intel®

CD2401

Multi-Protocol Communications Controller

Datasheet

The CD2401 is a four-channel synchronous/asynchronous communications controller, specifically designed to reduce host-system processing overhead and increase efficiency in a wide variety of communications applications. The CD2401 is available in a 100-pin MQFP package that offers eight clock/modem pins per channel. The device has four fully-independent serial channels that support asynchronous, bit-synchronous (HDLC/SDLC), bisync (byte-synchronous), and X.21 protocols.

The CD2401 is based on a proprietary, on-chip RISC processor that performs all the time-critical, low-level tasks that are otherwise normally performed by the host system.

The CD2401 boosts system efficiency with on-chip DMA, on-chip FIFOs, intelligent vectored interrupts, and intelligent protocol processing. The on-chip DMA controller provides 'fire-and-forget' transmit support — the host need only inform the CD2401 of the location of the packet to be sent. Similarly, on receive, the CD2401 automatically receives a complete packet with no host intervention or assistance. The DMA controller also has a transmit Append mode for use in asynchronous applications.

The DMA controller uses a dual-buffer scheme that easily implements simple or complex buffer schemes. Each channel and direction in the dual-buffer scheme has two active buffers.

The CD2401 can be programmed to interrupt the host at the completion of a frame or buffer. In applications where buffers are of a small, fixed size, the dual-buffer scheme allows large frames to be divided into multiple buffers.

For applications where a DMA interface is not desired, the devices can be operated as interrupt-driven or polled. This choice is available for each channel and each direction. For example, a channel can be programmed for DMA transmit and interrupt-driven receive.

In either case, 16-byte FIFOs on each channel and in each direction reduce latency time requirements, making both software and hardware designs less time-critical. Threshold levels on the FIFOs are user-programmable.

Efficient vectored interrupts are another way the CD2401 attains system efficiency. Separate interrupts are generated for transmit, receive, and modem-signal change with unique user-defined vectors for each type and channel. This allows flexible interfacing and fast, efficient interrupt coding. For example, the Good Data™ interrupt allows the host to vector directly to a routine that transfers the data — no status or error checking is required.

Datasheet

Contents

Figures

Tables

Revision History

Revision	Date	Description
1.0	5/01	Initial release.

Datasheet

intel.

1.0   Features

- Four full-duplex multi-protocol channels, each running up to 134.4 kbits/sec. (@ CLK = 35 MHz)

- Supports async, HDLC/SDLC (synchronous data link control; non-multidrop applications), bisync and X.21 on all channels

- 32-bit address, 16-bit data, double-buffered DMA controller for each transmitter and receiver; two independent bit-rate generators per channel for transmit and receive

- On-chip NRZ (nonreturn-to-zero), NRZI (nonreturn-to-zero inverted), and Manchester data encoding and decoding

- DPLL (digital phase-locked loop) on each receiver

- Two independent timers per channel

HDLC/SDLC (Non-Multidrop) Features

- Four 8-bit or two 16-bit frame address matching

- FCS generation and validation

- CRC (cyclic redundancy check) optionally readable

- Programmable leading-pad character transmission

- Supports shared flags on receive frames

- Programmable number of leading flags

Asynchronous Features

- User-programmable and automatic flow control modes

  — In-band (software) by XON/XOFF

  — Out-of-band (hardware flow control) by RTS/CTS and DTR/DSR

  — Line-break detection and generation

  — Special-character and character-range recognition and transmission

  — Transmit delay

- 5- to 8-bit character plus optional parity

- Enhanced features for UNIX® environment

  — Character expansion in transmit (for example, sending <LF> will be expanded to <CR> <LF> automatically)

  — Programmable translation of receiving character with error to different pattern (for example, character with parity error can be translated into FFh, 00h character on the system side)

  — Flow-control transparency and LNext

- Programmable timer closely coupled with character reception, especially for asynchronous receive DMA operation

Bisync Features

- Programmable for ASCII or EBCDIC encoding

- Support for transparent Bisync

- Recognition of all special characters enabling:

    — Block separation

    — CRC generation and validation

- Chaining of long receive blocks into multiple buffers

X.21 Features

- Full support for X.21 protocol

- Detection of steady-state conditions

- Transmission of steady-state conditions synchronized to modem lead

- Programmable SYN character

- 1 or 2 SYN character detect option

- Idle in any line condition

- Optional SYN strip on receive

On-Chip DMA Controller Features

- DMA Master or interrupt-selectable per channel and per direction

- Dual-configuration register sets to reduce realtime constraints

- Append and Block mode DMA

- Chain/unchain of long frames into multiple buffers

- 32-bit address and 8- or 16-bit data transfer

- Programmable gap in buffers following a receive character exception

Other Features

- Improved interrupt schemes

    — Vectored interrupts per channel allow direct jump into proper service routines

    — Good Data™ interrupts eliminate need for status checks

- Easily cascaded for multiple-device configurations

- 16-byte receive and transmit FIFOs

- Local and remote maintenance loopback modes

- Byte-endian-orientation selection pin that allows easy interface to 80X86 and 680X0 processors

- Eight clock/modem control signals per channel
  (in addition to TxD and RxD)

1.1   Benefits

- Substantially reduced host CPU overhead resulting in more channels and faster overall throughput

- No time-critical host software enabling faster and easier software development

- Smallest possible footprint for multi-channel device

Figure 1. Functional Block Diagram
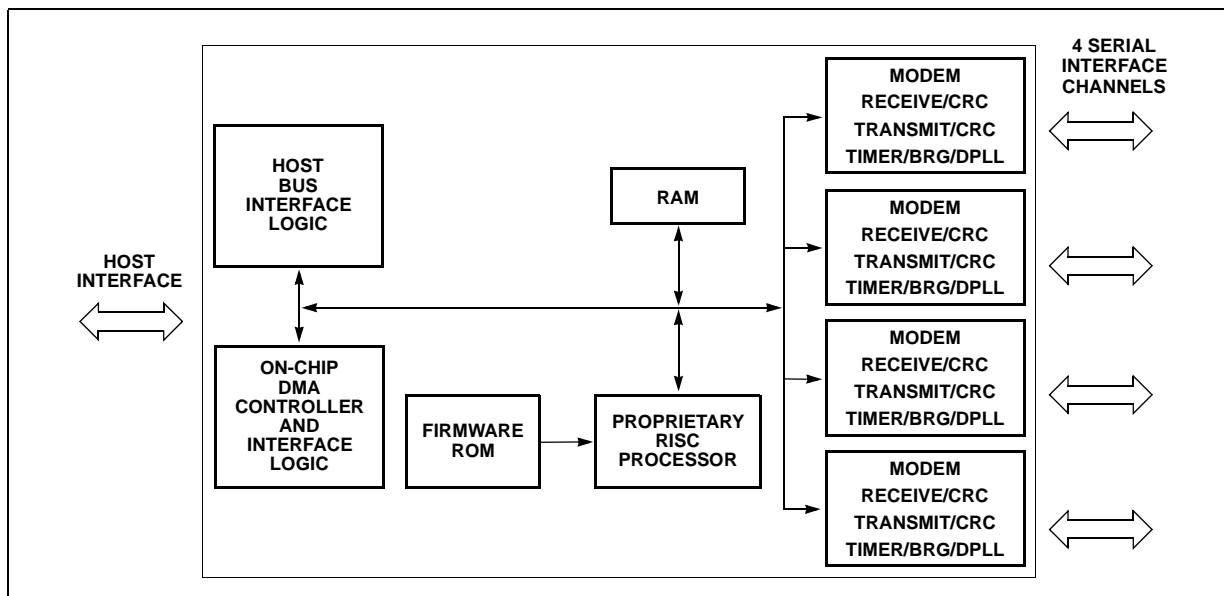
1.2   CD2XXX Family Compatibility

Features	CD2231	CD2401	CD2431	CD2481
Number of serial channels	2	4	4	4
Interrupt and on-chip DMA mechanism	√[1]	√	√	√
FIFO depth (per channel and per direction)	16 bytes	16 bytes	16 bytes	16 bytes
Data size (bits)	√	√	√	√
Async	√	√	√	√
SDLC/HDLC	√	√	√	√
X.21, bisync	–	√	–	–
Async-HDLC, PPP	√	–	√	√
SLIP	•[2]	–	•	•
MNP® 4	•	–	•	•
Serial data rate: sync / async (kbits/sec.)	256 / 230.4[3,4]	128 / 134.4[c]	128 / 134.4[c]	128 / 134.4[c]

Features	CD2231	CD2401	CD2431	CD2481
Number of modem leads (per channel, including RxD and TxD)	10	10	10	10
On-chip timers	√	√	√	√
UNIX® character processing[5]	√	√	√	√
Automatic in-band Rx flow control	•	–	•	•
Special character Tx and recognition	√	√	√	√
Package	100-pin MQFP	100-pin MQFP	100-pin MQFP	100-pin MQFP
System interface	√	√	√	√
Pin compatibility	**CD2401**/'2431	CD2431/'2231	**CD2401**/'2231	**CD2401**/'2231

NOTES:
1. √ indicates identical operation and register setting.
2. • indicates available in production revision (Revision B) and later.
3. A clock frequency of 35 MHz is required to obtain maximum bit-rates.
4. 134.4 kbps / 230.4 kbps in all async modes, 128 kbps / 256 kbps in sync modes: applies to Revision M or later CD2401; Revision D and later CD2431; Revision D or later CD2231.
5. UNIX character processing is available in Async mode only.

2.0 Conventions

Abbreviations

Symbol	Units of measure
°C	degree Celsius
μF	microfarad
μs	microsecond (1,000 nanoseconds)
Hz	hertz (cycle per second)
Kbit	kilobit (1,024 bits)
kbits/sec., kbps	kilobit (1,000 bits) per second
Kbyte	kilobyte (1,024 bytes)
kbytes/sec.	kilobyte (1,000 bytes) per second
kHz	kilohertz
kΩ	kilohm
Mbyte	megabyte (1,048,576 bytes)
MHz	megahertz (1,000 kilohertz)
mA	milliampere
ms	millisecond (1,000 microseconds)
ns	nanosecond
pV	picovolt
V	volt
W	watt

The use of 'tbd' indicates values that are 'to be determined', 'n/a' designates 'not available', and 'n/c' indicates a pin that is a 'no connect'.

Acronyms

Acronym	Definition
AC	alternating current
BCC	block check character
BRG	bit rate generator
bisync	byte synchronous
CMOS	complementary metal-oxide semiconductor
CRC[1]	cyclic redundancy check
DC	direct current
DCE	data communication equipment
DMA	direct-memory access
DPLL	digital phase-locked loop
DRAM	dynamic random-access memory
DTE	data terminal equipment
EOF	end-of-frame
ETC	embedded transmit command
FCS[1]	frame check sequence
FCT	flow control transparency
FIFO	first in/first out
HDLC	high-level data link control
ISA	industry standard architecture
ITB	intermediate text block
LRC	longitudinal redundancy check character
LSB	least-significant bit
MSB	most-significant bit
NRZ	nonreturn to zero
NRZI	nonreturn to zero inverted
PPP	point-to-point protocol
MQFP	metric quad-flat pack
RAM	random-access memory
R/W	read/write
SDLC	synchronous data link control
TTL	transistor-transistor logic

1. The terms CRC and FCS are used interchangeably in this document.

intel®

3.0 Pin Information

3.1 Pin Diagram

CD2401
100-Pin MQFP

Top pins (left to right, 50 to 31):
TXCIN[2] (50), RXCIN[2] (49), TXCIN[1] (48), RXCOUT[3] (47), RXCIN[1] (46), VDD (45), TXCIN[0] (44), RXCIN[0] (43), TXD[3] (42), TXD[2] (41), TXD[1] (40), TXD[0] (39), RXD[3] (38), RXCOUT[2] (37), RXD[2] (36), RXD[1] (35), RXD[0] (34), TEST (33), RXCOUT[1] (32), BYTESWAP (31)

Left pins (51 to 80):
RXCIN[3] (51)
TXCIN[3] (52)
DSR*[0] (53)
CTS*[0] (54)
TXCOUT/DTR*[0] (55)
RTS*[0] (56)
DSR*[1] (57)
CTS*[1] (58)
TXCOUT/DTR*[1] (59)
RTS*[1] (60)
DSR*[2] (61)
GND (62)
CTS*[2] (63)
TXCOUT/DTR*[2] (64)
RTS*[2] (65)
DSR*[3] (66)
CTS*[3] (67)
TXCOUT/DTR*[3] (68)
RTS*[3] (69)
GND (70)
A[7] (71)
A[6] (72)
A[5] (73)
A[4] (74)
A[3] (75)
A[2] (76)
A[1] (77)
A[0] (78)
VDD (79)
A/D[15] (80)

Right pins (30 to 1):
AEN* (30)
ADLD* (29)
DATEN* (28)
DATDIR* (27)
RESET* (26)
RXCOUT[0] (25)
IREQ*[3] (24)
IREQ*[2] (23)
CD*[3] (22)
IREQ*[1] (21)
GND (20)
IACKOUT* (19)
CD*[2] (18)
IACKIN* (17)
DTACK* (16)
DS* (15)
AS* (14)
R/W* (13)
BGACK* (12)
CD*[1] (11)
BGOUT* (10)
BGIN* (9)
VDD (8)
BR* (7)
BUSCLK (6)
CLK (5)
SIZ[1] (4)
SIZ[0] (3)
GND (2)
CS* (1)

Bottom pins (81 to 100):
A/D[14] (81), GND (82), A/D[13] (83), A/D[12] (84), CD*[0] (85), A/D[11] (86), A/D[10] (87), A/D[9] (88), A/D[8] (89), A/D[7] (90), A/D[6] (91), A/D[5] (92), A/D[4] (93), A/D[3] (94), A/D[2] (95), VDD (96), A/D[1] (97), A/D[0] (98), GND (99), BERR* (100)

3.2   Pin Functions

3.3   Pin Descriptions

The following conventions are used in the pin-description tables:

- (*) after a pin name indicates that the signal is active-low
- 'I' indicates the pin is input only
- 'O' indicates the pin is output only
- 'I/O' indicates the pin is bidirectional
- 'OD' indicates open-drain
- 'TS' indicates tristate
- '–' indicates ascending pin numbers
- ':' indicates descending pin numbers

intel

Table 1. Pin Descriptions (Sheet 1 of 4)

Symbol	Pin Number	Type	Description
CS*	1	I	**CHIP SELECT*:** When low, the CD2401 registers can be read or written by the host processor.
AS*	14	I/O (TS)	**ADDRESS STROBE*:** When the CD2401 is a bus master, this pin is an output which indicates that R/W*, A[7:0], and the externally latched A[31:8] are valid.
DS*	15	I/O (TS)	**DATA STROBE*:** When the CD2401 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 CD2401 is a bus master, DS* is an output used to control data transfer to and from system memory.
R/W*	13	I/O (TS)	**READ/WRITE*:** When the CD2401 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 CD2401 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*	16	I/O (OD)	**DATA TRANSFER ACKNOWLEDGE*:** When the CD2401 is not a bus master, this is an output and indicates to the host when a read or write to the CD2401 is complete. When BR* is driven low by the CD2401, DTACK* is an input which indicates that the system bus is no longer in use. When the CD2401 is a bus master, DTACK* is an input which indicates when system memory read and write cycles are complete.
SIZ[0–1]	3, 4	I/O (TS)	**SIZE [0–1]:** When not the active bus master, these are inputs that determine the size of the operand being read or written by the host. _see first SIZE table below_ When the CD2401 is a bus master, this is an output determining the size of the operand being transferred to or from system memory. _see second SIZE table below_
IACKIN*	17	I	**INTERRUPT ACKNOWLEDGE IN*:** This input, qualified with DS* and A[0–6], acknowledges CD2401 interrupts.

Within the SIZ[0–1] description, first table:

Bit		Size
SIZ1	SIZ0	
0	0	32 Bit[1]
0	1	Byte[2]
1	0	16 bit
1	1	3 bytes[1]

1. The CD2401 drives DTACK* even though the device does not respond to such byte alignment.
2. See BYTESWAP description.

Second table:

Bit		Size
SIZ1	SIZ0	
0	1	Byte[1]
1	0	16 bit

1. See BYTESWAP description.

Table 1. Pin Descriptions (Sheet 2 of 4)

Symbol	Pin Number	Type	Description
IACKOUT*	19	O	**INTERRUPT ACKNOWLEDGE OUT*:** This output is driven low during interrupt acknowledge cycles for which no internal interrupt is valid.
IREQ*[1–3]	21, 23, 24	I/O (OD)	**INTERRUPT REQUEST* [1–3]:** These outputs signal that the CD2401 has a valid interrupt for modem-lead activity (IREQ*1), transmit activity (IREQ*2), or receive activity (IREQ*3).
BR*	7	OD	**BUS REQUEST*:** This output is used to signal to the (open-drain) host processor or bus arbiter that bus mastership is required by the CD2401.
BGIN*	9	I	**BUS GRANT IN*:** This input indicates that the bus is available after the current bus master relinquishes the bus.
BGOUT*	10	O	**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*	12	I/O (OD)	**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*	100	I	**BUS ERROR*:** If this input becomes active while the CD2401 is a bus master, the current bus cycle is terminated, the bus relinquished, and an interrupt generated to indicate the error to the host processor.
A[0–7]	71–78	I/O (TS)	**ADDRESS [0–7]:** When the CD2401 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 CD2401 is a bus master, A[0–7] output the least-significant byte of the transfer address.
A/D[15:0]	80, 81, 83, 84, 86–95, 97, 98	I/O (TS)	**ADDRESS/DATA [15:0]:** When the CD2401 is not a bus master, these pins provide the 16-bit data bus for reading and writing to the CD2401 registers. When ADLD* is low, A/D[0–15] provide the upper address bits for external latching. When the CD2401 is a bus master, A/D[0–15] provide a multiplexed address/data bus for reading and writing to system memory.
ADLD*	29	O (TS)	**ADDRESS LOAD*:** This is a strobe used to externally latch the upper portion of the system address bus A[31:8]. While ADLD* is low, address bits 31:16 are available on A/D[15:0], and address bits 8 through 15 on A[7:0].
AEN*	30	O (TS)	**ADDRESS ENABLE*:** This output is used to output enable the external address bus drivers during CD2401 DMA cycles.
DATEN*	28	O (TS)	**DATA ENABLE*:** This output is active when either the CD2401 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.
DATDIR*	27	O (TS)	**DATA DIRECTION*:** This output is active when either the CD2401 is a bus master or the CS* pin is low. DATDIR* is used to control the external data buffers; when low, the buffers should be enabled in the CD2401 to system bus direction.
CLK	5	I	**CLOCK:** System clock.
BUSCLK	6	O	**BUS CLOCK:** This is the system clock (divided by 2) used internally to control certain bus operations. This pin is driven low during hardware reset.
RESET*	26	I	**RESET*:** This signal should stay valid for a minimum of 20 ns. The reset state of the CD2401 is guaranteed at the rising edge of this signal. When RESET* is removed, the CD2401 also performs a software initialization of its registers.

intel.

Table 1. Pin Descriptions (Sheet 3 of 4)

Symbol	Pin Number	Type	Description
TEST	33	I	**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 are in High-Impedance mode: BUSCLK, BGOUT*, IACKOUT*, RXCOUT[0–1], RTS*[1:0], DTR*[1:0], and TXD[1:0]. To ensure all CD2401 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 CD2401 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]	56, 60, 65, 69	O	**REQUEST TO SEND* [0–3]:** This output can be controlled automatically by the CD2401 to indicate that data is being sent on the TXD pin.
TXCOUT/DTR* [0–3]	55, 59, 64, 68	O	**TRANSMIT CLOCK OUT/DATA TERMINAL READY* [0–3]:** This output can be controlled automatically by the CD2401 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 is high and stays 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.
RXCOUT[0–3]	25, 32, 37, 47	O	**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 is 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]	54, 58, 63, 67	I	**CLEAR TO SEND* [0–3]:** This input can be programmed to control the flow of transmit data, for out-of-band flow control applications.
CD*[0–3]	85, 11, 18, 22	I	**CARRIER DETECT* [0–3]:** This pin is always visible in the MSVR. The CD* input can be programmed to validate receive data.
TXCIN[0–3]	44, 48, 50, 52	I	**TRANSMIT CLOCK [0–3]:** This pin inputs the transmit clock to the BRG.
RXCIN[0–3]	43, 46, 49, 51	I	**RECEIVE CLOCK [0–3]:** This pin inputs the receive clock to the BRG.
DSR*[0–3]	53, 57, 61, 66	I	**DATA SET READY* [0–3]:** This pin is always visible in the MSVR. The DSR input can be programmed to validate receive data.
TXD[0–3]	39, 40, 41, 42	O	**TRANSMIT DATA [0–3]:** Serial data output for each channel.
RXD[0–3]	34, 35, 36, 38	I	**RECEIVE DATA [0–3]:** Serial data input for each channel.

Table 1. Pin Descriptions (Sheet 4 of 4)

Symbol	Pin Number	Type	Description
BYTESWAP	31	I	**BYTESWAP:** 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 of A/D[7:0] precedes that of A/D[15:8] in a string of transmit or receive bytes; when BYTESWAP is low, A/D[15:8] precedes A/D[7:0]. When the BYTESWAP pin is high, bytes are transferred on A/D[7:0] when A[0] is low, and on A/D[15:8] when A[0] is high. When BYTESWAP is low, bytes are transferred on A/D[15:8] when A[0] is low, and A/D[7:0] when A[0] is high. A different register map is used, depending on the state of this pin. <table><tr><th>BYTESWAP</th><th>Byte Alignment</th></tr><tr><td>0</td><td>Motorola®</td></tr><tr><td>1</td><td>Intel®</td></tr></table>
V_{DD}	8, 45, 79, 96	–	**POWER**
GND	2, 20, 62, 70, 82, 99	–	**GROUND**

4.0   Register Table

The registers in the CD2401 are either Global or Per-Channel. The column 'Address mode' in the memory map in the following tables 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 CD2401 during interrupt service routines and DMA transfers.

In the following list, some register locations appear twice. They have different names and functions for asynchronous and synchronous protocol operations. See Chapter 8.0 on page 92 of this datasheet for detailed descriptions of all register functions.

4.1   Memory Map

The following notes are applicable for Section 4.1.1 through Section 4.1.7.

NOTES:

1. Address mode G: Global register — one set is always accessible.
   Address mode P: Per-Channel register — four sets, one per channel, accessible thgough CAR or interrupt context.

2. INT = address for Intel-style processor.

3. MOT = address for Motorola-style processor.

4.1.1   Global Registers

Name	Description	Addr. Mode[1]	INT[2]	MOT[3]	Size	Access	Page
GFRCR	Global Firmware Revision Code Register	G	82	81	B	R/W	92
CAR	Channel Access Register	G	EC	EE	B	R/W	92

4.1.2   Option Registers

Name	Description	Addr. Mode[1]	INT[2]	MOT[3]	Size	Access	Page
CMR	Channel Mode Register	P	18	1B	B	R/W	93
COR1	Channel Option Register 1	P	13	10	B	R/W	94
COR2	Channel Option Register 2	P	14	17	B	R/W	96
COR3	Channel Option Register 3	P	15	16	B	R/W	100
COR4	Channel Option Register 4	P	16	15	B	R/W	104
COR5	Channel Option Register 5	P	17	14	B	R/W	105
COR6	Channel Option Register 6	P	1B	18	B	R/W	106
COR7	Channel Option Register 7	P	04	07	B	R/W Async	108
SCHR1	Special Character Register 1	P	1C	1F	B	R/W Async	109
SCHR2	Special Character Register 2	P	1D	1E	B	R/W Async	110
SCHR3	Special Character Register 3	P	1E	1D	B	R/W Async	110
SCHR4	Special Character Register 4	P	1F	1C	B	R/W Async	110
SCRl	Special Character Range low	P	20	23	B	R/W Async	111
SCRh	Special Character Range high	P	21	22	B	R/W Async	111
LNXT	LNext Character	P	2D	2E	B	R/W Async	111
RFAR1	Receive Frame Address Register 1	P	1C	1F	B	R/W Sync	112
RFAR2	Receive Frame Address Register 2	P	1D	1E	B	R/W Sync	112
RFAR3	Receive Frame Address Register 3	P	1E	1D	B	R/W Sync	112
RFAR4	Receive Frame Address Register 4	P	1F	1C	B	R/W Sync	112
CPSR	CRC Polynomial Select Register	P	D4	D6	B	R/W Sync	113

4.1.3   Bit Rate and Clock Option Registers

Name	Description	Addr. Mode[1]	INT[2]	MOT[3]	Size	Access	Page
RBPR	Receive Bit Rate Period Register	P	C9	CB	B	R/W	113
RCOR	Receive Clock Option Register	P	CA	C8	B	R/W	114
TBPR	Transmit Bit Rate Period Register	P	C1	C3	B	R/W	115
TCOR	Transmit Clock Option Register	P	C2	C0	B	R/W	115

4.1.4   Channel Command and Status Registers

Name	Description	Addr. Mode[1]	INT[2]	MOT[3]	Size	Access	Page
CCR	Channel Command Register	P	10	13	B	R/W	116
STCR	Special Transmit Command Register	P	11	12	B	R/W	119
CSR	Channel Status Register	P	19	1A	B	R	119
MSVR-RTS	Modem Signal Value Registers	P	DC	DE	B	R/W	123
MSVR-DTR		P	DD	DF	B	R/W	123

4.1.5   Interrupt Registers

Name	Description	Addr. Mode[1]	INT[2]	MOT[3]	Size	Access	Page
LIVR	Local Interrupt Vector Register	P	0A	09	B	R/W	124
IER	Interrupt Enable Register	P	12	11	B	R/W	125
LICR	Local Interrupting Channel Register	P	25	26	B	R/W	126
STK	Stack Register	G	E0	E2	B	R	126

4.1.5.1   Receive Interrupt Registers

Name	Description	Addr. Mode[1]	INT[2]	MOT[3]	Size	Access	Page
RPILR	Receive Priority Interrupt Level Register	G	E3	E1	B	R/W	127
RIR	Receive Interrupt Register	G	EF	ED	B	R	128
RISR	Receive Interrupt Status Register	G	8A	88	W	R	129
RISRl	Receive Interrupt Status Register low	G	8A	89	B	R	129
RISRh	Receive Interrupt Status Register high	G	8B	88	B	R	132
RFOC	Receive FIFO Output Count	G	33	30	B	R	133
RDR	Receive Data Register	G	F8	F8	B	R	133
REOIR	Receive End Of Interrupt Register	G	87	84	B	W	134

4.1.5.2   Transmit Interrupt Registers

Name	Description	Addr. Mode[1]	INT[2]	MOT[3]	Size	Access	Page
TPILR	Transmit Priority Interrupt Level Register	G	E2	E0	B	R/W	135
TIR	Transmit Interrupt Register	G	EE	EC	B	R	135
TISR	Transmit Interrupt Status Register	G	89	8A	B	R	136
TFTC	Transmit FIFO Transfer Count	G	83	80	B	R	137
TDR	Transmit Data Register	G	F8	F8	B	W	137
TEOIR	Transmit End Of Interrupt Register	G	86	85	B	W	138

4.1.5.3   Modem Interrupt Registers

Name	Description	Addr. Mode[1]	INT[2]	MOT[3]	Size	Access	Page
MPILR	Modem Priority Interrupt Level Register	G	E1	E3	B	R/W	139
MIR	Modem Interrupt Register	G	ED	EF	B	R	139
MISR	Modem (/Timer) Interrupt Status Register	G	88	8B	B	R	140
MEOIR	Modem End Of Interrupt Register	G	85	86	B	W	141

4.1.6   DMA Registers

Name	Description	Addr. Mode[1]	INT[2]	MOT[3]	Size	Access	Page
DMR	DMA Mode Register (write only)	G	F4	F6	B	W	141
BERCNT	Bus Error Retry Count	G	8D	8E	B	R/W	142
DMABSTS	DMA Buffer Status	P	1A	19	B	R	142

4.1.6.1   DMA Receive Registers

Name	Description	Addr. Mode[1]	INT[2]	MOT[3]	Size	Access	Page
ARBADRL	A Receive Buffer Address Lower	P	40	42	W	R/W	143
ARBADRU	A Receive Buffer Address Upper	P	42	40	W	R/W	144
BRBADRL	B Receive Buffer Address Lower	P	44	46	W	R/W	144
BRBADRU	B Receive Buffer Address Upper	P	46	44	W	R/W	145
ARBCNT	A Receive Buffer Byte Count	P	48	4A	W	R/W	145
BRBCNT	B Receive Buffer Byte Count	P	4A	48	W	R/W	146
ARBSTS	A Receive Buffer Status	P	4C	4F	B	R/W	146
BRBSTS	B Receive Buffer Status	P	4D	4E	B	R/W	146
RCBADRL	Receive Current Buffer Address Lower	P	3C	3E	W	R	147
RCBADRU	Receive Current Buffer Address Upper	P	3E	3C	W	R	147

4.1.6.2   DMA Transmit Registers

Name	Description	Addr. Mode[1]	INT[2]	MOT[3]	Size	Access	Page
ATBADRL	A Transmit Buffer Address Lower	P	50	52	W	R/W	148
ATBADRU	A Transmit Buffer Address Upper	P	52	50	W	R/W	148
BTBADRL	B Transmit Buffer Address Lower	P	54	56	W	R/W	149
BTBADRU	B Transmit Buffer Address Upper	P	56	54	W	R/W	149
ATBCNT	A Transmit Buffer Byte Count	P	58	5A	W	R/W	150
BTBCNT	B Transmit Buffer Byte Count	P	5A	58	W	R/W	150
ATBSTS	A Transmit Buffer Status	P	5C	5F	B	R/W	150
BTBSTS	B Transmit Buffer Status	P	5D	5E	B	R/W	151
TCBADRL	Transmit Current Buffer Address Lower	P	38	3A	W	R	152
TCBADRU	Transmit Current Buffer Address Upper	P	3A	38	W	R	152

4.1.7 Timer Registers

Name	Description	Addr. Mode[1]	INT[2]	MOT[3]	Size	Access	Page
TPR	Timer Period Register	G	D8	DA	B	R/W	153
RTPR	Receive Timeout Period Register	P	26	24	W	R/W Async	153
RTPRl	Receive Timeout Period Register low	P	26	25	B	R/W Async	153
RTPRh	Receive Timeout Period Register high	P	27	24	B	R/W Async	154
GT1	General Timer 1	P	28	2A	W	R/W Sync	154
GT1l	General Timer 1 low	P	28	2B	B	R/W Sync	154
GT1h	General Timer 1 high	P	29	2A	B	R/W Sync	155
GT2	General Timer 2	P	2A	29	B	R/W Sync	155
TTR	Transmit Timer Register	P	2A	29	B	R/W Async	156

4.2 Register Definitions

4.2.1 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

4.2.2 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 Mode

AFLO	ClrDet	AdMde1	AdMde0	Flag3	Flag2	Flag1	Flag0

Asynchronous/Bisync/X.21 Modes

Parity	ParM1	ParM0	Ignore	Chl3	Chl2	Chl1	Chl0

Channel Option Register 2 (COR2)   14   17   B   R/W

HDLC Mode

0	FCSApd	0	CRCNinv	0	RtsAO	CtsAE	DsrAE

Asynchronous Mode

IXM	TxIBE	ETC	0	RLM	RtsAO	CtsAE	DsrAE

Bisync Mode

LRC	BCC	EBCDIC	CRCNinv	SYN3	SYN2	SYN1	SYN0

X.21 Mode

0	0	0	ETC	0	0	0	0

Channel Option Register 3 (COR3)   15   16   B   R/W

HDLC/Bisync Mode

sndpad	Alt1/S55	FCSPre	FCS	idle	pad2	pad1	pad0

Asynchronous Mode

ESCDE	RngDE	FCT	SCDE	Splstp	Stop2	Stop1	Stop0

X.21 Mode

SglSYN	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

Channel Option Register 6 (COR6) 1B 18 B R/W

Asynchronous Mode

IgnCR	ICRNL	INLCF	IgnBrk	NBrkInt	ParMrk	INPCK	ParInt

Bisync Mode

Special Frame Termination Character

X.21 Mode

SYN Character

Channel Option Register 7 (COR7) 04 07 B R/W

Asynchronous Mode

IStrip	LNE	FCErr	0	0	0	ONLCR	OCRNL

Special Character Register
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 Ranges
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 Register
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

0	0	0	0	0	0	0	poly

intel

4.2.3 Bit Rate and Clock Option Registers

Receive Bit Rate Period Register (RBPR)   **C9**   **CB**   **B**   **R/W**

Receive Bit Rate Period (Divisor)

Receive Clock Option Register (RCOR)   **CA**   **C8**   **B**   **R/W**

TLVal	res	DpllEn	Dpllmd1	Dpllmd0	ClkSel2	ClkSel1	ClkSel0

Transmit Bit Rate Period Register (TBPR)   **C1**   **C3**   **B**   **R/W**

Transmit Bit Rate Period (Divisor)

Transmit Clock Option Register (TCOR)   **C2**   **C0**   **B**   **R/W**

ClkSel2	ClkSel1	ClkSel0	res	Ext-1X	res	LLM	res

4.2.4 Channel Command and Status Registers

Channel Command Register (CCR)   **10**   **13**   **B**   **R/W**

Mode 1

0	ClrCh	InitCh	RstAll	EnTx	DisTx	EnRx	DisRx

Mode 2

1	ClrT1	ClrT2	ClrRcv[1]	0	0	0	0

1.Revision H and later; earlier revisions, this bit is '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 Mode

RxEn	RxFlag	RxFrame	RxMark	TxEn	TxFlag	TxFrame	TxMark

Asynchronous Mode

RxEn	RxFloff	RxFlon	0	TxEn	TxFloff	TxFlon	0

Bisync Mode

RxEn	RxITB	RxFrame	0	TxEn	TXITB	TxFrame	0

X.21 Mode

RxEn	0	RxSpc	0	TxEn	0	TxSpc	0

Modem Signal Value Registers (MSVR)　　　　　　　　　　　**R/W**
Modem Signal Value Registers (MSVR-RTS)　　**DC　DE　B　R/W**
Modem Signal Value Registers (MSVR-DTR)　　**DD　DF　B　R/W**

DSR	CD	CTS	DTRop	0	PortID	DTR	RTS

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]

4.2.4.1　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 Mode

0	EOF	RxAbt	CRC	OE	ResInd	0	ClrDct

Asynchronous Mode

Timeout	SCdet2	SCdet1	SCdet0	OE	PE	FE	Break

Bisync Mode

0	EOF	RxAbt	CRC	OE	0	0	0

X.21 Mode

LVal	SCdet2	SCdet1	SCdet0	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	NoTrans	Gap2	Gap1	Gap0

4.2.4.2 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	EOE	UE	BA/BB	0	TxEmpty	TxDat

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	Notrans	0	0	0

4.2.4.3   Modem/Timer Interrupt Registers

Modem Priority Interrupt Level Register (MPILR)   E1   E3   B   R/W

Modem Interrupt Register (MIR)   ED   EF   B   R

Men	Mact	Meo	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	Timer2	Timer1

Modem End Of Interrupt Register (MEOIR)   85   86   B   W

0	0	SetTm2	SetTm1	0	0	0	0

4.2.5   DMA Registers

DMA Mode Register (DMR)   F4   F6   F8   W

EnSync	0	0	0	ByteDMA	0	0	0

Bus Error Retry Count (BERCNT)   8D   8E   B   R/W

Binary Value

DMA Buffer Status (DMABSTS)   1A   19   B   R

TDAlign	RstApd	CrtBuf	Append	Ntbuf	Tbusy	Nrbuf	Rbusy

4.2.5.1 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/W
B Receive Buffer Status (BRBSTS)	4D	4E	B	R/W

Berr	EOF	EOB	0	0	0	0	2401own

Receive Current Buffer Address Lower (RCBADRL)	3C	3E	W	R
Receive Current Buffer Address Upper (RCBADRU)	3E	3C	W	R

4.2.5.2 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	UE	Append	0	INTR	2401own

Transmit Current Buffer Address Lower (TCBADRL)	38	3A	W	R
Transmit Current Buffer Address Upper (TCBADRU)	3A	38	W	R

4.2.6 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 (RTPRI)	26	25	B	R/W Async

Binary Value, bits 7:0

Receive Timeout Period Register high (RTPRh)   27   24   B   R/W Async

Binary Value, bits 15:8

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

intel.

5.0   Functional Description

5.1   Host Interface

The CD2401 is a synchronous device with an asynchronous bus interface. A stable input clock is required on the CLK pin — nominally 33 MHz. CLK is divided by two internally, and the resulting signal is an output on the BUSCLK pin. The baud-rate generators and timers are also related to CLK. The "AC Electrical Characteristics" in Chapter 9.0 shows that many input signal setup and output signal transitions are related to the edges of the CLK and BUSCLK signals. It is possible, however, to use the CD2401 in a purely asynchronous bus environment.

The CD2401 can act as either bus master during DMA transfers, or as a bus slave device during normal host read and write transfers. Both byte and word transfers are supported in each of the Bus Slave and DMA Bus Master modes. Figure 2 and Figure 3 show the signals involved in these transfers.

5.1.1   Host Read and Write Cycles

The host read and write cycles begin with the activation of the CS* and DS* signals. The DATADIR* and DATEN* signals control external data buffers. The falling edge of the DTACK* signal indicates that the transfer is complete. DTACK* is released when DS* is deasserted. At that time CS* should also be deasserted. The AS* is not used during slave cycles; it is an output during DMA transfers.

Note that the following open-drain and tristate outputs should have pull-up resistors attached: AEN*, AS*, DATADIR*, DATEN*, and DTACK*.

Figure 2. Host Read Cycle
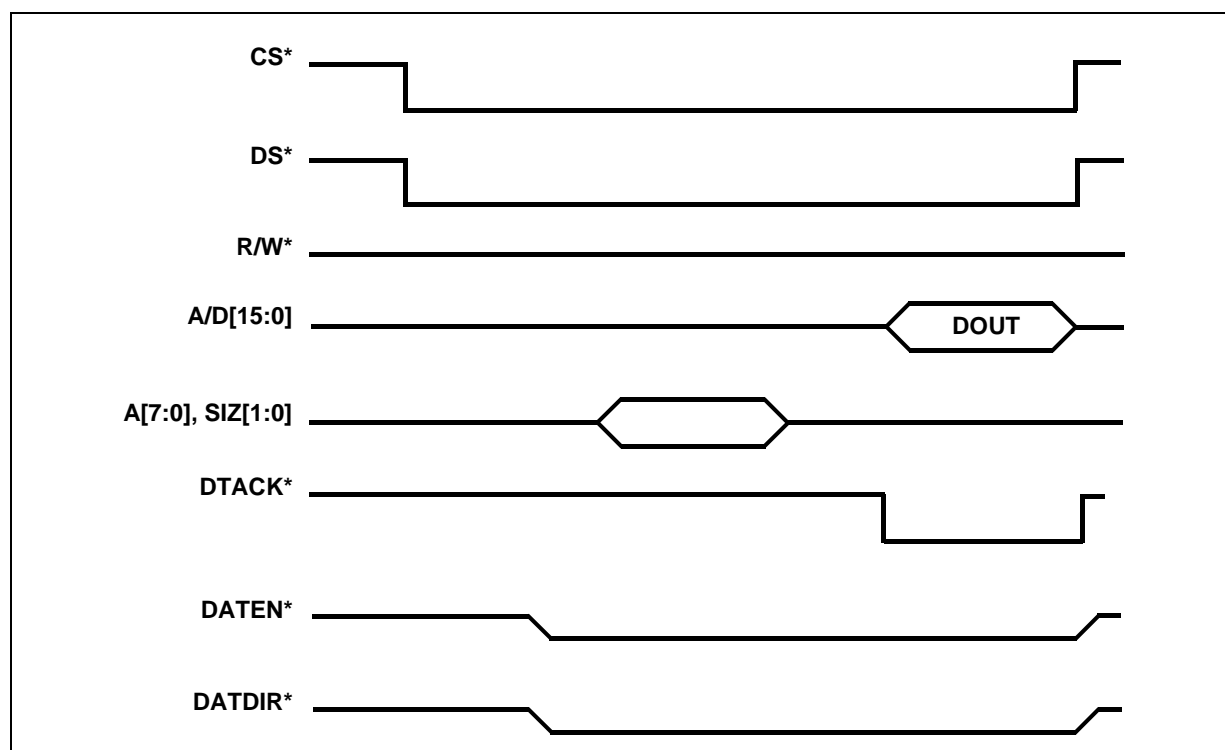
Figure 3. Host Write Cycle
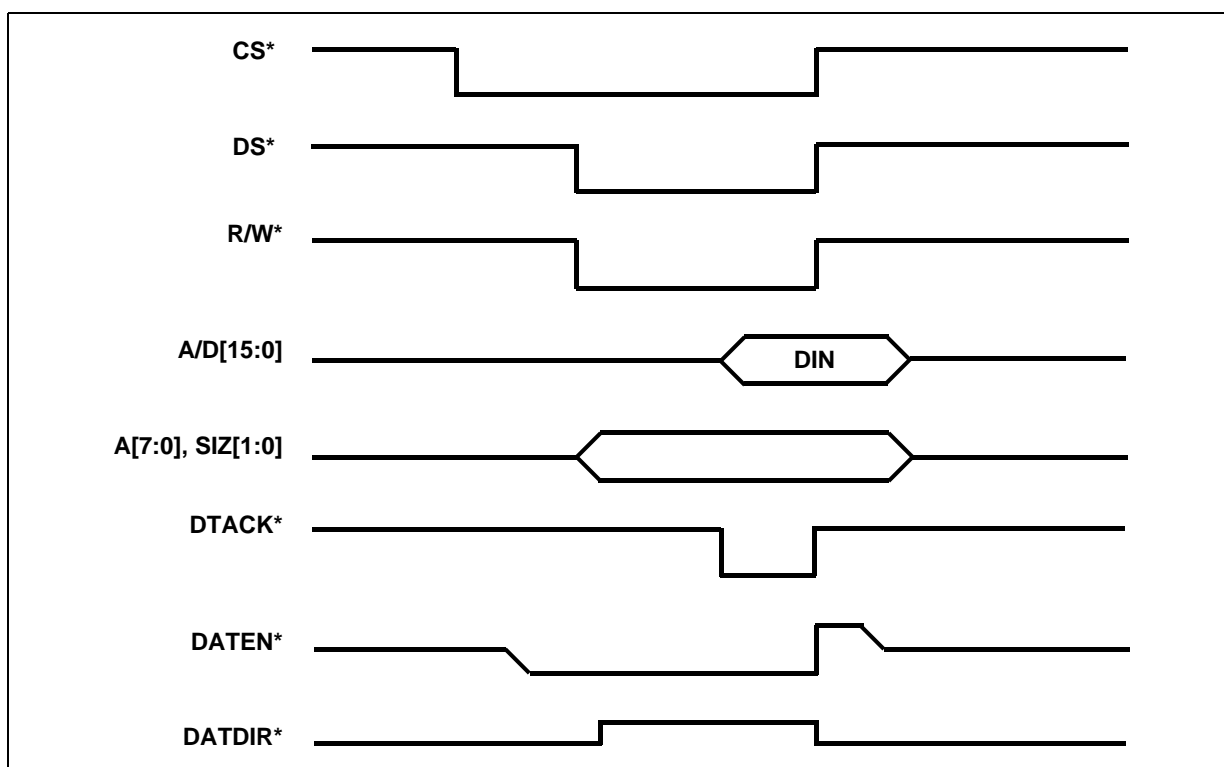
5.1.2   Byte and Word Transfers

Data can be moved to and from the CD2401 in either byte or word transfers. To accommodate various families of host processors, the BYTESWAP input pin is set to indicate the system byte-ordering scheme. SIZ[1:0] indicate whether the transfer is 1 or 2 bytes wide.

In systems where the even addresses represent the high-order byte, the BYTESWAP input pin should be tied low, and byte transfers occur on the A/D[15:8] pins for even addresses and on the A/D[7:0] pins for odd addresses. In systems where the high-order byte is on the odd address, the situation is reversed, and BYTESWAP should be tied high. Byte transfers to even addresses occur on the A/D[7:0] pins, and to odd addresses on the A/D[15:8] pins.

5.2   Interrupts

The CD2401 uses interrupt requests to alert the host that certain events have occurred. Interrupt operations on the CD2401 are tightly coupled with several registers described below. The concept of context affects the accessibility of these and other registers.

5.2.1 Contexts and Channels

The registers in the CD2401 are grouped into Global, Virtual, and four sets of Per-Channel registers. The CD2401 is normally in the background context, where the CAR selects the channel number for Per-Channel registers. The interrupt context begins with the interrupt acknowledge bus cycle, and ends with a write access to the appropriate End of Interrupt register. In the interrupt context, only the Per-channel registers for the channel number being serviced are available; the CAR has no effect. Most Global registers are available at all times, but some are shared by the four channels, such as the FIFO registers. These are called Virtual registers, and must be accessed only during an interrupt context.

Interrupt contexts can be nested so that a higher-priority interrupt service can preempt a lower priority interrupt already in progress. The CD2401 pushes the current interrupt context onto the stack, visible in the STK, and enters the context for the newly acknowledged interrupt. Any register accesses are in the new interrupt context until the host does a write to the appropriate EOIR for the top-level context. The CD2401 pops the top-level context off the stack and returns to the previous interrupt context.

5.2.2 Interrupt Registers

The IER and the LIVR are Per-Channel registers. IER contains bits to enable/disable the various interrupt sources within the CD2401. The LIVR value is output on the data bus during the interrupt acknowledge cycle. There are sets of three Global registers that correspond to the three types of interrupts: Receive, Transmit, and Modem. The Priority Interrupt Level registers (RPILR, TPILR, and MPILR) are programmed to contain the value that is present on the address bus during the interrupt acknowledge bus cycle for each type of interrupt. The Interrupt Status registers (RISR, TISR, or MISR) are examined during the interrupt service routine to determine the cause of each type of interrupt. TDR and RDR provide access to the FIFO buffers for each channel. These registers must not be accessed outside of the proper interrupt context. A write operation to the End of Interrupt registers (REOIR, TEOIR, or MEOIR) must be the last access to the CD2401 at the end of this handler routine to return it to its background context.

Figure 4. Interrupt Acknowledge Cycle

^{1} Interrupt vector is always on A/D[7:0].
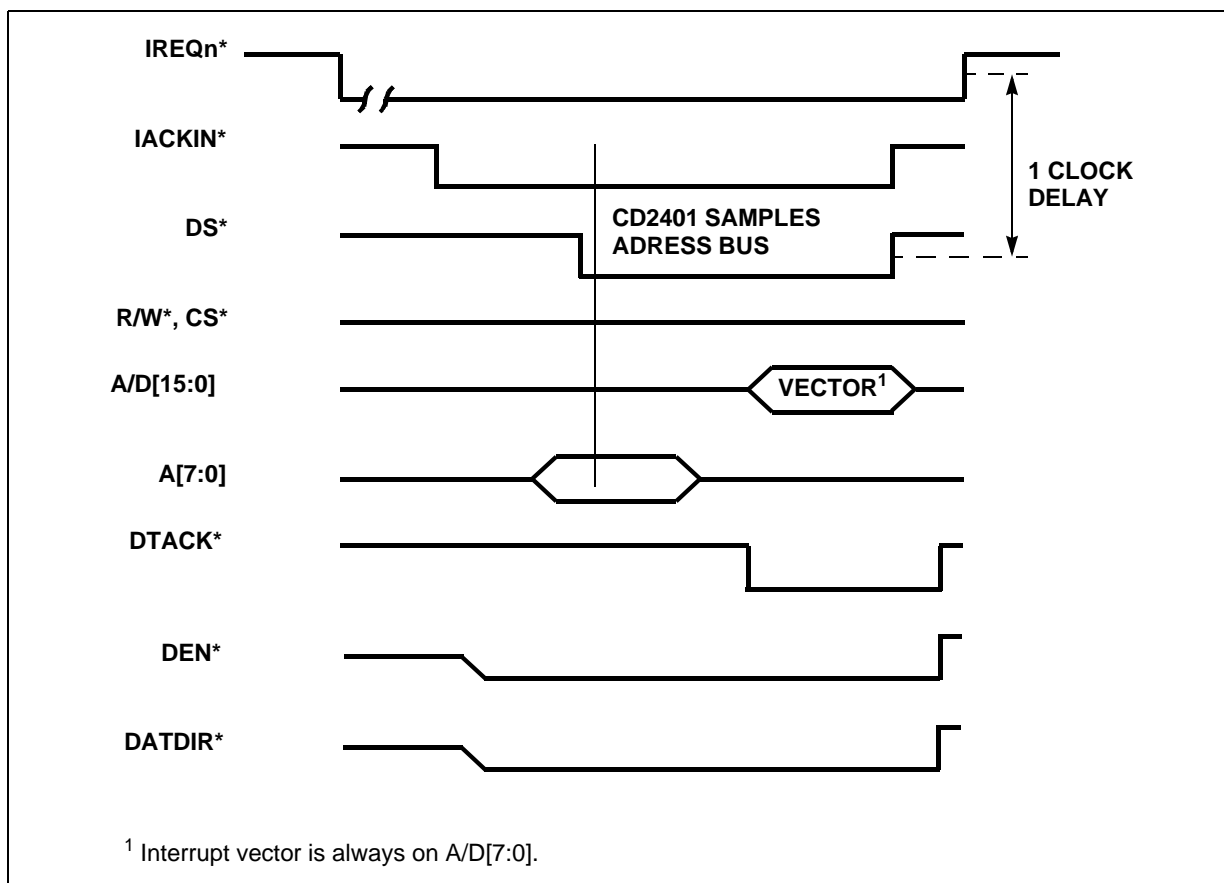
5.2.3   Groups and Types

There are two general reasons for the CD2401 to request service from the host processor — data transfer and exceptional conditions. Furthermore, interrupts are grouped into three categories, each with an associated Interrupt Request signal — IREQ1*, IREQ2*, and IREQ3*.

- Group 1 — Modem signal change/timer events

- Group 2 — Transmit interrupts

- Group 3 — Receive interrupts

Group 1 is used only for exceptions. Groups 2 and 3 include both data transfer and exceptions. Table 2 shows the possible causes of transmit and receive interrupt service requests. The cause of an interrupt request is encoded into the two least-significant bits of the vector presented on the data bus during the interrupt acknowledge cycle. The most-significant six bits of the vector come from the LIVR:

Interrupt Vector LSBs

00	Receive exception
01	Modem signal change or timer event
10	Transmit data or exception
11	Receive Good Data™

Table 2.   Transmit and Receive Interrupt Service Requests

Interrupt Cause	Async	HDLC	Bisync	X.21	Comments
Receive Good Data™	√	√	√	√	Not in DMA mode
BREAK detect	√				
Framing error	√				
Parity error	√		√	√	
Receive timeout, no data	√				
Special character match	√		√	√	
Transmitter empty	√	√	√	√	
Tx FIFO threshold		√	√	√	Not in DMA mode
Receive overrun	√	√	√	√	
Clear detect		√			
CRC error		√	√		
Residual bit count		√			
Receive abort		√	√		
End of frame		√	√	√	
Transmit underrun		√			
Bus error	√	√	√	√	DMA mode only
End of buffer	√	√	√	√	DMA mode only

5.2.4   Hardware Signals and IACK Cycles

The IACK bus cycle begins with IACKIN* and DS* asserted, and a value matching the appropriate PILR contents on the least-significant seven address bus bits (A[6:0]). If the IACK cycle is valid (that is, the PILR values match), the corresponding vector from the interrupting channel LIVR is driven onto the data bus and DTACK* is asserted. DTACK* is released after DS* is removed.

shows the interrupt acknowledge cycle timing. It is similar to the basic host read cycle except that IACKIN* is active, and CS* is inactive.

The three IREQn* pins are open-drain outputs requiring external pull-up resistors, nominally 4.7 k¾. The IACKOUT* is used to form a daisy-chain in systems with more than one CD2401 (see Section 5.2.5).

5.2.4.1   Programming the PILRs

The three Proirity Interrupt Level registers must be programmed with values that correspond to the least-significant seven address bits present on A[6:0] during the interrupt acknowledge bus cycle. Some CPUs output the priority level of the interrupts that are being acknowledged on the bus during the IACK cycle. In these systems the three PILR values are unique. In other systems that do not use this scheme, the PILR values can be the same or different depending on the specific design. When all of the PILRs contain the same value and multiple IREQn* lines are asserted, the CD2401 imposes the following priority scheme to determine which interrupt request is acknowledged:

Highest priority: Receive Interrupt register
Transmit Interrupt register

Lowest priority: Modem Interrupt register

5.2.4.2   Systems with Interrupt Controllers

Some systems use an interrupt controller that supplies its own vector during the interrupt acknowledge cycle. To function properly, the CD2401 needs an IACK cycle in response to its interrupt request. These systems can decode three distinct locations from the CD2401 to produce an IACKIN* instead of CS*. The PILRs should be programmed with the addresses of these three locations.

Alternatively, a single location can be decoded and the three PILRs given identical values as described above. In either case, the host should read one of these locations before the first access to the device in an interrupt service routine. The CD2401 enters its interrupt acknowledge context for the proper type and channel, and the data returned is the device interrupt vector from the LIVR.

5.2.5   Multi–CD2401 Systems

Multiple CD2401s can be chained together for systems requiring more than four channels. Each group of interrupt request lines (IREQn*) can be connected in a parallel wired-OR fashion. The system Interrupt Acknowledge signal is connected to the IACKIN* pin of the first device, and its IACKOUT* is then connected to the IACKIN* of the next device, and so on, forming a chain of CD2401s.

5.2.5.1   Keep-and-Pass Logic

The acceptance of an interrupt acknowledge cycle by the CD2401 depends on whether the part is requesting service and whether the least-significant seven address bits match the contents of the appropriate PILR. The following rules apply to the keep and pass logic.

1. If the CD2401 does not have an interrupt asserted, the interrupt acknowledge is passed out on IACKOUT*.

2. If the CD2401 is asserting one or more of its interrupts, but the interrupt priority levels driven on the address bus by the host do not match the contents of the appropriate PILR, this interrupt acknowledge is also passed out on IACKOUT*.

3. If the CD2401 is asserting an interrupt, and the interrupt priority level on the address bus matches the PILR for that interrupt type, the interrupt acknowledge is accepted by the CD2401 and the vector from the LIVR is driven onto the data bus.

5.2.5.2   Fair Share Scheme

When multiple CD2401 are chained, the Fair Share logic in these devices guarantees that the interrupts from all CD2401s in the system are presented to the host with equal urgency. There is no positional hierarchy in the interrupt scheme, that is, the CD2401 farthest from the host has as equal a chance of getting its interrupts through as that of the CD2401 nearest to the top of the interrupt chain. The Fair Share scheme is transparent to the user, and no enabling or disabling is required.

intel

When an interrupt request line is asserted, the Fair bit for that type of interrupt on the asserting device is cleared. The Fair bit remains cleared until the interrupt line returns to a high state. The CD2401 does not assert a new interrupt of that type while the corresponding Fair bit is cleared. Therefore, when multiple CD2401s assert interrupts together, each one is serviced in turn, before they can reassert the same interrupt type.

The IREQn* interrupt request lines are open-drain outputs that can be tied together in groups of the same type, creating a Fair Share scheme for each group of interrupts. Alternatively, all three groups can be tied to a common request using the CD2401 internal-priority scheme (see Section 5.2.4.1).

5.3 FIFO and Timer Operations

Each channel in the CD2401 has a 16-byte receive FIFO and a 16-byte transmit FIFO. The FIFOs are accessible through RDR and TDR. These Virtual registers are shared among the four channels; therefore, they cannot be accessed outside an interrupt context.

The threshold level of each channel is common for both FIFOs and is set by COR4, with a maximum threshold value of 12. The FIFO threshold is meaningful in both DMA and non-DMA modes. In DMA mode, the FIFO threshold determines when transfer bursts should occur. In non-DMA mode, the threshold level determines when transfer interrupts are asserted.

5.3.1 Receive FIFO Operation

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 Synchronous mode, an interrupt request for data transfer is initiated when the number of characters is greater than the FIFO threshold or an end of frame is reached.

5.3.2 Transmit FIFO Operation

The TxEmpty and TxD bits (IER[1:0]) control the generation of transmit FIFO interrupts. The CD2401 initiates an interrupt request for more data when the number of empty bytes in the FIFO is greater than the threshold set. During synchronous operation when the last byte of the frame is transferred to the FIFO, the CD2401 stops asserting transmit interrupts until the frame is sent.

5.3.3 Timers

The global TPR provides a timer prescale 'tick' as a clock source for the timers. The TPR counter is clocked by the system clock (CLK) divided by 2048. To maintain timer accuracy, the TPR should not be programmed with a value not less than 16 (10 hex) — a 'tick' of about 1 millisecond when CLK is 33 MHz.

Each channel has two timers, one 16-bit general timer 1 (GT1) and one 8-bit general timer 2 (GT2). Their operation and programming are different in synchronous and asynchronous protocols.

5.3.4   Timers in Synchronous Protocols

In synchronous protocols, the timers have no special significance for the CD2401; they are available to support the protocols. They are started by host commands or by interrupts generated by the CD2401. General timers 1 and 2 can be started in either of two ways:

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

2. By setting the SetTm2 or SetTm1 bits (R/T/MEOIR[5:4]) when terminating an interrupt service routine. In this case, the value should be written to the appropriate Interrupt Status register (RISR, TISR, or MISR).

These timers can be disabled by a command through the CCR.

5.3.5   Timers in Asynchronous Protocols

The receive timer is restarted from the value programmed in RTPR every time a character is received and loaded into the FIFO, or data is read by the host. For example, the receive FIFO threshold is set to eight, and six characters are stored in the receive FIFO. If no more characters are received and the receiver timer times-out, a receive interrupt is asserted (in DMA mode, DMA transfer occurs). The host is expected to retrieve all six characters from the receive FIFO. Assuming the host is still enabling this feature (that is, IER[5]) and if there is no character being received and receiver timer times-out, a receive exception timeout interrupt (a group 3 interrupt) is asserted. The timer can be disabled if the value in RTPR is set to '0' or the RET bit (IER[5]) is cleared.

5.3.6   Transmit Timer

The TTR is used only if the embedded transmit command is enabled in COR2. The delay transmit command specifies the delay period loaded in the TTR; no further transmit operations are performed until this timer reaches zero. The current state of the line is held at either '0' for send break or '1' for inter-character fill.

5.4   DMA Operation

The CD2401 uses a simple, but powerful, double-buffering method readily compatible with higher-level buffer control procedures (such as, circular queues, link lists, and buffer pools). Each transmitter and receiver is assigned an 'A' and 'B' buffer. When transmitting, the host processor alternately fills the A and B buffers, and commands the CD2401 to transmit the buffers one at a time. When receiving, the CD2401 fills the A and B buffers and informs the host processor when each is ready.

A simple Ownership Status bit is used for each buffer; this ensures that there are no deadlocks between the host and the CD2401 regarding the use of a particular buffer.

By using the simple and flexible DMA management of the CD2401, the user host processor is concerned with transmit/receive data on a block-by-block basis. The user need not be concerned with character-by-character transfers, or even filling and emptying the FIFOs. DMA control is user-selectable per-channel and operates independently of one another.

The CD2401 can perform DMA operations in any of the supported line protocols. A special Append mode feature can reduce host CPU overhead for asynchronous data streams. DMA operations are channel- and direction-specific. In each channel, either the transmitter, the receiver, or both, can be independently programmed for DMA mode by the CMR.

When the CD2401 acquires the bus for a DMA transfer, only data for one channel and in one direction is transferred; then, bus ownership is relinquished. A maximum of 16 bytes — the depth of the transmit and receive FIFOs — are transferred during any ownership cycle.

Whenever possible, DMA cycles are 16 bits wide, and buffers have the proper byte alignment. Unaligned buffers are sent using only 8-bit-wide transfers. If the buffer begins on an even address and contains an odd number of bytes, the CD2401 uses 16-bit transfers for all the words in the buffer, except the last transfer, which is 8 bits.

If one buffer in a chain ends on an odd address, the next buffer in the chain should start on an odd address to maintain proper alignment for the most efficient bus usage. In this case, only the last transfer of the first buffer and the first transfer of the next buffer is 8 bits wide; all others are 16 bits.

The CD2401 can be forced to perform only byte-wide DMA operations by setting the ByteDMA bit (DMR[3]).

5.4.1   Bus Acquisition Cycle

1. CD2401 asserts BR* and waits for BGIN*.
2. When BGIN* is detected, the CD2401 can access the bus after the current bus owner relinquishes control of the bus.
3. If BGACK* is high when BGIN* goes low, then the bus is free to access. Go to step 5.
4. If BGACK* is low when BGIN* goes low, then the bus is in use. The CD2401 waits for BGACK* to go high.
5. Once the CD2401 senses that BGACK* is high, it waits for the current bus cycle to terminate (DS* and DTACK* high), then asserts BGACK* by driving it low. At that time, the CD2401 'owns' the bus. After driving BGACK* low, the CD2401 drives BR* high.

Figure 5 is an example where the CD2401 was required to wait to access the bus.

5.4.2   DMA Data Transfer

After the CD2401 acquires the bus, it pulses ADLD* once. This loads the upper 24 address bits to the external 24-bit latch. This happens only once per DMA grant cycle. The AD[15:0] bits are remapped to Memory Address bits MA[31:16], and A[7:0] are mapped to MA[15:8]. If during DMA, the upper 24 bits need to change, the CD2401 relinquishes the bus and then re-acquires the bus.

During each DMA read and write cycle, the least-significant eight memory address bits, MA[7:0] come from A[7:0].

Figure 6 on page 46 is an example of one DMA access after bus is acquired.

Figure 5. Bus Acquisition Cycle

Another component owns the
bus and gives it up here. ⟶

The CD2401 owns the bus at this point. ⟶
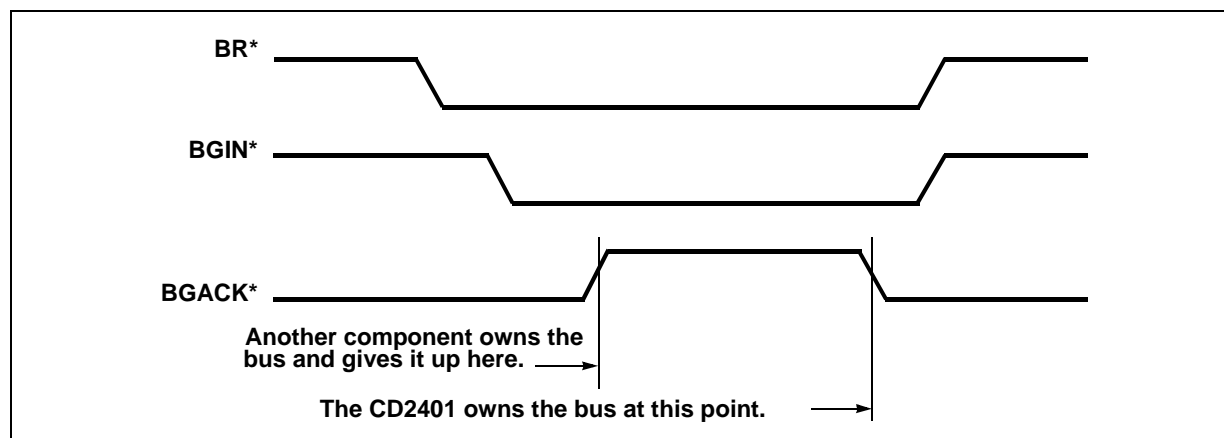
Figure 6. Data Transfer Timing
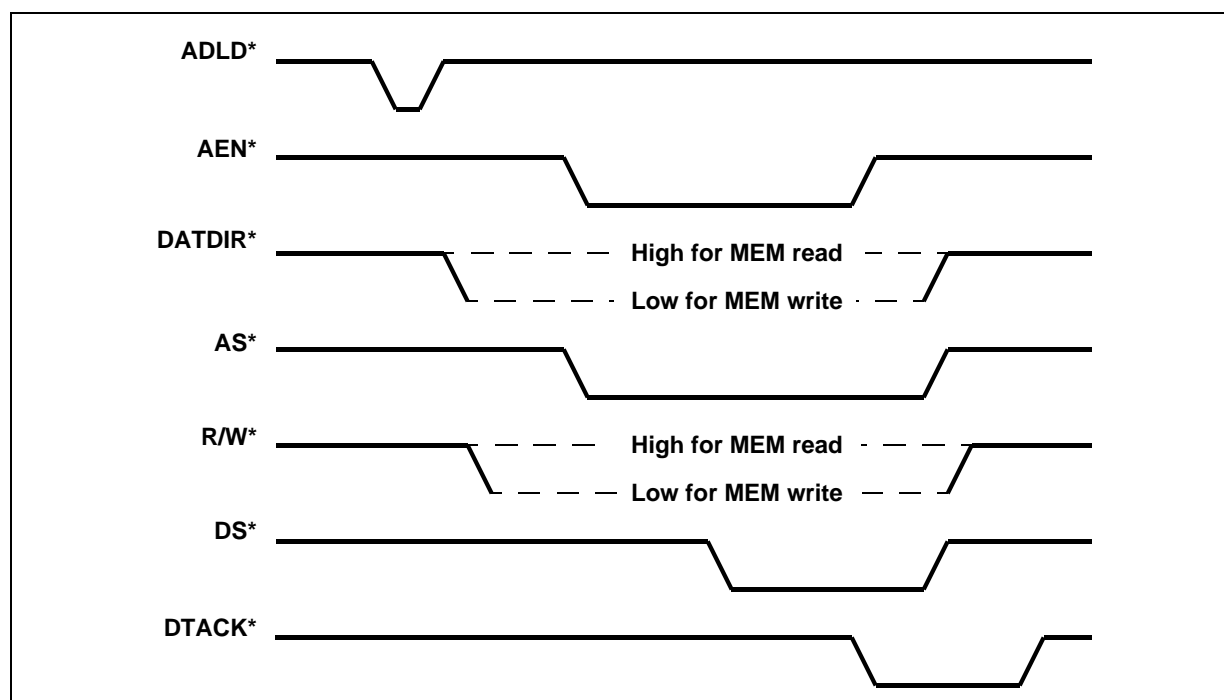
5.4.3   Bus Error Handling

When a bus error is detected during a DMA sequence, the CD2401 terminates the current bus cycle and relinquishes the bus. Any data transfer in the bus ownership cycle is ignored and the original conditions are restored. A subsequent retry attempt would start again from these original conditions.

If there is a non-zero value in the BERCNT register, the register is decremented and the failed transfer is retried automatically. If the BERCNT is zero, a bus error interrupt is generated and DMA transfers are suspended on the failing buffer until the interrupt is serviced.

5.4.4   A and B Buffers and Chaining

The buffer management of the CD2401 uses a dual-buffer scheme. There is an A and B buffer pair for each transmitter and each receiver. Each buffer is controlled by an Ownership Status bit, called 2401own. When 2401own is set to '1', the CD2401 owns the buffer. When 2401own is set to '0', the host owns the buffer. A simple rule prevents confusion in the buffer management: neither the CD2401 nor the host seizes buffer ownership. Each always relinquishes ownership to the other.

The host relinquishes ownership of a receive buffer to the CD2401 when the receive buffer is ready. The CD2401 is then free to write received data into the buffer. The CD2401 returns ownership of the receive buffer after the receive data is in the buffer. The host gives ownership of a transmit buffer to the CD2401 when the transmit buffer is ready to transmit. The CD2401 then transmits the contents of the buffer. When this is complete, the CD2401 returns ownership back to the host.

The CD2401 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 2401own bit and the 'next' bits is shown later. The receive buffers are handled in the same way using the Nrbuf (next receive buffer).

Table 3.   A and B Buffers and Chaining

Ntbuf	2401own Buffer A	2401own Buffer B	Transmit Action
0	0	0	Send nothing.
0	1	0	Host sets up buffer A.
1	1	0	CD2401 accepts buffer A and marks B as next.
1	0	0	CD2401 completes A transfer, and passes it to host.
1	0	1	Host sets up buffer B.
0	0	1	CD2401 accepts B and marks A as next.
0	1	1	Host sets up buffer A.
1	1	0	CD2401 completes B transfer, passes to host, accepts A and marks B as next.
1	0	0	CD2401 completes A transfer 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 RAM.

The EOF Status bit controls chaining in Synchronous modes. Chaining applies to both transmit and receive. For transmit, the host determines EOF bit; for receive, the CD2401 determines the EOF bit.

In Transmit DMA when the first buffer is supplied to the CD2401, 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 are appended, and the next buffer is again treated as the start of frame. If the EOF bit is not set, the CD2401 treats the buffer as the first part of a larger frame and chains into the next buffer (does not reset CRC); this process continues until a buffer is supplied with the EOF bit set.

5.4.5 Transmit DMA Transfer

As in receive data transfers, two buffers are available for DMA transmit transfers. The A/BTBADR[†] and A/BTBCNT registers contain the start address of and the byte count in the buffers. These registers are set by the host when initiating a transfer. The CD2401 makes a copy of the registers to perform the transfer, leaving the originals unchanged. Transfer of buffers between the host and the CD2401 is controlled by the A/BTBSTS registers.

Buffers can contain either complete frames or blocks of data, linked together to form a complete frame or a 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. Both are further 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 Chain 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 or block.

When the CRC bit is set, the CD2401 generates and transmits a cyclic redundancy check word for the frame using the polynomial selected by the CPSR. If the Interrupt Required bit is set, a host interrupt is generated after the buffer is transmitted.

Transmit buffers can be chained to support large frames. To minimize bus usage, the first buffer of the chain should begin on an even address in host memory. The CD2401 begins fetching a frame from a buffer performing DMA transfer, reading two bytes at a time. The CD2401 cannot realign data between external memory and the FIFO. If one buffer of the chain ends on an odd address, the next buffer in the chain should begin on an odd address. Otherwise, only single-byte transfers are made for the rest of the buffer.

Append Mode Transfer (Buffer A only)

Append mode transfers are available for buffer A in Asynchronous mode only. If buffer A is set to Append mode, the host can enable the CD2401 to transmit data in the buffer before it is completely filled. The CD2401 starts transmitting new data when it is appended to the buffer.

This mode is useful for terminal echo routines that 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 CD2401 by the host; ATBADR and ATBCNT are initialized. Subsequent triggering of DMA transfer occurs by programming ATBCNT with the accumulated byte count. In this case, ATBCNT should be written as a 16-bit word to avoid confusion between 2-byte operations. ATBADR should not be reprogrammed during Append mode. If the memory space has to be moved, the Append mode must first be disabled. When the final data is added to the append buffer and ATBCNT has been updated, the host should set the AppdCmp bit (STCR[5]). When the CD2401 has completed the final transmission, it clears the 2401own bit (ATBSTS[0]), and generates an end-of-buffer interrupt.

---

†. A/B is used as a Buffer register abbreviation indicating A buffer / B buffer followed by the register acronym.

Figure 7. Transmitter A and B Buffers

NOTE: The number of bits in each register is shown in parentheses (). Buffer A and buffer B do not need to be the same length.
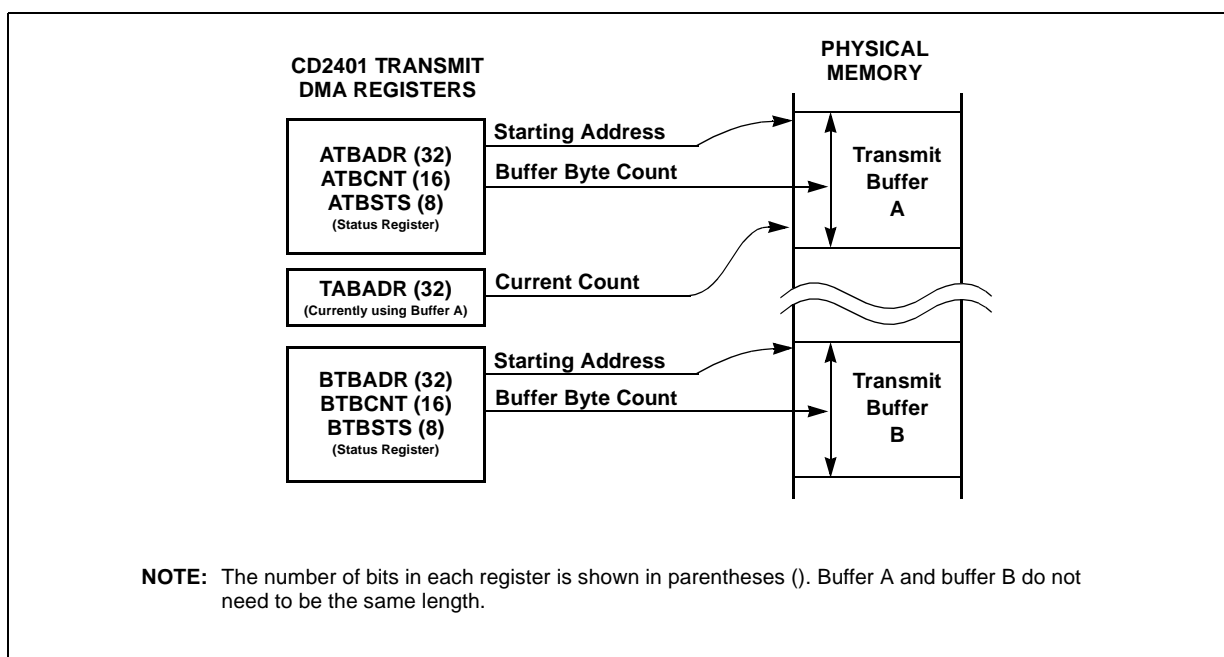
5.4.6 Synchronous Transmitter Examples

In Figure 7, buffers A and B are contained in RAM external to the CD2401. All else (DMABSTS, A/BTBADR, TCBADR, A/BTBCNT, and A/BTBSTS) is inside the CD2401.

Example 1

Transmit a frame out of channel 1 — no chaining.

1. The host checks the Ntbuf bit (DMABSTS[3]) for channel 1 to determine which buffer is next. In this example, Ntbuf is set to '0' indicating that buffer A is used next.

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

3. The host sets up the ATBSTS register. The EOF bit is set to indicate that there is no chaining. The 2401own bit is set to give ownership to the CD2401. By setting 2401own, the host commands the CD2401 to start transmission. Thus, everything must be ready (starting address, buffer data, byte count) prior to setting 2401own.

4. The CD2401 starts frame transmission out of channel 1. When transmission is started, the CD2401 sets Tbusy bit (DMABSTS[2]). As transmission progresses, the current buffer pointer (TCBADR) is updated by the CD2401. Also, at the start of transmission the Ntbuf bit is set to '1' to notify the host that buffer B is next.

5. The CD2401 completes frame transmission by adding any necessary CRCs and trailing-frame delimiters.

6. When the CD2401 completes the transmission, it clears the Tbusy bit. Then it sets the EOB bit and clears the 2401own bit (ATBSTS[0]). This notifies the host that the transmission is complete, and the CD2401 returns ownership of the buffer back to the host.

7. The CD2401 optionally interrupts the host, with EOF and EOB (TISR[6:5]) both set to indicate that the transmission is complete, and there was no chaining.

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 (DMABSTS[3]) for channel 0 to determine which buffer is next. In this example, Ntbuf is set to '1' indicating that buffer B is 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 is set to 100.

3. The host sets up the BTBSTS register. The EOF bit is cleared to indicate that this buffer is the first link in a chain. The 2401own bit is set to give ownership to the CD2401. By setting 2401own, the host commands the CD2401 to start transmission. Thus, everything must be ready (starting address, buffer, data count) prior to setting 2401own.

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

5. The CD2401 starts transmitting buffer B from channel 0. When this is started, the Ntbuf bit is cleared to '0' to indicate that buffer A is next. This helps the host keep track of which buffer is next. As transmission progresses, the current buffer pointer (TCBADR) is updated by the CD2401. During or prior to this, the host has readied buffer A. For buffer A, the EOF bit (ATBSTS[6]) 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 CD2401 does not add any CRCs or end-of-frame delimiters because there is more data for the current frame.

7. After the CD2401 has completed transmission of the first link out of buffer B, the CD2401 sets the EOB bit and clears the 2401own bit (BTBSTS[0]). This notifies the host that the transmission is complete, and returns ownership of the buffer back to the host.

8. The CD2401 optionally interrupts the host with EOF clear and EOB set (TISR[6:5]) to indicate that the transmission is complete, and that there was chaining.

9. The CD2401 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 CD2401 continues transmission of the current frame, but now transmission is from buffer A. This is the second link, and is 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 is set to 40 bytes.

11. After the CD2401 has completed transmission of the second link out of buffer A, the CD2401 sets the EOB bit and clears the 2401own bit (ATBSTS[0]). This notifies the host that the transmission is complete, and returns ownership of the buffer back to the host. As with the first link, the CD2401 does not add CRCs or ending frame delimiters to this link.

12. The CD2401 optionally interrupts the host with the EOF bit cleared and the EOB bit set (TISR[6:5]) to indicate that the transmission is complete, and there was chaining.

13. By this time, the host has set up a new buffer for buffer B. The EOF bit (BTBSTS[6]) is set to indicate that this is the last link in the chain.

14. The CD2401 then transmits buffer B in the same manner as above. As before, the CD2401 transmits the number of bytes indicated in the BTBCNT, which is 40 for the third segment.

15. When the CD2401 completes transmission, any necessary CRCs and ending frame delimiters are transmitted.

16. The CD2401 optionally interrupts the host with EOF and EOB set (TISR[6:5]) to indicate that the transmission is complete, and this is the last link in the chain.

5.4.7 Receive DMA Transfer

In all protocol modes, two host memory buffers can be made available to each receive channel, by the A/BRBADR and A/BRBCNT registers. To make a buffer available, the user must supply the buffer address in the A/BRBADR registers; the number of free bytes in the buffer must be written in the A/BRBCNT registers, and the buffer status must be updated in the A/BRBSTS register. The CD2401 is now free to use the buffer for receive data, and update the appropriate Buffer Status register. When the buffer is no longer in use, the CD2401 writes the number of bytes stored in the buffer in RBCNT and updates status in RBSTS. This frees the host to take control of this buffer and supply a new buffer in its place. The CD2401 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 CD2401 16-byte FIFO to respond and avoid loss of data.

Special actions are taken depending on the channel protocol. In HDLC mode, the end-of-frame/ data block boundaries are recognized by the CD2401. When a data-block boundary is detected, the current buffer is automatically terminated. If the other buffer is allocated and owned by the CD2401, it becomes the current buffer. End-of-frame and block interrupts are also generated to the host.

In Asynchronous mode, a host interrupt is generated when there are receive exceptions (framing error, special character, and so on), 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 is buffered internally in the FIFO until the host services the exception interrupt. The host has the following 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 by RCBADR.

3. A user-defined gap can be left in the buffer.

These selections are communicated to the CD2401 by the value written by the host to the REOIR, when the Receive interrupt service is complete. 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 RCBADR. 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 CD2401 automatically switches to the other buffer and advances the RCBADR enough to complete the desired gap. The CD2401 readjusts data alignment in its internal FIFO as needed to maintain alignment with the external buffer.

Receiver A and B Buffers

In Figure 8, buffers A and B are contained in RAM external to the CD2401. All others (DMABSTS, A/BRBADR, A/BRBCNT, A/BRBSTS, and RCBADR) are inside the CD2401.

Figure 8. Receiver A and B Buffers

NOTE: The number of bits in each register is shown in parentheses (). Buffer A and buffer B do not need to be the same length.
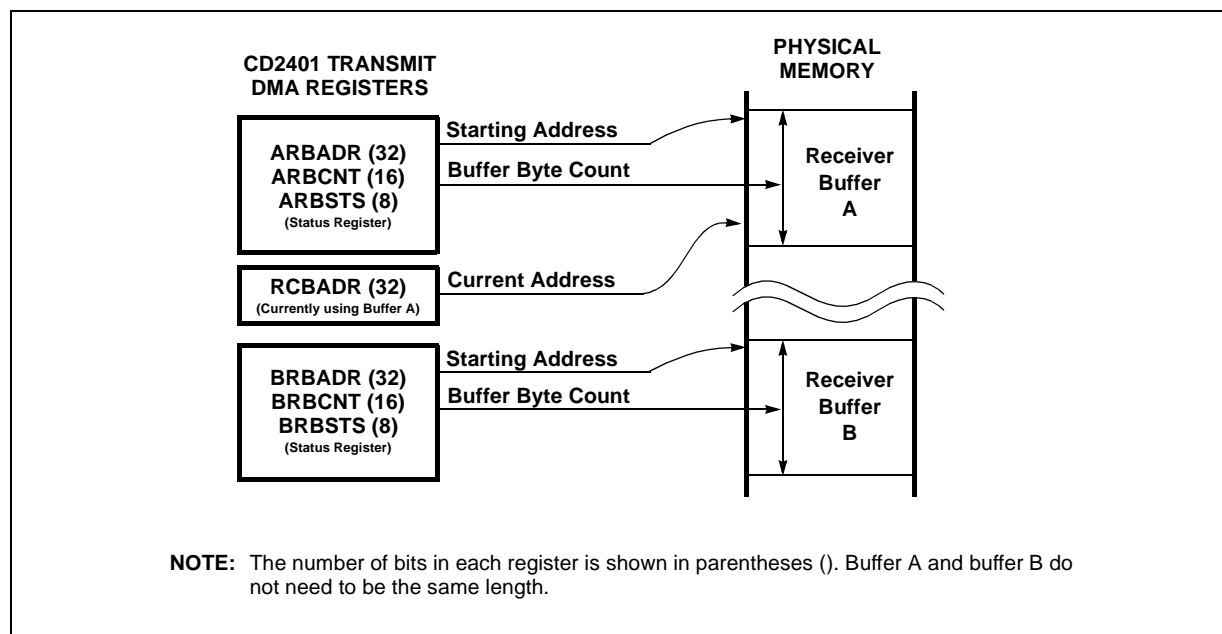
Example 1

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 (DMABSTS[1]) for channel 1 to determine which buffer is next. In this example, Nrbuf set to '0' indicates buffer A is 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 CD2401 by setting the 2401own bit (ARBSTS[0]). This notifies the CD2401 that it is now alright to write received.

4. The Rbusy bit (DMABSTS[0]) for channel 1 is '0' until a frame starts to be received. When frame data starts coming in, the CD2401 sets Nrbuf to notify the host that buffer B is next. As data bytes are written into the buffer, the current buffer pointer (RCBADR) is updated by the CD2401.

5. At the end of the received frame, the CD2401 tests for the correct end-of-frame delimiter and CRC. When the received frame is complete, the CD2401 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 CD2401 writes the value of the actual received byte count into the same register — ARBCNT. (Note that the host has written the maximum buffer size in ARBCNT when the buffer is given to the CD2401. But when the buffer is returned to the host, the CD2401 has written the actual byte count of the received buffer into ARBCNT.)

6. The CD2401 sets the EOB and EOF bits. This notifies the host that the end of the buffer and frame have been reached. The CD2401 also clears the 2401own bit to return the buffer to the host.

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 (DMABSTS[1]) for channel 0 to determine which buffer is next. In this example, Nrbuf set to '1' indicates buffer B is 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 to '100'.

3. The host then sets the 2401own bit to relinquish ownership to the CD2401.

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 is a receiver overrun, and the received frame is lost.

5. Suppose that the CD2401 starts receiving data into buffer B of channel 0. When this is started, the Nrbuf bit is cleared to '0' by the CD2401 to help the host keep track of which buffer is next. (During or prior to this, the host has made buffer A ready.)

6. After the CD2401 has received the first link of the frame into buffer B, it sets the EOB and SOB bits and clears the EOF bit. This indicates that the first link in a chain has been received. Also, the CD2401 clears the 2401own bit and returns ownership of the buffer 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 CD2401 optionally interrupts the host with EOF clear and EOB set (RISRh[6:5]) to indicate that the received buffer is complete, and that there was chaining.

8. The CD2401 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 goes into buffer A. This is the second link that is 100 bytes long. During this time, the host must set up a new buffer B for the third and final link.

10. After the CD2401 has received the second link into buffer A, it sets EOB bit and clears the 2401own bit (ARBSTS[0]). This returns ownership of the buffer 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 CD2401 optionally interrupts the host with EOF clear and EOB set (RISRh[6:5]) 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 CD2401 receives data into buffer B in the same manner previously explained.

14. In this example, the third link does not fill the buffer. Thus, when the end-of-frame delimiter is detected by the CD2401, the value of 40 (for 40 received bytes) is written into the received byte count (BRBCNT).

15. Next, the CD2401 sets the EOB and EOF bits to show that the buffer is complete, and that this is the last link in the chain.

16. The CD2401 optionally interrupts the host with EOF and EOB set (RISRh[6:5]) to indicate that the received frame is complete, and this was the last link in the chain.

5.4.7.1   Buffer Allocation

The CD2401 contains two DMA descriptors that can be loaded by the CPU to specify transmit buffers. These descriptors are designated A and B, and each consists of a 32-bit address (A/BTBADR), a 16-bit count (A/BTBCNT), and an 8-bit status (A/BTBSTS).

The Status register contains an Ownership Status bit (2401own). When this bit is set, the CD2401 owns the descriptor, it should not be written to by the CPU. When the bit is clear, the descriptor is owned by the CPU.

When DMA is selected and the channel is enabled, the CD2401 waits for ownership of buffer A. When ownership of A is given by setting the 2401own bit, the buffer is transmitted and the ownership bit is cleared. The CD2401 waits for ownership of buffer B; this process continues, toggling between the two buffer descriptors.

The DMABSTS register contains a status bit (NtBuf) that informs the CPU of the next buffer to transmit to ensure that the CPU and CD2401 stay in synchronization. This procedure ensures that a pipeline of data is available for the CD2401 to send, maximizing the bandwidth utilization and minimizing the possibility of underruns. Figure 9 illustrates this procedure.

5.4.7.2   Interrupts for Transmit DMA Buffers

Two types of transmit interrupts are available in DMA mode; they are enabled by the IER and controlled by the TxD and TxMpty bits.

When the TxMpty interrupt is enabled, interrupts are generated when there is no transmit data available to send. For example, the TxMpty interrupt can be used by the CPU to determine when line turn-around can occur on half-duplex lines.

Normally, the TxD interrupt indicates the end of each transmit buffer. The interrupt is scheduled internally when the last data is read from the transmit buffer into the FIFO.

Because only one interrupt is generated for each buffer, the TxD bit (IER[0]) can be left permanently enabled. If interrupts are required selectively for individual buffers, the INTR bit (A/BTBSTS[1]) registers can selectively enable interrupts.

5.4.7.3   Chained Buffers

In Synchronous modes when the frame size exceeds the maximum buffer size, a frame can be transmitted from a number of separate buffers. This is achieved simply by *not* setting the EOF bit (A/BTBSTS[6]) until the last buffer of the frame. The CD2401 transmits the buffers as one frame; it appends the CRC only when all the data is transmitted from the buffer with the EOF flag set.

If the above procedure for allocating buffers is used, the CPU has the transmission time of the last buffer to allocate the next to avoid possible underrun. The EOF bit (TISR[6]) is set for the interrupt associated with the last buffer.

intel®

Figure 9. DMA Transmit Buffer Selection

```
                        ┌─────────────┐
                        │    Start    │
                        └─────────────┘
                               │
                               ▼
                     ┌──────────────────┐
                     │      Read        │
                     │   DMABSTS to     │
                     │  Determine Next  │
                     │  Transmit Buffer │
                     │     (NtBuf)      │
                     └──────────────────┘
                               │
                               ▼
                    ◇ Next Buffer
         0 ────────◇ 2401own Bit ◇──────── 1
         │          ◇      ?      ◇          │
         ▼                                   │
  ┌──────────────┐                           │
  │    Update    │                           │
  │ Descriptor and│                          │
  │ Set 2401own  │                           │
  └──────────────┘                           │
         │                                   │
         ▼                                   │
      ◇ More Data                            │
  N ─◇ to Send ◇── Y ──────────────────────►│
     ◇    ?    ◇                             │
         │                                   ▼
         │                          ◇ Other Buffer
         │               0 ────────◇ 2401own Bit ◇── 1
         │               │          ◇      ?      ◇    │
         │               ▼                             │
         │      ┌──────────────┐                       │
         │      │    Update    │                       │
         │      │ Descriptor and│                      │
         │      │ Set 2401own  │                       │
         │      └──────────────┘                       │
         │               │                             │
         │               ▼                             │
         │        ┌─────────────┐                      │
         └───────►│     End     │◄─────────────────────┘
                  └─────────────┘
```
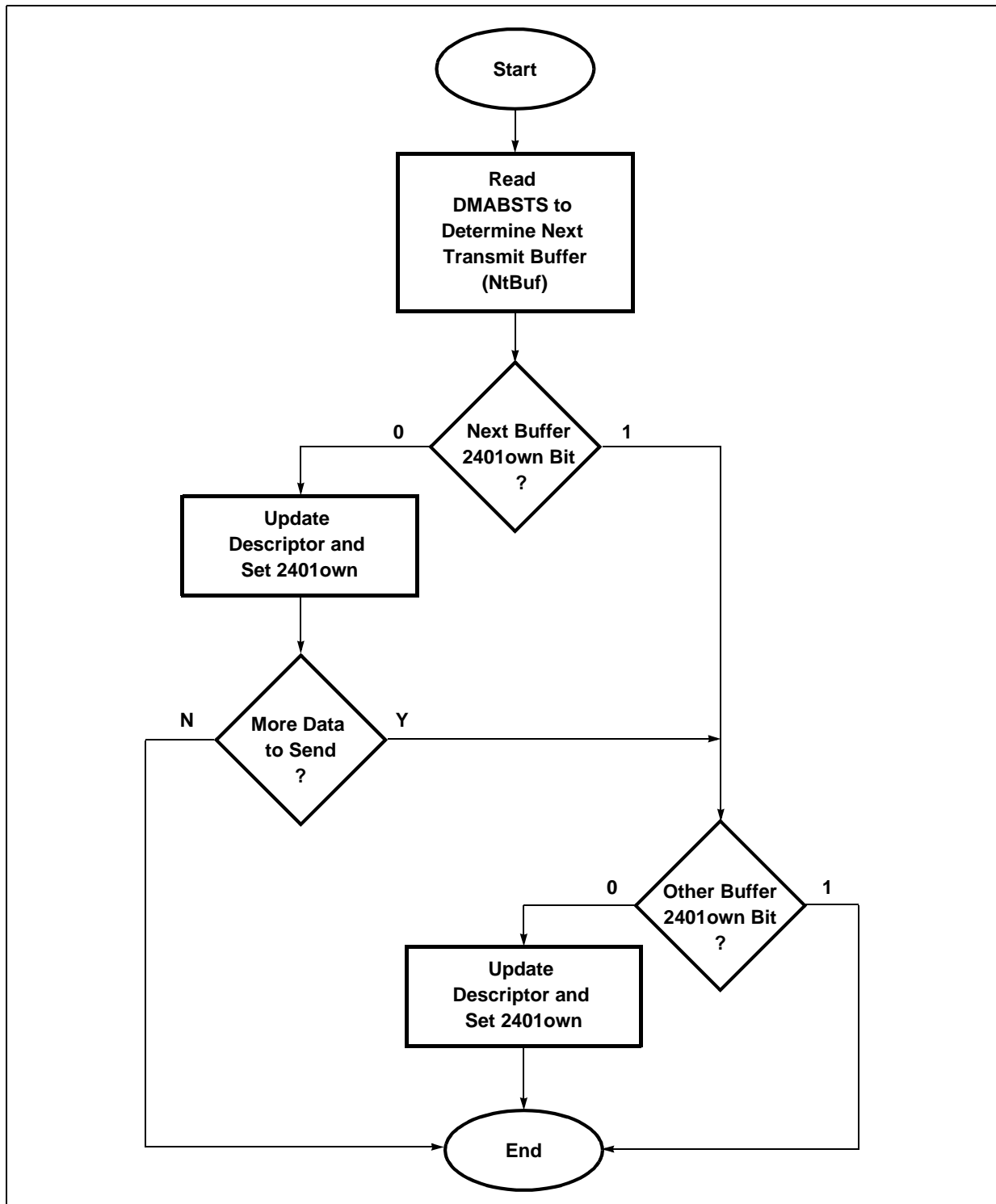
5.4.7.4   Append Mode

The Append mode reduces the CPU overhead required to provide asynchronous terminal echoing functionality; this is also necessary for any similar application that involves an unpredictable data stream. The A buffer can be set into Append mode by the ATBSTS register. This buffer can then be used for the echoed data, while the B buffer is used for all other output data. The append buffer allows data transmission to start from a buffer before all the data is available for transmission. For example, terminal echoing requires that each character is echoed (or translated and echoed) before the complete line is typed.

To operate in Append mode, the ATBADR and ATBCNT would be set as normal (the ATBCNT can be zero), and the 2401own and Append bits set (ATBSTS[0, 3]). When any data is available for transmission, it is placed in the RAM buffer by the CPU, and the total buffer byte count is updated in the ATBCNT. The CD2401 can now scan the ATBCNT register for any changes; if new data is found, it is read from the buffer and transmitted.

When no more data is found in the Append buffer, the CD2401 scans the B buffer for ownership. If the B buffer is owned by the CD2401, then the data in that buffer is transmitted uninterrupted; at the end of the transmission, the A buffer count continues to be scanned for new data.

For correct operation of this feature, the ATBCNT register should be updated with a word-write operation. If only byte access is possible, the value should not exceed 256 bytes. This mode allows multiple transfers to be performed through a single buffer; it saves CPU overhead by either processing multiple buffers or in handling interrupts with every character.

Line retransmission becomes as simple as 'stepping back' in the buffer and resending. To terminate Append mode, a command can be given by the STCR to cause the A buffer to be terminated when all current data has been sent.

5.4.7.5   Transmit Bus Errors

When a transmit bus error interrupt is generated, the TISR and A/BTBSTS registers indicate a bus error status. The current transfer address is available in the TCBADRs and the bus error occurred on the last transfer that started at this address. This means the actual error address can be up to 16 bytes further in the buffer.

Following a bus error condition, the CPU can either discontinue the current buffer or retry from the start of the last transfer. To discontinue the current buffer, the TermBuff bit should be set when TEOIR is written to at the end of the interrupt. In Synchronous mode, the frame is still in progress and needs to be aborted by the STCR.

To retry the frame, the CPU should set the 2401own bit (A/BTBSTS[0]), and not set the TermBuff bit when writing to TEOIR at the end of the interrupt. This causes the last transfer to be retried; should a bus error occur again, the above procedure is repeated. The CPU should check to ensure that a bad location is not continually retried.

5.4.7.6   Receive Buffer Interrupts

When a receive buffer is complete, the CD2401 generates an end-of-frame receive exception interrupt. It provides the CPU with RISR status and information on which buffer is complete.

When a receive error occurs, the device stops DMA at the point of error and generates a bus error receive exception interrupt. RISR indicates the cause of the exception, and RCBADR provides the next location in the receive buffer.

The CPU has the following five options:

1. Terminate the buffer.
2. Discard the exception.
3. Terminate the buffer and discard the exception.
4. Continue from the current position in the buffer.
5. Leave an 'n'-byte gap in the buffer and then continue.

The required option is written to the REOIR by the CPU to terminate the interrupt. If the terminate buffer option is chosen, the 2401own bit (A/BRBSTS[0]) should first be cleared by the CPU, or a new buffer can be supplied by the CPU.

5.4.7.7   Receive Timeout in Asynchronous DMA Mode

In Asynchronous DMA mode, the only way that the CD2401 releases the ownership is reaching the end-of-buffer. Receive timeout, or any exceptions, do not release the ownership if end-of-buffer condition is not met. The following illustrates recommended procedures to handle a receive timeout in Asynchronous DMA mode.

Scenario 1: Buffer A is currently selected; receive timeout occurs; host wants to continue on.
Recommendation: Do nothing in the receive timeout interrupt service routine.

Scenario 2: Buffer A is currently selected; receive timeout occurs; host no longer requires DMA.
Recommendation: Reset the 2401own bits (A/BRBSTS[0]) and set TermBuff (REOIR[7]) in the receive timeout interrupt service routine.

Scenario 3: Buffer A is currently used; receive timeout occurs; host wants to start DMA in buffer B.
Recommendation: Set TermBuff (REOIR[7]) in the receive timeout interrupt service routine. The CD2401 switches to buffer B.

Note: When a receive timeout occurs in buffer B, the CD2401 pops back to buffer A, unless the host clears both Ownership Status bits.

The above scenarios apply if buffer B is selected first.

5.4.7.8   Receive Bus Errors

When a receive bus error interrupt is generated, the RISR and A/BRBSTS registers indicate a bus error status. The current transfer address is available in RCBADR, the bus error occurred on the last transfer that started at this address. This means that the actual error address can be up to 16 bytes further in the buffer.

Following a bus-error condition, the CPU can either discontinue the current buffer or retry from the start of the last transfer. If the buffer is to be discontinued, the number of valid receive bytes can be calculated by subtracting the starting address (A/BRBADR) from the current address (RCBADR). The CPU should set the TermBuff bit (REOIR[7]) to terminate this buffer and move to the next.

The transfer that failed to the first buffer, due to the bus error, is still in the receive FIFO and is transferred to the next buffer following the end of the interrupt.

To retry the buffer from the failure point, the CPU should set the 2401own bit (A/BRBSTS[0]); the CPU should not set the TermBuff bit when writing to REOIR at the end of the interrupt. This causes the last transfer to be retried; should a bus error occur again, the above procedure is repeated. The CPU should check to ensure that a bad location is not continually retried.

5.5 Bit Rate Generation and Data Encoding

BRG and DPLL Operation

Data clocks are generated in the CD2401 by feeding one of a number of clock sources into a programmable divider. The clock source and divisor are user-programmable separately for each channel and direction. Clock options are programmed in TCOR and RCOR. The divisors are programmed in TBPR and RBPR. The possible clock sources are as follows:

Transmit

- Clk 0 – CLK input ÷ 8
- Clk 1 – CLK input ÷ 32
- Clk 2 – CLK input ÷ 128
- Clk 3 – CLK input ÷ 512
- Clk 4 – CLK input ÷ 2048
- TXCIN pin
- Receive bit clock

Receive

- Clk 0 – CLK input ÷ 8
- Clk 1 – CLK input ÷ 32
- Clk 2 – CLK input ÷ 128
- Clk 3 – CLK input ÷ 512
- Clk 4 – CLK input ÷ 2048
- RXCIN pin

The CLK input is nominally 35 MHz.

The divisor can be programmed for values from 1–255. 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 can be used, and it can 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, (1× 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 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.

Section 5.2 contains examples for programming standard bit rates. The value to be loaded to set a given bit rate is determined with the following equation:

$$\text{Bit rate divisor} \; = \; \frac{\text{Frequency of chosen clock source}}{\text{Desired bit rate}} - 1$$

Note: This equation, in general, yields 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

This example illustrates programming the bit rate generator at 64 kbps using the internal clock with a system clock frequency of 35 MHz.

Divisor loaded into R/TBPR[†] = 67 or 43h
Value loaded into R/TCOR = 00h, to select Clk0

Example 2

This example illustrates programming the bit rate generator at 56,000 bps using external clock. Again, the system clock frequency is assumed to be at 35 MHz.

The user provides a 1.25-MHz clock on the RXCIN or TXCIN pin.
Divisor loaded into R/TBPR = 21 or 15h
Value loaded into RCOR = 06h, to select External Clock mode
Value loaded into TCOR = C0h, to select External Clock mode

---

†. R/T is used as a register abbreviation indicating Receive / Transmit followed by the register acronym.

Figure 10. BRG and DPLL
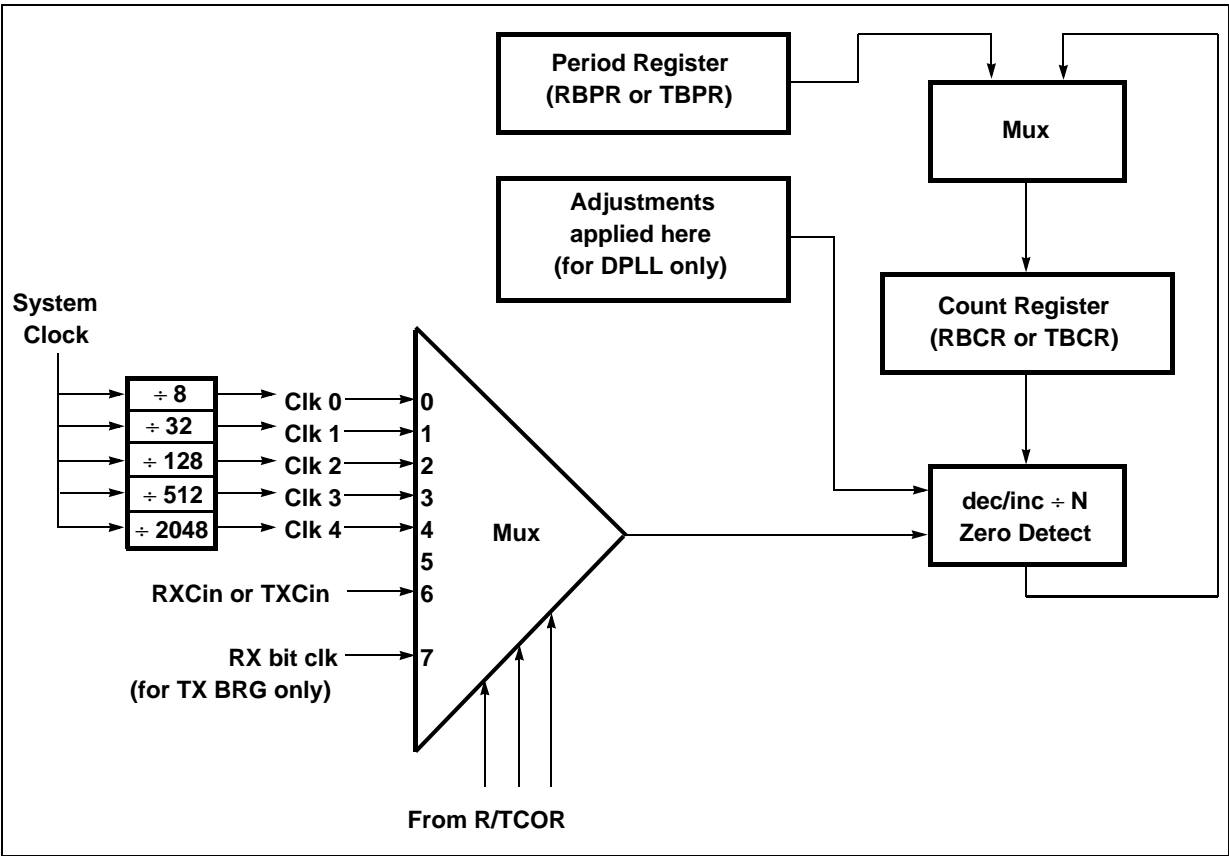
Receive Clock Option Register (RCOR)   CA   C8   B   R/W

TLVal	res	dpllEn	Dpllmd1	Dpllmd0	ClkSel2	ClkSel1	ClkSel0

Transmit Clock Option Register (TCOR)   C2   C0   B   R/W

ClkSel2	ClkSel1	ClkSel0	res	Ext-1X	res	LLM	res

Table 4. Clock Source Select (Sheet 1 of 2)

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	

Table 4. Clock Source Select (Sheet 2 of 2)

ClkSel2	ClkSel1	ClkSel0	Select	
1	1	0	External clock	
1	1	1	Reserved	(RCOR)
1	1	1	Receive clock	(TCOR)

Table 5. Bit Rate Constants, CLK = 20 MHz

Bit Rate	Divisor[1]	Clock	Error
50	c2	Clk 4	0.16%
110	58	Clk 4	0.25%
150	40	Clk 4	0.16%
300	81	Clk 3	0.16%
600	40	Clk 3	0.16%
1200	81	Clk 2	0.16%
2401	40	Clk 2	0.16%
3600	ad	Clk 1	0.22%
4800	81	Clk 1	0.16%
7200	56	Clk 1	0.22%
9600	40	Clk 1	0.16%
19200	81	Clk 0	0.16%
38400	40	Clk 0	0.16%
56000	2c	Clk 0	0.80%
64000	26	Clk 0	0.16%

1.All divisors are in hexadecimal.

Table 6. Bit Rate Constants, CLK = 25 MHz (Sheet 1 of 2)

Bit Rate	Divisor[1]	Clock	Error
50	f3	Clk 4	0.06%
110	6e	Clk 4	0.02%
150	50	Clk 4	0.47%
300	a2	Clk 3	0.15%
600	50	Clk 3	0.47%
1200	a2	Clk 2	0.15%
2401	50	Clk 2	0.47%
3600	d8	Clk 1	0.01%
4800	a2	Clk 1	0.15%
7200	6c	Clk 1	0.45%
9600	50	Clk 1	0.47%

Table 6. Bit Rate Constants, CLK = 25 MHz (Sheet 2 of 2)

Bit Rate	Divisor[1]	Clock	Error
19200	a2	Clk 0	0.15%
38400	50	Clk 0	0.47%
56000	37	Clk 0	0.35%
64000	30	Clk 0	0.35%
76800	28	Clk 0	0.76%

1.All divisors are in hexadecimal.

Table 7. Bit Rate Constants, CLK = 30 MHz

Bit Rate	Divisor[1]	Clock	Error
110	84	Clk 4	0.13%
150	61	Clk 4	0.35%
300	c2	Clk 3	0.16%
600	61	Clk 3	0.35%
1200	c2	Clk 2	0.16%
2401	61	Clk 2	0.35%
3600	40	Clk 2	0.16%
4800	c2	Clk 1	0.16%
7200	81	Clk 1	0.16%
9600	61	Clk 1	0.35%
19200	c2	Clk 0	0.16%
38400	61	Clk 0	0.35%
56000	42	Clk 0	0.05%
64000	3a	Clk 0	0.69%
76800	30	Clk 0	0.35%
115200	20	Clk 0	1.38%

1.All divisors are in hexadecimal.

Table 8. Bit Rate Constants, CLK = 35 MHz (Sheet 1 of 2)

Bit Rate	Divisor[1]	Clock	Error
110	9a	Clk 4	0.23%
150	71	Clk 4	0.06%
300	e3	Clk 3	0.06%
600	71	Clk 3	0.06%
1200	e3	Clk 2	0.06%
2401	71	Clk 2	0.06%
3600	4b	Clk 2	0.06%

Table 8. Bit Rate Constants, CLK = 35 MHz (Sheet 2 of 2)

Bit Rate	Divisor[1]	Clock	Error
4800	e3	Clk 1	0.06%
7200	97	Clk 1	0.06%
9600	71	Clk 1	0.06%
19200	e3	Clk 0	0.06%
38400	71	Clk 0	0.06%
56000	4d	Clk 0	0.16%
64000	43	Clk 0	0.53%
76800	38	Clk 0	0.06%
115200	25	Clk 0	0.06%
12800	21	Clk 0	0.53%
134400	20	Clk 0	1.38%

1. 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, and Manchester are data encoding schemes used in various synchronous protocols. In NRZ, the signal condition represents the data type: high for logic 1 and low for logic 0. In NRZ- and NRZI-type 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. Figure 11 through Figure 13 illustrate encoding methods. The eight data bits are '0110010'.

Example 3

This example illustrates programming the DPLL at 128 kbps in NRZI mode using the internal clock with a system clock frequency of 35 MHz.

Divisor loaded into R/TBPR = 33 or 22h.
Value loaded into RCOR = 28h to enable the DPLL, NRZI framing, and select Clk 0.

Example 4

This example illustrates programming the DPLL in ×1 External Clock mode, with Manchester encoding.

Divisor loaded into RBPR = 01h to enable ×1 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, (that is, for 33-MHz operation, the data clock should be less than 2 MHz). Note that R/TBPR are 8-bit registers; therefore the largest divisor value is 255.

Use the following equation to compute the divisor value.

$$\text{Bit rate divisor} = \frac{\text{Frequency of external clock source}}{\text{Desired bit rate}} - 1$$

Figure 11. Data Encoding
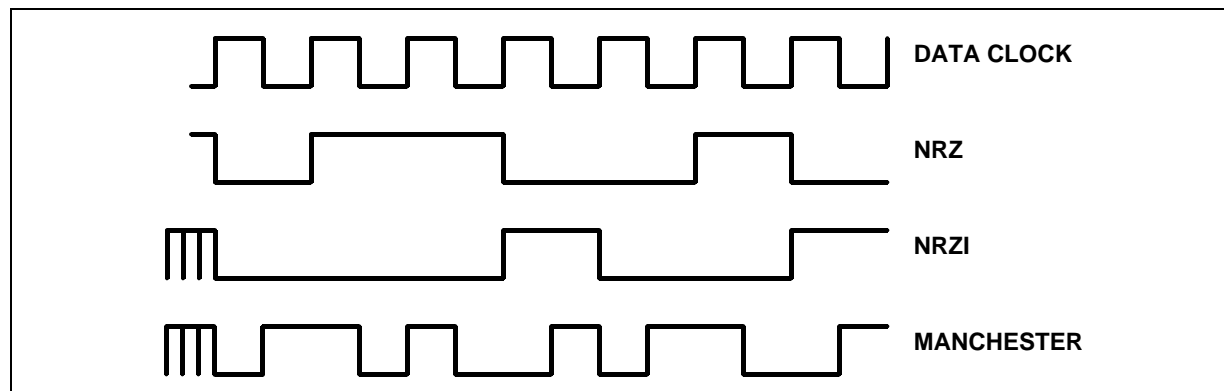
Figure 12. Transmit Data With External Clock In
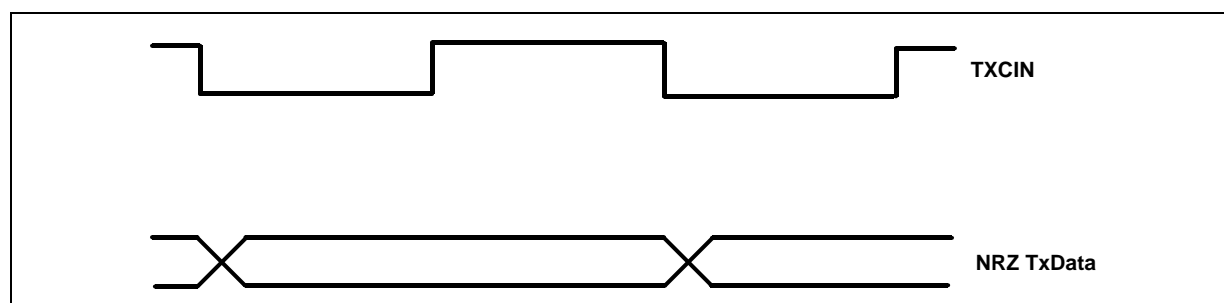
Note: When using the external receive clock in Receive mode, data is sampled on the low-to-high going edge of RXCIN.

Figure 13. Transmit Data With External Clock Out
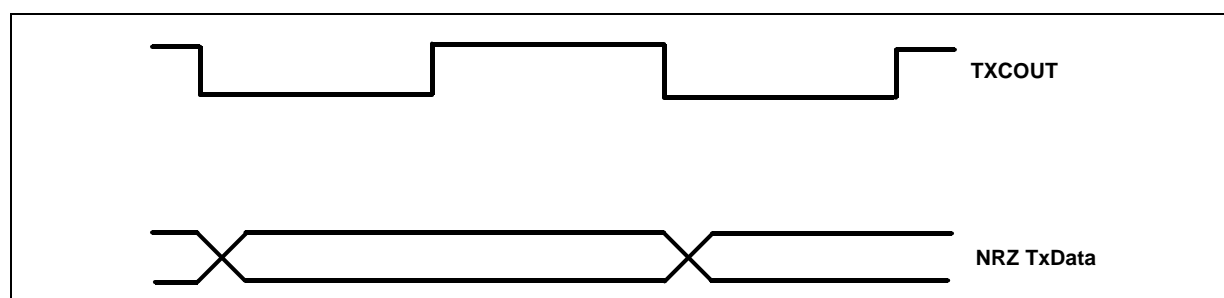
Table 9. Data Clock Selection Using External Clock @ 35 MHz

Bit Rate	External Clock Frequency	Divisor[1]
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	156.250 kHz	81
2401	156.250 kHz	40
3600	625.00 kHz	ef
4800	625.00 kHz	81
7200	1.250 MHz	ac
9600	1.250 MHz	81
19200	1.250 MHz	40
38400	1.250 MHz	1f
56000	1.250 MHz	15
64000	1.250 MHz	12
76800	1.250 MHz	0f
115200	2.00 MHz	10
128000	2.00 MHz	0f

1.All divisors are in hexadecimal.

5.6   Hardware Configurations

To demultiplex the A/D[15:0] 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–CD2401 applications. The common control lines (ADLD*, AEN*, DATDIR*, DATEN*) to the external devices are wire-OR'ed together. These pins are tristate, not open collector, but an external pull-up resistor (2.2–5.0 k¾) must be connected to each line to ensure logic 1 when no CD2401 is a bus master.

When no higher-priority alternate bus masters are present, a daisy-chain priority scheme can be implemented by wire OR'ing 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 CD2401. If a priority encoded scheme is required, the BR* signals must be prioritized externally and BG* signals routed to individual devices.

5.6.1   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 CD2401 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 CD2401 always drives all 16 data bits during a register read or a DMA write operation, regardless of the size of the actual transfer.

5.6.1.1   DMA Connections for the

Figure 14. DMA Connections for the CD2401
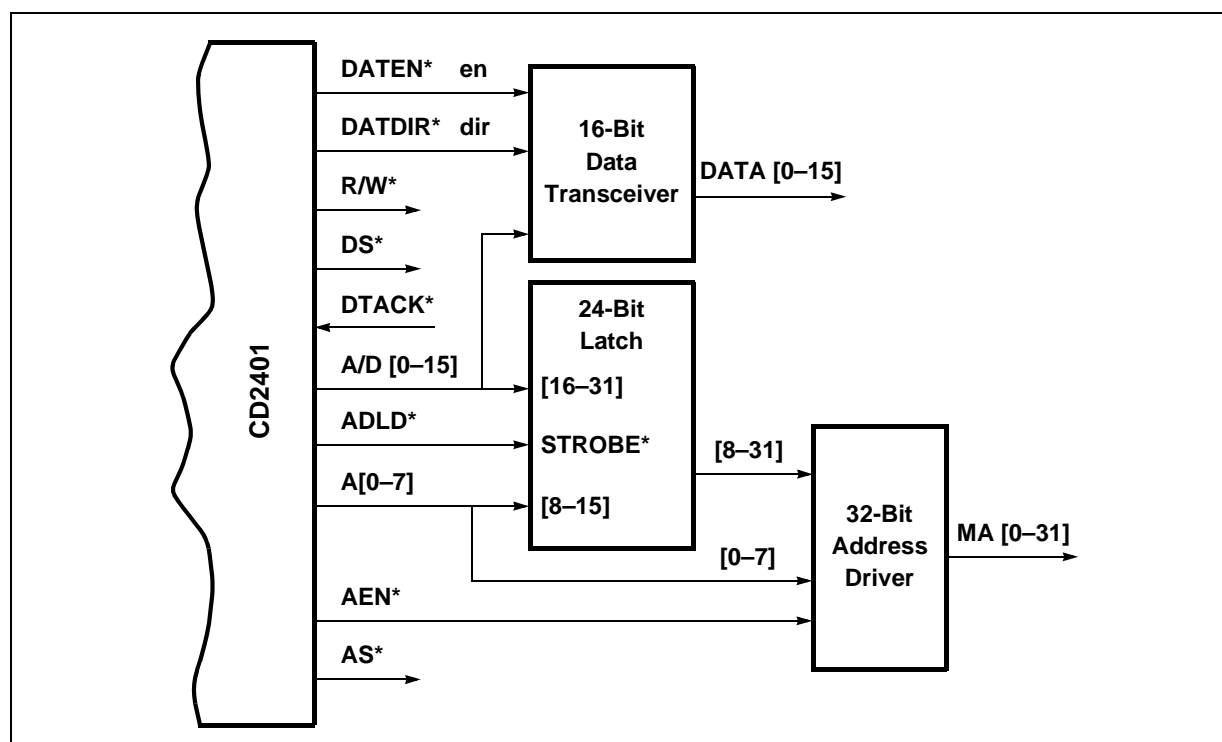
NOTES:
1. The 24-bit latch is *required*.
2. The 16-bit transceiver is *optional* depending on application.
3. The 32-bit driver is *optional* depending on drive requirements.

5.6.2   CD2401 as a DTE and DCE Interface[†]

Table 10 shows the recommended DTE (data terminal equipment) connections between the CD2401 and RS-232C, X.21, and X.21 bis[††] standard interfaces.

Table 10. DTE Connections

CD2401	RS-232C	X.21	X.21 bis (V.24)
RXD	BB	R	104
TXD	BA	T	103
RTS*	CA	C	105
CTS*	CB	I	106
DSR*	CC	–	107
TXCOUT/DTR*	–/CD	–	108.2
RXCIN	DD	S	115
TXCIN	DB	S	114
RXCOUT	DA	–	113
CD*	–/CF	–	109

Table 11 shows the recommended DCE (data communication equipment) connections between the CD2401 and RS-232C, X.21, and X.21 bis standard interfaces.

Table 11. DCE Connections

CD2401	RS-232C	X.21	X.21 bis (V.24)
RXD	BA	T	103
TXD	BB	R	104
RTS*	CB	I	106
CTS*	CA	C	105
DSR*	CD	–	108.2
TXCOUT/DTR*	DB/CC	S	114/107
RXCIN	–	–	–
TXCIN	DA	–	113
RXCOUT	DD	–	115
CD*	DA/–	–	–

---

†. Reference: CCITT 1988 Blue Book.

††. X.21 is completely different from X.21 bis.

6.0 Protocol Processing

6.1 HDLC Processing

6.1.1 Frame Check Sequence

FCS is a 16-bit standard computation used in HDLC and defined in ISO 3309. This FCS algorithm is the same used with the synchronous HDLC operation of the CD2401. The basic characteristics of the FCS are the following:

Accumulation: FCS computation starts after the opening flag and continues to the closing flag.

Polynomial: The standard polynomial is:

$$x^{**}16 + x^{**}12 + x^{**}5 + 1$$

Pre-load: The FCS 16-bit accumulator is preset to all '1's.

Transmit order: The FCS bits are identified as X15 to X0. The most-significant bit is X15, and is transmitted first. Thus, the first FCS character transmitted has bits X15–X8 in character positions D1–D8, respectively. The second FCS character has bits X7–X0 in character positions D1–D8, respectively.

Transmit polarity: Inverted.

Correct remainder: The receiver calculates the entire received frame, including the received FCS field. If the frame is received error-free, then the correct remainder in the FCS accumulation is 'F 0 B 8' (X15 is the leftmost bit).

The FCS can be individually enabled or disabled for the transmitter and receiver.

If enabled for the transmitter, the device appends the FCS on transmitted frames. If disabled, the device adds no FCS at the end of the frame.

If enabled for the receiver, the device computes the received FCS and reports the results. If the FCS buffer is enabled, the device includes the 2-byte FCS in the received data presented to the host. If disabled, the device does not test for received FCS.

6.1.2 HDLC Transmit Mode

The transmitter can be programmed to idle in either flag (01111110) or mark (continuous 1's) mode through the idle bit (COR3[3]). When Idle in Mark mode is selected, frame transmission can be programmed to be pre-pended by a programmable number of pad characters and 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 for Manchester encoding, the 00 character guarantees a transition every bit time, and the AA character guarantees exactly one transition per bit time.

If the transmitter is idle in Mark mode, frame transmission is started when data is made available to the transmitter, either by the TDR (Transmit Data register) or a DMA buffer. First, the programmable number of pad characters are transmitted, then the programmable number of flag characters. Data characters are then transmitted and a CRC value accumulated using each data character.

When end-of-frame status is passed to the CD2401 by the TEOIR or the ATBSTS/BTBSTS, and the remaining data is transmitted, the CRC and a closing flag are appended to the frame. If a new frame is available immediately, the correct number of opening flags are transmitted and data transmission starts. If data is not available, the line is returned to its idle condition.

If data underrun occurs, the CD2401 does not append a CRC, but aborts the transmission by sending eight continuous '1's, and then reverts to the idle condition. An underrun interrupt is generated, and if interrupt transfer is being used, the CPU provides an EOF response in TEOIR. If DMA Transfer mode is being used, the CD2401 discards DMA buffers until an EOF buffer is found; transmission then resumes from the next buffer. This ensures correct operation when a multiple buffer frame underruns.

When programmed in NRZI mode and idle in Mark mode, after the closing flag and the first eight '1's are transmitted, the transmit data line is sampled to determine if it is a logic high or low. If it is low, an extra '0' 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 is 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 '1's are transmitted, and then continuous flags and an underrun interrupt are generated.

6.1.3 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. The following two modes of address recognition are available:

- First byte of address field only (four possible matches available against RFAR1–4).

- First and second byte address field (two possible matches available against RFAR1–2 and RFAR3–4).

For the purposes of address matching, the Address Extension bit is not interpreted by the device. The address matching occurs on either the complete first byte, or the 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 EOF (End of Frame) interrupt is generated. The CRC can be either validated or ignored. If the CD2401 does not check the CRC, it is passed onto the host. A validated CRC can be discarded or passed onto the host for diagnostic purposes.

The next non-flag/abort character restarts the process; the current state of the receive process is visible to the CPU by 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 by 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 is discarded, and a clear-detect indication is passed to the CPU by the RISR. However, the channel remains in HDLC mode until modified by the CPU.

6.2 Async Processing

Data is transmitted according to the format options defined in the CORs. These options determine the character length, parity, and Stop bit length. New data sent from the host is transmitted in a continuous stream, unless one of the following occurs:

- Transmitter disabled — transmission terminated at the end of the current character until transmitter enabled.

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

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

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

- Send special character command from host — the current character is completed and the special character transmitted, after which normal transmission resumes.

6.2.1 Transmitter In-Band Flow Control

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

Transmit in-band flow control is enabled when the TxIBE bit (COR2[6]) is set to '1'. When TxIBE is set to '0', in-band flow control is disabled, the IXM bit (COR2[7]) has no meaning. The XON and XOFF characters are defined in SCHR1 and SCHR2.

When in-band flow control is enabled (TxIBE = 1) and an XOFF character is received, the channel stops transmission after the current character in the transmit shift register; the current character in the transmit holding register is transmitted. When IXM is set to '0', transmission restarts after an XON character is received. When IXM is set to '1', transmission restarts after any character is received.

The FCT Mode bit (COR3[5]) determines whether to pass the received flow control characters to the host. If FCT is '1', the characters are not passed to the host. If FCT is '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 CSR. The TxFloff (transmit flow off) and TxFlon (transmit flow on) bits (CSR[2:1]) are used.

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

TxFlon = 0 is normal. TxFlon = 1 indicates 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.

6.2.2 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 are transmitted by setting the SndSpc bit (STCR[3]) to '1'.

The CSR contains status bits RxFloff (receive flow off) and RxFlon (receive flow on), which are used for receiver in-band flow control.

RxFloff = 0 is normal. RxFloff = 1 indicates the channel has requested that the remote stop transmission. This bit is reset to '0' when the channel requests that the remote restart 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 indicates that the channel has requested that the remote restart 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.

6.2.3 Out-of-Band Flow Control

Receive out-of-band flow control is enabled when the CtsAE bit (COR2[1]) is set to '1'. In this mode, character transmission begins only after the CTS* pin is active (low). In asynchronous transmission if CTS* goes inactive (high) after transmission starts, the channel stops 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 stops transmission after the current frame. In either case, transmission restarts after CTS* goes active.

The CD2401 can automatically flow control the remote device by the DTR* pin. This mode is selected by setting a non-zero DTR* threshold in COR5; when both thresholds in COR4 and COR5 are exceeded, the CD2401 sets the DTR* pin high. When the data in the FIFO falls below the DTR* threshold, the DTR* pin is automatically driven low.

Each channel of the CD2401 has four pins that can be used either as a modem control or general-purpose input/output pins. The modem signal names assigned to these four pins were selected to provide an easy reference for system designers. In fact, they are all simply general-purpose inputs and outputs (if automatic out-of-band flow-control is not used) that can be individually controlled by the Modem Signal Value register(s). Since the pins are general-purpose, system designers can opt to connect the pins in a manner to suit the application.

However, when the system software design employs automatic out-of-band flow control with the pins, then the signal naming convention no longer holds true in some cases, depending on whether the device is used as DCE or DTE. In this case, it is best to think of the pins in terms of their actual uses within the CD2401 and connect them accordingly, without regard to their names; the RTS* and CTS* pins are associated with transmitter, and the DTR* and DSR* pins are associated with the receiver. Table 12 on page 72 shows the recommended signal hookup if automatic out-of-band flow control is desired.

Table 12. Recommended Signal Connection

Mode		CD2401 Pin Name	Out-of-Band Flow Control
DCE	DTE		
CTS		DTR	Signal remote to transmit.
RTS			Not implemented in this direction.
	RTS	RTS	Request remote permission to transmit.
	CTS	CTS	Enable transmitter.

For example, if the CD2401 is designed to be DCE and automatic out-of-band flow control is desired, connect the DTR pin to the remote CTS input. If the CD2401 is to be used as the DTE side, then connect the CD2401 CTS output to the remote CTS input.

Note that if automatic out-of-band flow control is implemented, the activity of the DTR and DSR pins do not implement the function assigned to those signal names by the signaling conventions of the CCITT and other standards organizations. These names would only apply to these pins if they are under program control and not under automatic CD2401 control. In fact, the 'DTR' function enables the modem to go on- and off-line, depending on the state of the pin. If automatic control is used, then DTR goes inactive when the receive FIFO reached the programmed threshold, thus causing the modem to drop the connection (carrier) to the remote, which would not be the correct function based on the state of the receive FIFO.

6.2.4 Line Break Detection and Generation

A line break on the receiver occurs when the input at the RXD (receive data) pin is all zeros (low) for at least one full character time. This is indicated when the Break bit (RISRl[0]) is set to '1'.

Line-break generation out of the transmitter is possible when the ETC bit (COR2[5]) is set to '1'. A line break is generated when the output at the TXD pin is all zeroes (low) for at least one full character time.

Line breaks can be transmitted by embedding certain sequences in the data stream (defined in Table 13). These sequences are valid for transmitting breaks only if ETC is set to '1'. The embedded sequences to transmit a BREAK are listed in Table 13.

Table 13. BREAK Sequencing

Sequence	Function
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, use the insert delay sequence. The inserted delay is xx, where xx is a binary number. The delay is xx times the 'tick' set by the TPR. The minimum period of TPR should be one millisecond. If the insert delay sequence is not preceded by a Send BREAK sequence, there is an inserted delay of all '1's (high) on the output for duration xx.
00h–83h	Stop BREAK – This must follow a Send BREAK or Insert Delay sequence.
00h–00h	Send NUL – If the user needs to send a NULL character and ETC is set to '1', the user can embed 00h–00h to send one NULL character. If there are less than 8-bits per character, the user can also send a NULL character by 'sending' 80h.

Note: In addition to insert delay, a 'break' can also be increased beyond one full character by transmitting more than one 'Send BREAK' sequence at a time.

6.2.5 Special Characters Special Character Transmission

Selected special characters can be sent preemptively by setting the SndSpc bit (STCR[3]). The CD2401 channel acknowledges the command by clearing the STCR. Along with the SndSpc bit, the host needs to setup the three SSPC bits (STCR[2:0]) to select which character is to be sent.

When the host commands a special character transmission, the channel completes transmitting any characters in the transmit shift and transmit holding registers, and then transmit the special character sequence. Any other characters awaiting transmission in the FIFO or through DMA are transmitted after the special character.

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

Table 14. SSPC[x] Settings

SSPC2	SSPC1	SSCP0	Function
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

6.2.6   Special Character Recognition and Range

Special character recognition is enabled when the SCDE bit (COR3[4]) is set to '1'. The special characters are programmed in the SCHRs, and are the same characters used for the transmitter.

If the FCT bit (COR3[5]) is set to '1', the channel processes the flow control characters and discards them. If FCT is set to '0', the received flow control characters are processed and passed onto the host by an exception interrupt.

In the event of an error (framing and/or parity) in a received character sequence, the channel does not interpret this character as a special character. But, if an overrun condition occurred after a special character was detected, the new character is lost and the overrun status is set. In this condition, the CD2401 gives both an overrun exception and a special character recognition status.

6.2.7   Special Character Range

The SCRs define an inclusive range for special character recognition in Asynchronous mode. This is 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 SCHR1.

Special characters and range detection is through three special character detect bits (RISRl[6:4]). The function of the SCdet[x] bits is listed in Table 15.

Table 15. SCdet[x] Settings

SCdet2	SCdet1	SCdet0	Function
0	0	0	No special characters and 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	The hex value of the receive character is within the range: SCRl $\leq$ receive character $\leq$ SCRh.

6.2.8   UNIX® Support Features

COR6 provides several functions required by UNIX TTY drivers to further reduce the amount of character-by-character processing the CPU is required to perform. Separate receive and transmit bits are provided to perform CR/NL (carriage return/new line) 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 and character error conditions. Break conditions can be handled in the normal way (by a receive status interrupt), the condition can be discarded, or the break can be translated to NULL (00) and passed as normal data to the CPU. Parity and framing errors can either be handled as normal (by 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[6]) 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 by the LNext register. The 'strip' feature (COR7[7]) strips the eighth bit off each error-free received character. This has no effect on the transmitted data. Figure 15 on page 76 through page 78 shows the exact order of the CD2401 character processing steps and the receive character processing flow for Async mode in flowchart format.

Figure 15. CD2401 Async Receive Character Processing
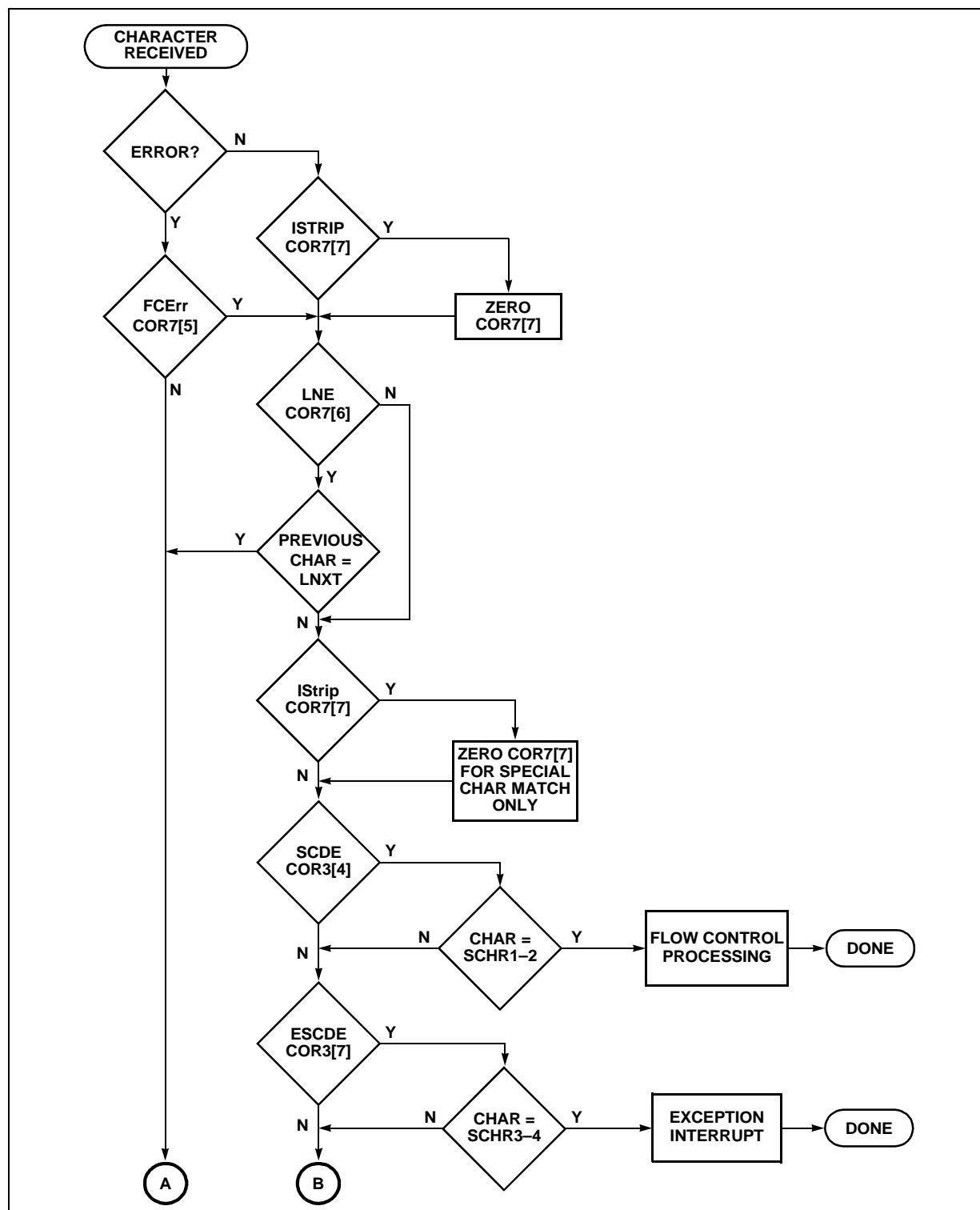
Figure 15. CD2401 Async Receive Character Processing (Continued)

A

CHAR = BREAK — N

Y

Process Break Options		COR6
IgnBrk	NBrkInt	Action
0	0	Exception interrupt
1	0	Discard character
1	1	Replace with 0

DONE

Process Parity Options			COR6[3:0]
ParMrk	INPCK	ParInt	Action
0	0	0	Exception interrupt
0	0	1	Replace with 0 (NULL)
0	1	0	Ignore error
0	1	1	Discard character
1	0	1	Replace with FF + 00 + Char

DONE

Figure 15. CD2401 Async Receive Character Processing (Continued)

intel.
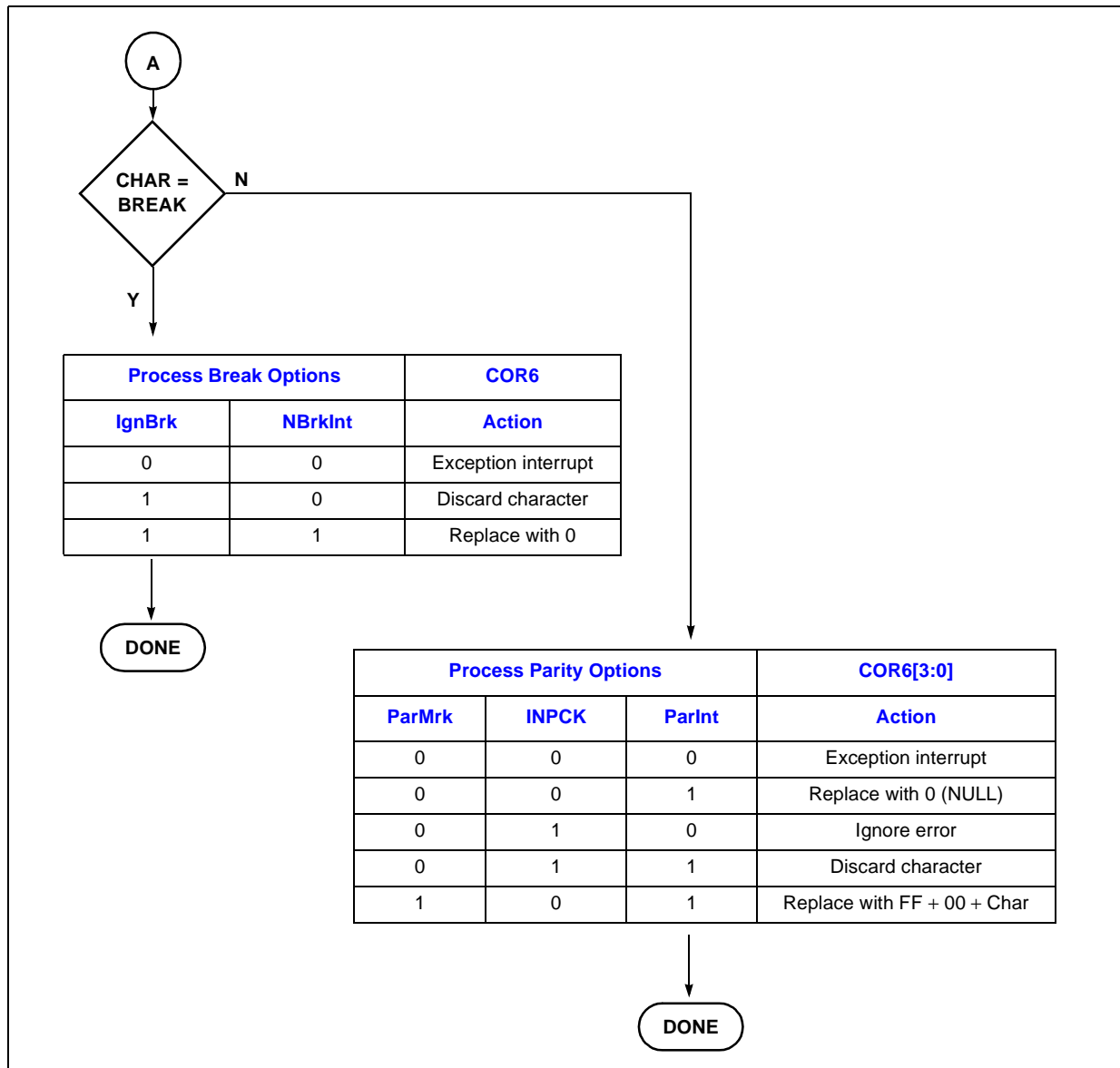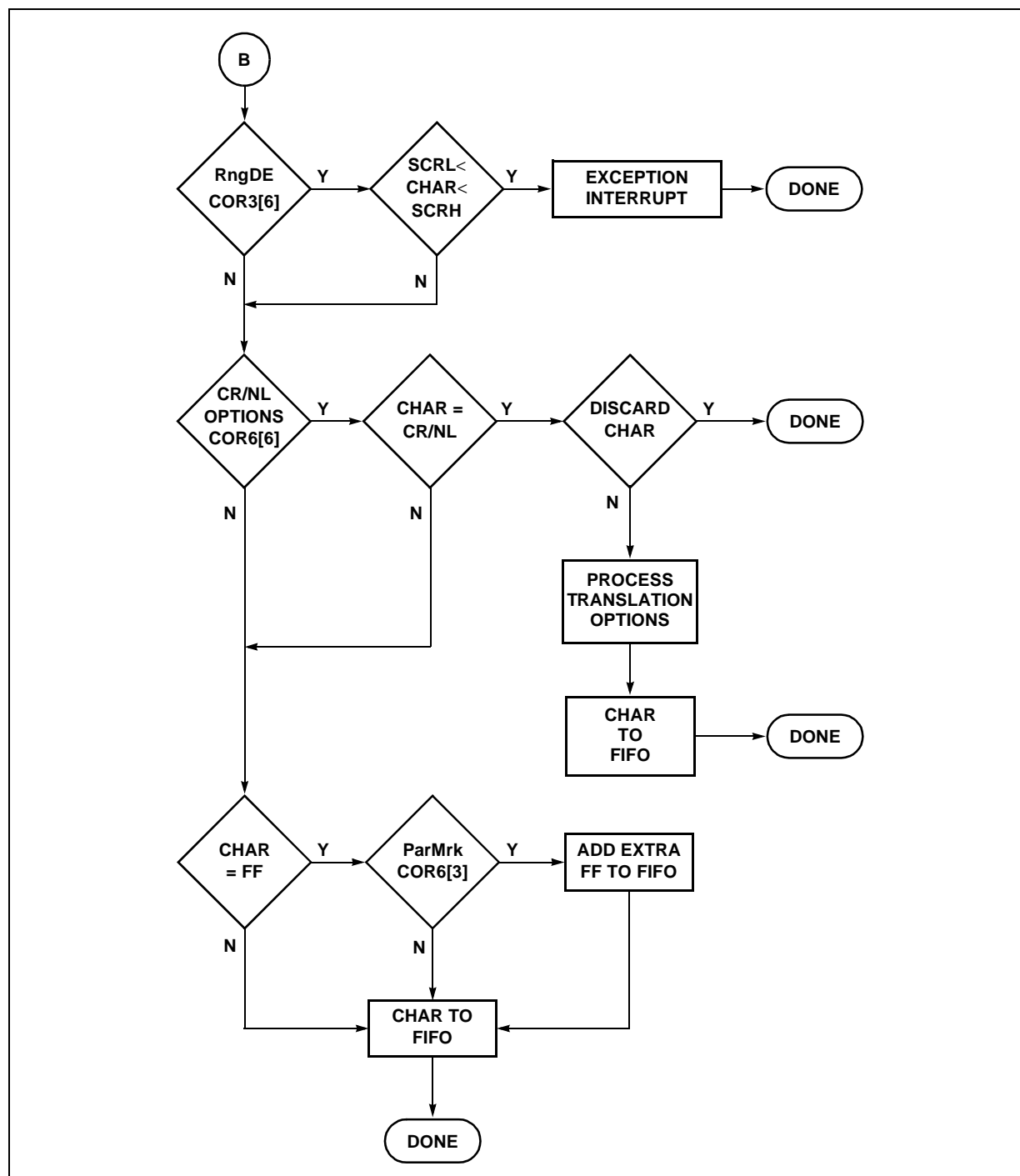
6.3 Bisync Processing

In both transmit and receive, the CD2401 interprets the first characters of data to determine the type of frame and compile the corresponding BCC. The host uses COR1 to program parity options and character length, and COR2 to program the character set (ASCII or EBCDIC) and determine whether to use CRC-16 or LRC.

6.3.1 Bisync Transmit Processing

The CD2401 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 starts; any required leading pads are sent, followed by a SYN pair and the CPU-supplied data. The CD2401 monitors the transmit data to determine frame type and compute the correct BCC, thus eliminating unnecessary characters from the calculation. If SYN sequences are embedded in the data supplied by the CPU, they are transmitted — but excluded — from the BCC calculation.

If a frame transmission is aborted by the STCR, an EOT and a trailing pad are transmitted and the line returned to its idle state. A frame is terminated normally when an EOF indication is passed to the CD2401, either in TEOIR or in 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 returns to the idle state.

6.3.2 Bisync Receive Processing

After initialization, the receiver starts in Synchronous Hunt mode and discards 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). If required, the BCC is compiled excluding any characters that should not be part of the calculation. When a frame terminating condition is detected and if a BCC was accumulated, it is checked and the EOF information is passed to the CPU by the RISRs. If the frame is terminated with an ENQ condition, the BCC is not checked and an abort indication is passed to the CPU by the RISRs.

An extra-frame-termination process is available by programming an extra-frame-termination character into COR6. When this character is detected, the receive frame is terminated immediately and no BCC is computed. Following an initialize channel command, COR6 is set to the value of DLE (10 hex) by the internal code; the user can alter this to any other value. To detect the condition where the frame termination character was corrupted on a non-transparent line, COR6 can be programmed to the idle condition, FF hex. To use this on a transparent line, the data should not equal FF; if the value in COR6 is preceded by the DLE character, it does not cause frame termination.

Note: This feature is not available in revisions prior to Revision 'H'.

Short-Frame Processing

Short frames in Bisync mode are generally terminated with minimum two bytes of XOFF. The CD2401 reports these frames as follows:

Frame = SYN SYN STX ENQ FF FF

Reported as:
        receive CRC error
        receive abort
STX and ENQ are passed to host as data.

6.3.3   CRC Calculation in Bisync Mode

In Bisync mode there are several conditions where the CD2401 varies the way it computes the FCS or BCC/CRC. Which data is included in the current frame depends on how the previous frame ended and the transparency in the middle of a frame.

The following illustrations show the point in the current frame where FCS computation begins based on how the previous frame ended. From the start of a frame when the previous frame ended in DLE-ITB, the following data streams have CRC computed as:

SYN	SYN	Data	Data
SYN	SYN	STX	Data
SYN	SYN	SOH	Data
		CRC Begins	

From the start of a frame when the previous frame did not end in DLE-ITB, the following four data streams have CRC computed as:

SYN	SYN	DLE	STX	Data
SYN	SYN	DLE	SOH	Data
				CRC Begins

SYN	SYN	SOH	Data
			CRC Begins

Transition from a non-transparent to a transparent frame:

SYN	SYN	SOH	Data	Data	DLE	STX	Data
			CRC			CRC	

Within a non-transparent frame:

Data	Data	SYN	Data	Data	SYN	SYN	Data	Data
CRC			CRC				CRC	
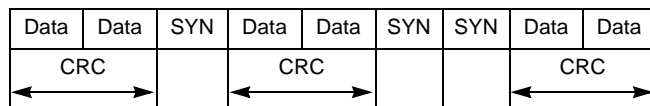
At the end of a non-transparent frame:

Data	Data	ETX	BCC	BCC
Data	Data	ETB	BCC	BCC
Data	Data	ITB	BCC	BCC
CRC Ends				
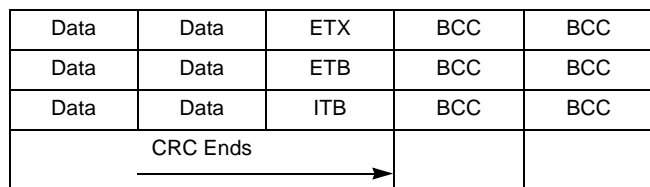
At the end of a transparent frame:

Data	Data	DLE	ETX	BCC	BCC
Data	Data	DLE	ETB	BCC	BCC
Data	Data	DLE	ITB	BCC	BCC
CRC			CRC		
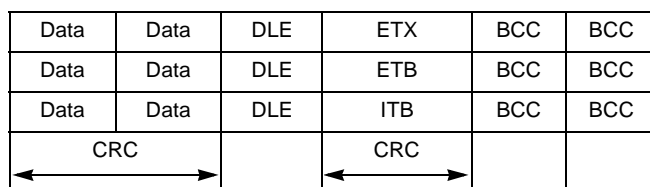
Note: For the transparent bisync frames (receive only), the first DLE is not included in the CRC calculation (except DLE-SYN, where the SYN is also not included.

Transparency Support

In transmit, the CD2401 generates CRC according to the previous illustrations, however the host must manually insert the DLE in the data stream.

On receive, the CD2401 checks CRC according to the previous illustrations and strips the DLE from the data stream.

6.3.4   BCC Computation Formulas

In Bisync mode, the CD2401 can use either CRC-16 or LRC. The mode used is determined by the setting of the LRC bit (COR2[7]). CRC-16 uses the polynomial in the following equation, preset to all zeroes.

$$x^{**}16 + x^{**}15 + x^{**}2 + 1$$

LRC performs a parity check on each bit of each character in the frame in a longitudinal or 'vertical' manner. For example, in the following 3-byte frame, LRC with even parity is computed as:

```
Character 1:  10010001
Character 2:  11000001
Character 3:  11000001
LRC:          10010001
```

Note: If parity is used, parity is computed/checked on each character, but the LRC of the parity bits is not checked.

6.3.5 Receive State Table

Table 16 on page 82 and Table 17 on page 83 show the bisync state transitions, within bisync receive microcode character processing, and the definition of the states.

Table 16. Bisync Receive State Transition

Character	BGETHDR 0	BESCF 1	BESCL 2	BGETXT 3	BBCC 4	BBCC2 5	BWEOT 6	BXP 7
SYN	0	7	7	3	5 c	c[1]	6	7 c
DLE (RxITB = 0)	1 i	6	7 c	2 c	5 c	c[a]	6	2
DLE (RxITB = 1)	2 i c	6	7 c	2 c	5 c	c[a]	6	2
ENQ (RxITB = 0)	6	Hunt	Hunt	Hunt	5 c	c[a]	6	7 c
ENQ (RxITB = 1)	Hunt	Hunt	Hunt	Hunt	5 c	c[a]	6	7 c
STX	3 i[2]	7 i	7 c	3 c	5 c	c[a]	6	7 c
SOH	3 i[b]	7 i	7 c	3 c	5 c	c[a]	6	7 c
EOT (RxITB = 0)	6	Hunt	Hunt	Hunt	5 c	c[a]	6	7 c
EOT (RxITB = 1)	3 i	Hunt	Hunt	Hunt	5 c	c[a]	6	7 c
ETX (RxITB = 1)	Hunt	6	4 c 0-RxITB	4 c 0-RxITB	5 c	c[a]	6	7 c
ETB (RxITB = 1)	Hunt	6	4 c 0-RxITB	4 c 0-RxITB	5 c	c[a]	6	7 c
ITB	6	6	4 c 1-RxITB	4 c 1-RxITB	5 c	c[a]	6	7 c
ETX (RxITB = 0)	6	6	4 c 0-RxITB	4 c 0-RxITB	5 c	c[a]	6	7 c
ETB (RxITB = 0)	6	6	4 c 0-RxITB	4 c 0-RxITB	5 c	c[a]	6	7 c
PAD	6	6	7 c	3 c	5 c	c[a]	Hunt	7 c
COR6 [3] (RxITB = 0)	6	6	7 c	Hunt	5 c	c[a]	6	Hunt
COR6 [c] (RxITB = 1)	3 i[b]	6	7 c	Hunt	5 c	c[a]	6	Hunt
None of the above (RxITB = 0)	6	6	7 c	3 c	5 c	c[a]	6	7 c
None of the above (RxITB = 1)	3 i[b]	6	7 c	3 c	5 c	c[a]	6	7 c

1. In the BBCC2 state: if RxITB = 1, go to state BGETHDR, else go to SYN Hunt state.
2. If RxITB = 1, calculate BCC, else do not calculate BCC.
3. The special frame termination character is activated if COR6 is not equal to x'10. If COR6 = x'10, disregard these two rows.

Key:

Hunt = Go to SYN Hunt state
i = initialize BCC
c = calculate CRC
0 = clear state flag, for example 0-RxITB clears the RxITB flag
1 = set state flag

Note: When returning to the SYN Hunt state, RxITB is cleared.

Table 17. Description of States

Number	Name	Description
0	BGETHDR	SYN-SYN received, looking for next non-SYN character.
1	BESCF	The first DLE after opening SYNs received.
2	BESCL	Received DLE, not the first for this frame.
3	BGETXT	The first non-SYN character after opening SYNs received. The first character is either: STX, SOH, ETX (if RxITB = 1), or ETB (if RxITB = 1).
4	BBCC	A proper end-of-text received, the channel is ready for the first BCC character. Proper protocol: ETX, ETB, or ITB in non-transparent mode; DLE-ETX, DLE-ETB, or DLE-ITB in transparent mode.
5	BBCC2	Received the first BCC character, CD2401 is ready for the second BCC character.
6	BWEOT	Continue receiving in this state until a PAD character is detected.
7	BXP	Transparent receive: look for escape DLE, else keep receiving.

NOTES:
1. The SYN Hunt state not shown. The SYN Hunt state consists of two substates that look for a received SYN-SYN pattern. Upon receiving SYN-SYN, the state moves to BGETHDR.
2. Conditions for the state table:
   CRC-16 enabled
   EBCDIC character set

6.4 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 steady-state conditions (all '1's, all '0's, alternating '1's and '0's) 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. To enable detection of steady-state conditions, the SSDE bit (COR3[4]) must be set.

6.4.1 X.21 Transmit

To minimize CPU intervention in the transmit direction, a modified version of the ETC is used. ETC mode is controlled through COR2. When enabled, ETC mode 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 CD2401 as normal transmit data, in the format shown in Table 18.

Table 18. ETC Byte Sequence

Byte 1	This byte must be 80 hex to indicate the start of a command sequence.
Byte 2	This byte indicates the required state of the C lead: 00 = sets the C lead off 01 = sets the C lead on 02–FF= reserved
Byte 3	This byte is the data character to transmit as is with no parity added. If parity is required, it must be included in the byte (that is, to transmit '+' with odd parity, load the value AB).
Byte 4	This byte is a count of the number of times to transmit the character. If the count is '0', the character is continuously sent until new data is made available (this is the normal mode for the steady-state conditions). In this case, when the count is '0', the CD2401 always sends the data a minimum of three times, even if new data is made available before all three characters are sent.

6.5   X.21 Receive

In receive, the CD2401 validates the steady-state conditions, passing just the change-of-state information to the CPU. Steady-state conditions validated are:

- all '1's
- all '0's
- alternating '1's and '0's
- SYN
- 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 for DTE, or 'C' lead for DCE).

For X.21 mode, the SSDE bit (COR3[6]) must be set to enable the detection of the steady-state conditions; this enables the detection of the all '1's, all '0's, and the alternating '1's and '0's conditions with a stable value on the CTS* pin. When the SSDE bit is set, the StrpSYN and SCDE bits (COR3[5:4]) can also be set if required (if SSDE is not set, then the StrpSYN and SCDE bits have no effect).

Note: The SglSYN bit (COR3[7]) must be '0' because X.21 mode always requires two SYN characters for synchronization.

Enable the SSDE and StrpSYN bits to prevent SYN characters from entering the receive data FIFO. When set, the StrpSYN bit treats SYN characters the same way as steady-state conditions (that is, when two valid SYN characters are detected, a receive special character interrupt is generated and the next 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 the same way as steady-state conditions. When detected for two consecutive character times, a special character detect interrupt is generated and the next repetitions of the same character are stripped from the receive data (for example, to detect the 'BEL off' condition for a DTE incoming call, then strip that repetition until the next state change). Character synchronization must be achieved before SCHR1–3 can be detected.

In certain phases of X.21 call setup, there is no character synchronization. When a data change occurs in a non-character synchronous phase, a partial character can be detected before the steady is detected or character sync is achieved. In these conditions, the partial character is passed to the host as normal data.

Note: The CD2401 passes all data received to the host before it receives SYN characters.

Example

Assume the SSDE, StrpSYN, and SCDE bits are set, that is the CD2401 detects steady-state conditions, strip SYN characters, and special characters from the incoming data. Under these conditions, the following data stream:

Incoming data:

SS	SS	Junk	SS	SS	SYN	SYN	Bell	Bell	Data	Data	SS	SS
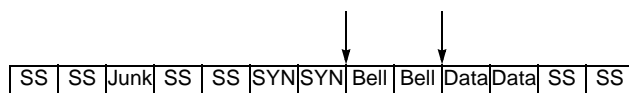
Where SS = steady-state condition.

Note: A SYN character detect interrupt is generated after the second SYN. A special character detect interrupt is generated after the second Bell.

Incoming data is passed to host as:

Junk	Data	Data							
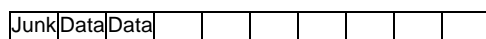
6.6   Extended X.21 Mode

Proper selection of options can extend the X.21 mode for some synchronous applications. Extended X.21 mode does not perform a true programmable sync mode, because the Extended-X.21 mode keeps passing characters to host though it is in SYN Hunt mode.

6.6.1   Extended X.21 Transmit

Without the ETC bit (COR2[5]) enabled, data supplied to the channel is transmitted unaltered. When no data is available in the transmit FIFO, the ASCII SYN character (16) is sent to fill idle time; the normal operating mode in X.21. If a different character is required, two methods are available.

First, the idle character can be supplied as normal data; the drawback is that host intervention is required to either fill the FIFO or supply a new DMA buffer periodically while there is no real data to send. The second method is to use the ETC to repeat the required data pattern until transmit data is available. The ETC command functions as shown in Table 19 (COR2[5] is set).

When an 80 hex character is encountered in the transmit data, it and the following three characters are treated as a special command. The byte format is:

Table 19. Byte Format — ETC Bit Set

Byte 1	This byte must be equal to 80 hex to start a command sequence.
Byte 2	This byte indicates the required state of the RTS* pin. 00 = the RTS* pin is set inactive (high) 01 = the RTS* pin is set active (low) 02–FF= reserved
Byte 3	This byte is the required character for transmission. It is sent as an 8-bit character without parity (any required parity value is included by the host).
Byte 4	This byte is the number of times the above character should be sent. If set to '0', the character is sent until new data is supplied. In this case, the CD2401 always sends the data a minimum of three times, even if new data is made available before all three characters are sent.

To idle in mark with the RTS* line off, write the command '80 00 FF 00'. To send the character '80' while the ETC command is enabled and the RTS* pins asserted, write the sequence '80 01 80 01'.

Note: This feature is not available in revisions prior to revision 'H'.

6.6.2 Extended X.21 Receive

In Receive mode, the SYN character can be programmed in COR6 to be any required value. When the channel is initialized, COR6 is initially set to the value '16' by the internal code; host software can reprogram COR6 to the required SYN value after channel initialization is complete. If a parity mode is enabled, the parity bit should not be set in COR6. Synchronization can be achieved with either a single- or double-SYN pattern; this is controlled by the SglSYN bit (COR3[7]).

Since the channel does not look for an end-of-frame delimiter, the host must determine when to end the frame and/or to terminate the buffer (if DMA is used) with software. It must also detect when to start another frame.

Extended X.21 mode still passes received data to host before and/or after it receives the SYN character(s). The host can discard all the junk characters before the SYN character(s) whenever the device detects SYN character(s) and generates an interrupt.

If the SSDE bit (COR3[4]) is not set, Extended X.21 mode does not strip SYN characters and special characters from the incoming data stream.

Note: Strip SYN and SCDE are functional only if the SSDE is set.

Example 1: (Two SYN mode)

Incoming data:

Junk	Junk	SYN	SYN	Data	Data	Data	Data	Data	Data	Data	Data
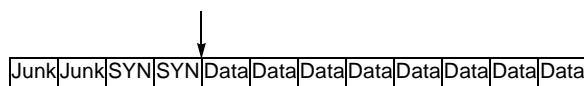
Note: A SYN character detect interrupt is generated after the second SYN.

Data is passed to the host:

Junk	Junk	SYN	SYN	Data	Data	Data	Data	Data	Data	Data	Data

Example 2: (Single SYN mode)

Incoming data:

Junk	Junk	SYN	Data	Data	Data	Data	Data	Data	Data	Data	Data
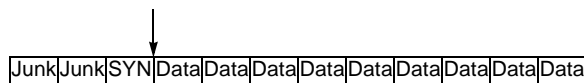
Note: A SYN character detect interrupt is generated after the SYN.

Data is passed to the host as:

Junk	Junk	SYN	Data	Data	Data	Data	Data	Data	Data	Data

6.7 Non-8-bit Data Transfers

In Asynchronous mode it is possible to transmit and receive less than 8 bits per character. There can be 5, 6, 7, or 8 bits per character.

For HDLC mode, 8-bits per character are always transmitted. The CD2401 transmits only byte-aligned frames. The CD2401 receives 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 and EOF bits (RISRl[2, 6]) are set.

7.0   Programming Examples

This section provides some examples of CD2401 programming. Included are examples of Global and Per-Channel initialization, and two interrupt service routines. The code was written in Borland® Turbo C++.

Figure 16. Initialization Sequence for the CD2401
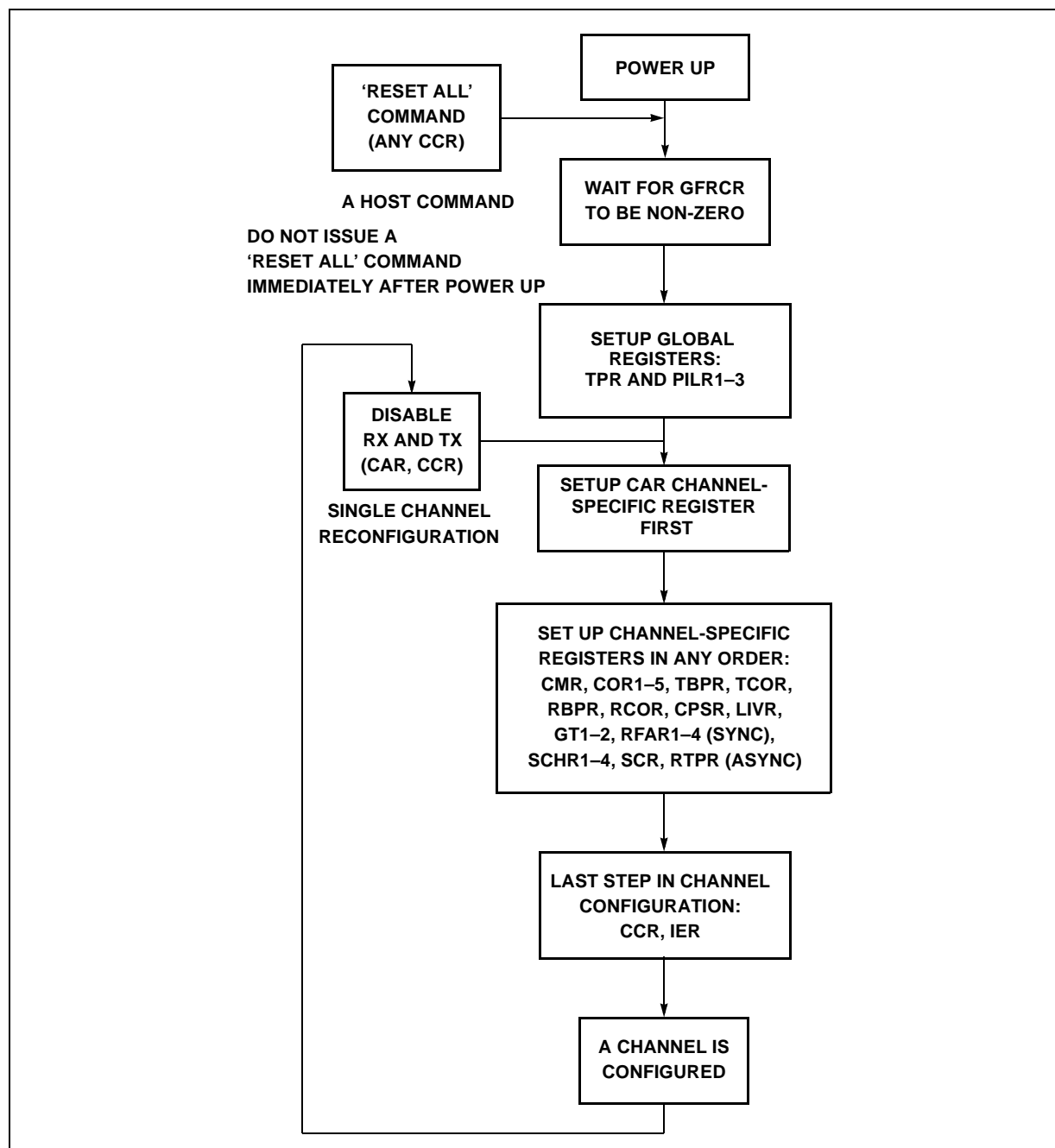
7.1   Global Initialization

The following code segment is an example of global initialization. The host waits for a hardware reset, determined by a non-zero value in the GFRCR. A 'Reset All' command is sent to the CD2401 by the CCR. The CD2401 internal processor will put a non-zero value into the GFRCR when initialization is complete.

It is a good practice to manually clear the GFRCR before issuing the 'Reset All' command, as it takes a finite amount of time for the internal processor to detect the presence of the command and clear the register. If the host is sufficiently fast, it might read the GFRCR before the command execution commences and incorrectly assume the command is complete. Note, unlike all other CCR commands, the clearing the CCR is not an indication of 'Reset All' command completion; clearing the CCR is a normal process of the internal register reset operation during 'Reset All' command execution, and occurs long before the reset operation is complete. When the GFRCR is loaded with the firmware revision number, the command is complete.

The PILRs should be loaded with the value that will appear on the seven address lines (A[6:0]) during interrupt acknowledge cycles. The TPR loads the dividing counter that provides input to each of the other timers in the CD2401. The DMA Mode and Bus Error Count registers are used in DMA modes only. After the global portion is done, the Per-Channel registers need to be initialized. Transfers and interrupts should be enabled after all other initialization is complete.

```
// Global Initialization
    while( !inportb( GFRCR ) )                  // wait for hardware reset
                 ; // wait
    outportb( GFRCR, 0x00 );                    // manually clear GFRCR
    outportb( CCR, RESET_ALL );                 // Reset command
    while( !inportb( GFRCR ) )                   // wait for reset command
                                                ; // wait
    outportb( PILR1, 0x02 );                    // Priority Interrupt
    outportb( PILR2, 0x04 );                    // Level registers
    outportb( PILR3, 0x06 );
    outportb( TPR, 0x40 );                      // Set timer prescale
    outportb( BERCNT, 0 );                      // Bus error count
    outportb( DMR, 0 );                         // DMA mode — 16-bit

// per-channel initialization

    for( i=0; i<4; i++ ) {
            outportb( CAR, i );                 // set channel number
            init_chan( cor, bpr );              // initialize channel
            outportb( CMR, MODE );              // set protocol/DMA or Intr
            outportb( CCR, INIT_CH | EN_RX | EN_TX );
            while( inportb(CCR) )
                    ; // wait
            outportb( IER, TX_DATA|RX_DATA ); // enable interrupts
    }
```

7.2   Async Interrupt Setup Example

This section contains a code example for an asynchronous channel running at 19,200 bps, with 8 bits/character, 1 Stop bit, and no parity. The sample code enables in-band flow control and implied XON mode. This code assumes that the proper channel is set by the CAR.

```
        outportb( LIVR, 0x40 );
        outportb( RCOR, 0 );  // Receive clock option
```

```
outportb( RBPR, 0x81 );                    // Baud Rate divisor
outportb( TCOR, 0 );                        // Transmit clock option
outportb( TBPR, 0x81 );                    // Baud Rate divisor

outportb( CMR, ASYNC );                    // Async Mode, interrupt

outportb( COR1, PARIGN | CHAR8 );          // 8 bit chars, no parity
outportb( COR2, IXM | TXIBE );             // in-band flow,implied XON
outportb( COR3, STOP1 | FCT );             // 1 stop, flow control
outportb( COR4, thresh );                  // FIFO threshold
outportb( COR5, 0 );
```

7.3   HDLC DMA Channel Setup Example

This per-channel initialization code example is for the HDLC protocol at 38.4 kbps with NRZI encoding. The setup specifies two extra opening flags before frames, no address matching, and that DMA transfers should be used.

```
outportb( LIVR, 0x30 );                    // Set interrupt vector

outportb( RCOR, DPLL_NRZI );               // Receive clock option
outportb( RBPR, 64 );                       // Baud rate divisor
outportb( TCOR, 0 );                        // Transmit clock option
outportb( TBPR, 64 );                       // Baud rate divisor

outportb( CMR, RX_DMA | TX_DMA | HDLC );   // Mode register
outportb( CPSR, CPSR_CRC_V41 );            // CRC polynomial select

outportb( COR1, NO_ADDR | FLAG_2 );        // No address matching,
outportb( COR2, CRC_V41 );                 // 2 opening flags
outportb( COR3, 0 );
outportb( COR4, thresh );                  // FIFO threshold
outportb( COR5, 0);
```

7.4   Receive DMA Interrupt Service Routine

The following code example shows an interrupt service routine for the CD2401 in DMA mode. The buffer class array 'ib[ ]' is used for notational convenience, and its exact implementation is user-defined. The upper ( ) and lower ( ) functions should return the upper and lower 16 bits of the DMA address for the current buffer segment. The 'nxt_buf( )' accesses the next segment.

If the system uses separate interrupt handlers for receive, transmit, and modem interrupts, the channel number can be obtained from the lower two bits of the Interrupt register (RIR, TIR, or MIR). Otherwise, first use LIVR to determine the type of interrupt. Receive 'Good Data' interrupts should not occur during DMA transfers. The normal exception is when an end-of-frame is received.

The DMABSTS register shows which buffer the CD2401 expects to use next. Fill the descriptor registers for that buffer, including the 2401own bit and return. The last access to the CD2401 during the service routine is the REOIR.

```
int risrl = inportb( RISRL );              // low status
int ch = inportb( RIR ) & 0x03;            // channel number
```

intel®

```
switch( inport(LIVR) & 0x03 ) {
case LIVR_GOODDATA:                              // shouldn't happen in DMA
    break;

case LIVR_EXCEPTION:                             // EOF is 'normal' exception
    if( risrl & RISR_EOF ) {
                                                 if( inportb(DMABSTS) & DMABS_NRBUF )
{   // buffer B next
            outport( BRBADRU, ib[ch].upper() );
            outport( BRBADRL, ib[ch].lower() );
            outport( BRBCNT, BUF_MAX );
            outport( BRBSTS, 2401_OWN );
            ib[ch].nxt_buf();                // get next buffer
        } else {                             // buffer A next
            outport( ARBADRU, ib[ch].upper() );
            outport( ARBADRL, ib[ch].lower() );
            outport( ARBCNT, BUF_MAX );
            outport( ARBSTS, 2401_OWN );
            ib[ch].nxt_buf();                // get next buffer
        }
    }
}
outportb( REOIR, ZERO );
```

7.5   Transmit Interrupt Service Routine

The following code example is a transmit interrupt service handler example. When using a synchronous protocol, transmitters must declare an end of frame if an underrun occurs. If the end of buffer is encountered, and no data is transferred by this interrupt service, then the Notrans bit (TEOIR[3]) should be set along with EOF (TEOIR[6]). TEOIR is always the last access of an interrupt service routine.

```
int teoir = ZERO;                          // default
int tisr = inportb( TISR );                // status
int ch = inportb( TIR ) & 0x03;            // channel number

switch( tisr ) {
case TISR_UE:
    teoir = TEOIR_EOF;                     // underflow
    break;
case TISR_TXDATA:
    tftc = inportb( TFTC );                // FIFO count
    for( i = 0; i < tftc; i++) {
        if( ob[ch].is_eob() ) {            // end of buffer?
            ob[ch].nxt_buf();              // get next buffer
            teoir = TEOIR_EOF;
            if( i == 0 )
                                           teoir |= NOTRANS;// no data
transferred
            break;
        }
        else outportb( TDR, ob[ch].nxt_char() );//send next character
    }
}
outportb( TEOIR, teoir );
```

8.0 Detailed Register Descriptions

8.1 Global Registers

8.1.1 Global Firmware Revision Code Register (GFRCR)

Register Name: GFRCR						Intel Hex Address: x'82	
Register Description: Global Firmware Revision Code						Motorola Hex Address: x'81	
Default Value: x'0D							
Access: Byte Read/Write							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Firmware Revision Code							

This register serves two functions in providing the host with information about the CD2401. When the CD2401 is initialized by a hardware RESET* signal or by a software 'Reset All' command issued through any CCR, the CD2401 zeros this register at the start of the initialization. At the conclusion of the initialization, the CD2401 writes the firmware revision code to the GFRCR. All valid CD2401 revision codes are non-zero, the revision code is incremented by one with each new release (for example, GFRCR for Revision M = 0D hex).

Host software must confirm that the GFRCR contents are non-zero before proceeding to configure the CD2401 for normal operation.

8.1.2 Channel Access Register (CAR)

Register Name: CAR						Intel Hex Address: x'EC	
Register Description: Channel Access						Motorola Hex Address: x'EE	
Default Value: x'03							
Access: Byte Read/Write							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Reserved						C1	C0

This register contains the channel number used for the channel-oriented host read/write operations when the host is not in an interrupt service routine. The CD2401 supplies the interrupting channel number during all interrupt service operations. The CAR contents are not used during interrupt service. Note that this indicates that an interrupt service routine is restricted to accessing only the register set of the interrupting Channel and Global registers.

Bits 7:2   Reserved – *must* be written as '0'; read back as a 'don't care'.

Bits 1:0   Channel number

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

8.2 Option Registers

8.2.1 Channel Mode Register (CMR)

Register Name: CMR Register Description: Channel Mode Default Value: x'02 Access: Byte Read/Write					Intel Hex Address: x'18 Motorola Hex Address: x'1B		
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 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 – *must* be '0'.

Bits 2:0   Protocol mode select [2:0]
If these options are changed, an initialization command must be given to the CD2401 through the CCR.

chmd2	chmd1	chmd0	Mode
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

8.2.2 Channel Option Register 1 (COR1)

COR1 — HDLC Mode

Register Name: COR1					Intel Hex Address: x'13		
Register Description: Channel Option 1					Motorola Hex Address: x'10		
Default Value: x'00							
Access: Byte Read/Write							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
AFLO	ClrDet	AdMde1	AdMde0	Flag3	Flag2	Flag1	Flag0

An Initialization command must be given to CD2401 through the CCR if any options specified in this register are changed.

Bit 7   Address Field Length Option
0 = address field is one octet in length
1 = address field is two 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 zeros receive characters, with the CTS* pin high.

Bits 5:4   Addressing modes [1:0]
00 = no address recognition
01 = 4 × 1 byte
10 = 2 × 2 byte.
If this bit is set, RFARs should contain the address to be matched. If AFLO (COR1[7]) is set to '1', an address match is made against the RFAR1 and RFAR2 pair, or the RFAR3 and RFAR4 pair.

Bits 3:0   Inter-frame flag option [3:0]
These bits define the minimum number of flags transmitted before a frame is started.

Flag3	Flag2	Flag1	Flag0	Function
0	0	0	0	Minimum of one opening flag, with shared closing/opening flags permitted.
0	0	0	1	Minimum number of opening flags sent.
through				
1	1	1	1	

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

COR1 — Asynchronous, Bisync, and X.21 Modes

Register Name: COR1
Register Description: Channel Option 1
Default Value: x'00
Access: Byte Read/Write

Intel Hex Address: x'13
Motorola Hex Address: x'10

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

Bit 7   Parity
1 = odd parity
0 = even parity

Bits 6:5   Parity mode 1 and 0 – Defines Parity mode for both transmitter and receiver.

ParM1	ParM0	Parity
0	0	None
0	1	Force (odd parity = force 1, even = force 0)
1	0	Normal
1	1	Reserved

Bit 4   Ignore parity
0 = evaluate parity on received characters.
1 = no parity evaluation of received characters.

Bits 3:0   Character Length [3:0]

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

8.2.3 Channel Option Register 2 (COR2)

COR2 — HDLC Mode

Register Name: COR2 Register Description: Channel Option 2 Default Value: x'00 Access: Byte Read/Write						Intel Hex Address: x'14 Motorola Hex Address: x'17	
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
0	FCSApd	0	CRCNinv	0	RtsAO	CtsAE	DsrAE

Bit 7   Reserved – *must* be '0'.

Bit 6   FCS Append
0 = receive CRC is not passed to host at end of frame.
1 = receive CRC passes to host at end of frame.

Bit 5   Reserved – *must* be '0'.

Bit 4   CRC Inverted
0 = CRC is transmitted inverted (that is, CRC V.41).
1 = CRC is not transmitted inverted (that is, CRC-16).

Bit 3   Reserved – *must* be '0'.

Bit 2   RTS Automatic Output enable
When this bit is set and the channel is enabled, the CD2401 automatically asserts the RTS* output when it has characters to send. When Idle in Mark mode is selected, RTS* is asserted prior to opening flags and remains asserted until after a closing flag has been transmitted.

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

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

COR2 — Asynchronous Mode

Register Name: COR2 Register Description: Channel Option 2 Default Value: x'00 Access: Byte Read/Write						Intel Hex Address: x'14 Motorola Hex Address: x'17	
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
IXM	TxIBE	ETC	0	RLM	RtsAO	CtsAE	DsrAE

Bit 7   Implied XON Mode
IXM has meaning only if TxIBE is set.
If transmission stops due to a received XOFF character, then:

0 = transmission resumes only after the receipt of an XON character or a transmit enable command by the CCR.

1 = transmission resumes after the receipt of any character or a transmit enable command by the CCR.

Bit 6   Transmit In-band flow control Enable
0 = no in-band flow control.

1 = transmission stops after the receipt of an XOFF character (cntl-S or 13 hex). Immediately after receiving an XOFF, any character in the Transmit Shift or Holding registers are transmitted and character transmission halts. Thus, no more than two characters are sent after receiving an XOFF.

Depending on the state of the IXM bit, either the receipt of an XON (cntl-Q or 11 hex) character or any character (IXM = 1) restarts the transmission. A transmit enable command by the CCR also restarts transmission.

Bit 5   Embedded Transmitter Command enable (Async)
When this bit is set, the embedded special transmitter command functions are enabled. The NULL (all zeros) 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 one 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 – *must* be '0'.

Bit 3   Remote Loopback Mode
1 = enables Remote Loopback mode.
0 = disables Remote Loopback mode.

Bit 2   RTS* Automatic Output enable
1 = the RTS* output pin remains enabled during DMA or character bursts from the transmit FIFO. If the CTS* input pin goes high, then RTS* goes high and transmission stops after the current burst is complete.

Bit 1   CTS* Automatic Enable
0 = the transmitter output enable is independent of the CTS* input pin.

1 = CTS* is evaluated prior to the transmission of each character. If CTS* is asserted low, that character is transmitted completely. If CTS* is high, that character transmission is held until CTS* goes low.

Bit 0   DSR* Automatic Enable
0 = the receiver input enable is independent of the DSR* input pin.

1 = DSR* is evaluated at the end of each received character. If DSR* is asserted low, the receiver input is enabled for the next character. If DSR* is high, the receiver is disabled until DSR* goes low.

COR2 — Bisync Mode

Register Name: COR2   Intel Hex Address: x'14
Register Description: Channel Option 2   Motorola Hex Address: x'17
Default Value: x'00
Access: Byte Read/Write

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
LRC	BCC	EBCDIC	CRCNinv	SYN3	SYN2	SYN1	SYN0

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 character set in use.
1 = EBCDIC character set in use.

Bit 4   CRC Transmit Inverted
0 = CRC is transmitted inverted (CRC V.41).
1 = CRC is not transmitted inverted (CRC-16).

Bits 3:0   Extra SYN characters [3:0]
This field determines the number of extra SYN characters that are transmitted before a frame starts. The two required SYNs are not included in this count. This is a binary encode field in which the two required SYNs are followed by the encoded number of characters (0000 = no extra characters; 1111 = 15 extra characters).

Note: In ASCII mode, data is 7 bits with LRC (odd parity) checking only. In EBCDIC mode, data is 8 bits with CRC checking only.

8.2.3.1   COR2 — X.21 Mode

Register Name: COR2 Register Description: Channel Option 2 Default Value: x'00 Access: Byte Read/Write						Intel Hex Address: x'14 Motorola Hex Address: x'17	
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
0	0	ETC	0	0	0	0	0

Bits 7:6   Reserved – *must* be '0'.

Bit 5   Embedded Transmitter Command enable
                1 = embedded command in the FIFO is detected and acted upon.

Bit 4:0   Reserved – *must* be '0'.

FIFO Data Sequence to Implement ETC

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 (refer to Section 6.4 on page 83). The sequence of bytes placed in the datastream to implement this are:

Byte1   This must be equal to 80 hex to start a command sequence.

Byte 2   This byte 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, do not use

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

Byte 4   This is the count of the number of times the character should be sent. If set to '0', the character is sent continuously until more data is provided to the transmitter (but always a minimum of three times).

8.2.4   Channel Option Register 3 (COR3)

COR3 — HDLC Mode

Register Name: COR3 Register Description: Channel Option 3 Default Value: x'00 Access: Byte Read/Write					Intel Hex Address: x'15 Motorola Hex Address: x'16		
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Sndpad	Alt1	FCSPre	FCS	idle	pad2	pad1	pad0

In Synchronous mode COR3 specifies the learning pattern (pad character) sent by the CD2401 to synchronize the DPLL at the remote end. The pad character (00h or AAh) sent depends on the type of encoding used.

Bit 7   Send Pad character
0 = CD2401 does not send any pad character.
1 = CD2401 sends pad character(s) before sending flag when coming out of the Idle In Mark mode.

Bit 6   Send sync pattern
0 = 00h is sent as pad character (NRZI encoding).
1 = AAh is sent as pad character (Manchester/NRZ encoding).

Bit 5   FCS Preset
0 = FCS is preset to all '1's (CRC V.41).
1 = FCS is preset to all '0's (CRC-16).

Bit 4   FCS mode
0 = normal FCS mode. The CD2401 generates and appends CRC on transmit and validates CRC on receive using CRC polynomial selected through the CPSR.
1 = disable FCS generation and checking. The CD2401 treats the entire frame as data.

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

Bits 2:0   Character count [2:0]
These bits specify the number of synchronous characters sent.

pad2	pad1	pad0	Function
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.

pad2	pad1	pad0	Function
1	0	1	Reserved
through			
1	1	1	

COR3 — Asynchronous Mode

Register Name: COR3						Intel Hex Address: x'15	
Register Description: Channel Option 3						Motorola Hex Address: x'16	
Default Value: x'00							
Access: Byte Read/Write							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
ESCDE	RngDE	FCT	SCDE	Splstp	Stop2	Stop1	Stop0

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

Bit 6 — Range Detect Enable
0 = range detect disabled.
1 = characters between SCRl and SCRh (inclusive) generate special character interrupts.

Bit 5 — Flow Control Transparency mode
0 = flow control characters received are passed to the host by receive exception interrupts.
1 = flow control characters received are not passed to the host.
This bit has no effect unless both TxIBE (COR2) and SCDE (COR3) are set.

Bit 4 — Special Character Detection Enable
0 = special character detect for SCHR1 and SCHR2 disabled.
1 = special character detect for SCHR1 and SCHR2 enabled.

This bit must be set along with TxIBE (COR2[6]) before FCT (COR3[5]) becomes effective.

Bit 3 — Special Character I-Strip
When this bit is set, the receive character is I-Stripped (COR3[7] = 0) for the special character matching functions only. The character passed to the host is unaffected. This function allows special character processing of data without knowing if the data is 8-bits no parity or 7-bits with parity.

Bits 2:0 — Stop bit length [2:0]
These bits specify the length of the Stop bit.

Stop2	Stop1	Stop0	Stop Bit Length
0	1	0	1
0	1	1	1.5
1	0	0	2
0	0	0	
through			
0	0	1	
1	1	0	Reserved
through			
1	1	1	

COR3 — Bisync Mode

Register Name: COR3
Register Description: Channel Option 3
Default Value: x'00
Access: Byte Read/Write

Intel Hex Address: x'15
Motorola Hex Address: x'16

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Sndpad	S55	FCSPre	FCS	idle	pad2	pad1	pad0

Bit 7　　Send Pad character
　　　　　0 = CD2401 does not send any pad characters.
　　　　　1 = CD2401 sends pad characters before sending SYN when coming out of the Idle-In Mark mode.

Bit 6　　Send pad pattern
　　　　　0 = hex AA is sent as pad character.
　　　　　1 = hex 55 is sent as pad character.

Bit 5　　FCS Preset
　　　　　0 = FCS is preset to all '0's.
　　　　　1 = FCS is preset to all '1's.

Bit 4　　FCS mode
　　　　　0 = normal FCS mode. The CD2401 generates and appends CRC on transmit and validates CRC on receiving, using the CRC polynomial selected through CPSR.
　　　　　1 = disable FCS generation and checking. The CD2401 treats the entire frame as data.

Bit 3　　Idle mode
　　　　　0 = idle in SYN.
　　　　　1 = idle in mark.

Bits 2:0　Pad count [2:0] – transmit frame leading pads
　　　　　These bits specify the number of pad characters to be sent when coming out of Idle-In Mark mode.

intel

pad2	pad1	pad0	Number of Leading Pads
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	1	Reserved – do not use.
through			
1	1	1	

COR3 — X.21 Mode

Register Name: COR3 Register Description: Channel Option 3 Default Value: x'00 Access: Byte Read/Write					Intel Hex Address: x'15 Motorola Hex Address: x'16		
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
SglSYN	SSDE	StrpSYN	SCDE	0	0	0	0

Bit 7   Single SYN
This bit determines the number of SYN characters that need to be received before Character Synchronization mode is considered received.
0 = two SYN characters are required.
1 = one SYN character is required.

Note: The SglSYN option is only available for Extended X.21 mode; standard X.21 mode always requires two SYN characters. When the SglSYN option is set, StpSYN and SCDE options are disabled.

Note: This option is not available in revisions prior to Revision 'H'.

Bit 6   Steady State Detect Enable
When set, this bit enables the checking of special receive conditions relevant to X.21. These conditions are:

1. All '0's

2. All '1's

3. Alternating '0's and '1's.

4. Change in the condition of the CTS* pin (CTS* is used as the 'I' lead for DTE, or the 'C' lead for 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 the RISR. After detection of the special condition, no further data is passed to the host until different data is received.

There is no character synchronization in certain phases of X.21 call setup. When a data change occurs 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 stripped from the incoming datastream. If this bit is not set, the 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
SCDE is only available when SSDE mode (see above) is enabled. If enabled, the characters programmed in SCHR1–3 are treated as the steady-state conditions in SSDE mode. They are validated for two character times, a special character interrupt is generated and subsequent repetitions of the same data pattern are filtered from the datastream.

Bits 3:0   Reserved – *must* be '0'; reads back as '0'.

8.2.5   Channel Option Register 4 (COR4)

Register Name: COR4   Intel Hex Address: x'16
Register Description: Modem Change Options and FIFO Transfer Threshold   Motorola Hex Address: x'15
Default Value: x'00
Access: Byte Read/Write

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

Bit 7   Detect one-to-zero transition on DSR*
1 = detect one-to-zero transition on the DSR* input (zero-to-one transition on MSVR[7]).

Bit 6   Detect one-to-zero transition on CD*
1 = detect one-to-zero transition on the CD* input (zero-to-one transition on MSVR[6]).

Bit 5   Detect one-to-zero transition on CTS*
1 = detect one-to-zero transition on the CTS* input (zero-to-one transition on MSVR[5]).

Bit 4   Reserved – *must* be '0'.

Bits 3:0   FIFO threshold in characters
Note that 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, in both Interrupt and DMA Transfer modes.

In Asynchronous mode, a Good Data transfer is initiated for the number of characters in the FIFO greater than the specified threshold. Receive timeout and the occurrence of a receive data exception are also cause to initiate a receive transfer.

In Synchronous mode, data transfer is initiated when the number of characters in the FIFO is greater than the specified threshold. An EPF is also cause to initiate a receive transfer.

For transmit operation, the CD2401 attempts 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 CD2401 stops refilling the transmit FIFO once the last character in the frame transfers to the FIFO.

8.2.6   Channel Option Register 5 (COR5)

Register Name: COR5   Intel Hex Address: x'17
Register Description: Channel Option 5   Motorola Hex Address: x'14
Default Value: x'00
Access: Byte Read/Write

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

This register defines the current state change options to be monitored.

Bit 7   Detect zero-to-one transition on DSR*
1 = detect zero-to-one transition on DSR* input (one-to-zero transition on MSVR[7])

Bit 6   Detect zero-to-one transition on CD*
1 = detect zero-to-one transition on CD* input (one-to-zero transition on MSVR[6])

Bit 5   Detect zero-to-one transition on CTS*
1 = detect zero-to-one transition on CTS* input (one-to-zero transition on MSVR[5])

Bit 4   Reserved – *must* be '0'.

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 where the automatic hardware (DTR/DSR) flow control is activated. A threshold value of '0' 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.

8.2.7 Channel Option Register 6 (COR6)

COR6 — Async Mode

Register Name: COR6
Register Description: Channel Option 6
Default Value: x'00
Access: Byte Read/Write

Intel Hex Address: x'1B
Motorola Hex Address: x'18

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

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

Bits 7:5   These three bits enable translation of received CR/NL characters as:

IgnCr	ICRNL	INLCR	Function
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.

Bits 4:3   Break action
These bits determine the action taken after a BREAK condition is received.

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

Bits 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 is generated with the Timeout bit (RISRl[7]) set when the end-of-break is detected. The RET interrupt must be enabled (IER[5]) to enable this feature.

ParMrk	INPCK	ParInt	Function
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[1]	0	1	Translate to a sequence of FF NULL and the error character, pass on as Good Data.
1	1	0	Reserved
1	1	1	Reserved

1. When ParMrk = 1 and ParInt = 1, each occurrence of FF hex in the datastream is preceded by FF hex to distinguish it from a parity error sequence.

COR6 — Bisync Mode

Register Name: COR6 Register Description: Channel Option 6 Default Value: x'00 Access: Byte Read/Write						Intel Hex Address: x'1B Motorola Hex Address: x'18	
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Special Frame Termination Character							

In Bisynchronous mode, this register provides a frame-termination method for a receive character other than the one already defined in the bisynchronous specification. When an initialize channel command in the CCR is processed, COR6 is set to the value of DLE (hex 10); this deactivates the function because DLE processing is always performed before a match with COR6 is performed. Following the completion of the initialize channel command, the user can program COR6 to any character value where frame termination is required.

One use of this might be to terminate non-transparent frames where the termination character has been corrupted. If the idle line character (hex FF in 8-bit mode, hex 7F in 7-bit mode) is programmed in COR6 when a normal termination character is corrupted (that is, ETX hex 03 is changed to hex 07), the BCC is received as data and the idle line condition causes the frame to terminate. The idle line character is the last character in the frame. No attempt is made to perform a CRC on the receive data under these conditions.

Note: This feature is not available in revisions prior to Revision 'H'.

COR6 — X.21 Mode

Register Name: COR6				Intel Hex Address: x'1B			
Register Description: Channel Option 6				Motorola Hex Address: x'18			
Default Value: x'00							
Access: Byte Read/Write							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
SYN Character							

In X.21 mode, this register contains the character to be used for find character synchronization with the receive datastream. When the initialize channel command in the CCR is processed, COR6 is set to the value ASCII SYN (hex 16). If the user requires a different synchronization character, COR6 can be loaded with that character after the initialize channel command completes, as in the following sequence:

1. Issue initialize channel command.

2. Wait for command to complete (CCR returns to hex 00).

3. Reprogram COR6 to the desired character.

4. Issue receive enable command by CCR.

Note: This feature is not available in revisions prior to Revision 'H'.

8.2.8   Channel Option Register 7 (COR7) — Async Mode only

Register Name: COR7				Intel Hex Address: x'04			
Register Description: Channel Option 7				Motorola Hex Address: x'07			
Default Value: x'00							
Access: Byte Read/Write							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 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   I-Strip
When this bit is set, the most-significant bit of receive characters is stripped, leaving 7-bit characters. IStrip is applied before special character processing, but after all other character processing.

Bit 6   LNext Enable
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.

intel.

LNE provides a mechanism to transfer flow control and special characters as normal data, without invoking flow control action in the CD2401 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 are not processed for special character/flow control matching.
1 = all receive characters, even those with errors, are processed for special character/ flow control processing.

Bits 4:2   Reserved – *must be '0'*.

Bits 1:0   Transmit processing for CR and NL; these bits define the Translation mode when CR and/or NL are present in the transmit data.

ONLCR	OCRNL	Function
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.

8.2.9   Special Character Registers — Async Mode only

Special Character registers can be used for detecting specific receive characters in the incoming datastream and can be used to transmit characters (by STCR) preempting any data in the transmit FIFO.

8.2.9.1   Special Character Register 1 (SCHR1)

Register Name: SCHR1						Intel Hex Address: x'1C	
Register Description: Special Character 1						Motorola Hex Address: x'1F	
Default Value: x'00							
Access: Byte Read/Write							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
User-Defined Special Character XON Character							

8.2.9.2   Special Character Register 2 (SCHR2)

Register Name: SCHR2 Register Description: Special Character 2 Default Value: x'00 Access: Byte Read/Write				Intel Hex Address: x'1D Motorola Hex Address: x'1E			
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
User-Defined Special Character XOFF Character							

SCHR1 and 2 are used in conjunction with the SCDE bit (COR3[4]) to detect specific incoming characters. When both the SCDE and TxIBE (COR2[6]) bits are set, they define the in-band flow control characters XON and XOFF. The values are user-defined, regardless how used.

SCHR1 = XON
SCHR2 = XOFF

In addition to the SCDE and TxIBE bits, if the FCT bit (COR3[5]) is set when flow control characters are received, they are stripped from the datastream.

8.2.9.3   Special Character Register 3 (SCHR3)

Register Name: SCHR3 Register Description: Special Character 3 Default Value: x'00 Access: Byte Read/Write				Intel Hex Address: x'1E Motorola Hex Address: x'1D			
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
User-Defined Special Character							

8.2.9.4   Special Character Register 4 (SCHR4)

Register Name: SCHR4 Register Description: Special Character 4 Default Value: x'00 Access: Byte Read/Write				Intel Hex Address: x'1F Motorola Hex Address: x'1C			
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
User-Defined Special Character							

SCHR3 and 4 are used in conjunction with the ESCDE bit (COR3[7]) to detect characters in the receive datastream and generate receive special character interrupts.

Note: SCHR3 and 4 are not stripped from the datastream if FCT mode is enabled.

8.2.10 Special Character Range Register — Async Mode only

8.2.10.1 Special Character Range – Low (SCRl)

Register Name: SCRl Register Description: Special Character Range – low Default Value: x'00 Access: Byte Read/Write					Intel Hex Address: x'20 Motorola Hex Address: x'23		
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
User-Defined Special Character Range Detect – low							

8.2.10.2 Special Character Range – High (SCRh)

Register Name: SCRh Register Description: Special Character Range – high Default Value: x'00 Access: Byte Read/Write					Intel Hex Address: x'21 Motorola Hex Address: x'22		
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
User-Defined Special Character Range Detect – high							

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

8.2.11 LNext Character Register (LNXT) — Async Mode only

Register Name: LNXT Register Description: Literal Next Character Default Value: x'00 Access: Byte Read/Write					Intel Hex Address: x'2D Motorola Hex Address: x'2E		
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
User-Defined Literal Next Character							

This register defines the LNext character. If the LNext function is enabled (COR7[6]), the CD2401 examines received characters and compares them against this value. If a match occurs, this character and the next are placed in the FIFO without any special processing. In effect, the LNext function causes the CD2401 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 'ERROR' character, LNext is placed in the FIFO, and the following next character is treated as it normally would be for these error conditions.

8.2.12 Receive Frame Address Registers (RFAR) — HDLC Sync Mode only

8.2.12.1 Receive Frame Address Register 1 (RFAR1)

Register Name: RFAR1 Register Description: Receive Frame Address 1 Default Value: x'00 Access: Byte Read/Write						Intel Hex Address: x'1C Motorola Hex Address: x'1F	
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Frame Qualification Address 1							

8.2.12.2 Receive Frame Address Register 2 (RFAR2)

Register Name: RFAR2 Register Description: Receive Frame Address 2 Default Value: x'00 Access: Byte Read/Write						Intel Hex Address: x'1D Motorola Hex Address: x'1E	
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Frame Qualification Address 2							

8.2.12.3 Receive Frame Address Register 3 (RFAR3)

Register Name: RFAR3 Register Description: Receive Frame Address 3 Default Value: x'00 Access: Byte Read/Write						Intel Hex Address: x'1E Motorola Hex Address: x'1D	
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Frame Qualification Address 3							

8.2.12.4 Receive Frame Address Register 4 (RFAR4)

Register Name: RFAR4 Register Description: Receive Frame Address 4 Default Value: x'00 Access: Byte Read/Write						Intel Hex Address: x'1F Motorola Hex Address: x'1C	
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Frame Qualification Address 4							

Reception of an HDLC frame can be qualified with a matched 1- or 2-byte address field either as four 1-byte alternatives or two 2-byte alternatives. Control of how the RFARs are used for address recognition is detailed in the description of the CORs (see page 94 through page 108).

8.2.13 CRC Polynomial Select Register (CPSR)

Register Name: CPSR Register Description: CRC Polynomial Select Default Value: x'00 Access: Byte Read/Write					Intel Hex Address: x'D4 Motorola Hex Address: x'D6		
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
0	0	0	0	0	0	0	Poly

Bits 7:1   Reserved – *must* be '0'.

Bit 0   Polynomial select
0 = CRC V.41 polynomial (normally used for HDLC protocol and preset to 1's)
$x^{**}16 + x^{**}12 + x^{**}5 + 1$

1= CRC-16 polynomial (generally used for Bisync but will work in HDLC mode, preset to 0's)
$x^{**}16 + x^{**}15 + x^{**}2 + 1$

8.3 Bit Rate and Clock Option Registers

8.3.1 Receive Bit Rate Generator Registers

8.3.1.1 Receive Bit Rate Period Register (RBPR)

Register Name: RBPR Register Description: Receive Bit-Rate Period Default Value: x'81 Access: Byte Read/Write					Intel Hex Address: x'C9 Motorola Hex Address: x'CB		
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Receive Bit Rate Period (Divisor)							

This register contains the preload value for the receive bit 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, determines the receive bit rate. If a 1× external clock is used, a value of 01h must be loaded in the RBPR.

8.3.1.2   Receive Clock Option Register (RCOR)

Register Name: RCOR						Intel Hex Address: x'CA	
Register Description: Receive Clock Option						Motorola Hex Address: x'C8	
Default Value: x'00							
Access: Byte Read/Write							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
TLVal	0	DpllEn	Dpllmd1	Dpllmd0	ClkSel2	ClkSel1	ClkSel0

This register is used to select the DPLL mode and the desired clock source for the receive BRG.

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

Bit 6   Reserved – *must* be '0'.

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

Bits 4:3   DPLL Mode select [1:0]
             These bits 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   Clock Select [2:0]
             These three bits select the clock source for the receive BRG 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 the clock options in Section 5.5 on page 58.

8.3.2 Transmit Bit Rate Generator Registers

8.3.2.1 Transmit Bit Rate Period Register (TBPR)

Register Name: TBPR Register Description: Transmit Bit Rate Period Default Value: x'81 Access: Byte Read/Write						Intel Hex Address: x'C1 Motorola Hex Address: x'C3	
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Transmit Bit Rate Period (Divisor)							

This register contains the preload value for the transmit bit 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, determines the transmit bit rate. If a 1× external clock or the receive clock is used, a value of 01h must be loaded in the TBPR.

8.3.2.2 Transmit Clock Option Register (TCOR)

Register Name: TCOR Register Description: Transmit Clock Option Default Value: x'00 Access: Byte Read/Write						Intel Hex Address: x'C2 Motorola Hex Address: x'C0	
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
ClkSel2	ClkSel1	ClkSel0	0	Ext-1X	0	LLM	0

This register controls the transmit BRG and Local Loopback mode.

Bits 7:5　　Clock Select [2:0]

These bits select the clock source for the transmit BRG.

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	Receive clock

Note:　See the detailed description of clock options in .

Bit 4   Reserved – *must* be '0'.

Bit 3   Times 1 external clock.
               This bit is set to '1' when the user supplies the data clock on TXCIN pin where the
               frequency is equal to the transmit data rate. When using the external 1× clock or the
               clock from the receiver's DPLL, the TBPR must be programmed to '01h'.

Bit 2   Reserved – *must* be '0'.

Bit 1   Local Loopback Mode
               1 = enables the Local Loopback mode
               0 = disables the Local Loopback mode

Bit 0   Reserved – *must* be '0'.

8.4   Channel Command and Status Registers

There are two CCR command sets. Mode 1 (bit 7 = 0) commands affect basic channel control. In
Mode 2 (bit 7 = 1), additional commands that control timer functions are available.

8.4.1   Channel Command Register (CCR)

Mode 1

Register Name: CCR   Intel Hex Address: x'10
Register Description: Channel Command Register – Mode 1   Motorola Hex Address: x'13
Default Value: x'00
Access: Byte Read/Write

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

The various command and control bits in this register perform largely independent functions. The
host can assert multiple command bits to achieve the desired effect. The CD2401 clears the register
to '1' after it accepts and acts on a host command. The host must verify that the contents of this
register are '0' 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 is:

 1. Clear channel

 2. Initialize channel

 3. Enable receive

 4. Disable receive

 5. Enable transmit

 6. Disable transmit

intel®

Note: Processing CCR commands is a low-priority task for the internal firmware, since they seldom occur. The user must take care when waiting for command completions at critical times, that is, during interrupt service routines.

Bit 7   For mode 1, this bit *must* be '0'.

Bit 6   Clear Channel command
When this command is issued, the CD2401 clears the data FIFOs and current transmit and receive status of the channel in the CSR. 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. Channel parameters are not affected by a Clear Channel 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 CMR or COR1, the channel must be reinitialized by this command. The Initialize Channel command causes the internal protocol-specific registers to be initialized.

Warning: If the Initialize Channel command is issued after a channel is already in operation, then a Clear Channel command must be issued prior to, or coinciding with the Initialize Channel command. Failure to observe this requirement will result in unpredictable device behavior.

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 can issue. None of the other command bits is interpreted if the Reset All command is given. The host must reinitialize the CD2401 following the execution of this command, just like after a hardware power-on reset. When this command is complete, the GFRCR is updated with the firmware revision code.

Bit 3   Enable Transmitter
This bit enables the transmitter by setting the TxEn bit (CSR[3]). In Asynchronous mode, this command also clears the transmit flow control options.

Bit 2   Disable Transmitter
This bit disables the transmitter by clearing the TxEn bit (CSR[3]). In Asynchronous mode, the transmit flow control bits are cleared.

Bit 1   Enable Receiver
This bit enables the receiver by setting the RxEn bit (CSR[7]). In Asynchronous mode, the receive flow control bits are also cleared.

Bit 0   Disable Receiver
This bit disables the receiver by clearing the RxEn bit (CSR[7]). In Asynchronous mode, the Receive Flow Control bits are also cleared.

Mode 2

Register Name: CCR Register Description: Channel Command – Mode 2 Default Value: x'00 Access: Byte Read/Write						Intel Hex Address: x'10 Motorola Hex Address: x'13	
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
1	ClrT1	ClrT2	0 ClrRcv[1]	0	0	0	0

1. Only available on Revision H and later.

Either one or both timers can be cleared with a single command. 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 7   For Mode 2, this bit *must* be '1'.

Bit 6   Clear Timer 1
General Timer 1 is cleared.

Bit 5   Clear Timer 2
General Timer 2 is cleared.

Bit 4   Clear Receiver Command
This command only affects the receiver. It resets all receiver functions like a combination of clear channel, initialize channel and enable receiver commands. ClrRcv clears the receive FIFO and clears receive status in the CSR register, except for the RcvEn bit. ClrRcv clears the receive DMA buffer status in A/BRBSTS and receive status in DMABSTS. Clearing the 2401own bits in both Receive Buffer Status registers means that DMA buffers *must* be returned to the CD2401 before receive transfers can begin again.

For Synchronous modes, this command puts the receiver back into SYN/Flag Hunt mode.

Note: This command is not available in revisions prior to Revision 'H'.

Bits 3:0   Reserved – *must* be '0'.

8.4.2   Special Transmit Command Register (STCR)

Async and HDLC Modes only

Register Name: STCR						Intel Hex Address: x'11	
Register Description: Special Transmit Command						Motorola Hex Address: x'12	
Default Value: x'00							
Access: Byte Read/Write							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
0	AbortTx	AppdCmp	0	SndSpc	SSPC2	SSPC1	SSPC0

The CD2401 clears STCR to '0' when it accepts a host CPU command.

Bit 7   Reserved – *must* be '0'.

Bit 6   Abort Transmission (HDLC mode)
             This bit terminates 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 is made to the append buffer.

Bit 4   Reserved – *must* be '0'.

Bit 3   Send Special character(s) command
             Use this command in Asynchronous mode to send a user-defined special character
             or special character sequence. The special character is transmitted ahead on any data
             remaining in the FIFO.

Bits 2:0   Select Special Character [2:0]

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

8.4.3   Channel Status Register (CSR)

This status register stores the current state of the channel. It can 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.

HDLC Mode

Register Name: CSR Register Description: Channel Status Default Value: x'00 Access: Byte Read/Write					Intel Hex Address: x'19 Motorola Hex Address: x'1A		
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
RxEn	RxFlag	RxFrame	RxMark	TxEn	TxFlag	TxFrame	TxMark

Bit 7   Receiver Enable
0 = receiver disabled.
1 = receiver enabled.

Bit 6   Receive Flag
0 = currently not receiving flag/SYN.
1 = currently receiving flag/SYN.

Bit 5   Receive Frame
0 = currently not receiving frame.
1 = currently receiving frame.

Bit 4   Receive Mark
0 = currently not receiving continuous mark.
1 = currently receiving continuous mark.

Bit 3   Transmitter Enable
0 = transmitter disabled.
1 = transmitter enabled.

Bit 2   Transmit Flag
0 = currently not transmitting flag.
1 = currently transmitting flag.

Bit 1   Transmit Frame
0 = currently not transmitting frame.
1 = currently transmitting frame.

Bit 0   Transmit Mark
0 = currently not transmitting continuous '1's.
1 = currently transmitting continuous '1's.

Asynchronous Mode

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

intel。

Register Name: CSR
Register Description: Channel Status
Default Value: x'00
Access: Byte Read/Write

Intel Hex Address: x'19
Motorola Hex Address: x'1A

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

Bit 7   Receiver Enable
0 = receiver disabled.
1 = receiver enabled.

Bit 6   Receive Flow Off
0 = normal
1 = the CD2401 requested the remote to stop transmission (Send XOFF command given to the channel). This bit is reset when the CD2401 requests the remote to restart transmission, the receiver is enabled/disabled, or the channel is reset.

Bit 5   Receive Flow On
0 = normal
1 = the CD2401 requested the remote to restart character transmission (Send XON command given to the channel). This bit is reset when the next (non-flow control) character is received, the receiver is enabled/disabled, or the channel is reset.

Bit 4   Reserved – always returns '0' when read.

Bit 3   Transmitter Enable
0 = transmitter disabled.
1 = transmitter enabled.

Bit 2   Transmit Flow Off
0 = normal
1 = the CD2401 was requested by the remote to stop transmission. This bit is reset when the CD2401 receives a request to resume transmission, the transmitter is enabled/disabled, or the channel is reset.

Bit 1   Transmit Flow On
0 = normal
1 = the CD2401 was requested by the remote to resume transmission. This bit is reset once character transmission is resumed, the transmitter is enabled/disabled, or the channel is reset.

Bit 0   Reserved – always returns '0' when read.

Bisynchronous Mode

Register Name: CSR					Intel Hex Address: x'19		
Register Description: Channel Status					Motorola Hex Address: x'1A		
Default Value: x'00							
Access: Byte Read/Write							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
RxEn	RxITB	RxFrame	0	TxEn	TxITB	TxFrame	0

Bit 7   Receiver Enabled status
            0 = receiver disabled.
            1 = receiver 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 is included in the BCC calculation (see Table 16 on page 82).

Bit 5   Receive Frame
            When this bit is set, it indicates that the CD2401 is currently receiving a frame.

Bit 4   Reserved – always returns '0' when read.

Bit 3   Transmitter Enabled status
            0 = transmitter disabled.
            1 = transmitter enabled.

Bit 2   Transmit ITB
            This bit is set if the last frame transmitted ended with an ITB character (that is, the leading character of the next frame is included in the BCC calculation).

Bit 1   Transmit Frame status
            When this bit is set, it indicates that the CD2401 is currently transmitting a frame.

Bit 0   Reserved – always returns '0' when read.

X.21 Mode

Register Name: CSR					Intel Hex Address: x'19		
Register Description: Channel Status					Motorola Hex Address: x'1A		
Default Value: x'00							
Access: Byte Read/Write							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
RxEn	0	RxSpc	0	TxEn	0	TxSpc	0

Bit 7   Receiver Enabled status
            0 = receiver disabled.
            1 = receiver enabled.

Bit 6   Reserved – always returns '0' when read.

Bit 5   Receive Special
               When this bit is set, it indicates that the CD2401 is currently in a steady-state condi-
               tion. Such conditions generate a receive special character interrupt.

Bit 4   Reserved – always returns '0' when read.

Bit 3   Transmitter Enabled status
               0 = transmitter disabled.
               1 = transmitter enabled.

Bit 2   Reserved – always returns '0' when read.

Bit 1   Transmit Special
               When this bit is set, it indicates that the CD2401 is currently transmitting an ETC
               command, as defined in COR2.

Bit 0   Reserved – always returns '0' when read.

8.4.4   Modem Signal Value Registers (MSVR)

8.4.4.1   Modem Signal Value Register (MSVR-RTS)

Register Name: MSVR-RTS Register Description: Modem Signal Value – RTS Default Value: x'00 Access: Byte Read/Write						Intel Hex Address: x'DC Motorola Hex Address: x'DE	
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
DSR	CD	CTS	DTRop	0	0	DTR	RTS

8.4.4.2   Modem Signal Value Register (MSVR-DTR)

Register Name: MSVR-DTR Register Description: Modem Signal Value – DTR Default Value: x'00 Access: Byte Read/Write						Intel Hex Address: x'DD Motorola Hex Address: x'DF	
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
DSR	CD	CTS	DTRop	0	0	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 the control of the DTR* and RTS* output pins: a write to MSVR-
DTR only affects DTR*; a write to MSVR-RTS only affects RTS*.

Bit 7   This bit reflects the current state of DSR*.

Bit 6   This bit reflects the current state of CD*.

Bit 5   This bit reflects the current state of CTS*.

Bit 4   DTR option (written by MSVR-DTR)
                 0 = the value of MVSR-DTR[1] is output on TXCOUT/DTR*.
                 1 = the transmit clock is output on TXCOUT/DTR*.

Note: If the transmit clock source is a 1× clock on TXCIN/CD*, this signal cannot be driven on TXCOUT/DTR*.

Bit 3   Reserved – *must* be '0'.

Bit 2   Unused – always returns '0' when read; writing has no effect.

Bit 1   This bit reflects the current state of DTR*.

Bit 0   This bit reflects the current state of RTS*.

8.5   Interrupt Registers

8.5.1   General Interrupt Registers

8.5.1.1   Local Interrupt Vector Register (LIVR)

Register Name: LIVR Register Description: Local Interrupt Vector Default Value: x'00 Access: Byte Read/Write						Intel Hex Address: x'0A Motorola Hex Address: x'09	
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
X	X	X	X	X	X	IT1	IT0

The host effectively controls bits 7:2; the device provides bits 1:0 within an interrupt acknowledge context.

The CD2401 has one Local Interrupt Vector register per channel, each with six host-defined bits. The host can opt to embed the channel number and the protocol in use in the channel vector. The CD2401 supplies two modified bits signifying the type of interrupt service required.

Bits 7:2   User-defined. These six bits can be used as the CD2401 device ID number.

Bits 1:0   Interrupt Type [1:0]

                 These two bits indicate the group/type of interrupt occurring.

IT1	IT0	Group/Type
0	1	Group 1: modem signal change interrupt/general timer interrupt.
1	0	Group 2: transmit data interrupt.
1	1	Group 3: receive data interrupt.
0	0	Group 3: receive exception interrupt.

Note: Because the CD2401 provides a unique Local Interrupt Vector register for each channel, the host has the option to include the channel number within the interrupt vector.

8.5.1.2   Interrupt Enable Register (IER)

Register Name: IER
Register Description: Interrupt Enable
Default Value: x'00
Access: Byte Read/Write

Intel Hex Address: x'12
Motorola Hex Address: x'11

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

Bit 7 — Modem pin change detect enable
This is the Master interrupt enable for modem change detect functions. The host can select which modem pins are monitored for input change and select either, or both, directions of change by programming the change detect option bits in COR4 and COR5. A Group 1-type interrupt (see LIVR description) is generated from this enable.

Bit 6 — Reserved – *must* be '0'.

Bit 5 — Receive Exemption Timeout (Async 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 – *must* be '0'.

Bit 3 — Receive Data
The receive FIFO threshold was reached in Interrupt Transfer mode, causing a Group 3 receive data interrupt. Any receive exception causes a Group 3 receive exception interrupt.

Bit 2 — General timer(s) timeout enable
In Synchronous mode this bit enables a Group 1 interrupt when either timer reaches zero.

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

Bit 0   Transmit Data
Any transmit exception or transmit FIFO threshold reached in Interrupt Transfer mode. Group 2 interrupts are generated at the end of transmit DMA buffers or when the FIFO threshold is reached in Interrupt Transfer mode.

8.5.1.3   Local Interrupting Channel Register (LICR)

Register Name: LICR Register Description: Local Interrupting Channel Default Value: C1:C0 contain the channel number Access: Byte Read/Write						Intel Hex Address: x'25 Motorola Hex Address: x'26	
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
X	X	X	X	C1	C0	X	X

These Per-Channel registers are initialized with each channel number. The locations are RAM registers and can be used for any purpose.

Bits 7:4   User-defined

Bits 3:2   Channel nuber [1:0]

These bits define the interrupting channel number.

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

Bits 1:0   User-defined

8.5.1.4   Interrupt Stack Register (STK)

Register Name: STK Register Description: Interrupt Stack Default Value: x'00 Access: Byte Read only						Intel Hex Address: x'E0 Motorola Hex Address: x'E2	
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 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 contains the internal interrupt nesting history. The stack is pushed from bits 7 and 0 toward the center during an interrupt acknowledge cycle, and popped from the center during a write to an end of interrupt register.

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

CLvl [1]	CLvl [0]	Function
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[1:0] These bits hold a previously-active interrupt now nested.

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

8.5.2 Receive Interrupt Registers

8.5.2.1 Receive Priority Interrupt Level Register (RPILR)

Register Name: RPILR
Register Description: Receive Priority Interrupt Match
Default Value: x'00
Access: Byte Read/Write

Intel Hex Address: x'E3
Motorola Hex Address: x'E1

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
			User-Assigned Priority Match Value				

This register must be initialized by the host to contain the codes that are presented on the address bus by the host system to indicate which of the three CD2401 interrupt types (that is, modem, transmit, or receive) is being acknowledged when IACKIN* is asserted. The CD2401 compares bits 0–6 in this register with A[6:0] 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 this register always reads back as '0'. When each of the three PILRs is programmed with the same value, automatic internal prioritization is activated, with receive as the highest priority, followed by transmit and modem.

8.5.2.2   Receive Interrupt Register (RIR)

Register Name: RIR					Intel Hex Address: x'EF		
Register Description: Receive Interrupt Register					Motorola Hex Address: x'ED		
Default Value: x'00							
Access: Byte Read only							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Ren	Ract	Reoi	0	Rvct [1]	Rvct [0]	Rcn [1]	Rcn [0]

Bit 7   Receive Enable
               This bit is set by the CD2401 to initiate a receive interrupt request sequence. This bit
               is cleared during a valid receive interrupt acknowledge cycle.

Bit 6   Receive Active
               This bit is set automatically when Ren is set, and the Fair Share logic allows the
               assertion of a receive interrupt request. This bit is cleared when the host CPU writes
               to the REIOR.

Bit 5   Receive End of Interrupt
               This bit is automatically set when the host CPU writes to the REIOR during 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 complete.

Bit 4   Unused – always returns '0' when read.

Bits 3:2   Receive Vector [1:0]
               These bits are set by the CD2401 to provide the lower 2 bits of the vector supplied
               to the host CPU during an interrupt acknowledge cycle.
               The Receive Good Data vector is decoded as: Rvct [1] = 1, Rvct [0] = 1
               The Receive exception vector is decoded as: Rvct [1] = 0, Rvct [0] = 0

Bits 1:0   Receive Channel Number [1:0]
               These bits are set by the CD2401 to indicate the channel requiring receive interrupt
               service.

8.5.2.3   Receive Interrupt Status Register (RISR)

Register Name: RISR					Intel Hex Address: x'8A		
Register Description: Receive Interrupt Status					Motorola Hex Address: x'88		
Default Value: x'00							
Access: Word Read only							
15	14	13	12	11	10	9	8
RISR high							
7	6	5	4	3	2	1	0
RISR low							

This register reports the status of the channel during the receive interrupt service. It is a 16-bit register, with the lower byte displaying current receive character oriented status while the upper byte displays current DMA interrupt status. The upper byte is not used if DMA mode is not active.

HDLC Mode

Register Name: RISRI					Intel Hex Address: x'8A		
Register Description: Receive Interrupt Status - low					Motorola Hex Address: x'89		
Default Value: x'00							
Access: Byte Read only							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
0	EOF	RxAbt	CRC	OE	ResInd	0	ClrDct

Bit 7   Reserved – always returns '0' when read

Bit 6   End of Frame
This bit indicates that a valid end of frame was received, and the frame is essentially complete.

Bit 5   Receive Abort
This bit indicates that an abort sequence terminating the frame was received.

Bit 4   Receive CRC error
This bit indicates that a frame with a valid end of frame was received, but the FCS was not correct.

Bit 3   Overrun Error
This bit indicates that new data has arrived, but the CD2401 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 is discarded.

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

Bit 1   Reserved – always returns '0' when read

Bit 0   Clear Detect
This bit 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 in COR1.

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

Asynchronous Mode

Register Name: RISRI Register Description: Receive Interrupt Status - low Default Value: x'00 Access: Byte Read only				Intel Hex Address: x'8A Motorola Hex Address: x'89			
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Timeout	SCdet2	SCdet1	SCdet0	OE	PE	FE	Break

Bit 7   Timeout
This bit 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 this bit is set.

Bits 6:4 Special Character Detect [2:0]

SCdet2	SCdet1	SCdet0	Status
0	0	0	None detected.
0	0	1	Special character 1 matched.
0	1	0	Special character 2 matched.
0	1	1	Special character 3 matched (only if COR3[7] is enabled).
1	0	0	Special character 4 matched (only if COR3[7] is enabled).
1	1	1	Character is within the inclusive range of the characters in the SCRs (only if COR3[6] is enabled). Special character match can be enabled for error characters by COR7.

Bit 3   Overrun Error
This bit indicates that new data has arrived, but the CD2401 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
This bit indicates that a parity error occurred.

Bit 1   Framing Error
This bit indicates that a bad stop bit is detected.

Bit 0   Break
This bit indicates that a BREAK is detected.

Bisynchronous Mode

Register Name: RISRI						Intel Hex Address: x'8A	
Register Description: Receive Interrupt Status - low						Motorola Hex Address: x'89	
Default Value: x'00							
Access: Byte Read only							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
0	EOF	RxAbt	CRC	OE	0	0	0

Bit 7   Reserved – reads as '0'.

Bit 6   End of Frame
This bit indicates that a valid end of frame was received, and the frame is essentially complete.

Bit 5   Receive Abort
This bit indicates that an abort sequence terminating the frame was received.

Bit 4   Receive CRC error
This bit indicates that a frame with a valid end of frame was received, but the FCS was not correct.

Bit 3   Overrun Error
This bit indicates that the receiver buffer and FIFO were overrun. At least one new character was received, but lost since there was no room available in the receiver buffer and/or FIFO. The OE status is set on the last character received before the overrun occurred.

Bit 2:0   Reserved – reads as '0'.

RISRI - X.21 Mode

Register Name: RISRI						Intel Hex Address: x'8A	
Register Description: Receive Interrupt Status Register - low						Motorola Hex Address: x'89	
Default Value: x'00							
Access: Byte Read/Write							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
LVal	SCdet2	SCdet1	SCdet0	OE	PE	0	LChg

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

Bit 7   Lead Value
0 = OFF
1 = ON

Bits 6:4   Special Character Detect [2:0]
This indication occurs if two consecutive characters matching the value indicated in the table below are received.

SCdet2	SCdet1	SCdet0	Status
0	0	0	None detected.
0	0	1	Matched the value in SCHR1.
0	1	0	Matched the value in SCHR2.
0	1	1	Matched the value in SCHR3.
1	0	0	All '0' condition.
1	0	1	All '1' condition.
1	1	0	Alternating '0' and '1' condition.
1	1	1	SYN detect.

Bit 3   Overrun Error
This bit indicates that the receiver buffer and FIFO were overrun. At least one new character was received, but lost since there was no room available in the receiver buffer and/or FIFO. This status is set on the last character received before the overrun occurred.

Bit 2   Parity Error
This bit indicates that a parity error occurred on this character.

Bit 1   Reserved – always returns '0' when read.

Bit 0   Lead Change
This bit 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 Lead Change indication can precede a special character interrupt.

During an interrupt service routine, the host can use this register to provide a timer value as detailed in the REOIR.

8.5.2.4   Receive Interrupt Status Register high (RISRh)

Register Name: RISRh
Register Description: Receive Interrupt Status - high
Default Value: x'00
Access: Byte Read only

Intel Hex Address: x'8B
Motorola Hex Address: x'88

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

Note: This register is used in DMA mode only.

Bit 7   Bus Error (written by CD2401)
0 = no bus error.
1 = bus error detected on last transfer. The actual address at which the error occurred

is available in the A/BRCBAR. 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 by setting REOIR[7] and move onto the next.

Bit 6   End of Frame
         Reception of a data frame is complete (Sync DMA mode only).

Bit 5   End of Buffer
         The end of a receive buffer is reached. Used only for DMA supported transmission. The end of one of the host-supplied receive buffers is reached.

Bit 4   Reserved – always returns '0' when read.

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

Bits 2:0   Reserved – always returns '0' when read

8.5.2.5   Receive FIFO Output Count (RFOC)

Register Name: RFOC   Intel Hex Address: x'33
Register Description: Receive FIFO Output Count   Motorola Hex Address: x'30
Default Value: x'00
Access: Byte Read only

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

Bits 7:5   Reserved – always returns '0' when read.

Bits 4:0   Receive Data Count [4:0]
            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.

8.5.2.6   Receive Data Register (RDR)

Register Name: RDR   Intel Hex Address: x'F8
Register Description: Receive Data   Motorola Hex Address: x'F8
Default Value: x'00
Access: Byte Read only

Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 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 conventions for the positioning of valid data on the bus. If the BYTESWAP pin is high, data is valid on A/D[7:0]; if BYTESWAP is low, data is valid on A/D[15:8]. This is true because the RDR is on an even address.

8.5.2.7   Receive End of Interrupt Register (REOIR)

Register Name: REOIR   Intel Hex Address: x'87
Register Description: Receive End of Interrupt   Motorola Hex Address: x'84
Default Value: x'00
Access: Byte Write only

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

This must be written to by the host receive interrupt service routine to signal to the CD2401 that the current interrupt service is concluded. This must be the last access to the CD2401 during an interrupt service routine. A write to this register generates an internal end-of-interrupt signal that pops the CD2401 interrupt context stack.

Depending on the circumstances of an individual interrupt service, the host may be required to pass a parameter to the CD2401 through these registers. The CD2401 interprets the values written to this register at the completion of all receive interrupts as commands to be executed.

Bit 7   Terminate current DMA Buffer
               If this bit is set, the current receive buffer is terminated and data transfer is switched to the other buffer. This bit should only be set in response to an Async Exception interrupt. If the buffer is terminated in response to an Exception Character (that is, Parity Error) interrupt and the DiscExc bit is not set, the exception character is written at the start of the next buffer.

               Before writing the terminate buffer command to REOIR, a new buffer descriptor can be written to the current buffer.

Bit 6   Discard Exception Character (DMA mode only)
               When this bit is set in response to an Async Exception interrupt, the exception character is not transferred to memory.

Bit 5   Set General Timer 2 (Synchronous modes)
               0 = do not set General Timer 2.
               1 = load the value provided in RISRl to General Timer 2.

Bit 4   Set General Timer 1 (Synchronous modes)
               0 = do not set General Timer 1.
               1 = load the value provided in RISRl to the high byte of General Timer 1.

               At the end of an interrupt service routine, the user can set a timer by setting a timer value in RISR. When the timer reaches zero, the CD2401 generates a modem/timer group interrupt to the host.

Bit 3   No Transfer of Data
               If no data is transferred from the receive FIFO during a receive interrupt, this bit must be set by the host.

Bits 2:0   DMA buffer gaps [2:0]
                  These bits set the size of the optional gaps to be left in DMA buffer starting at the
                  current location before resuming data transfer. The CD2401 moves its buffer address
                  pointer forward the selected number of bytes. It does 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 completes and the gap continues in the other receive buffer. If the discard
                  exception character is not selected, the character where the exception occurred is
                  written to the buffer following the gap.

8.5.3   Transmit Interrupt Registers

8.5.3.1   Transmit Priority Interrupt Level Register (TPILR)

Register Name: TPILR Register Description: Transmit Priority Interrupt Match Default Value: x'00 Access: Byte Read/Write				Intel Hex Address: x'E2 Motorola Hex Address: x'E0			
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
User-Assigned Priority Match Value							

This register must be initialized by the host to contain the codes that are presented on the address
bus by the host system to indicate which of the three CD2401 interrupt types (that is, modem,
transmit, or receive) is being acknowledged when IACKIN* is asserted. The CD2401 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 this register has no effect on the IACK cycle.

TPILR must contain the code used to acknowledge transmit interrupts.

Note: Bit 7 of this register always reads back as '0'. When each of the three PILRs are programmed with
          the same value, automatic internal prioritization is activated, with receive as the highest priority,
          followed by transmit and modem.

8.5.3.2   Transmit Interrupt Register (TIR)

Register Name: TIR Register Description: Transmit Interrupt Default Value: None, value varies Access: Byte Read only				Intel Hex Address: x'EE Motorola Hex Address: x'EC			
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Ten	Tact	Teoi	0	Tvct [1]	Tvct [0]	Tcn [1]	Tcn [0]

Bit 7   Transmit Enable
                  This bit is set by the CD2401 to initiate a transmit interrupt request sequence. It is
                  cleared during a valid transmit interrupt acknowledge cycle.

Bit 6   Transmit Active
This bit 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 TEOIR.

Bit 5   Transmit End of Interrupt
This bit is set automatically when the host CPU writes to TEOIR 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 complete.

Bit 4   Unused – always returns '0' when read.

Bits 3:2   Transmit Vector [1:0]
These bits are set by the CD2401 to provide the least-significant two bits of the vector supplied to the host CPU during an interrupt acknowledge cycle.
Transmit vector is decoded as: Tvct [1] = 1, and Tvct [0] = 0.

Bits 1:0   Transmit Channel Number [1:0]
These bits are set by the CD2401 to indicate the channel requiring transmit interrupt service.

8.5.3.3   Transmit Interrupt Status Register (TISR)

Register Name: TISR						Intel Hex Address: x'89	
Register Description: Transmit Interrupt Status						Motorola Hex Address: x'8A	
Default Value: x'00							
Access: Byte Read only							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Berr	EOF	EOB	UE	BA/BB	0	TxEmpty	TxDat

When the host receives a transmit interrupt, the following status are provided in this register.

Bit 7   Bus Error (written by the CD2401)
0 = no bus error.
1 = bus error detected on the last transfer.

Bit 6   Transmit End of Frame indication (DMA mode)
This interrupt occurs when the final data character of a transmit frame is transferred to the transmit FIFO.

Bit 5   Transmit End of Buffer indication (DMA mode)

Bit 4   Transmit underrun
               Data was not available in time during the synchronous frame transmission and an
               underrun error occurred. After a transmit underrun, all data up to the next EOF is dis-
               carded.

Bit 3   Buffer A or B interrupt
               0 = transmit buffer A.
               1 = transmit buffer B.

Bit 2   Reserved – always returns '0' when read.

Bit 1   Transmitter Empty
               All characters were completely transmitted and the serial output is idle.

Bit 0   Transmit Data
               This bit indicates that the number of characters in the FIFO is below the threshold.

8.5.3.4   Transmit FIFO Transfer Count (TFTC)

| Register Name: TFTC   Intel Hex Address: x'83 |
| Register Description: Transmit FIFO Transfer Count   Motorola Hex Address: x'80 |
| Default Value: x'00 |
| Access: Byte Read only |

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

Bits 7:5   Reserved – always returns '0' when read.

Bits 4:0   Transmit Data Count [4:0]
               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.

8.5.3.5   Transmit Data Register (TDR)

| Register Name: TDR   Intel Hex Address: x'F8 |
| Register Description: Transmit Data   Motorola Hex Address: x'F8 |
| Default Value: x'00 |
| Access: Byte Write only |

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

This 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 for
positioning valid data on the bus as outlined in Section 7.4 on page 90. If the BYTESWAP pin is
high, data must be valid on A/D[7:0]; if BYTESWAP is low, data must be valid on A/D[15:8]
because the TDR is on an even address.

8.5.3.6   Transmit End of Interrupt Register (TEOIR)

Register Name: TEOIR Register Description: Transmit End of Interrupt Default Value: x'00 Access: Byte Write only				Intel Hex Address: x'86 Motorola Hex Address: x'85			
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
TermBuff	EOF	SetTm2	SetTm1	Notrans	0	0	0

This register must be written to by the host transmit interrupt service routine to signal to the CD2401 that the current interrupt service is concluded. This must be the last access to the CD2401 during an interrupt service routine. A write to this register generates an internal end-of-interrupt signal that pops the CD2401 interrupt context stack.

Depending on the circumstances of an individual interrupt service, the host may be required to pass a parameter to the CD2401 through these registers. The CD2401 interprets the values written to this register at the completion of all receive interrupts as commands to be executed.

Bit 7   Terminate Buffer

1 = terminates buffer in DMA mode forces the current buffer to be discarded.

Note: If current interrupt is a transmit end-of-buffer interrupt, setting this bit at the end of the service routine causes the next buffer to also be terminated.

Bit 6   End of Frame (Synchronous modes)
This bit uses an interrupt-driven data transfer.
0 = this data transfer does not complete the frame/block.
1 = this data transfer completes the frame/block.

Bit 5   Set General Timer 2 (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 (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 can set a timer by setting a timer value in TISR. When the timer reaches zero, the CD2401 generates a modem/timer group interrupt to the host.

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

Bits 2:0   Reserved – *must* be zero.

8.5.4 Modem Interrupt Registers

8.5.4.1 Modem Priority Interrupt Level Register (MPILR)

Register Name: MPILR Register Description: Modem Priority Interrupt Match Default Value: x'00 Access: Byte Read/Write					Intel Hex Address: x'E1 Motorola Hex Address: x'E3		
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
User-Assigned Priority Match Value							

This register must be initialized by the host to contain the codes that are presented on the address bus by the host system to indicate which of the three CD2401 interrupt types (that is, modem, transmit, or receive) is being acknowledged when IACKIN* is asserted. The CD2401 compares bits 6:0 in this register with A[6:0] to determine if the acknowledge level is correct. The value programmed in the MSB of this 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 always reads back as '0'. When each of the three PILRs is programmed with the same value, automatic internal prioritization is activated, with receive as the highest priority, followed by transmit and modem.

8.5.4.2 Modem Interrupt Register (MIR)

Register Name: MIR Register Description: Modem Interrupt Default Value: x'00 Access: Byte Read only					Intel Hex Address: x'ED Motorola Hex Address: x'EF		
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Men	Mact	Meo	0	Mvct 1	Mvct 0	Mcn 1	Mcn 0

Bit 7   Modem Enable
This bit is set by the CD2401 to initiate a modem interrupt request sequence. It is cleared during a valid modem interrupt acknowledge cycle.

Bit 6   Modem Active
This bit is automatically set when Men is set and the Fair Share logic allows the assertion of a modem interrupt request. This bit is cleared when the host CPU writes to MEOIR.

Bit 5   Modem End of interrupt
This bit is automatically set when the host CPU writes to the MEOIR while in a modem interrupt routine.

Men	Mact	Meo	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 complete.

Bit 4   Unused – always returns '0' when read.

Bits 3:2   Modem Vector [1:0]
These bits are set by the CD2401 to provide the least-significant 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   Modem Channel Number [1:0]
These bits are set by the CD2401 to indicate the channel requiring modem interrupt service.

8.5.4.3   Modem (/Timer) Interrupt Status Register (MISR)

Register Name: MISR   Intel Hex Address: x'88
Register Description: Modem Interrupt Status   Motorola Hex Address: x'8B
Default Value: x'00
Access: Byte Read/Write

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DSRChg	CDChg	CTSChg	0	0	0	Timer2	Timer1

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

Bit 7   DSR Changed
1 = a change is detected on the DSR* input. The change detect is programmed in COR4 and COR5.

Bit 6   CD Changed
1 = a change is detected on the CD* input. The change detect is programmed in COR4 and COR5.

Bit 5   CTS Changed
1 = a change is detected on the CTS* input. The change detect is programmed in COR4 and COR5.

Bit 4:2   Reserved – always returns '0' when read.

Bit 1   General Timer 2 timed out
The count reached zero before being reset or disabled.

Bit 0   General Timer 1 timed out
The count reached zero before being reset or disabled.

During an interrupt service routine, the host can use this register to provide a binary timer value to one of the timers (Sync modes only), as detailed in MEOIR. The host can only load one of the two timers in each interrupt service routine.

8.5.4.4   Modem End of Interrupt Register (MEOIR)

Register Name: MEOIR
Register Description: Modem End of Interrupt
Default Value: x'00
Access: Byte Write only

Intel Hex Address: x'85
Motorola Hex Address: x'86

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

Bits 7:6   Reserved – *must* be '0'.

Bit 5   Set General Timer 2 (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 (Synchronous modes)
                0 = do not set General Timer 1.
                1 = load the value provided in MISR to the high byte of General Timer 1.

Bits 3:0   Reserved – *must* be '0'.

At the end of an interrupt service routine, the user can set one of the timers by setting a timer value in the MISR and setting the appropriate bit in this register. When the timer reaches zero, the CD2401 generates a modem/timer group interrupt to the host.

8.6   DMA Registers

8.6.1   DMA Mode Register (DMR)

Register Name: DMR
Register Description: DMA Mode
Default Value: x'00
Access: Byte Write only

Intel Hex Address: x'F4
Motorola Hex Address: x'F6

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

This register is write only. No misoperation occurs if the register is read, but the read value is not consistent.

Bit 7   Internal DTACK* Synchronization Enable
                  If external synchronization of DTACK* with BUSCLK is not provided, an internal
                  synchronization can be enabled by setting this bit (revision M and later).

Bits 6:4   Reserved – *must* be '0'; reads back as '0'.

Bit 3   Byte DMA
                  0 = The CD2401 attempts 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 CD2401 always performs 8-bit DMA transfers, the position of the data on
                  the bus still follows the normal rules relating to the BYTESWAP pin.

Bits 2:0   Reserved – *must* be '0'; reads back as '0'.

8.6.2   Bus Error Retry Count (BERCNT) Register

Register Name: BERCNT Register Description: Bus Error Retry Count Default Value: x'00 Access: Byte Read/Write						Intel Hex Address: x'8D Motorola Hex Address: x'8E	
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Binary Value							

When this register is programmed to '0', 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 CD2401 retries the
same DMA operation and decrements the register value by one. When the value reaches zero, the
next bus error causes an interrupt, at which time a new count can be loaded by the host CPU.

8.6.3   DMA Buffer Status (DMABSTS) Register

Register Name: DMABSTS Register Description: DMA Buffer Status Default Value: x'00 Access: Byte Read only						Intel Hex Address: x'1A Motorola Hex Address: x'19	
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
TDAlign	RstApd	CrtBuf	Append	Ntbuf	Tbusy	Nrbuf	Rbusy

When the CD2401 requires an external buffer for DMA transfers, it checks Ntbuf/Nrbuf bits to
decide which buffer to use. Once the CD2401 starts using the buffer, it toggles Ntbuf/Nrbuf bits
and sets Tbusy/Rbusy bits. At system initialization, Ntbuf and Nrbuf are set to buffer A.

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

Bit 6   Reset Append
               This bit is set after the terminate append buffer command in STCR is recognized.
               This bit is cleared after the remaining data is flushed from the buffer.

Bit 5   Current Transmit Buffer
               This bit is used internally to mark the actual buffer in use.

Bit 4   Append (only buffer A can be used as an append buffer)
               This bit is the transmit append buffer usage indicator.
               0 = Append buffer is not in use.
               1 = Append buffer is in use.

Bit 3   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 (that is, when data is
               first read from buffer A). This bit is set to indicate that buffer B is next.

Bit 2   Current Transmit Buffer Busy
               0 = no buffer in use.
               1 = current transmit buffer in use.

Bit 1   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 (that is, when data is
               first written to buffer A). This bit is set to indicate buffer B is next.

Bit 0   Current Receive Buffer Busy
               0 = no buffer in use.
               1 = current receive buffer in use.

8.6.4   DMA Receive Registers

8.6.4.1   A Receive Buffer Address – Lower (ARBADRL) Register

Register Name: ARBADRL						Intel Hex Address: x'40	
Register Description: Receive Buffer 'A' 32-bit Address – lower word						Motorola Hex Address: x'42	
Default Value: x'0000							
Access: Word Read/Write							
Bit 15	*Bit 14*	*Bit 13*	*Bit 12*	*Bit 11*	*Bit 10*	*Bit 9*	*Bit 8*
Binary Address Value, 32-bit Address bits 15:8							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Binary Address Value, 32-bit Address bits 7:0							

8.6.4.2 A Receive Buffer Address – Upper (ARBADRU) Register

Register Name: ARBADRU Register Description: Receive Buffer 'A' 32-bit Address – upper word Default Value: x'0000 Access: Word Read/Write				Intel Hex Address: x'42 Motorola Hex Address: x'40			
Bit 15	*Bit 14*	*Bit 13*	*Bit 12*	*Bit 11*	*Bit 10*	*Bit 9*	*Bit 8*
Binary Address Value, 32-bit Address bits 23:16							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Binary Address Value, 32-bit Address bits 31:24							

This register contains the start addresses of the A external buffer used by the CD2401 to store the receive data block. This register is written to by the host and copied internally to control the data transfer to the memory.

8.6.4.3 B Receive Buffer Address – Lower (BRBADRL) Register

Register Name: BRBADRL Register Description: Receive Buffer 'B' 32-bit Address – lower word Default Value: x'0000 Access: Word Read/Write				Intel Hex Address: x'44 Motorola Hex Address: x'46			
Bit 15	*Bit 14*	*Bit 13*	*Bit 12*	*Bit 11*	*Bit 10*	*Bit 9*	*Bit 8*
Binary Address Value, 32-bit Address, bits 15:8							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Binary Address Value, 32-bit Address, bits 7:0							

8.6.4.4 B Receive Buffer Address – Upper (BRBADRU) Register

Register Name: BRBADRU Register Description: Receive Buffer 'B' 32-bit Address – upper word Default Value: x'0000 Access: Word Read/Write					Intel Hex Address: x'46 Motorola Hex Address: x'44		
Bit 15	*Bit 14*	*Bit 13*	*Bit 12*	*Bit 11*	*Bit 10*	*Bit 9*	*Bit 8*
Binary Address Value, 32-bit Address, bits 23:16							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Binary Address Value, 32-bit Address, bits 31:24							

This register contains the start addresses of the B external buffer used by the CD2401 to store the receive data block. This register is written to by the host and copied internally to control the data transfer to the memory.

8.6.4.5 A Buffer Receive Byte Count (ARBCNT) Register

Register Name: ARBCNT Register Description: Receive Buffer A Byte Count Default Value: x'0000 Access: Word Read/Write					Intel Hex Address: x'48 Motorola Hex Address: x'4A		
Bit 15	*Bit 14*	*Bit 13*	*Bit 12*	*Bit 11*	*Bit 10*	*Bit 9*	*Bit 8*
Binary Count Value, 16-bit Count bits 15:8							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Binary Count Value, 16-bit Count bits 7:0							

8.6.4.6   B Buffer Receive Byte Count (BRBCNT) Register

Register Name: BRBCNT Register Description: Receive Buffer B Byte Count Default Value: x'0000 Access: Word Read/Write				Intel Hex Address: x'4A Motorola Hex Address: x'48			
Bit 15	*Bit 14*	*Bit 13*	*Bit 12*	*Bit 11*	*Bit 10*	*Bit 9*	*Bit 8*
Binary Count Value, 16-bit Count bits 15:8							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Binary Count Value, 16-bit Count bits 7:0							

These registers contain the number of bytes stored in the external data buffers by the CD2401. The count is updated after a block of data is moved to memory and the buffer is terminated. As initially written by the host, these registers contain the number of bytes that the buffer can hold.

8.6.4.7   A Receive Buffer Status (ARBSTS) Register

Register Name: ARBSTS Register Description: Receive Buffer A Status Default Value: x'00 Access: Byte Read/Write				Intel Hex Address: x'4C Motorola Hex Address: x'4F			
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Berr	EOF	EOB	0	0	0	0	2401own

8.6.4.8   B Receive Buffer Status (BRBSTS) Register

Register Name: BRBSTS Register Description: Receive Buffer B Status Default Value: x'00 Access: Byte Read/Write				Intel Hex Address: x'4D Motorola Hex Address: x'4E			
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Berr	EOF	EOB	0	0	0	0	2401own

These registers contain the current status of associated receive buffers and enable the buffers to be passed between the host and the CD2401.

Bit 7   Bus Error (set by the CD2401 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 CD2401 and cleared by the host CPU)
                 0 = this buffer does not terminate a frame.
                 1 = this buffer terminates a frame.

Bit 5   End of Buffer (set by the CD2401 and cleared by the host CPU)
                 0 = buffer not complete.
                 1 = buffer complete.
                 When the EOB bit is set by the CD2401, 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).

Bits 4:1   Reserved – *must* be '0'; reads as '0'.

Bit 0   Ownership of the Transfer buffer (set by the host CPU and cleared by the CD2401)
                 0 = buffer not free for use by the CD2401.
                 1 = buffer free for use by the CD2401.

8.6.4.9   Receive Current Buffer Address – Lower (RCBADRL) Register

Register Name: RCBADRL   Intel Hex Address: x'3C
Register Description: Current Receive Buffer Address – lower word   Motorola Hex Address: x'3E
Default Value: x'0000
Access: Word Read only

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Binary Address Value, 32-bit Address bits 15:8							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Binary Address Value, 32-bit Address bits 7:0							

8.6.4.10   Receive Current Buffer Address – Upper (RCBADRU) Register

Register Name: RCBADRU   Intel Hex Address: x'3E
Register Description: Current Receive Buffer Address – upper word   Motorola Hex Address: x'3C
Default Value: x'0000
Access: Word Read only

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Binary Address Value, 32-bit Address bits 31:24							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Binary Address Value, 32-bit Address bits 23:16							

These registers contain the address of the current DMA buffer being used for receive data. These
registers are updated at the end of receive data transfers. These registers are *only* for the use of the
CD2401 to manage DMA transfers. In Asynchronous mode, the host can read these registers

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 needs this information to process newly arrived data in the buffer if used in Append mode and a data timeout has occurred. The address is also needed if an exception has occurred and a gap is to be left in the DMA buffer (see the description of the Gap[x] bits in Section 8.5.2.7 on page 134) for the insertion of status information by the host. For a bus error during receive data transfer, this register provides the start address of the transfer causing the bus error.

8.6.5 DMA Transmit Registers

8.6.5.1 A Transmit Buffer Address – Lower (ATBADRL) Register

Register Name: ATBADRL
Register Description: Transmit Buffer A 32-bit Address – lower word
Default Value: x'0000
Access: Word Read/Write

Intel Hex Address: x'50
Motorola Hex Address: x'52

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Binary Address Value, 32-bit Address bits 15:8							

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Binary Address Value, 32-bit Address bits 7:0							

8.6.5.2 A Transmit Buffer Address – Upper (ATBADRU) Register

Register Name: ATBADRU
Register Description: Transmit Buffer A 32-bit Address – upper word
Default Value: x'0000
Access: Word Read/Write

Intel Hex Address: x'52
Motorola Hex Address: x'50

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Binary Address Value, 32-bit Address bits 23:16							

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Binary Address Value, 32-bit Address bits 31:24							

This register holds the start addresses of the A external buffer used by the CD2401 to transmit the next data block. This register is written by the host and copied internally to control the data transfer from the memory to the CD2401 FIFO.

8.6.5.3 B Transmit Buffer Address – Lower (BTBADRL) Register

Register Name: BTBADRL Register Description: Transmit Buffer B 32-bit Address – lower word Default Value: x'0000 Access: Word Read/Write					Intel Hex Address: x'54 Motorola Hex Address: x'56		
Bit 15	*Bit 14*	*Bit 13*	*Bit 12*	*Bit 11*	*Bit 10*	*Bit 9*	*Bit 8*
Binary Address Value, 32-bit Address bits 15:8							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Binary Address Value, 32-bit Address bits 7:0							

8.6.5.4 B Transmit Buffer Address – Upper (BTBADRU) Register

Register Name: BTBADRU Register Description: Transmit Buffer B 32-bit Address – upper word Default Value: x'0000 Access: Word Read/Write					Intel Hex Address: x'56 Motorola Hex Address: x'54		
Bit 15	*Bit 14*	*Bit 13*	*Bit 12*	*Bit 11*	*Bit 10*	*Bit 9*	*Bit 8*
Binary Address Value, 32-bit Address bits 23:16							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Binary Address Value, 32-bit Address bits 31:24							

This register holds the start addresses of the B external buffer used by the CD2401 to transmit the next data block. This register is written by the host and copied internally to control the data transfer from the memory to the CD2401 FIFO.

8.6.5.5 A Buffer Transmit Byte Count (ATBCNT) Register

Register Name: ATBCNT Register Description: Transmit Buffer A Byte Count Default Value: x'0000 Access: Word Read/Write					Intel Hex Address: x'58 Motorola Hex Address: x'5A		
Bit 15	*Bit 14*	*Bit 13*	*Bit 12*	*Bit 11*	*Bit 10*	*Bit 9*	*Bit 8*
Binary Count Value, 16-bit Count bits 15:8							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Binary Count Value, 16-bit Count bits 7:0							

8.6.5.6 B Buffer Transmit Byte Count (BTBCNT) Register

Register Name: BTBCNT Register Description: Transmit Buffer B Byte Count Default Value: x'0000 Access: Word Read/Write					Intel Hex Address: x'5A Motorola Hex Address: x'58		
Bit 15	*Bit 14*	*Bit 13*	*Bit 12*	*Bit 11*	*Bit 10*	*Bit 9*	*Bit 8*
Binary Count Value, 16-bit Count bits 15:8							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Binary Count Value, 16-bit Count bits 7:0							

These registers contain the count of bytes to be transmitted from the respective buffer.

8.6.5.7 A Transmit Buffer Status (ATBSTS) Register

Register Name: ATBSTS Register Description: Transmit Buffer A Status Register Default Value: x'00 Access: Byte Read/Write					Intel Hex Address: x'5C Motorola Hex Address: x'5F		
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Berr	EOF	EOB	UE	Append	0	INTR	2401own

8.6.5.8 B Transmit Buffer Status (BTBSTS) Register

Register Name: BTBSTS		Intel Hex Address: x'5D
Register Description: Transmit Buffer B Status Register		Motorola Hex Address: x'5E
Default Value: x'00		
Access: Byte Read/Write		

Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Berr	EOF	EOB	UE	Append	0	INTR	2401own

These registers contain the status of the associated transmit buffer and enables successive buffers to be passed between the host and the CD2401.

Bit 7   Bus Error (set by the CD2401 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   End of a transmit buffer reached (DMA supported transfers only)

The end of one of the host-supplied transmit buffers has been reached. This bit is set by the CD2401 and cleared by the host CPU.

Bit 4   Underrun Error
When this bit is set it indicates that a transmit underrun occurred because the buffer was not available.

Bit 3   Append (Asynchronous mode; set and cleared by the host CPU)
0 = no data is appended to the buffer.
1 = data can be appended to buffer after transfer starts.

Bit 2   Reserved – *must* be '0'; reads as '0'.

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

Bit 0   Ownership of transfer buffer (set by the host CPU and cleared by the CD2401)
0 = buffer not ready to be used by the CD2401.
1 = buffer is ready for the CD2401 to transmit.

To start transmission of a buffer, the host must set the A/BTBADR and A/BTBCNT registers, and then set the 2401own bit. If the CD2401 is to generate and send the CRC for the frame, the FCSApd (COR2[6]) 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 is set by the CD2401, and the 2401own bit is reset allowing a new buffer to be allocated.

When the Append is bit data can be added to the buffer after transmission begins. In this mode, the host sets ATADR and ATCNT as normal, but when new data is appended to the buffer, the A/BTBCNT can be updated. When the A buffer is used in Append mode, the CD2401 does not set the

EOB bit. When the host completes use of the buffer, it must issue the append complete command through STCR. The CD2401, upon transmitting the last characters from the buffer, sets EOB, thus allowing the host to allocate a new transmit buffer.

8.6.5.9 Transmit Current Buffer Address Register – Lower (TCBADRL)

Register Name: TCBADRL						Intel Hex Address: x'38	
Register Description: Current Transmit Buffer Address – lower word						Motorola Hex Address: x'3A	
Default Value: x'0000							
Access: Word Read only							
Bit 15	*Bit 14*	*Bit 13*	*Bit 12*	*Bit 11*	*Bit 10*	*Bit 9*	*Bit 8*
Binary Address Value, 32-bit Address bits 15:8							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Binary Address Value, 32-bit Address bits 7:0							

8.6.5.10 Transmit Current Buffer Address Register – Upper (TCBADRU)

Register Name: TCBADRU						Intel Hex Address: x'3A	
Register Description: Current Transmit Buffer Address – upper word						Motorola Hex Address: x'38	
Default Value: x'0000							
Access: Word Read only							
Bit 15	*Bit 14*	*Bit 13*	*Bit 12*	*Bit 11*	*Bit 10*	*Bit 9*	*Bit 8*
Binary Address Value, 32-bit Address bits 31:24							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Binary Address Value, 32-bit Address bits 23:16							

These registers contain the address into the current DMA buffer being used for transmit data. This address is updated at the end of transmit data transfers. In the case of a bus error during transmit data transfer, this register contains the starting address of the transfer causing the bus error.

intel

8.7 Timer Registers

8.7.1 Timer Period Register (TPR)

Register Name: TPR						Intel Hex Address: x'D8	
Register Description: Timer Period						Motorola Hex Address: x'DA	
Default Value: x'FF							
Access: Byte Read/Write							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Binary Value							

This register provides the initialization value for the timer prescaler that 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 RTPR, TTR, and 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.

8.7.2 Receive Timeout Period Register (RTPR) – Async Mode only

Register Name: RTPR						Intel Hex Address: x'26	
Register Description: Receive Timeout Period, 16-bits						Motorola Hex Address: x'24	
Default Value: x'0000							
Access: Word Read/Write – Async mode only							
Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Binary Value bits 15:8							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Binary Value bits 7:0							

8.7.2.1 Receive Timeout Period Register – Low (RTPRI) – Async Mode only

Register Name: RTPRI						Intel Hex Address: x'26	
Register Description: Receive Timeout Period – low byte						Motorola Hex Address: x'25	
Default Value: x'00							
Access: Byte Read/Write – Async mode only							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Binary Value							

8.7.2.2   Receive Timeout Period Register – High (RTPRh) – Async Mode only

Register Name: RTPRh Register Description: Receive Timeout Period – high byte Default Value: x'00 Access: Byte Read/Write – Async mode only						Intel Hex Address: x'27 Motorola Hex Address: x'24	
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
RxEn	RxFloff	RxFlon	0	TxEn	TxFloff	TxFlon	0

The value in this register sets the receive data timeout period in Async mode; this register is used as general-purpose timers in Sync modes (see Section 8.7.3). As each character is moved into 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 RTPR. The receive timer decrements on each 'tick' of the prescaler counter, this period is controlled in TPR. If the receive timer reaches zero, it causes a receive data interrupt.

The register can be accessed as a single 16-bit word, or as two byte values.

8.7.3   General Timer 1 (GT1) Register – Sync Modes only

Register Name: GT1 Register Description: General Timer 1 Default Value: x'00 Access: Word Read/Write						Intel Hex Address: x'28 Motorola Hex Address: x'2A	
Bit 15	*Bit 14*	*Bit 13*	*Bit 12*	*Bit 11*	*Bit 10*	*Bit 9*	*Bit 8*
Binary Value bits 15:8							
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Binary Value bits 7:0							

8.7.3.1   General Timer 1 – Low (GT1l) – Sync Modes only

Register Name: GT1l Register Description: General Timer 1 – low byte Default Value: x'00 Access: Byte Read/Write						Intel Hex Address: x'28 Motorola Hex Address: x'2B	
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Binary Value							

8.7.3.2   General Timer 1 – High (GT1h) – Sync Modes only

Register Name: GT1h Register Description: General Timer 1 – high byte Default Value: x'00 Access: Byte Read/Write					Intel Hex Address: x'29 Motorola Hex Address: x'2A		
Bit 15	*Bit 14*	*Bit 13*	*Bit 12*	*Bit 11*	*Bit 10*	*Bit 9*	*Bit 8*
Binary Value							

This 16-bit timer can be started by the user whenever it is inactive by writing a 16-bit timeout value to this register. When this register is non-zero, it decrements on each prescaler clock 'tick'. When it reaches zero, a modem/timer group interrupt is generated to the host. The timer can be disabled by a CCR command. In addition, during an interrupt the user can 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. In a single-interrupt routine, only one general timer can be restarted this way.

8.7.4   General Timer 2 (GT2) Register — Sync Modes only

Register Name: GT2 Register Description: General Timer 2 Default Value: x'00 Access: Byte Read/Write					Intel Hex Address: x'2A Motorola Hex Address: x'29		
Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
Binary Value							

This 8-bit timer can be started by the user whenever it is inactive by writing an 8-bit timeout value to the timer register. When this register is non-zero, it decrements on each prescaler clock 'tick'. When it reaches zero, a modem/timer group interrupt is generated to the host. The timer can be disabled by a CCR command if the timer's current value is greater than '1'. In addition, during a receive or transmit interrupt, the user can 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. In a single-interrupt routine, only one general timer can be restarted this way.

8.7.5   Transmit Timer Register (TTR) — Async Modes only

Register Name: TTR Intel Hex Address: x'2A
Register Description: Transmit Timer Motorola Hex Address: x'29
Default Value: x'00
Access: Byte Read only

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Binary Value							

This Asynchronous mode timer is managed by the CD2401 to implement embedded transmit delays when that option is used by the host (see description of COR2). This register should not be modified by the host under any circumstances.

intel

9.0 Electrical Specifications

Note: Verify with your local sales office that you have the latest datasheet before finalizing a design.

9.1 Absolute Maximum Ratings

Operating ambient temperature (T_A) 0°C to 70°C

Storage temperature −65°C to 150°C

All voltages with respect to ground −0.5 V to V_{CC} +0.5 V (volts)

Supply voltage (V_{CC}) +7.0 V

Power dissipation 0.25 W (watt)

Note: Stresses above those listed under Absolute Maximum Ratings can 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 can affect device reliability.

9.2 DC Electrical Characteristics

(@ $V_{CC} = 5$ V $\pm$ 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 except CLK, RESET*, and BGIN*)	2.0		V_{CC}	V
V_{IH}	Input high voltage for CLK, RESET*, and BGIN*	2.7	V_{CC}	V	
V_{OL}	Output low voltage	0.4		V	$I_{OL} = 2.4$ mA (I_{OL} for OD pins = 10 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 tristate 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 = 35 MHz
C_{IN}	Input capacitance		10	pF	
C_{OUT}	Output capacitance		10	pF	

Note: The maximum CLK of 35 MHz applies to Revision M and later devices only; Revisions K and L remain specified at 33 MHz maximum, revisions prior to K remain specified at 20 MHz maximum. All values in the following tables apply to the 35-MHz specification.

9.3 AC Electrical Characteristics

Symbol	Parameter	MIN	MAX
t_{PERIOD}	Period of CLK input (35 MHz maximum)	28.57	
t_1	CLK high to BUSCLK high		20
t_2	CLK high to BUSCLK low		20
Bus Arbitration			
t_{11}	CLK high to BGACK* tristate		25
t_{12}	BGIN* low to address valid[1]		40
t_{13}	Address hold after CLK high	0	
t_{14}	CLK high to address tristate		25
t_{15}	CLK high to ADLD* low		25
t_{16}	CLK high to ADLD* high		20
t_{17}	Address setup to ADLD* high	15	
t_{18}	CLK high to AEN*/DATEN*/DATDIR* high		25
t_{19}	CLK high to AEN*/DATEN*/DATDIR* tristate		25
t_{20}	CLK high to AEN*/DATEN*/DATDIR* low		25
DMA Read			
t_{21}	Data setup to CLK high	10	
t_{22}	Data hold after CLK high	15	
t_{23}	CLK high to address valid		30
t_{24}	CLK low to AS* low		25
t_{25}	CLK high to AS* high		20
t_{26}	CLK low to DS* low		25
t_{27}	CLK high to DS* high		20
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		40
t_{32}	Data hold after CLK high	0	
t_{33}	CLK low to DS* low		25
t_{34}	CLK high to DS* high		20
t_{35}	DTACK* low setup to CLK high	10	
t_{36}	DTACK* high setup to CLK high (to avoid false termination)	50	
Host Read/Write			
t_{41}	DS* and CS* low setup to CLK high	7	
t_{42}	Reserved		
t_{43}	Reserved		

intel

Symbol	Parameter	MIN	MAX
t_{44}	R/W* setup to CLK high	5	
t_{45}	CLK high to data valid		25
t_{46}	Data setup time to CLK high	6	
t_{47}	Data hold time after CLK high	15	
t_{48}	Address setup time to CLK high	5	
t_{49}	Address hold time after CLK high	15	
t_{50}	CLK high to DTACK* low (read cycle)		25
t_{51}	CLK high to DTACK* low (write cycle)		25
t_{52}	(CS* and DS*) low to DATEN*/DATDIR* low		28
t_{53}	DS* high to DATEN*/DATDIR* tristate		25
t_{54}	DS* high to data bus tristate		25
t_{55}	DS* high to DTACK* high-impedance		25
Interrupt Acknowledge			
t_{61}	CLK high to IACKIN*, DS* setup	20	
t_{63}	CLK high to data valid		35
t_{64}	Address setup to IACKIN* low	0	
t_{65}	Address hold after IACKIN* high	0	
t_{66}	CLK high to DTACK* low		25
t_{67}	(IACKIN* and DS*) low and BUSCLK high to DATEN* and DATDIR* low		40

1. This timing assumes the following conditions: BGACK* high, DTACK* high, DS* high, and BUSCLK high.

Figure 17. CLK/BUSCLK/RESET* Timing Relationship

During RESET* active period, BUSCLK is held low. BUSCLK will transition high and begin running at one/half CLK frequency on the first rising edge of CLK after RESET* is released.
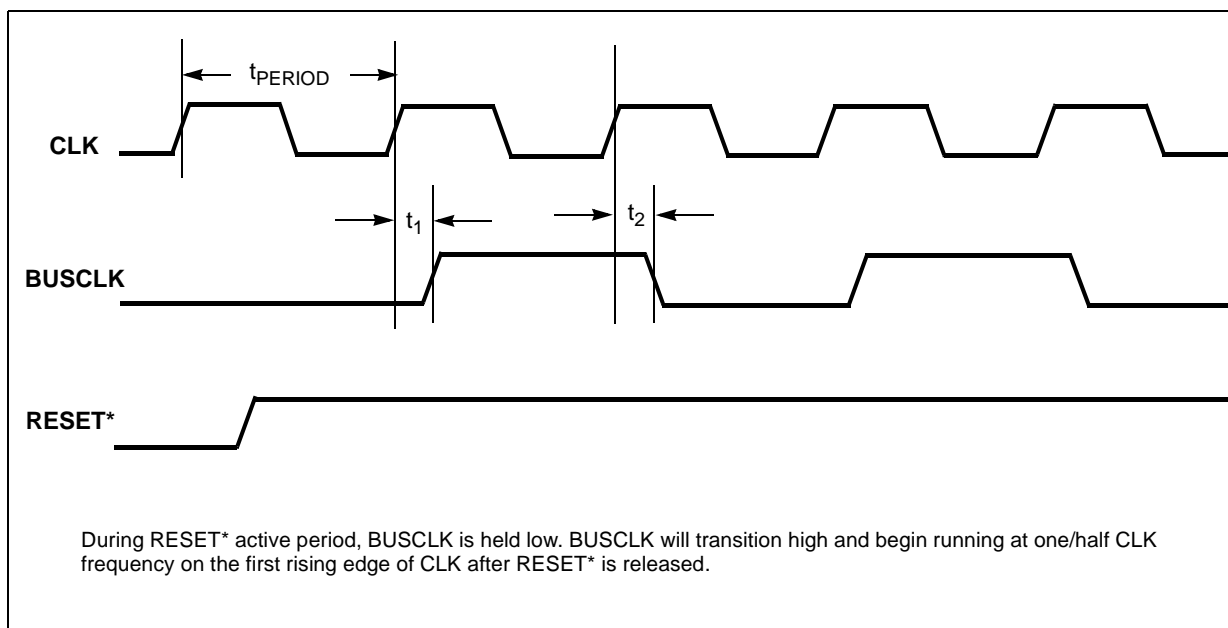
Figure 18. Slave Read Cycle Timing
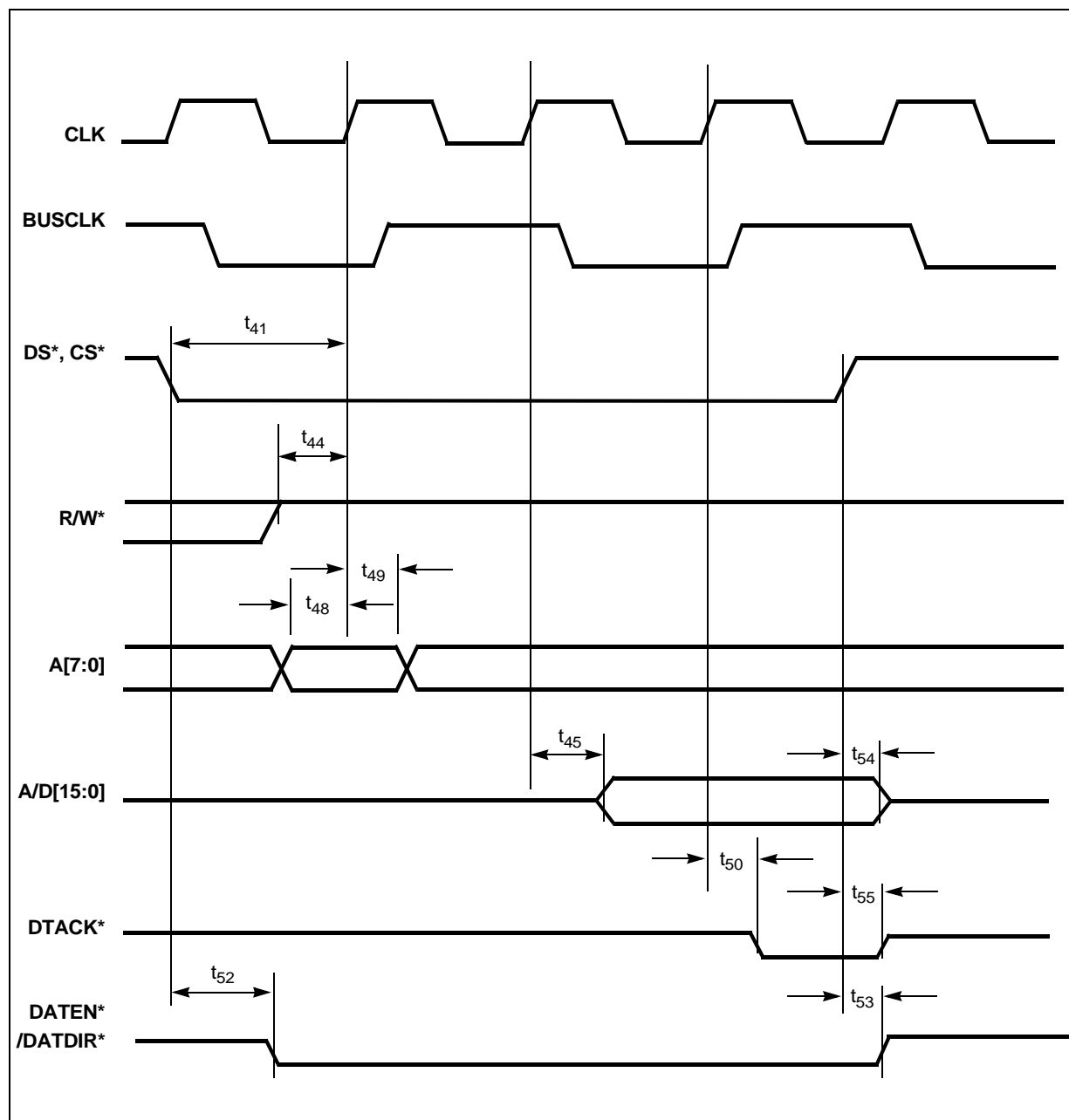
Figure 19. Slave Write Cycle Timing
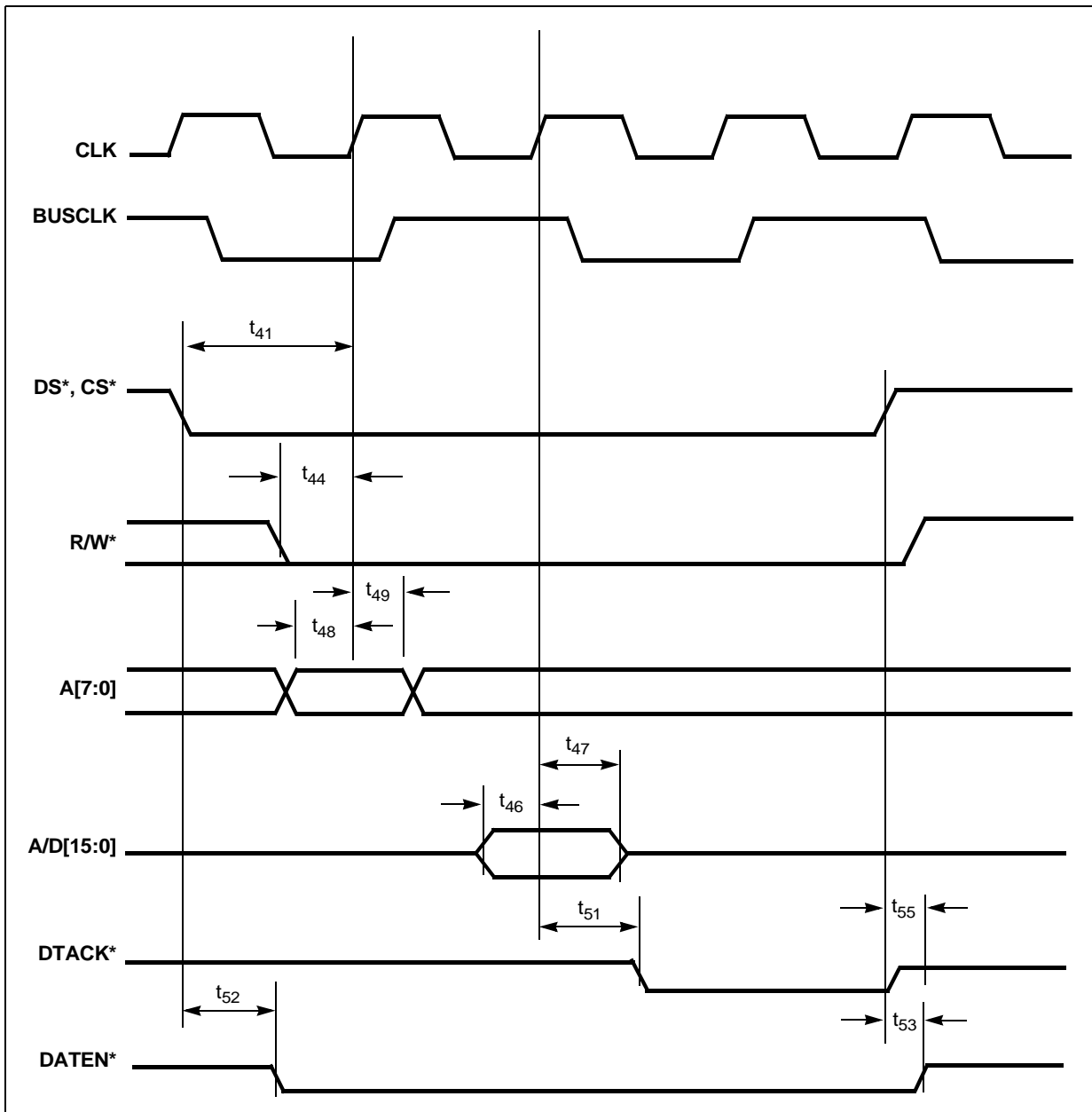
Figure 20. Interrupt Acknowledge Cycle Timing

intel
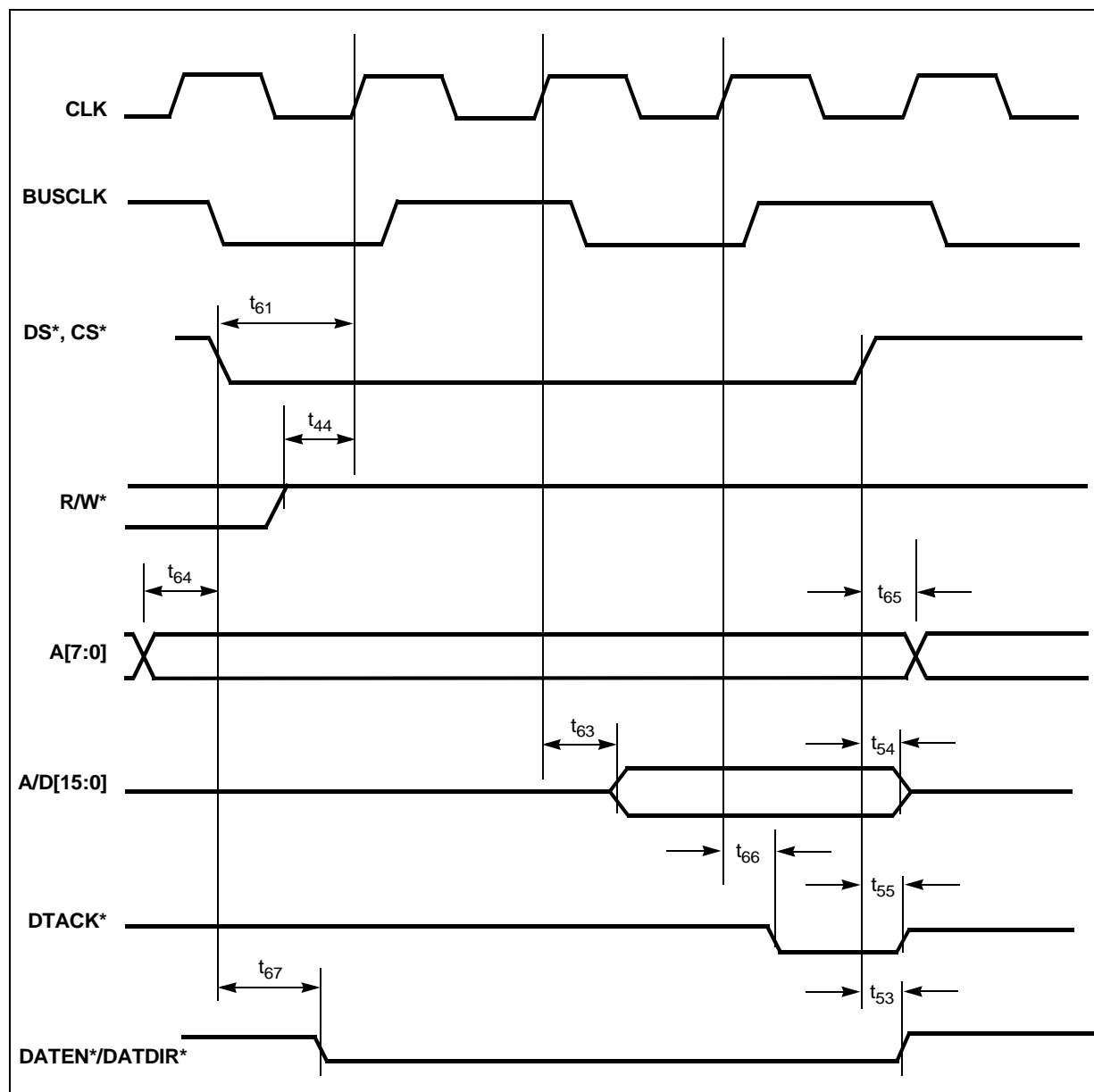
Figure 21. Bus Arbitration Cycle Timing

NOTE: In DMA Read cycle, these pins will be tristate; in DMA Write cycle, these pins will be D[15:0].
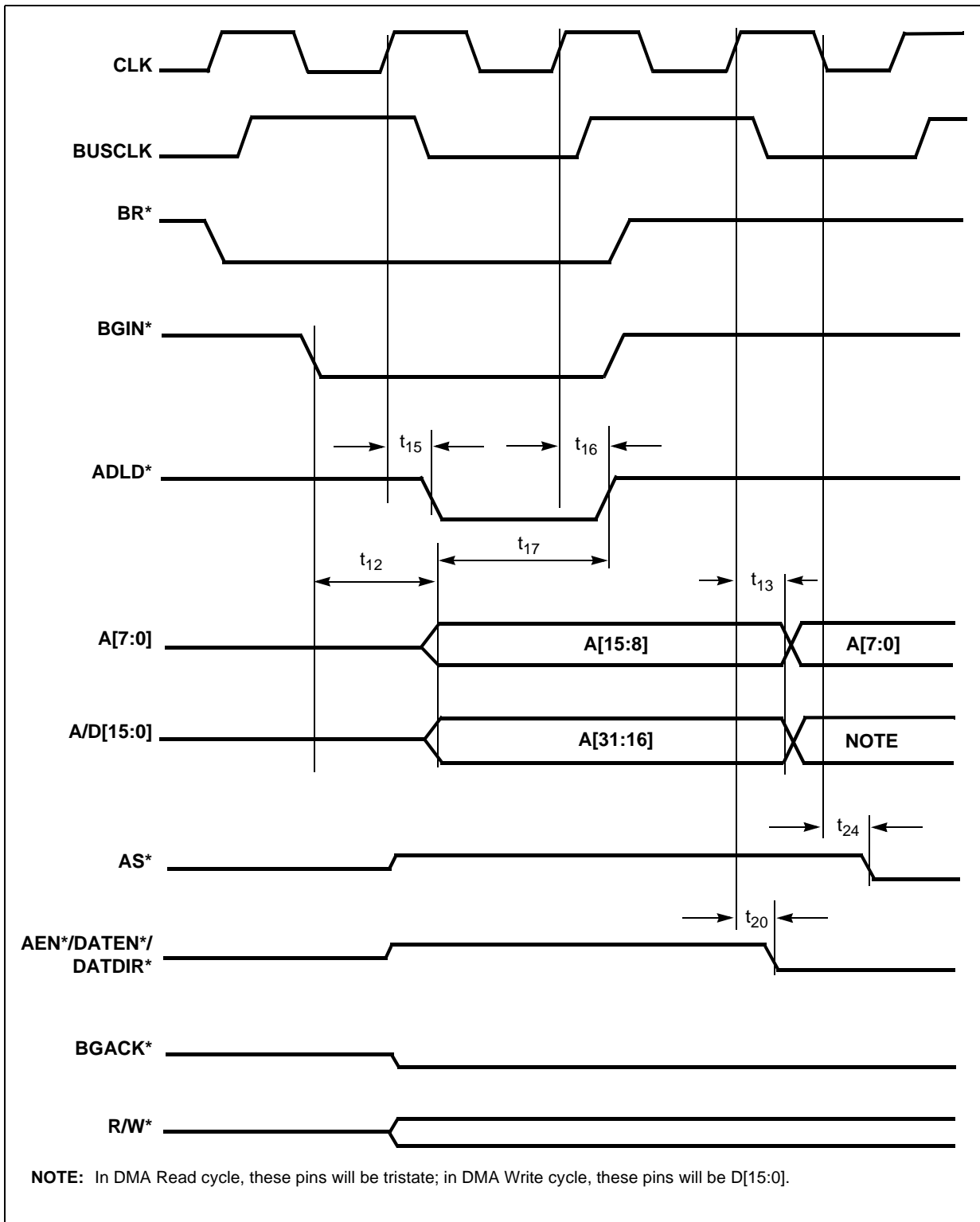
Figure 22. Bus Release Timing
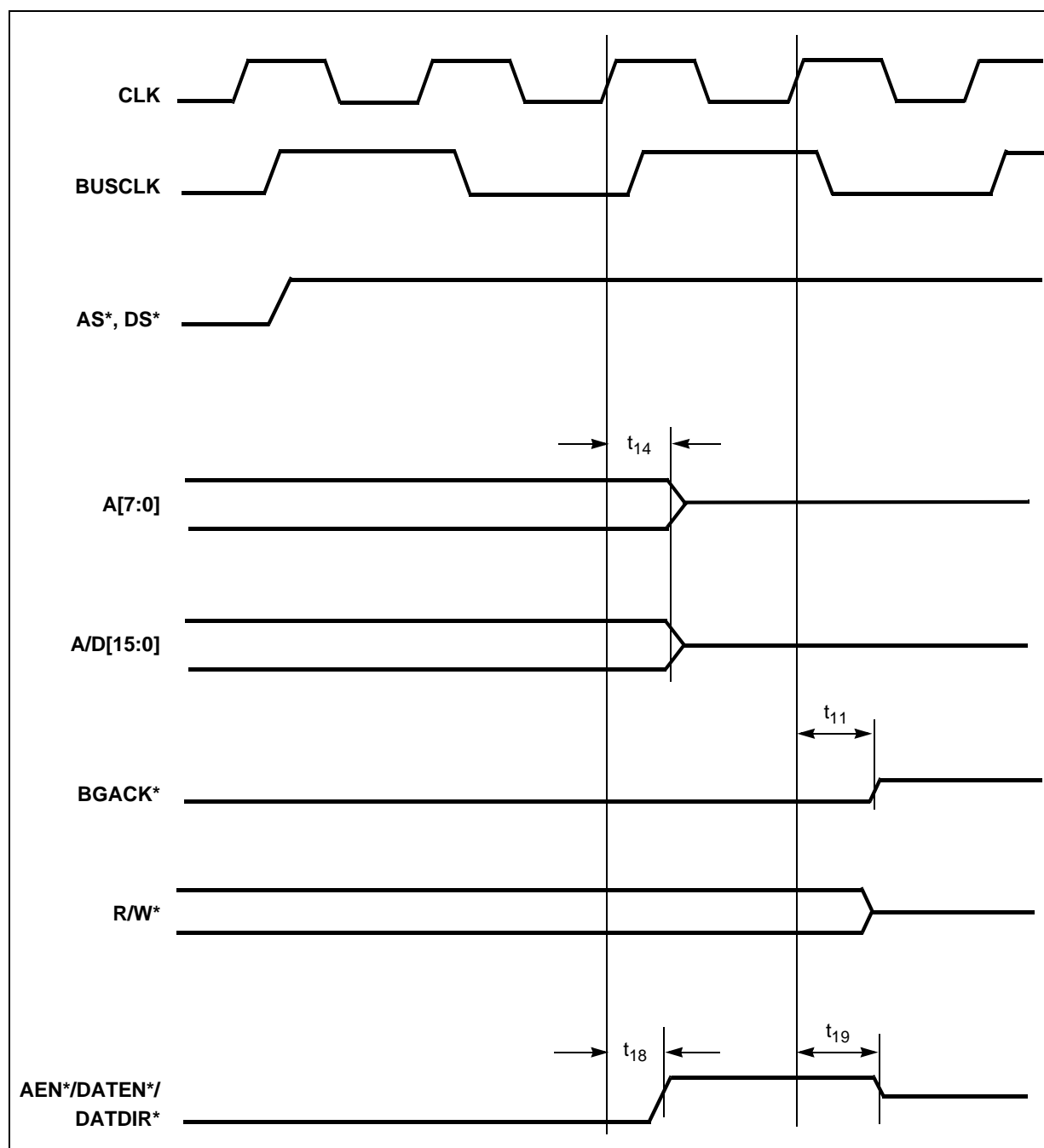
Figure 23. DMA Read Cycle Timing

BERR* Timing:
t_S = setup time to CLK rising edge = 10 ns
t_H = hold time after CLK rising edge = 20 ns
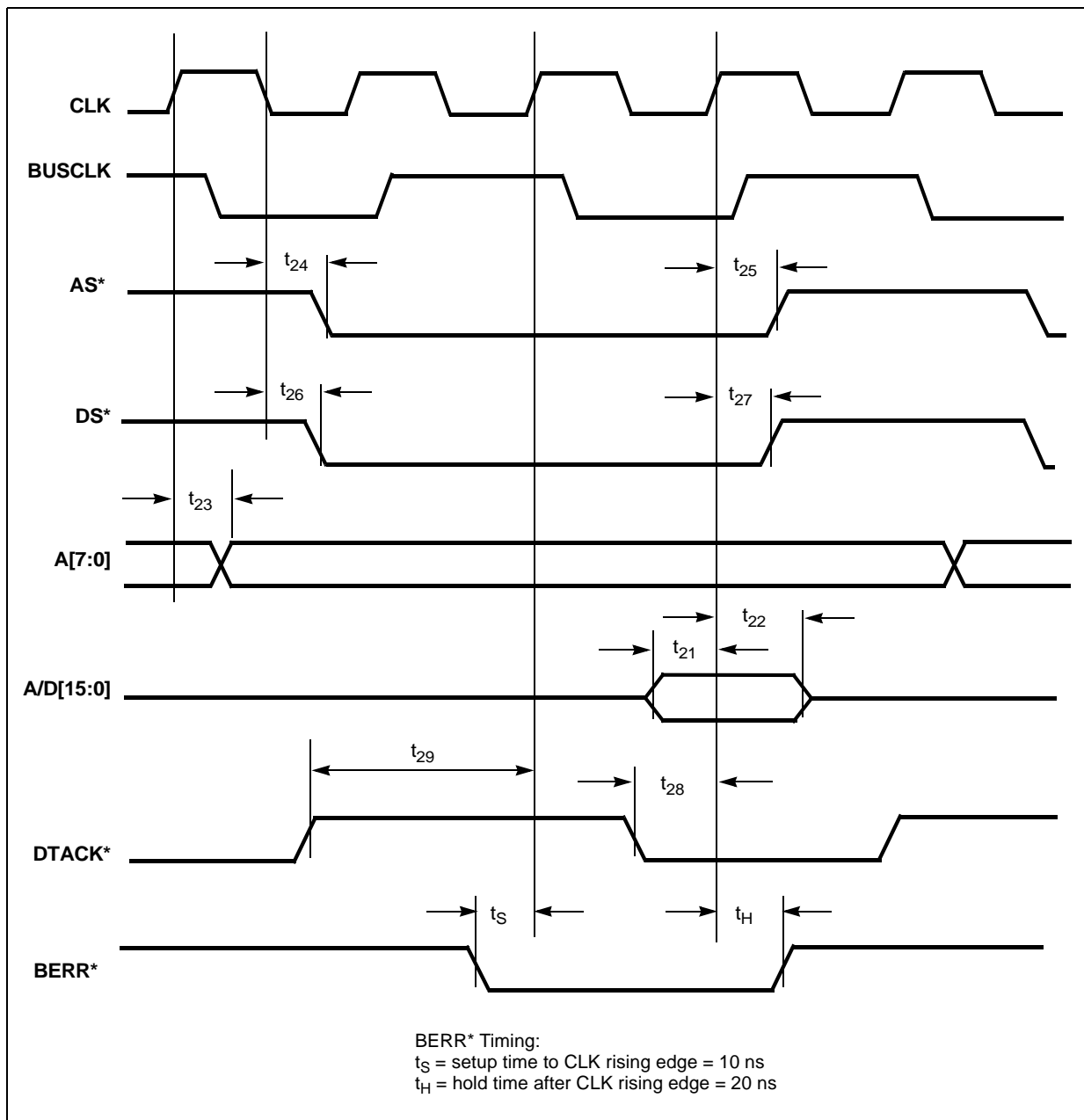
Figure 24. DMA Write Cycle Timing

BERR* Timing:
t_S = setup time to CLK rising edge = 10 ns
t_H = hold time after CLK rising edge = 20 ns

10.0 Package Specifications

CD2401
100-Pin MQFP (JEDEC)

Pin 1 Indicator

Pin 100

Pin 1

22.95 (0.904)
23.45 (0.923)

19.90 (0.783)
20.10 (0.791)

0.22 (0.009)
0.38 (0.015)

0.65
(0.0256)
BSC

13.90 (0.547)
14.10 (0.555)

16.95 (0.667)
17.45 (0.687)

0.65 (0.026)
0.95 (0.037)

2.57 (0.101)
2.87 (0.113)

1.60 (0.063) REF

0.13 (0.005)
0.23 (0.009)

0° MIN
7° MAX

3.40
(0.134)
MAX

0.25
(0.010)
MIN
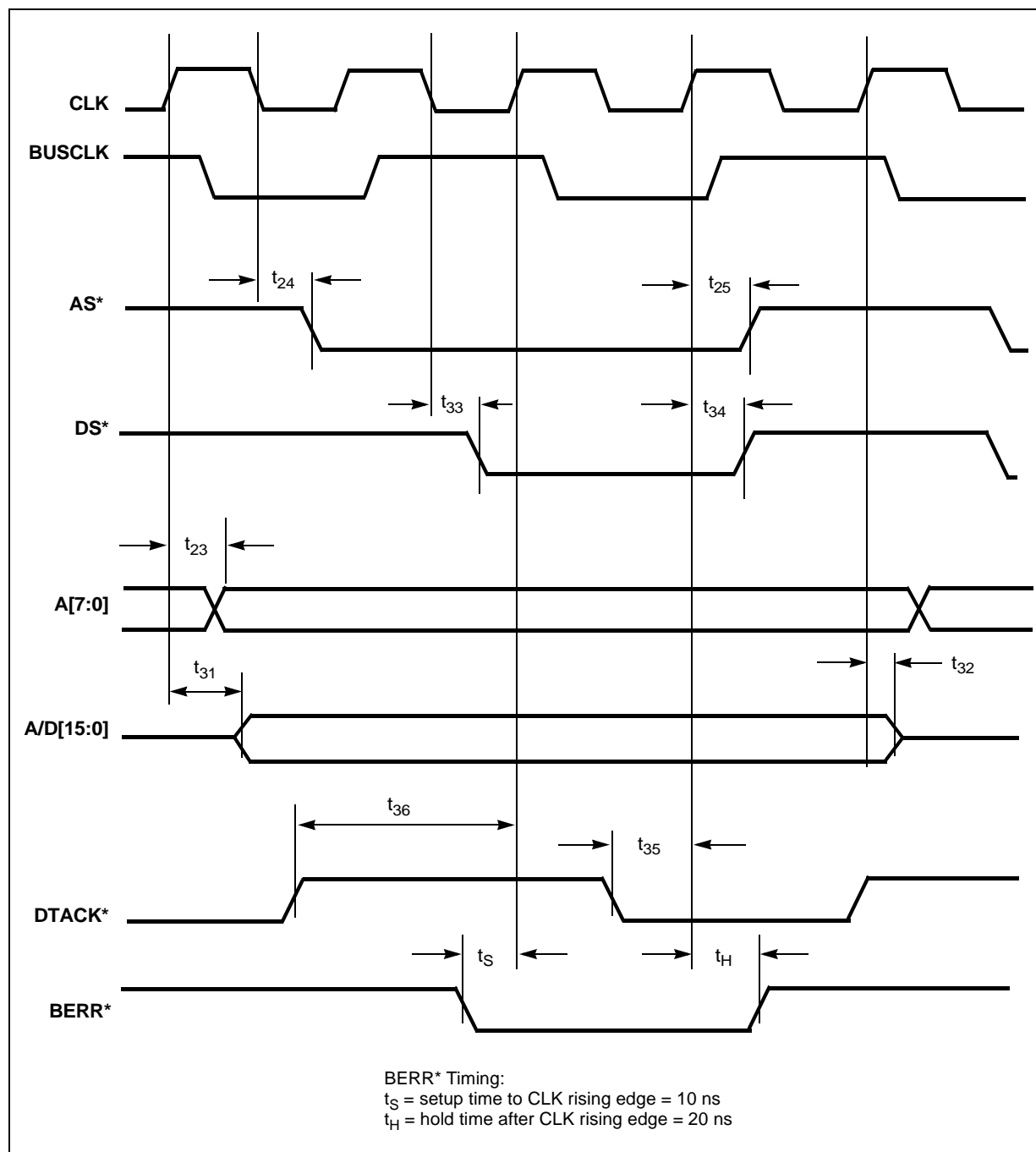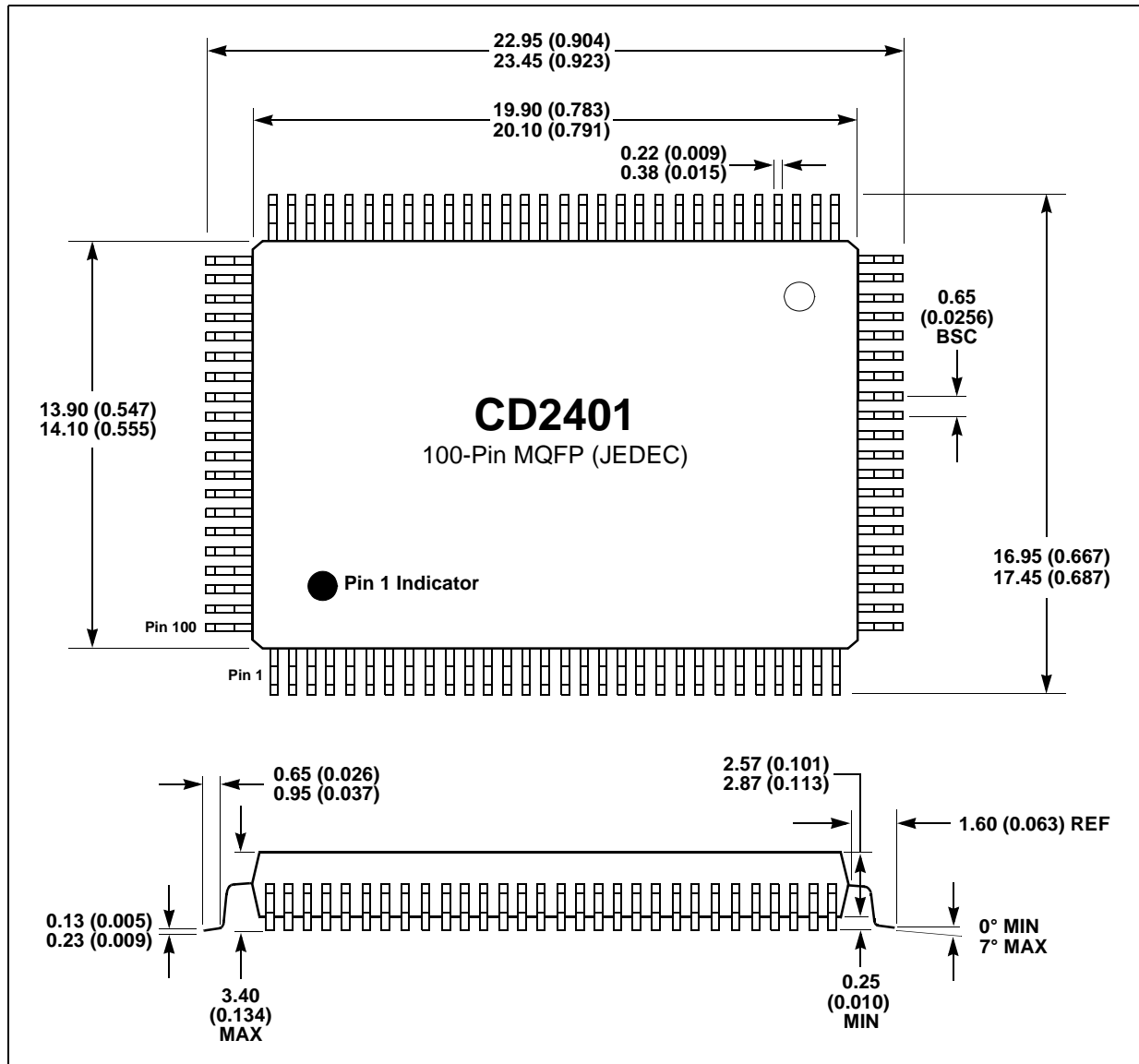
NOTES:
1. Revision I or older are available in the EIAJ package; Revision M and newer are available in the JEDEC package.
2. Dimensions are in millimeters (inches), and controlling dimension is millimeter.
3. Before beginning any new design with this device, please contact Intel for the latest package information.

11.0   Ordering Information Example

SCD240110QCM

Revision †

Product line:
Communications, Data

Part number

Internal reference number

Temperature range:
C = Commercial

Package type:
MQFP (metric quad flat pack)
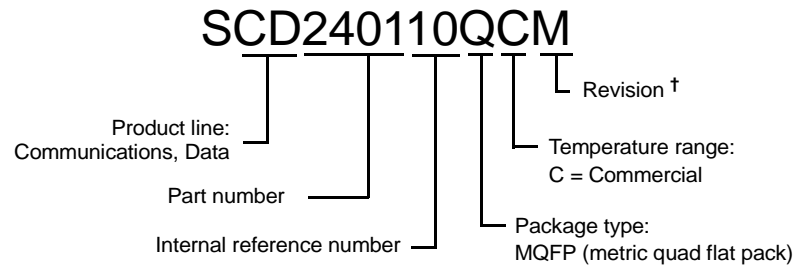
† Contact Intel Corporation for up-to-date information on revisions.

intel.

Index

A

abbreviations 13
absolute maximum ratings 157
AC electrical characteristics
    bus arbitration 158
    DMA read 158
    DMA write 158
    host read/write 158
    interrupt acknowledge 159
acronyms 14
Addressing mode 94
Append mode 56
Async-HDLC/PPP mode 31, 102, 119, 122, 131
Asynchronous 96
Asynchronous DMA mode 57
Asynchronous mode 19, 27, 98, 101, 106, 120, 130

B

bit rate generation 58
BRG operation 58
buffer allocation 54
buffers and chaining 47
bus acquisition cycle 45, 46
bus arbitration 158
bus error handling 46
byte and word transfers 37

C

cascading 42
CLK/BUSCLK/RESET* Timing Relationship 159
Clock mode 19
contexts and channels 38
conventions 13
cycles

bus acquisition 45, 46
hardware signals and IACK 41
host read 36
host read and write 35
host write 37
interrupt acknowledge 39

D

data clock selection 65
data encoding 58, 64
DC electrical characteristics 157
DMA data transfer 45
DMA operation 44
DMA read 158
DMA write 158
DPLL mode 114
DPLL operation 58

E

electrical specifications 157

F

Fair Share scheme 42
FCS (frame check sequence) 68
FCS mode 100
FIFO and timer operations 43
Flow Control Transparency (FCT) mode 101
frame check sequence (FCS) 68
functional description 35

H

hardware configurations
    32-bit data bus 66
    DMA connections 66
    DTE and DCE interface 67
hardware signals and IACK cycles 41
HDLC mode 27, 94, 96, 98, 100, 120, 129

High-Impedance mode 19
host interface 35
host read and write cycles 35
host read/write 158

I

Idle in Mark mode 100
Idle mode 100
interrupt acknowledge 159
interrupts
    acknowledge cycle 39
    contexts and channels 38
    groups and types 39
    IACK cycles 41
    keep and pass logic 42
    multi–CD2401 systems 42
    registers 38
    systems with interrupt controllers 42
    transmit and receive interrupt service requests 41

K

keep and pass logic 42

L

Local Loopback mode 115

M

memory map 21
modes
    Addressing mode 94
    Append mode 56
    Async-HDLC/PPP mode 31, 102, 119, 122, 131
    Asynchronous DMA mode 57
    Asynchronous mode 19, 27, 98, 101, 106, 120, 130
    Chain mode 48
    Clock 19
    DPLL mode 114
    FCS mode 100

Flow Control Transparency (FCT) mode 101
    HDLC 98
    HDLC mode 27, 94, 96, 100, 120, 129
    High-Impedance mode 19
    Idle in Mark mode 100
    Idle mode 100
    Local Loopback mode 115
    Parity mode 95
    PPP 98
    Protocol mode 93
    Receive Transfer mode 93
    Remote Loopback mode 97
    SYN/Flag Hunt mode 118
    Synchronous mode 100
    Transmit Transfer mode 93
    XON mode 97
multi–CD2401 systems 42

O

operations
    BGR 58
    DMA 44
    DPLL 58
    FIFO and timer 43
    receive FIFO 43
    transmit FIFO 43
ordering information example 168

P

package specifications 167
Parity mode 95
pin descriptions 16
pin diagram
    CD2401 15
pin functions
    CD2401 16
pin information 15
PPP mode 98
programming the PILR registers 41
Protocol mode 93
protocol processing 68

intel

intel.

intel.

Bit Index

intel®