CIRRUS LOGIC®

CL-CD1190

Data Sheet

FEATURES

- **Centronics®- compatible**
- **Up to 250 kbytes/second transfer rate**
- **192-byte FIFO**
- **Improved interrupt schemes**
  - Vectored interrupt to allow jump into service routine
  - Interrupt on FIFO threshold
  - Interrupts on timer expiration and external status, i.e., PAPER EMPTY, ERROR, TIMEOUT, etc.
- **Programmable FIFO threshold for receive and transmit**
- **Bi-directional operation**
- **Independent general-purpose timer**
- **Multiple CL-CD1190s can be cascaded using interrupt daisy chain scheme**
- **Programmable strobe and acknowledge widths from 0.8 µs to 50 µs**
- **Six general-purpose I/O pins**
  - Three inputs and three outputs

Other Features

- **Local loopback capability**
- **System clock up to 25 MHz**
- **Packaged in 68-pin PLCC**
- **Pin-compatible with CL-CD1400**
- **Advanced, low-power CMOS process technology**
- **Cascadable to multiple CL-CD1190s using Fair Share™ daisy chain scheme**

Intelligent Printer/Scanner Interface Controller

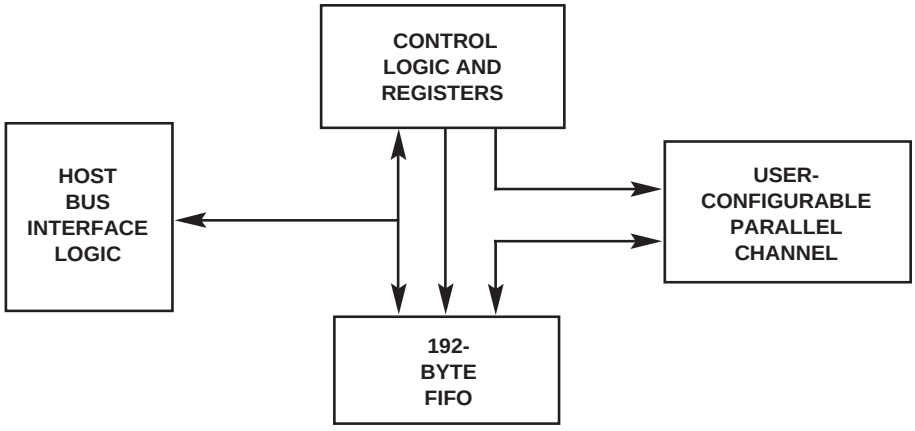
OVERVIEW

The CL-CD1190 is an intelligent, high-speed interface for bi-directional parallel transfers at rates up to 250 kbytes/second. It is a flexible, buffered interface controller for parallel data transfer to and from printers and scanners. A 192-byte FIFO greatly reduces host overhead, while providing the necessary logic for both the parallel interface and an interrupt or polled host interface. It also provides a flexible parallel interface compatible with industry-standard specifications like those used on Centronics high-speed line printers.

The standard interface has been extended within the CL-CD1190 to include a bi-directional capability to accommodate devices such as optical page scanners. The device is flexible enough to be used on either the controller or peripheral side of the parallel interface, providing a high-speed solution for both controller and peripheral manufacturers.

(cont. next page)

Functional Block Diagram



Before beginning any new design with this device, please contact Cirrus Logic, Inc., for the latest errata information. See the back cover of this document for sales office locations and phone numbers. This data sheet applies to CL-CD1190 Revision C or later devices.

OVERVIEW (cont.)

Two CL-CD1190s can be connected in tandem to provide a simple but fast bi-directional interface between two host systems.

The most important advantage of using the CL-CD1190 is the reduction in CPU overhead. It automatically handles the handshake signals, and the deep FIFO allows the host to transfer up to 192 bytes without further intervention. The CL-CD1190 can operate in either an interrupt-driven or polled environment, thus providing complete flexibility in system implementation.

To accommodate a variety of end applications, the CL-CD1190 provides programmable strobe and acknowledge pulse widths that can be varied from 800 ns to 50 µs (based on a 20-MHz system clock) in 200 ns increments.

By automatically controlling all data transfers on the parallel channel, the CL-CD1190 will, for example, generate an interrupt when its FIFO is empty during send operations. The host can then transfer up to 192 bytes directly to the controller at full bus speed. The CL-CD1190 will place the first byte on the printer interface pins, assert STROBE* and wait for ACK*. When it receives ACK*, it will place the next byte on the interface pins and repeat the process. It will continue this until the FIFO is empty, and then generate another interrupt to the host.

The CL-CD1190 does not need host support when data is being moved from the FIFO to the parallel pins, thus freeing the host to perform other functions.

Depending on the system clock frequency, the mode of operation (controller/peripheral), the direction (input/output), and the programmed duration of the handshake signals, the CL-CD1190 can sustain data transfer rates of up to 250 kbytes per second.

The CL-CD1190 is fabricated using an advanced low-power CMOS technology. The device operates on a 25-MHz system clock and is packaged in a 68-pin PLCC.

Industry-Standard Compatibility

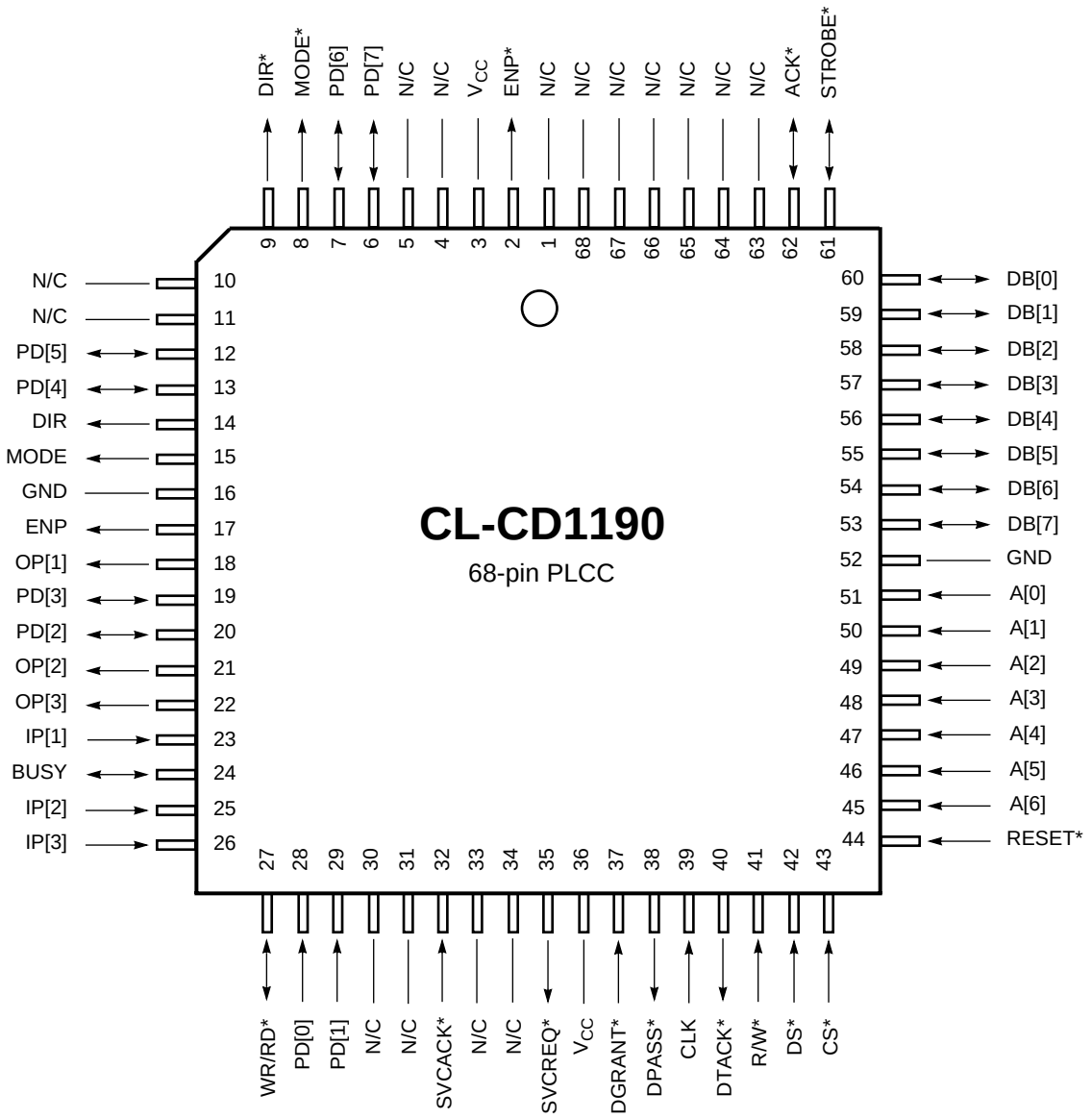
CL-CD1190	Centronics Peripheral	Dataproducts Peripheral	Hewlett Packard Controller
Data Signals			
PD0-I/O	D1-I	D1-I	DATA1-I/O
PD1-I/O	D2-I	D2-I	DATA2-I/O
PD2-I/O	D3-I	D3-I	DATA3-I/O
PD3-I/O	D4-I	D4-I	DATA4-I/O
PD4-I/O	D5-I	D5-I	DATA5-I/O
PD5-I/O	D6-I	D6-I	DATA6-I/O
PD6-I/O	D7-I	D7-I	DATA7-I/O
PD7-I/O	D8-I-optional	D8-I-optional	DATA8-I/O
Control Signals			
BUSY-I/O	BUSY-O	READY-O	BUSY-I/O
STROBE*-I/O	STROBE*-I	STROBE*-I	nSTROBE-I/O
WR/RD*-I/O	—	—	WR/RD*-O
ACK*-I/O	ACKNLG*-O	DEMAND*-O	nACKNLG-I
Auxiliary Control Signals			
IP [1,2,3]	FAULT*-O	PARITY-ERROR*-O	nFAULT-I
OP [1,2,3]	PE-O	PAPER-O	ERROR-I
—	SLCT-O	ON-LINE-O	SELECT-I
±ENP,	—	INPUT-PRIME*-I	nRESET*-O
±MODE, ±DIR	INPUT-PRIME*-I	—	nSELECTIN-O

NOTE: The auxilliary signals shown above indicate possible uses and connections for the IP and OP signal groups, as well as the Enabled, Mode and Direction signals. There is not a one-to-one correlation.

Table of Contents

1. PIN INFORMATION

1.1 Pin Diagram — Top View



541190-1

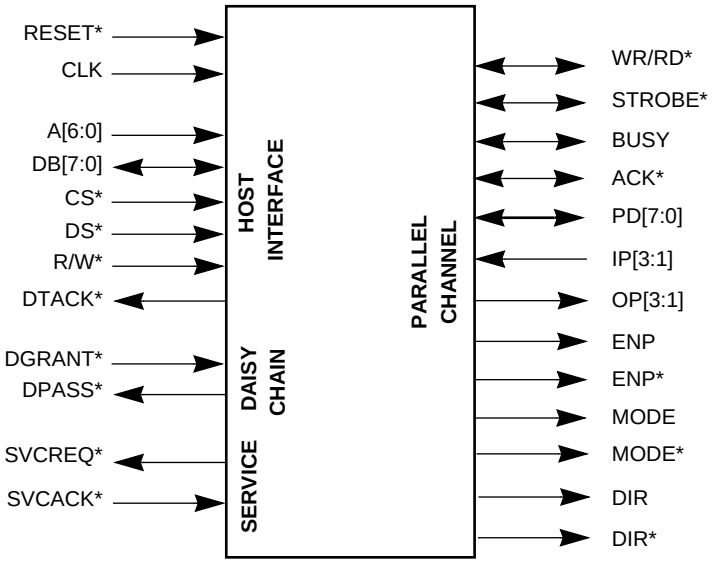
NOTES: An asterisk (*) indicates that signal is active-low.

N/C means no connection.

To reduce address space, A[4] can be tied to ground, and A[6] and A[5] can be tied together; therefore, only 32 address locations are occupied.

DIR*, MODE*, and ENP* are signals available on Revision C or later devices.

1.2 Pin Functions



541190-2

1.3 Pin List

The following conventions are used in the pin assignment tables: (*) denotes an active-low signal; I = Input, I/O = Input/Output, O = Output, OD = Open Drain.

NOTE: All Open Drain outputs must be terminated to +5V through a 1K ohm resistor.

Pin Name	DIR	# of Pins	Pin #
RESET*	I	1	44
CLK	I	1	39
CS*	I	1	43
DS*	I	1	42
R/W*	I	1	41
DTACK*	OD	1	40
A[6:0]	I	7	45-51
DB[7:0]	I/O	8	53-60
SVCREQ*	OD	1	35
SVCACK*	I	1	32
DGRANT*	I	1	37
DPASS*	O	1	38
STROBE*	I/O	1	61
ACK*	I/O	1	62
OP[3]	O	1	22
OP[2]	O	1	21
IP[3]	I	1	26
IP[2]	I	1	25
IP[1]	I	1	23

Pin Name	DIR	# of Pins	Pin #
BUSY	I/O	1	24
WR/RD*	I/O	1	27
MODE	O	1	15
MODE*	O	1	8
ENP	O	1	17
ENP*	O	1	2
DIR	O	1	14
DIR*	O	1	9
PD[0]	I/O	1	28
PD[1]	I/O	1	29
PD[2]	I/O	1	20
PD[3]	I/O	1	19
PD[4]	I/O	1	13
PD[5]	I/O	1	12
PD[6]	I/O	1	7
PD[7]	I/O	1	6
V_{CC}		2	3, 36
GND		2	16, 52

1.4 Pin Descriptions

Symbol	Pin Number	Type	Description
RESET*	44	I	**RESET**: This signal asynchronously resets the CL-CD1190. RESET must be active for a minimum of ten system clocks. When RESET is removed, the CL-CD1190 will perform a software initialization of its registers, disable all transmitters and receivers, and when complete, place the firmware revision number in the FRR.
CLK	39	I	**CLOCK**: (System Clock). The CL-CD1190 requires a nominal 20-MHz clock for proper operation. The system clock is divided by two, internally, to generate all on-chip timing clocks.
CS*	43	I	**CHIP SELECT**: When active, CS*, in conjunction with DS*, initiates a host I/O cycle with the CL-CD1190.
DS*	42	I	**DATA STROBE**: During an active I/O cycle, DS* strobes data into on-chip registers during a write cycle or enables data onto the data bus during read cycles.
R/W*	41	I	**READ/WRITE**: R/W* sets the direction of the data transfer between the host and the CL-CD1190. When high, the cycle is a read cycle, and when low it is a write cycle. This signal should not be confused with the WR/RD* Signal, which controls the direction of the parallel interface.
DTACK*	40	OD, O	**DATA TRANSFER ACKNOWLEDGE**: When the CL-CD1190 has completed internal operations associated with a host I/O cycle, it activates DTACK* to indicate the end of the cycle. The host may terminate the cycle as soon as DTACK* becomes active.
A[6:0]	45-51	I	**ADDRESS[6:0]**: These signals select the on-chip register being accessed during a host I/O cycle.
DB[7:0]	53-60	I/O	**DATA BUS[7:0]**: These eight bidirectional signals are the data interface between the host and internal CL-CD1190 registers.
SVCREQ*	35	OD, O	**SERVICE REQUEST**: When the CL-CD1190 needs host service, it activates this signal.
SVCACK*	32	I	**SERVICE ACKNOWLEDGE**: The host activates this signal to start an interrupt service. This starts a special-case read cycle, during which the CL-CD1190 places the contents of the interrupt vector register on the data bus. CS* is not activated during a SVCACK* cycle.
DGRANT*	37	I	**DAISY GRANT**: This input, qualified with DS* and a valid service acknowledge (SVCACK*), activates the CL-CD1190 interrupt service cycle.

1.4 Pin Descriptions *(cont.)*

Symbol	Pin Number	Type	Description
DPASS*	38	O	**DAISY PASS**: This output is driven low when no valid service request exists for the type of service acknowledge active. In multiple-CL-CD1190 designs, this signal is normally connected to the proceeding CL-CD1190 DGRANT* output, forming an interrupt acknowledge daisy chain.
STROBE*	61	I/O	**STROBE**: STROBE* indicates that data on the Parallel Data Port (PD[7:0]) is ready. It is driven active after an appropriate data setup time. Data is held for an appropriate hold time after STROBE* is deactivated. STROBE* is a bi-directional signal; when the CL-CD1190 is configured as a sender, it drives STROBE*, and when it is a receiver, STROBE* is an input.
ACK*	62	I/O	**ACKNOWLEDGE**: ACK* is a bidirectional signal. It indicates that the data transfer is complete. When the CL-CD1190 is a peripheral, it drives ACK* once it has taken the data on the parallel data port. When it is a controller, it uses ACK* as an indication that the peripheral has accepted the data.
BUSY	24	I/O	**BUSY**: BUSY is a bi-directional signal. It is an input when the CL-CD1190 is in one of the output modes. It is an output when it is in one of the input modes. During data input operations, the CL-CD1190 drives BUSY active after receiving the strobe from the sender. When it has taken the data, it deaserts BUSY. During data output operations, the state of BUSY is monitored and the CL-CD1190 will not assert STROBE* when BUSY is active.
WR/RD*	27	I/O	**WRITE/READ***: The WR/RD* indicates the direction of the parallel data transfer. When the CL-CD1190 is in the Controller Mode, this signal is an output; when in the Peripheral Mode, it is an input. The state of the output is controlled by the value written into Bit 3 of the SCR Register when the CL-CD1190 is in Controller Mode. Bit 7 of the SCR always shows the current state of the WR/RD* Output, regardless of the enabled mode. This signal should not be confused with the R/W* Signal, which controls the direction of the data bus during host I/O cycles.

1.4 Pin Descriptions *(cont.)*

Symbol	Pin Number	Type	Description
PD[0]	28	I/O	**PARALLEL DATA 0**
PD[1]	29	I/O	**PARALLEL DATA 1**
PD[2]	20	I/O	**PARALLEL DATA 2**
PD[3]	19	I/O	**PARALLEL DATA 3**
PD[4]	13	I/O	**PARALLEL DATA 4**
PD[5]	12	I/O	**PARALLEL DATA 5**
PD[6]	7	I/O	**PARALLEL DATA 6**
PD[7]	6	I/O	**PARALLEL DATA 7** These signals provide the eight-bit parallel data interface for the CL-CD1190. Their direction is controlled by the current mode of operation as set by the DIR Bit in the GCR Register. (See the register descriptions for detailed information on register bit assignments).
OP[1]	18	O	**GENERAL-PURPOSE OUTPUT 1**
OP[2]	21	O	**GENERAL-PURPOSE OUTPUT 2**
OP[3]	22	O	**GENERAL-PURPOSE OUTPUT 3** These three signals are general-purpose outputs. Their state is controlled by the lower three bits of the SCR Register. (See the register descriptions for detailed information on register bit assignments).
IP[1]	23	I	**GENERAL-PURPOSE INPUT 1**
IP[2]	25	I	**GENERAL-PURPOSE INPUT 2**
IP[3]	26	I	**GENERAL-PURPOSE INPUT 3** These three signals are general-purpose inputs. Their state can be monitored via the upper four bits of the SCR Register. (See the register descriptions for detailed information on register bit assignments).
DIR	14	O	**DIRECTION**: The DIR Signal is provided to aid in the control of external buffers. It is high for input and low for output and is a direct reflection of the DIR Bit in the GCR.
DIR*	9	O	**DIRECTION***: This signal is the inverse fo DIR.
ENP	17	O	**ENABLE PARALLEL TRANSFER**: ENP is an active-high output that indicates the current state of the CL-CD1190. When the device is enabled for data transfers via CCR command, this output will be high; otherwise, it will be low.
ENP*	2	O	**ENABLE PARALLEL TRANSFER***: ENP* is the inverse of ENP.

1.4 Pin Descriptions *(cont.)*

Symbol	Pin Number	Type	Description
MODE	15	O	**CONTROLLER/PERIPHERAL MODE**: This output indicates the current mode of the CL-CD1190. When it is low, the CL-CD1190 is in the Controller Mode; when high, it is programmed as a peripheral. It is a direct reflection of the Mode Bit in the GCR.
MODE*	8	O	**CONTROLLER/PERIPHERAL MODE***: This output is the inverse of MODE.

2. FUNCTIONAL DESCRIPTION

2.1 Device Architecture

The CL-CD1190 consists of four main functional blocks, as shown in Figure 2-1. These are the host interface, the parallel port interface, a 192-byte FIFO and control logic. The host bus interface block controls and sequences host I/O activity, arbitrating accesses between the host and internal logic preventing clashes between the two. The FIFO is made up of a large static RAM array, with pointers and thresholds managed by the control logic block. In addition to FIFO management, the control block also performs the necessary control of the parallel data port strobes and signals. The parallel port block provides all of the interface functions to allow the device to be connected to a variety of peripherals and controllers.

2.2 Host Interface

The host interface to the CL-CD1190 is made up of an 8-bit bidirectional data bus, a 7-bit address bus and various strobes that identify the type and direction of the I/O cycle that is taking place. Although the strobe names and basic timing are similar to that of the Motorola 68000 family, the CL-CD1190 easily fits into any CPU environment.

In most cases, when the host reads or writes an internal CL-CD1190 location, it actually accesses a location in a RAM array that serves as a bank of registers. However, some locations are mapped to actual hardware resources, such as when a true signal value is required; an example is the current state of the WR/RD* Pin in the Signal Control Register.

The CL-CD1190 is designed to be socket-compatible with the CL-CD1400; therefore, it has seven (7) address lines. The CL-CD1190, however, does not actually require an address space of this magnitude. The address space can be minimized by connecting the device, as shown in Figure 2-2 on the following page, and using the 'CA' (Compressed Address) values shown in Section 4.1, *CL-CD1190 Register Map*.

The CL-CD1190 is, by design, a synchronous device. All internal operations take place on edges and levels (phases) of the internal clock. Note that the internal clock is generated by dividing the external (system) clock by two. When the host performs an I/O cycle with the CL-CD1190, its strobes, address and data are sampled on falling edges of the internal clock. As can be seen in the timing diagrams in Section 5, external control signals must meet setup times with respect to clock edges. Once a cycle has started, the sequence of events is locked to the CL-CD1190 internal clock, with events (address setup, write data setup and read data available) occurring at predictable times.

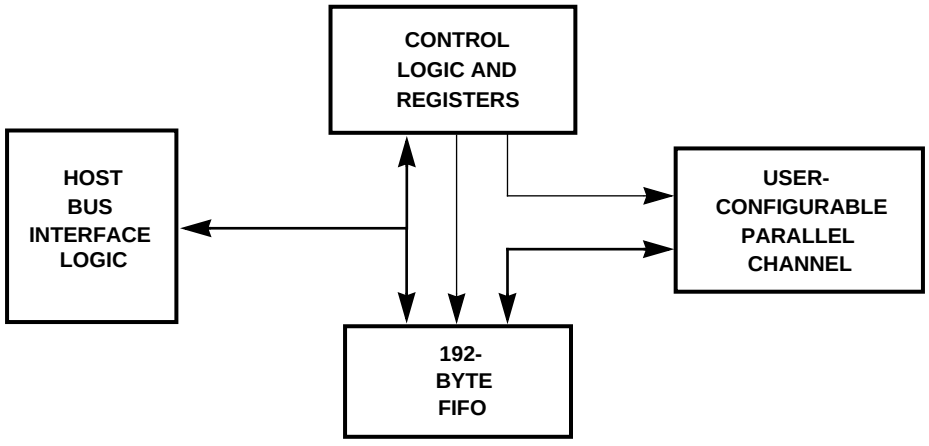


Figure 2–1. CL-CD1190 Functional Block Diagram

It is not necessary, however, to design a synchronous interface to the CL-CD1190. In an asynchronous design, the Data Transfer Acknowledge (DTACK*) Signal is used as an indication that the CL-CD1190 has completed the requested data transfer. Thus, DTACK* can be an input to wait state generation logic that will hold the host CPU until the operation is complete. If the strobes (Chip Select and Data Strobe — CS* and DS*) do not meet the minimum setup time with respect to a clock edge, the CL-CD1190 will not detect the I/O request, and the cycle will be delayed two full-system clock cycles, thus meeting the setup time. The I/O cycle will then commence and follow the synchronous timing, with DTACK* signaling the end of the cycle.

2.2.1 Host Read Cycles

Read cycles are initiated when the CL-CD1190 senses that both the CS* and DS* inputs are active and the Read/Write (R/W*) Input is high. All strobes and address inputs must meet setup times as specified in the timing specifications in Section 3. It is important to note that *both* the CS* and DS* signals must be valid for a cycle to start, thus cycle times are measured from whichever of the two signals goes active *last*. The CL-CD1190 signals the fact that it has completed the read cycle (placing the data from the addressed register on the data bus pins) by activating the DTACK* Signal. DTACK* is released when the host removes CS* or DS*.

2.2.2 Host Write Cycles

Write cycle timing and strobe activity is nearly identical to read cycles, except that the R/W* Signal must be held low. Write data, strobes and address inputs must meet setup and hold times as specified in the timing diagrams in Section 3. Again, the DTACK* Signal is used to indicate that the cycle is complete and the CL-CD1190 has taken the data. Removing either CS* or DS* releases DTACK*.

2.2.3 Host Service Acknowledge Cycles

Service acknowledge cycles are a special-case read cycle. Timing is basically the same as a normal read cycle, except that the SVCACK* and DGRANT* inputs are activated instead of the CS* Input (a longer setup time is required on the SVCACK* Input than on the CS* Input). The data that the CL-CD1190 provides during the read cycle is the contents of the Interrupt Vector Register. (See description of service request procedures later in this section). As in read and write cycles, DTACK* will indicate the end of the cycle and removing DS* releases DTACK*.

2.3 Service Requests and Interrupts

Four bits in the Interrupt Configuration Register (ICR) are used to specify the events that will cause the host to be notified. When an enabled event occurs, the CL-CD1190 initiates a Service

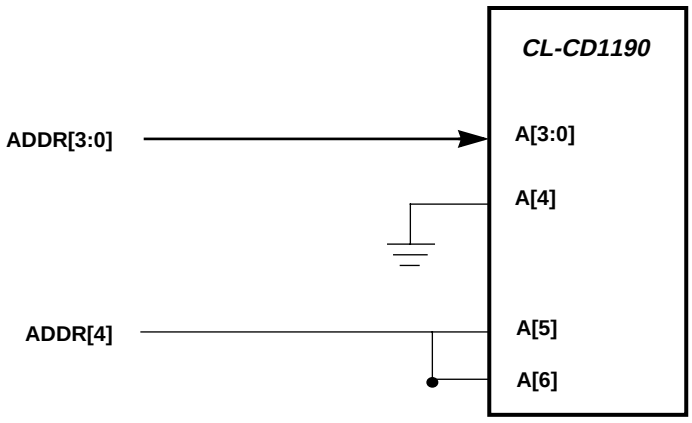


Figure 2–2. Compressed Address Connections

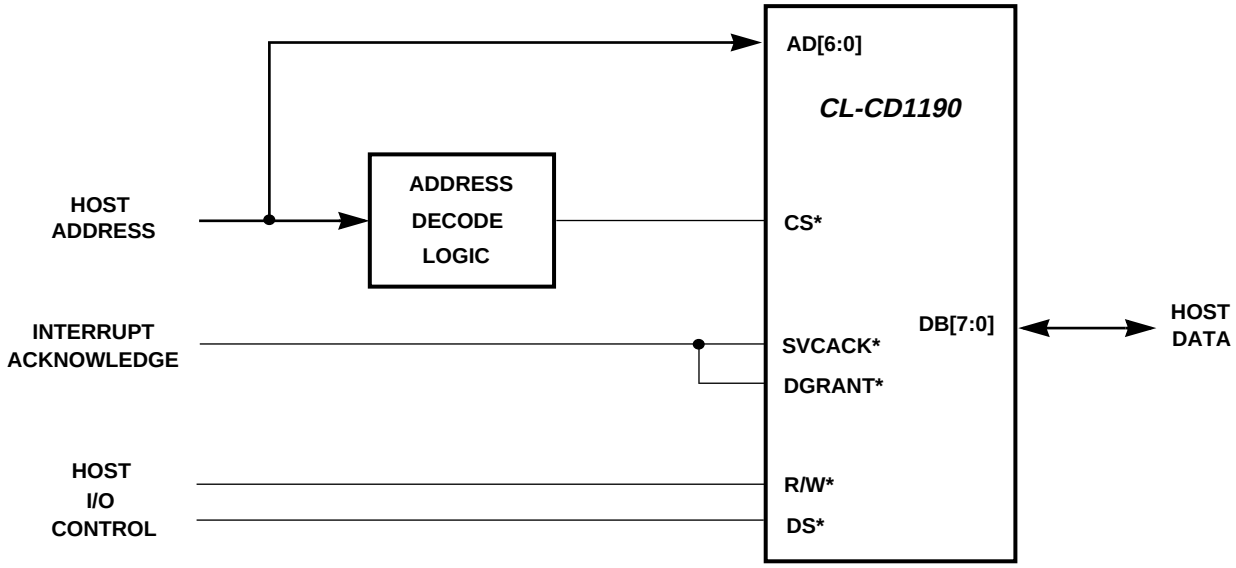
Request. The cause of the request is recorded in the Interrupt Status Register (ISR), and an internal Service Pending Flag is set. If the IEN Bit in the ICR is set, indicating Interrupt Mode, the SVCREQ* Pin will be asserted low. The host CPU becomes aware of the service request either by periodically reading the ISR, Polled Mode, or from the SVCREQ* Signal via system interrupt logic. In Interrupt Mode, the host CPU responds with an interrupt acknowledge cycle that activates the SVCACK* Pin. During the acknowledge cycle, the CL-CD1190 will present the contents of the Interrupt Vector Register (IVR) onto the data bus. The acknowledge cycle is not necessary when using Polled Mode. In either mode, the last action of the service request handler routine must be a write operation to the End of Service Register.

2.3.1 Service Pending Flag and End of Service Register (ESR)

The Service Pending Flag is an internal flag used by the CL-CD1190 to keep track of whether a Service Request is in progress. The flag is invisible to the user, but its state does affect operation of the CL-CD1190. Once a Service Request is posted, there is no way of knowing exactly when the host will respond to it. If the CL-CD1190 attempts to update ISR after a request is posted, there could be a collision between its access and that of the external host. If this occurs, status may be lost. To prevent this situation, the CL-CD1190 suspends certain operations while the Service Pending Flag is set.

During the time that this flag is set, the CL-CD1190 will stop sampling the general-purpose input pins to detect changes in their state. Since signal change detect is *not* intended to detect fast pulses (it is intended to detect conditions like 'paper empty'), a pause in sampling will not cause a problem; the change in level will be detected when sampling is resumed. The CL-CD1190 also stops threshold detection, but both functions are enabled again as soon as the Service Pending Flag is cleared by writing to the ESR at the end of a service routine.

The CL-CD1190 starts its service acknowledge cycle when both DS* and DGRANT* are active together. If SVCREQ* and SVCACK* are both active before that time, then the CL-CD1190 will place its IVR contents on the data bus and assert DTACK*. Otherwise, the CL-CD1190 will pass on the grant by asserting DPASS*. When the grant is passed on, no DTACK* will be asserted, and DPASS* will remain active as long as the inputs stay unchanged.



541190-5

Figure 2–3. Control Signal Configuration

The diagrams show SVCACK* and DGRANT* tied together. This can only be done if DS* is guaranteed to be later than SVCACK*. Otherwise, it is necessary to delay DGRANT* until after SVCACK*. See the timing diagrams in Section 5. Chip Select (CS*) must remain inactive during the entire duration of a service acknowledge cycle.

A summary of Interrupt Mode service request actions is as follows:

1. CL-CD1190 asserts a low level on its SVCREQ* Output Pin.
2. The host responds by performing an interrupt acknowledge cycle that activates the SVCACK* Input Pin.
3. The host uses the value obtained from the Interrupt Vector Register during Step 2 if it is significant.
4. The host executes the interrupt handler routine (data transfer, etc.).
5. The host writes to the ESR to terminate the service and clear the service pending flag.

2.3.2 Systems with Interrupt Controllers

Some systems use an interrupt controller that supplies its own vector during the interrupt acknowledge cycle. These systems can decode a location distinct from the CL-CD1190 to produce a SVCACK* instead of CS*. The host should read this location before the first access to the device in an interrupt service routine. The CL-CD1190 will enter its interrupt context, and the data returned will be the interrupt vector from IVR.

2.3.3 Multiple CL-CD1190s and the Interrupt Daisy Chain

The CL-CD1190 provides a means of 'daisy chaining' the service acknowledgments of two or more devices together. Figure 2–4 shows how two CL-CD1190s would be connected. Priority is based on the position in the chain. The open-drain SVCREQ* outputs of the two devices are wire-ORed together to form one request. Similarly, the SVCACK* inputs are also connected together. The DPASS* Output of the first CL-CD1190 drives the DGRANT* Input of the second, forming the chain. The respective IVR of the two chips must be loaded with unique patterns, so the host will know

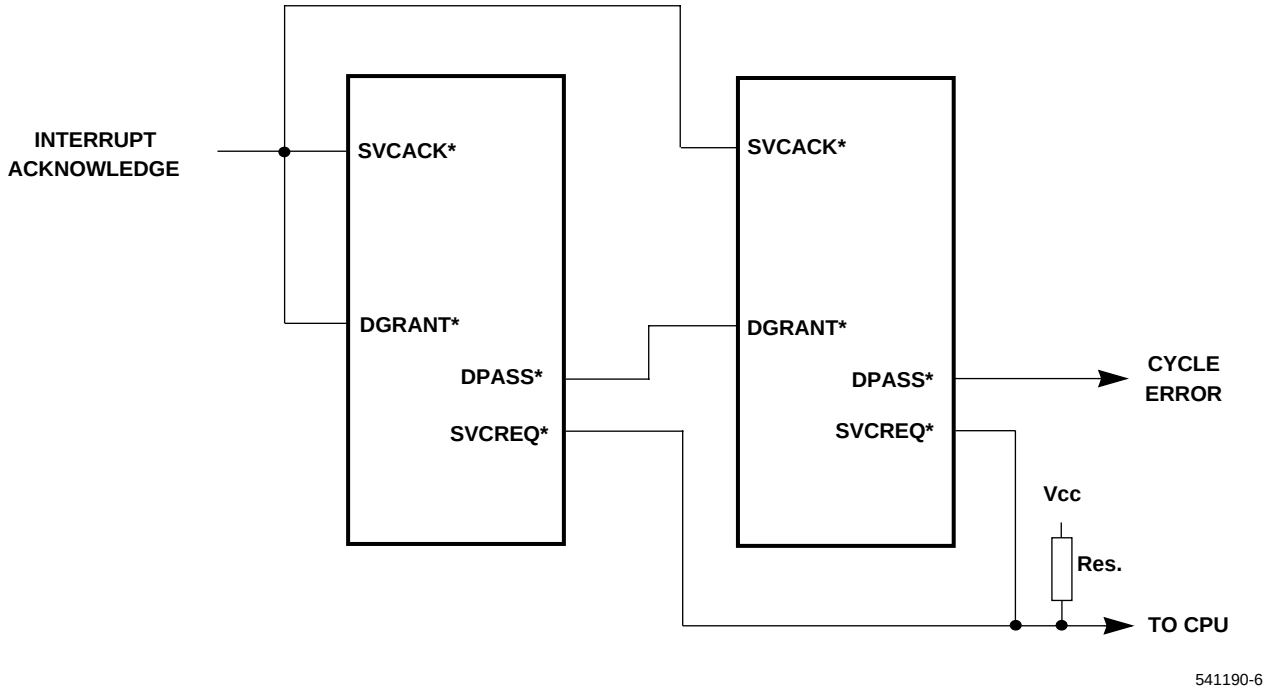


Figure 2–4. CL-CD1190 Daisy Chain Connections

which CL-CD1190 responded to the service acknowledge.

The DGRANT* Input of the first CL-CD1190 may be connected to SVCACK* if DS* will be later than SVCACK*. If not, then DGRANT* must be delayed until after SVCACK* is stable. On the second and subsequent devices, DGRANT* will be propagated from previous CL-CD1190 DPASS* outputs, and thus DGRANT* will always be later than SVCACK*. See the timing diagrams in Section 5.

When the host acknowledges the request, both CL-CD1190s will receive the acknowledge via the SVCACK* Input. However, only the first will receive the DGRANT*. If it has an active SVCREQ* request pending, it will take the acknowledge and drive its vector register onto the data bus.

If it does not have a request pending, it will pass the grant by asserting DPASS*, which is connected to the second CL-CD1190 DGRANT* Input. Assuming that the second has an active request pending, the second device will take the acknowledge and drive its vector register onto the data bus.

CAUTION: If neither CL-CD1190 has a pending request, the DPASS* will be output by the second and neither will respond with DTACK*, possibly causing the bus cycle to hang. The actual implementation is system-dependent; however, it is important for the designer to provide some way for the host to know that the cycle did not complete normally, if no device exists at the end of the chain.

2.3.4 Polled-Mode Operations

In Polled Mode, the IEN Bit in the Interrupt Configuration Register is cleared so that the SVCREQ* Pin will never be asserted. The host periodically checks the Interrupt Status Register (ISR) to see if there is any request for service pending. If the host detects a non-zero value in ISR, it should proceed by executing its service routine, thus clearing the cause of the service request. Once the service is complete, the operation should be followed by a write to the ESR Register.

In polled systems, the SVCREQ*, SVCACK* and DGRANT* signals should all be terminated to Vcc through a 1K - 5K ohm resistor. ***Under no circumstances should the SVCACK* Input be activated if the SVCREQ* Output is not active.*** The DPASS* Output will be asserted if SVCACK*, DGRANT* and DS* inputs are active together. There is generally no need for a vector in a Polled Mode environment.

A summary of Polled Mode service request actions is as follows:

1. CL-CD1190 sets ISR bits according to the cause of the service request.

2. The host scans the ISR periodically, checking the INT Bit; if it is true (‘1’), host service is requested.

3. The host performs a service-handler routine (i.e., FIFO data transfer, if that is the source of the request).

4. The host writes to the ESR to terminate the service and clear the internal service pending flag.

2.4 Timer Operations

The CL-CD1190 provides two on-chip timers utilized as follows: one general-purpose timer is used by the host system for any purpose; the other is used to set a maximum time that data will remain in the FIFO, if insufficient data is received to trigger the programmed FIFO threshold. Each of these timers is implemented as a two-stage counter.

2.4.1 The Prescale Counter

The first stage is common to both timers (see Figure 2–5). The CL-CD1190 input clock is divided by 512, and the resulting signal is used to decrement the prescale counter. The prescale counter is loaded from a holding register, whose value comes from the Timer Prescale Register (TPR).

Each time the prescale timer is decremented to zero, it is automatically reloaded from the holding register. A CCR command (timer enable) is required to update the holding register from the TPR. The output of the prescaler counter is called a ‘tick’, and it is used by the upper-level counters in both the general-purpose and data timers.

2.4.2 The General-Purpose Timer

The period for the upper-level counter of the general-purpose timer is supplied by the user in the Timer Multiplier Register (TMR), as shown in Figure 2–6. This timer was designed for single time-outs and must be re-enabled via CCR command each time it needs to be started. The counter is loaded by the CCR timer enable command and decremented by the TPR tick. When the counter decrements to zero, a Timer Expired service request will be initiated if the ITE Bit in the Interrupt Configuration Register (ICR) is set.

2.4.3 The Data FIFO Timer

When receiving data from the parallel interface, it is possible that the end of the transfer will come before the FIFO threshold point is reached. The data timer is provided to force a Data Ready interrupt after a specified period of time.

The period for the upper level counter of this timer is supplied by the user in the Data Time-out Register (DTR). This timer is shown in Figure 4-11. The DTR counter is decremented by the TPR tick, and the counter is reloaded each time a new character is received from the parallel channel. If there are any data bytes waiting in the FIFO when counter decrements to zero, a Data Ready service request will be initiated, if IDR is enabled in the ICR.

2.5 FIFO Operations

The CL-CD1190 contains a 192-byte FIFO for buffering data transfers. The FIFO Data Register (FDR) is the host interface end of the FIFO, and is a single location used for both input and output. FDR should only be accessed during a service request handler routine.

The FIFO Threshold Register (FTR) is used to set the point at which the CL-CD1190 informs the host that more space is left in the FIFO for additional data. Upon entering a service routine, the value in the FIFO Count Register (FCR) can be used to determine the minimum number of bytes available for transfer.

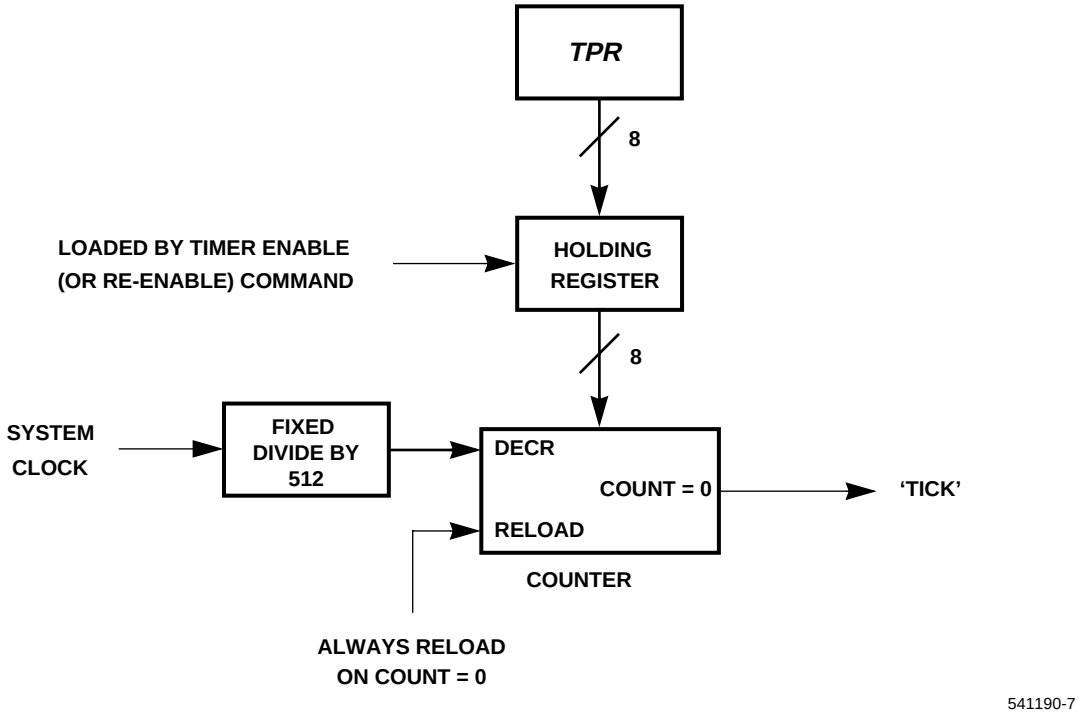


Figure 2–5. Timing Period ('Tick') Generation

2.5.1 Input Transfer Modes

When the number of bytes collected in the FIFO during input transfers becomes more than the value indicated by FTR Register, a Data Ready service request will be initiated, if the DRI Bit in the Interrupt Configuration Register (ICR) is set. In input modes, the value in the FCR is the number of bytes available to the host for reading.

When receiving data from the parallel interface, it is possible that the end of the transfer will come before the threshold point in FTR is reached. The Data Time-out Register (DTR) timer is provided to force a Data Ready interrupt after a specified period of time.

2.5.2 Output Transfer Modes

When the number of bytes remaining in the FIFO during output transfers becomes less than the value indicated by FTR Register, a Data Ready service request will be initiated, if the DRI Bit in the ICR is set. In output modes, the value in the FCR is the number of bytes waiting to be sent. The number of open spaces available for writing by the host is: N = (192 - FCR) bytes. The DTR timer is not used in output modes.

2.5.3 FIFO Loopback Test

This feature is provided for system diagnostics. It allows the host to test the entire data path through the interface controller to the CL-CD1190 FIFO. The data read back should be checked against the data written, either directly or by checksum, etc. Comparison of individual bytes is necessary to verify that data was delivered in the correct order.

The following steps should be performed to execute the FIFO loopback test:

1. The host writes a CCR command with the PAR and FLUSH bits set.

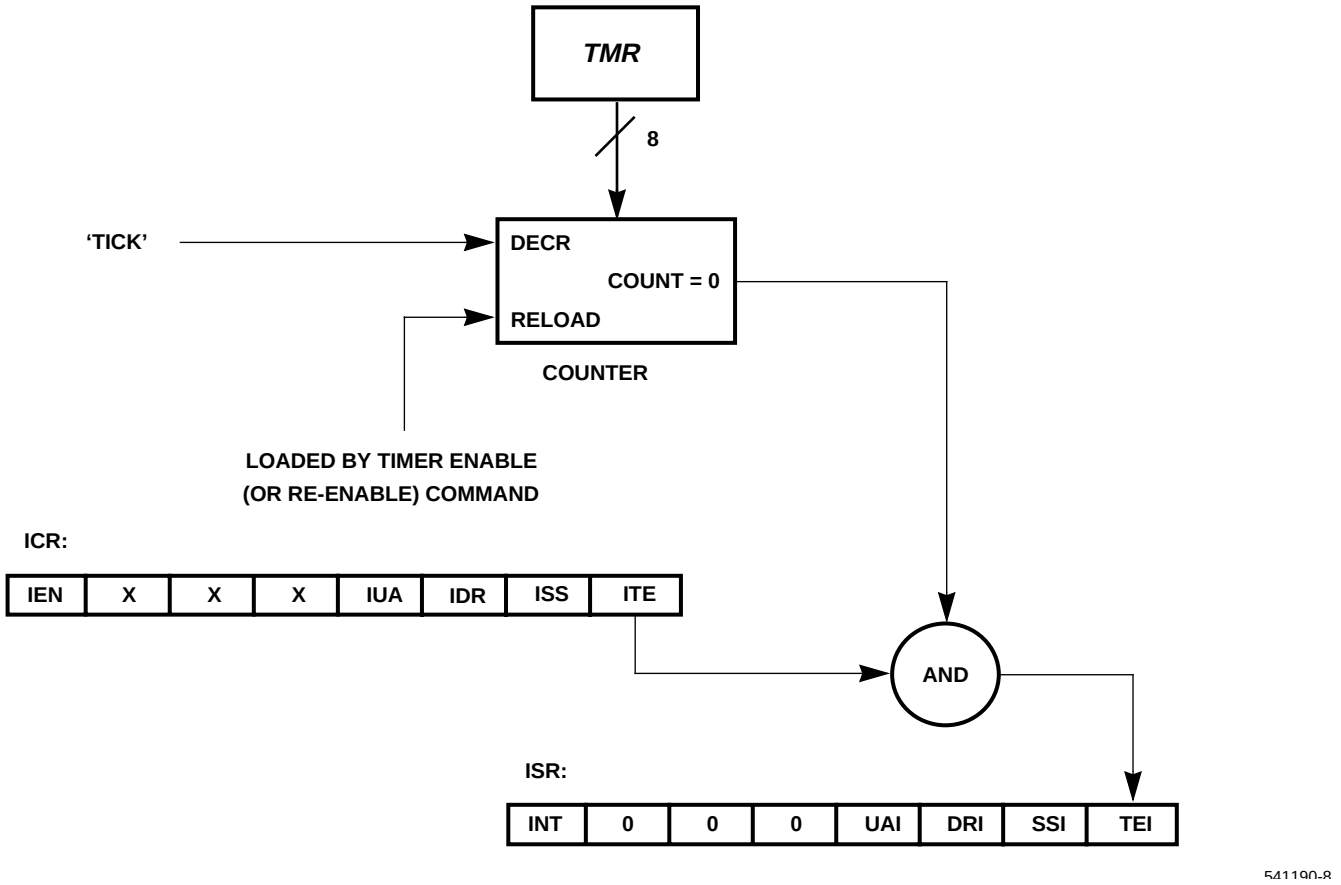


Figure 2–6. General-Purpose Timer

2. Write 192 bytes to the FIFO Data Register (FDR).
3. Read 192 bytes from the FIFO Data Register.
4. Verify the data.

2.6 Parallel Interface — Theory of Operation

This section describes the overall structure of the parallel interface, including some history and important terminology.

2.6.1 History — The Centronics Specification

The original Centronics specification described the interface for a parallel printer. This interface has become a de-facto industry standard and is widely used as the basis for parallel printer interfaces (with some modifications). It is limited, however, because the interface makes no provision for bidirectional operation. In addition, the maximum speed is limited by the timing requirements. Various companies have used simple modifications to the interface to enhance its capabilities. The CL-CD1190 is flexible enough to accommodate many of these modifications, including those used with high-speed Dataproducts® line printers and the fully bidirectional Hewlett Packard® Scan Jet™ optical page scanner.

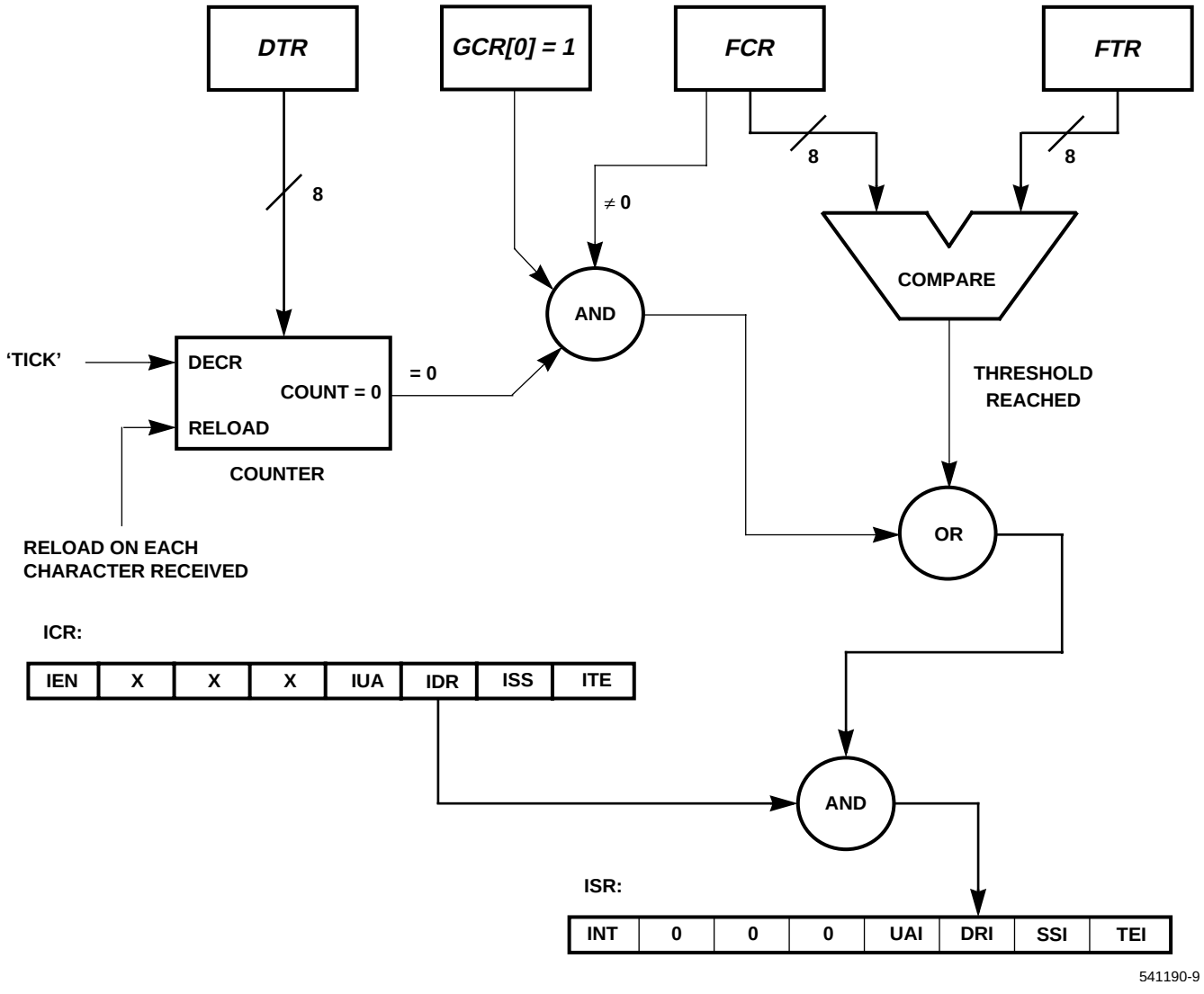


Figure 2–7. FIFO Timer

2.6.2 Terminology—Controllers, Peripherals and Hosts

In this data sheet, the terms controller and peripheral are used to identify the different modes under which the CL-CD1190 can operate. The controller is always the master of the interface and determines the direction of transfer. The normal direction of data flow is from the controller to the peripheral. The peripheral can only send data to the controller after it has been instructed to do so by the controller.

The term sender is used to indicate the device (in an output mode) transmitting data. Receiver means the device (in an input mode) to which the data is being sent. These terms are independent of the controller and peripheral. The host refers to any processor that has direct control of the CL-CD1190, regardless of whether the CL-CD1190 is acting as controller or peripheral.

2.6.3 Control Signals

The mode of operation is set in the Global Configuration Register (GCR). The MODE and DIR pins reflect the values stored in the GCR; the ENP Pin is active when the CL-CD1190 is enabled for transfer. The direction and function of the four primary parallel control signals are affected by the current mode and direction of operation, controller or peripheral, and input or output. These signals are WR/RD*, STROBE*, BUSY, and ACK*.

The controller determines the direction of transfer with the WR/RD* Output. The peripheral uses this signal as an input to see which direction the controller expects data to flow. STROBE* is always driven in the same direction as the data flow. It is driven by the sender to indicate that the data on the PD lines is ready for sampling. The receiver should use the STROBE* pulse to latch the data. BUSY is the flow control signal. Its direction is always opposite to that of the STROBE* Signal. A sender may not assert STROBE* while the receiver has BUSY asserted. The ACK* Signal is always driven by the peripheral. The ACK* Signal is pulsed in response to a STROBE*, or in response to a change in the state of the WR/RD* Input. The response to STROBE* is automatic, but the response to changing WR/RD* requires host intervention, because the host must reprogram the GCR to change the direction of data transfer.

2.6.4 Protocol

The CL-CD1190 provides the primitives necessary to construct a protocol that allows bidirectional transfers between the controller and the peripheral. A protocol is necessary because the parallel interface is half-duplex in nature. Data can only travel in one direction at a time. Some form of communication is required between the controller and peripheral to agree to change direction. Changing directions also requires host intervention, to reconfigure the part for data transfer in the opposite direction. The transfer protocol is defined by the user.

2.6.5 Changing Directions

When the controller host is ready to receive data from the peripheral, it can send a special character sequence that the peripheral host understands as a request for data. The controller then changes the WR/RD* Pin to a low level to indicate that the peripheral can begin to send data back. The peripheral sends a pulse on the ACK* Signal to confirm that the change of state of WR/RD* was detected. The peripheral may send an unlimited amount of data, with a special character sequence to indicate the end of transmission. The controller will again change the state of WR/RD* back to a high level, and the peripheral acknowledges this with an ACK* pulse. The CL-CD1190 can detect these ACK* pulses that are not part of the normal data transfer handshake in response to STROBE*. These are called unsolicited acknowledges.

2.6.6 CL-CD1190 Modes of Operation

The CL-CD1190 uses four basic modes of operation. The part can take on the role of either controller or peripheral. The controller is always the master of the interface and decides the direction of transfers. The peripheral is the slave, which performs as instructed by the master. In each of these two roles, the CL-CD1190 can transfer data in either direction. Thus the four basic modes are Controller Output, Controller Input, Peripheral Out, and Peripheral Input. There is a handshake option in Controller Output Mode, which creates a

fifth mode called No Acknowledge Mode. The function and direction of the CL-CD1190 control signals depends on the mode into which the device is programmed.

Note that most if not all designs will fix the CL-CD1190 in either Controller or Peripheral Mode, depending upon the type of product in which the device is used. When the CL-CD1190 is built into a printer product, the device should be configured as a peripheral. If the CL-CD1190 is used in a controller product, it should be configured as a controller.

2.6.7 Controller Output Mode

When the CL-CD1190 is configured for Controller Output Mode, it will send data whenever the part is enabled and there are one or more characters remaining in the FIFO. Service requests can be programmed to occur when the FIFO becomes empty or when a user-programmable threshold has been reached.

The CL-CD1190 sends data by placing each byte on the parallel data pins and asserting a pulse on the STROBE* Signal. The setup time, hold time, and width of the STROBE* pulse, are all equal, and this value is programmed into the SWR Register. The CL-CD1190 waits for the peripheral device to respond with a pulse on the ACK* Input, which indicates that the data has been accepted. The peripheral device may also activate the BUSY Signal to indicate that it is busy and cannot accept any more data at this time. The CL-CD1190 will not assert another STROBE* pulse while BUSY is high. Once ACK* and BUSY have returned to their normal states, the CL-CD1190 will place the next data byte on the port (if the FIFO is not empty), and the cycle repeats.

When the NOACK Bit in the GCR is set, the CL-CD1190 ignores ACK*. In No Acknowledge Mode, only BUSY is used for handshake.

Figure 2–8 shows a typical connection for a CL-CD1190 in Controller Output Mode, attached to a printer.

2.6.8 Controller Input Mode

When the CL-CD1190 is configured for Controller Input Mode, it will accept data whenever the part is enabled and there are one or more character spaces remaining in the FIFO. Service requests can be programmed to occur when the FIFO becomes full or when the FIFO threshold has been reached.

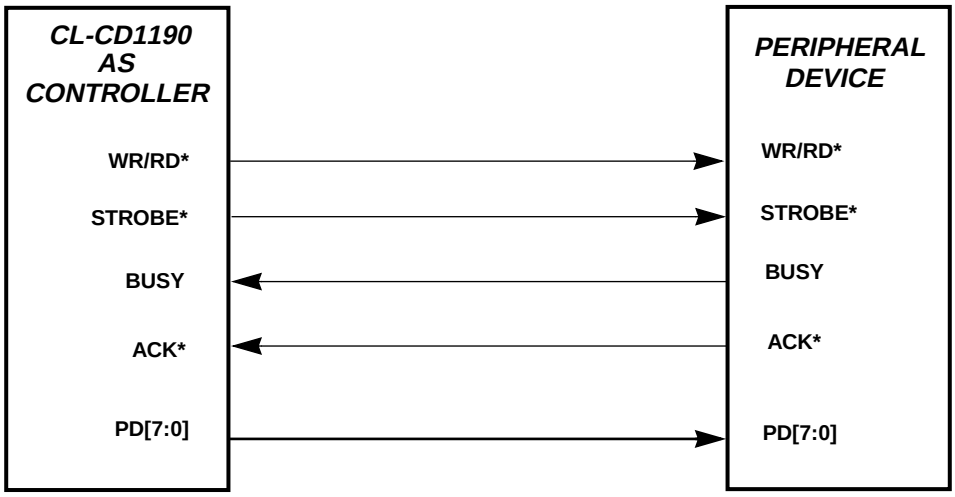


Figure 2–8. CL-CD1190 Controller Output Mode

The CL-CD1190 waits for incoming pulses on the STROBE* Pin. Shortly after the falling edge of this signal, the data on the parallel port is sampled and copied to the FIFO. Only the BUSY Pin is used for handshake. The controller raises BUSY at the falling edge of STROBE*, and lowers it when ready for the next data transfer. If the FIFO is full, BUSY will not be lowered until at least one byte is removed from the FIFO during a host service routine. The peripheral must not assert STROBE* with new data until BUSY has been cleared.

The ACK* Pin is an input when the CL-CD1190 is in Controller Mode, so this pin is not part of the normal transfer handshake in this mode. ACK* is only used for the direction-change protocol.

Figure 2–9 shows the connections between the CL-CD1190, acting as a controller, and a peripheral that sends data, such as a scanner.

2.6.9 Peripheral Output Mode

When the CL-CD1190 is configured in Peripheral Output Mode, operations are similar to Controller Output Mode, except that the ACK* Pin is not used for data transfer handshaking; it is only used for the direction-change protocol. The peripheral host should examine the WR/RD* Signal to verify that the controller expects the peripheral to send. The peripheral CL-CD1190 waits until BUSY is low and then asserts a pulse on the STROBE* Output. The data setup time is identical to the width of STROBE* and both are determined by the value in the SWR. The controller is expected to raise BUSY in response to the STROBE*, but not to send an ACK* pulse. The peripheral will not send new data with another STROBE* until BUSY has returned to its low state.

Figure 2–10, on the following page, shows the connections between a receiving controller and the CL-CD1190 in Peripheral Output Mode. The CL-CD1190 is acting as a sending device, such as a scanner.

2.6.10 Peripheral Input Mode

The operation of the CL-CD1190 in Peripheral Input Mode is similar to the Controller Input Mode, except that the peripheral always responds to a STROBE* pulse with a pulse on the ACK* Signal. BUSY handshake is identical to Controller Input Mode.

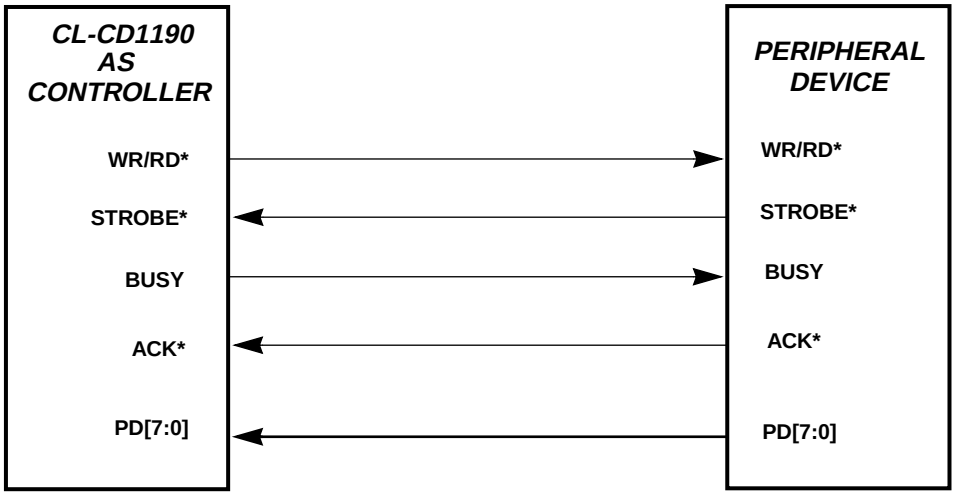


Figure 2–9. CL-CD1190 Controller Input Mode

The peripheral CL-CD1190 waits for an incoming STROBE* pulse from the controller. The peripheral responds with a pulse on the ACK* Output, which indicates that the data has been accepted, and by raising the BUSY Signal to indicate that it cannot accept any more data at this time. The controller must not assert another STROBE* pulse while BUSY is high. Once the CL-CD1190 has processed the data, BUSY will be returned to its normal state, and the controller may send the next byte.

On the following page, Figure 2–11 shows the connections between the controller and the CL-CD1190 as a receiving device in a printer application.

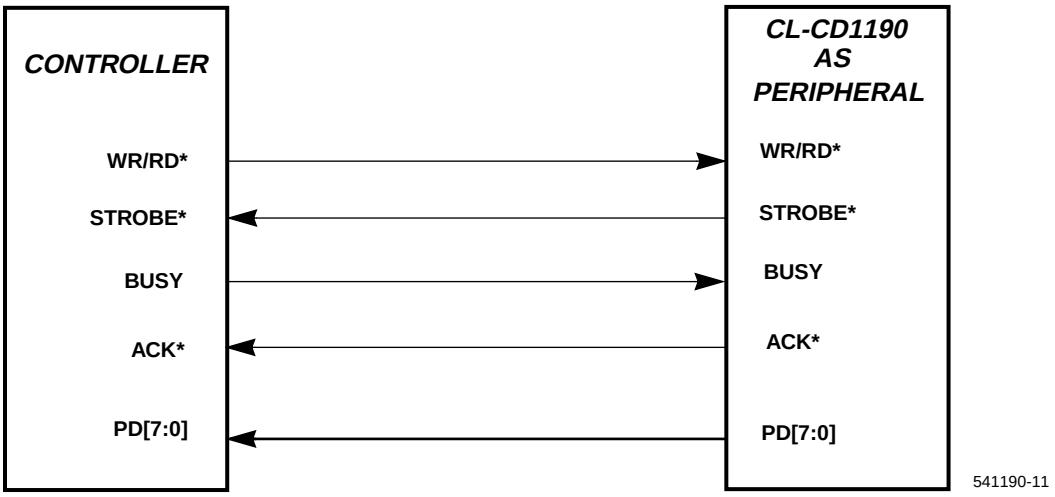


Figure 2–10. CL-CD1190 Peripheral Output Mode

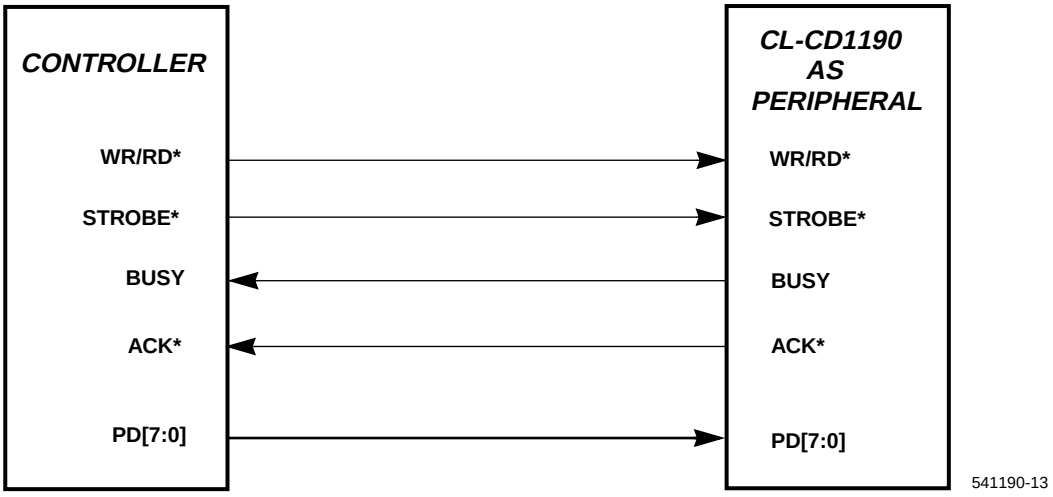


Figure 2–11. CL-CD1190 Peripheral Input Mode

2.7 Hardware Configurations

The simplicity of the host interface to the CL-CD1190 allows it to be built into systems making use of popular microprocessors, such as the Intel® 80X86 family (8086, 80286, 80386, etc.), the Motorola® family (68000, 68010, 68020, etc.), and the National® 32X32 family (32CG16, 32332, 32532, 32GX32, etc.).

2.7.1 Interfacing an Intel® Microprocessor-Based System

With very little extra logic, the CL-CD1190 can interface any system based on a processor in the Intel 80X86 family. The figure below shows a generalized view of an I/O-mapped interface with an 80286-based system. To provide the proper strobes and controls, the IOR* and IOW* control strobes are used to synthesize the DS* and R/W* signals. DTACK* is used as an input to wait state generation logic that will hold the processor (if necessary until the CL-CD1190 has completed the I/O request.

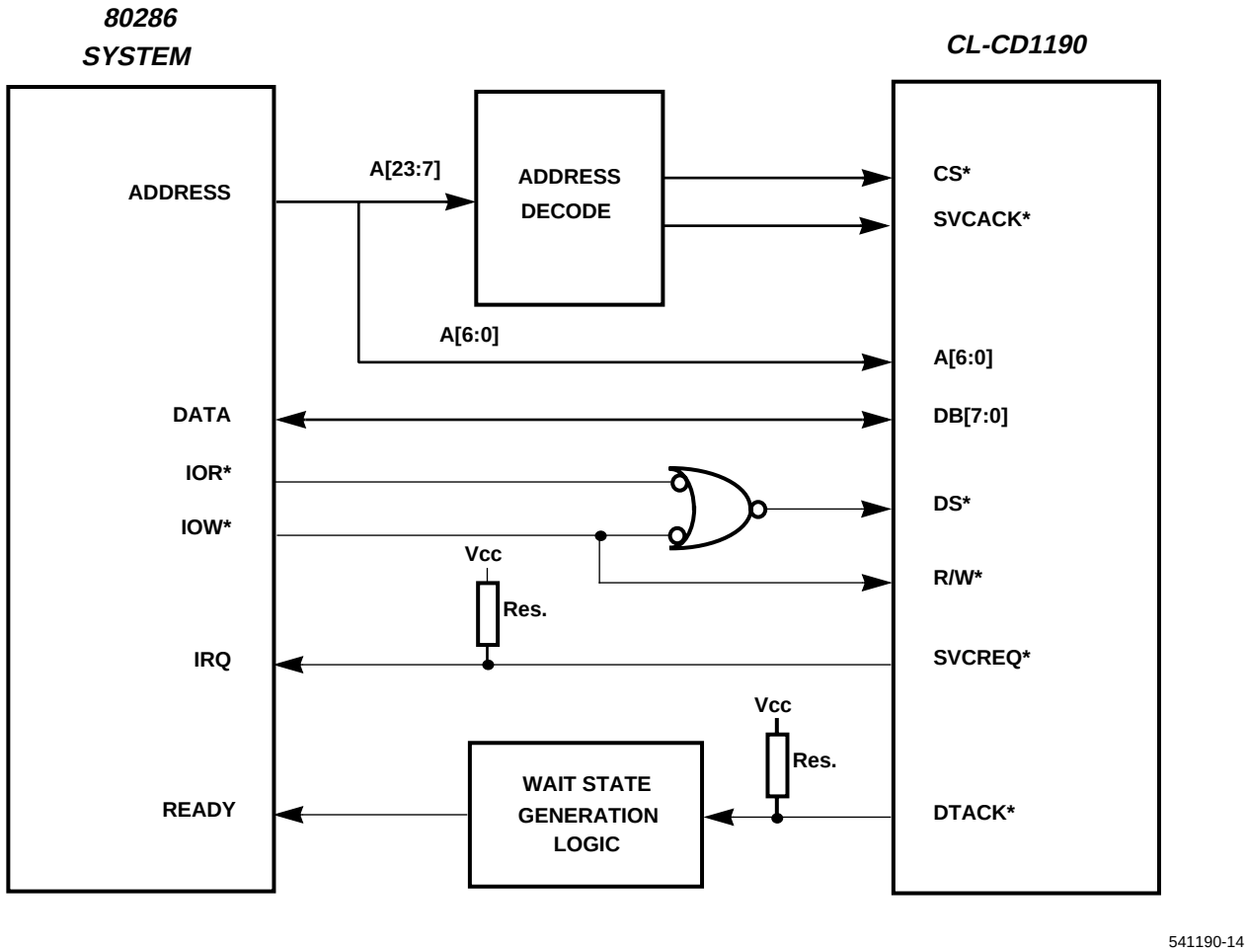
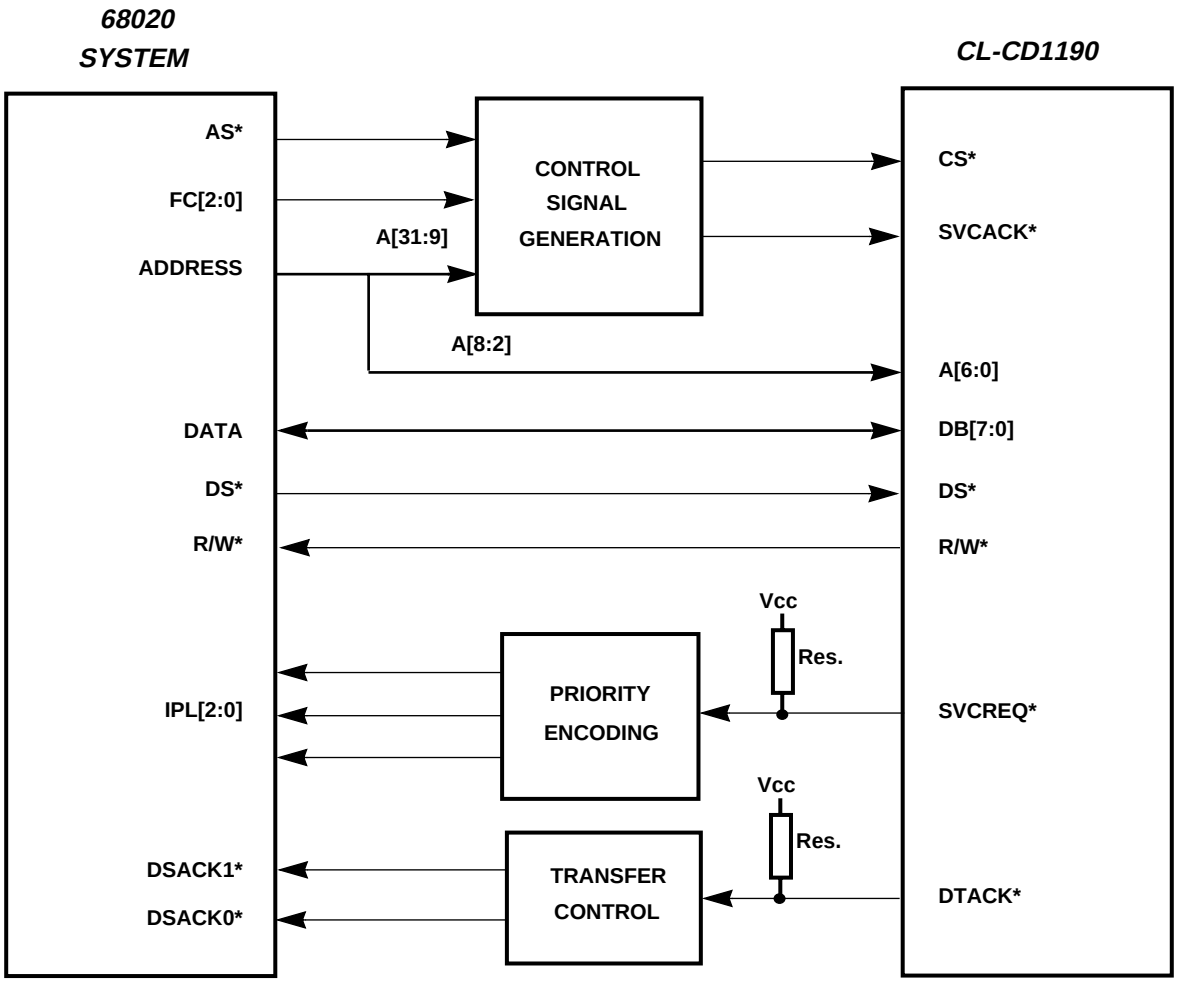


Figure 2–12. Intel® 80X86 Family Interface

2.7.2 Interfacing a Motorola® Microprocessor-Based System

Interfacing a 68000 family device is not difficult. The bus timing and the interface signal definitions very closely match those of the 68000 microprocessor, thus allowing direct connection in most cases. With later versions (68020, 68030), some additional logic is required to generate the DSACK0* and DSACK1* functions that replace the DTACK* on the earlier devices. The example below is a generalized interface to a 68020 device.



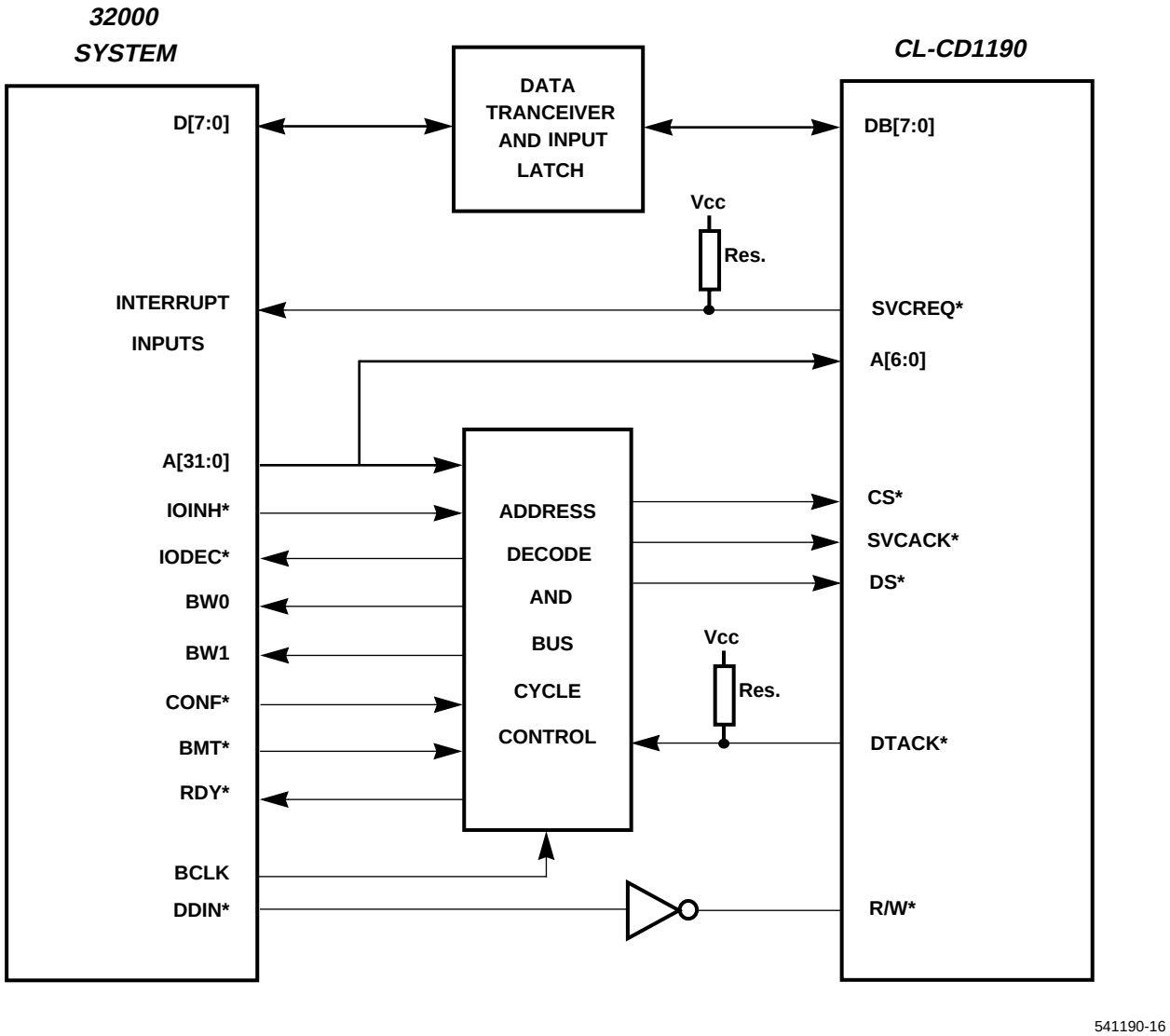
541190-15

Figure 2–13. Motorola® 68020 Interface

2.7.3 Interfacing a National Semiconductor® Microprocessor-Based System

The connections between the CL-CD1190 and a NS32000 (32GX320, 32CG16, etc.) embedded controller are also relatively simple. As with the Intel devices, cycles are controlled by the DS, CS and R/W Signals that have been synthesized from the available I/O control signals and I/O cycle extensions (wait states) are generated by logic connected to the DTACK Signal. Some additional consideration is required when implementing memory-mapped I/O to prevent multiple read and write cycles with the CL-CD1190 FIFOs due to the pipelined architecture of the 32000 device but all of the necessary controls are available.

Figure 2-14 depicts a simplified interface example.



541190-16

Figure 2–14. National® 32000 Interfac

Notes

3. CL-CD1190 PROGRAMMING

3.1 Overview

The CL-CD1190 host interface is made up of an array of registers. These registers control aspects of chip behavior. Most of the registers will only be set up once, during initialization, and only rarely modified during normal operation. This section describes these aspects, as well as the methods of interacting with the CL-CD1190 for parallel service needs.

3.2 Initialization

To operate a CL-CD1190, several procedures must be completed. These include chip initialization, programming global functions, such as interrupt vectors, and setting parallel port timing parameters. In most cases, initialization routines will only be executed once, during overall system boot-up. The following subsections describe these steps in detail.

3.2.1 Chip Initialization

The chip reset procedures will normally be executed after a power-up or system-wide reset; therefore, the CL-CD1190 has performed its own internal initialization by the hardware reset control signal, RESET*. However, it is a good practice to issue a software chip reset to verify completion prior to chip initialization. The following steps can be followed to accomplish this. (Figure 3–1 on the following page is a flow-chart version of the same steps).

1) Wait for FRR to become non-zero.

2) Write hexadecimal 81 (0x81) to the Channel Command Register (CCR).

This command causes the CL-CD1190 to perform a complete internal logic reset of all internal registers to their power-up reset values. The FIFO is flushed, and the port is disabled.

3) Wait for the firmware revision code to be written into the FRR.
This operation is used by the control logic to flag completion of the reset procedure. After reset, the FRR is one of the first registers to be cleared and is the last register set before normal operation begins. The initialization routine *must* wait for this register to become non-zero before beginning any other programming of CL-CD1190 registers. However, if the host code is sufficiently fast, it may begin testing the FRR before the control logic clears it; thus, one may assume that the CL-CD1190 has completed its internal initialization when, in fact, it has not. To avoid this error, the host software should look for the FRR to change to a zero and then to the current revision code. Alternatively, the host can clear the FRR just prior to issuing the global reset and then poll for the correct revision code. This would be useful in slow systems that cannot guarantee that the host will be able to check the register after it has been cleared, and before it is loaded with the revision code.

This procedure can also be used as part of a diagnostic test suite. The device will complete internal initialization within 500 µsec (with a 20-MHz clock). Therefore, a timer (software or hardware) can be used to detect if the operation does not finish within this time and if the chip is functional or not.

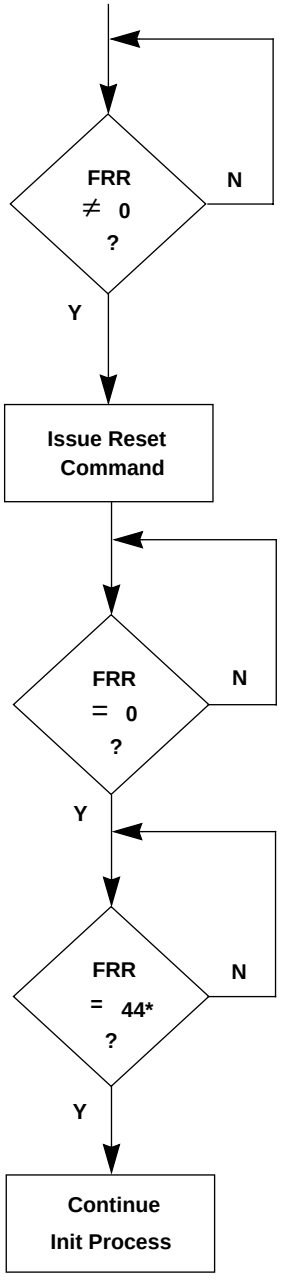
3.2.2 Global Function Initialization

Once chip reset has been completed, the next step is to set the global operating mode and other operational controls, such as timer period, strobe widths, etc.

1) Set the desired signal level on the WR/RD and general-purpose output (OP[2-0]) pins*

The signal level of the WR/RD* Output Pin must be set to the appropriate value, depending on the global operating mode (Controller Input, Controller Output). It may also be necessary to set the signal level of the general-purpose output port, depending on the design requirements.

All of these signal levels are set via the Signal Control Register (SCR).



* Revision code for
Revision C device = 44

Future revisions, if necessary, will increment this by one; for example, a Revision D device would have an FRR value of 45.

541190-17

Figure 3–15. Flow Diagram of CL-CD11990 Master Initialization Sequence

2) Set Interrupt Vector Register

The interrupt vector value supplied by the CL-CD1190 during the acknowledge cycle is important if hardware service request/acknowledge cycles are employed. It should be loaded into the Interrupt Vector Register (IVR) during the initialization sequence.

3) Set strobe widths

The Strobe Width (SWR) and Acknowledge Width (AWR) Registers are loaded with the appropriate value to generate the desired pulse widths. These registers can set a pulse width of any length, from 800 ns to 50 µs in 200-ns increments (@ CLK = 20 MHz); the minimum value is 4 and maximum value is 255.

4) Set the global configuration via the GCR and the CGC command of the CCR

The global configuration determines the mode of operation of the device. After reset, the GCR is set to all 0's. Thus, the CL-CD1190 defaults to the Controller Output Mode. If this is the intended mode for the device, nothing further needs to be done with this register. Otherwise, the mode must be set via the Control Bits (2-0). This is done by issuing the Change Global Configuration command to the CCR.

5) Set the Timer Prescale Register (TPR)

The TPR sets the master time 'tick' for the CL-CD1190. It is a binary value that sets the constant by which the system clock is divided (after a fixed prescale of 512) to produce the internal clock for the on-chip timers (*not* the strobe and acknowledge pulse widths, however). This clock is used for FIFO time-out generation and the master clock that drives the on-chip timer. For example, to generate a timer clock of 1 ms, the value is computed as:

$$\left(\frac{20MHz}{512}\right) \times 1ms = 39.0625$$

Or, for a 25-MHz system clock:

$$\left(\frac{25MHz}{512}\right) \times 1ms = 48.828001$$

The value 39 (49) would be loaded into the TPR and, in effect, selects an approximate 1 KHz clock as the clock source for the timers. This register sets the value that will be loaded into the master timer each time it reaches zero.

This value of 39 is the recommended minimum value that should be placed in the TPR. Values that generate a time period of less than 1 ms adversely affect the performance of the device and, thus, overall data transfer performance.

6) Set the Timer Multiplier (TMR) and Data Time-out (DTR) Registers

The TMR and DTR should be loaded with an appropriate value divisor to generate the desired time-out periods (timer and FIFO, respectively).

3.3 Program Examples

This section provides some examples of CL-CD1190 programming. Included are a chip initialization sequence, and interrupt driven send and receive routines. All examples are written in Borland Turbo C® and should be easily converted to other versions of 'C' or other languages.

3.3.1 Sample CL-CD1190 Initialization Routine

```
/* This procedure initializes the CL-CD1190 for controller output mode with a low
 * FIFO threshold (16 bytes) for systems with a short interrupt response time
 */
void    cd1190_init()
{
        int     i=0;
        outportb( SCR, 0x08 );              /* set WR/RD*, op[3,2,1] low */
        outportb( IVR, 0x30 );              /* set irq vector */
        outportb( SWR, 0x08 );              /* 1.6 µs STROBE* width, assuming 20-MHz CLK */
        outportb( AWR, 0x08 );              /* 1.6 µs ACK* width, assuming 20-MHz CLK */
        outportb( TPR, 39 );                /* timer period (39 decimal = 1 ms) */
        outportb( TMR, 0x40 );              /* timer multiplier */
        outportb( SR0, 0 );                 /* disable transitions to zero */
        outportb( SR1, 0 );                 /* disable transitions to one  */
        outportb( FTR, 0x10 );              /* Threshold = 16  */
        outportb( DTR, 0x40 );              /* data timer multiplier */
        outportb( GCR, CON|OUT );           /* set controller output mode */
        while( inportb(CCR) != 0 )          /* wait for CCR command to complete */
                i++;
        outportb( CCR, CGC );               /* tell controller GCR changed*/
        outportb( CCR, PAR | FLUSH | ENP | TIM|ENT ); /* enable parallel channel and timer */
        while( inportb(CCR) != 0 )/          * wait for CCR command to complete */
                i++;
        outportb( ICR, 0x84 );              /* data ready and master interrupts enabled */
}
```

3.3.2 Sample Output Interrupt Transfer Routine

```
/* This is an example of a routine to service an output data interrupt request*/

void interrupt cd1190_output()
{
        int     i, ist, fcr, txn;

        ist = inportb( ISR );

        if( ist & DRI )
        {
                fcr = inportb(FCR);             /* FIFO Count Reg */
                txn = 192 - fcr;                /* Number of chars to transmit */
                                                 /* use 128 for Rev. B, 192 for later */
                for( i = 0; i < txn; i++ )
                {
                        if( bufpos < buflen )
                        {
                                outportb( FDR, outbuf[bufpos] );
                                bufpos++;
                        }
                else if( rotate )
                        bufpos = 0;             /* Example loops on one buffer */
                                                /* Buffer management goes in this area */
                        else
                        {
                                outportb( ICR, 0 );
                                break;
                        }
        }
        outportb( ESR, 0 );
}
```

3.3.3 Sample Input Interrupt Transfer Routine

/* This is an example of a routine to service an input data interrupt request*/

```
void interrupt cd1190_input()
{
        int     i, ist, fcr;

        ist = inportb( ISR );

        if( ist & DRI )
        {
                fcr = inportb( FCR );                   /* read number of chars available in FIFO */
                for( i = 0; i < fcr; i++ )
                {
                        if( bufpos > BUFSIZ )           /* Example uses only one buffer */
                                bufpos = 0;             /* Buffer management goes in this area */

                        inbuf[ bufpos ] = inportb( FDR );
                        bufpos++;
                }
        }
        outportb( ESR, 0 );/                            * Write End Of Interrupt Register to terminate */
}
```

4. REGISTERS

All communication with the CL-CD1190 takes place through an array of registers. The tables on the following pages define the register symbols, names, read and write access modes, and the internal offset address for each register in the CL-CD1190; these are referenced to the A[6:0] inputs applied during a host I/O cycle. The offset address is shown in two ways: the absolute address as applied to all seven address inputs and a compressed address that is formed by connecting A[4] to ground and connecting A[6] and A[5] together. The compressed address allows the device address range to be reduced from occupying 128 consecutive address locations down to 32 address locations. This may be beneficial in some system applications. A detailed description of the host interface is presented in Section 2. The detailed descriptions of register bit definitions and programming are presented in Section 4.3.

Note that the addresses are shown relative to the CL-CD1190 definition of address lines. In 16- and 32-bit systems, it is a common practice to connect 8-bit peripherals to only one byte lane. Thus, in 16-bit systems, the CL-CD1190 appears at every other address, that is, the CL-CD1190 A0 is connected to the host A1, etc. In 32-bit systems, the CL-CD1190 appears at every fourth address, that is, the CL-CD1190 A0 is connected to the host A2, etc. In either of these cases, the addresses used by the programmer will be different than what is shown.

For instance, in a 16-bit Motorola 68000-based system, the CL-CD1190 is placed on data lines D0-D7, which are at odd addresses in the Motorola manner of addressing. The CL-CD1190 A0 is connected to the 68000 A1, etc. Thus, CL-CD1190 address x'08 becomes x'11 to the programmer. It is 'left-shifted' 1 bit; and A0 must be '1'. for low-byte (D0-D7) accesses.

4.1 CL-CD1190 Register Map

Symbol	Register Name	R/W	A[6:0]	(Hex)	(CA)[1]	Page[2]
AWR	ACK* Width Register	R/W	000 0011	03	03	43
CCR	Controller Command Register	R/W	000 1011	0B	0B	38
DTR	Data Time-out Register	R/W	000 1001	09	09	44
ESR	End Of Service Request Register	W	110 0000	60	10	48
FCR	FIFO Count Register	R	000 1110	0E	0E	47
FDR	FIFO Data Register	R/W	110 0010	62	12	47
FRR	Firmware Revision Register	R/W	000 1111	0F	0F	36
FTR	FIFO Threshold Register	R/W	000 1000	08	08	47
GCR	Global Configuration Register	R/W	000 1010	0A	0A	36
ICR	Interrupt Configuration Register	R/W	000 0001	01	01	46
IRR	Interrupt Request Register	R	110 0111	67	17	48
ISR	Interrupt Status Register	R/W	000 1100	0C	0C	46
IVR	Interrupt Vector Register	R/W	000 0000	00	00	45
PSR	Parallel Status Register	R	110 1100	6C	1C	44
SCR	Signal Control Register	R/W	110 1110	6E	1E	44

4.1 CL-CD1190 Register Map *(cont.)*

Symbol	Register Name	R/W	A[6:0]	(Hex)	(CA)[1]	Page[2]
SR0	Specification Register ZEROes	R/W	000 0110	06	06	45
SR1	Specification Register ONEs	R/W	000 0111	07	07	45
SSR	Signal Status Register	R/W	000 1101	0D	0D	45
SWR	STROBE* Width Register	R/W	000 0010	02	02	43
TMR	Timer Multiplier Register	R/W	000 0101	05	05	43
TPR	Timer Prescale Register	R/W	000 0100	04	04	43

NOTES: 1)The column labeled CA in the tables above is the compressed address. See Section 2 for a description of device addressing.

2)The page numbers shown in the table indicate the page where the detailed description of the register is located.

4.2 Register Definitions — Overview

ACK* Width Register **(AWR)** **03** **Read/Write**

Binary Value, Range 4 - 255

Controller Command Register **(CCR)** **0B** **Read/Write**

CGC	PAR	SIG	TIM	EN/DS	C2	C1	C0

Format 1: Change Global Configuration Command

CGC	0	0	0	0	0	0	RESET

Format 2: Parallel Commands

0	PAR	X	X	X	DIP	FLUSH	ENP

Format 3: Signal Commands

0	0	SIG	X	X	S_BSY	ACK	C_BSY

Format 4: Timer Commands

0	X	X	TIM	EN/DS	X	X	X

Data Timeout Register **(DTR)** **09** **Read/Write**

Binary Value, Range 0 - 255

4.2 Register Definitions — Overview *(cont.)*

End of Service Request Register (ESR) 60 (10) Write Only

X	X	X	X	X	X	X	X

FIFO Count Register (FCR) 0E Read Only

Binary Count Value (Range 0 - 192)

FIFO Data Register (FDR) 62 (12) Read/Write

8 - Bit Data to/from FIFO

Firmware Revision Register (FRR) 0F Read/Write

Firmware Revision Code

FIFO Threshold Register (FTR) 08 Read/Write

Binary Threshold Value (Range: 0 - 192)

Global Configuration Register (GCR) 0A Read/Write

VTYPE	FRSHR	0	ONLCR	OCRNL	NOACK	MODE	DIR

Interrupt Configuration Register (ICR) 01 Read/Write

IEN	X	X	X	IUA	IDR	ISS	ITE

Interrupt Status Register (ISR) 0C Read/Write

INT	0	0	0	UAI	DRI	SSI	TEI

Interrupt Request Register (IRR) 67 (17) Read Only

0	0	0	0	0	0	EXT*	INT*

Interrupt Vector Register (IVR) 00 Read/Write

Binary Value

4.2 Register Definitions — Overview *(cont.)*

Parallel Status Register (PSR) 6C (1C) Read Only

ENP	BUSY	ACK*	STROBE*	0	0	0	0

Signal Control Register (SCR) 6E (1E) Read/Write

Read Format

WR/RD*	IP3	IP2	IP1	WR/RD*	OP3	OP2	OP1

Write Format

X	X	X	X	WR/RD*	OP3	OP2	OP1

Specification Register ZEROes (SR0) 06 Read/Write

WR/RD*	IP3	IP2	IP1	0	0	0	0

Specification Register ONEs (SR1) 07 Read/Write

WR/RD*	IP3	IP2	IP1	0	0	0	0

Signal Status Register (SSR) 0D Read/Write

WR/RD*	IP3	IP2	IP1	0	0	0	0

STROBE* Width Register (SWR) 02 Read/Write

Binary Value, Range 4 - 255

Timer Multiplier Register (TMR) 05 Read/Write

Binary Value, Range 0 - 255

Timer Prescale Register (TPR) 04 Read/Write

Binary Value, Range 0 - 255

4.3 Detailed Register Descriptions

This section presents a complete, detailed description of each register. Registers have two formats: full eight bits, where the entire content defines a single function; or, the register is a collection of bits, grouped singly or in multiples, defining a function. In the second case, the descriptions break the register down into its component parts and describe the bits individually. The order of register presentation follows that given in the brief register descriptions on the previous pages.

Firmware Revision Register (FRR) 0F Read/Write

Firmware Revision Code

The FRR serves two purposes in the CL-CD1190. First, it displays the revision number of the firmware in the chip. When a revision to the CL-CD1190 is required, the revision number of the firmware is incremented by one. Beginning with Revision C, the code is hex 44. Later revisions will increment this by one; for example, Revision D would be hex 45, and so on.

Secondly, this register can be used by the system programmer as an indication of when the internal logic has completed reset procedures, after either a power-on reset (via the RESET* Input) or a software global reset (via the reset command in the CCR). Immediately after the reset operation begins, this register is cleared. When reset operations are complete, and the CL-CD1190 is ready to accept host accesses, the register is loaded with the revision code.

Global Configuration Register (GCR) 0A Read/Write

VTYPE	FRSHR	0	ONLCR	OCRNL	NOACK	MODE	DIR

The seven flags in the GCR are used to control the overall mode of operation for the CL-CD1190.

The MODE and DIR (direction) Bits define the operational mode of the CL-CD1190: Controller Output, Controller Input, Peripheral Output and Peripheral Input. See Section 2 for a complete description of the four modes. The values contained in these two bits are reflected on the MODE and DIR output pins (pins 15 and 14 respectively). These outputs can be used to control external buffers and other interface logic. Changes to the MODE and DIR pins do not take effect until the CGC command has been executed (see the explanation of controller commands under the Controller Command Register (CCR) heading).

The NOACK Bit specifies that only the BUSY Input Signal is to be used for handshake with the peripheral in Controller Output Mode. When this mode is selected, the ACK* Input is ignored by the CL-CD1190. The unsolicited acknowledge detect function is also disabled when NOACK Mode is selected. Setting NOACK in modes other than Controller Output have no effect on device operation.

The OCRNL and ONLCR Bits enable specific character manipulation of the output data stream that can be useful with some operating systems (UNIX®, for example) and some printers. OCRNL causes the device to convert any carriage return (CR) characters in the FIFO into New Line (NL) characters. The conversion takes place when the character has been removed from the FIFO and is being preparred for transmission. ONLCR causes a CR character to be inserted before any NL characters in the output stream. As with OCRNL, the insertion takes place when the NL is removed from the FIFO and before the NL is transmitted. Thus, in neither case is any extra FIFO space used and it remains as 192 bytes deep.

4.3 Detailed Register Descriptions *(cont.)*

If both ONLCR and OCRNL are set, then CR characters are converted to NL and NL characters are prepended with a CR. (Note that a prepended CR character is not then converted to a NL character.)

The FRSHR enables Fair Share™ interrupt prioritization. If this bit is set, the CL-CD1190 logic examines the state of the SVCREQ* signal before posting a service request and, if active (low), will not post the request until it becomes inactive. This mode is compatible with the Fair Share modes of other devices in the Cirrus Logic Data Communications family. Fair Share prevents one device from having exclusive access to the CPU, allowing other devices on the same interrupt request line equal access to the CPU.

If set, the VTYPE Bit will cause the CL-CD1190 to modify the contents of the IVR, during a SVCACK* cycle, with a code in the least significant two bits that identifies the interrupt request type to which it is responding. If multiple requests are pending internally, the CL-CD1190 prioritizes them as UAI, DRI, SSI and TEI (highest priority to lowest priority). For example, if both DRI and TEI are pending, it will respond with the DRI and then, once that request has been serviced, post another request for TEI. If VTYPE is set, the vector is modified according to the following table (see the description of the IVR for bit placement):

IT1	IT0	Type and Priority
0	0	TEI (Lowest)
0	1	SSI
1	0	DRI
1	1	UAI (Highest)

Register bit encoding is as follows:

Bit 7 VTYPE
Bit 6 FRSHR
Bit 5 Must be zero.
Bit 4 ONLCR
Bit 3 OCRNL
Bit 2 NOACK: No Acknowledge (functional in Controller Output Mode only).
Bit 1-0 Mode and Direction Bits, encoded as:

MODE	DIR	Function
0	0	Controller Output
0	1	Controller Input
1	0	Peripheral Output
1	1	Peripheral Input

4.3 Detailed Register Descriptions *(cont.)*

Controller Command Register (CCR) 0B Read/Write

CGC	PAR	SIG	TIM	EN/DS	C2	C1	C0

Format 1: Change Global Configuration Commands

CGC	0	0	0	0	0	0	Reset

Format 2: Parallel Commands

0	PAR	X	X	X	DIP	FLUSH	ENP

Format 3: Signal Commands

0	0	SIG	X	X	S_BSY	ACK	C_BSY

Format 4: Timer Commands

0	X	X	TIM	EN/DS	X	X	X

The Controller Command Register is used to issue commands directly to the control logic to control or change some functions of device operation. The upper four bits indicate which of four command types is being issued, and the lower four bits are parameters to those commands. When the command has been executed by the CL-CD1190, it will zero out the CCR. Therefore, two consecutive commands must wait for the CCR to be cleared after the first is issued, before the second is issued. In some cases, multiple commands may be issued at the same time, depending on the command/parameter bits that are set. This will be explained at the end of this register definition.

4.3 Detailed Register Descriptions *(cont.)*

Format 1: Change Global Configuration Commands

CGC	0	0	0	0	0	0	Reset

Bit 7 Change Global Configuration.

Whenever the host changes the MODE and DIR Bits in the Global Configuration Register, it must inform the on-chip processor by issuing the CGC command. This command causes parallel transfers to be disabled. They can be re-enabled via the Parallel commands (see below).

Bits 6-1 Must be zero.

Bit 0 Reset.

NOTE: When Bit 0 is set in conjunction with Bit 7 (CGC), a full-chip reset operation is initiated. This has the same effect as activation of the RESET* input pin. Parallel transfers are disabled and the device assumes the default conditions. When the command is issued, the FRR is cleared; when internal reset and initialization operations are complete, the FRR is loaded with the device firmware revision code. Clearing of the CCR is not an indication that the reset operation is complete; host software must wait for the FRR to be loaded with the firmware revision before further I/O operations with the device are attempted.

Format 2: Parallel Commands

0	PAR	X	X	X	DIP	FLUSH	ENP

This command format is used to activate/de-activate the parallel channel and flush the data FIFO. Commands are interpreted in a left to right order; thus the channel can be disabled, flushed and re-enabled in a single operation.

Bit 7 Must be zero.

Bit 6 Parallel command. Bit 6 is used in conjunction with Bits 2-0 to initiate operations on the parallel channel.

Bits 5-3 Don’t care.

Bit 2 DIP – disable parallel transfers. Parallel data transfers are disabled and the ENP Output Pin is reset.

Bit 1 FLUSH – flush FIFO. This command flushes all data from the FIFO, regardless of the state of the channel (enabled or disabled).

Bit 0 ENP – enable parallel transfers. Parallel data transfers are enabled and the ENP Output Pin is set.

The encoding of the three control parameters is shown in the table on the following page.

4.3 Detailed Register Descriptions *(cont.)*

DIP	FLUSH	ENP	Encoding
0	0	0	Not used
0	0	1	Enable parallel data transfers
0	1	0	Flush FIFO
0	1	1	Flush FIFO and enable parallel data transfers
1	0	0	Disable parallel data transfers
1	0	1	Disable and then re-enable parallel data transfers
1	1	0	Disable parallel data transfers and flush the FIFO
1	1	1	Disable parallel data transfers, flush FIFO, then enable parallel data transfers

Format 3: Signal Command:

0	0	SIG	X	X	S_BSY	ACK	C_BSY

The signal command is used to manually control the state of the BUSY and ACK* output pins. While these signals are normally under automatic control of the CL-CD1190, some special circumstances require the host to override the automatic functions. The CL-CD1190 imposes some restrictions on the use of the commands. For example, only peripherals are allowed to drive the ACK* Output; therefore, commands issued while in the Controller Mode will not affect the signal. Likewise, BUSY is only an output during input modes, so commands issued to change its state will not reach the pin during output modes.

Commands are interpreted in a left-to-right order, so a single command can set and clear (pulse) the BUSY Output. The duration of the pulse depends upon what the device is doing when the command is issued.

Issuing the ACK command will cause a pulse to be generated on the ACK* Output. The duration of the pulse is determined by the programmed value that is set by the AWR Register.

Encoding of the bits is as follows:

Bit 7-6 Must be zero.
Bit 4 SIG – Issue the signal control command.
Bits 4-3 Don’t care.
Bit 2 S_BSY – set the BUSY Output.
Bit 1 ACK – generate a single pulse on the ACK* Output.
Bit 0 C_BSY – clear the BUSY Output.

4.3 Detailed Register Descriptions *(cont.)*

S_BSY	ACK	C_BSY	Encoding
0	0	0	Not used
0	0	1	Clear (reset) the BUSY Output
0	1	0	Pulse the ACK* Output
0	1	1	Pulse ACK* and clear BUSY outputs
1	0	0	Set the BUSY Output
1	0	1	Set then clear the BUSY Output
1	1	0	Set the BUSY and pulse the ACK* outputs
1	1	1	Set BUSY, pulse the ACK* Output, then clear BUSY

Format 4: Timer Commands

0	X	X	TIM	EN/DS	X	X	X

Timer commands are used to enable and disable the on-chip timer. Additionally, this command can be issued in conjunction with signal or parallel commands since it does not use the lower three bits for parameters. More information on command mixing is provided below.

Bit 7 Must be zero.
Bits 6-5 Don't care.
Bit 4 TIM – issue timer command.
Bit 3 EN/DS – Enable/Disable parameter to the timer command.
Bits 2-0 Don't care.

The table below shows the bit encoding of the TIM and EN/DS Bits:

TIM	EN/DS	Function
0	0	No command executed
0	1	No command executed
1	0	Timer disabled
1	1	Timer enabled

4.3 Detailed Register Descriptions *(cont.)*

Command Mixing

In some cases, multiple commands may be issued at the same time. This ability depends upon what commands are issued. Commands that need parameters passed in the lower three bits may not be mixed because multiple commands could not share the information. However, commands may be mixed with the timer command since it does not require parameters. Commands are evaluated in a left-to-right order, so commands that are specified by the more significant bits will take precedence over lower significant bits. Thus, commands that set both Bits 6 and 5 (parallel and signal commands) will result in only the parallel command be executed; the signal command will be ignored.

The CGC and RESET commands require that all other bits (6-1) be zero so any commands issued with the upper four bits equal to 1001 through 1111 will result in no action.

As stated previously, timer commands may be issued with any of the parallel and signal commands, but not with the CGC or RESET command.

The table below summarizes the various bit settings and the resultant action taken by the CL-CD1190 in response to them.

CGC	PAR	SIG	TIM	Action
0	0	0	0	No action
0	0	0	1	Timer only
0	0	1	0	Signal only
0	0	1	1	Signal and Timer
0	1	0	0	Parallel only
0	1	0	1	Parallel and Timer
0	1	1	0	Parallel only (parallel takes precedence over signal)
0	1	1	1	Parallel and Timer (parallel takes precedence over signal)
1	0	0	0	Change GCR (or RESET)
1	0	0	1	No action
1	0	1	0	No action
1	0	1	1	No action
1	1	0	0	No action
1	1	0	1	No action
1	1	1	0	No action
1	1	1	1	No action

4.3 Detailed Register Descriptions *(cont.)*

Strobe Width Register **(SWR)** **02** **Read/Write**

Binary Value, Range 4 - 255

The Strobe Width Register (SWR) is used to program the width of the STROBE* pulse when the CL-CD1190 is in the Controller or Peripheral Output Mode (that is, STROBE* is an output). The value loaded in this register provides the constant for a counter that is driven by the master clock (CLK) divided by four. Each count represents a 160 ns time period, based on a 25 MHz clock, thus the range of pulse widths is from 0.64 µs to 40.8 µs. After power-on or host commanded reset, the CL-CD1190 loads this register with a default value of four (4). If no other initialization is performed, the device will generate 1.0 µs STROBE* pulses.

Acknowledge Width Register **(AWR)** **03** **Read/Write**

Binary Value, Range 4 - 255

The Acknowledge Width Register (AWR) is used to program the width of the ACK* pulse when the CL-CD1190 is in the Peripheral Input Mode (ACK* is an output). The value loaded in this register provides the constant for a counter that is driven by the master clock (CLK) divided by four. Each count represents a 160 ns time period, based on a 25 MHz clock, thus the range of pulse widths is from 0.64 µs to 40.8 µs. After power-on or host-commanded reset, the CL-CD1190 loads this register with a default value of four (4). If no other initialization is performed, the device will generate 1.0 µs ACK* pulses.

Timer Prescale Register **(TPR)** **04** **Read/Write**

Binary Value, Range 0 - 255

The TPR sets the divisor that will be used to generate the time period for CL-CD1190 timer operations. It can be set to any value between 0 and 255 (x'FF). The TPR is clocked by the system clock prescaled (divided) by 512. With a 25 MHz clock, each count of the TPR provides a 20.48 µs time duration. A value of zero represents a count of 256 yielding a time duration of 5.243 ms. The internal counter is reloaded from the TPR each time the count expires (reaches zero). By default, this register is loaded with the value hex 40 during device initialization.

Timer Multiplier Register **(TMR)** **05** **Read/Write**

Binary Value, Range 0 - 255

The Timer Multiplier Register (TMR) is decremented once each time the TPR counter expires. When the TMR decrements to zero, a timer interrupt will be generated if the enable (ITE) bit is set in the Interrupt Configuration Register (ICR). A value of zero represents a count of 256, and yields a maximum timeout of 1.34 seconds, based on a 25MHz system clock and the maximum TPR count of 256 (see TPR above).By default, this register is loaded with the value hex 40 during device initialization.

4.3 Detailed Register Descriptions *(cont.)*

Data Time-out Register (DTR) 09 Read/Write

Binary Value, Range 0 - 255

This register sets the period for FIFO data time-outs during receive operations. The time-out function is implemented with a two-stage countdown timer. The DTR holds the time constant for an internal counter (DTC — not host accessible), which is reloaded whenever new data is received from the parallel interface and placed in the FIFO. The default value for this reigster is zero.

The DTR is decremented each time the TPR counter expires. If no new data is received (thus reloading the counter) before the counter reaches zero, the FIFO count register is checked. If data is in the FIFO, a FIFO data interrupt is generated. This prevents 'stale' data from collecting in the FIFO if the amount of data does not reach the programmed threshold.

Parallel Status Register (PSR) 6C (1C) Read/Write

ENP	BUSY	ACK*	STROBE*	0	0	0	0

The Parallel Status Register provides a means for the host to examine the state of three of the physical pins on the CL-CD1190. The ENP Bit indicates when parallel data transfers are enabled or disabled in response to an enable or disable command issued via the CCR Register. When ENP is a '1', transfers are enabled and when '0', disabled.

BUSY, STROBE* and ACK* are made available to the host for the purpose of detecting extraordinary conditions, such as a stuck interface or when a peripheral has been initialized. Internal control logic makes use of this same register to handle the handshake for data transfers so the data is a direct reflection of current activity in the interface. When the STROBE* or ACK* Pin are active (low), the STROBE* or ACK* Bit in the PSR will be a '1'. When the BUSY Pin is high, the BUSY Bit will be a '1'.

Signal Control Register (SCR) 6E (1E) Read/Write

Read Format

WR/RD*	IP3	IP2	IP1	WR/RD*	OP3	OP2	OP1

Write Format

X	X	X	X	WR/RD*	OP3	OP2	OP1

The Signal Control Register allows the host access to the general-purpose input and output ports (IP[3:1], OP[3:1]) and the WR/RD* Pin of the CL-CD1190. The register has two formats, one for read and one for write. During read cycles, the CL-CD1190 will return the states of all of the defined I/O pins. During write operations, only the least significant four bits are meaningful; data written to the input port bits is 'don't care'.

4.3 Detailed Register Descriptions *(cont.)*

Since WR/RD* can be either an input or an output depending on the mode of operation (controller/peripheral), it appears as both an input and an output in the SCR. WR/RD* is only active as an output during Controller Mode Operation so the device must be in this mode before writes to Bit 3 will activate the output pin. In all operational modes, Bit 7 reflects the true state of the WR/RD* Pin.

After reset, the CL-CD1190 defaults to Controller Output Mode; therefore, the SCR is loaded with hex 08 to activate the WR/RD* Pin for output operation and to clear the bits in the output port.

Specification Register ZEROes (SR0) 06 Read/Write

WR/RD*	IP3	IP2	IP1	X	X	X	X

Specification Register ONEes (SR1) 07 Read/Write

WR/RD*	IP3	IP2	IP1	X	X	X	X

These two registers specify the conditions on the input port that will cause a Signal Status Interrupt to be posted in the Interrupt Status Register. SR0 enables the detection of signal changes from logic one to logic zero on the corresponding bit of the input port. SR1 specifies changes from logic zero to logic one.

Whenever a change is detected in the value of one of the inputs in the SCR, the changed bits are masked with the values in both Specification Registers. If a change is indicated, the changed bits are set in the Signal Status Register and the SSI interrupt is posted in the ISR.

Signal Status Register (SSR) 0D Read/Write

WR/RD*	IP3	IP2	IP1	0	0	0	0

The Signal Status Register is used by the CL-CD1190 to indicate that one of the input pins has changed according to the conditions set by the programming in the SR0 and SR1 Registers. When a signal change occurs, the appropriate bit is set in the SSR and an Signal Change Interrupt (SSI) will be generated if enabled. The SSR is cleared after the host terminates a service acknowledge sequence by writing the ESR Register. Since the clearing operation may be delayed, high-performance polled mode systems may write zeroes to this register.

Interrupt Vector Register (IVR) 00 Read/Write

8-Bit Vector Value

This register specifies the value that will be driven onto the data bus during a service acknowledge cycle. The value programmed can be any value host software requires.

4.3 Detailed Register Descriptions *(cont.)*

Interrupt Configuration Register (ICR) 01 Read/Write

IEN	X	X	X	IUA	IDR	ISS	ITE

The Interrupt Configuration Register is used to select which of the four possible interrupting sources is enabled to produce a service request. The register also is used to arm the overall interrupt structure.

The four interrupt sources are:

ITE Interrupt on Timer Expired
ISS Interrupt on Signal Status
IDR Interrupt on Data Ready (FIFO threshold or FIFO timer)
IUA Interrupt on Unsolicited Acknowledge

When any of these conditions occur, the CL-CD1190 examines the ICR to determine if the condition is enabled to generate a service request to the host. If it is enabled, the corresponding bit is set in the ISR. If the Interrupt Enabled (IEN) Bit is also set in the ICR, the Service Request (SVCREQ*) Output Pin will also be activated.

The ITE Bit allows a service request to be posted whenever the programmed time-out period has been reached, as programmed by the TPR and TMR registers pair. The ISS Bit enables service requests when any of the enabled signal changes occur (SR0, SR1). The IDR Bit is used to enable a service request whenever the data in the FIFO reaches the programmed threshold level or when the FIFO timer expires. Lastly, the IUA Bit enabled service requests to be posted whenever an acknowledge pulse is received that is not associated with a strobe pulse.

Interrupt Status Register (ISR) 0C Read/Write

INT	X	X	X	UAI	DRI	SSI	TEI

The Interrupt Status Register provides current status of pending interrupt conditions within the CL-CD1190. When a service request is asserted, the ISR will contain bits corresponding to the cause (or causes) of the service request. The ISR is automatically cleared after a write access to the ESR, but clear may be delayed. High-performance polled mode systems should write zeroes into this register after it has been read.

Each of the bits corresponds to a bit in the ICR: TEI to ITE, SSI to ISS, DRI to IDR and UAI to IUA. The INT Bit is a simple OR of the cause bits. It will be set whenever one of the status bits is set, regardless of the state of the IEN Bit. Systems that operate in a poll mode may examine this bit to determine if any service requests are pending. The bits are defined on the following page.

4.3 Detailed Register Descriptions *(cont.)*

Interrupt Status Register Bits

TEI Timer Expired Interrupt
SSI Signal Status Interrupt
DRI Data Ready Interrupt (FIFO threshold or FIFO timer)
UAI Unsolicited Acknowledge Interrupt
INT Interrupt Request

FIFO Count Register (FCR) 0E Read Only

FC[7]	FC[6]	FC[5]	FC[4]	FC[3]	FC[2]	FC[1]	FC[0]

The FIFO Count Register contains the current count of the number of bytes in the FIFO. It is a binary value that can be in the range of 0 to 192 (0x00 - 0xC0). The value is used by host software to determine how many bytes may be read during a receive data FIFO service.

FIFO Threshold Register (FTR) 08 Read/Write

FT[7]	FT[6]	FT[5]	FT[4]	FT[3]	FT[2]	FT[1]	FT[0]

The FIFO Threshold Register is used in both input and output modes. It sets the data level at which the CL-CD1190 will post a FIFO service request and can be set to any value in the range of 0 to 192 (0x00 - 0xC0).

During an Input Mode Operation, whenever the FIFO level reaches the level programmed in the FTR, a service request will be posted via the DRI Bit in the ISR. During an Output Mode Operation, a DRI service request will be posted whenever the FIFO level falls below the programmed value in the FTR.

4.3.1 Virtual Registers

The CL-CD1190 has two registers that must only be accessed during a service acknowledge routine, and as such are considered to be ‘virtual’ registers. These are the FDR and ESR Registers.

FIFO Data Register (FDR) 62 (12) Read/Write

8-Bit Value

The FIFO Data Register is the host access port to the CL-CD1190 FIFO. When the device is programmed for input, the host may read any number of bytes from this location up to the number specified in the FCR. During an Output Mode Operation, the host may write up to 192 bytes into this location, depending upon the number of free locations available, as determined by the FCR.

4.3.1 Virtual Registers *(cont.)*

It is important to note that this register may not be accessed until the CL-CD1190 has requested FIFO service via the DRI service request. The FDR is available for host access only while the CL-CD1190 is in the 'service pending' context.

End of Service Register (ESR) 60 (10) Write Only

X	X	X	X	X	X	X	X

The End of Service Register is used by the host to tell the CL-CD1190 that the service acknowledge sequence is complete. When the host has read/written all the data to/from the FIFO, it must perform a write to this location. The data written is 'don't care'; the CL-CD1190 only needs the write cycle to determine if the service is complete and cause it to switch out of the service pending context.

Interrupt Request Register (IRR) 67 (17) Read Only

0	0	0	0	0	0	EXT*	INT*

The Interrupt Request Register allows the host to examine the state of both the internal and external service request functions. This can be useful in polled daisy chain systems made up of multiple CL-CD1190s (or other devices, all of which are sharing the same interrupt request line). If this CL-CD1190 is posting a service request (SVCREQ* is active), both the INT* and EXT* Bits will be '1'. If the INT* Bit is '0' and the EXT* Bit is '1', then some other device is driving the interrupt request line.

Note: On revision B and earlier versions of the CL-CD1190, this location does not exist. Unpredictable behavior of the device may occur if this location is accessed.

5. ELECTRICAL SPECIFICATIONS

5.1 Absolute Maximum Ratings

Supply voltage (V_{CC}) ..... +7.0 Volts

Input voltages, with respect to ground ..... -0.5 Volts to V_{CC} +0.5 Volts

Operating temperature (T_A) ..... 0° C to 70° C

Storage temperature ..... -65° to 150° C

Power dissipation ..... 0.25 Watt

NOTE: Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only, and functional operation of the device at these or any conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

5.2 Recommended Operating Conditions

Supply voltage (V_{CC}) ..... 5V ± 5%

Operating free air ambient temperature (T_A) ..... 0° C < T_A < 70° C

System clock ..... 25 MHz

Before beginning any new design with this device, please contact Cirrus Logic, Inc., for the latest errata information. See the back cover of this document for sales office locations and phone numbers. This data sheet applies to CL-CD1190 Revision C or later devices.

5.3 DC Electrical Characteristics

(@ V_{CC} = 5V ± 5%, T_A = 0° C to 70° C)

Symbol	Parameter	MIN	MAX	Units	Test Conditions
V_{IL}	Input Low Voltage	-0.5	0.8	V	
V_{IH}	Input High Voltage	2.0	V_{CC}	V	Note 1
V_{OL}	Output Low Voltage		0.4	V	I_{OL} = 2.4 mA, Note 2
V_{OH}	Output High Voltage	2.4		V	I_{OH} = -400 µA
I_{IL}	Input Leakage Current	-10	10	µA	0 < V_{IN} < V_{CC}
I_{LL}	Data Bus 3-state				
	Leakage current	-10	10	µA	0 < V_{OUT} < V_{CC}
I_{OC}	Open Drain Output				
	Leakage current	-10	10	µA	0 < V_{OUT} < V_{CC}
I_{CC}	Power Supply Current		50	mA	CLK = 25 MHz, Note 3

5.3 DC Electrical Characteristics *(cont.)*

Symbol	Parameter	MIN	MAX	Units	Test Conditions
C_{IN}	Input Capacitance		10	pF	
C_{OUT}	Output Capacitance		10	pF	

NOTES: 1) V_{IH} is 2.7V minimum on RESET* and CLK, and 2.2V minimum on Pin 61.
2) V_{OL} for open drain signals is 0.5V @ 16.0 mA sinking.
3) Typical I_{CC} is 25 mA.

NOTE: While the CL-CD1190 is a highly dependable device, there are a few guidelines that will help to ensure that the maximum possible level of overall system reliability is achieved. First, the PC board should be designed to provide maximum isolation of noise. A four-layer board is preferable, but a two-layer board will work if power and ground distribution is good. In either case, decoupling capacitors mounted close to the CL-CD1190 are strongly recommended. Noise typically occurs when either the CL-CD1190 data bus drivers come out of tristate to drive the bus during a read, or when an external bus buffer turns on during a write cycle. This noise, a rapid rate-of-change of supply current, causes 'ground bounce' in the power distribution traces. This ground bounce, a rise in the voltage of the ground pins, effectively raises the input logic thresholds of all devices in the vicinity, resulting in the possibility of a '1' being interpreted as a '0'.

To reduce the possibility of ground bounce affecting the operation of the CL-CD1190, we have specified the input-high voltage (V_{IH}) of the CLK and RESET* Pins at 2.7 volts, instead of the TTL-standard 2.0 volts. We have found that this eliminates any sensitivity to ground bounce, even in very noisy systems.

Although 2.7 volts is higher than the industry-standard 2.4 volt output (V_{OH}) specified for TTL, there are several simple ways to meet this specification. One choice is to use any of the advanced CMOS logic families which are on the market (FACT, ACL, etc.). These CMOS output buffers will pull up to close to V_{CC} when not heavily loaded. In addition, AS and ALS TTL may be used if the output of the TTL device is only driving 1 or 2 CMOS loads. As noted in the Texas Instruments *ALS/AS Logic Data Book* (1986), pages 4-18 and 4-19, the V_{OH} output of these families exceeds 3.0 volts at low current loading. Other manufacturers publish similar data. Cirrus Logic recommends the use of one of these two options for the CLK input, to insure fast clean edges. Note that the RESET* Pin may, if desired, be pulled up passively with a 1K ohm or less resistor.

5.4 AC Characteristics

5.4.1 Index of Timing Information

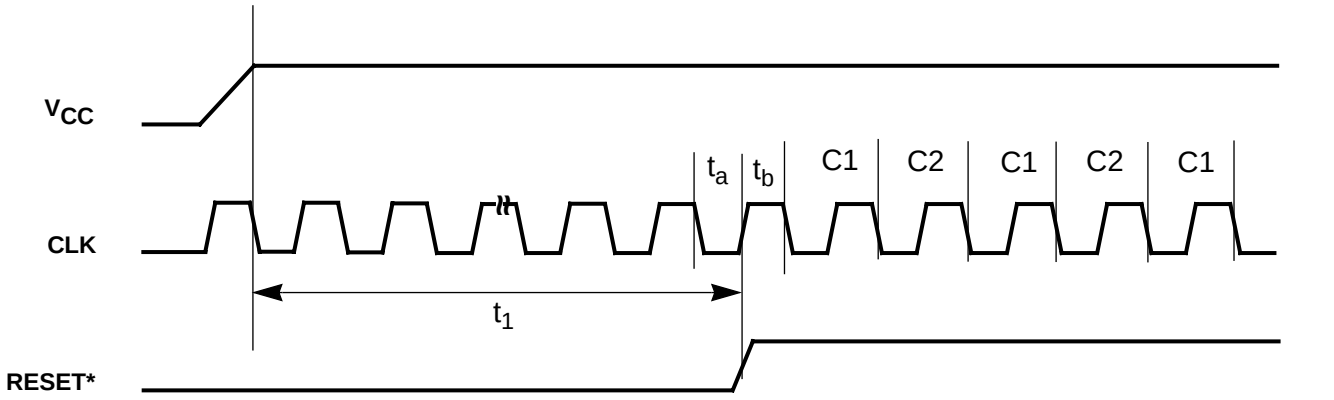
5.4.2 Asynchronous Timing

Refer to Figures 5–1a through 5–1e on the following pages for the reference numbers in the table below. (@ V_{CC} = 5V ± 5%, T_A = 0° C to 70° C)

Table 5–1. Transmit Timing *(Figures 5–1a and 5–1c)*

Ref. #	Fig.	Parameter	MIN	MAX	Unit
t_1	5–1a	RESET* Low pulse width	10		T_{CLK}
t_2	5–1c	Address setup time to CS* or DS*		-20	ns
t_3	5–1c	R/W* setup time to CS* or DS*		-10	ns
t_4	5–1c	Address hold time after CS*	0		ns
t_5	5–1c	R/W* hold time after CS*	0		ns
t_6	5–1c	Read data valid to DTACK* low		10	ns
t_7	5–1c	DTACK* low from CS* or DS*	$3T_{CLK}$	$5T_{CLK}$+40	ns
t_8	5–1c	Data Bus Tri-state after CS* or DS* high	0	30	ns
t_9	5–1c	CS* or DGRANT* high from DTACK* low	0		ns
t_{10}	5–1c	DTACK* inactive from CS* or DGRANT* and DS* high		40	ns
t_{11}	5–1c	DS* high pulse width	10		ns
t_{12}	5–1d	Write data valid from CS* and DS* low		$2\ T_{CLK}$	ns
t_{13}	5–1d	Write data hold time after DS* high	0		ns
t_{14}	5–1b	Clock period (TCLK) (Note)	40	500	ns
t_{15}	5–1b	Clock low time (Note)	$0.4T_{CLK}$	$0.6T_{CLK}$	ns
t_{16}	5–1b	Clock high time (Note)	$0.4T_{CLK}$	$0.6T_{CLK}$	ns
t_{17}	5–1e	Propagation delay, DGRANT* and DS* to DPASS*		35	ns
t_{18}	5–1e	Setup time, SVCACK* to DS* and DGRANT*	10		ns

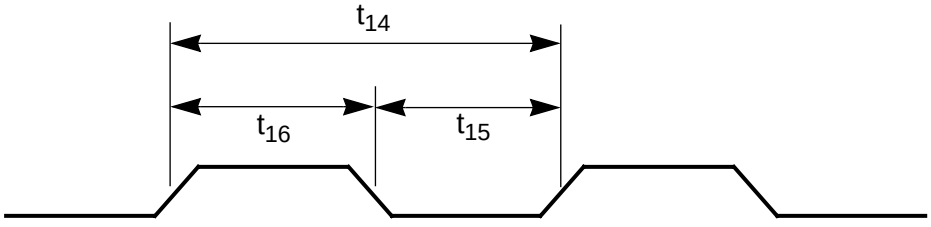
NOTE: Timing numbers for RESET* and CLK in the table above are valid for both asynchronous and synchronous specifications. The device will operate on any clock with a 40-60 duty cycle or better.



541190-18

NOTE: For synchronous systems, it is necessary to know the clock cycle number so that interface circuitry can stay in lock-step with the device. CLK numbers can be determined if RESET* is released within the range t_a - t_b. t_a is defined as 10 ns minimum after the falling edge of the clock; t_b is defined as 5 ns minimum before the next falling edge of the clock. If these conditions are met, the cycle starting after the second falling edge will be known to be C1. See the synchronous timing diagrams for additional information. Asynchronous systems need not be concerned with clock cycle numbers.

Figure 5–1a. Reset Timing



541190-19

Figure 5–1b. Clock Timing

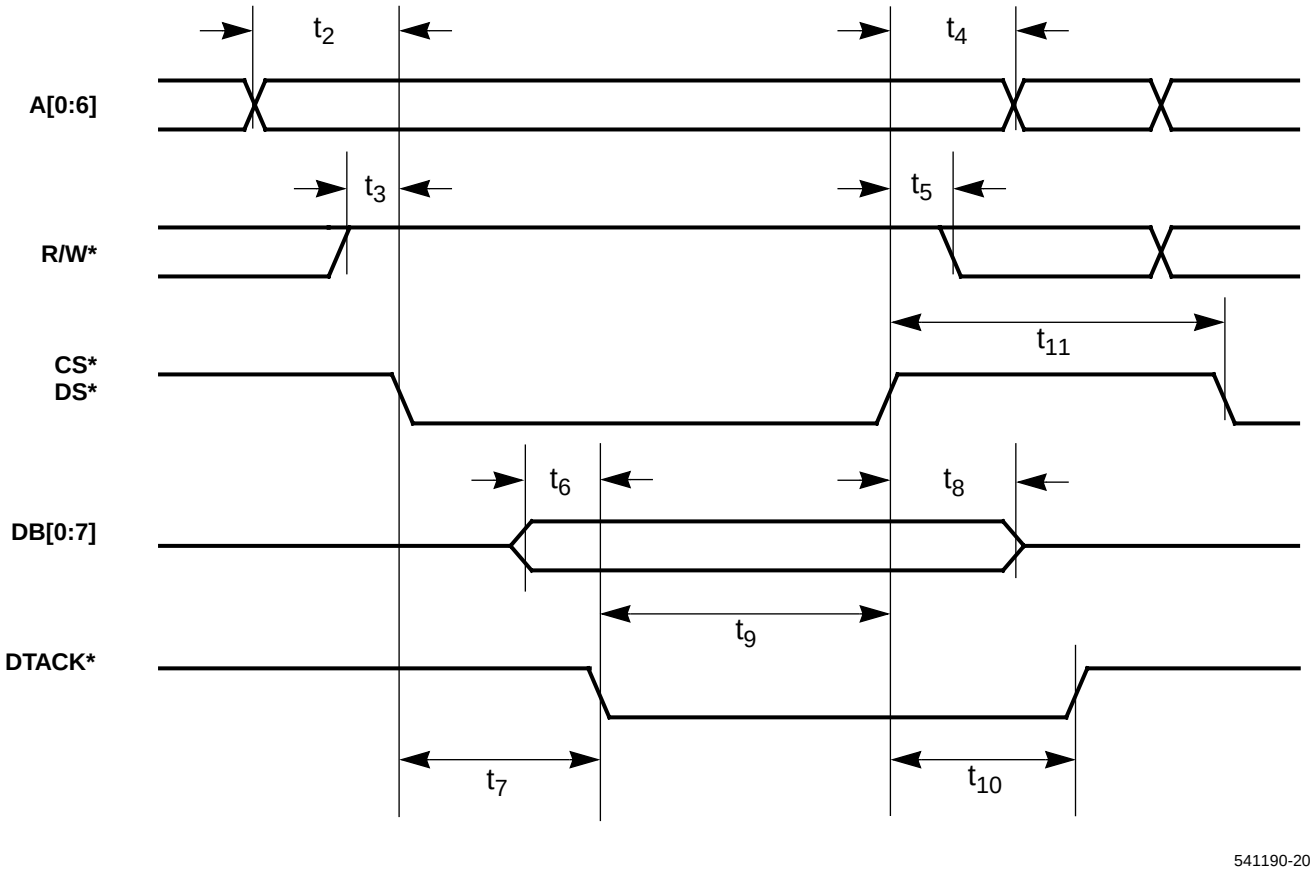


Figure 5–1c. Asynchronous Read Cycle Timing

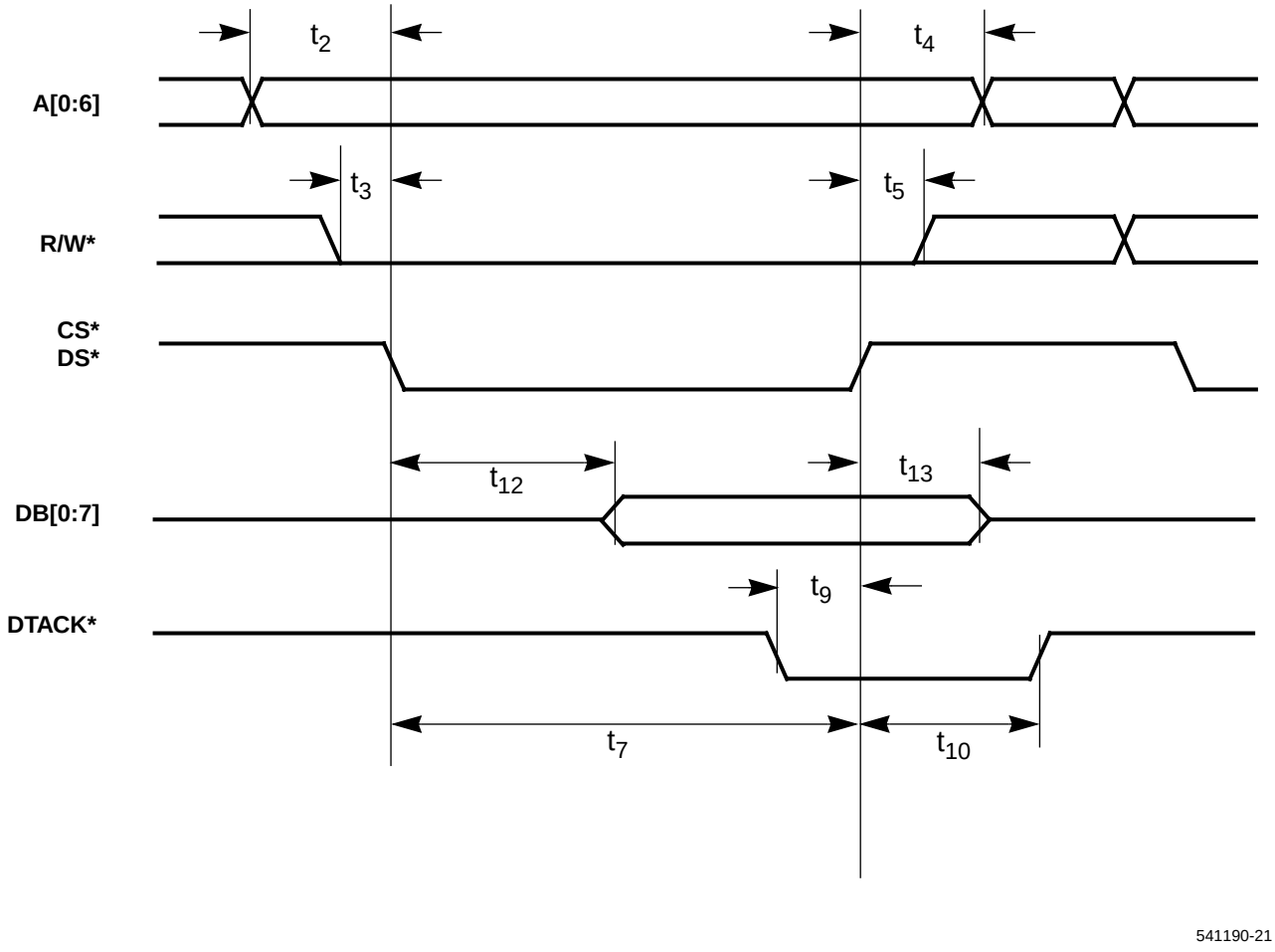


Figure 5–1d. Asynchronous Write Cycle Timing

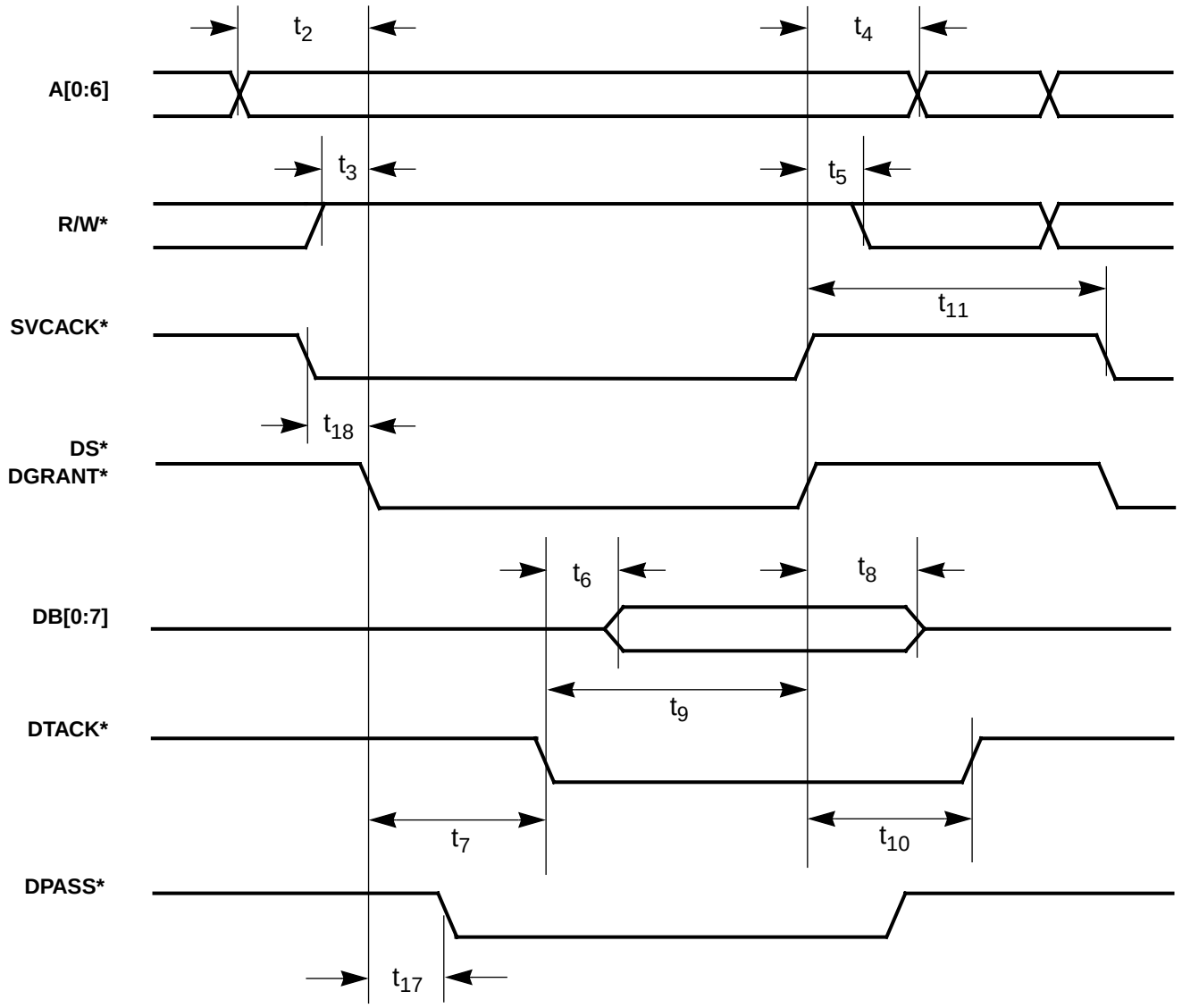


Figure 5–1e. Asynchronous Service Acknowledge Cycle Timing

5.4.3 Synchronous Timing

Refer to Figures 5–2a through 5–2c on the following pages for the reference numbers in the table below.

Table 5–2. Transmit Timing *(Figures 5–2a and 5–2c)*

Ref. #	Fig.	Parameter	MIN	MAX	Unit
t_1	5–2a	Setup time, CS* and DS* to C1 falling edge	5		ns
t_2	5–2a	Setup time, R/W* to C1 falling edge		-10	ns
t_3	5–2a	Setup time, address valid to C1 falling edge		-20	ns
t_4	5–2a	C3 falling edge to data valid		60	ns
t_5	5–2a	DTACK* low from C4 falling edge		40	ns
t_6	5–2a	CS* and DS* trailing edge to data bus high-impedance		30	ns
t_7	5–2a	CS* and DS* inactive between host accesses	10		ns
t_8	5–2a	Hold time, R/W* after C4 falling edge	20		ns
t_9	5–2a	Hold time, address valid after C4 falling edge	0		ns
t_{10}	5–2b	Setup time, write data valid to C3 falling edge	0		ns
t_{11}	5–2c	Setup time, DS* and DGRANT* to C1 falling edge	20		ns
t_{12}	5–2c	Setup time, SVCACK* to DS* and DGRANT* (last active)	10		ns
t_{13}	5–2c	Hold time, write data valid after C4 falling edge	0		ns
t_{14}	5–2c	Propagation delay, DS* and DGRANT* to DPASS*		35	ns

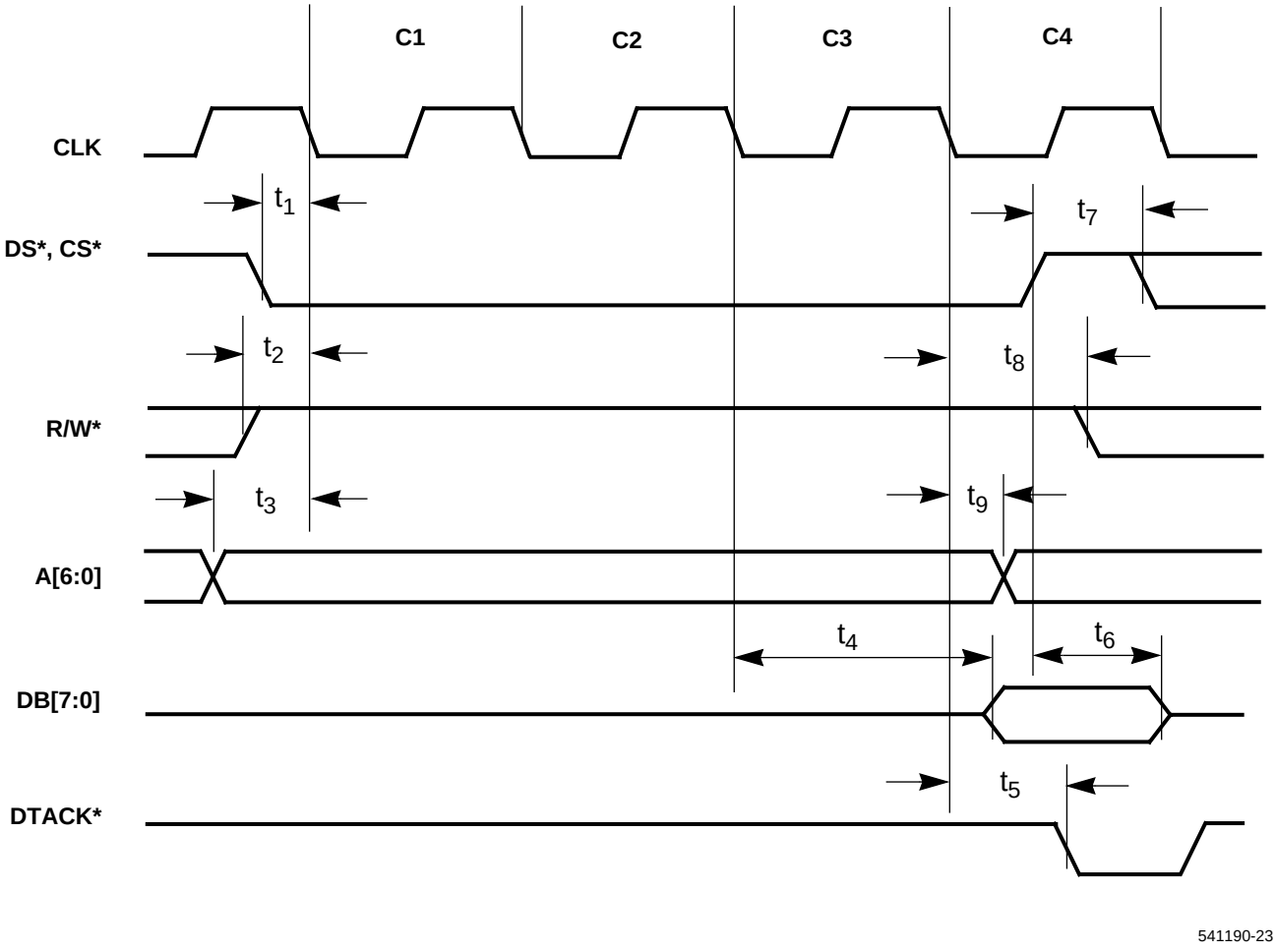


Figure 5–2a. Synchronous Read Cycle Timing

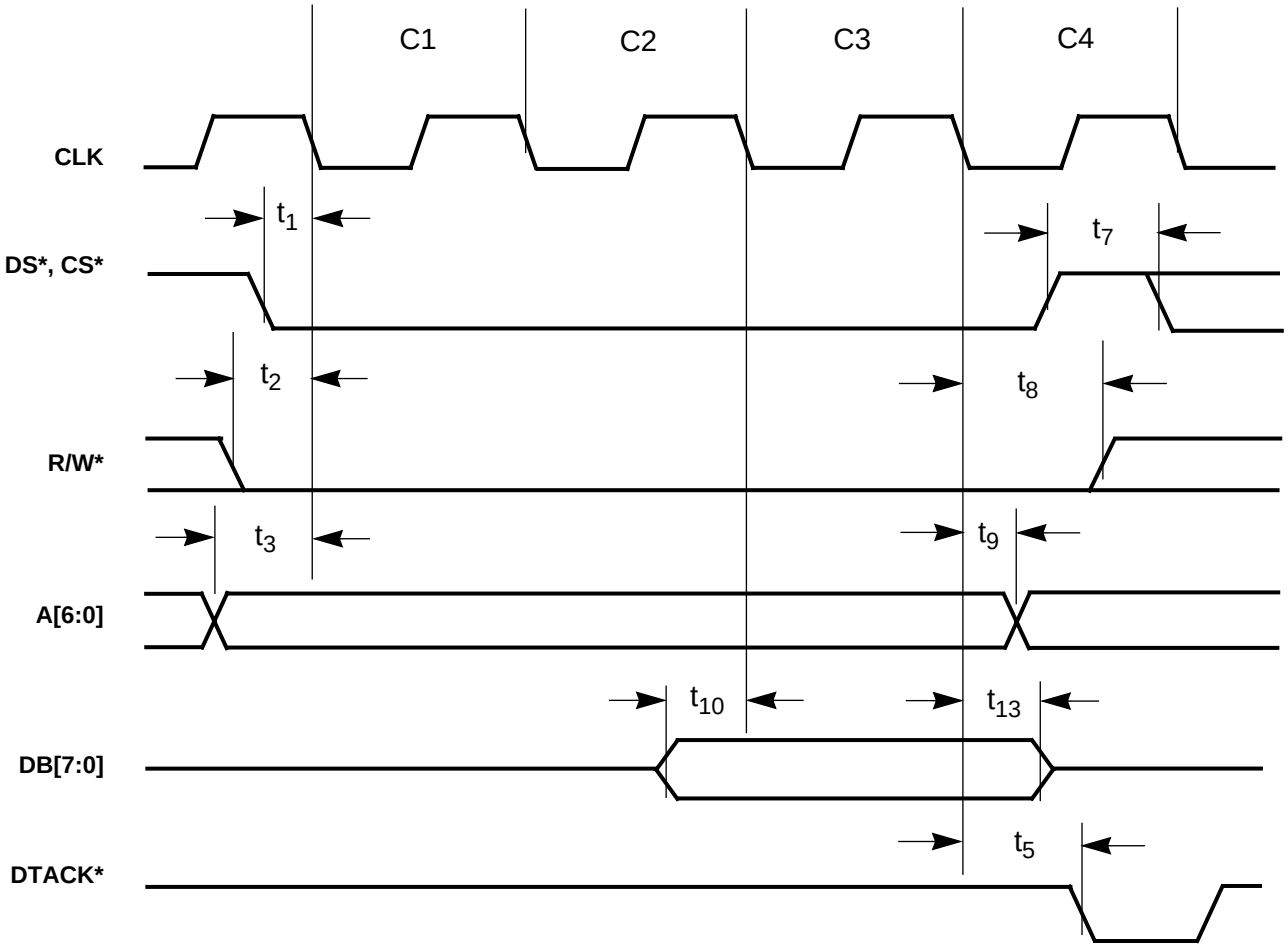


Figure 5–2b. Synchronous Write Cycle Timing

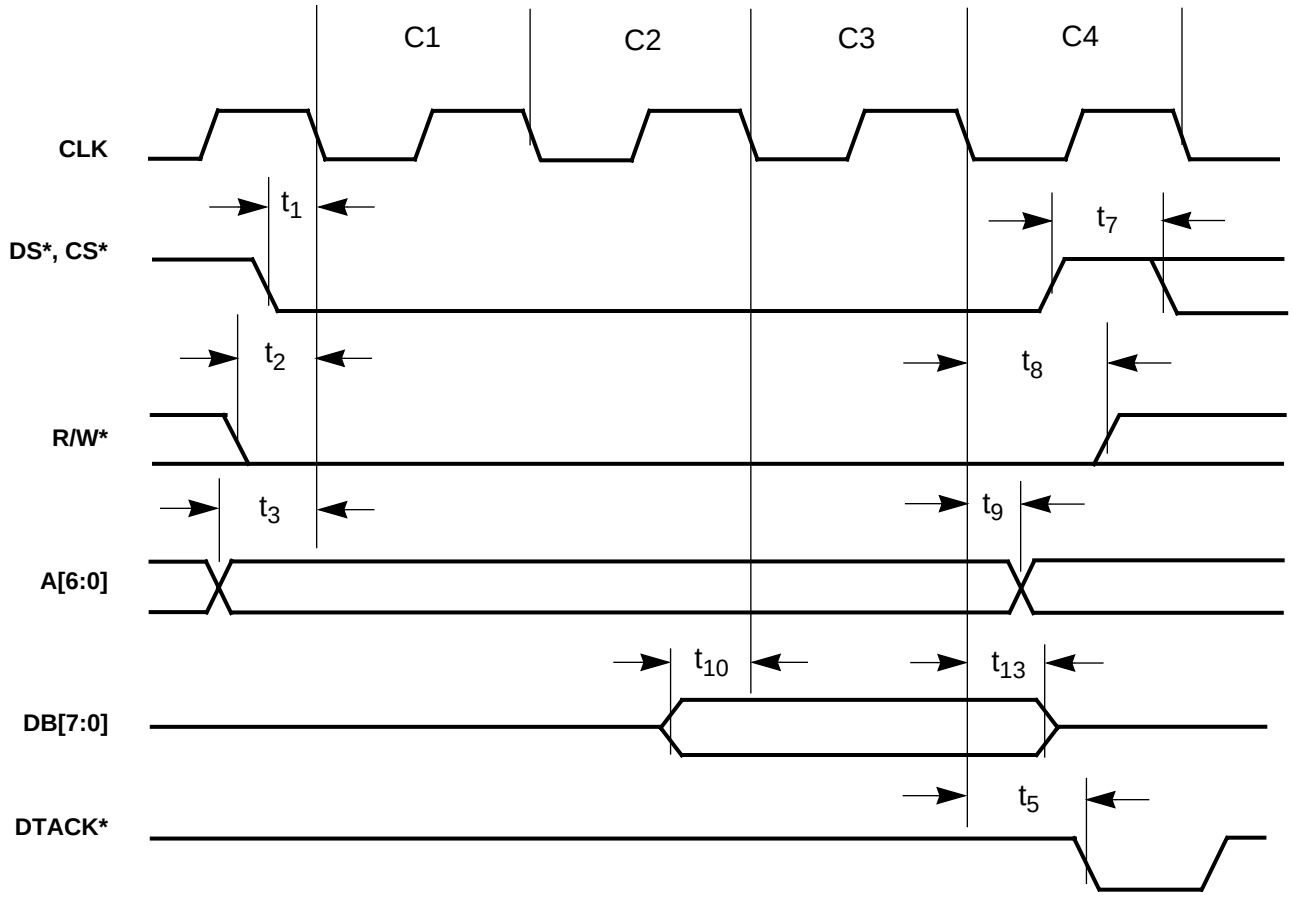


Figure 5–2c. Synchronous Service Acknowledge Cycle Timing

5.4.4 Parallel Port Timing Specifications

Refer to Figures 5–3a through 5–3c for identification of reference numbers in the following tables.

The following table shows the timing specifications for the parallel port when it is programmed in the transmit mode for either controller or peripheral.

Table 5–3a. Transmit Timing *(Figures 5–3a and 5–3c)*

Ref. #	Fig.	Parameter	MIN	MAX	Unit
t_p1	5–3a/5–3c	Setup time, PD[7:0] to STROBE* falling edge			Note 1
t_p2	5–3a/5–3c	STROBE* pulse width	0.64	40.8	µs (Note 2)
t_{p3}	5–3a/5–3c	Hold time, PD[7:0] after STROBE* rising edge			Note 3
t_p4	5–3a	ACK* pulse width	0.64	40.8	µs (Note 2)
t_p5	5–3a/5–3c	Hold time, PD[7:0] after BUSY falling edge	0.5	3.0	µs
t_p6	5–3c	STROBE* rising edge to next STROBE* falling edge			Note 4

NOTES: 1) Data setup time on the parallel data port is equal to the programmed value in the SWR Register plus 100 ns.

2) The width of the STROBE* and ACK* pulse is set by the programmed value in the SWR and AWR Registers. The numbers in the table are based on a 25-MHz operating frequency.

3) The data hold time on the parallel data port is equal to the basic data transfer cycle time minus the sum of t_p1 and t_p2; t_p4 = cycle - (t_p1 + t_p2). The basic cycle time is dependent upon the mode of operation (controller or peripheral) and the rate at which the receiver can process and acknowledge the data transfer.

4) This time value is dependent upon the length of time required by the receiver to process and acknowledge the data transfer. The CL-CD1190 will not begin the next cycle until the previous cycle has been completed by the activation of ACK* and deactivation of BUSY (if present).

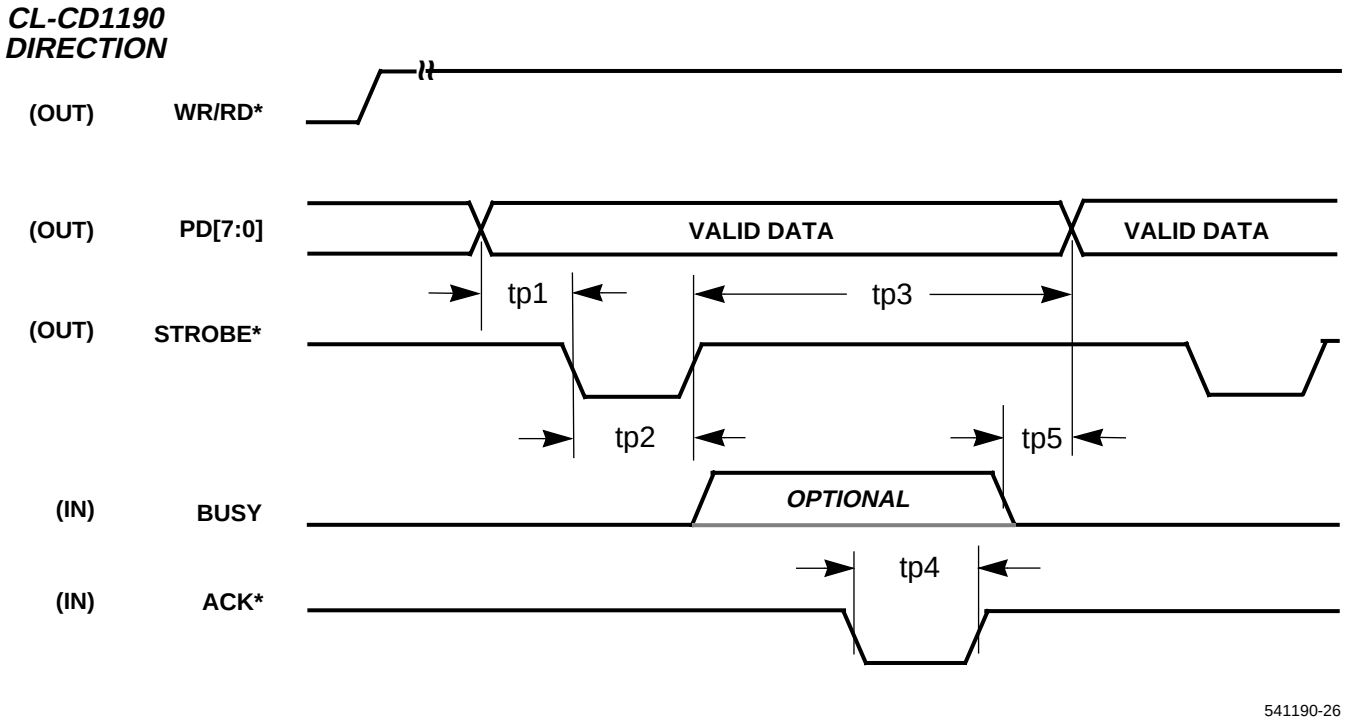
5.4.5 Parallel Port Timing Specifications *(cont.)*

The following table shows the timing specifications for the parallel port when it is programmed in the receive mode for either controller or peripheral.

Table 5–3b. Receive Timing *(Figures 5–3b and 5–3d)*

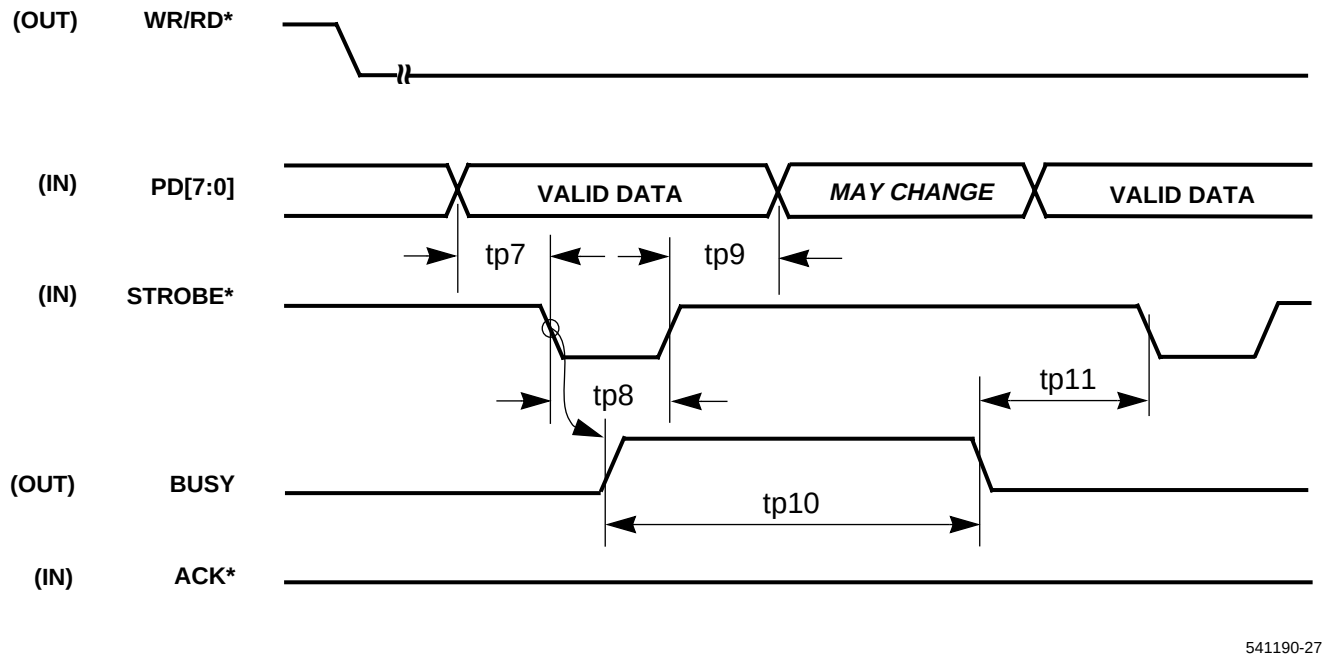
Ref. #	Fig.	Parameter	MIN	TYP	MAX	Unit
t_p7	5–3b/–3d	Setup time, PD[7:0] to STROBE* falling edge	200			ns
t_p8	5–3b/–3d	STROBE* pulse width	0.64		40.8	µs
t_p9	5–3b/–3d	Hold time, PD[7:0] after STROBE* rising edge	200			ns
t_p10	5–3b/–3d	BUSY pulse width	3.0	4.0		µs
t_p11	5–3b/–3d	BUSY falling edge to next STROBE* falling edge	0			ns
t_p12	5–3d	ACK* pulse width	0.64		40.8	µs (Note 1)

NOTE: The width of the STROBE* and ACK* pulse is set by the programmed value in the SWR and AWR Registers. The numbers in the table are based on a 25MHz operating frequency.



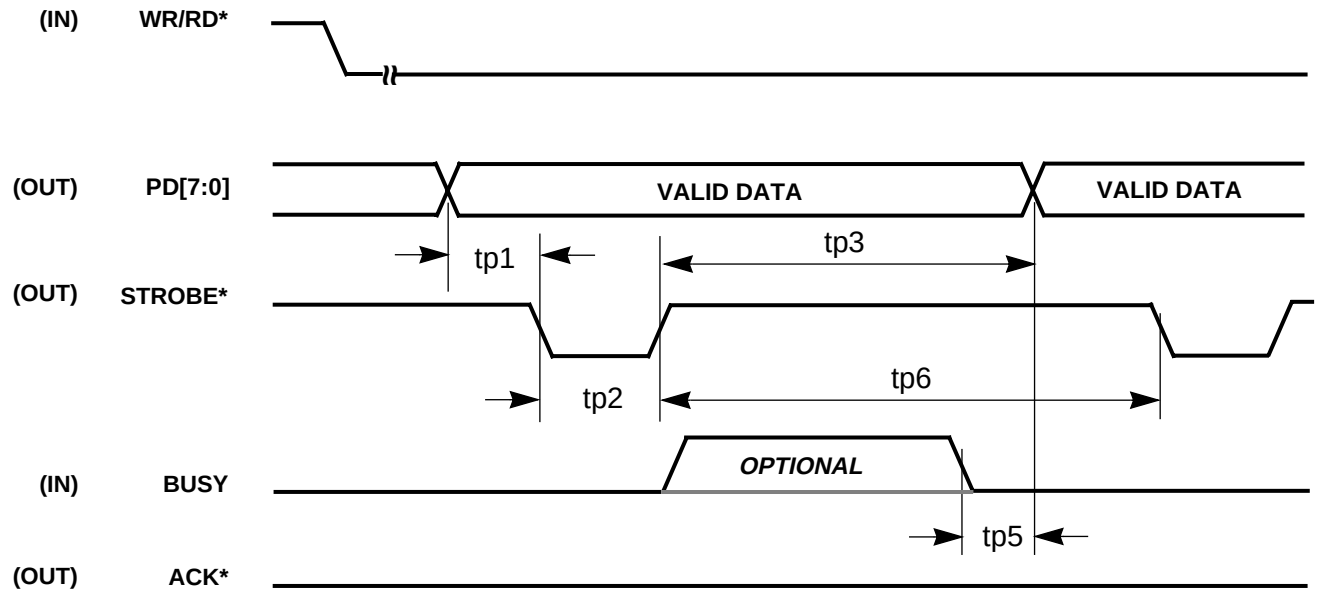
NOTE: The CL-CD1190 does not require that the receiver generate a BUSY. If present it will use it as a handshake signal; otherwise transfers happen continuously: setup, strobe, hold, setup, strobe, hold, etc.

Figure 5–3a. Parallel Port Transmit Timing (Controller Output Mode)



541190-27

Figure 5–3b. Parallel Port Receive Timing (Controller Input Mode)



541190-28

NOTE: The CL-CD1190 does not require that the receiver generate a BUSY. If present it will use it as a handshake signal; otherwise transfers happen continuously: setup, strobe, hold, setup, strobe, hold, etc.

Figure 5–3c. Parallel Port Transmit Timing (Peripheral Output Mode)

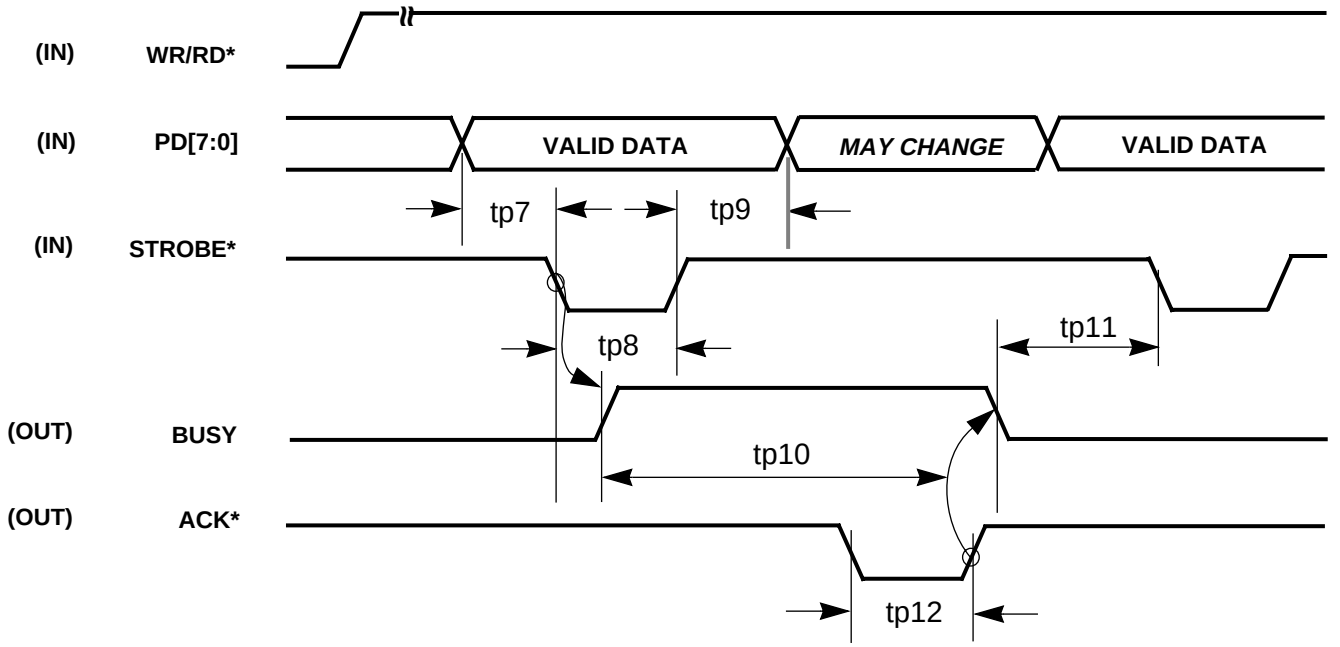
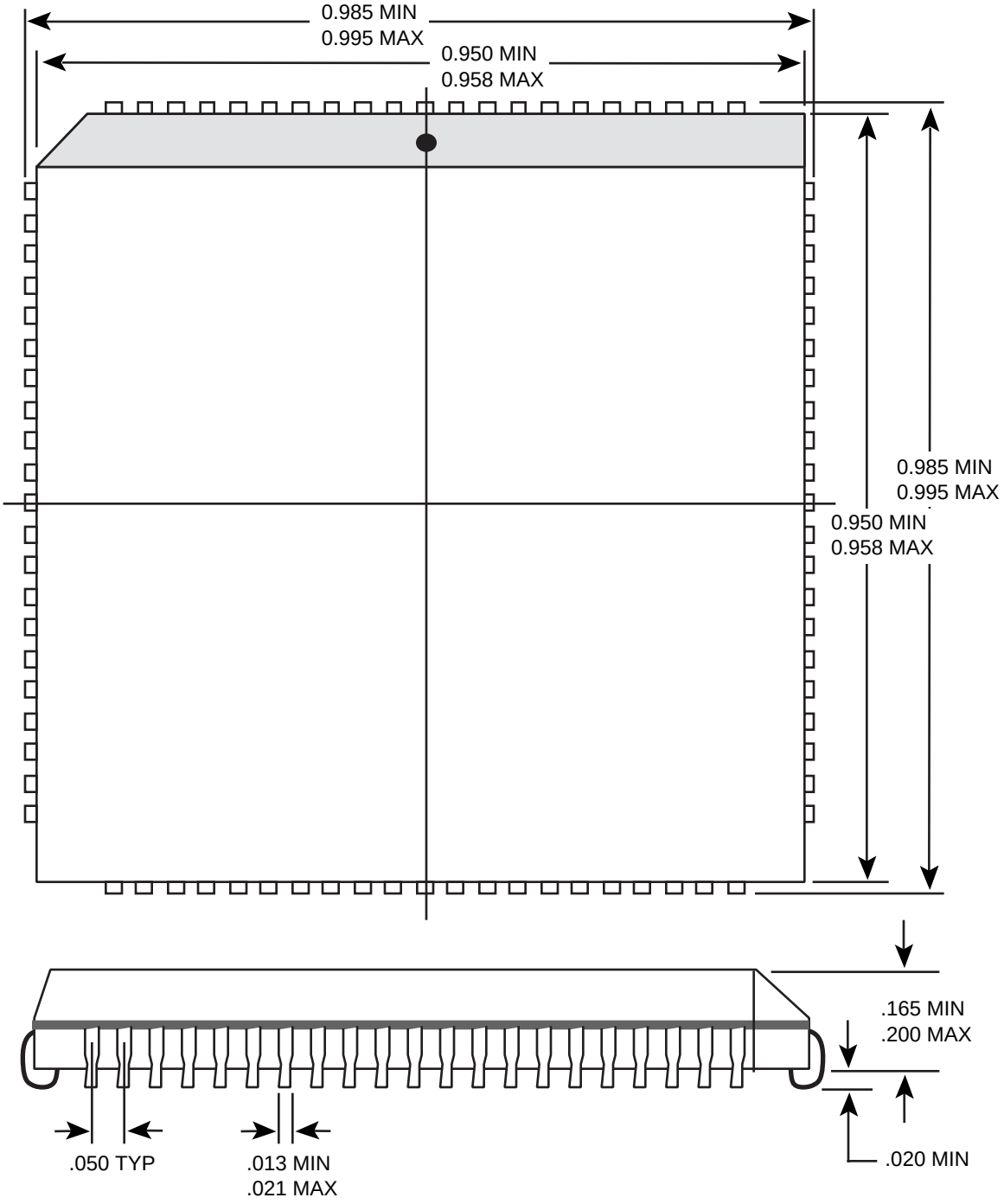


Figure 5–3d. Parallel Port Receive Timing (Peripheral Input Mode)

6. PACKAGE DIMENSIONS — 68-Pin PLCC

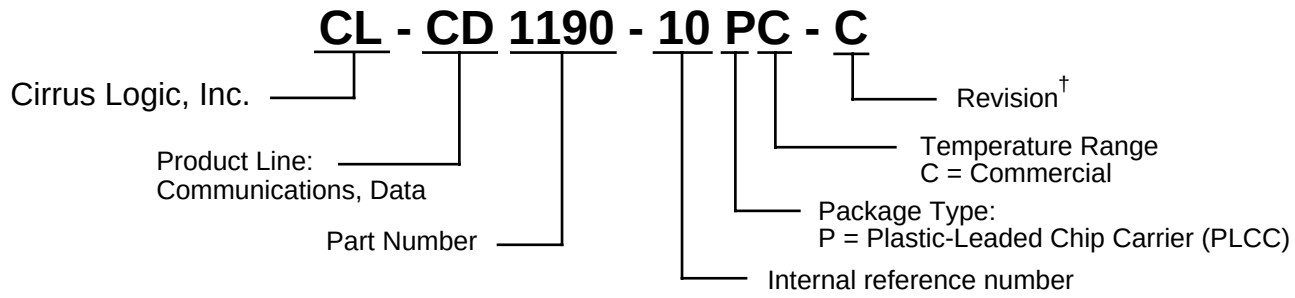


NOTE: All dimensions are in inches.

541190-30

7. ORDERING INFORMATION

When ordering the CL-CD1190, use the following format:

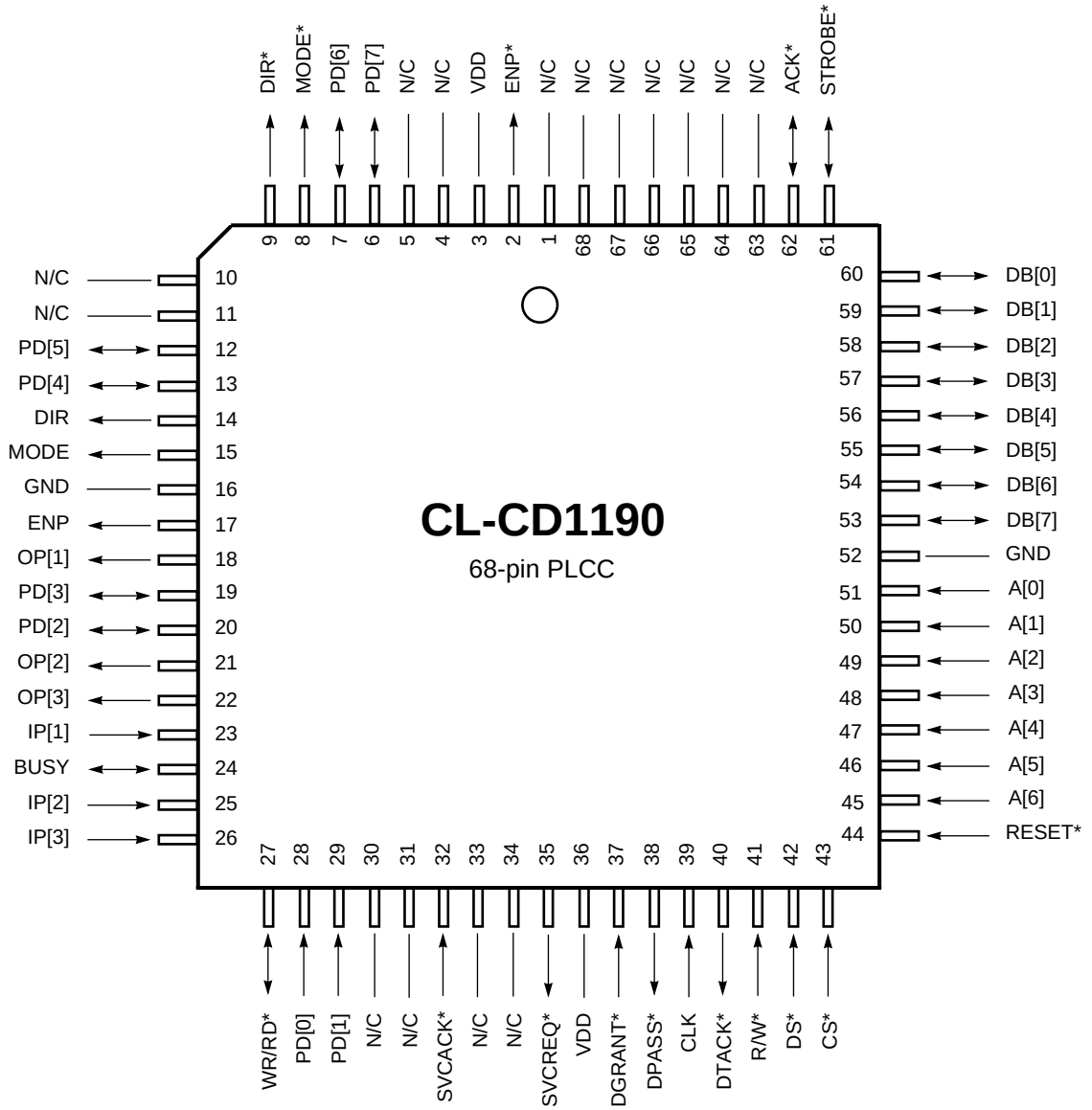


† Contact Cirrus Logic, Inc., for up-to-date information on revisions.

541190-31

8. QUICK REFERENCE

8.1 Pin Diagram — Top View



541190-1

NOTES: N/C means no connection.

To reduce address space, A[4] can be tied to ground, and A[6] and A[5] can be tied together; therefore, only 32 address locations are occupied.

8.2 CL-CD1190 Register Map

Symbol	Register Name	R/W	A[6:0]	(Hex)	(CA)[1]	Page[2]
IVR	Interrupt Vector Register	R/W	000 0000	00	00	45
ICR	Interrupt Configuration Register	R/W	000 0001	01	01	46
SWR	STROBE* Width Register	R/W	000 0010	02	02	43
AWR	ACK* Width Register	R/W	000 0011	03	03	43
TPR	Timer Prescale Register	R/W	000 0100	04	04	43
TMR	Timer Multiplier Register	R/W	000 0101	05	05	43
SR0	Specification Register ZEROes	R/W	000 0110	06	06	45
SR1	Specification Register ONEs	R/W	000 0111	07	07	45
FTR	FIFO Threshold Register	R/W	000 1000	08	08	47
DTR	Data Time-out Register	R/W	000 1001	09	09	44
GCR	Global Configuration Register	R/W	000 1010	0A	0A	36
CCR	Controller Command Register	R/W	000 1011	0B	0B	38
ISR	Interrupt Status Register	R/W	000 1100	0C	0C	46
SSR	Signal Status Register	R/W	000 1101	0D	0D	45
FCR	FIFO Count Register	R	000 1110	0E	0E	47
FRR	Firmware Revision Register	R/W	000 1111	0F	0F	36
ESR	End Of Service Request Register	W	110 0000	60	10	48
FDR	FIFO Data Register	R/W	110 0010	62	12	47
IRR	Interrupt Request Register	R	110 0111	67	17	48
PSR	Parallel Status Register	R	110 1100	6C	1C	44
SCR	Signal Control Register	R/W	110 1110	6E	1E	44

NOTES: 1) The column labeled CA denotes the compressed address. See Section 4 for a description of device addressing.

2) The page numbers shown in the table indicate the page where the detailed description of the register may be found.

8.3 Bit Definitions

Interrupt Vector Register (IVR) 00 R/W

Binary Value

Interrupt Configuration Register (ICR) 01 R/W

IEN	X	X	X	IUA	IDR	ISS	ITE

STROBE* Width Register (SWR) 02 R/W

Binary Value, Range 4 - 255

ACK* Width Register (AWR) 03 R/W

Binary Value, Range 4 - 255

Timer Prescale Register (TPR) 04 R/W

Binary Value, Range 0 - 255

Timer Multiplier Register (TMR) 05 R/W

Binary Value, Range 0 - 255

Specification Register ZEROes (SR0) 06 R/W

WR/RD*	IP3	IP2	IP1	0	0	0	0

Specification Register ONEs (SR1) 07 R/W

WR/RD*	IP3	IP2	IP1	0	0	0	0

FIFO Threshold Register (FTR) 08 R/W

Binary Threshold Value (Range: 0 - 192)

Data Timeout Register (DTR) 09 R/W

Binary Value, Range 0 - 255

Global Configuration Register (GCR) 0A R/W

VTYPE	FRSHR	0	ONLCR	OCRNL	NOACK	MODE	DIR

Controller Command Register (CCR) 0B R/W

CGC	PAR	SIG	TIM	EN/DS	C2	C1	C0

Format 1: Change Global Configuration Command

CGC	0	0	0	0	0	0	RESET

Format 2: Parallel Commands

0	PAR	X	X	X	DIP	FLUSH	ENP

Format 3: Signal Commands

0	0	SIG	X	X	S_BSY	ACK	C_BSY

Format 4: Timer Commands

0	X	X	TIM	EN/DS	X	X	X

Interrupt Status Register (ISR) 0C R/W

INT	0	0	0	UAI	DRI	SSI	TEI

Signal Status Register (SSR) 0D R/W

WR/RD*	IP3	IP2	IP1	0	0	0	0

FIFO Count Register (FCR) 0E R Only

Binary Count Value (Range 0 - 192)

Firmware Revision Register (FRR) 0F R/W

Firmware Revision Code

End of Service Request Register (ESR) 60 (10) W Only

X	X	X	X	X	X	X	X

FIFO Data Register (FDR) 62 (12) R/W

8-Bit Data to/from FIFO

Interrupt Request Register (IRR) 67 (17) R Only

0	0	0	0	0	0	EXT*	INT*

Parallel Status Register (PSR) 6C (1C) R Only

ENP	BUSY	ACK*	STROBE*	0	0	0	0

Signal Control Register (SCR) 6E (1E) R/W

Read Format

WR/RD*	IP3	IP2	IP1	WR/RD*	OP3	OP2	OP1

Write Format

X	X	X	X	WR/RD*	OP3	OP2	OP1

CIRRUS LOGIC

CL-CD1190

Errata

Version 1.2 **4 June 1992**

Revision B **CL-CD1190-10PC-B** **GFRCR value 43 Hex**

Known Bugs in the Revision B Device

DGRANT - DPASS Daisy Chain

Problem: In multiple chip designs, the DGRANT – DPASS daisy chain does not work properly when acknowledging service requests in hardware. An internal race condition causes the affected CD1190 to neither pass the acknowledge cycle nor keep it. This, in turn, can cause a system lockup condition because DTACK* will not be asserted by any device in the daisy-chain to indicate the end of the I/O cycle.

Workaround: Perform software acknowledgments as described in the polled mode operation section of the data sheet (pg. 15) if designing daisy-chained systems.

Corrected: Will be corrected in revision C - date TBD.

Sensitivity to Long Duration Input Strobe (STROBE* or ACK*) Pulses

Problem: The device is sensitive to ACK* and STROBE* input pulse length. In either case, the length of the strobe should not exceed 5.0µsec in duration at an operating frequency of 20MHz. At frequencies lower than the specified maximum, the length of the pulse that can be tolerated is longer and can be estimated by multiplying the operating clock cycle time by 100.

Workaround: There is no workaround other than to be sure no strobes (STROBE* or ACK*) exceed 5.0µsec in length or clock the device slower.

Corrected: Will be corrected in revision C - date TBD.

Signal Change Detect Non-Functional

Problem: The signal change detection function does not work. Once the parallel channel has been enabled, changes on the IP[1:3] inputs (either high-to-low or low-to-high) will not be detected automatically.

Workaround: In order to detect changes, host software must manually test the IP bits in the Signal Control Register (SCR) and compare a previously read value with the current value. The software can be aided in this process through the use of the on-chip timer; the timer can be programmed to produce a periodic interrupt that will cause the host to poll the SCR.

Corrected: Will be corrected in revision C - date TBD.

CIRRUS LOGIC

Data Sheet Errors

V_{IH} Min is 2.2V on STROBE* (Pin 61)

Problem: The data sheet specifies that V_{IH} for all pins other than CLK and RESET* is 2.0V. It should have indicated the V_{IH} for STROBE* is 2.2V. This is only relevant when STROBE* is an input (all parallel input modes).

Workaround: There is no workaround; this is a voltage specification based on the type of pad used in the design.

Corrected: Will be corrected in next printing of the data sheet- date: TBD.

Incorrect Specification of Performance

Problem: The data sheet specifies that the device can send/receive data at rates up to 250K bytes/second. This number was based on best case conditions in a simulation environment that assumed the CL-CD1190 was connected to a sender or receiver that was faster than the device itself. Actual maximum performance in applications has been shown to be 175K bytes/second depending on the mode of operation (peripheral/controller and input/output). The best performance has been shown to occur with peripheral input mode for receive and controller output with the *No Acknowledge* mode enabled for transmit.

Workaround: Not Applicable.

Corrected: Not Applicable.

Direct Sales Offices

Domestic

N. CALIFORNIA
Fremont
TEL: 510/623-8300
FAX: 510/252-6020

S. CALIFORNIA
Westlake Village
TEL: 805/371-5860
FAX: 805/371-5861

NORTHWESTERN AREA
Portland, OR
TEL: 503/620-5547
FAX: 503/620-5665

SOUTH CENTRAL AREA
Austin, TX
TEL: 512/255-0080
FAX: 512/255-0733

Irving, TX
TEL: 972/252-6698
FAX: 972/252-5681

Houston, TX
TEL: 281/257-2525
FAX: 281/257-2555

NORTHEASTERN AREA
Andover, MA
TEL: 508/474-9300
FAX: 508/474-9149

SOUTHEASTERN AREA
Raleigh, NC
TEL: 919/859-5210
FAX: 919/859-5334

Boca Raton, FL
TEL: 561/241-2364
FAX: 561/241-7990

International

CHINA
Beijing
TEL: 86/10-6428-0783
FAX: 86/10-6428-0786

FRANCE
Paris
TEL: 33/1-48-12-2812
FAX: 33/1-48-12-2810

GERMANY
Herrsching
TEL: 49/81-52-92460
FAX: 49/81-52-924699

HONG KONG
Tsimshatsui
TEL: 852/2376-0801
FAX: 852/2375-1202

ITALY
Milan
TEL: 39/2-3360-5458
FAX: 39/2-3360-5426

JAPAN
Tokyo
TEL: 81/3-3340-9111
FAX: 81/3-3340-9120

KOREA
Seoul
TEL: 82/2-565-8561
FAX: 82/2-565-8565

SINGAPORE
TEL: 65/743-4111
FAX: 65/742-4111

TAIWAN
Taipei
TEL: 886/2-718-4533
FAX: 886/2-718-4526

UNITED KINGDOM
London, England
TEL: 44/1727-872424
FAX: 44/1727-875919

High-Value ‘Systems in Silicon’

Headquartered in Fremont, California, Cirrus Logic is a leading manufacturer of advanced integrated circuits for the personal computer, consumer, and industrial markets. The Company’s software-rich ‘systems in silicon’ add high value to major brands worldwide in applications that span multimedia (graphics, audio, video), communications (enterprise networking and remote data access), and mass storage (magnetic and optical moving media).

With a focus on innovative microperipheral chip solutions, Cirrus Logic is committed to technology leadership in the Interactive Age.

Cirrus Logic’s manufacturing strategy ensures maximum product quality and availability, as well as access to world-class processing technologies through joint ventures with IBM® and Lucent Technologies®.

Contact one of our systems and applications specialists to see how your company can benefit from the high value that Cirrus Logic adds to its customers’ products.

Cirrus Logic, Inc.
3100 West Warren Ave., Fremont, CA 94538
TEL: 510/623-8300 FAX: 510/252-6020

Publications Ordering: 800/359-6414 (USA) or 510/249-4200
World Wide Web: http://www.cirrus.com
551190-002