	0	0	0

Modem Priority Interrupt Level Register {MPILR} E1 E3 B R/W
Modem Interrupt Register {MIR} ED EF B R

Men	Mact	Meoi	0	Mvct [1]	Mvct [0]	Mcn [1]	Mcn [0]

Modem (/Timer) Interrupt Status Register {MISR} 88 8B B R

DSRChg	CDChg	CTSChg	res	res	res	Timer 2	Timer 1

Modem End Of Interrupt Register {MEOIR} 85 86 B W

0	0	SetTm2	SetTm1	0	0	0	0

2.2.6 DMA Registers

DMA Mode Register (Write Only) {DMR} F4 F6 F8 W

0	0	0	0	ByteDMA	0	0	0

Bus Error Retry Count {BERCNT} 8D 8E B R/W

DMA Buffer Status {DMABSTS} 1A 19 B R

TDAlign	RstApd	CrtBuf	Append	Ntbuf	Tbusy	Nrbuf	Rbusy

2.2.6.1 DMA Receive Registers

A Receive Buffer Address Lower	{ARBADRL}	40	42	W	R/W
A Receive Buffer Address Upper	{ARBADRU}	42	40	W	R/W
B Receive Buffer Address Lower	{BRBADRL}	44	46	W	R/W
B Receive Buffer Address Upper	{BRBADRU}	46	44	W	R/W
A Buffer Receive Byte Count	{ARBCNT}	48	4A	W	R
B Buffer Receive Byte Count	{BRBCNT}	4A	48	W	R
A Receive Buffer Status	{ARBSTS}	4C	4F	B	R
B Receive Buffer Status	{BRBSTS}	4D	4E	B	R

Receive Buffer Status (8-bit) {A/B-RBSTS}

Berr	EOF	EOB	0	0	0	0	2400own

Receive Current Buffer Address Lower {RCBADRL} 3C 3E W R

Receive Current Buffer Address Upper {RCBADRU} 3E 3C W R

Receive Buffer Count (16-bit) {A/B-RBCNT}

2.2.6.2 DMA Transmit Registers

A Transmit Buffer Address Lower	{ATBADRL}	50	52	W	R/W
A Transmit Buffer Address Upper	{ATBADRU}	52	50	W	R/W
B Transmit Buffer Address Lower	{BTBADRL}	54	56	W	R/W
B Transmit Buffer Address Upper	{BTBADRU}	56	54	W	R/W
A Buffer Transmit Byte Count	{ATBCNT}	58	5A	W	R/W
B Buffer Transmit Byte Count	{BTBCNT}	5A	58	W	R/W
A Transmit Buffer Status	{ATBSTS}	5C	5F	B	R/W
B Transmit Buffer Status	{BTBSTS}	5D	5E	B	R/W

Berr	EOF	EOB	0	Append	0	INTR	2400own

Transmit Current Buffer Address Lower	{TCBADRL}	38	3A	W	R
Transmit Current Buffer Address Upper	{TCBADRU}	3A	38	W	R

2.2.7 Timer Registers

Timer Period Register	{TPR}	D8	DA	B	R/W
Binary Value					

Receive Timeout Period Register	{RTPR}	26	24	W	R/W Async
Receive Timeout Period Register low	{RTPRl}	26	25	B	R/W Async
Receive Timeout Period Register high	{RTPRh}	27	24	B	R/W Async
General Timer 1	{GT1}	28	2A	W	R Sync
General Timer 1 low	{GT1l}	28	2B	B	R Sync
General Timer 1 high	{GT1h}	29	2A	B	R Sync
General Timer 2	{GT2}	2A	29	B	R Sync
Transmit Timer Register	{TTR}	2A	29	B	R Async

3. BIT RATE GENERATION AND DATA ENCODING

3.1 BRG and DPLL Operation

Data clocks are generated in the CL-CD2400 by feeding one of a number of clock sources into a programmable divider. The clock source and divisor are programmable separately for each channel and direction by the user. Clock options are programmed in the Transmit Clock Option Register and the Receive Clock Option Register. The divisors are programmed in the Transmit Bit Rate Period Register and the Receive Bit Rate Period Register. The possible clock sources are:

Transmit:

1. Clk 0 – CLK input/8
2. Clk 1 – CLK input/32
3. Clk 2 – CLK input/128
4. Clk 3 – CLK input/512
5. Clk 4 – CLK input/2048
6. TXCLK Pin
7. Receive bit clock

Receive:

1. Clk 0 – CLK input/8
2. Clk 1 – CLK input/32
3. Clk 2 – CLK input/128
4. Clk 3 – CLK input/512
5. Clk 4 – CLK input/2048
6. RXCLK Pin

The CLK input is nominally 20 MHz.

The divisor can be programmed for values from 1 to 255. To maximize the accuracy of edge detection in Asynchronous and DPLL Modes, the highest frequency clock and largest divisor combination should be selected.

An external clock input may be used, and it may be at a multiple of the desired bit rate. If so, the appropriate divisor value must be loaded into the Bit Rate Period Register. If the external clock is at the desired bit rate, (1x clock) a value of 01h must be loaded into the associated Bit Rate Period Register.

The receive bit rate generator can also be programmed to act as a Digital Phase Locked Loop (DPLL). In that mode, the clock select and divisor are programmed to be as near as possible to the nominal receive bit rate. Clock Phase adjustments are made by the DPLL logic to lock to the incoming data stream. The receive bit clock is an optional input to the transmitter. This makes it possible to use the DPLL derived clock to synchronize the transmit data stream.

Table 3.0 shows examples for programming standard bit rates. The value to be loaded to set a given bit rate is determined by the following equation:

$$\text{Bit Rate divisor} = \left[\frac{\text{Frequency of chosen clock source}}{\text{Desired Bit Rate}} \right] - 1$$

The above equation will, in general, yield a non-integer result. The nearest integer value, along with the clock source, is the optimum choice for that bit rate. The value loaded in the period register must be that integer expressed as an 8-bit binary value. The bit rate error is the difference between the integer value and the ideal value, expressed as a percentage.

Example 1: Illustrates programming the bit rate generator at 19.2 kbps using internal clock, at a system clock frequency of 20 MHz.

Divisor loaded into R(T)BPR = 129 or 81H
Value loaded into R(T)COR = 00H, to select clk0

Example 2: Illustrates programming the bit rate generator at 56000 bps using external clock. Again, the system clock frequency is assumed to be at 20 MHz.

The user provides a 1.25 MHz clock on the RxCin or TxCin Pin.
Divisor loaded into R(T)BPR = 21 or 15H
Value loaded into RCOR = 06H, to select External Clock Mode
Value loaded into TCOR = C0H, to select External Clock Mode

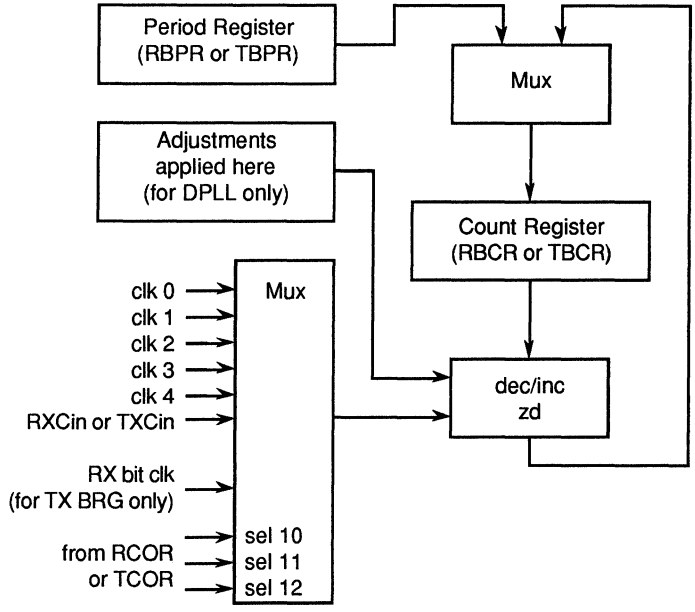


Figure 3–0. BRG/DPLL

Table 3–0. Data Clock Selection Using the Bit Rate Generators

	Clock freq. = 20 MHz			*Clock freq. = 19.66 MHz*		
Bit Rate	**Divisor**	**Clock**	**Error(%)**	**Divisor**	**Clock**	**Error(%)**
50	C2	clk4	-0.16	BF	clk4	+0.00
110	58	clk4	-0.25	56	clk4	-0.31
150	40	clk4	+0.16	3F	clk4	+0.00
300	81	clk3	+0.16	7F	clk3	+0.00
600	40	clk3	+0.16	3F	clk3	+0.00
1200	81	clk2	+0.16	7F	clk2	+0.00
2400	40	clk2	+0.16	3F	clk2	+0.00
3600	AD	clk1	-0.22	AA	clk1	-0.20
4800	81	clk1	+0.16	7F	clk1	+0.00
7200	56	clk1	-0.23	54	clk1	+0.39
9600	40	clk1	+0.16	3F	clk1	+0.00
19200	81	clk0	+0.16	1B	clk0	+0.00
38400	40	clk0	+0.16	3F	clk0	+0.00
56000	2C	clk0	-0.79	2B	clk0	-0.26
64000	26	clk0	+0.16	25	clk0	+1.05

	Clock freq. = 20.48 MHz			*Clock freq. = 18.432 MHz*		
Bit Rate	**Divisor**	**Clock**	**Error(%)**	**Divisor**	**Clock**	**Error(%)**
9600	42	clk1	-0.51	EF	clk0	0.00
19200	84	clk0	+0.25	77	clk0	0.00
38400	42	clk0	-0.51	3B	clk0	0.00
56000	2D	clk0	-0.63	28	clk0	+0.36
64000	27	clk0	0.00	23	clk0	0.00

	Clock freq. = 20.608 MHz			*Clock freq. = 19.712 MHz*		
Bit Rate	**Divisor**	**Clock**	**Error(%)**	**Divisor**	**Clock**	**Error(%)**
19200	85	clk0	+0.12	7F	clk0	+0.26
38400	42	clk0	+0.12	3F	clk0	+0.26
56000	2D	clk0	0.00	2B	clk0	0.00
64000	27	clk0	+0.64	26	clk0	-1.30

	Clock freq. = 17.92 MHz		
Bit Rate	**Divisor**	**Clock**	**Error(%)**
19200	74	clk0	-0.29
38400	39	clk0	+0.57
56000	27	clk0	0.00
64000	22	clk0	0.00

NOTE: All divisors are in hexadecimal.

Transmit and receive data can be encoded/decoded in NRZ, NRZI or Manchester formats. For NRZI, at the start of transmission, a learning data stream of contiguous zeros achieves bit synchronization; for Manchester, an alternating pattern of ones and zeros is required.

NRZ, NRZI, Manchester – Data encoding schemes used in various synchronous protocols. In NRZ, the signal condition represents the data type, high for a logic 1 and low for a logic 0. In NRZ & NRZI type of encoding, the transitions of the data stream occur at the beginning of the bit cell. In NRZI, the signal condition switches to the opposite state to send a binary 0. In Manchester encoding, the transitions are always in the middle of the bit cell. A high-to-low transition is made to send a logic 1, and a low-to-high transition to send a logic 0. The timing diagrams below illustrate the encoding method. The data bits are 0110010.

Example 3: This example illustrates programming the Digital Phase locked loop at 64 kbps in NRZI Mode using the internal clock, at a system clock frequency of 20 MHz.

Divisor loaded into RCOR = 38 or 26h
Value loaded into RCOR = 28h, to enable the DPLL, NRZI framing and select clk0.

Example 4: This example illustrates programming the Digital Phase locked loop in the x1 External Clock Mode, with Manchester encoding.

Divisor loaded into RBPR = 01h, to enable x1 external clock
Value loaded into RCOR = 36h, to enable the DPLL, select Manchester framing, and external clock.

When using an n-times external clock, the highest possible clock frequency and largest divisor combination is recommended. The frequency of an external clock should be less than the system CLK input divided by 16, (i.e., for 20 MHz operation, the data clock should be less than 1.25 MHz). Note that R(T)BPR is an 8-bit register; therefore, the largest divisor value is 255.

The equation to compute the divisor value is

$$\text{Bit Rate divisor} = \left[\frac{\text{Frequency of external clock source}}{\text{Desired Bit Rate}} \right] - 1$$

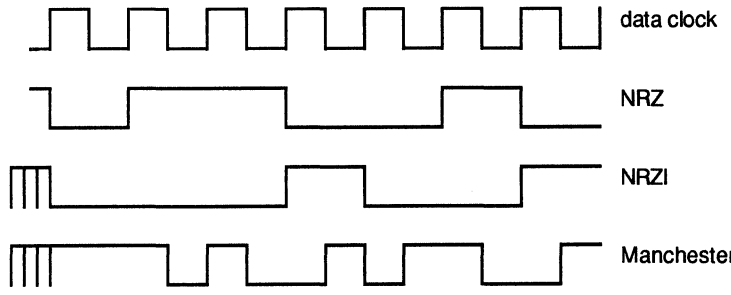


Figure 3–1. Data Encoding

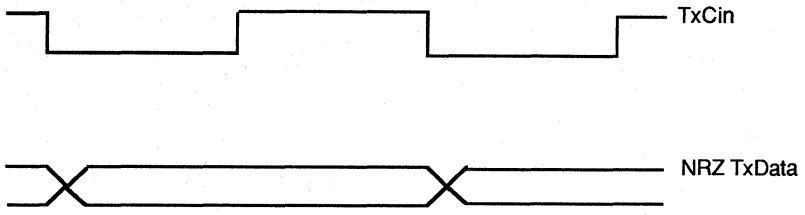


Figure 3–2. Transmit Data With External Clock In

When using external receive clock in the Receive Mode, data is sampled on the low-to-high going edge of RXCin.

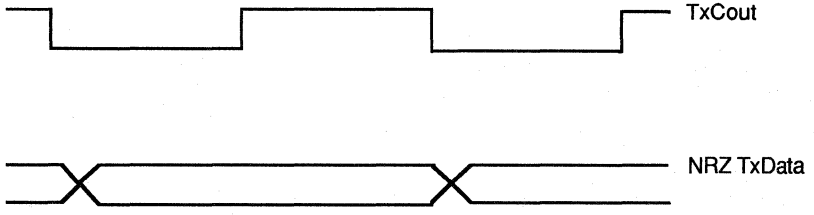


Figure 3–3. Transmit Data With External Clock Out

Table 3–1. Data Clock Selection Using External Clock

Bit Rate	*Clock = 20 MHz* Ext. clk. freq.	Divisor (hex)
50	9.765 KHz	C2
110	9.765 KHz	57
150	9.765 KHz	40
300	39.062 KHz	81
600	39.062 KHz	40
1200	156.25 KHz	81
2400	156.25 KHz	40
3600	625.0 KHz	EF
4800	625.0 KHz	81
7200	1.25 MHz	AC
9600	1.25 MHz	81
19200	1.25 MHz	40
38400	1.25 MHz	1F
56000	1.25 MHz	15
64000	1.25 MHz	12

3.2 Timer Operation

Each channel has two timers, one 16-bit (Timer 1) and one 8-bit (Timer 2). Their operation and programming are different in Synchronous and Asynchronous Modes. There is a global Timer Period Register, TPR (8-bit) which provides a timer prescaler value for the timers clock source which is clocked by system clk/2048.

system clk/2048/{TPR} ---> clock source for timers (max period = 1 ms)

To maintain timer accuracy, the TPR should not be programmed to a value less than 0aH.

3.2.1 Timers in Synchronous Protocols

In synchronous protocols, the timers have no special significance to the CL-CD2400. They are started by host commands or by interrupts generated by the CL-CD2400.

General Timers 1 and 2 can be started in two ways:

1. By loading a new value to GT1 or GT2, if the timer is not running.

2. By setting either the SetTm1 or SetTm2 bits in the relevant End Of Interrupt Register, when terminating a modem, transmit or receive interrupt. In this case, the value to load in GT1h or GT2 should be written to the relevant Interrupt Status Register (MISR, TISR, RISR) prior to writing to the End of Interrupt Register.

The timers can be disabled via a command through the Channel Command Register CCR.

3.2.2 Timers in Asynchronous Protocol Receive Timer

The receive timer is restarted from the value programmed in RTPR every time a character is loaded to the FIFO, or receive data is read by the host. When the timer reaches 0 and data remains in the FIFO, the data is transferred to the host (either via DMA or interrupt control). If the timer reaches 0 and there is no data in the receive FIFO, an optional timer interrupt is generated (IER, Bit 5). If the value in RTPR is zero, the timer is disabled.

Transmit Timer

The Transmit Timer Register (TTR) is used only if embedded transmit command is enabled. See COR2 Bit Mask, found in Section 6, for a description of embedded transmit command, as well as a description in Section 4.3.3. The Delay Transmit command specifies delay period loaded in TTR, and no further transmit operations are performed until this timer reaches 0 (the current state of the line is held either 0 for send break or 1 for inter-character fill).

4. PROTOCOL PROCESSING

4.1 HDLC Processing

4.1.1 HDLC Transmit Mode

The transmitter can be programmed to idle in either flag (01111110) or mark (continuous 1's) mode via Idle Bit in Channel Option Register 3 (COR3). When idle in mark is selected, frame transmission can be programmed to be prepended by a programmable number of pad characters and a programmable number of flags. The pad character can be selected as either 00 or AA; the pad characters allow the remote receivers Phase Locked Loop to synchronize quickly to the data. When NRZI encoding is used, the 00 character guarantees a transition every bit time; for Manchester encoding, AA guarantees exactly one transition per bit time.

If the transmitter is idling in mark, frame transmission is started when data is made available to the transmitter, either via the Transmit Data Register (TDR) or a DMA buffer. First the programmable number of pad characters will be transmitted, and then the programmable number of flag characters. Data characters will then be transmitted and a CRC value accumulated using each data character.

When end of frame status is passed to the CL-CD2400 via the TEOIR or the A/BTBSTS, and the remaining data is transmitted, the CRC and a closing flag will be appended to the frame. If a new frame is available immediately, the correct number of opening flags will be transmitted and data transmission started. If data is not available, the line will be returned to its idle condition.

If data underrun occurs, the CL-CD2400 will not append a CRC, but will abort the transmission by sending eight continuous 1's, and then revert to the idle condition. An underrun interrupt will be generated, and if interrupt transfer is being used, the CPU should provide an EOF response in TEOIR. If DMA Transfer Mode is being used, the CL-CD2400 will discard DMA buffers until an EOF buffer is found; transmission will then resume from the next buffer. This ensures correct operation when a multiple buffer frame underruns.

When programmed in NRZI Mode and idle in mark, after the closing flag and the first eight ones are transmitted, the transmit data line is sampled to determine if it is a logic high or low. If it is low, an extra zero is transmitted to force the line to be a logic high.

When Idle in Flag Mode is selected, the send pad and opening number of flags have no significance; transmission will be started when data is first made available in the FIFO. If no data underrun occurs, the frame is terminated normally with a CRC, and then continuous flags are generated. If an underrun does occur, then no CRC is appended, eight ones are transmitted, and then continuous flags and an underrun interrupt will be generated.

4.1.2 HDLC Receive Mode

When enabled, the receiver enters Flag Hunt Mode. When the first flag is detected, the next non-flag/abort character is treated as the start of frame. If no address recognition is enabled, frame reception then continues; if Address Recognition Mode is enabled, the incoming data is compared with the receive address registers. Two modes of address recognition are available: 1. First byte of address field only, (four possible matches available against RFAR[1:4]); 2. First and second byte address field, (two possible matches available against RFAR[1:2], RFAR[3:4]).

For the purposes of address matching, the address extension bit is not interpreted by the chip. The address matching occurs on either the complete first byte, or complete first and second byte of the frame. If no address match is recognized, Flag Hunt Mode is once again entered, thereby discarding the current frame. If a match is found, normal frame reception continues. When the closing flag of the frame is detected, the data remaining in the FIFO is passed to the CPU, either through DMA transfers or good data interrupts, and then an End Of Frame (EOF) interrupt is generated. The CRC can be either ignored and passed to the CPU, validated and discarded, or validated and passed to the CPU.

The next non-flag/abort character will restart the process; the current state of the receive process is visible to the CPU via the CSR Register, which

indicates whether data, flag or mark are currently being received. To support the Data Phase of an X.21 connection, a Clear Detect feature can be enabled via COR1. When enabled, the receive data and CTS* Pin are monitored for the Clear Indication (0, off) from the remote. If detected, the remainder of the current frame will be discarded, and a clear detect indication will be passed to the CPU via the RISR. However, the channel remains in HDLC Mode until modified by the CPU.

4.2 Bisync Processing

In both transmit and receive, the CL-CD2400 interprets the data to determine the type of frame, and to compile the corresponding BCC. The CPU must program which character set to use, ASCII/EBCDIC, and whether to use a CRC or LRC in Channel Option Register 2 (COR2).

4.2.1 Bisync Transmit Processing

The CL-CD2400 can be programmed to idle in either SYN or mark. When idling in mark, a programmable number of leading pad characters can be transmitted before each data frame. The leading pads ensure the Remote Phase locked loop has sufficient transitions to achieve bit synchronization before data starts. The leading pad character can be programmed as AA (suitable for NRZ and Manchester), or 00 (suitable for NRZI).

When data is available in the FIFO, transmission will be started; any required leading pads will be sent, followed by a SYN pair and the CPU-supplied data. The CL-CD2400 monitors the transmit data to determine frame type and compute the correct BCC, eliminating unnecessary characters from the calculation. If SYN sequences are embedded in the data supplied by the CPU, they will be transmitted, but excluded, from the BCC calculation.

If a frame transmission is aborted via the Special Transmit Command Register, an EOT and trailing pad will be transmitted and the line returned to its idle state. A frame is terminated normally when an End Of Frame (EOF) indication is passed to the CL-CD2400, either in Transmit End Of Interrupt Register (TEOIR), or in the A/B Transmit DMA Buffer Status Register (A/BTBSTS). If the frame ends with an EOT or ENQ condition, the trailing pad is appended and transmission is complete; otherwise, any accumulated BCC is appended, followed by the trailing pad, and the line returned to its idle state.

4.2.2 Bisync Receive Processing

After initialization, the receiver starts in Synchronous Hunt Mode, and will discard data until a pair of SYN characters are detected. The next non-SYN data is assumed to be the start of frame. The receive data is continuously monitored to determine the type of frame (transparent/non-transparent, BCC/no BCC). The BCC, if required, is compiled, excluding any characters which should not be part of the calculation. When a frame terminating condition is detected, and if a BCC has been accumulated, it is checked and the end of frame information passed to the CPU via the Receive Interrupt Status Register. If the frame is terminated with an ENQ condition, the BCC is not checked, and an abort indication is passed to the CPU in RISR.

4.3 Async Processing

Data is transmitted according to the format options defined in the Channel Option Registers. These options determine the character length, parity, and stop bit length. New data sent from the host will be transmitted in a continuous stream, unless one of the following happens:

1. Transmitter disabled – transmission terminated at the end of the current character until transmitter enabled.

2. XOFF received from line – transmission terminated at end of the current character until XON received or transmitter enabled.

3. Out of band flow control – transmission terminated at the end of the current character until out of band flow control removed.

4. In-line command received in data stream from host – in-line command is executed and transmission resumed.

5. Send special character command from host – the current character is completed and the special character is transmitted after which normal transmission is resumed.

4.3.1 In-Band Flow Control

For in-band flow control modes to be active, the Special Character Detect Mode must be enabled.

Transmitter In-Band Flow Control

Transmit in-band flow control is enabled when the Transmit In-Band Enable (TxIBE) Bit in COR2 is set to 1. When TxIBE = 0, in-band flow control is disabled, the Implied XON Mode (IXM) Bit, also in COR2, has no meaning. The XON and XOFF characters are defined in the Special Character Registers SCHR[1:2].

When in-band flow control is enabled.(TxIBE = 1), upon receipt of an XOFF character, the channel will stop transmission after the current character in the Transmit Shift Register and the current character in the Transmit Holding Register are transmitted. When IXM = 0, transmission will restart after an XON character is received. When IXM = 1, transmission will restart after any character is received.

The Flow Control Transparency (FCT) Mode Bit in COR3 is used to determine if the received flow control characters are to be passed to the host. If FCT = 1, the characters are not passed to the host. If FCT = 0, they are passed to the host as exception characters. This bit does not affect non-flow control special characters.

Additional status information about transmitter in-band flow control is available in the Channel Status Register (CSR). The Transmit Flow Off (TxFloff) Bit and the Transmit Flow On (TxFlon) Bit are used.

TxFloff = 0 is normal. TxFloff = 1 means that the channel has been requested by the remote to stop transmission. This bit is reset to 0 when the channel receives restart, as described above. This bit is reset to 0 when the transmitter is enabled or disabled, or the channel is reset.

TxFlon = 0 is normal. TxFlon = 1 means that the channel has been requested by the remote to restart transmission. This bit is reset to 0 once the channel has restarted transmission. This bit is reset to 0 when the transmitter is enabled or disabled, or the channel is reset.

Receiver In-Band Flow Control

The channel can request the remote to stop transmission by sending an XOFF character. Likewise, the channel can request the remote to restart transmission by sending an XON characters. The XON/XOFF characters is transmitted by setting the SndSpc Bit in STCR to a 1.

The CSR contains status bits Receive Flow Off (RxFloff) and Receive Flow On (RxFlon) which are used for receiver in-band flow control.

RxFloff = 0 is normal. RxFloff = 1 means the channel has requested that the remote stop its transmission. This bit is reset to 0 when the channel requests the that the remote restart its transmission. This bit is reset to 0 when the receiver is enabled or disabled, or the channel is reset.

RxFlon = 0 is normal. RxFlon = 1 means that the channel has requested that the remote restart its transmission. This bit is reset to 0 when the next non-flow control character is received. This bit is reset to 0 when the receiver is enabled or disabled, or the channel is reset.

4.3.2 Out-of-Band Flow Control

Receive out-of-band flow control is enabled when the CTS Automatic Enable (CtsAE) Bit is set to 1. In this mode, character transmission will begin only after the CTS* Pin is active (low). In asynchronous transmission, if CTS* goes inactive (high) after transmission has started, the channel will stop transmission after the current character in the Transmit Shift Register, and the current character in the Transmit Holding Register are transmitted. In synchronous modes, if CTS* goes inactive, the channel will stop transmission after the current frame. In either case, transmission will restart after CTS* goes active.

The CL-CD2400 can automatically flow control the remote device via the DTR*Pin. This mode is selected by setting a non-zero DTR* threshold in COR5; when both the threshold in COR4 and the threshold in COR5 are exceeded, the CL-CD2400 will set the DTR* Pin high. When the data in the FIFO falls below the DTR* threshold, the DTR* Pin is automatically driven low.

4.3.3 Line Break Detection and Generation

A line break on the receiver occurs when the input at the receive data (RXD) Pin is all zeros (low) for at least one full character time. This is indicated when the Break Bit in RISR1 is set to 1.

Line break generation out of the transmitter is possible when the Embedded Transmit Command (ETC) Bit in COR2 is set to 1. A line break is generated when the output at the transmit data (TXD) Pin is all zeroes (low) for at least one full character time.

Line breaks may be transmitted by embedding certain sequences in the data stream as defined below. These sequences are valid for transmitting breaks only of ETC = 1. The embedded sequences to transmit a break are:

00h-81h	Send BREAK – Send a line break for at least one character time.
00h-82h-xxh	Insert Delay – To increase the break generation beyond one character time, the Insert Delay sequence may be used. The inserted delay will be xx, where xx is a binary number. The delay will be xx times the 'tick' set by the Timer Period Register (TPR). The minimum period of TPR should be 1 millisecond. If the Insert Delay sequence is not preceded by a Send BREAK sequence, then there will be an inserted delay of all 1's (high) on the output for duration xx.
00h-83h	Stop BREAK – This must follow the Send BREAK sequence, or the Insert Delay sequence.
00h-00h	Send NUL – If the user needs to send a NUL character, and ETC = 1, the user may embed 00h-00h to send one NUL character. If there are less than 8-bits per character, the user may also send a NUL character by 'sending' an 80h.

In addition to Insert Delay, a 'break' may also be increased beyond one full character by transmitting more than one 'Send BREAK' sequence at a time.

4.3.4 Special Characters Special Character Transmission

Selected special characters may be sent preemptively by setting the Send Special Character Command (SndSpc) Bit in the STCR. The CL-CD2400 channel acknowledges the command by clearing the STCR. Along with the SndSpc Bit, the host needs to set-up the three Special Character Select (SSPC0, SSPC1, SSPC2) bits, also in the STCR, to select which character is to be sent.

When the host commands a special character transmission, the channel will complete transmitting any characters in the Transmit Shift Register and Transmit Holding Register, and then transmit the special character sequence. Any other characters awaiting transmission in the FIFO or through DMA will be transmitted after the special character.

If the transmitter is off due to in-band flow control, the special characters will override and be sent. Special characters will override out-of-band flow control. Also, if the transmitter is disabled, the special character send command will override, and the character will be sent.

SSPC2	SSPC1	SSCP0	
0	0	1	Send Special Character #1
0	1	0	Send Special Character #2
0	1	1	Send Special Character #3
1	0	0	Send Special Character #4
0	0	0	reserved
1	0	1	reserved
1	1	0	reserved
1	1	1	reserved

Special Character Recognition and Special Character Range

Special character recognition is enabled when the Special Character Detection (SCDE) Bit in COR3 is set to 1. The special characters are programmed in the SCHR registers, and are the same characters used for the transmitter.

If the Flow Control Transparency (FCT) Bit in COR3 is set to 1, the channel will process the flow control characters and discard them. Otherwise, if FCT = 0, the received flow control characters will be processed and passed onto the host via exception interrupt.

In the event of an error (framing and/or parity) in a received character sequence, the channel will not interpret this character as a special character. But, if an overrun condition occurred after a special character is detected, the new character is lost and the overrun status is set. In this condition, the CL-CD2400 will give both an overrun exception and a special character recognition status.

Special Character Range

The Special Character Range low and high (SCRl and SCRh) registers define an inclusive range for special character recognition in the Asynchronous Mode. It may be useful for identifying that a received character is within a certain range, such as a control character. To disable this function, if special character detection is enabled, make both SCRl and SCRh equal to Special Character #1 (SCHR1).

Special characters and range detection is through the three special character detect (SCdet0, SCdet1, SCdet2) bits in the RISR1 Register. The meanings of these bits are:

SCdet2	SCdet1	SCdet0	
0	0	0	No special characters/range detected
0	0	1	Special character 1 matched
0	1	0	Special character 2 matched
0	1	1	Special character 3 matched if character 1 and 3 sequence not enabled
1	0	0	Special character 4 matched if character 2 and 4 sequence not enabled
1	1	1	Received character is within range given by SCRl and SCRh 5.4

4.3.5 UNIX Support Features

Channel Option Register 6 (COR6) provides several functions required by UNIX TTY drivers, to further reduce the amount of character by character processing that the CPU is required to perform.

Separate receive and transmit bits are provided to perform carriage return(CR)/new line(NL) translations. In transmit NL can be converted to CR NL, or CR converted to NL. In receive, CR can be discarded, NL converted to CR, or CR converted to NL.

In receive processing separate modes are provided to handle break conditions and character error conditions. Break conditions can be handled in the normal way (via a receive status interrupt), or the condition can be discarded, or the break can be translated to a NULL (00) and passed as normal data to the CPU. Parity and framing errors can either be handled as normal (via receive status interrupts),

discarded, translated to a NULL (00) and passed to the CPU as normal data, or the character can be passed to the CPU as normal data, preceded by the sequence FF 00.

The LNext option (COR7—LNE) provides a mechanism to transfer flow control and other special characters without invoking flow control or special character interrupts at the receiver. If the LNext option is enabled when the LNext character is received, the following character is just passed to the CPU as a normal character. The LNext character is programmed via the LNext Register.

The IStrip feature (COR7) strips the eighth bit off each error free received character. It has no effect on the transmitted data.

4.4 X.21 Call Setup Mode

The X.21 call setup protocol uses a combination of synchronous data and control leads to control call progress, between a DCE and a DTE. The data can be in the form of steady state conditions (all ones, all zeros, alternating ones and zeros), or character synchronous data. The control leads are used in conjunction with the steady state data conditions to pass change of state information between the DCE and DTE.

4.4.1 X.21 Transmit

In order to minimize the CPU intervention in the transmit direction, a modified version of the Embedded Transmit Command (ETC) is used. The ETC Mode is controlled via COR2, and when enabled, provides a means of transmitting steady state, or repetitive data patterns synchronized to the control lead. The ETC consists of a sequence of four bytes passed to the CL-CD2400 as normal transmit data, the format of the bytes is as follows.

Byte 1 – 80 this indicates the start of a command sequence

Byte 2 – this indicates the required state of the control lead

00 = set the control lead off
01 = set the control lead on
02-FF reserved

Byte 3 – data character to be transmitted, this is transmitted as is, with no parity added. If parity is required, it must be included in the byte, i.e., to transmit '+' with odd parity, the value AB would be loaded.

Byte 4 – is a count of the number of times to transmit the character. If the count is zero the character will be sent indefinitely until new data is made available (this would be the normal mode for the steady state conditions). When the count is zero the CL-CD2400 will always send the data a minimum of three times even if new data is made available sooner.

4.4.2 X.21 Receive

In receive, the CL-CD2400 validates the steady state conditions passing just the change of state information to the CPU. Steady state conditions validated are the all ones, all zeros, alternating ones and zeros, SYN, and the characters programmed in SCHR1:3. To be validated a condition must be present for two character times with a stable value on the CTS* Pin (CTS* is used in X.21 as the I lead in a DTE or C lead in a DCE).

To enable the detection of the steady state conditions, the SSDE Bit must be set in COR3, this enables the detection of the all ones, all zeros and the alternating ones and zeros conditions with a stable value on the CTS* Pin. When the SSDE Bit is set the StrpSyn and SCDE may also be set if required, (if SSDE is not set then the StrpSyn and SCDE bits have no effect).

The StrpSyn Bit, when set, treats SYN characters in the same way as the steady state conditions, that is when two valid SYN characters are detected a receive special character interrupt is generated and following SYN characters are stripped from the incoming data stream. If the StrpSyn Bit is not set, the SYN character is still used to achieve character synchronization, but all received SYN characters are passed to the CPU, as normal receive data.

The SCDE Bit enables the detection of the special characters defined in SCHR1-3, in the same manner as the steady state conditions. When detected for two consecutive character times, a special character detect interrupt will be generated, and following repetitions of the same

character will be stripped from the receive data. An example of the use of this bit would be to detect the 'BEL off' condition for a DTE incoming call, and then strip its repetition until the next state change. Character synchronization must be achieved before SCHR1-3 can be detected.

4.5 Non-8-Bit Data Transfers

In Asynchronous Mode and in non-HDLC synchronous modes, it is possible to transmit and receive less than 8 bits per character. For asynchronous, there may be 5, 6, 7, or 8 bits per character. For some synchronous (non-HDLC) applications, there may be 7-bit ASCII with parity handled by the chip. In each case where there is less than 8 bits per character, the data transfer will always be from the low-order bit locations.

For HDLC, there are always 8 bits per character transmitted. The CL-CD2400 will transmit only byte-aligned frames. The CL-CD2400 will receive HDLC frames using transfers of 8 bits per character, except for the last character received before the FCS. If this last character is not aligned to an 8-bit boundary, the ResInd (Residual Indication) Bit will be set, along with the EOF Bit in RISR.

5. HOST INTERFACE

5.1 Host Read/Write Cycle

The CL-CD2400 can either be a bus slave for I/O transfers or a bus master for DMA transfers. Sections 6.1 and 6.2 describe the host interface to the CL-CD2400 in Slave Mode, namely, I/O read/write and interrupt. In this document, the term 'Service Request' is used to represent interrupt request by the CL-CD2400. Note that 'Service Request' has a broader meaning than Interrupt:

Interrupt is one way of requesting service from the host CPU. More detailed information on service request is in Section 6.2.

The CL-CD2400 requires a stable input clock (CLK) to be present in order to function; the maximum frequency of this clock is 20 MHz. This clock is divided by 2 internally, and output on the BUSCLK Pin. The divided clock is used to perform internal bus cycles in the CL-CD2400. The maximum serial data rate for each channel is also related to the clock. Reference: Sections 4.1 and 4.2 on BRG and DPLL.

The CL-CD2400 is basically following an asynchronous timing scheme. In other words, all control signals to the CL-CD2400 are not required to have a time relationship with the clock. The AC characteristics show some timing parameters that are related to the clock, to represent minimum access time to the CL-CD2400. The Address Strobe input AS* is not used during slave operations; it is an output during DMA cycles.

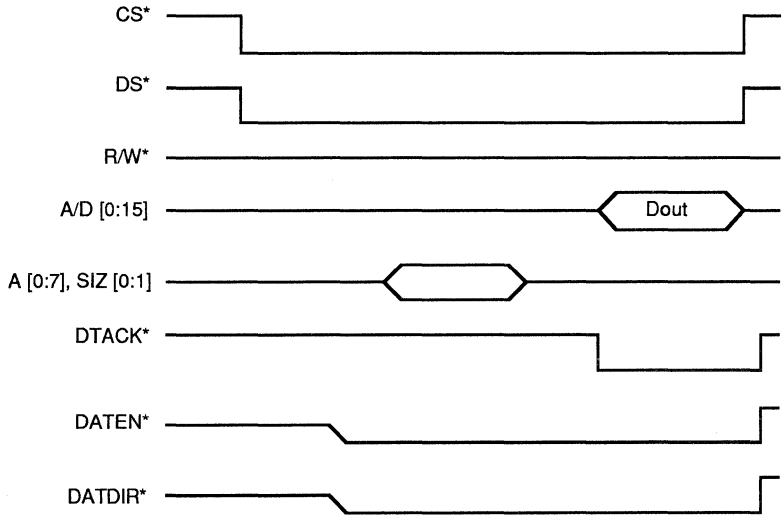


Figure 5–0. Host Read Cycle

Figure 5–0 shows the host read cycle. Note that DS*, R/W*, DATDIR* and DATEN* have to be connected to pull-up resistors. DTACK* is an output which indicates data is valid. When DS* is deactivated, DTACK* and data revert to high impedance. DATEN* and DATDIR* are active during the read cycle, DATDIR* will be driven low, indicating that data direction is from CL-CD2400 to host.

Figure 5–1 shows the host write cycle. Note that DS*, R/W*, DATDIR* and DATEN* have to be connected to pull-up resistors. DTACK* is an output which indicates that the data has been written. When DS* is deactivated, DTACK* reverts to high impedance. DATEN* and DATDIR* are active during the write cycle; DATDIR* is high, indicating that the data direction is from host to CL-CD2400.

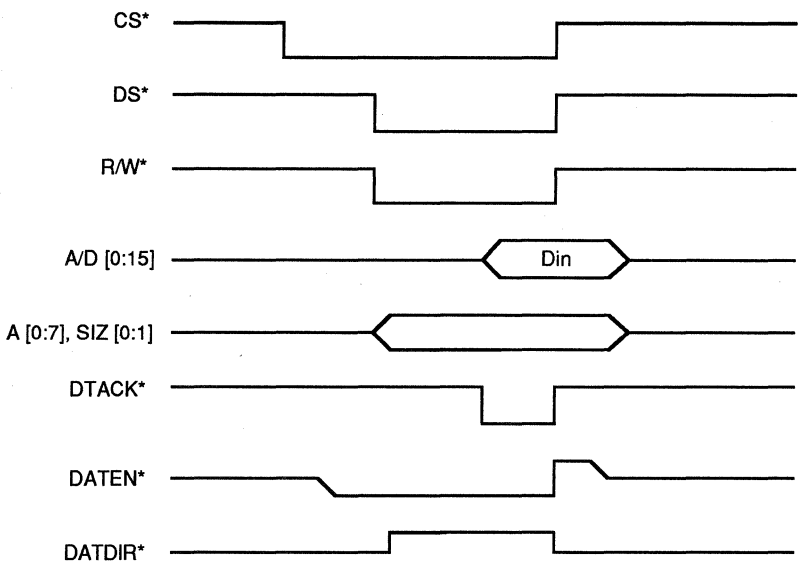


Figure 5–1. Host Write Cycle

5.2 Host Service Request Cycle

The most common way of implementing a service request from the host is by an interrupt. For applications where interrupt service is not available for the CL-CD2400, a simulated interrupt service can be implemented by a processor polling routine. The following section describes how the CL-CD2400 performs the service request mechanism.

The interrupt sources inside the CL-CD2400 are:

Receive/receive exception	from IREQ3* Pin
Transmit	from IREQ2* Pin
Modem	from IREQ1* Pin

For each interrupt source, the causes of the interrupt can be:

Receive:
- Receive good data available (Interrupt Mode)

Receive exception Interrupt:
- Time out, no data (Async)
- Special character/sequence match (Async/X.21)
- Receive overrun (Async, Sync)
- Parity error (Async)
- Framing error (Async)
- Break detection (Async)
- End of frame (Interrupt Mode, Sync)
- Receive abort (Sync)
- CRC error (Sync)
- Residual bit count (Sync)
- Bus error (DMA Mode)
- End of frame (DMA Mode, Sync)
- End of receive buffer (DMA Mode)
- Clear detect (Sync Mode)

Transmit Interrupt:
- Bus error (DMA Mode)
- End of frame (Sync)
- End of transmit buffer (DMA Mode)
- Transmit underrun (Sync)
- Transmitter empty (Async, Sync)
- Transmitter threshold (Interrupt Mode)

Modem Interrupt:
Modem signals change (DSR*, CD* & CTS* transitions in either direction)
General timer1 timeout (see Section 4.2)
General timer2 timeout (see Section 4.2)

Detail description for each interrupt condition is in Section 6, [M,T,R] Interrupt Status Registers.

The following paragraphs describe the registers involved in setting and controlling interrupts.

Interrupt Enable Register, IER (per-channel register)

This register is used to enable/disable the interrupt sources previously described.

Local Interrupt Vector Register, LIVR (per-channel register)

This register has six host-defined bits. The other 2 bits signify the type of interrupt service required, namely Rx exception, Rx data, Tx data, and modem signals change/general timer interrupts. The value of this register is output during the interrupt acknowledge cycle.

Modem Priority Interrupt Register, MPILR (global register)

Transmit Priority Interrupt Register, TPILR (global register)

Receive Priority Interrupt Register, RPILR (global register)

These registers must be initialized by the host to contain the codes that will be present on the address bus during an interrupt acknowledge cycle. The CL-CD2400 will compare the value on the address bus with that in the [M,T,R] PILR, to verify if this cycle is for an interrupt it has asserted. MPILR must contain the code used to acknowledge modem/timer interrupt. TPILR must contain the code used to acknowledge Transmit interrupt. RPILR must contain the code used to acknowledge Receive interrupt.

Modem Interrupt Register, MIR (global register)

Transmit Interrupt Register, TIR (global register)

Receive Interrupt Register, RIR (global register)

These three read only registers provide the host with information on the internal state of the next pending or currently active interrupts in the CL-CD2400.

Stack Register, STK (global register)

This read only register can be used to determine the currently active interrupt type (receive, transmit or modem) and on the interrupt nesting history.

Modem Interrupt Status Register, MISR (global register)

Transmit Interrupt Status Register, TISR (global register)

Receive Interrupt Status Register, RISR (global register)

These three registers provide the status or cause for an interrupt of the corresponding level. In addition, they can be used to pass the value of a timer to the CL-CD2400 at the end of an interrupt (Section 4.2). The advantage is to allow reprogramming the timer when any particular interrupt is being serviced.

Modem Interrupt Register, MIR (global register)
Transmit Interrupt Register, TIR (global register)
Receive Interrupt Register, RIR (global register)

Modem End of Interrupt Register, MEOIR (global register)
Transmit End of Interrupt Register, TEOIR (global register)
Receive End of Interrupt Register, REOIR (global register)

These registers should be written to at the end of the interrupt service routine, so that the CL-CD2400 can execute these commands as the consequence to the interrupt request. This should be the last access to the CL-CD2400 during an interrupt routine.

Recommended Programming Sequence:

Program LIVR	<--------- Chip initialization
Program [M,T,R]PILR	<--------- Chip initialization
Program IER	<--------- Channel initialization
Read [M,T,R]ISR	<--------- Check status during the interrupt service routine
Write [M,T,R]EOIR	<--------- Last instruction to the CL-CD2400 at the interrupt service routine

CL-CD2400 Interrupt Mechanism:

Once the CL-CD2400 is initialized (see previous page), when an interrupt condition exists and the interrupt source is enabled, the CL-CD2400 will assert the IREQn* until a valid interrupt acknowledge cycle is executed. A valid interrupt acknowledge cycle requires IACKIN* and DS* to be asserted and the correct value (corresponding to [M,T,R]PILR contents) on A[0:6]. If the interrupt acknowledge cycle is valid, the corresponding vector will be output and DTACK* asserted.

Figure 5–2 shows the interrupt acknowledge cycle timing. Basically, it is a host read cycle except CS* is inactive (high) and IACKIN* is active.

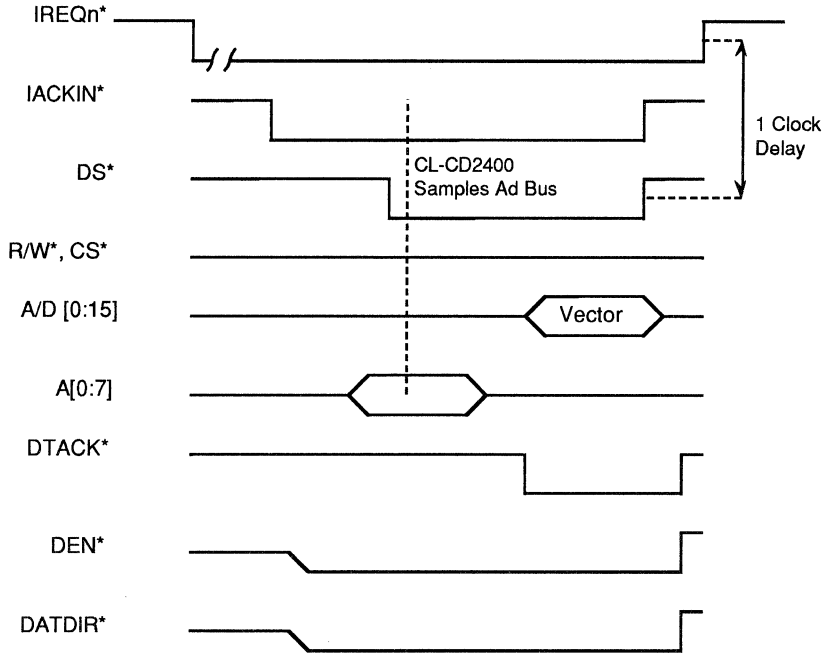


Figure 5–2. Interrupt Acknowledge Cycle

Multiple CL-CD2400s can be daisy-chained together. In this configuration, they share the same interrupt request lines. Interrupt acknowledge is connected to IACKIN* of the first in the chain, and IACKOUT* is then connected to the IACKIN* of the next in the chain. Each individual CL-CD2400 will still be sampling A[0-6] when both IACKIN* and DS* are active. The highest CL-CD2400 in the chain will drive a vector and activate DTACK* when the [M,T,R]PILR matches the A[0:6] bus.

A Fair Share Interrupt mechanism is implemented by hardware in the CL-CD2400 to prevent starving the CL-CD2400s at the end of the chain from interrupt service. The CL-CD2400 monitors the state of the IREQn* Pins, and will not assert a new interrupt until the pin has returned to a high state since it last asserted it. This mechanism can be deactivated by gating each IREQn* Pin with a single direction driver and resistively pulling up each IREQn* Pin. Hence, the CL-CD2400 will always see its IREQn* Pin go inactive after the interrupt acknowledge cycle, even though some other CL-CD2400s is/are still pulling the same interrupt line low.

For polling implementation, the CL-CD2400 has to be programmed as previously described. In this case, the IREQn* signals from the CL-CD2400 will be connected to some discrete logic, so that the host can read from it as an I/O cycle (polling). The said discrete logic can be implemented as a latch, a priority encoder, or an I/O port, depending on the design. Once the host detects one or more service requests, it will simulate an I/O read cycle to the CL-CD2400, which looks like an interrupt acknowledge cycle to the CL-CD2400.

As mentioned before, this simulated interrupt acknowledge cycle should first output A[0-6] to the CL-CD2400, with value equal to the corresponding [M,T,R]PILR value. The subsequent vector being read back by the host will tell the cause of the service request. The host still needs to end the service by writing to the [M,T,R]EOIR.

Figure 5–3 shows the comparison between traditional Poll Mode and CL-CD2400 Poll Mode.

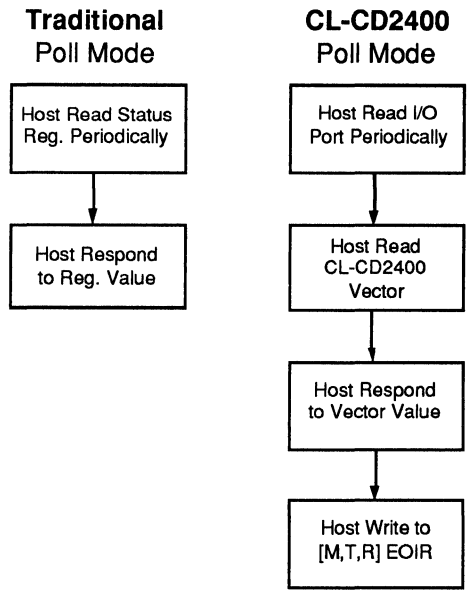


Figure 5–3. Traditional & CL-CD2400 Poll Mode Sequence

NOTE: In either Interrupt Driver or polling environment, the CPU can read/write data to the FIFO only when the interrupt condition exits, i.e., after reading the vector and before writing to the EIOR.

5.3 DMA Cycle

Data Transfer

In Asynchronous Mode, a good data transfer will be initiated for a number of characters in the FIFO greater than the specified threshold. The data transfer can be by DMA or interrupt driven. Receive timeout and receive exception can also initiate a receive transfer. In Synchronous Mode, data transfer will be initiated when the number of characters in the FIFO is greater than the specified threshold, or an end of frame or receive exception is detected.

For transmit operation, the CL-CD2400 will initiate a data transfer when the empty space in the FIFO is greater than the set threshold. Whenever possible, the CL-CD2400 will attempt to perform 16-bit transfers, to reduce the loading on the bus. The CL-CD2400 can be forced to perform 8-bit operations in DMA Mode by setting the Byte DMA Bit in the DMA Mode Register (DMR). For these operations, the upper and lower data buses are used for the transfers as specified in the BYTESWAP pin description. The DMR is writable only if read an inconsistent value will be returned.

DMA Features

There are eight DMA channels on-chip, one for each transmitter and one for each receiver. All necessary signals for DMA handshake are included in this product. Each DMA channel contains the starting address and byte count needed for buffer control.

The user may also optionally choose to have an interrupt generated each time a buffer is completed. In other words, after a buffer has been transmitted, or after a buffer has been received, the CL-CD2400 would generate an appropriate interrupt.

DMA Buffer Management

The CL-CD2400 uses a simple, but powerful, double buffering method. By using a simple method, the CL-CD2400 is readily compatible with higher buffer control procedures, such as circular queues, link lists, buffer pools, or any other. Each transmitter and each receiver is assigned an 'A' buffer and a 'B' buffer. When transmitting, the host processor will alternately fill the A and B buffers, and command the CL-CD2400 to transmit the buffers one at a time. When receiving, the CL-CD2400 will fill the A and B buffers, and inform the host processor when each is ready.

A simple ownership bit is used for each buffer to insure that there are no deadlocks between the host and the CL-CD2400, over who is currently using a particular buffer.

To improve buffer utilization efficiency, a method of buffer chaining is available as a user option. This is useful to break large packets into smaller blocks.

By using the simple and flexible DMA management of the CL-CD2400, the user's host processor is concerned with transmit/receive data on a block-by-block basis. The user does not need to be concerned with character-by-character transfers, or even filling/emptying the FIFOs. The DMA control is user-selectable per channel and operates independent of each other.

5.3.1 Bus Acquisition Cycle

1. CL-CD2400 asserts BR* and waits for BGIN*.

2. When BGIN* is detected, the CL-CD2400 knows that it will have the bus after the current bus owner gives up the bus.

3. If BGACK* is high when BGIN* goes low, then no one else currently owns the bus. In this case, go to step 5.

4. If BGACK* is low when BGIN* goes low, then someone else owns the bus. The CL-CD2400 will wait for BGACK* to go high.

5. Once the CL-CD2400 sees that BGACK* is high, the CL-CD2400 will wait for the current bus cycle to terminate (DS* and DTACK* high) and then assert BGACK* by driving it low. At that time, the CL-CD2400 owns the bus. After driving BGACK* low, the CL-CD2400 will drive BR* high.

Example in which the CL-CD2400 was required to wait for someone else to relinquish the bus:

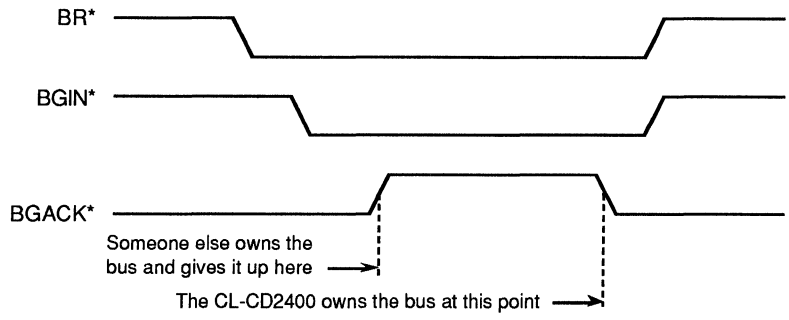


Figure 5–4. Bus Acquisition Cycle

5.3.2 DMA Data Transfer

1. After the CL-CD2400 acquires the bus, it will pulse ADLD* once. This will load the upper 24 address bits to the external 24-bit latch. This will happen only once per DMA grant cycle. The AD[0:15] bits are remapped to memory address (MA) bits MA[16:31], and A[0:7] are mapped to MA[8:15]. If during DMA, the upper 24 bits need to change, the CL-CD2400 will relinquish the bus and then re-acquire the bus.

2. During each DMA read/write cycle, the lower eight memory address bits, MA[0:7] come from A[0:7].

Example of one DMA access after bus is acquired:

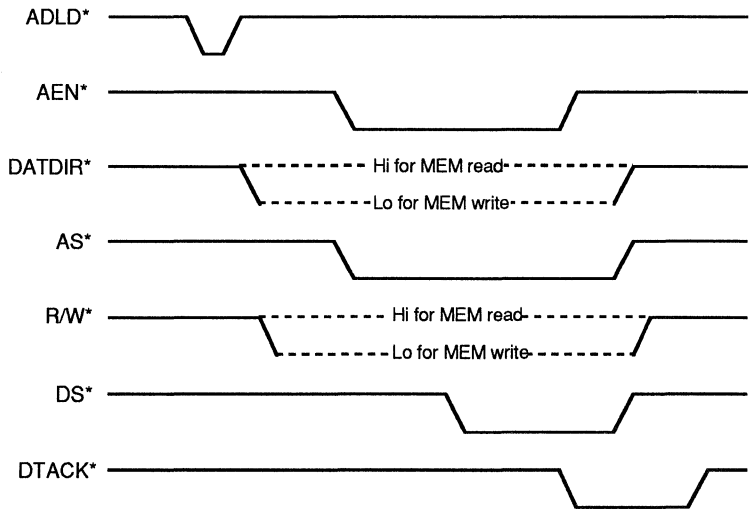


Figure 5–5. Data Transfer Timing

5.3.3 A/B Buffers and Chaining

The buffer management of the CL-CD2400 uses a dual buffer scheme. There is an A/B buffer pair for each transmitter and each receiver. Each buffer is controlled by an ownership status bit, called 2400OWN. When 2400OWN = 1, the CL-CD2400 'owns' the buffer. When 2400OWN = 0, the host 'owns' the buffer. A simple rule prevents confusion in the buffer management: Neither the CL-CD2400 nor the host will seize buffer ownership. Each will always relinquish ownership to the other. The host gives ownership of a receive buffer to the CL-CD2400 when the receive buffer is ready. The CL-CD2400 is then free to write received data into the buffer. The CL-CD2400 gives ownership of the receive buffer back after the receive data is in the buffer. The host gives ownership of a transmit buffer to the CL-CD2400 when the transmit buffer is ready to transmit. The CL-CD2400 will then transmit the contents of the buffer. When this is complete, the CL-CD2400 will give ownership back to the host.

The CL-CD2400 keeps track of which buffer (A or B) is to be used next in the status bits: Ntbuf for transmit and Nrbuf for receive. The relationship between the 2400OWN Bit and the 'next' bits is shown in the example below. The receive buffers are handled in the same way using the Nrbuf (next receive buffer).

Ntbuf	2400 OWN-A	2400 OWN-B	Transmit Action
0	0	0	Send nothing
0	1	0	Host sets up buffer A
1	1	0	CL-CD2400 accepts buffer A and marks B as next
1	0	0	CL-CD2400 completes A tx, and passes it to host
1	0	1	Host sets up buffer B
0	0	1	CL-CD2400 accepts B and marks A as next
0	1	1	Host sets up buffer B
1	1	0	CL-CD2400 completes B tx, passes to host, accepts A and marks B as next
1	0	0	CL-CD2400 completes A tx and passes it to host

Chaining is used to break up relatively long frames into shorter blocks in memory, and is useful where there are frequent smaller frames and occasional long frames. Chaining allows more efficient use of the user's RAM.

The EOF Status Bit is used to control chaining in synchronous modes. Chaining applies to both transmit and receive. For transmit, the host determines EOF Bit. For receive, the CL-CD2400 determines the EOF Bit.

In Transmit DMA, when the first buffer is supplied to the CL-CD2400, it is treated as the start of frame – the CRC is reset and leading pad/flag/syn characters are transmitted, followed by the data. If the EOF Bit is set, the CRC and closing flag/syn will be appended, and the next buffer will again be treated as the start of frame. If the EOF Bit is not set, the CL-CD2400 treats the buffer as the first part of a larger frame and chains into the next buffer (does not reset CRC); this process then continues until a buffer is supplied with the EOF Bit set.

5.3.4 Transmit DMA Transfer

As in receive data transfers, two buffers are available for DMA transmit transfers. The Transmit Buffer Address and Transmit Buffer Count Registers (ATBADR/BTBADR and ATBCNT/BTBCNT) contain the start address of and the byte count in the buffers. These registers are set by the host, when initiating a transfer. The CL-CD2400 makes a copy of the registers to perform the transfer, leaving the originals unchanged. Transfer of buffers between the host and the CL-CD2400 is controlled by the Transmit Buffer Status (ATBSTS/BTBSTS) Registers.

Buffers can contain either complete frames/blocks of data, linked together to form a complete frame/block, or used in an Append Mode to transmit data as it arrives from another process. The first two transfer types are Block Mode transfers, the last is the Append Mode, and both are described below. The management of the buffers reduces the processor overhead associated with short data transfers and increases the minimum response time requirements for frame-based transmissions.

Chain Mode Transfer

In this mode, the frame should be complete in buffers in memory before transmission is

started. The Append Status Bit should not be set, the Start of Frame Bit must be set to begin transmission, and the last buffer bit must be set if this buffer is the last in a chained block or is a complete frame/block.

When the CRC Bit is set, the CL-CD2400 will generate and transmit a Cyclic Redundancy Check word for the frame using the polynomial selected by the CRC Polynomial Select Register (CPSR). A host interrupt will be generated after the buffer is transmitted, if the interrupt required bit is set.

Transmit buffers must be of an even byte size if they are to be chained to support large frames. They must also begin on an even address in host memory. The CL-CD2400 will begin fetching a frame from a buffer performing DMA transfers reading two characters at a time. If buffers do not begin and end on even memory addresses, alignment between the CL-CD2400 internal FIFO and the external buffers cannot be maintained.

Append Mode Transfer

This is available for Buffer A in Asynchronous Mode only. If Buffer A is set to Append Mode, the host can enable the CL-CD2400 to transmit data in the buffer before it is completely filled. The CL-CD2400 will start transmitting new data when it is appended to the buffer.

This mode is useful for terminal echo routines, which do not wait for a complete block to be formed before starting transmission. In this mode,transmission is started when the buffer is made available to the CL-CD2400 by the host: the ATBADR[0-3] and the ATBCNT[L,H] are initialized. Subsequent triggering of DMA transfer takes place by programming the ATBCNT[L,H] with the accumulated byte count. The ATBCNT should be written as a 16-bit word in this case, to avoid confusion between two byte operations. The ATBADR[0-3] should not be reprogrammed during the Append Mode. If the memory space has to be moved, the Append Mode has to be disabled first. When the final data is added to the append buffer and ATBCNT has been updated, the host should set the AppdCmp Bit in STCR. When the CL-CD2400 has completed the final transmission, it will clear the 2400OWN Bit in the ATBSTS Register, and generate an end of buffer interrupt. Only the 'A' buffer can be used in the Append Mode.

Transmitter A/B Buffers

The DMABSTS (8) Register applies to each channel. The buffer specific registers shown below apply to each channel.

In this drawing, Buffers A and B are contained in RAM external to the CL-CD2400. All else (DMABSTS, ATBADR, TCBADR, ATBCNT, ATBSTS, BTBADR, BTBCNT and BTBSTS) are inside the CL-CD2400.

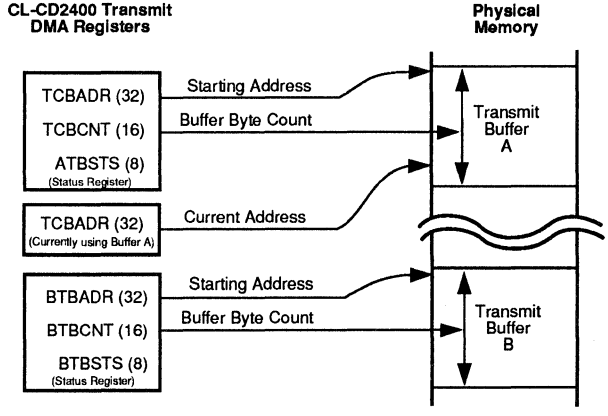


Figure 5–6. Transmitter A/B Buffers

NOTE: Number of bits in each register is shown in parentheses ( ). Buffer A and Buffer B do not necessarily need to be the same length.

Synchronous Transmission Example #1:

Transmit a frame out of channel 1, with no chaining.

1. The host checks the Ntbuf Bit in the DMABSTS Register for channel 1 to determine which buffer is next. In this example, Ntbuf = 0 indicating that Buffer A will be used next.

2. The host sets up the buffer data, the starting address, ATBADR, and the buffer byte count, ATBCNT.

3. The host then sets up the ATBSTS (A buffer status) Register. The EOF Bit is set to indicate that there is no chaining. The 2400OWN Bit is set to give ownership to the CL-CD2400. By setting 2400OWN, the host commands the CL-CD2400 to start transmission. Thus, everything must be ready (starting address, buffer data, byte count) prior to setting 2400OWN.

4. The CL-CD2400 will start frame transmission out of channel 1. When transmission is started, the CL-CD2400 will set bit Tbusy in DMABSTS. As transmission progresses, the current buffer pointer, TCBADR, is updated by the CL-CD2400. Also, at the start of transmission, the Next Buffer Bit, Ntbuf, is set to 1 to tell the host that Buffer B will be next.

5. The CL-CD2400 will complete frame transmission by adding any necessary CRCs and trailing frame delimiters.

6. When the CL-CD2400 completes the transmission, the CL-CD2400 will clear the Tbusy Bit. Then, the CL-CD2400 will set the EOB Bit and clear the 2400OWN Bit in the ATBSTS. This will tell the host that the transmission is complete, and give ownership of the buffer back to the host.

7. The CL-CD2400 will then optionally interrupt the host, with EOF and EOB in the TISR both set to indicate that the transmission is complete, and there was no chaining.

Synchronous Transmission Example #2:

Transmit out of channel 0, and chain three buffers into one frame. The frame is 240 bytes long, and the maximum buffer size is 100.

1. The host checks the Ntbuf Bit in the DMABSTS Register for channel 0 to determine which buffer is next. In this example, Ntbuf = 1 indicates Buffer B will be used next.

2. The host sets up the buffer data, the starting address, BTBADR, and the buffer byte count, BTBCNT, for the first 'link' of the chain to be transmitted. For this example, BTBCNT = 100.

3. The host then sets-up the BTBSTS (B buffer status) Register. The EOF Bit is clear to indicate that this buffer is the first link in a chain. The 2400OWN Bit is set to give ownership to the CL-CD2400. By setting 2400OWN, the host commands the CL-CD2400 to start transmission. Thus, everything must be ready (starting address, buffer, data count) prior to setting 2400OWN.

4. At this point, the host knows that it has the time it takes to transmit 100 bytes to set up the next buffer link. If the host fails to do this in time, there will be a transmitter underrun, and the frame is aborted in HDLC or bisynchronous.

5. The CL-CD2400 starts transmitting Buffer B from channel 0. When this is started, the Ntbuf Bit is cleared to 0 to show that Buffer A will be next. This helps the host keep track of which buffer is next. As transmission progresses, the current buffer pointer, TCBADR is updated by the CL-CD2400. During this, or prior, the host will have made Buffer A ready. For Buffer A, the EOF Bit in the ATBSTS Register is cleared by the host, indicating that the buffer is not the end of the chain.

6. At the end of transmission of this buffer, the CL-CD2400 does not add any CRCs nor end of frame delimiters because there is more data for the current frame.

7. After the CL-CD2400 has completed transmission of the first link out of Buffer B, the CL-CD2400 will set the EOB Bit and clear the 2400OWN Bit in the BTBSTS. This will tell the host that the transmission is complete, and give ownership of the buffer back to the host.

8. The CL-CD2400 will then optionally interrupt the host with EOF clear and EOB set in the TISR to indicate that the transmission is completed, and that there was chaining.

9. The CL-CD2400 now sees from the ATBSTS Register that it has ownership of Buffer A for transmission of the next 'link'. It also sees that the EOF is clear so that this link is not the last link in the transmitted chain.

10. The CL-CD2400 will continue transmission of the current frame, but now transmission is from Buffer A. This will be the second link which will be 100-bytes-long. During this time, the host must set up a new Buffer B for the third, and final, link. The BTBCNT for the last link will be set to 40 bytes.

11. After the CL-CD2400 has completed transmission of the second link out of Buffer A, the CL-CD2400 will set the EOB Bit and clear the 2400OWN Bit in the ATBSTS. This will tell the host that the transmission is completed, and give ownership of the buffer back to the host. As with the first link, the CL-CD2400 will not add CRCs nor ending frame delimiters to this link.

12. The CL-CD2400 will then optionally interrupt the host with EOF clear and EOB set in the TISR to indicate that the transmission is completed, and there was chaining.

13. By this time, the host has set up a new buffer for Buffer B. The EOF Bit in the BTBSTS is set to indicate that this is the last link in the chain.

14. The CL-CD2400 then transmits Buffer B in the same manner shown above. As before, the CL-CD2400 transmits the number of bytes indicated in the BTBCNT, which is 40 for the third segment.

15. When the CL-CD2400 completes transmission, any necessary CRCs and ending frame delimiters are transmitted.

16. The CL-CD2400 will then optionally interrupt the host with EOF and EOB set in the TISR to indicate that the transmission is complete, and this was the last link in the chain.

5.3.5 Receive DMA Transfer

In all protocol modes, two host memory buffers can be made available to each receive channel, via the Receive Buffer Address and Receive Buffer Count Registers (ARBADR/BRBADR and ARBCNT/BRBCNT). To make a buffer available, the user must supply the buffer address in the Receive Buffer Address Registers; the number of free bytes in the buffer must be written in the Receive Buffer Count Registers, and the buffer status must be updated in Receive Buffer Status Register (ARBSTS/BRBSTS). The CL-CD2400 is then free to use the buffer for receive data, and will update the Buffer Status Register as appropriate. When the buffer is no longer in use, the CL-CD2400 will write the number of bytes stored in the buffer in RBCNT and update status in RBSTS. This frees the host to take control of this buffer and supply a new buffer in its place. The CL-CD2400 automatically switches to the other buffer whenever one buffer becomes full or the end of a frame has been reached. If the other buffer has not been allocated, the host still has the time required to fill the CL-CD2400 16-byte FIFO in which to respond and avoid loss of data.

Receive buffers must begin on an even address in host memory. They must also be of an even size in bytes, if they are to be chained to support large frames. The CL-CD2400 will begin storing a frame in an empty buffer performing DMA transfers writing two characters at a time. If buffers do not begin and end on even memory addresses, alignment between the CL-CD2400 internal FIFO and the external buffers cannot be maintained.

Special actions are taken depending on the channel protocol. In HDLC and bisync, the end of frame/data block boundaries are recognized by the CL-CD2400. When a data block boundary is detected, the current buffer is automatically terminated. If the other buffer is

allocated and described to the CL-CD2400, it will become the current buffer. End of frame/block interrupts will also be generated to the host.

In Asynchronous Mode, a host interrupt is generated when there are receive exceptions (framing error, special character, etc.), but the buffer is not terminated. The data and exception status are made available to the host, just as when the Asynchronous Mode is purely interrupt driven. New data will be buffered internally in the FIFO, until the host services the exception interrupt. The host has three options when terminating an exception interrupt:

1. The exception character can be discarded.
2. The buffer can be terminated; if it is, no additional interrupt would be generated. The transfer count is not provided in A/BRBCNT, but can be calculated via Receive Current Buffer Address (RCBADR).
3. A user-defined gap can be left in the buffer.

These selections are communicated to the CL-CD2400 by the value written by the host to the Receive End Of Interrupt Register, when the Receive Interrupt service is completed.

Leaving an 'n' byte gap enables the host to insert status of its own in the current buffer, while continuing to receive data in the same buffer. This eliminates the overhead of allocating a new buffer. The host must have noted the starting location of the gap while in the exception interrupt. This is done by reading the Receive Current Buffer Address Register. The address in this register is guaranteed to be stable during the Receive Interrupt, and to point to the next free character location in the current DMA buffer.

If the size of the gap supplied by the host is sufficient to fill or complete the current buffer, the CL-CD2400 will automatically switch to the other buffer and advance the Receive Current Buffer Address enough to complete the desired gap.

The CL-CD2400 will readjust data alignment in its internal FIFO as needed to maintain alignment with the external buffer.

Receiver A/B Buffers

The DMABSTS (8) Register applies to each channel. The buffer specific registers shown below apply to each channel.

In the following drawing, Buffers A and B are contained in RAM external to the CL-CD2400. All else (DMABSTS, ARBADR, RCBADR, ARBCNT, ARBSTS, BRBADR, BRBCNT and BRBSTS) are inside the CL-CD2400.

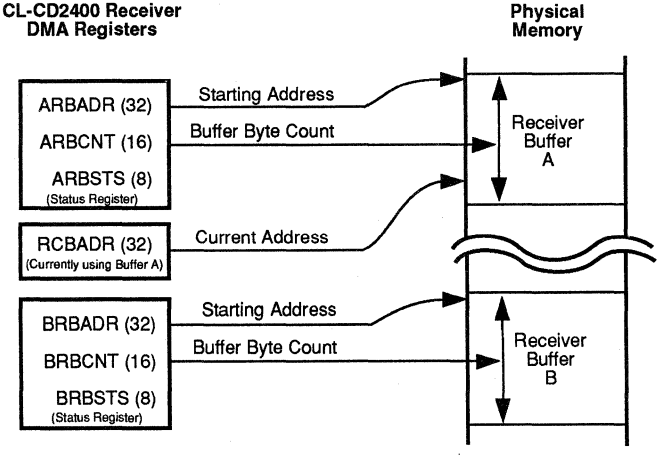


Figure 5–7. Receiver A/B Buffers

NOTE: Number of bits in each register is shown in parentheses (). Buffer A and Buffer B do not necessarily need to be the same length.

Synchronous Receiver Example #1:

Receive a frame from channel 1; no chaining.

1. The host must first make a receive buffer available before a frame can be received. Thus, the host checks the Nrbuf Bit in the DMABSTS Register for channel 1 to determine which buffer is next. In this example, Nrbuf = 0 indicates Buffer A will be used next.

2. The host sets up the starting address, ARBADR, and the buffer byte count, ARBCNT. When the host writes the count, ARBCNT, the host has defined the size limit for the buffer.

3. The host then gives the buffer to the CL-CD2400 by setting the 2400OWN Bit in the status register, ARBSTS. This tells the CL-CD2400 that it is now OK to write received data from channel 1 into that buffer.

4. The Rbusy Bit in the DMABSTS Register for channel 1 is 0 until a frame starts to be received. When frame data starts coming in, the CL-CD2400 will set Rbusy to tell the host that Buffer B is next. As data bytes are written into the buffer, the current buffer pointer, RCBADR, is updated by the CL-CD2400.

5. At the end of the received frame, the CL-CD2400 tests for correct end of frame delimiter and CRC. When the received frame is complete, the CL-CD2400 clears the Rbusy Bit. In this example, there is no receive chaining, so the received frame byte count is less than or equal to the buffer size count, ARBCNT. The CL-CD2400 will write the value of the actual received byte count into the same register, ARBCNT. [Recognize the host has written the maximum buffer size in ARBCNT when the buffer is given to the CL-CD2400. But when the buffer is given back to the host, the CL-CD2400 has written the actual byte count of the received buffer into ARBCNT.]

6. Then, the CL-CD2400 will set the EOB and EOF bits. This tells the host that the end of the buffer and frame have been reached. The CL-CD2400 will also clear the 2400OWN Bit to give the buffer back to the host.

Synchronous Receiver Example #2:

Receive a frame on channel 0, which consists of three buffers chained together. The frame is 240 bytes long, and the maximum buffer size is 100.

1. The host checks the Nrbuf Bit in the DMABSTS Register for channel 0 to determine which buffer is next. In this example, Nrbuf = 1 indicates Buffer B will be used next.

2. The host sets up the starting address, BRBADR. Buffer size is set to 100 in this example. Thus, the host sets BRBCNT = 100.

3. The host then sets the 2400OWN Bit to give ownership to the CL-CD2400.

4. The host should know the amount of time it takes to receive 100 bytes, because this is the minimum time the host has to set up the next buffer link. If the host fails to do this in time, there will be a receiver overrun, and the received frame will be lost.

5. Suppose that the CL-CD2400 starts receiving data into Buffer B of channel 0. When this is started, the Nrbuf Bit is cleared to 0 by the CL-CD2400 to help the host keep track of which buffer is next. (During this time, or prior, the host will have made Buffer A ready.)

6. After the CL-CD2400 has received the first link of the frame into Buffer B, the CL-CD2400 will set the EOB and SOB bits and clear the EOF Bit. This will indicate that the first link in a chain has been received. Also, the CL-CD2400 will clear the 2400OWN Bit, and give ownership of the buffer back to the host.

   For the first received link, the received byte count, BRBCNT, remains unchanged at 100, since the received data filled the buffer.

7. The CL-CD2400 will then optionally interrupt the host with EOF clear and EOB set in the RISR to indicate that the received buffer is complete, and that there was chaining.

8. The CL-CD2400 now sees from the ARBSTS Register that it has ownership of Buffer A for transmission of the next 'link.'

9. As the frame continues to be received, the data will go into Buffer A. This will be the second link, which will be 100 bytes long. During this time, the host must set up a new Buffer B for the third, and final, link.

10. After the CL-CD2400 has received the second link into Buffer A, the CL-CD2400 will set EOB Bit and clear the 2400OWN Bit in the ARBSTS. This will give ownership of the buffer back to the host.

    As with the first link, the received byte count, ARBCNT, remains unchanged at 100 since the received data filled the buffer.

11. The CL-CD2400 will then optionally interrupt the host with EOF clear and EOB set in the RISR to indicate that the received buffer is complete, and there was chaining.

12. By this time, the host has set up a new buffer for Buffer B.

13. The CL-CD2400 will then receive data into Buffer B in the same manner shown above.

14. In this example, the third link does not fill the buffer. Thus, when the end of frame delimiter is detected by the CL-CD2400, the value of 40 (for 40 received bytes) will be written into the received byte count, BRBCNT.

15. Next, the CL-CD2400 will set the EOB and EOF bits to show that the buffer is complete, and that this is the last link in the chain.

16. The CL-CD2400 will then optionally interrupt the host with EOF and EOB set in the RISR to indicate that the received frame is complete, and this was the last link in the chain.

5.4 Byte, Word Transfers

5.4.1 Byte Transfers

In either the DMA or Non-DMA Mode, data may be transferred one byte at a time. If the BYTESWAP Pin is high, byte transfers are performed on A/D[0:7] if the address is even and A/D[8:15] if the address is odd. The reverse is true if BYTESWAP is low.

5.4.2 Word Transfers

During DMA data transfers, buffers should always start on an even byte boundary, and if chained, contain an even number of bytes. The DMA cycles will, therefore, mainly consist of 16-bit bus cycles. At the end of a synchronous frame or during asynchronous transfers, byte bus cycles may occur. When the CL-CD2400 ends a transfer with a byte bus cycle, it will always be on the even address. If it is to continue from the same buffer next transfer, it will start the next sequence of DMA bus cycles with a byte transfer to the odd address, and then continue with 16-bit transfers. This maintains the alignment of data between internal and external memory.

For example, suppose the chip is set up for word transfers in DMA. Suppose further that the host gives the chip five bytes of data to transmit. The chip would do two-word transfers and one-byte transfers for those five bytes.

6. DETAILED REGISTER DESCRIPTIONS

6.1 Global Registers

Global Firmware Revision Code Register		INT	MOT	SIZ	ACCESS
	{GFRCR}	82	81	B	R

7	6	5	4	3	2	1	0
Firmware Revision code							

This register serves two functions in providing the host with information about the CL-CD2400. When the CL-CD2400 is initialized by a hardware RESET signal, or by a software Reset All command issued through any channel Channel Command Register, the CL-CD2400 zeros his register at the start of the initialization. At the conclusion of the initialization, the CL-CD2400 writes its firmware revision code to the GFRCR. All valid CL-CD2400 revision codes are non-zero, the revision code is incremented by one with each new release, e.g., GFRCR for Revision G = 07 hex.

Host software must confirm that the GFRCR contents are non-zero before proceeding to configure the CL-CD2400 for normal operation.

Channel Access Register			EC	EE	B	R/W
	{CAR}					

7	6	5	4	3	2	1	0
0	0	0	0	0	0	C1	C0

This register contains the channel number used for the channel-oriented host read or write operations, when the host is not in an interrupt service routine. The CL-CD2400 supplies the interrupting channel number during all interrupt service operations. The Channel Access Register contents are not used during interrupt service. Note that this means that an interrupt service routine is restricted to accessing only the register set of the interrupting channel and global registers.

Bits 7-2 Reserved

Bits 1-0 Channel Number

C1	C0	Channel Number
0	0	Channel 0
0	1	Channel 1
1	0	Channel 2
1	1	Channel 3

6.2 Option Registers

Channel Mode Register {CMR} | INT 18 | MOT 1B | SIZ B | ACCESS R/W

7	6	5	4	3	2	1	0
RxMode	TxMode	0	0	0	chmd2	chmd1	chmd0

Bit 7 Receive Transfer Mode
0 – Interrupt
1 – DMA

Bit 6 Transmit Transfer Mode
0 – Interrupt
1 – DMA

Bits 5-3 Reserved

Bits 2-0 Protocol Mode select

An initialize command must be given to the CL-CD2400 through the Channel Command Register, if these options are changed.

chmd2	chmd1	chmd0	
0	0	0	HDLC
0	0	1	bisync
0	1	0	async
0	1	1	X.21
1	0	0	reserved
1	0	1	reserved
1	1	0	reserved
1	1	1	reserved

Channel Option Register 1 {COR1} 13 10 B R/W
HDLC Mode

An Initialize command must be given to CL-CD2400 through the Channel Command Register if any options specified in this register are changed.

7	6	5	4	3	2	1	0
AFLO	ClrDet	AdMde1	AdMde0	Flags3	Flags2	Flags1	Flags0

Bit 7 Address Field Length Option
0 = Address field is 1 octet in length
1 = Address field is 2 octets in length

Bit 6 Clear Detect for X.21 Data Transfer Phase
0 = Clear detect disabled
1 = Clear detect enabled
A clear is defined as two consecutive all zero receive characters with the CTS* Pin high.

Bits 5-4 Addressing Modes
00 = no address recognition
01 = 4 * 1 byte
10 = 2 * 2 byte. (If this bit is set, RFAR1, RFAR2, RFAR3, RFAR4 should contain the address to be matched. If AFLO is set to 1, an address match will be made against the RFAR1 and RFAR2 pair or the RFAR3 and RFAR4 pair.

Bits 3-0 Inter-frame flag option
Defines the minimum number of flags transmitted before a frame is started.

Flags 3	Flags 2	Flags 1	Flags 0	
0	0	0	0	— minimum of 1 opening flag, with shared closing/opening flags permitted
0	0	0	1	
	through			— minimum number of opening flags sent
1	1	1	1	

The minimum number of opening flags will always precede a frame when idle in mark mode is set or will always separate two consecutively transmitted frames. No restriction is placed on the number of flags between received frames.

Channel Option Register 1 *(cont.)*
Asynchronous, Bisynchronous and X.21 Modes

7	6	5	4	3	2	1	0
Parity	ParM1	ParM0	Ignore	Chl3	Chl2	Chl1	Chl0

Bit 7 Parity – '1' = odd parity
'0' = even parity

Bits 6-5 Parity Mode 1 & 0 – Defines Parity Mode for both transmitter and receiver:

ParM1	ParM0	Parity
0	0	no parity
0	1	force parity (odd parity = force 1, even = force 0)
1	0	normal parity
1	1	reserved

Bit 4 Ignore – Ignore Parity
0 = evaluate parity on received characters
1 = do not evaluate parity on received characters

Bits 3-0 Character Length

Chl3	Chl2	Chl1	Chl0	Character Length
0	1	0	0	5 bits
0	1	0	1	6 bits
0	1	1	0	7 bits
0	1	1	1	8 bits

Channel Option Register 2 {COR2} 14 17 B R/W
HDLC Mode

7	6	5	4	3	2	1	0
0	FCSApd	0	CRCNinv	0	RtsAO	CtsAE	DsrAE

Bit 6 FCS Append
0 = Receive CRC is not passed to the host at end of frame
1 = Receive CRC is passed to the host at end of frame

Bit 4 CRCNinv
0 = CRC is transmitted inverted (i.e., CRC V.41)
1 = CRC is not transmitted inverted (i.e., CRC-16)

Bit 2 RTS Automatic Output Enable. When set, if the channel is enabled, the CL-CD2400 will automatically assert the RTS* output when it has characters to send. When Idle-in Mark Mode is selected, RTS* will be asserted prior to opening flags and will remain asserted until after a closing flag has been transmitted.

Bit 1 CTS Automatic Enable. Enables CTS* input to be used as automatic transmitter enable/disable. If enabled, the CTS input is checked before frame transmission is started.

Bit 0 DSR Automatic Enable. Enable the DSR* input as automatic receiver enable/disable. If enabled, the pin is checked at the beginning of each received frame.

Channel Option Register 2 {COR2} 14 17 B R/W
Asynchronous Mode

7	6	5	4	3	2	1	0
IXM	TxIBE	ETC	0	RLM	RtsAO	CtsAE	DsrAE

Bit 7 Implied XON Mode (Async)
This bit only has meaning when TXIBE is enabled. During Transmit In-Band Flow Control Mode, the CL-CD2400 stops transmission upon detection of an XOFF character. The IXM Bit determines whether the CL-CD2400 should restart transmission based on receipt of an XON character or any character.

IXM = 0 Following receipt of an XOFF character, transmission will only be resumed by receipt of an XON character or a transmit enable command via CCR.

IXM = 1 Following receipt of an XOFF character, transmission will be resumed by the receipt of any character or a transmit enable command via CCR. If the character that restarts transmission is not an XON character, it is not subjected to the Flow Control Transparency feature, i.e., it will be passed to the host CPU. The XOFF character will restart transmission if the channel is in a 'flow-off' conditions.

Bit 6 Transmit In-Band (XON/XOFF) Flow Control Enable
The CL-CD2400, upon receipt of the XOFF character, terminates transmission after the current character in the Transmit Shift Register, and the character in transmit holding register, are sent. The CL-CD2400 will resume transmission upon receipt of the XON character, any character (depending on the state of the IXM Bit) or a transmit enable command via CCR.

The XON and XOFF character must be programmed in SCHR1 and SCHR2 respectively. When they are programmed to the same value, the flow control state is toggled on each reception.

Channel Option Register 2
Asynchronous Mode *(cont.)*

Bit 5 Embedded Transmitter Command Enable (Async)
If set, the embedded special transmitter command functions are enabled. (See Section 4.3.3)

The null (all zero) character is used as the ESCape character. The following functions are supported:

00H 00H – Send one 00H character as normal data

00H 81H – Send Break – Enter line Break condition for at least 1 character time. (If the insert delay special character sequence immediately follows the Send Break sequence, the duration of the break transmission is extended by the amount of the programmed delay.)

00H 82H XXH – Insert Delay – Insert a delay of 'XX' (interpreted as an unsigned binary number) times the programmed timer 'tick' set by the Prescaler Period Registers. (A zero delay count results in no delay.)

00H 83H – Stop Break – Exit line Break condition and resume normal character transmission.

Bit 4 Reserved.

Bit 3 Remote Loopback Mode
'1' enables the Remote Loopback Mode
'0' disables the Remote Loopback Mode

Bit 2 RTS Automatic Output Enable. When set, if the channel is enabled, the CL-CD2400 will automatically assert the RTS* output when it has characters to send. RTS* will not be deasserted until after a stop bit has been transmitted.

Bit 1 CTS Automatic Enable. Enables CTS* input to be used as automatic transmitter enable or disable. In Asynchronous Mode, the option is evaluated prior to each character transmission. In Sync modes, the option is checked prior to each frame transmission.

Bit 0 DSR Automatic Enable. Enable the DSR* input as automatic receiver enable/disable. In Asynchronous Mode, the option is evaluated at the end of each character received. In Sync modes, the option is checked at the beginning of each received frame.

Channel Option Register 2 *(cont.)*
Bisynchronous Mode

7	6	5	4	3	2	1	0
LRC	BCC	EBCDIC	CRCNinv	0	0	0	0

Bit 7 Longitudinal Redundancy Check
0 = CRC16 used for BCC
1 = LRC used for BCC

Bit 6 BCC Append
0 = Receive BCC is not passed to the host at end of frame
1 = Receive BCC is passed to the host at end of frame

Bit 5 EBCDIC
0 = ASCII used as character set
1 = EBCDIC used as character set

Bit 4 CRCNinv
0 = CRC is transmitted inverted (i.e., CRC V.41)
1 = CRC is not transmitted inverted (i.e., CRC-16)

X.21 Mode

7	6	5	4	3	2	1	0
0	0	ETC	0	0	0	0	0

In X.21 Mode, the CL-CD2400 uses the RTS* Pin as the C lead. C ON is defined as RTS* low and C OFF as RTS* high.

Bit 5 Embedded Transmitter Command Enable. If set, the embedded special transmitter command functions are enabled.

In X.21 Mode, this feature is provided to simplify the transmission of both repetitive data, and data synchronized to the C lead. The command is a sequence of four consecutive bytes supplied as normal transmit data by the host processor.

Byte 1 This must be equal to 80 to start a command sequence.

Byte 2 This indicates the required state of the C lead to be synchronized with the transmit data.
00 = set the C lead to OFF
01 = set the C lead to ON
02-FF reserved

Byte 3 This is the required data character for transmission. It will be sent as an 8-bit character without parity (any required parity should be included in the character by the host).

Byte 4 This is a count of the number of times the character should be sent, if set to zero the character will be sent until more data is provided to the transmitter (but always a minimum of three times).

Channel Option Register 3 {COR3} 15 16 B R/W

In Synchronous Mode, COR3 is used to specify the learning pattern (pad character) sent by the CL-CD2400 to synchronize the DPLL at the remote end. The pad character sent depends on the kind of encoding used.

HDLC Mode

7	6	5	4	3	2	1	0
sndpad	Alt1	FCSPre	FCS	Idle	pad2	pad1	pad0

In Synchronous Mode, COR3 is used to specify the learning pattern(pad character) sent by the CL-CD2400 to synchronize the DPLL at the remote end. The pad character (00h or AAh) sent depends on the kind of encoding used.

Bit 7 1 = CL-CD2400 will send pad character(s) before sending flag when coming out of the Mark Idling Mode.
0 = CL-CD2400 will not send any pad character.

Bit 6 alt1 – send sync pattern.
1 = AAh (Manchester/NRZ Encoding) is sent as pad character.
0 = 00h (NRZI Encoding) is sent as pad character.

Bit 5 FCS Preset
0 = FCS is preset to all ones.
1 = FCS is preset to all zeros.

Bit 4 FCS Mode.
1 = Disable FCS generation and checking, CL-CD2400 treats the entire frame as data.
0 = Normal FCS Mode. The CL-CD2400 generates and appends CRC on transmit and validates CRC on receive using CRC polynomial selected through CRC Polynomial Select Register.

Bit 3 Idle Mode
0 = idle in flag
1 = idle in mark

Bits 2-0 Character Count – Specifies the number of synchronous characters sent.

pad2	pad1	pad0	
0	0	0	reserved
0	0	1	1 pad character sent
0	1	0	2 pad characters sent
0	1	1	3 pad characters sent
1	0	0	4 pad characters sent

101 - 111 are reserved

Channel Option Register 3 *(cont.)*
Asynchronous Mode

7	6	5	4	3	2	1	0
ESCDE	RngDE	FCT	SCDE	0	Stop2	Stop1	Stop0

Bit 7 ESCDE = Extended Special Character Detect Enable
0 = Special Character detect for SCHR 3 and 4 is disabled.
1 = Special Character detect for SCHR 3 and 4 is enabled; a special character interrupt will be generated following the receipt of a character matching SCHR3 or SCHR4.

Bit 6 RngDE = Range Detect Enable
0 = Range Detect disabled.
1 = Characters between SCRI and SCRh (inclusive) generate special character interrupts.

Bit 5 FCT – Flow Control Transparency Mode
0 = Flow control characters received will be passed to the host via receive exception interrupts.
1 = Flow control characters received will not be passed to the host.

This bit has no effect unless both TxIBE (COR2) and SCDE (COR3) are set.

Bit 4 Special Character Detection
0 = Special character detect for SCHR1 and 2 is disabled.
1 = Special character detect for SCHR1 and 2 is enabled.

This bit must be set along with TxIBE (COR2) before FCT (COR3) becomes effective.

Bit 3 Reserved. Must be zero.

Bits 2-0 Stop Bit Length – Specifies the length of the stop bit

Stop2	Stop1	Stop0	Stop Bit Length
0	1	0	1 stop bit
0	1	1	1.5 stop bits
1	0	0	2 stop bits

000- 001 and 110 - 111 Reserved

Channel Option Register 3 *(cont.)*
Bisynchronous Mode

7	6	5	4	3	2	1	0
sndpad	S55	FCSPre	FCS	Idle	pad2	pad1	pad0

Bit 7 1 = CL-CD2400 will send pad character(s) before sending SYN when coming out of the Mark Idle Mode.
0 = CL-CD2400 will not send any pad character.

Bit 6 S55 = Send pad pattern
1 = 55h is sent as pad character.
0 = AAh is sent as pad character.

Bit 5 FCS Preset
0 = FCS is preset to all ones.
1 = FCS is preset to all zeros.

Bit 4 FCS Mode
1 = Disable FCS generation and checking, CL-CD2400 treats the entire frame as data.
0 = Normal FCS Mode. The CL-CD2400 generates and appends CRC on transmit and validates CRC on receive using CRC polynomial selected through CRC Polynomial Select Register.

Bit 3 Idle Mode
0 = idle in SYN
1 = idle in mark

Bits 2-0 character count – specifies the number of pad characters sent.

pad2	pad1	pad0	
0	0	0	reserved
0	0	1	1 pad character sent
0	1	0	2 pad characters sent
0	1	1	3 pad characters sent
1	0	0	4 pad characters sent

101 - 111 are reserved

Channel Option Register 3 *(cont.)*
X.21 Mode

7	6	5	4	3	2	1	0
0	SSDE	StrpSyn	SCDE	0	0	0	0

Bit 7 Reserved – ***must be zero.***

Bit 6 Steady State Detect Enable, when set to a 1, this bit enables the checking of special receive conditions relevant to X.21. The special conditions are:
1. All zeros
2. All ones
3. Alternating zeros and ones
4. Change in the condition of the CTS* Pin (CTS* is used as the I lead for a DTE, or C lead for a DCE).

To be detected as a special condition, a change must be present for at least 16-bit times. When detected, a receive exception interrupt is generated with the relevant status set in RISR. After detection of the special condition, no further data is passed to the host until different data is received.
In certain phases of the X.21 call setup, there is no character synchronization. When a data change takes place in a Non-Character Synchronous Phase, a partial character can be detected before the steady state is detected or character sync is achieved. In these conditions, the partial character is passed to the host as normal data.

Bit 5 Strip SYN. When this bit is set, SYN characters are treated as special receive conditions; when two SYN characters are received, a special character interrupt is generated (see RISR), and following SYN characters are filtered out. If the bit is not set, SYN characters are treated as normal data and passed to the host in good data interrupts; they are still used to obtain character synchronization with the data.

Bit 4 Special Character Detect Enable is only available when the SSDE Mode is enabled. If enabled, the characters programmed in SCHR1-3 are treated as the steady state conditions in the SSDE Mode; they are validated for two character times, a special character interrupt is generated, and then further repetitions of the same data pattern are filtered from the data stream.

Bit 3-0 Reserved – *must be zero.*

Channel Option Register 4 {COR4} 16 15 B R/W

(Modem Change Options and FIFO Transfer Threshold)

7	6	5	4	3	2	1	0
DSRzd	CDzd	CTSzd	0	FIFO Threshold			

Bit 7 DSRzd = 1
Detect one to zero transition on the DSR* input (zero to one transition of DSR (MSVR) Bit)

Bit 6 CDzd = 1
Detect one to zero transition on the CD* input (zero to one transition of CD (MSVR) Bit)

Bit 5 CTSzd = 1
Detect one to zero transition on the CTS* input (zero to one transition of CTS (MSVR) Bit)

Bits 3-0 FIFO Threshold in characters – Note: the maximum value allowable for this field is 12 (0C hex). These 4 bits, binary encoded field, set the FIFO transfer threshold FOR BOTH TRANSMIT AND RECEIVE FIFOs, FOR BOTH INTERRUPT AND DMA TRANSFER MODES.

In the Asynchronous Mode, a Good Data transfer will be initiated for the number of characters in the FIFO greater the specified threshold. Receive timeout and the occurrence of a receive data exception are also cause to initiate a receive transfer. In Synchronous modes, data transfer will be initiated when the number of characters in the FIFO is greater than the specified threshold. An end of frame is also cause to initiate a receive transfer.

For transmit operation, the CL-CD2400 will attempt to refill the transmit FIFO when the empty space in the FIFO is greater than the set threshold. In the case of synchronous frame transmissions, the CL-CD2400 will stop refilling the transmit FIFO once the last character in the frame has been transferred to the FIFO.

Channel Option Register 5 {COR5} 17 14 B R/W

7	6	5	4	3 2 1 0
DSRod	CDod	CTSod	0	Rx Flow Control Threshold

This register is used to define the current state change options to be monitored.

Bit 7 DSRod = 1
Detect zero to one transition on DSR input (one to zero transition of DSR (MSVR) Bit)

Bit 6 CDod = 1
Detect zero to one transition on CD input (one to zero transition of CD (MSVR) Bit)

Bit 5 CTSod = 1
Detect zero to one transition on CTS input (one to zero transition of CTS (MSVR) Bit)

Bits 3-0 Receive Flow Control FIFO Threshold
These 4 bits, binary encoded field, define the Receive FIFO hardware flow control threshold. It sets the threshold in the receive FIFO at which automatic hardware (DTR/DSR) flow control is activated. A threshold value of zero disables the hardware flow control mechanism. When the number of characters in the Receive FIFO exceeds this threshold, the DTR* Pin deasserts; when the number of characters is equal to or less than the threshold, DTR* is asserted.

Channel Option Register 6 {COR6} 1B 18 B R/W
Asynchronous Mode

7	6	5	4	3	2	1	0
IgnCR	ICRNL	INLCR	IgnBrk	NBrkInt	ParMrk	INPCK	ParInt

CR is defined as 0D hex, NL as 0A hex and NULL as 00 hex.

Bit 7-5 These three bits are used to enable translation of received CR/NL characters as follows:

IgnCr	ICrRNL	INLCR	
0	0	0	– no special action on CR and NL
0	0	1	– NL translated to CR
0	1	0	– CR translated to NL
0	1	1	– CR translated to NL and NL translated to CR
1	0	0	– CR discarded
1	0	1	– CR discarded and NL translated to CR
1	1	0	– CR discarded
1	1	1	– CR discarded and NL translated to CR

Bit 4-3 Break Action – These bits determine the action taken after a break condition is received.

IgnBrk	NBrkInt	
0	0	– Generate an exception interrupt
0	1	– Translate to a NULL character
1	0	– Reserved
1	1	– Discard character

Bit 2-0 Parity/framing error actions – These bits determine the action taken when a parity or framing error is received.

Following the generation of a break exception interrupt, a receive exception interrupt will be generated with RET Bit (RISRI) set when the end of break is detected. The RET interrupt must be enabled in IER to enable this feature.

ParMrk	INPCK	ParInt	
0	0	0	– Generated an exception interrupt
0	0	1	– Translated to a NULL character
0	1	0	– Ignore error; character passed on as good data
0	1	1	– Discard error character
1	0	0	– Reserved
1	0	1	– Translate to a sequence of FF NULL and the error character and pass on as good data
1	1	0	– Reserved
1	1	1	– Reserved

Channel Option Register 7 {COR7} 04 07 B R/W
Asynchronous Mode

7	6	5	4	3	2	1	0
IStrip	LNE	FCErr	0	0	0	ONLCR	OCRNL

CR is defined as 0D hex., NL as 0A hex. and NULL as 00 hex.

Bit 7 IStrip – when this bit is set, the receive characters are stripped to 7 bit, after all other processing.

Bit 6 LNext – this bit enables the LNext option.
0 = All receive characters are processed for special character detection.
1 = The character following the LNext character is not processed for special character matching or flow control.

This provides a mechanism to transfer flow control and special characters as normal data, without invoking flow control action in the CL-CD2400, and without generating special interrupts. The LNext character is defined in the LNXT Register, and, when processed, is always passed to the host CPU as normal data.

Bit 5 Flow control on error characters
0 = Characters received with an error will not be processed for special character/flow control matching.
1 = All receive characters, even those with errors, will be processed for special character/flow control processing.

Bit 4-2 Reserved

Bit 1-0 Transmit processing for CR and NL; these bits define Translation Mode when CR and/or NL are present in the transmit data.

ONLCR	OCRNL	
0	0	– No special action
0	1	– CR translated to NL
1	0	– NL translated to the sequence CR NL
1	1	– CR translated to NL, and NL translated to the sequence CR NL

Special Character Registers
Asynchronous Mode

Special character registers can be used for detecting specific receive characters in the incoming data stream, and can be used to transmit character (via STCR) preempting any data in the transmit FIFO.

Special Character Register1	{SCHR1}	1C	1F	B	R/W Async
Special Character Register2	{SCHR2}	1D	1E	B	R/W Async

Special characters 1 and 2 are used in conjunction with the SCDE Bit of COR3 to detect incoming characters; when both SCDE and TxIBE (COR2) are set, they define the in-band flow control characters XON and XOFF.

SCHR1 = XON
SCHR2 = XOFF

In addition to the SCDE and TxIBE bits, if the FCT Bit (COR3) is set when flow control characters are received, they will be stripped from the data stream.

Special Character Register3	{SCHR3}	1E	1D	B	R/W Async
Special Character Register4	{SCHR4}	1F	1C	B	R/W Async

Special characters 3 and 4 are used in conjunction with the ESCDE Bit of COR3 to detect characters in the receive data stream and generate receive special character interrupts.

NOTE: Special characters 3 and 4 are not stripped from the data stream if Flow Control Transparency (FCT) Mode is enabled.

Special Character Range

Special Character Range low	{SCRl}	20	23	B	R/W Async
Special Character Range high	{SCRh}	21	22	B	R/W Async

(Async) Special Character Range low and high

These bytes define an inclusive range for special character recognition in the Asynchronous Mode. It may be useful for identifying that a received character is within a user defined range and is, for example, a control character.

LNext Character	{LNXT}	2D	2E	B	R/W

(Async)

This register defines the LNext character. If the LNext function is enabled (Bit 6 of OCR7), the CL-CD2400 will examine received characters and compare them against this value. If a match occurs, this character and the following will be placed in the FIFO without any special processing. In effect, the LNext function causes the CL-CD2400 to ignore characters with special meaning, such as flow control characters. There are two exceptions. If the character following the LNext character is either a break or an errored character, LNext will be placed in the FIFO, and the following character will be treated as it normally would be for these error conditions.

Rx Frame Address Registers

Receive Frame Address Register1	{RFAR1}	1C	1F	B	R/W Sync
Receive Frame Address Register2	{RFAR2}	1D	1E	B	R/W Sync
Receive Frame Address Register3	{RFAR3}	1E	1D	B	R/W Sync
Receive Frame Address Register4	{RFAR4}	1F	1C	B	R/W Sync

Synchronous Modes

Reception of a frame can be qualified with a matched one or two byte address field either as 4 one-byte alternatives or 2 two-byte alternatives. Control of how the ADR Registers are used for address recognition is detailed in the description of the Channel Option Registers.

CRC Polynomial Select Register {CPSR} D4 D6 B R/W

7	6	5	4	3	2	1	0
res	res	res	res	res	res	res	poly

Bits 7-1 Reserved

Bit 0 Polynomial select
1 = CRC-16 polynomial (normally used for bisync protocol and preset to 0's) [x**16 + x**15 + x**2 + 1].
0 = CRC V.41 polynomial (normally used for HDLC protocol and preset to 1's) [x**16 + x**12 + x**5 + 1]

6.3 Bit Rate and Clock Options Registers

Receive Baud Rate Generator Registers

Receive Baud Rate Period Register {RBPR} C9 CB B R/W

7	6	5	4	3	2	1	0
Receive Baud Rate Period (Divisor)							

This register contains the preload value for the receive baud rate counter. When using an internal clock option or an n-times external clock, the preload value, in conjunction with the receiver clock source chosen, will determine the receive baud rate. If a 1x external clock is used, a value of 01h must be loaded in the RBPR.

Receive Clock Option Register {RCOR} CA C8 B R/W

7	6	5	4	3	2	1	0
TLVal	res	dpllEn	Dpllmd1	Dpllmd0	ClkSel2	ClkSel1	ClkSel0

This register is used to select the DPLL Mode, and the desired clock source for the receive baud rate generator.

Bit 7 TLVal – Transmit Line Value
This reflects the logical value of the transmit data pin.

Bit 6 Reserved

Bit 5 DPLL enable
1 = DPLL is enabled
0 = DPLL is disabled

Bits 4-3 DPLL Mode select and used to select the type of data encoding used.

Dpllmd1	Dpllmd0	Encoding
0	0	NRZ
0	1	NRZI
1	0	Manchester
1	1	Reserved

Bits 2-0 These three bits are used to select the clock source for the receive baud rate generator or DPLL.

clkSel2	clkSel1	clkSel0	clock source
0	0	0	Clk 0
0	0	1	Clk 1
0	1	0	Clk 2
0	1	1	Clk 3
1	0	0	Clk 4
1	0	1	Reserved
1	1	0	External clock
1	1	1	Reserved

NOTE: See the detailed description of clock options in Section 4.

6.3 Bit Rate and Clock Options Registers *(cont.)*

Transmit Baud Rate Generator Registers

Transmit Baud Rate Period Register {TBPR} C1 C3 B R/W

7	6	5	4	3	2	1	0
Transmit Baud Rate Period (Divisor)							

This register contains the preload value for the transmit baud rate count. When using one of the internal clocks or an n-times external clock, the preload value, in conjunction with the transmitter clock source chosen, will determine the transmit baud rate. If a 1x external clock or the Receive clock is used, a value of 01h must be loaded in the TBPR.

Transmit Clock Option Register {TCOR} C2 C0 B R/W

7	6	5	4	3	2	1	0
ClkSel2	ClkSel1	ClkSel0	res	Ext-1X	res	LLM	res

Transmit baud rate generator and to control Local Loopback Mode.

Bits 7-5 Are used to select the clock source for the transmit baud rate generator.

ClkSel2	ClkSel1	ClkSel0	Select
0	0	0	Clk 0
0	0	1	Clk 1
0	1	0	Clk 2
0	1	1	Clk 3
1	0	0	Clk 4
1	0	1	Reserved
1	1	0	External clock
1	1	1	Receive clock

NOTE: See the detailed description of clock options in Section 4.

Bit 4 Reserved

Bit 3 Times 1 external clock. This bit is set to 1 when user supplies the data clock on TxC[i] Pin whose frequency is equal to the transmit data rate. When in this mode, the TBPR must be programmed to 01h.

Bit 2 Reserved

Bit 1 Local Loopback Mode
'1' enables the Local Loopback Mode
'0' disables the Local Loopback Mode

Bit 0 Reserved

6.4 Channel Command and Status Registers

Channel Command Register {CCR} 10 13 B R/W

7	6	5	4	3	2	1	0
0	ClrCh	InitCh	RstAll	EnTx	DisTx	EnRx	DisRx

7	6	5	4	3	2	1	0
1	ClrT1	ClrT2	0	0	0	0	0

There are two CCR command sets. If Bit 7 is zero, the commands affect basic channel control. If Bit 7 is one, additional commands are available which control other functions.

The various command and control bits in this register perform largely independent functions. The host may assert multiple command bits to achieve the desired effect. The CL-CD2400 will clear the register to zero when it accepts and acts on a host command. The host must verify that the contents of this register are zero prior to issuing a new command. If the Reset All command is issued, all other commands are ignored. All other combinations are legal, and the order of processing will be as follows:

1. Clear Channel
2. Initialize Channel
3. Enable Receive
4. Disable Receive
5. Enable Transmit
6. Disable Transmit

Channel Control Commands

Bit 7 Must be zero

Bit 6 Clear Channel Command
When this command is issued, the CL-CD2400 clears the data FIFOs and current transmit and receive status of the channel. If the channel is currently transmitting a frame in synchronous protocol, the host should issue the Transmit Abort, Special Transmit Command, prior to issuing a Clear Command Command. Channel parameters will not be affected by a Channel Clear command. The Clear Channel command causes both receive and transmit FIFOs to be cleared, the transmitter and receiver to be disabled and all DMA Status Registers (DMABSTS, A/BRBSTS and A/BTBSTS) to be cleared.

Bit 5 Initialize Channel
If any change is made to the Protocol Mode Select bits in the Channel Mode Register (CMR) or to the Channel Option Register 1 (COR1), the channel must be reinitialized via this command. The command causes the internal protocol-specific registers to be initialized.

6.4 Channel Command and Status Registers

Channel Command Register *(cont.)*

Bit 4 Reset All
An on-chip firmware initialization of ***all channels*** is performed. All channel and global parameters are reset to their power on reset condition. This command is the strongest the host may issue. None of the other command bits is interpreted if the Reset All command is given. The host must re-initialize the CL-CD2400 following the execution of this command just as after a hardware Power-On Reset. When this command is completed, the GFRCR will be updated with the firmware revision code.

Bit 3 Enable Transmitter
Enables the transmitter by setting TxEn in the Channel Status Register (CSR). In Asynchronous Mode, this command also clears the transmit flow control options.

Bit 2 Disable Transmitter
Disables the transmitter by clearing TxEn in the Channel Status Register (CSR). In Asynchronous Mode, the transmit flow control bits will be cleared.

Bit 1 Enable Receiver
Enables the receiver by setting the RxEn Bit in the CSR. In Asynchronous Mode, the receive flow control bits will also be cleared.

Bit 0 Disable Receiver
Disables the receiver by clearing the RxEn Bit in the CSR. In Asynchronous Mode, the receive flow control bits will also be cleared.

Miscellaneous Channel Commands

Bit 7 Must be one. Either one or both of the timers may be cleared with a single command.

Bit 6 Clear Timer 1
General Timer 1 is cleared. Note that if the running timer value is 01h at the time this command is issued, there is a small chance that the timer will expire and cause a timer interrupt before the command is processed.

Bit 5 Clear Timer 2
General Timer 2 is cleared. Note that if the running timer value is 01h at the time this command is issued, there is a small chance that the timer will expire and cause a timer interrupt before the command is processed.

Bits 4-0 Reserved. Must be zero

Special Transmit Command Register {STCR} 11 12 B R/W

7	6	5	4	3	2	1	0
0	AbortTx	AppdCmp	0	SndSpc	SSPC2	SSPC1	SSPC0

The CL-CD2400 will clear the register to zero when it accepts a host CPU command.

Bit 6 Abort Transmission (HDLC)
Terminate the frame currently in transmission with an abort sequence. In DMA Mode, all data up to the next EOF is discarded.

Bit 5 Append Complete (Asynchronous DMA Mode)
This bit should be set by the host when the last addition has been made to the append buffer.

Bit 4 Reserved

Bit 3 Send Special Character(s) Command
A command used in Asynchronous Mode to send a user-defined special character or special-character sequence. The special character will be transmitted ahead on any data remaining in the FIFO.

Bits 2-0 Special Character Select

SSPC2	SSCP1	SSPC0	Function
0	0	0	Reserved
0	0	1	Send Special Character #1
0	1	0	Send Special Character #2
0	1	1	Send Special Character #3
1	0	0	Send Special Character #4
1	0	1	Reserved
1	1	0	Reserved
1	1	1	Reserved

Channel Status Register

This status register stores the current state of the channel. It may be read by the host at any time. The states of the RxEn and the TxEn Bits are controlled by host CPU commands to the CCR Register.

HDLC Mode

{CSR} 19 1A B R

7	6	5	4	3	2	1	0
RxEn	RxFlag	RxFrame	RxMark	TxEn	TxFlag	TxFrame	TxMark

Bit 7 Receiver Enable
0 = Receiver is disabled
1 = Receiver is enabled

Bit 6 Rx Flag
0 = Currently not receiving flag/sync
1 = Currently receiving flag/sync

Bit 5 Rx Frame
0 = Currently not receiving frame
1 = Currently receiving frame

Bit 4 Rx Mark
0 = Currently not receiving continuous mark
1 = Currently receiving continuous mark

Bit 3 Transmitter Enable
0 = Transmitter is disabled
1 = Transmitter is enabled

Bit 2 Tx Flag
0 = Currently not transmitting flag
1 = Currently transmitting flag

Bit 1 Tx Frame
0 = Currently not transmitting frame
1 = Currently transmitting frame

Bit 0 Tx Mark
0 = Currently not transmitting continuous ones
1 = Currently transmitting continuous ones

Channel Status Register *(cont.)*
Asynchronous Mode

If the host determines that a flow control state is inappropriate, it may be cleared by enabling or disabling the transmitter or receiver by CCR command.

7	6	5	4	3	2	1	0
RxEn	RxFloff	RxFlon	0	TxEn	TxFloff	TxFlon	0

Bit 7 Receiver Enable
0 = Receiver is disabled
1 = Receiver is enabled

Bit 6 Receive Flow Off
0 = Normal
1 = The CL-CD2400 has requested the remote to stop transmission (Send XOFF (command has been given to the channel). This bit will be reset when the CL-CD2400 has requested the remote to restart transmission, or when the receiver is enabled or disabled, or the channel is reset.

Bit 5 Receive Flow On
0 = Normal
1 = The CL-CD2400 has requested the remote to restart character transmission (Send XON command has been given to the channel). This bit is reset when the next (non-flow control) character is received, or when the receiver is enabled, or disabled or the channel is reset.

Bit 4 Reserved

Bit 3 Transmitter Enable
0 = Transmitter is disabled
1 = Transmitter is enabled

Bit 2 Transmit Flow Off
0 = Normal
1 = The CL-CD2400 has been requested by the remote to stop transmission. This bit is reset when the CL-CD2400 receives a request to resume transmission, or when the transmitter is enabled or disabled, or the channel is reset.

Bit 1 Transmit Flow On
0 = Normal
1 = The CL-CD2400 has been requested by the remote to resume transmission. This bit is reset once character transmission is resumed, or when the transmitter is enabled or disabled, or the channel is reset.

Bit 0 Reserved

Channel Status Register *(cont.)*
Bisynchronous Mode

7	6	5	4	3	2	1	0
RxEn	RxITB	RxFrme	0	TxEn	TxITB	TxFrme	0

Bit 7 Receiver Enable
0 = Receiver is disabled
1 = Receiver is enabled

Bit 6 Receive ITB. This bit indicates that the last frame received was terminated with an ITB. This means that the leading character of the next receive frame will be included in the BCC calculation.

Bit 5 Receive frame. This bit indicates that the CL-CD2400 is currently receiving a frame.

Bit 3 Transmitter Enable
0 = Transmitter is disabled
1 = Transmitter is enabled

Bit 2 Transmit ITB. This bit is set if the last frame transmitted ended in an ITB character. This means that the leading character of the next frame will be included in the BCC calculation.

Bit 1 Transmit Frame. When set, this bit indicates a frame is currently being transmitted.

X.21 Mode

7	6	5	4	3	2	1	0
RxEn	0	RxSpc	0	TxEn	0	TxSpc	0

Bit 7 Receiver Enable
0 = Receiver is disabled
1 = Receiver is enabled

Bit 5 Receive Special. When set, this indicates that the channel is in a steady state condition. Such conditions generate a receive special character interrupt.

Bit 3 Transmitter Enable
0 = Transmitter is disabled
1 = Transmitter is enabled

Bit 1 Transmit Special. When set this indicates that the CL-CD2400 is currently transmitting an ETC command, as defined in COR2.

Modem Signal Value Registers	{MSVR}			R/W
Modem Signal Value Register	{MSVR-RTS}	DC	DE	B R/W
	{MSVR-DTR}	DD	DF	B R/W

7	6	5	4	3	2	1	0
DSR/RxCk	CD/TxCk	CTS	DTRopt	res	PortID	DTR	RTS

Either of these registers is read to determine the current input levels on the input modem pins. Note that the pin definitions for these signals is negative true while the register values are positive true. Two registers are provided for control of the outputs DTR and RTS. Writing to the MSVR-DTR Register affects only the DTR Pin. Writing to the MSVR-RTS Register affects only the RTS Pin.

Bit 7 DSR – Current state of Data Set Ready input.
Note that this pin may optionally be used as Receive Clock input in synchronous modes. Reading the Receive Clock input gives an inverted sample of the state of the Receive Clock at the time the read occurs.

Bit 6 CD – Current state of Carrier Detect input.
Note that this pin may optionally be used as Transmit Clock input in synchronous modes. Reading the Transmit Clock input gives an inverted sample of the state of the Transmit Clock at the time the read occurs.

Bit 5 CTS – Current state of Clear to Send input.

Bit 4 DTR option – written via MSVR-DTR Register.
0 = value of DTR Bit is output on TXCout/DTR* Pin
1 = Transmit clock is output on TXCout/DTR* Pin

NOTE: If the Transmit clock source is a 1x clock on the TXCin/CD* Pin, this signal cannot be driven on TXCout/DTR*.

Bit 3 Reserved

Bit 2 PortID – This bit is read only. It can be used to determine which modem clock option pins are available.
0 = Device is the CL-CD2401
1 = Device is the CL-CD2400.

Bit 1 DTR – Current state of Data Terminal Ready output.

Bit 0 RTS – Current state of Request to Send output.

6.5 Interrupt Registers

Local Interrupt Vector Register {LIVR} 0A 09 B R/W

(The host effectively controls Bits 7 through 2; the chip provides bits 1 and 0 ***during an IACK Cycle***).

7	6	5	4	3	2	1	0
X	X	X	X	X	X	IT1	IT0

The CL-CD2400 has one Local Interrupt Vector Register per channel, each with 6 host-defined bits. The host may choose to embed the channel number and the protocol in use on the channel in the channel vector. The CL-CD2400 will supply two modified bits signifying the type of interrupt service required.

Bits 7-2 User-defined. These six bits may be used as the CL-CD2400 chip ID number.

Bits 1-0 Interrupt type 1-0. These three bits indicate the group/type of interrupt occurring.

IT[1:0]	GROUP/TYPE
01	Group 1: Modem Signal Change Interrupt/ General Timer Interrupt
10	Group 2: Transmit Data Interrupt
11	Group 3: Receive Data Interrupt
00	Group 3: Receive Exception Interrupt

6.5 Interrupt Registers *(cont.)*

Interrupt Enable Register {IER} 12 11 B R/W

7	6	5	4	3	2	1	0
Mdm	0	RET	0	RxD	TIMER	TxMpty	TxD

Bit 7 Modem Pin Change Detect
Master interrupt enable for modem change detect functions. The host may select which modem pins are watched for input change and select either or both directions of change by programming the change detect option bits in COR4 and COR5. A group1 type interrupt (see LIVR description) is generated from this enable.

Bit 6 Reserved

Bit 5 RET (Async)
In Asynchronous Mode, this bit enables a group 3 Receive Exception Timeout interrupt when a receive data timeout occurs with an empty receive FIFO. This provides a mechanism for the host to manage a partially full receive buffer when receive data stops.

Bit 4 Reserved

Bit 3 Rx Data
The receive FIFO threshold has been reached in Interrupt Transfer Mode, causing a group 3 Receive Data Interrupt. Any receive exception causes a group 3 Receive Exception Interrupt.

Bit 2 Timer
General Timer(s) Timeout
In Synchronous Mode this bit enables a group 1 interrupt when either timer reaches zero.

Bit 1 Tx Mpty
Transmitter Empty. When enabled, a group 2 interrupt is generated when the channel is completely empty of transmit data.

Bit 0 Tx Data
Any transmit exception or transmit FIFO threshold reached in Interrupt Transfer Mode. Group 2 interrupts will be generated at the end of transmit DMA buffers or when the FIFO threshold is reached in Interrupt Transfer Mode.

6.5 Interrupt Registers *(cont.)*

Local Interrupting Channel Register {LICR} 25 26 B R/W

7	6	5	4	3	2	1	0
X	X	X	X	C1	C0	X	X

This register contains the number of the interrupting channel being served. The channel number is always that of the current acknowledged interrupt. If no interrupt is acknowledged, the channel number is determined by the value in CAR.

Bits 7-4 User-defined.

Bits 3-2 Defines the interrupting channel number

C1	C0	Channel Number
0	0	Channel 0
0	1	Channel 1
1	0	Channel 2
1	1	Channel 3

Note that because the CL-CD2400 provides a unique Local Interrupt Vector Register for each channel, the host has the option to include the channel number within the interrupt vector.

Bits 1-0 User-defined.

6.5 Interrupt Registers *(cont.)*

Stack Register {STK} E0 E2 B R

7	6	5	4	3	2	1	0
CLvl [1]	MLvl [1]	TLvl [1]	0	0	TLvl [0]	MLvl [0]	CLvl [0]

This register is a four-deep-by-two-bit-wide stack that holds the internal interrupt nesting history. The stack is pushed from bits 7 and 0 towards the center during an interrupt acknowledge cycle and popped from the center during a write to an end of interrupt register.

CLvl [0:1] These bits provide the currently active interrupt level.

CLvl [1]	CLvl [0]	
0	0	No interrupt active; CAR provides the current channel number
0	1	Currently in a modem interrupt service, MIR provides the current channel number.
1	0	Currently in a transmit interrupt service, TIR provides the current channel number.
1	1	Currently in a receive interrupt service, RIR provides the current channel number.

MLvl [0:1] These bits hold a previously active interrupt now nested.

TLvl [0:1] These bits hold the oldest interrupt now nested 2 deep.

6.5.1 Receive Interrupt Registers

Receive Priority Interrupt Level Register {RPILR} E3 E1 B R/W

This register must be initialized by the host to contain the codes that will be presented on the address bus by the host system to indicate which of the three CL-CD2400 interrupt types (i.e., modem, transmit or receive) is being acknowledged when IACKIN* is asserted. The CL-CD2400 compares bits 0-6 in this register with A[0-6] to determine if the acknowledge level is correct. The value programmed in the MSB of the register has no effect on the IACK cycle.

RPILR must contain the code used to acknowledge receive interrupts.

NOTE: Bit 7 of the register is always read back as '0'. When each of the three Priority Interrupt Level Registers is programmed with the same value, they are internally prioritized, with receive as the highest priority, followed by transmit and modem.

Receive Interrupt Register {RIR} EF ED B R

7	6	5	4	3	2	1	0
Ren	Ract	Reoi	0	Rvct [1]	Rvct [0]	Rcn [1]	Rcn [0]

Bit 7 Ren
Receive enable is set by the CL-CD2400 to initiate a receive interrupt request sequence. It is cleared during a valid receive interrupt acknowledge cycle.

Bit 6 Ract
Receive active is set automatically when Ren is set, and the Fair Share logic allows the assertion of a receive interrupt request. It is cleared when the host CPU writes to the Receive End of Interrupt Register.

Bit 5 Reoi
Receive end of interrupt is set automatically when the host CPU writes to the receive end of interrupt register while in a receive interrupt routine.

Ren	Ract	Reoi	Sequence of Events
0	0	0	Idle
1	0	0	Receive interrupt requested, but not asserted
1	1	0	Receive interrupt is asserted
0	1	0	Receive interrupt is acknowledged
0	0	1	Receive interrupt service routine is completed

Bit 3-2 Rvct [0:1]
Receive vector bits are set by the CL-CD2400 to provide the lower two bits of the vector supplied to the host CPU during an interrupt acknowledge cycle. Receive good data vector is decoded as follows: Rvct [1] = 1, and Rvct [0] = 1. Receive exception vector is decoded as follows: Rvct [1] = 0, and Rvct [0] = 0.

Bits 1-0 Rcn [0:1]
Receive channel number is set by the CL-CD2400 to indicate the channel requiring receive interrupt service.

6.5.1 Receive Interrupt Registers *(cont.)*

Receive Interrupt Status Register {RISR} 8A 88 W R/W
Receive Interrupt Status Register low {RISRl} 8A 89 B R/W
HDLC Mode

7	6	5	4	3	2	1	0
0	EOF	rxabt	CRC	OE	ResInd	0	ClrDct

Bit 6 Receiving of a data frame is essentially complete.

Bit 5 Received Abort sequence terminating the frame.

Bit 4 CRC Error on current frame.

Bit 3 Overrun Error – indicates that new data has arrived, but the CL-CD2400 FIFO or holding registers are full. The new data is lost, and the overrun indication is flagged on the last character received before the overrun occurred. In HDLC and Bisync Modes, the remainder of a frame, following an overrun, will be discarded.

Bit 2 Residual Indication – indicates that the last character of the frame was a partial character.

Bit 0 Clear Detect – indicates an X.21 Data Transfer Phase Clear Signal has been detected. This is defined as two consecutive all-zero receive characters with the CTS* Pin high. Clear Detect Mode is enabled via COR1.

During an interrupt service routine, the host may use this register to provide a timer value as detailed in the Receive End of Interrupt Register. The host can only load one of the two timers in the interrupt service routine.

6.5.1 Receive Interrupt Registers *(cont.)*

Asynchronous Mode

7	6	5	4	3	2	1	0
Timeout	SCdet2	SCdet1	SCdet0	OE	PE	FE	Break

Bit 7 Timeout – indicates that the Receive FIFO is empty, and no data has been received within the receive timeout period. There is no data character associated with this status, and no other status bits are valid if the Timeout Bit is set.

Bits 6-4 Special Character Detect

SCdet[2:0]	Status
000	None detected
001	Special Character 1 matched
010	Special Character 2 matched
011	Special Character 3 matched (only if ESCDE is enabled in COR3)
100	Special Character 4 matched (only if ESCDE is enabled in COR3)
111	Character is within the inclusive range of the characters in the Special Character Range low and high registers (only if RngDE is enabled in COR3)

Special character match can be enabled for error characters via COR7.

Bit 3 Overrun Error – indicates that new data has arrived, but the CL-CD2400 FIFO or holding registers are full. The new data is lost and the overrun indication is flagged on the last character received before the overrun occurred.

Bit 2 Parity Error – indicates that a parity error has occurred.

Bit 1 Framing Error – indicates that a bad stop bit has been detected.

Bit 0 Break – indicates that a Break has been detected.

Bisynchronous Mode

7	6	5	4	3	2	1	0
0	EOF	rxabt	CRC	OE	0	0	0

Bit 6 Receiving of a data frame is essentially complete.

Bit 5 Received Abort sequence terminating the frame.

Bit 4 CRC error on current frame.

Bit 3 Overrun Error – indicates that new data has arrived, but the CL-CD2400 FIFO or holding registers are full. The new data is lost, and the overrun indication is flagged on the last character received before the overrun occurred.

During an interrupt service routine, the host may use this register to provide a timer value as detailed in the Receive End of Interrupt Register.

6.5.1 Receive Interrupt Registers *(cont.)*

X.21 Mode

7	6	5	4	3	2	1	0
LVal	SCdet2	SCdet1	SCdet0	OE	PE	O	LChg

For X.21 Operation, the CTS* Pin is used as the I lead for DTE or C lead for a DCE, a low level on CTS* is interpreted as an ON condition and a high level on CTS* as an OFF condition.

Bit 7 Lead value
0 = OFF
1 = ON

Bit 6-4 Special Character Detect all the following conditions must be met two consecutive character times.

SCdet[2:0]	Status
000	None detected
001	Matched the value in SCHR1
010	Matched the value in SCHR2
011	Matched the value in SCHR3
100	All 0 condition
101	All 1 condition
110	Alternating 1 0 condition
111	SYN detect

Bit 3 Overrun Error – indicates that new data has arrived, but the CL-CD2400 FIFO or holding registers are full. The new data is lost and the overrun indication is flagged on the last character received before the overrun occurred.

Bit 2 Parity Error – indicates that a parity error has occurred.

Bit 0 Lead Change this indicates a change of state on the CTS* Pin from the previous character time. Because there is no character sync during some phases of the X.21 call setup, an LChg indication may precede a special character interrupt.

During an interrupt service routine, the host may use this register to provide a timer value as detailed in the Receive End of Interrupt Register.

6.5.1 Receive Interrupt Registers *(cont.)*

Receive Interrupt Status Register high {RISRh} 8B 88 B R/W

7	6	5	4	3	2	1	0
Berr	EOF	EOB	0	BA/BB	0	0	0

This register is used in DMA Mode only.

Bit 7 Bus error (written by CL-CD2400)
0 = No bus error.
1 = Bus error was detected on the last transfer. The actual address at which the error occurred is available in the Receive Current Buffer Address Register. In response to a bus error status, the host has two possible options.
1. Retry from the next position in the buffer.
2. Terminate this buffer through setting TermBuff Bit in REOIR, and move onto the next.

Bit 6 Reception of a data frame is complete (Sync DMA Mode only).

Bit 5 The End Of a receive Buffer has been reached. Used only for DMA supported transmission. The end of one of the host supplied receive buffers has been reached.

Bit 3 Status occurs during buffer A or buffer B data transfer
0 = Buffer A
1 = Buffer B

Receive FIFO Output Count {RFOC} 33 30 B R

7	6	5	4	3	2	1	0
0	0	0	RxCt4	RxCt3	RxCt2	RxCt1	RxCt0

Bits 7-5 Reserved

Bits 4-0 Receive data count. If the receive channel is interrupt driven, a non-zero value in this bit field is the number of data characters available for transfer within the current receive interrupt.

Receive Data Register {RDR} F8 F8 B R
(Read Only)

7	6	5	4	3	2	1	0
D7	D6	D5	D4	D3	D2	D1	D0

This virtual register accesses the receive data FIFO of a channel interrupting for receive data transfer. This register address is used for all channels to transfer receive FIFO data to the host, if programmed in Interrupt Transfer Mode. Data must be read as bytes, and follows the rules of Section 6.3 for the positioning of valid data on the bus. If the BYTESWAP Pin is high, data is valid on A/D[0:7], if BYTESWAP is low, data is valid on A/D[8:15]. This is true because the RDR is on an even address.

6.5.1 Receive Interrupt Registers *(cont.)*

Receive End Of Interrupt Register {REOIR} 87 84 B W

7	6	5	4	3	2	1	0
TermBuff	DiscExc	SetTm2	SetTm1	Notransf	Gap2	Gap1	Gap0

The CL-CD2400 interprets values written to this register at the completion of all Receive interrupts.

Bit 7 Terminate Current DMA Buffer. If this bit is set, the current buffer will be switched to the other buffer. If the buffer is terminated and the discard exception character is not selected, the character on which the exception occurred, is written to the next buffer (DMA Mode only).

Bit 6 Discard Exception Character. (DMA Mode only) Data Transfer Mode.

Bit 5 Set general timer 2 in Synchronous Modes.
0 = do not set general timer
1 = load the value, to general timer 2, provided in RISRI.

Bit 4 Set general timer 1 in Synchronous Modes.
0 = do not set general timer 1
1 = load the value, to the high byte of general timer 1, provided in RISRI.

At the end of an interrupt service routine, the user may set a timer by setting a timer value in the Receive Interrupt Status Register. When the timer reaches 0, CL-CD2400 will generate a modem/timer group interrupt to the host.

Bit 3 No transfer of data. This bit must be set by the host, if no data is transferred from the receive FIFO during a receive interrupt.

Bits 2-0 Size of the optional gaps to be left in DMA Buffer, starting at the current location, before resuming data transfer. The CL-CD2400 will move its buffer address pointer forward the selected number of bytes. It will not write to any location 'in the gap.' If the gap is large enough to complete or extend beyond the end of the current buffer, it will be completed, and the gap continued in the other receive buffer. If the discard exception character is not selected, the character on which the exception occurred is written to the buffer following the gap.

6.5.2 Transmit Interrupt Registers

Transmit Priority Interrupt Level Register {TPILR} E2 E0 B R/W

This register must be initialized by the host to contain the codes that will be presented on the address bus by the host system to indicate which of the three CL-CD2400 interrupt types (i.e., modem, transmit or receive) is being acknowledged when IACKIN* is asserted. The CL-CD2400 compares bits 0-6 in this register with A[0-6] to determine if the acknowledge level is correct. The value programmed in the MSB of the register has no effect on the IACK cycle.

TPILR must contain the code used to acknowledge transmit interrupts.

NOTE: Bit 7 of this register is always read back as '0'. When each of the three Priority Interrupt Level Registers is programmed with the same value, they are internally prioritized, with receive as the highest priority, followed by transmit and modem.

Transmit Interrupt Register {TIR} EE EC B R

7	6	5	4	3	2	1	0
Ten	Tact	Teoi	0	Tvct [1]	Tvct [0]	Tcn [1]	Tcn [0]

Bit 7 Ten
Transmit enable is set by the CL-CD2400 to initiate a transmit interrupt request sequence. It is cleared during a valid transmit interrupt acknowledge cycle.

Bit 6 Tact
Transmit active is set automatically when Ten is set, and the Fair Share logic allows the assertion of a transmit interrupt request. It is cleared when the host CPU writes to the Transmit End of Interrupt Register.

Bit 5 Teoi
Transmit end of interrupt is set automatically when the host CPU writes to the transmit end of interrupt register while in a transmit interrupt routine.

Ten	Tact	Teoi	Sequence of Events
0	0	0	Idle
1	0	0	Transmit interrupt requested, but not asserted
1	1	0	Transmit interrupt is asserted
0	1	0	Transmit interrupt is acknowledged
0	0	1	Transmit interrupt service routine is completed

Bits 3-2 Tvct [0:1]
Transmit vector bits are set by the CL-CD2400 to provide the lower two bits of the vector supplied to the host CPU during an interrupt acknowledge cycle. Transmit vector is decoded as follows: Tvct [1] = 1, and Tvct [0] = 0.

Bits 1-0 Tcn [0:1]
Transmit channel number is set by the CL-CD2400 to indicate the channel requiring transmit interrupt service.

6.5.2 Transmit Interrupt Registers *(cont.)*

Transmit Interrupt Status Register {TISR} 89 8A B R/W

7	6	5	4	3	2	1	0
Berr	EOF	EOB	UE	BA/BB	0	TxEmpty	TxDat

When the host receives a transmit interrupt, the following status is provided in this register:

Bit 7 Bus error (written by CL-CD2400)
0 = no bus error
1 = bus error was detected on the last transfer. The actual address at which the error was detected is available in the Transmit Current Buffer Address Register. In response to a bus error status, the host has two possible options:
1. Continue from the next position in the buffer, by resetting (0) TermBuff Bit in TEOIR.
2. Terminate this buffer through TEOIR and move onto the next.

Bit 6 Transmit end of frame indication in DMA Mode. The interrupt for this condition is generated when the final data character of a transmit frame is transferred to the internal transmit FIFO. To maximize frame throughput, the host will wish to initiate the next frame transmission when EOFrame becomes true. If a half-duplex serial line must be 'turned around,' the host must wait for TxEmpty to become true. The Interrupt Enable Register contains bits that permit the host to choose which of these conditions will actually cause the interrupt.

Bit 5 The End Of a transmit Buffer has been reached. Used only for DMA supported transmission.

Bit 4 Transmit Underrun. Data was not available in time during synchronous frame transmission, and an underrun error has occurred. Following a transmit underrun, all data up to the next EOF is discarded.

Bit 3 Buffer that has exception
0 = Buffer A
1 = Buffer B

Bit 1 Transmitter Empty, the frame or character transmission is completed, and the transmit serial output is in its selected idle condition.

Bit 0 Transmit Data is below the FIFO threshold.

During an interrupt service routine, the host may use this register to provide a timer value as detailed in the Transmit End of Interrupt Register. The host can only load one of the two timers in each interrupt service routine.

6.5.2 Transmit Interrupt Registers *(cont.)*

Transmit FIFO Transfer Count {TFTC} 83 80 B R

7	6	5	4	3	2	1	0
0	0	0	TxCt4	TxCt3	TxCt2	TxCt1	TxCt0

Bits 7-5 Reserved

Bits 4-0 Transmit data count. If the Transmit channel is interrupt driven, a non-zero value is a request for data. These bits give the number of spaces available in the Transmit FIFO.

End Of Interrupt Registers

The End Of Interrupt Registers must be written to by the corresponding host interrupt service routines to signal to the CL-CD2400 that the current interrupt service is concluded. This must be the last access to the CL-CD2400 during an interrupt service routine. Writing to these registers will generate an internal End of Interrupt signal which pops the CL-CD2400 interrupt context stack.

Depending on the circumstances of an individual interrupt service, the host may be required to pass a parameter to the CL-CD2400 through these registers.

Transmit Data Register {TDR} F8 F8 B W
(Write Only)

7	6	5	4	3	2	1	0
D7	D6	D5	D4	D3	D2	D1	D0

This virtual register accesses the transmit data FIFO of a channel interrupting for transmit data transfer. This register address is used for all channels to transfer transmit FIFO data to the host, if programmed in Interrupt Transfer Mode. Data must be written as bytes, and follows the rules of Section 6.3 for positioning valid data on the bus. If the BYTESWAP Pin is high, data must be valid on A/D[0:7]; if BYTESWAP is low, data must be valid on A/D[8:15], because the TDR is on an even address.

6.5.2 Transmit Interrupt Registers *(cont.)*

Transmit End Of Interrupt Register {TEOIR} 86 85 B W

7	6	5	4	3	2	1	0
TermBuff	EOF	SetTm2	SetTm1	Nontransf	0	0	0

Bit 7 1 = Terminate buffer in DMA Mode forces the current buffer to be discarded.

Bit 6 End of Frame in synchronous modes using interrupt-driven data transfer.
0 = this data transfer does not complete the frame/block.
1 = this data transfer does complete the frame/block.

Bit 5 Set general timer 2 in synchronous modes.
0 = do not set general timer 2.
1 = load the value, provided in TISR, to general timer 2.

Bit 4 Set general timer 1 in synchronous modes.
0 = do not set general timer 1.
1 = load the value, provided in TISR, to the high byte of general timer 1.

At the end of an interrupt service routine, the user may set a timer by setting a timer value in the Transmit Interrupt Status Register. When the timer reaches 0, CL-CD2400 will generate a modem/timer group interrupt to the host.

Bit 3 No transfer of data. This bit must be set by the host, if no data is transferred to the transmit FIFO during a data transfer interrupt.

Bits 2-0 Reserved

6.5.3 Modem/Timer Interrupt Registers

Modem Priority Interrupt Level Register {MPILR} E1 E3 B R/W

This register must be initialized by the host to contain the codes that will be presented on the address bus by the host system to indicate which of the three CL-CD2400 interrupt types (i.e., modem, transmit or receive) is being acknowledged when IACKIN* is asserted. The CL-CD2400 compares bits 0-6 in this register with A[0-6] to determine if the acknowledge level is correct. The value programmed in the MSB of the register has no effect on the IACK cycle.

MPILR must contain the code used to acknowledge modem/timer interrupts.

NOTE: Bit 7 of this register is always read back as '0'. When each of the three Priority Interrupt Level Registers is programmed with the same value, they are internally prioritized, with receive as the highest priority, followed by transmit and modem.

Modem Interrupt Register {MIR} ED EF B R

7	6	5	4	3	2	1	0
Men	Mact	Meoi	0	Mvct [1]	Mvct [0]	Mcn [1]	Mcn [0]

Bit 7 Men
Modem enable is set by the CL-CD2400 to initiate a modem interrupt request sequence. It is cleared during a valid modem interrupt acknowledge cycle.

Bit 6 Mact
Modem active is set automatically when Men is set, and the Fair Share logic allows the assertion of a modem interrupt request. It is cleared when the host CPU writes to the Modem End of Interrupt Register.

Bit 5 Meoi
Modem end of interrupt is set automatically when the host CPU writes to the Modem End of Interrupt Register while in a modem interrupt routine.

Men	Mact	Meoi	Sequence of Events
0	0	0	Idle
1	0	0	Modem interrupt requested, but not asserted
1	1	0	Modem interrupt is asserted
0	1	0	Modem interrupt is acknowledged
0	0	1	Modem interrupt service routine is completed

Bits 3-2 Mvct [0:1]
Modem vector bits are set by the CL-CD2400 to provide the lower two bits of the vector supplied to the host CPU during an interrupt acknowledge cycle. Modem vector is decoded as follows: Mvct [1] = 0,and Mvct [0] = 1.

Bits 1-0 Mcn [0:1]
Modem channel number is set by the CL-CD2400 to indicate the channel requiring modem interrupt service.

6.5.3 Modem/Timer Interrupt Registers *(cont.)*

Modem (/Timer) Interrupt Status Register {MISR} 88 8B B R/W

7	6	5	4	3	2	1	0
DSRChg	CDChg	CTSChg	res	res	res	Timer 2	Timer 1

When the host receives a modem interrupt, the following status is provided in this register:

Bit 7 DSR Changed. A logic '1' indicates that a change has been detected on the DSR* input. The change detect is programmed in COR4 and COR5.

Bit 6 CD Changed. A logic '1' indicates that a change has been detected on the CD* input. The change detect is programmed in COR4 and COR5.

Bit 5 CTS Changed. A logic '1' indicates that a change has been detected on the CTS* input. The change detect is programmed in COR4 and COR5.

Bits 4-2 Reserved

Bit 1 General Timer 2 timed out (count reaches zero before being reset or disabled).

Bit 0 General Timer 1 timed out (count reaches zero before being reset or disabled).

During an interrupt service routine, the host may use this register to provide a timer value as detailed in the Modem End of Interrupt Register. The host can only load one of the two timers in each interrupt service routine.

Modem End Of Interrupt Register {MEOIR} 85 86 B W

7	6	5	4	3	2	1	0
0	0	SetTm2	SetTm1	0	0	0	0

Bit 5 Set general timer 2 in synchronous modes.
0 = do not set general timer 2.
1 = load the value, provided in MISR, to general timer 2.

Bit 4 Set general timer 1 in synchronous modes.
0 = do not set general timer 1.
1 = load the value, provided in MISR, to the high byte of general timer 1.

At the end of an interrupt service routine, the user may set the timer by setting a timer value in the Modem Interrupt Status Register. When the timer reaches 0, CL-CD2400 will generate a modem/timer group interrupt to the host.

6.6 DMA Registers

DMA Mode Register (write only) {DMR} F4 F6 B W

7	6	5	4	3	2	1	0
0	0	0	0	ByteDMA	0	0	0

This register is write only and not available as read, no misoperation will occur if the register is read but the read value will not be consistent.

Bit 7-4 Reserved – When writing to this register, these bits must be written as zero.

Bit 3 Byte DMA 0 = The CL-CD2400 will attempt to perform 16-bit data transfers whenever possible, and 8-bit only when necessary (when only one byte is available or odd address boundaries).
1 = The CL-CD2400 will always perform 8-bit DMA transfers, the position of the data on the bus will still follow the normal rules relating to the BYTESWAP Pin.

Bit 2-0 Reserved – When writing to this register, these bits must be written as zero.

Bus Error Retry Count {BERCNT} 8D 8E B R/W

7	6	5	4	3	2	1	0
Binary Value							

When this register is programmed to zero, any bus error causes a receive/transmit interrupt to be generated and DMA Operations suspended to the buffer in error, until the interrupt is processed by the host CPU.

When this register contains a non-zero value and when a bus error occurs, the CL-CD2400 will retry the same DMA Operation and decrement the register value by one. When the value reaches zero, the next bus error will cause an interrupt, at which time a new count may be loaded by the host CPU.

6.6 DMA Registers *(cont.)*

DMA Buffer Status {DMABSTS} 1A 19 B R

7	6	5	4	3	2	1	0
TDAlign	RstApd	CrtTBuf	Append	NtBuf	Tbusy	Nrbuf	Rbusy

When CL-CD2400 requires an external buffer for DMA transfer, it checks Ntbuf/Nrbuf Bits to decide which buffer to use. Once CL-CD2400 starts using the buffer, it toggles Ntbuf/Nrbuf Bits, and sets Tbusy/Rbusy Bits. Ntbuf and Nrbuf are set to Buffer A at system initialization.

Bit 7 This status bit is used internally to manage data alignment in the transmit FIFO.

Bit 6 Reset Append Mode is set after the terminate append buffer command in STCR has been recognized, and is cleared after the remaining data has been flushed from the buffer.

Bit 5 Current transmit buffer is used internally to mark the actual buffer in use.

Bit 4 Append (Only Buffer A can be used as an append buffer)
Transmit append buffer usage indicator
0 = Append buffer is not in use.
1 = Append buffer is in use.

Bit 3 Ntbuf
Next transmit buffer
0 = Buffer A is the next transmit buffer
1 = Buffer B is the next transmit buffer

This bit is toggled when transmission is started from a buffer, i.e., when data is first read from Buffer A, the bit is set to indicate that Buffer B is next.

Bit 2 Tbusy
Current transmit buffer is in use?
0 = No buffer is in use.
1 = Current transmit buffer is in use.

Bit 1 Nrbuf
Next receive buffer
0 = Buffer A is the next receive buffer
1 = Buffer B is the next receive buffer

This bit is toggled when receive data is first written to a buffer, i.e., when data is first written to Buffer A, the bit is set to indicate Buffer B is next.

Bit 0 Rbusy
Current receive buffer is in use?
0 = No buffer is in use.
1 = Current receive buffer is in use.

6.6.1 DMA Receive Registers *(cont.)*

A Receive Buffer Address Lower	{ARBADRL}	40	42	W	R/W
A Receive Buffer Address Upper	{ARBADRU}	42	40	W	R/W
B Receive Buffer Address Lower	{BRBADRL}	44	46	W	R/W
B Receive Buffer Address Upper	{BRBADRU}	46	44	W	R/W

Receive Buffer Address Registers A and B (32-bit)

These registers contain the start addresses of two external buffers which will be used by the CL-CD2400 to store the next two receive data blocks. They are written to by the host and copied internally to control the data transfer to the memory.

A Receive Buffer Byte Count	{ARBCNT}	48	4A	W	R
B Receive Buffer Byte Count	{BRBCNT}	4A	48	W	R

These registers contain the number of bytes stored in the external data buffers by the CL-CD2400. The count is updated after a block of data is moved to memory and the buffer is terminated. As initially written by the host, the register contains the number of bytes which the buffer can hold.

A Receive Buffer Status	{ARBSTS}	4C	4F	B	R
B Receive Buffer Status	{BRBSTS}	4D	4E	B	R

Receive Buffer Status (8-bit)

7	6	5	4	3	2	1	0
Berr	EOF	EOB	0	0	0	0	2400own

These registers contain the current status of associated receive buffers and enable the buffers to be passed between the host and CL-CD2400. Status bits are defined as:

Bit 7 — Bus error (set by the CL-CD2400 and cleared by the host CPU)
0 = no bus error
1 = bus error occurred on the last transfer; the suspect address is available in RCBADR

Bit 6 — End of frame (set by the CL-CD2400 and cleared by the host CPU)
0 = this buffer does not terminate a frame
1 = this buffer terminates a frame

Bit 5 — Buffer complete (set by the CL-CD2400 and cleared by the host CPU)
0 = buffer not complete
1 = buffer complete

Bit 0 — Ownership of the transfer buffer (set by the host CPU and cleared by the CL-CD2400)
0 = buffer not free to be used by CL-CD2400
1 = buffer free to be used by CL-CD2400

When the buffer completed bit is set by the CL-CD2400, the buffer is free for the host to process (RBCNT information is updated to the number of bytes available in the buffer, and a new buffer can be allocated).

6.6.1 DMA Receive Registers *(cont.)*

Receive Current Buffer Address Lower	{RCBADRL}	3C	3E	W	R
Receive Current Buffer Address Upper	{RCBADRU}	3E	3C	W	R

Contains the address of the current DMA buffer being used for receive data. Updated at the end of receive data transfers. This register is for the private use of the CL-CD2400 in managing DMA transfers. In Asynchronous Mode, the host may read this register during a Receive Exception Interrupt to determine how much data is in the buffer. The address is the location of the next character to be transferred to the buffer. The host will need that information to process newly arrived data in the buffer if it is being used in the Append Mode, and the data timeout has occurred. It is also needed if an exception has occurred, and a gap is to be left in the FIFO for the insertion of status information by the host. In the case of a bus error during receive data transfer, this register provides the start address of the transfer causing the bus error.

6.6.2 DMA Transmit Registers

A Transmit Buffer Address Lower	{ATBADRL}	50	52	W	R/W
A Transmit Buffer Address Upper	{ATBADRU}	52	50	W	R/W
B Transmit Buffer Address Lower	{BTBADRL}	54	56	W	R/W
B Transmit Buffer Address Upper	{BTBADRU}	56	54	W	R/W

Transmit Buffer Address Registers A and B (32-bit)

These registers contain the start addresses of two external buffers which will be used by the CL-CD2400 to transmit the next data blocks. They are written to by the host and copied internally to control the data transfer from the memory to the CL-CD2400 FIFO.

A Transmit Buffer Byte Count	{ATBCNT}	58	5A	W	R/W
B Transmit Buffer Byte Count	{BTBCNT}	5A	58	W	R/W

These registers contain the count of bytes in the buffers to be transmitted.

6.6.2 DMA Transmit Registers *(cont.)*

A Transmit Buffer Status	{ATBSTS}	5C	5F	B	R/W
B Transmit Buffer Status	{BTBSTS}	5D	5E	B	R/W

7	6	5	4	3	2	1	0
Berr	EOF	EOB	0	Append	0	INTR	2400own

This register contains the status of the associated transmit buffer, and enables successive buffers to be passed between the host and CL-CD2400. Status bits within the register are defined as:

Bit 7 Bus error (set by the CL-CD2400 and cleared by the host CPU)
0 = no bus error
1 = bus error occurred on the last transfer; the suspect address is available in TCBADR

Bit 6 End of Frame (set and cleared by host CPU)
0 = this buffer is not the last in frame/block
1 = this buffer is the last in frame/block

Bit 5 The End Of a Transmit Buffer has been reached. Used only for DMA supported transfer. The end of one of the host supplied transmit buffers has been reached. This bit is set by the CL-CD2400 and cleared by the host CPU.

Bit 3 Append (Asynchronous Mode; set and cleared by the host CPU)
0 = no data will be appended to the buffer
1 = data may be appended to buffer after tx started

Bit 2 Reserved

Bit 1 Interrupt
0 = no interrupt required after the buffer is sent
1 = interrupt required after the buffer is sent

Bit 0 Ownership of the transfer buffer (set by the host CPU and cleared by the CL-CD2400)
0 = buffer not ready to be used by CL-CD2400
1 = buffer is ready for CL-CD2400 to transmit

To start transmission of a buffer, the host must set the Transmit Buffer Address (ATBADR/BTBADR) and Transmit Buffer Count (ATBCNT/BTBCNT) Registers, and then set the 2400OWN Bit. If the CL-CD2400 is to generate and send the CRC for the frame, the CRC Bit in COR1 must be set. If the buffer contains the end of a frame, the EOF Bit must also be set. When the buffer has been sent, the EOB Bit will be set by the CL-CD2400, and 2400OWN will be reset, allowing a new buffer to be allocated.

Setting the Append Bit allows data to be added to the buffer after transmission has begun. In this mode, the host sets ATADR and ATCNT as normal, but when new data is appended to the buffer, the Transmit Buffer Count (ATBCNT/BTBCNT) can be updated. When the A buffer is used in Append Mode, the CL-CD2400 will not set the EOB Bit. When the host has completed use of the buffer, it must issue the Append Complete command through STCR. The CL-CD2400, upon transmitting the last characters from the buffer, will set EOB, thus allowing the host to allocate a new transmit buffer.

6.6.2 DMA Transmit Registers *(cont.)*

Transmit Current Buffer Address Lower	{TCBADRL}	38	3A	W	R
Transmit Current Buffer Address Upper	{TCBADRU}	3A	38	W	R

Contains the address into the current DMA buffer being used for transmit data. Updated at the end of transmit data transfers. In the case of a bus error during transmit data transfer, this register contains the start address of the transfer causing the bus error.

6.7 Timer Registers

Timer Period Register {TPR} D8 DA B R/W

7	6	5	4	3	2	1	0
Binary Value							

This register provides the initialization value for the timer prescaler which is itself clocked by a prescaled clock equal to system clock/2048. The timer prescaler establishes the clock for the various on-chip timers (including the general timers available to the host in the synchronous modes). The minimum value loaded in this register to maintain accuracy in the timer is 0A hex.

6.7 Timer Registers *(cont.)*

Receive Timeout Period Register	{RTPR}	26	24	W	R/W Async
Receive Timeout Period Register low	{RTPRl}	26	25	B	R/W Async
Receive Timeout Period Register high	{RTPRh}	27	24	B	R/W Async

(Async) Receive Timeout Period Register (16-bits)

This value sets the receive data timeout period. As each character is moved to the receive FIFO or the last data is transferred from the FIFO to the host, the Receive Timer (an internal timer) is reloaded with the Receive Timeout Period Register.. The Receive Timer is decremented on each 'tick' of the prescaler counter. If the Receive Timer reaches zero, it will cause a receive data interrupt.

General Timer 1	{GT1}	28	2A	W	R Sync
General Timer 1 low	{GT1l}	28	2B	B	R Sync
General Timer 1 high	{GT1h}	29	2A	B	R Sync

(Sync) General Timer 1

This 16-bit timer may be started by the user whenever it is inactive by writing a 16-bit timeout value to the register. When non-zero, it is decremented on each prescaler clock 'tick.' When it reaches zero, a Modem/Timer group interrupt is generated to the host. The timer may be disabled by Channel Command Register command. In addition, during a Receive or Transmit interrupt, the user may reload a running timer (high byte only) by providing a reload value in the Interrupt Status Register and a reload timer command in the End of Interrupt Register for the interrupt being serviced. Only one general timer may be restarted this way in a single-interrupt routine.

General Timer 2	{GT2}	2A	29	B	R Sync

(Sync) General Timer 2

This 8-bit timer may be started by the user whenever it is inactive by writing an 8-bit timeout value to the timer register. When non-zero, it is decremented on each prescaler clock 'tick.' When it reaches zero, a Modem/Timer group interrupt is generated to the host. The timer may be disabled by Channel Command Register command if the timer's current value is greater than 1. In addition, during a Receive or Transmit interrupt, the user may reload a running timer by providing a reload value in the Interrupt Status Register and a reload timer command in the End of Interrupt Register for the interrupt being serviced. Only one general timer may be restarted this way in a single-interrupt routine.

Transmit Timer Register	{TTR}	2A	29	B	R Async

(Async) Transmit Timer Register

This asynchronous mode timer is managed by the CL-CD2400 to implement embedded transmit delays when that option is used by the host.

7. APPLICATIONS

7.1 Cascading CL-CD2400s

In order to demultiplex the A/D[0:15] bus into separate address and data buses, external buffers and latches are required. To reduce external circuitry, these external devices can be shared in multi-CL-CD2400 applications. The common control lines (ADLD*, AEN*, DATDIR*, DATEN*) to the external devices are wire-ORed together. These pins are tri-state not open collector, but an external pull-up resistor (2.2K-5.0K) must be connected to each line to ensure a logic one when no CL-CD2400 is a bus master.

When no higher priority alternate bus masters are present, a daisy chain priority scheme can be implemented by wire ORing the BR* and BGACK* and connecting directly to the 680X0. The 680X0 BG* signal is then connected to the first device in the chain and daisy chained to the remaining devices. A lower priority bus master can then be connected at the end of the chain.

If a higher priority bus master is present, the BG* signal must be qualified before being passed into the highest priority CL-CD2400. If a priority encoded scheme is required, the BR* signals must be prioritized externally and BG* signals routed to individual devices.

7.2 Interface to a 32-Bit Data Bus

To interface to a 32-bit data bus, two 16-bit data buffers must be used to isolate the CL-CD2400 A/D[0:15] pins from either half of the 32-bit bus. The A[1] address pin determines if the lower or upper half of the data bus is in use for a particular bus cycle. The CL-CD2400 always drives all 16 data bits during a Register Read or a DMA Write Operation, regardless of the size of the actual transfer.

7.3 Initialization of the CL-CD2400

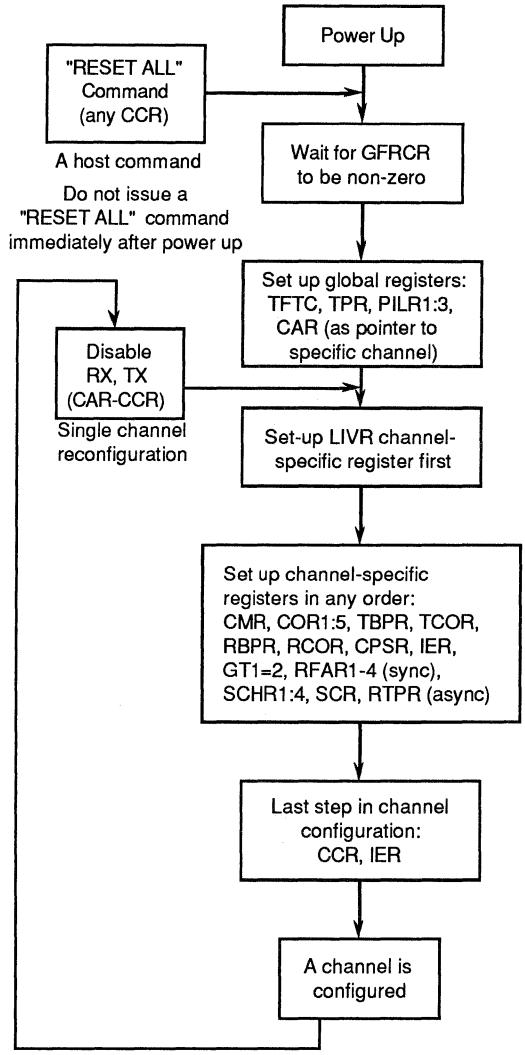


Figure 7–0. Initialization Sequence for the CL-CD2400

7.4 DMA Connections for the CL-CD2400

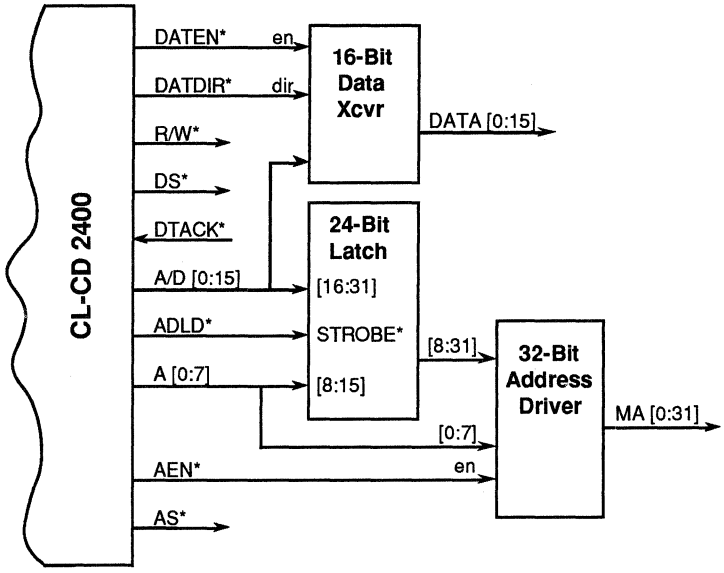


Figure 7–1. DMA Connections for the CL-CD2400

The 24-bit latch is a MUST.

The 16-bit xcvr is OPTIONAL depending on application.

The 32-bit driver is OPTIONAL depending on drive needs.

7.5 Interrupts

7.5.1 Hardware Interface Signals

The CL-CD2400 has five control signals that are used for the interrupt operation; IREQ[1], IREQ[2] and IREQ[3] are used to assert timer/modem signal changes, transmit and receive interrupts, respectively, to the host. They are open-drain outputs, therefore, an external pull-up resistor of 4-5K ohm is required on the IREQ lines. The interrupt acknowledge input, IACKIN*, and interrupt acknowledge output, IACKOUT*, are the other interrupt control signals. The CL-CD2400 is designed for systems that may require more than four channels. When using multiple CL-CD2400s, the interrupt request lines are connected in parallel; the IACKOUT* of the first is connected to the IACKIN* of the next, and so on, forming a daisy chain for the interrupt acknowledge input. The host drives the IACKIN* of the first CL-CD2400. Thus, when the leading CL-CD2400 sees an interrupt without an IREQ line asserted or does not see a match with the priority level, it will pass that interrupt acknowledge out to the next CL-CD2400 on its IACKOUT*. Optionally, the IREQ outputs can be wire-ORed together without altering the interrupt acknowledge cycle.

7.5.2 Registers of Interest

The following are registers of interest when using the interrupt schemes available in the CL-CD2400.

Interrupt Enable Register (IER): The bits in this register are used to enable or disable the various interrupt sources in the device.

Local Interrupt Vector Register (LIVR): The value in this register is output during the interrupt service routine. The upper 6 bits in this register are host-defined, and the other 2 bits are set by the CL-CD2400 to encode the type of interrupt service being requested. These 2 bits are valid only during an interrupt service cycle.

Modem Priority Interrupt Status Register(MPILR)

Transmit Priority Interrupt Status Register (TPILR)

Receive Priority Interrupt Status Register (RPILR)

The user must program these registers with the value that will be present on the address bus during an interrupt acknowledge cycle. MPILR must be programmed to contain the value used during a modem/timer interrupt acknowledge, TPILR must be programmed to contain the value used during a transmit interrupt acknowledge, and RPILR must be programmed to contain the value used during a receive interrupt acknowledge.

Modem Interrupt Status Register (MISR)

Transmit Interrupt Status Register (TISR)

Receive Interrupt Status Register (RISR)

These registers provide the status or cause for an interrupt of the corresponding type. In addition, they can also be used to pass the value of a timer to the CL-CD2400 at the end of an interrupt.

Modem End Of Interrupt Register (MEOIR)

Transmit End Of Interrupt Register (TEOIR)

Receive End Of Interrupt Register (REOIR)

There are several commands that can be issued through these registers as a result of the interrupt request. These registers must be written to at the end of the interrupt service routine; this should be the last access to the CL-CD24C0 during the interrupt service routine.

7.5.3 Programming the PILR Registers

As a part of the system initialization, the user must program the Priority Interrupt Level Registers (Receive, Modem and Transmit) with the value that will be present on the address bus during the interrupt acknowledge cycle. In most applications, the values contained in these registers are unique, but the user has the option of programming all of these registers with the same priority level. This section will discuss how the CL-CD2400 will behave during an interrupt acknowledge cycle when two or more PILR Registers are programmed with the same value. Also refer to Section 5.2.

When the CL-CD2400 determines that one or more groups of interrupts need service from the host, it asserts the IREQ line. More than one IREQ line can be active at the same time. For example, if only one IREQ line is active, the host asserts the interrupt acknowledge line and drives the priority level on the address bus.

Upon a match between the address bus and the PILR of the corresponding interrupt level, the CL-CD2400 drives the appropriate interrupt vector on the data bus to signify the type of interrupt service being requested.

When all of the PILR Registers are programmed to the same value, and multiple-IREQ lines are asserted, the CL-CD2400 uses the following hierarchy to determine which interrupt vector to drive on data bus:

Highest priority

Receive interrupt

Transmit interrupt

Modem interrupt

Lowest priority

Thus, the Receive interrupt is serviced first, then the Transmit interrupt, followed by the Modem interrupt.

7.5.4 Interrupts in the CL-CD2400

There are two possible reasons for the CL-CD2400 to interrupt the host processor:

1. CL-CD2400 wants (additional) data from the host system because number of free bytes in a transmit FIFO is at or below the threshold. Or, the CL-CD2400 wants to transfer (additional) data to the host system because the number of characters in a receive FIFO has reached the threshold. This interrupt is a request for transfer of data.
2. The CL-CD2400 has detected status changes that need to be relayed to the host processor in a relatively short time. An example of this type of interrupt is when the CL-CD2400 detects a bus error while its performing a DMA operation.

Interrupts in CL-CD2400 are grouped into three categories:

Group 1 – Modem/Timer signal change interrupt
Group 2 – Transmit interrupt
Group 3 – Receive interrupt

Each interrupt group can be enabled or disabled individually. The CL-CD2400 asserts interrupts to the host processor through the IREQ lines; the IREQ line conveys the source of the interrupt request from the Receive, Transmit or Modem Signal Change interrupt groups. In response to the interrupt request, the host asserts the interrupt acknowledge signal and drives the priority level on the address bus. The host conveys which interrupt source it is acknowledging by placing the appropriate priority level on the address bus. If the priority level on the address bus matches any of the PILR Registers, then the CL-CD2400 responds to the interrupt acknowledge cycle by driving the appropriate interrupt vector (on lower 2 bits of the data bus, the upper 6 bits are user-defined. Refer to LIVR Register). to signify the type of interrupt service being requested.

Interrupt vector	Interrupt service type
00	Receive Exception
01	Modem signal change
10	Transmit
11	Receive data

Thus, from this information, the host software driver will vector to the Modem, Transmit or Receive interrupt service routine to process the interrupt request. Note that the CL-CD2400 does not associate any hierarchy across interrupt groups or channels; it asserts interrupts to the host as the need arises. It is left to the host driver to prioritize the interrupt service routines.

7.5.5 Context Switching in the CL-CD2400

The CL-CD2400 can be a slave during the bus read/write operations, or a bus master for DMA transfers. The registers in the CL-CD2400 are grouped into global and per-channel (local). The global registers are available at all times, and the per-channel registers are accessed through the Channel Access Register (CAR). There are, of course, four groups of per-channel registers; the group that is active at any given time is determined by the value set in the CAR Register. The user is free to modify the value in CAR, at any time, to access that channel registers. During interrupt acknowledge, the channel number that is initiating the interrupt is set by the CL-CD2400 and is conveyed to the host through LICR or LIVR (if the host chooses to embed that information in the upper bits of this register). This channel number is determined by the current context. There are three different contexts in the CL-CD2400, one for each group of interrupts. (When it is not in an interrupt

context, then it is in the background context). The CL-CD2400 nests contexts as interrupts nest; therefore, when an interrupt is acknowledged, it pushes the current context onto an internal stack and enters the new context for the acknowledged interrupt. Any register accesses during the interrupt acknowledge will be from that channel, until the host performs the mandatory write to the EOIR Register. Upon receiving the end of interrupt signal, the CL-CD2400 pops the interrupt stack and restores the previous context. Also, note that some register accesses are meaningful only within the interrupt context. For example, the host processor should write to TDR only when acknowledging a transmit interrupt.

7.5.6 FIFOs in the CL-CD2400

Each channel in the CL-CD2400 has a 16-byte Receive FIFO and a 16-byte Transmit FIFO. The threshold level for both FIFOs is set through COR4 Register. The maximum allowable threshold value is 12 (0Ch).

Receive FIFO: In the Asynchronous Mode, a 'good data' interrupt is initiated when the number of characters in the FIFO is greater than the FIFO threshold. Note that Receive timeout and Receive data exception conditions also cause an interrupt to the host. In the Synchronous Mode, data transfer is initiated when the number of characters is greater than the FIFO threshold or an end of frame is reached.

Transmit FIFO: If TxD Bit in the IER Register is set, the CL-CD2400 will initiate a data transfer (to fill the FIFO) when the number of empty bytes in the FIFO is greater than the threshold set. During synchronous operation, when the last byte of a frame is transferred into the FIFO, the CL-CD2400 will stop writing data into the FIFO.

7.5.7 Pass/Keep Logic

The following three examples describe the action taken by the device during an interrupt acknowledge cycle.

1. The CL-CD2400 does not have an interrupt asserted, but there is a valid interrupt acknowledge cycle in progress, (i.e., IACKIN* and DS* are active) then this interrupt acknowledge is passed out on IACKOUT*.
2. The CL-CD2400 is asserting one or more of its interrupts, but the interrupt priority level driven on the address bus by the host does not match the contents of any PILR Registers; this interrupt acknowledge is passed out on IACKOUT*.
3. The CL-CD2400 is asserting an interrupt, and the interrupt priority level on the address bus matches the PILR Register for that interrupt type; this means the interrupt acknowledge is is not passed out on IACKOUT*.

7.5.8 Fair Share Scheme

When multiple CL-CD2400s are daisy-chained, the fair-share logic in these devices guarantees that interrupts from all CL-CD2400s in the system are presented to the host with equal urgency. There is no positional hierarchy in the interrupt scheme, i.e., the CL-CD2400 that is farthest from the host processor has an equal chance of getting its interrupts through as the CL-CD2400 that is nearest to the top of the interrupt daisy chain. The IREQ*[3-1] interrupt request lines are open-drain outputs, therefore, in a multichip design, each interrupt-type output can be wire-ORed together, creating a Fair Share scheme for each group of interrupts.

When an interrupt request line is asserted, the fair bit for that type of interrupt on the asserting device is reset. The fair bit will remain reset until the interrupt line returns to a logic 1. The CL-CD2400 cannot assert a new interrupt while the corresponding fair bit is reset Therefore, when multiple CL-CD2400s assert an interrupt together, each one will be serviced, in turn, before they can reassert the same interrupt type.

The fair-share scheme is totally transparent to the user, and no enabling or disabling is needed.

8. SAMPLE PACKAGES

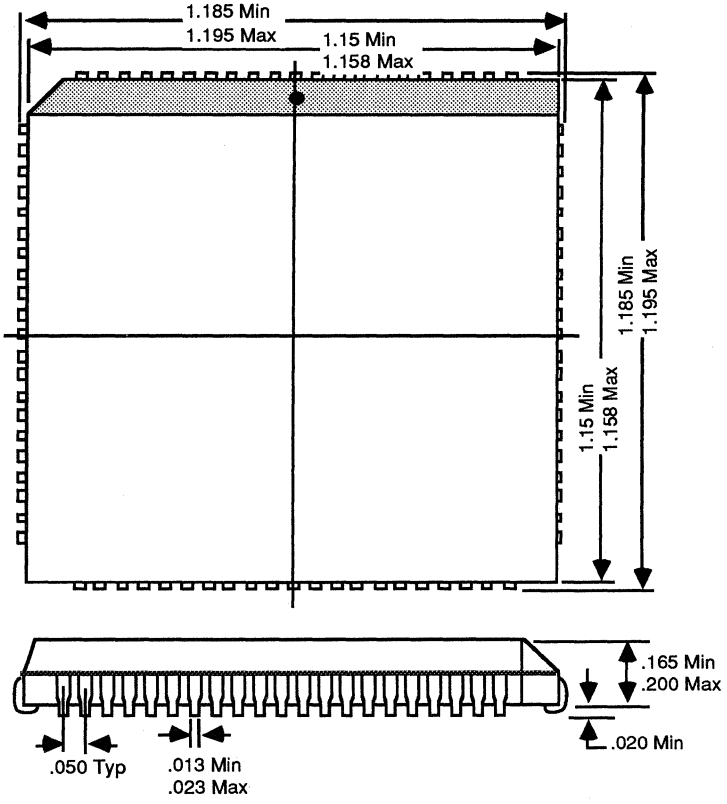


Figure 8–0. CL-CD2400 (84-pin PLCC) Sample Package

NOTE: Dimensions for the PLCC package are in millimeters.

8. SAMPLE PACKAGES *(cont.)*

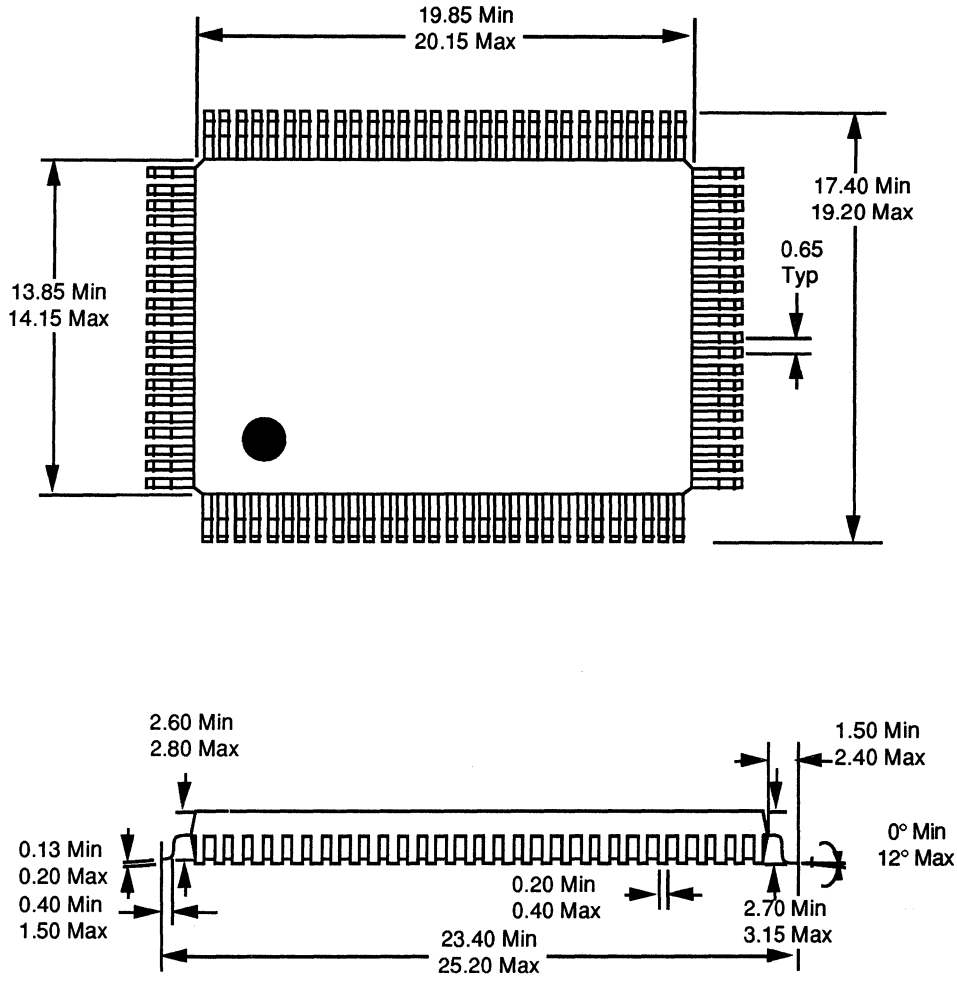


Figure 8–1. CL-CD2401 (100-pin QFP) Sample Package

NOTE: Dimensions for the QFP package are in millimeters.

9. ELECTRICAL SPECIFICATIONS

9.1 Absolute Maximum Ratings

Operating ambient temperature ..... 0° C to 70° C

Storage temperature ..... -65° C to 150° C

All voltages with respect to ground ..... -0.5 to Vcc +0.5 Volts

Supply voltage (Vcc) ..... +7.0 Volts

Power dissipation ..... 0.5 Watt

NOTE: Stresses above those listed under *Absolute Maximum Ratings* may cause permanent damage to the device. These are stress ratings only and functional operation of the device at these or any conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

9.2 D.C. Electrical Characteristics

(@ V_{CC} = 5V ± 5%, T_A = 0° C to 70° C)

Symbol	Parameter	MIN	MAX	Units	Test Conditions
V_{IL}	Input Low Voltage	-0.5	0.8	V	
V_{IH}	Input High Voltage (ALL PINS*)	2.0	V_{CC}	V	
V_{OL}	Output Low Voltage	0.4		V	I_{OL} = 2.4 mA
V_{OH}	Output High Voltage	2.4		V	I_{OH} = -400 µA
I_{IL}	Input Leakage Current	-10	10	µA	0 < V_{in} < V_{CC}
I_{LL}	Data Bus 3-state Leakage current	-10	10	µA	0 < V_{out} < V_{CC}
I_{OC}	Open Drain Output Leakage	-10	10	µA	0 < V_{out} < V_{CC}
I_{CC}	Power Supply Current		50	mA	CLK = 20 MHz
C_{in}	Input Capacitance		10	pF	
C_{out}	Output Capacitance		10	pF	
* Except:					
V_{IH}	Input High Voltage for CLK*, RESET* and BGIN*	2.7	V_{CC}	V	

9.3 A.C. Electrical Characteristics

CL-CD2400 Timing Parameters

Parameter	MIN	MAX
t_1 CLK high to BUSCLK high		35
t_2 CLK high to BUSCLK low		35
Bus Arbitration		
t_{11} CLK high to BGACK* tristate		30
t_{12} BGIN* low to address valid		60
t_{13} Address hold after CLK high	15	
t_{14} CLK high to address tristate		35
t_{15} CLK high to ADLD* low		35
t_{16} CLK high to ADLD* high		35
t_{17} Address setup to ADLD* high	15	
DMA Read		
t_{21} Data setup to CLK high	20	
t_{22} Data hold after CLK high	30	
t_{23} CLK high to address valid		60
t_{24} CLK low to AS* low		35
t_{25} CLK high to AS* high		35
t_{26} CLK low to DS* low		35
t_{27} CLK high to DS* high		35
t_{28} DTACK* low setup to CLK high	10	
t_{29} DTACK* high setup to CLK high (to avoid false termination)	50	
DMA Write		
t_{31} CLK high to data valid		70
t_{32} Data hold after CLK high	5	
t_{33} CLK low to DS* low		35
t_{34} CLK high to DS* high		35
t_{35} DTACK* low setup to CLK high	10	
t_{36} DTACK* high setup to CLK high (to avoid false termination)	50	
t_{37} CLK high to DATDIR* low		40

Parameter	MIN	MAX
Host Read/Write		
t_{41} DS* and CS* low setup to CLK high	5	
t_{42} Reserved		
t_{43} Reserved		
t_{44} R/W* setup to CLK high	0	
t_{45} CLK high to data valid		70
t_{46} Data setup time to CLK high	0	
t_{47} Data hold time after CLK high	25	
t_{48} Address setup time to CLK high	5	
t_{49} Address hold time after CLK high	30	
t_{50} CLK high to DTACK* low (read cycle)		35
t_{51} CLK high to DTACK* low (write cycle)		35
t_{52} CS* and DS* low and BUSCLK high to DATEN*/DATDIR* low		40
t_{52} (CS* and DS*) low to DEN* low		40
t_{53} (CS* or DS*) high to DEN* high		40
t_{54} (CS* and DS*) low to DATDIR* low		40
Interrupt Acknowledge		
t_{61} CLK high to IACKIN* setup		50
t_{62} CLK high to DS* setup		50
t_{63} CLK high to Data valid		70
t_{64} Address setup to IACKIN* low	10	
t_{65} Address hold after IACKIN* high	0	
t_{66} CLK high to DTACK* low		40
t_{67} (IACKIN* and DS*) low and BUSCLK high to DATEN* and DATDIR* low		70

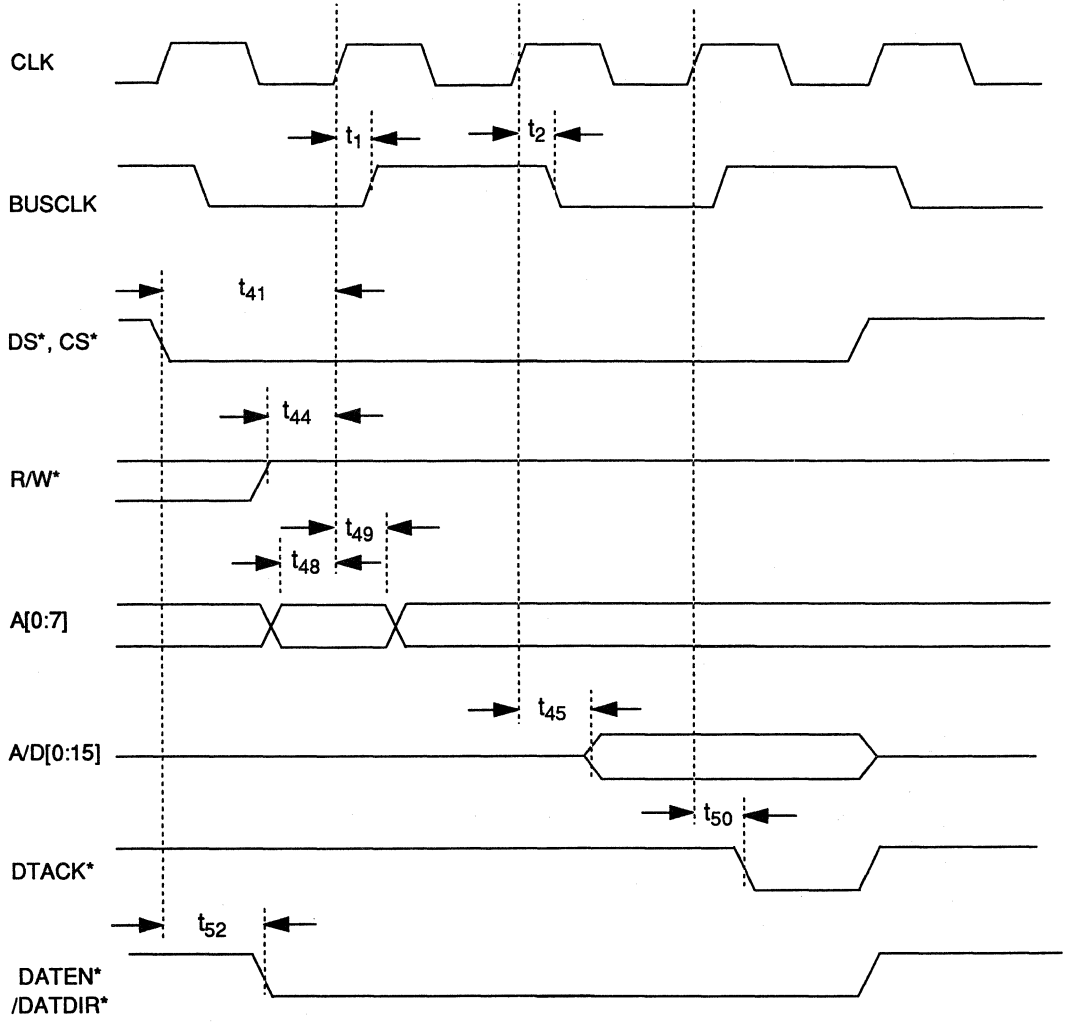


Figure 9–0. Slave Read Cycle Timing

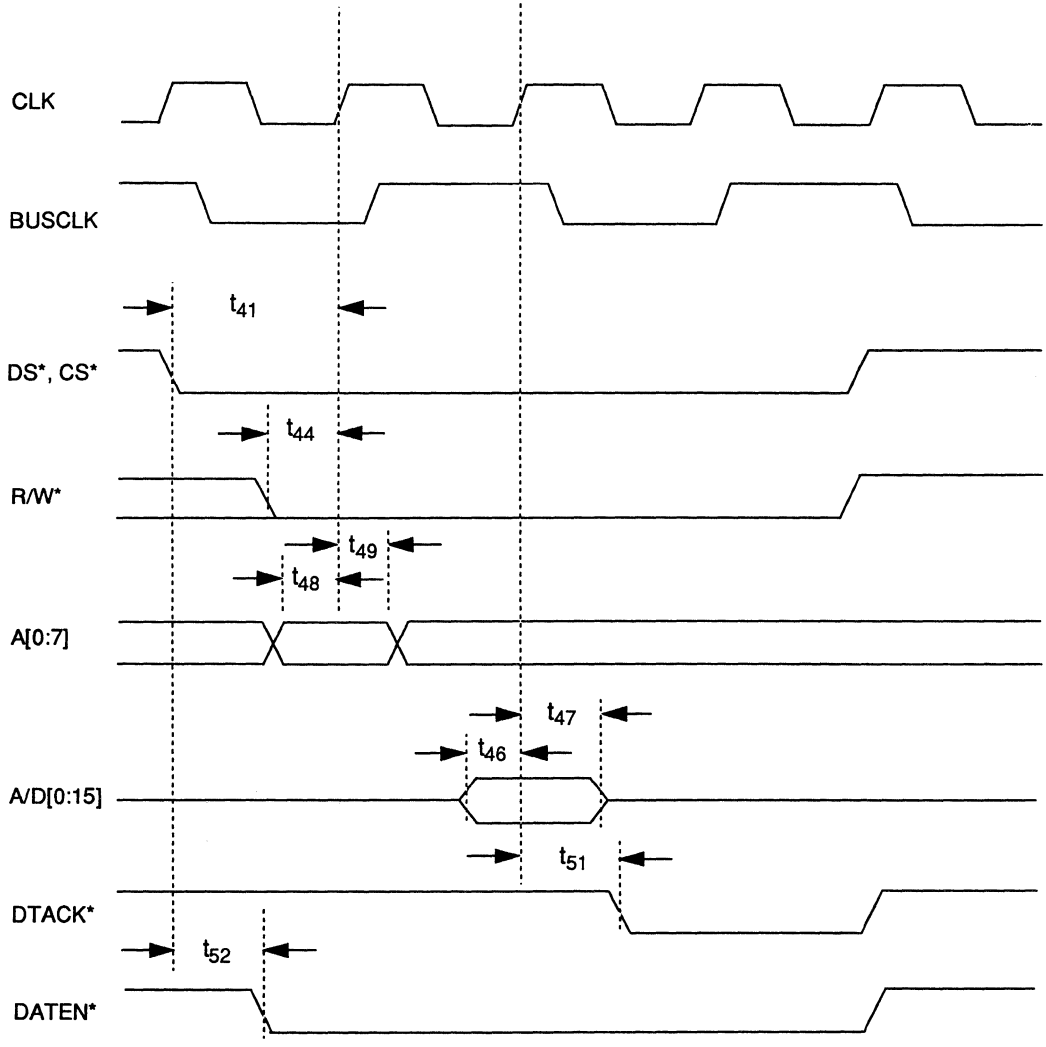


Figure 9–1. Slave Write Cycle Timing

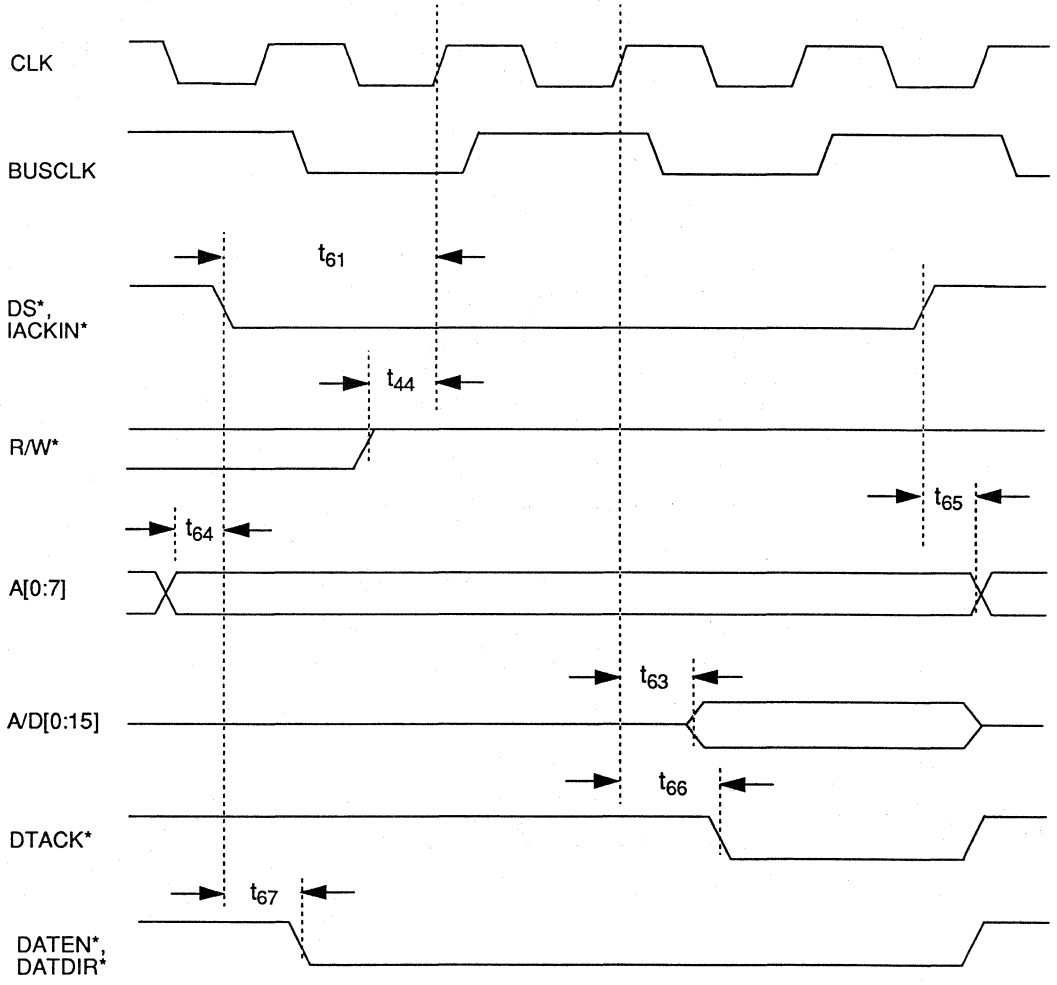


Figure 9–2. Interrupt Acknowledge Cycle Timing

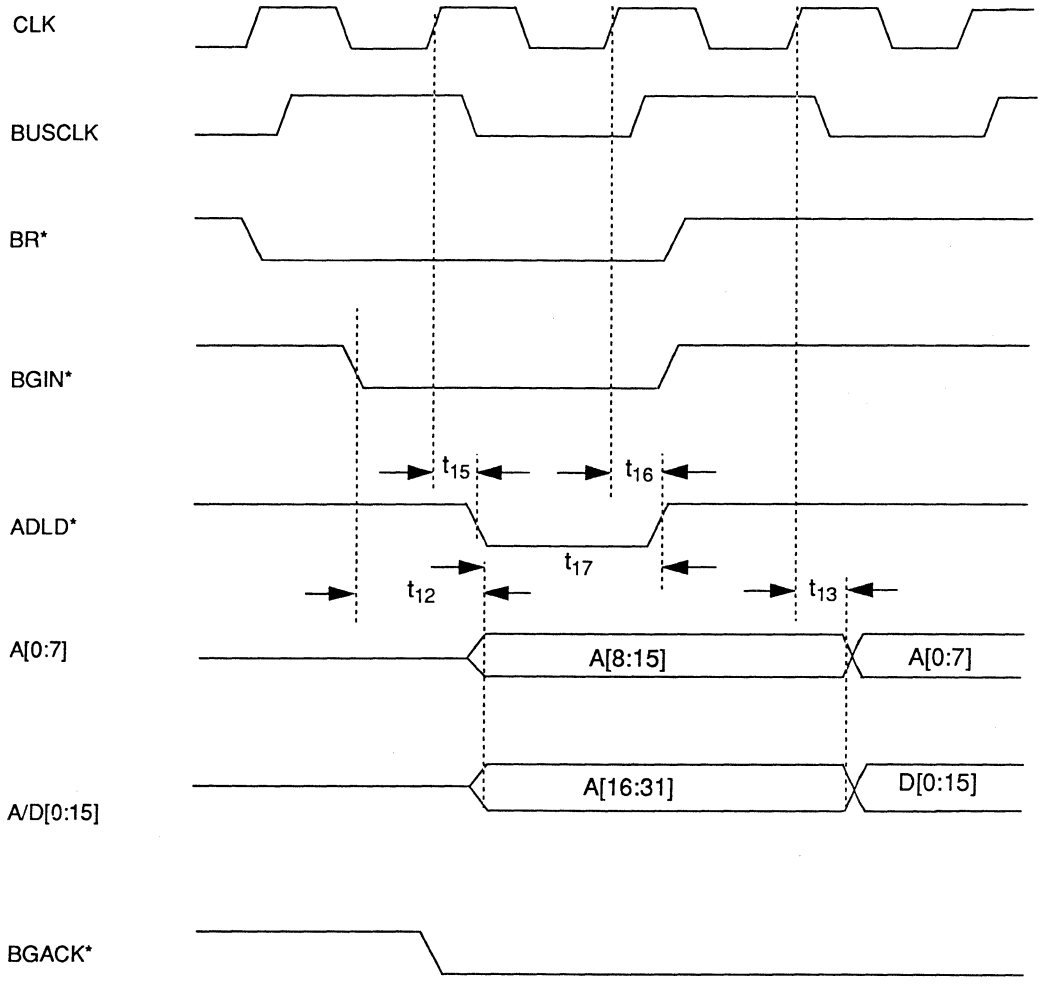


Figure 9–3. BUS Arbitration Cycle Timing

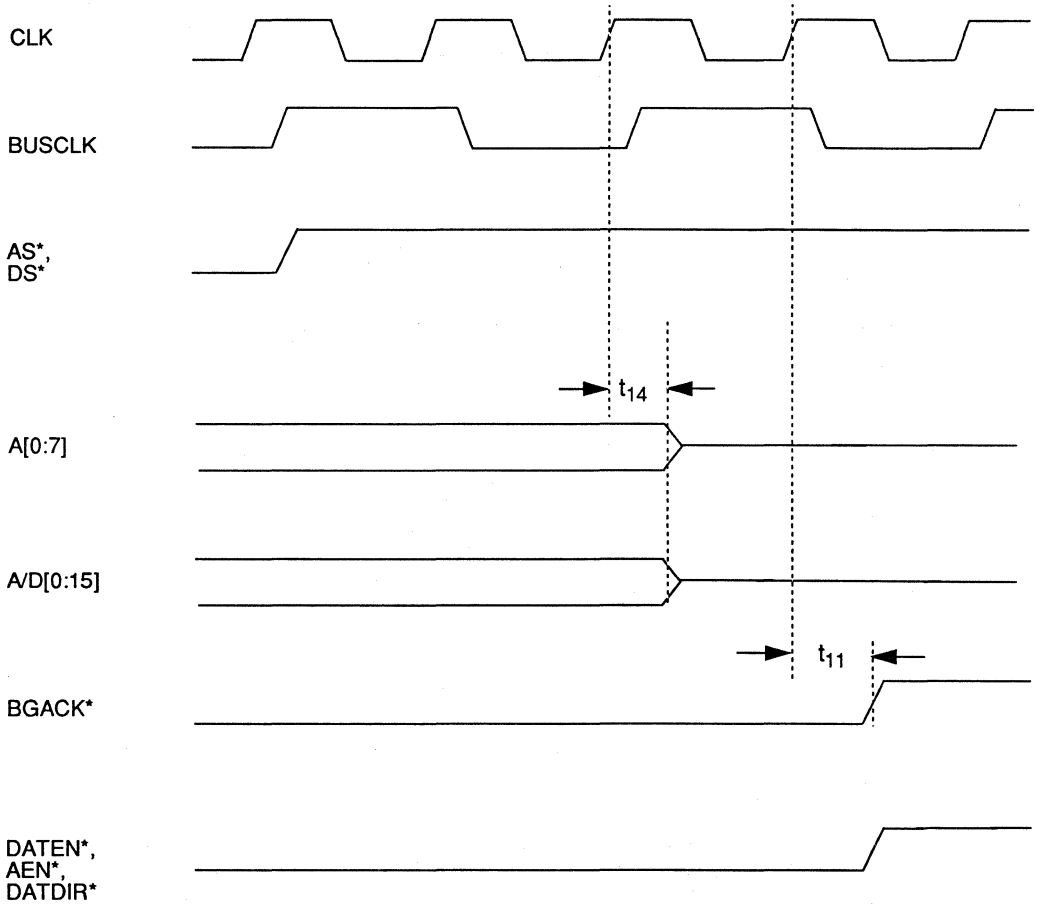


Figure 9–4. BUS Release Timing

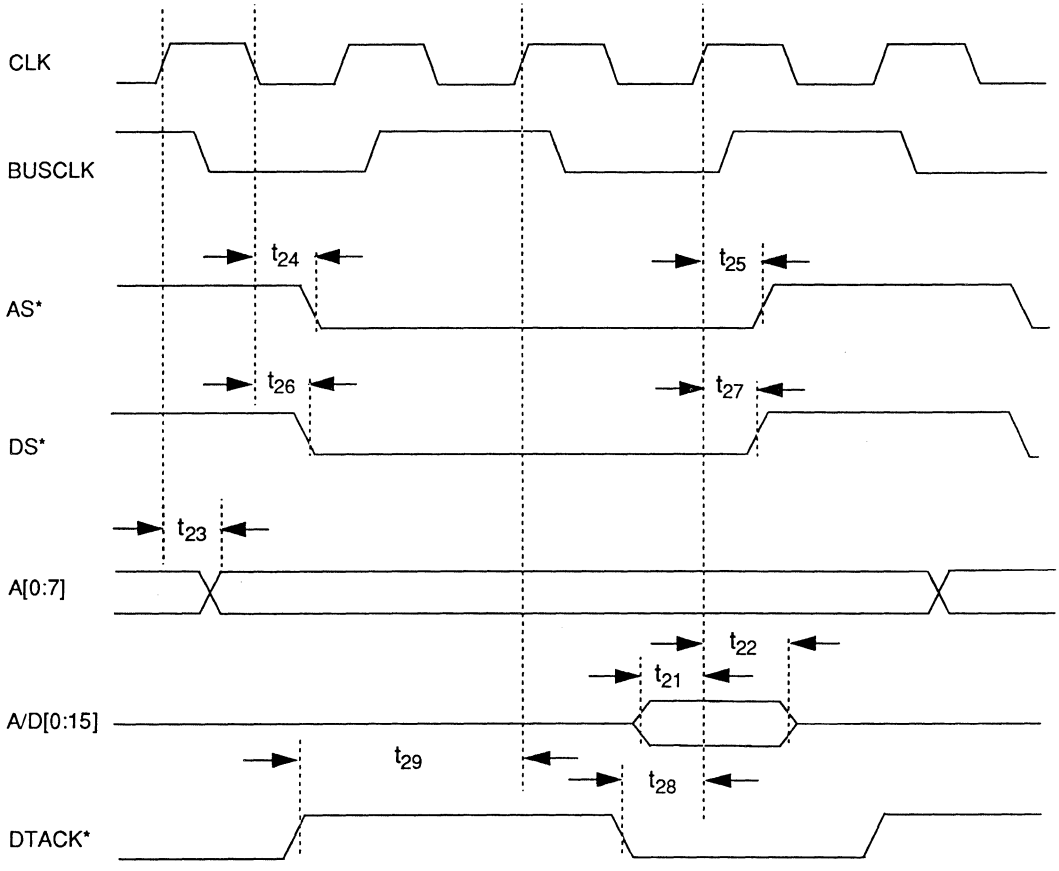


Figure 9–5. DMA Read Cycle Timing

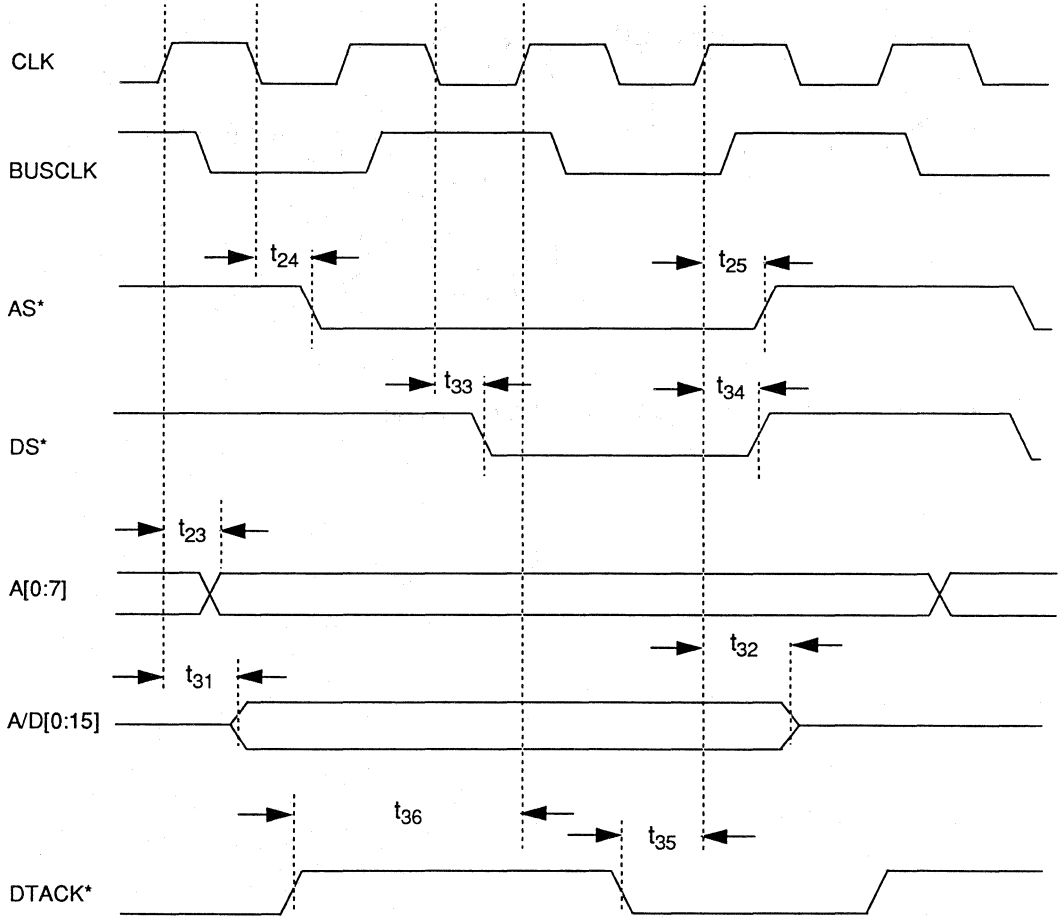
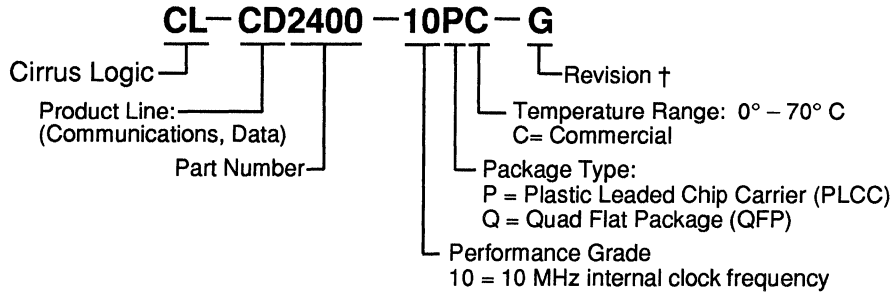


Figure 9–6. DMA Write Cycle Timing

10. ORDERING INFORMATION



† Contact Cirrus Logic, Inc., for up-to-date information on revisions.

Notes

Notes

Notes

Notes

Notes

Notes

Direct Sales Offices

Domestic

N. CALIFORNIA
San Jose
TEL: 408/436-7110
FAX: 408/437-8960

S. CALIFORNIA
Tustin
TEL: 714/258-8303
FAX: 714/258-8307

Thousand Oaks
TEL: 805/371-5381
FAX: 805/371-5382

ROCKY MOUNTAIN AREA
Boulder, CO
TEL: 303/939-9739
FAX: 303/442-6388

NORTH CENTRAL AREA
Westchester, IL
TEL: 708/449-7715
FAX: 708/449-7804

SOUTH CENTRAL AREA
Austin, TX
TEL: 512/794-8490
FAX: 512/794-8069

NORTHEASTERN AREA
Andover, MA
TEL: 508/474-9300
FAX: 508/474-9149

Philadelphia, PA
TEL: 215/251-6881
FAX: 215/651-0147

SOUTH EASTERN AREA
Boca Raton, FL
TEL: 407/994-9883
FAX: 407/994-9887

Atlanta, GA
TEL: 404/263-7601
FAX: 404/729-6942

International

GERMANY
Herrsching
TEL: 49/08152-2030
FAX: 49/08152-6211

JAPAN
Kanagawa-Ken
TEL: 81/462-76-0601
FAX: 81/462-76-0291

SINGAPORE
TEL: 65/3532122
FAX: 65/3532166

TAIWAN
Taipei
TEL: 886/2-718-4533
FAX: 886/2-718-4526

UNITED KINGDOM
Berkshire, England
TEL: 44/344-780-782
FAX: 44/344-761-429

The Company

Cirrus Logic, Inc., produces high-integration peripheral controller circuits for mass storage, graphics, and data communications. Our products are used in leading-edge personal computers, engineering workstations, and office automation equipment.

The Cirrus Logic formula combines proprietary S/LA™† IC design automation with system design expertise. The S/LA design system is a proven tool for developing high-performance logic circuits in half the time of most semiconductor companies. The results are better VLSI products, on-time, that help you win in the marketplace.

Cirrus Logic's fabless manufacturing strategy, unique in the semiconductor industry, employs a full manufacturing infrastructure to ensure maximum product quality, availability and value for our customers.

Talk to our systems and applications specialists; see how you can benefit from a new kind of semiconductor company.

† U.S. Patent No. 4,293,783