- **Dual TL16C550C Universal Asynchronous Receiver/Transmitters (UARTs)**
- **IEEE 1284 Bidirectional Parallel Data (PD) Port**
  - **– Compatible With Standard Centronics Parallel Interface**
  - **– Support for Parallel Protocols: Extended Capability Port (ECP) and Enhanced Parallel Port (EPP)**
  - **– Data Path 16-Byte FIFO Buffer**
  - **– Direct Memory Access (DMA) Transfer**
  - **– Decompression of Run Length Encoded Data in ECP Reverse Mode**
  - **– Direct Connection to Printer, No External Transceiver is Needed**
- **Serial Ports Have Infrared Data Association (IrDA) Inputs and Outputs**
  - **– 1200 bps to 115.2 kbps Data Rate**
- **16-Byte FIFOs Reduce CPU Interrupts**
- **12 mA Drive Current for All 1284 Control Terminals and Parallel Port Data Terminals**
- **Programmable Auto Flow Control on the UARTs**
- **Capable of Running With All Existing TL16C450 Software**
- **After Reset, All Registers Are Identical to the TL16C450 Register Set**
- **Up to 16-MHz Clock Rate for Up to 1-Mbaud Operation**
- **In the TL16C450 Mode, Hold and Shift Registers Eliminate the Need for Precise Synchronization Between the CPU and Serial Data**
- **Programmable Baud-Rate Generator Allows Division of Any Input Reference Clock by 1 to (2^{16} – 1) and Generates an Internal 16× Clock**
- **Adds or Deletes Standard Asynchronous Communication Bits (Start, Stop, and Parity) to or From the Serial Data Stream**
- **On-Board Prescaler With Programmable Divisor Values From 0 to 33**
- **Independent Control of Transmit, Receive, Line Status, and Data-Set Interrupts on Each Channel**
- **Fully Programmable Serial-Interface Characteristics:**
  - **– 5-, 6-, 7-, or 8-Bit Characters**
  - **– Even-, Odd-, or No-Parity Bit Generation and Detection**
  - **– 1-, 1 1/2-, or 2-Stop Bit Generation**
  - **– Baud Generation (DC to 1 Mbit Per Second)**
- **False Start-Bit Detection**
- **Complete Status Reporting Capabilities**
- **3-State Output TTL Drive Capabilities for Bidirectional Data Bus and Control Bus**
- **Line Break Generation and Detection**
- **Internal Diagnostic Capabilities:**
  - **– Loopback Controls for Communications Link-Fault Isolation**
  - **– Break, Parity, Overrun, and Framing Error Simulation**
- **Fully Prioritized Interrupt System Controls**
- **Modem-Control Functions ($\overline{\text{CTS}}$, $\overline{\text{RTS}}$, $\overline{\text{DSR}}$, $\overline{\text{DTR}}$, $\overline{\text{RI}}$, and $\overline{\text{DCD}}$)**
- **Available in 80-Pin Quad Flatpack (QFP) Package**

description

The TL16PIR552 has a dual-channel universal asynchronous receiver/transmitter (UART). The UART is similar to the TL16C550C. The device serves two serial input/output ports simultaneously in microcomputer or microprocessor-based systems. Each channel performs serial-to-parallel conversion on data characters received from peripheral devices or modems and parallel-to-serial conversion on data characters transmitted by the CPU. The complete status of each channel of the dual UART can be read by the CPU at any time during functional operation. The information obtained includes the type and condition of the transfer operation being performed and the error condition.

description (continued)

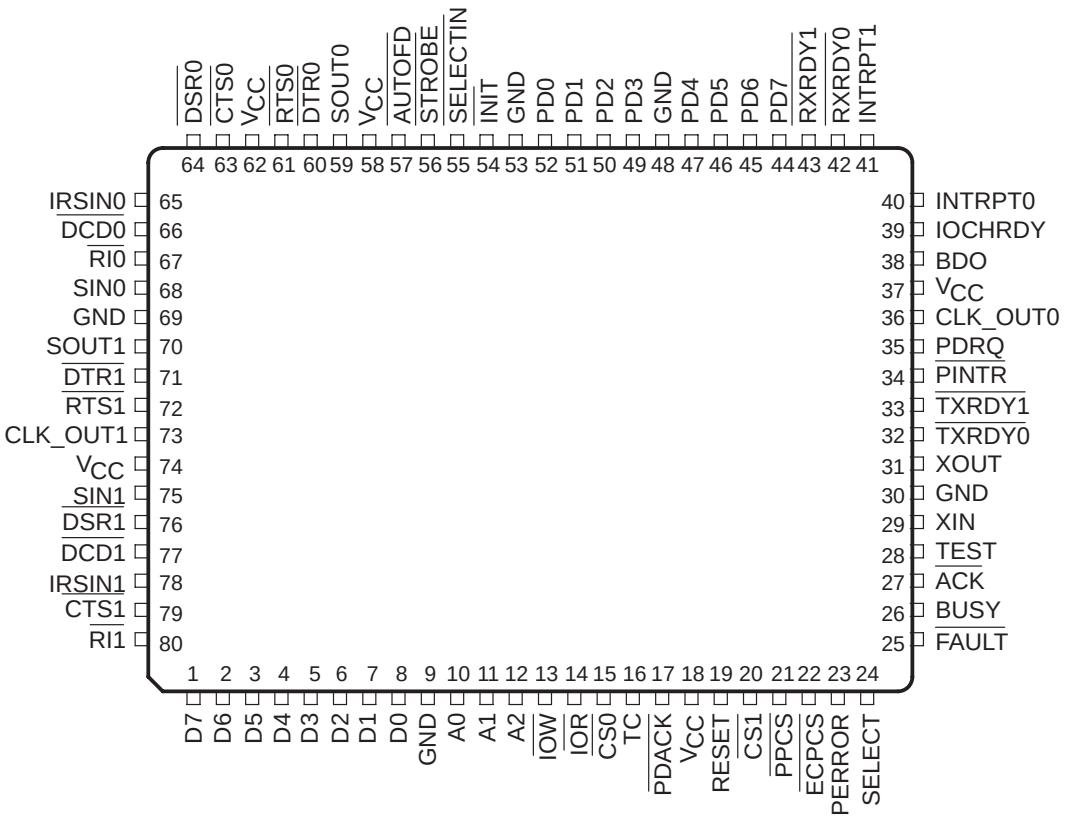
The receiver and transmitter FIFOs in the UARTs store up to 16 bytes including three additional bits of error status per byte for the receiver FIFO. In the FIFO mode, there is a selectable autoflow control feature that can significantly reduce software overload and increase system efficiency by automatically controlling serial data flow through $\overline{RTS}$ output and $\overline{CTS}$ input signals.

The TL16PIR552 UART includes a programmable baud-rate generator capable of dividing a reference clock by divisors from 1 to 65535 and producing a 16× reference clock for the internal transmitter logic. Provisions are also included to use this 16× clock for the receiver logic. The UART accommodates a 1-Mbaud serial rate (16-MHz input clock) so that a bit time is 1 μs and a typical character time is 10 μs (start bit, eight data bits, stop bit).

Each serial channel has a prescaler with programmable divisor values from 0 to 33. The serial ports also have a dedicated infrared serial data input (IRSIN0/1) and the serial data outputs multiplex between a RS-232-type serial output or an infrared serial data output. This is selected through an internal register bit and uses the same SOUT0/1 output terminals. The same UART circuit is used for the data path for the IrDA or the RS-232 operations. Channel 0 is powered up at IR0 and channel 1 is powered up during the RS-232 mode.

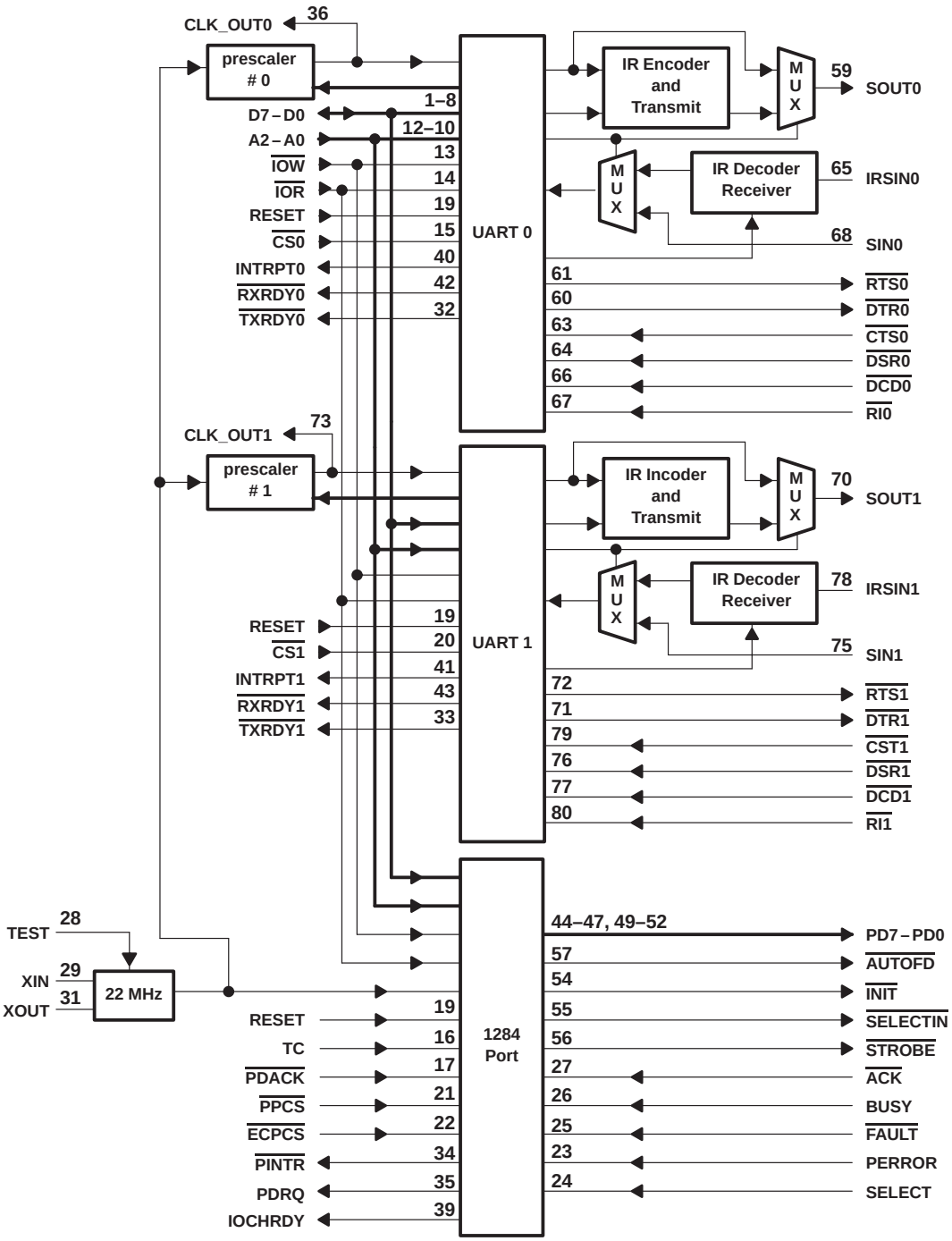
In addition to dual communication capabilities, the TL16PIR552 provides the user with an IEEE 1284 host side compatible, bidirectional, parallel data port. The parallel port operates in a compatible, FIFO, extended capability port (ECP) with RLE data decompression mode, and in a enhanced parallel port (EPP) mode. The default mode of operation is compatible with the Centronics printer port. The parallel port and the two serial ports provide IBM PC/AT™-compatible computers with a single device to serve a 3-port system.

PH PACKAGE
(TOP VIEW)



IBM and PC/AT are trademarks of International Business Machines Corporation.

functional block diagram



NOTE A: Terminal numbers shown are for the PH package.

Terminal Functions

TERMINAL NAME	TERMINAL NO.	I/O	DESCRIPTION
A0–A2	10–12	I	Register select. A0–A2 are address lines that select the internal registers in the device.
$\overline{\text{ACK}}$	27	I	Data acknowledge. In compatibility mode $\overline{\text{ACK}}$ is pulled low by the peripheral device to acknowledge transfer of a data byte from the host. In ECP mode, $\overline{\text{ACK}}$ is used in a closed loop handshake with the host AUTOFD to transfer data from the peripheral device to the host. It is asserted low by the peripheral device to indicate data is available. In EPP mode, $\overline{\text{ACK}}$ is used by the peripheral device to interrupt the host. This signal is active high and is positive-edge triggered.
$\overline{\text{AUTOFD}}$	57	O	Autofeed. In compatibility mode $\overline{\text{AUTOFD}}$ is set low in conjunction with $\overline{\text{SELECTIN}}$ being set high to request a 1284 mode. Then $\overline{\text{AUTOFD}}$ is set high after the peripheral device acknowledges the signal by setting $\overline{\text{ACK}}$ low. In EPP mode, $\overline{\text{AUTOFD}}$ is an active low output that is used to denote data read or write operations. It also provides a ninth data bit that is used to determine whether address or data information is present on the data lines in the forward mode. In EPP mode this signal is active low to denote data read or write operations. In ECP mode, $\overline{\text{AUTOFD}}$ requests a byte of data from the peripheral when asserted, handshaking with $\overline{\text{ACK}}$ in the reverse direction. In the forward direction $\overline{\text{AUTOFD}}$ indicates whether the data lines contain the ECP address or data. The host drives this signal to flow control in the reverse direction. It is an “interlocked” handshake with $\overline{\text{ACK}}$. $\overline{\text{AUTOFD}}$ also provides command information in the forward phase.
BDO	38	O	Bus buffer output. BDO output is active (high) when the CPU is not reading data. It controls the system bus driver.
BUSY	26	I	Busy. In compatibility mode BUSY is driven high to indicate that the peripheral is not ready to receive data. In the ECP mode, BUSY is driven high to indicate that the peripheral is not ready to receive data and is driven low to indicate that the peripheral is ready to receive data in forward mode. In reverse mode, BUSY is low when the information on the data lines are commands (RLE) and it is high when the information on the data lines is data. In EPP mode, BUSY is active low. It is driven inactive as a positive acknowledgment from the peripheral device that data or address information is completed. It is active when the peripheral is ready for the next data and address transfer. In ECP mode, BUSY deasserts to indicate that the peripheral can accept data. It handshakes with $\overline{\text{STROBE}}$ in the forward direction. In the reverse direction BUSY indicates whether the data lines contain the ECP command information or data. The peripheral uses this signal to control flow in the forward direction. It is an “interlocked” handshake with $\overline{\text{STROBE}}$. BUSY also provides command information in the reverse direction.
CLK_OUT0, CLK_OUT1	36, 73	O	Prescaler Outputs. CLK_OUT0 and CLK_OUT1 drive the UARTs.
$\overline{\text{CS0}}$, $\overline{\text{CS1}}$	15,20	I	Chip select. $\overline{\text{CS0}}$ and $\overline{\text{CS1}}$ are active low inputs that act as an enable for the write operation and a read operation for the UART. $\overline{\text{CS0}}$ enables UART0 and $\overline{\text{CS1}}$ enables UART1.
$\overline{\text{CTS0}}$, $\overline{\text{CTS1}}$	63, 79	I	Clear to send. $\overline{\text{CTS0}}$ and $\overline{\text{CTS1}}$ are modem-status signals whose condition can be verified by reading bit 4 (CTS) of the MSR. Bit 0 (ΔCTS) of the MSR indicates that $\overline{\text{CTS0}}$ or $\overline{\text{CTS1}}$ has changed states since the last read operation from the MSR. When the modem-status interrupt is enabled, $\overline{\text{CTS0}}$ or $\overline{\text{CTS1}}$ changes states, and an interrupt is generated. $\overline{\text{CTS0}}$ or $\overline{\text{CTS1}}$ is also used in the auto-$\overline{\text{CTS}}$ mode to control the transmitter.
D7–D0	1–8	I/O	Data bus. Eight data lines with 3-state outputs provide a bidirectional path for data, control, and status information between the CPU and the device.
$\overline{\text{DCD0}}$, DCD1	66, 77	I	Data carrier detect. $\overline{\text{DCD0}}$ and $\overline{\text{DCD1}}$ are modem status signals whose condition can be verified by reading bit 7 (DCD) of the modem status register (MSR). Bit 3 (ΔDCD) of the MSR indicates that $\overline{\text{DCD0}}$ or $\overline{\text{DCD1}}$ has changed state since the last read from MSR. If the modem status interrupt is enabled when $\overline{\text{DCD0}}$ or $\overline{\text{DCD1}}$ changes state, an interrupt is generated.
$\overline{\text{DSR0}}$, DSR1	64, 76	I	Data set ready. $\overline{\text{DSR0}}$ and $\overline{\text{DSR1}}$ are modem status signals whose condition can be verified by reading bit 5 (DSR) of the MSR. Bit 1 (ΔDSR) of the MSR indicates that $\overline{\text{DSR0}}$ or $\overline{\text{DSR1}}$ has changed state since the last read from MSR. If the modem status interrupt is enabled when $\overline{\text{DSR0}}$ or $\overline{\text{DSR1}}$ changes state, an interrupt is generated.
$\overline{\text{DTR0}}$, DTR1	60, 71	O	Data terminal ready. When active, (low), $\overline{\text{DTR0}}$ or $\overline{\text{DTR1}}$ informs a modem or data set that the UART is ready to establish communication. $\overline{\text{DTR0}}$ or $\overline{\text{DTR1}}$ is placed in the active state by setting bit 0 of the modem-control register (MCR) to 1. $\overline{\text{DTR}}$x is placed in the inactive state either as a result of a master reset, during loop-mode operation, or resetting bit 0 of the MCR.
$\overline{\text{ECPCS}}$	22	I	Chip select. $\overline{\text{ECPCS}}$ is used for the ECP parallel port internal registers, and is an active low signal.

NOTE 1: All parallel port control outputs are open drain outputs in the Centronics mode, and push-pull outputs in other modes.

Terminal Functions

TERMINAL NAME	TERMINAL NO.	I/O	DESCRIPTION
$\overline{\text{FAULT}}$	25	I	Fault indication. In compatibility mode $\overline{\text{FAULT}}$ is set high to acknowledge the 1284 mode requested. The EPP mode is user defined. In ECP mode $\overline{\text{FAULT}}$ generates an error interrupt when asserted. It provides a mechanism for peer-to-peer communication. This signal is valid only in the forward direction. During ECP mode the peripheral is permitted (but not required) to drive this terminal low to request a reverse transfer. The request is merely a “hit” to the host; the host has ultimate control over the transfer direction. $\overline{\text{FAULT}}$ is typically used to generate an interrupt to the host CPU.
GND	9, 30, 48, 53, 69		Ground terminal.
$\overline{\text{INIT}}$	54	O	Initiation. In compatibility mode $\overline{\text{INIT}}$ is pulsed low to reset the interface and force a return to the compatibility mode idle phase.In ECP mode $\overline{\text{INIT}}$ is driven low to place the channel in the reverse direction and it allows the peripheral to drive the bidirectional data lines when $\overline{\text{SELECTIN}}$ is high. In EPP mode $\overline{\text{INIT}}$ is active low. When driven low, this signal initiates a termination cycle that results in the interface returning to the compatibility mode.
INTRPT0, INTRPT1	40,41	O	Interrupt (0–1). When active (high), INTRPT0 or INTRPT1 informs the CPU that the UART has an interrupt to be serviced. Four conditions that cause an interrupt to be issued include a receiver error, received data is available, an empty transmitter holding register, or an enabled modem-status interrupt.
IOCHRDY	39	O	ISA channel ready. IOCHRDY is an open drain output that extends the length of a bus cycle when it is inactive.
$\overline{\text{IOR}}$	14	I	Read input. $\overline{\text{IOR}}$ is an active low input signal that enables the selected channel to output data to D7–D0. The data output depends upon the register selected by the address A2–A0 inputs and chip select.
$\overline{\text{IOW}}$	13	I	Write input. $\overline{\text{IOW}}$ is an active low input signal that enables the data to be input to either a UART or to the parallel port. The data destination depends upon the register selected by the address inputs A2–A0 and chip select.
IRSIN0, IRSIN1	65, 78	I	Serial data. IRSIN0 and IRSIN1 are serial inputs from an IR serial data communication device.
PD0–PD7	52–49, 47–44	I/O	Parallel data bits (0–7). PD0–PD7 provide a byte wide input or 47–44 output port to the system. These bits contain address, data, or RLE command data.
$\overline{\text{PDACK}}$	17	I	Parallel port DMA acknowledge. $\overline{\text{PDACK}}$ is an active low input.
PDRQ	35	O	DMA Request. PDRQ is used for parallel port DMA requests during ECP and FIFO modes.
PERROR	23	I	Peripheral error. In compatibility mode PERROR is driven high when the device encounters an error in the paper path. In ECP mode the peripheral drives PERROR low to acknowledge a reverse request ($\overline{\text{INIT}}$). Based on this signal the host determines when it is permitted to drive the data bus. In EPP mode the signal is user defined.
$\overline{\text{PINTR}}$	34	O	Parallel port interrupt. $\overline{\text{PINTR}}$ is a 3-state output. In EPP mode this is an active high, positive-edge triggered input.
$\overline{\text{PPCS}}$	21	I	Chip select. $\overline{\text{PPCS}}$ is used for the parallel port internal registers and is an active-low signal.
$\overline{\text{RI0}}$, RI1	67,80	I	Ring Indicator. $\overline{\text{RI0}}$ and $\overline{\text{RI1}}$ are modem-status signals whose condition can be verified by reading bit 6 (RI) of the MSR. Bit 2 (TERI) of the MSR indicates that the RI0/RI1 input has transitioned from a low to a high level since the last read operation from MSR. If the modem-status interrupt is enabled when this transition occurs, an interrupt is generated.
RESET	19	I	Reset. RESET is an active high reset that when asserted, clears all UARTs and parallel port printer internal registers.
$\overline{\text{RTS0}}$, RTS1	61 72	O	Request to send. When active, $\overline{\text{RTS0}}$ or $\overline{\text{RTS1}}$ informs the modem or data set that the UART is ready to receive data. $\overline{\text{RTS0}}$ or $\overline{\text{RTS1}}$ is set to its active level by setting the $\overline{\text{RTSx}}$ modem-control register bit and is set to inactive (high) either as a result of master reset or during loop-mode operations or by resetting bit 1 (RTS) of the MCR. In the auto-$\overline{\text{RTS}}$ mode, $\overline{\text{RTSx}}$ is set to its inactive level by the receiver threshold-control logic.

NOTE 1: All parallel port control outputs are open drain outputs in the Centronics mode, and push-pull outputs in other modes.

Terminal Functions

TERMINAL NAME	TERMINAL NO.	I/O	DESCRIPTION
RXRDY0, RXRDY1	42,43	O	Receiver ready. Receiver direct-memory access (DMA) signaling is available with RXRDY0 or RXRDY1. When operating in the FIFO mode, one of two types of DMA signalling can be selected using the FIFO control-register bit 3 (FCR3). When operating in the TL16C450 mode, only DMA mode 0 is allowed. Mode 0 supports single-transfer DMA in which a transfer is made between CPU bus cycles. Mode 1 supports multitransfer DMA in which multiple transfers are made continuously until the receiver FIFO has been emptied. In DMA mode 0 (FCR0 = 0 or FCR0 = 1, FCR3 = 0), if there is at least one character in the receiver FIFO or receiver holding register, RXRDY0 and RXRDY1 are active (low). When RXRDYx has been active but there are no characters in the FIFO or holding register, RXRDYx goes inactive (high). In DMA mode 1 (FCR0 = 1, FCR3 = 1), when the trigger level or the timeout has been reached, RXRDYx goes active (low); when it has been active but there are no more characters in the FIFO or holding register, it goes inactive (high).
SELECT	24	I	Select. In compatibility mode SELECT is set high to indicate that the printer is on line. In ECP mode SELECT indicates an affirmative response for each extensibility byte. It is high when the requested mode is supported. In EPP mode the signal is user defined.
SELECTIN	55	O	Select. In compatibility mode SELECTIN is set low by the host to select the peripheral device. It is set high to request the 1284 mode. In ECP mode SELECTIN is driven high by the host. It is driven low by the host to terminate the ECP mode and return to the compatibility mode. In EPP mode SELECTIN is an active low output that is used to denote address read or write operations.
SIN0, SIN1	68,75	I	Serial data. SIN0 and SIN1 are inputs from a connected communication device.
SOUT0, SOUT1	59, 70	O	Serial outputs. Either IR output format or UART output format. Composite serial data outputs are to be connected to a communication device. SOUT0 and SOUT1 are set to the marking state (1) as a result of a master reset operation.
STROBE	56	O	Data strobe. In compatibility mode STROBE is set active low to transfer data into the input latch of the peripheral device. Data is valid while STROBE is low. In ECP mode STROBE is used in a closed-loop handshake with BUSY to transfer data or address information from the host to the peripheral device. In EPP mode this signal is set low to denote an address or data write operation to the peripheral and is set high to denote and address or data read operation from the peripheral.
TC	16	I	Terminal count. TC is an active high input during DMA and when PDACK is low. TC indicates that the data transfer is complete.
TEST	28	I	Test. TEST is tied low during normal operation. To turn the oscillator off and measure ICCQ current, TEST is tied active (high).
TXRDY0, TXRDY1	32,33	O	Transmitter ready. Transmitter DMA signaling is available with TXRDY0 and TXRDY1. When operating in the FIFO mode, one of two types of DMA signalling can be selected through FCR3. When operating in the TL16C450 mode, only DMA mode 0 is allowed. Mode 0 supports single-transfer DMA in which a transfer is made between the CPU bus cycles. Mode 1 supports multitransfer DMA in which multiple DMA transfers are made continuously until the transmit FIFO has been filled.
V_{CC}	18, 37, 58 62, 74		5-V supply voltage.
XIN XOUT	29 31	I/O	Crystal input and output terminals. A 22-MHz clock is required to meet the internal timing required by the 1284 parallel port (minimum 40 to 60% duty cycle).

NOTE 1: All parallel port control outputs are open drain outputs in the Centronics mode, and push-pull outputs in other modes.

detailed description

autoflow control

Autoflow control is comprised of auto-CTS and auto-RTS. With auto-CTS, the CTSx input must be active before the transmitter FIFO can emit data (see Figure 1). With auto-RTS, RTSx becomes active when the receiver needs more data and notifies the sending serial device (see Figure 1). When RTSx is connected to CTSx, data transmission does not occur unless the receiver FIFO has space for the data; thus, overrun errors are eliminated when UART1 and UART2 are TL16PIR552s with enabled autoflow control. If not, overrun errors occur when the transmit-data rate exceeds the receiver FIFO read latency.

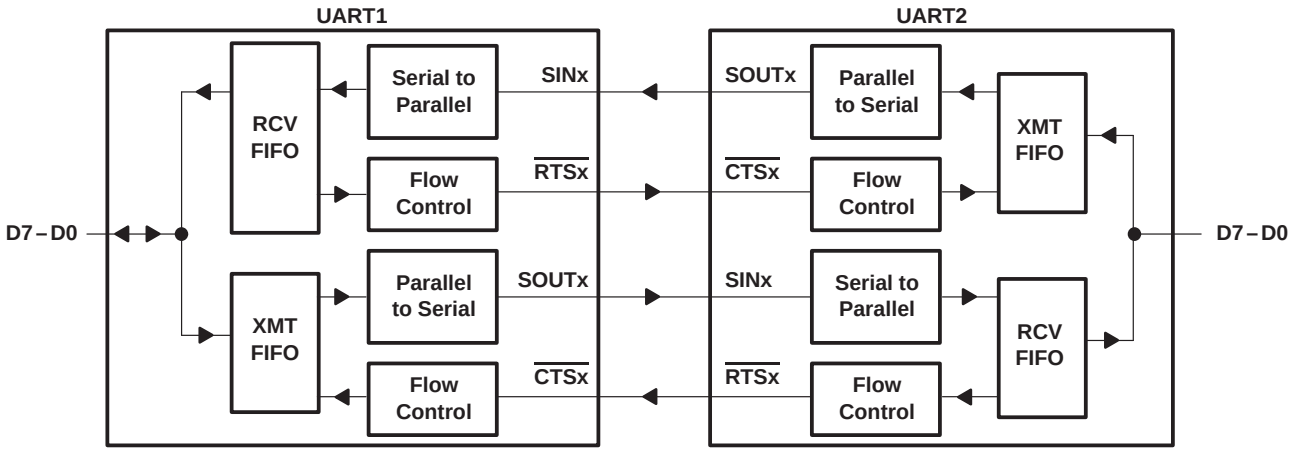


Figure 1. Autoflow Control (Auto-$\overline{\text{RTS}}$ and Auto-$\overline{\text{CTS}}$) Example

auto-$\overline{\text{RTS}}$ (see Figure 1)

Auto-$\overline{\text{RTS}}$ data-flow control originates in the receiver timing and control block (see functional block diagram) and is linked to the programmed receiver-FIFO trigger level. When the receiver-FIFO level reaches a trigger level of 1, 4, or 8 (see Figure 3), $\overline{\text{RTSx}}$ is deasserted. With trigger levels of 1, 4, and 8, the sending UART may send an additional byte after the trigger level is reached (assuming the sending UART has another byte to send) because it may not recognize the deassertion of $\overline{\text{RTSx}}$ until after it has begun sending the additional byte. $\overline{\text{RTSx}}$ is automatically reasserted once the receiver (RCV) FIFO is emptied by reading the receiver buffer register(RBR).

If the trigger level is 14 (see Figure 6), $\overline{\text{RTSx}}$ is deasserted after the first data bit of the sixteenth character is present on the SINx line. $\overline{\text{RTSx}}$ is reasserted when the RCV FIFO has at least one available byte space.

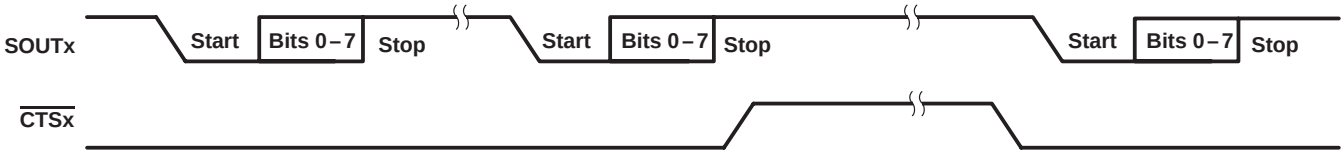
auto-$\overline{\text{CTS}}$ (see Figure 1)

The transmitter circuitry checks $\overline{\text{CTSx}}$ before sending the next data byte. When $\overline{\text{CTSx}}$ is active, it sends the next byte. To stop the transmitter from sending the following byte, $\overline{\text{CTSx}}$ must be released before the middle of the last stop bit that is currently being sent (see Figure 2). The auto-$\overline{\text{CTS}}$ function reduces interrupts to the host system. When flow control is enabled, the $\overline{\text{CTSx}}$ level changes do not trigger host interrupts because the device automatically controls its own transmitter. Without auto-$\overline{\text{CTS}}$, the transmitter sends any data present in the transmit FIFO and a receiver overrun error may result.

enabling autoflow control and auto-$\overline{\text{CTS}}$

Autoflow control is enabled by setting modem-control register (MCR) bit 5 (autoflow enable or AFE) and bit 1 (RTS) to 1. Autoflow incorporates both auto-$\overline{\text{RTS}}$ and auto-$\overline{\text{CTS}}$. When only auto-$\overline{\text{CTS}}$ is desired, bit 1 in the MCR should be reset to 0 (this assumes a control signal is driving $\overline{\text{CTSx}}$).

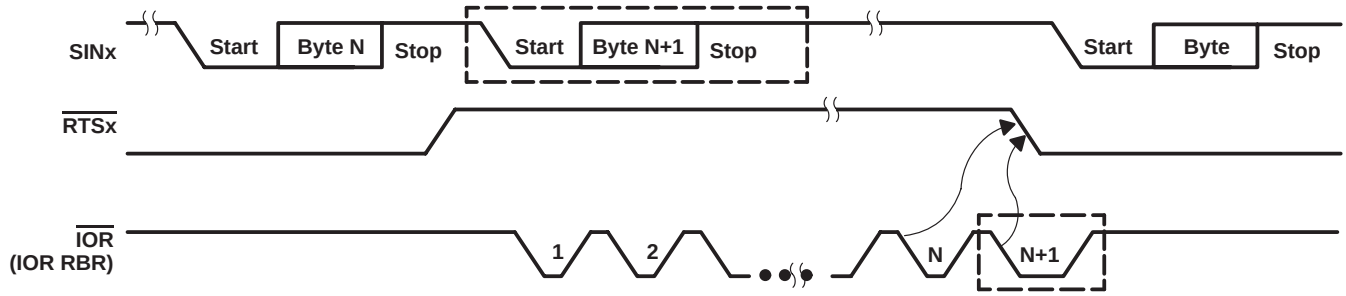
auto-$\overline{\text{CTS}}$ and auto-$\overline{\text{RTS}}$ functional timing



NOTE A: When $\overline{\text{CTSx}}$ is low, the transmitter keeps sending serial data out. When $\overline{\text{CTSx}}$ goes high before the middle of the last stop bit of the current byte, the transmitter finishes sending the current byte but it does not send the next byte. When $\overline{\text{CTSx}}$ goes from high to low, the transmitter begins sending data again.

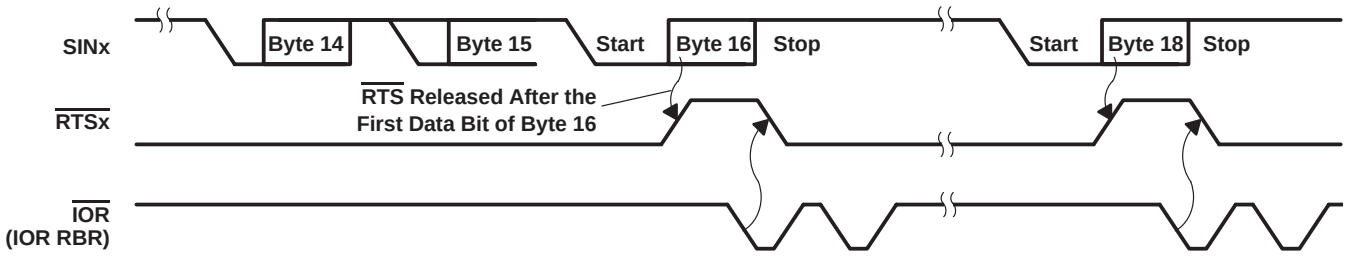
Figure 2. $\overline{\text{CTS}}$ Functional Timing

The receiver-FIFO trigger level can be set to 1, 4, 8, or 14 bytes. These are described in Figures 3 and 4.



NOTES: A. N = RCV-FIFO trigger level (1, 4, or 8 bytes)
B. The two blocks in dashed lines cover the case where an additional byte is sent as described in the preceding *auto-RTS* section.

Figure 3. $\overline{\text{RTS}}$ Functional Timing, RCV-FIFO Trigger Level = 1,4, or 8 Bytes



NOTE A: $\overline{\text{RTSx}}$ is deasserted when the receiver receives the first data bit of the sixteenth byte. The receive FIFO is full after finishing the sixteenth byte. $\overline{\text{RTSx}}$ is asserted again when there is at least one byte of space available and no incoming byte is in processing or there is more than one byte of space available. When the receive FIFO is full, the first receive buffer-register read reasserts $\overline{\text{RTSx}}$.

Figure 4. $\overline{\text{RTS}}$ Functional Timing, Receiver - FIFO Trigger Level = 14 Bytes

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)†

Supply voltage range, V_{CC} . . . . . –0.5 V to 6 V
Input voltage range, V_I . . . . . –0.5 V to V_{CC} + 0.5 V
Output voltage range, V_O . . . . . –0.5 V to V_{CC} + 0.5 V
Input clamp current, I_{IK} (V_I < 0 or V_I > V_{CC}) (See Note 2) . . . . . ±20 mA
Output clamp current, I_{OK} (V_O < 0 or V_O > V_{CC}) (See Note 3) . . . . . ±20 mA
Output clamp current, level shift, I_{OK} (V_O < 0 or V_O > V_{CC}) (See Note 3) . . . . . ±20 mA
Virtual junction, T_J . . . . . 150°C
Storage temperature range, T_{stg} . . . . . –65°C to 150°C

† Stresses beyond those listed under “absolute maximum ratings” may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under “recommended operating conditions” is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

NOTES: 2. This applies to external input and bidirectional buffers. V_I < V_{CC} does not apply to fail-safe terminals.
3. This applies to external output and bidirectional buffers. V_O < V_{CC} does not apply to fail-safe terminals.

recommended operating conditions

	MIN	NOM	MAX	UNIT
Supply voltage, V_{CC}	4.75	5	5.25	V
Input voltage, V_I	0		V_{CC}	V
High-level input voltage, V_{IH}	2		V_{CC}	V
Low-level input voltage, V_{IL}	–0.5		0.8	V
Input transition (rise and fall) time, t_t	0		25	ns
Operating ambient temperature range, T_A	0	25	70	°C
Virtual junction temperature, T_J	0	25	115	°C

electrical characteristics over recommended ranges of supply voltage and operating free-air temperature (unless otherwise noted)

PARAMETER		TEST CONDITIONS	COMMERCIAL		UNIT
			MIN	MAX	
V_{OH}	High-level output voltage	I_{OH} = Rated†	V_{CC} –0.8		V
V_{OL}	Low-level output voltage	I_{OL} = Rated†		0.5	V
V_{IT+}	Positive-going input threshold voltage‡	TTL compatible		2	V
V_{IT-}	Negative-going input threshold voltage‡	TTL compatible	0.8		V
V_{hys}	Hysteresis‡ ($V_{IT+} - V_{IT-}$)	TTL compatible	0.25	0.7	V
I_{OZ}	3-state-output Hi-Z current	V_I = V_{CC} or GND§		±10	μA
I_{IL}	Low-level input current	V_I = GND		–1	μA
I_{IH}	High-level input current	V_I = V_{CC}		1	μA

† 12 mA for 1284 control, 3 mA for all other outputs
‡ Applies to external input and bidirectional buffers with hysteresis. All input buffers have hysteresis.
§ A 3-state or open-drain output must be in the high-impedance state.
NOTE 4: All 1284 output and data terminals are low-noise TTL 12 mA drivers. The type of the driver, push/pull or open drain, is switched dynamically based on the 1284 mode.

system timing requirements over recommended ranges of supply voltage and operating free-air temperature

		ALT. SYMBOL	FIGURE	TEST CONDITIONS	MIN	MAX	UNIT
t_{cR}	Cycle time, read ($t_{w7} + t_{d8}$)	RC			65		ns
t_{cW}	Cycle time, write ($t_{w6} + t_{d5}$)	WC			59		ns
t_{w1}	Pulse duration, $\overline{IOW}$	t_{WR}	5		50		ns
t_{w2}	Pulse duration, $\overline{IOR}$	t_{RD}	6		50		ns
t_{w3}	Pulse duration, RESET	t_{MR}			1		µs
t_{su1}	Setup time, data valid before $\overline{IOW}\uparrow$	t_{DS}	5		20		ns
t_{su2}	Setup time, $\overline{CTS}\uparrow$ before midpoint of stop bit		16			10	ns
t_{h1}	Hold time, CS valid after $\overline{IOW}\uparrow$	t_{WCS}	5		15		ns
t_{h2}	Hold time, address valid after $\overline{IOW}\uparrow$	t_{WA}	5		15		ns
t_{h3}	Hold time, data valid after $\overline{IOW}\uparrow$	t_{DH}	5		15		ns
t_{h4}	Hold time, chip select valid after $\overline{IOR}\uparrow$	t_{RCS}	6		20		ns
t_{h5}	Hold time, address valid after $\overline{IOR}\uparrow$	t_{RA}	6		20		ns
t_{d1}	Delay time, CS valid before $\overline{IOW}\downarrow$	t_{CSW}	5		10		ns
t_{d2}	Delay time, address valid before $\overline{IOW}\downarrow$	t_{AW}	5		13		ns
t_{d3}	Delay time, CS valid to $\overline{IOR}\downarrow$	t_{CSR}	6		10		ns
t_{d4}	Delay time, address valid to $\overline{IOR}\downarrow$	t_{AR}	6		13		ns
t_{d5}	Delay time, $\overline{IOR}\downarrow$ to data valid	t_{RVD}	6	C_L = 75 pF	20		ns
t_{d6}	Delay time, $\overline{IOR}\uparrow$ to floating data	t_{HZ}	6	C_L = 75 pF	10		ns

receiver switching characteristics over recommended ranges of supply voltage and operating free-air temperature (see Note 5)

PARAMETER		ALT. SYMBOL	FIGURE	TEST CONDITIONS	MIN	MAX	UNIT
t_{d7}	Delay time, stop to INTRPTx$\uparrow$ or read RBR$\uparrow$ to LSI INTRPT$\uparrow$ or stop to $\overline{RXRDY}\downarrow$	t_{SINT}	7, 8, 9, 10, 11			1	RCLK cycle
t_{d8}	Delay time, read RBR/INTRPT or read to $\overline{RXRDY}\uparrow$ or $\overline{IOR}\uparrow$ to INTRPTx$\downarrow$	t_{RINT}	7, 8, 9, 10, 11	C_L = 75 pF		70	ns

NOTE 5: In the FIFO mode, the read cycle (RC) = 425 ns (min) between reads of the receive FIFO and the status registers (interrupt-identification register or line-status register).

transmitter switching characteristics over recommended ranges of supply voltage and operating free-air temperature

PARAMETER		ALT. SYMBOL	FIGURE	TEST CONDITIONS	MIN	MAX	UNIT
t_{d9}	Delay time, INTRPTx↓ to SOUTx↓	t_{IRS}	12		8	24	baud-out cycles
t_{d10}	Delay time, start to INTRPTx↑	t_{STI}	12		8	10	baud-out cycles
t_{d11}	Delay time, $\overline{IOW}$↑ to INTRPTx↓	t_{HR}	12	C_L = 75 pF		50	ns
t_{d12}	Delay time, $\overline{IOW}$↓ to INTRPTx↑	t_{SI}	12		16	34	baud-out cycles
t_{d13}	Delay time, $\overline{IOR}$ IIR↓ to INTRPTx↓	t_{IR}	12	C_L = 75 pF		35	ns
t_{d14}	Delay time, $\overline{IOW}$↑ to $\overline{TXRDYx}$↑	t_{WXI}	13,14	C_L = 75 pF		30	ns
t_{d15}	Delay time, SOUTx↓ to $\overline{TXRDYx}$↓	t_{SXA}	13, 14	C_L = 75 pF		9	baud-out cycles

† THRE = transmitter holding-register empty; IIR = interrupt-identification register.

modem-control switching characteristics over recommended ranges of supply voltage and operating free-air temperature, C_L = 75 pF

PARAMETER		ALT. SYMBOL	FIGURE	MIN	MAX	UNIT
t_{d16}	Delay time, $\overline{CTSx}$, $\overline{DSRx}$, $\overline{DCDx}$↓ to INTRPTx↑ or $\overline{RI}$↑ to INTRPTx↑	t_{RIM}	15		35	ns
t_{d17}	Delay time, $\overline{IOR}$↑ to INTRPTx↓		15		24	baudout cycles
t_{d18}	Delay time, $\overline{CTSx}$↓ to SOUTx↓		16		2	baudout cycles
t_{d19}	Delay time, midpoint of stop bit to $\overline{RTSx}$↑		17		2	baudout cycles
t_{d20}	Delay time, $\overline{IOR}$↓ to $\overline{RTSx}$↓		17		2	baudout cycles
t_{d21}	Delay time, first data bit of the sixteenth character to $\overline{RTSx}$↑		18		2	baudout cycles
t_{d22}	Delay time, $\overline{IOR}$↓ to $\overline{RTSx}$↓	t_{MDO}	18		40	ns
t_{d23}	Delay time, $\overline{IOW}$↑ to $\overline{RTSx}$, $\overline{DTRx}$↑↓	t_{SIM}	15		35	ns

IR signal switching characteristics over recommended ranges of supply voltage and operating free-air temperature, C_L = 75 pF

PARAMETER		ALT. SYMBOL	FIGURE	MIN	MAX	UNIT
t_{d24}	Delay time, internal SOUTx↓ to SOUTx↑ (IR mode)		19		8	baud-out cycles
t_{d25}	Delay time, incoming IRSINx↑ to internal SINx↓		19		15	ns

parallel-port timing requirements over recommended ranges of supply voltage and operating free-air temperature

PARAMETER		FIGURE	MIN	MAX	UNITS
t_{d26}	Delay time, $\overline{FAULT}$↓ to PINTR (ECP)↓	20		20	ns
t_{d27}	Delay time, $\overline{ACK}$↓ to PINTR (EPP)↓	20		20	ns
t_{w4}	Pulse duration, PINTR (ECP)↓ to PINTR (ECP)↑ (ECP and EPP modes)	20	80	100	ns

parallel-port EPP data or address write cycle timing requirements over recommended ranges of supply voltage and operating free-air temperature

PARAMETER		FIGURE	MIN	MAX	UNITS
t_{h6}	Hold time, BUSY↓ to PD invalid	21	60	140	ns
t_{d28}	Delay time, $\overline{\text{IOW}}$↓ to PD valid	21	10	30	ns
t_{d29}	Delay time, BUSY↓ to $\overline{\text{STROBE}}$↑	21	60	100	ns
t_{d30}	Delay time, $\overline{\text{STROBE}}$↓ to $\overline{\text{AUTOFD}}$↓	21	0	10	ns
t_{d31}	Delay time, BUSY↑ to $\overline{\text{AUTOFD}}$↑	21	60	100	ns
t_{d32}	Timeout, $\overline{\text{IOW}}$↓ to BUSY↑	21	10	12	µs
t_{d33}	Delay time, $\overline{\text{SELECTIN}}$↑ to BUSY↓	21	200		ns
t_{d34}	Delay time, $\overline{\text{IOW}}$↓ to IOCHRDY↓	21	0	25	ns
t_{d35}	Delay time, BUSY↑ to IOCHRDY↑	21	60	120	ns
t_{d36}	Delay time, IOCHRDY↑ to $\overline{\text{IOW}}$↑	21	30		ns
t_{d37}	Delay time, $\overline{\text{IOW}}$↓ to $\overline{\text{STROBE}}$↓	21	0	40	ns
t_{d38}	Delay time, $\overline{\text{IOW}}$↑ to $\overline{\text{IOW}}$ or $\overline{\text{IOR}}$↓	21	60		ns
t_{d39}	Delay time, BUSY↓ to $\overline{\text{STROBE}}$↓	21	20	40	ns

parallel-port EPP data or adress read cycle timing requirements over recommended ranges of supply voltage and operating free-air temperature

PARAMETER		FIGURE	MIN	MAX	UNITS
t_{h7}	Hold time, $\overline{\text{AUTOFD}}$, $\overline{\text{SELECTIN}}$↑ to PD hi-Z	22	10		ns
t_{h8}	Hold time, $\overline{\text{IOR}}$↑ to D hi-Z (Hold Time)	22	0	20	ns
t_{d40}	Delay time, BUSY↑ to $\overline{\text{AUTOFD}}$, $\overline{\text{SELECTIN}}$↑	22	120	200	ns
t_{d41}	Delay time, $\overline{\text{AUTOFD}}$, $\overline{\text{SELECTIN}}$↓ to PD Valid	22	110		ns
t_{d42}	Delay time, $\overline{\text{IOR}}$↓ to IOCHRDY↓	22	0	25	ns
t_{d43}	Delay time, BUSY↑ to IOCHRDY↑	22	80	130	ns
t_{d44}	Delay time, PD valid to D valid	22	0	25	ns
t_{d45}	Timeout, $\overline{\text{IOR}}$↓ to $\overline{\text{BUSY}}$↑	22	10	12	µs
t_{d46}	Delay time, $\overline{\text{IOR}}$↑ to $\overline{\text{IOW}}$ or $\overline{\text{IOR}}$↓	22	60		ns
t_{d47}	Delay time, IOCHRDY↑ to $\overline{\text{IOR}}$↑	22	30		ns

parallel-port FIFO timing requirements over recommended ranges of supply voltage and operating free-air temperature

PARAMETER		FIGURE	MIN	MAX	UNITS
t_{su3}	Setup time, PD valid to $\overline{\text{STROBE}}$↓	23	550		ns
t_{h9}	Hold time, PD hold from $\overline{\text{STROBE}}$↑	23	500		ns
t_{w5}	Pulse duration, $\overline{\text{STROBE}}$ pulse width low	23	500		ns
t_{d48}	Delay time, $\overline{\text{STROBE}}$↓ to BUSY↑ active	23		125	ns
t_{d49}	Delay time, BUSY↓ to $\overline{\text{STROBE}}$↓	23	600		ns

parallel-port (ECP) reverse timing requirements over recommended ranges of supply voltage and operating free-air temperature

	PARAMETER	FIGURE	MIN	MAX	UNITS
t_{su4}	Setup time, PD valid to $\overline{ACK}\downarrow$	24	0		ns
t_{d50}	Delay time, $\overline{AUTOFD}\downarrow$ to PD changed	24	0		ns
t_{d51}	Delay time, $\overline{ACK}\uparrow$ to $\overline{AUTOFD}\downarrow$	24	80	200	ns
t_{d52}	Delay time, $\overline{ACK}\uparrow$ to $\overline{AUTOFD}\downarrow$	24	80	120	ns
t_{d53}	Delay time, $\overline{AUTOFD}\downarrow$ to $\overline{ACK}\downarrow$	24	25		ns
t_{d54}	Delay time, $\overline{AUTOFD}\uparrow$ to $\overline{ACK}\uparrow$	24	150		ns

parallel-port (ECP) forward timing requirements over recommended ranges of supply voltage and operating free-air temperature

	PARAMETER	FIGURE	MIN	MAX	UNITS
t_{su5}	Setup time, PD valid to $\overline{STROBE}\downarrow$	25	0	90	ns
t_{h10}	Hold time, BUSY$\downarrow$ to PD changed	25	30	180	ns
t_{d55}	Delay time, $\overline{AUTOFD}$ valid to $\overline{STROBE}\downarrow$	25	0	45	ns
t_{d56}	Delay time, BUSY$\downarrow$ to $\overline{AUTOFD}$ changed	25	25	180	ns
t_{d57}	Delay time, $\overline{STROBE}\downarrow$ to BUSY$\uparrow$	25	370		ns
t_{d58}	Delay time, $\overline{STROBE}\uparrow$ to BUSY$\downarrow$	25	295		ns
t_{d59}	Delay time, BUSY$\downarrow$ to $\overline{STROBE}\downarrow$	25	80	120	ns
t_{d60}	Delay time, BUSY$\uparrow$ to $\overline{STROBE}\uparrow$	25	20	60	ns

PARAMETER MEASUREMENT INFORMATION

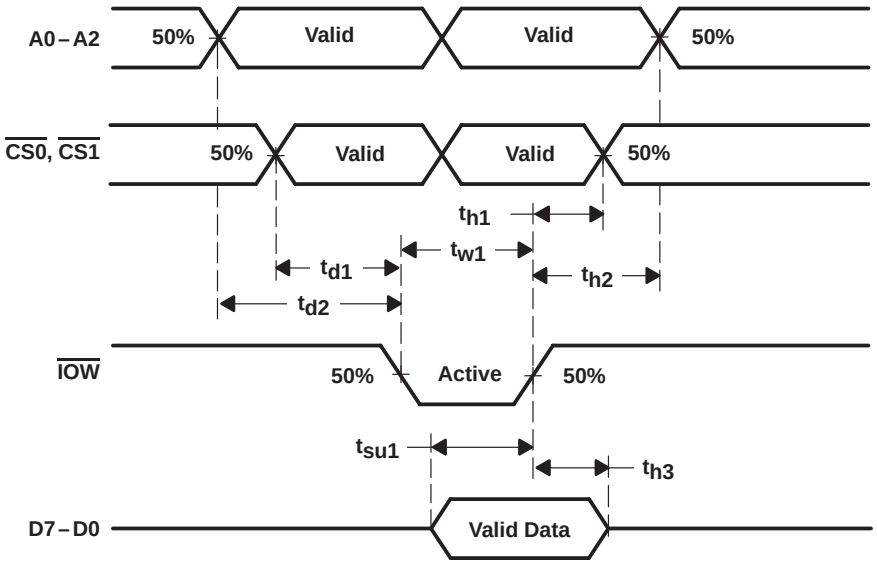


Figure 5. Write-Cycle Timing

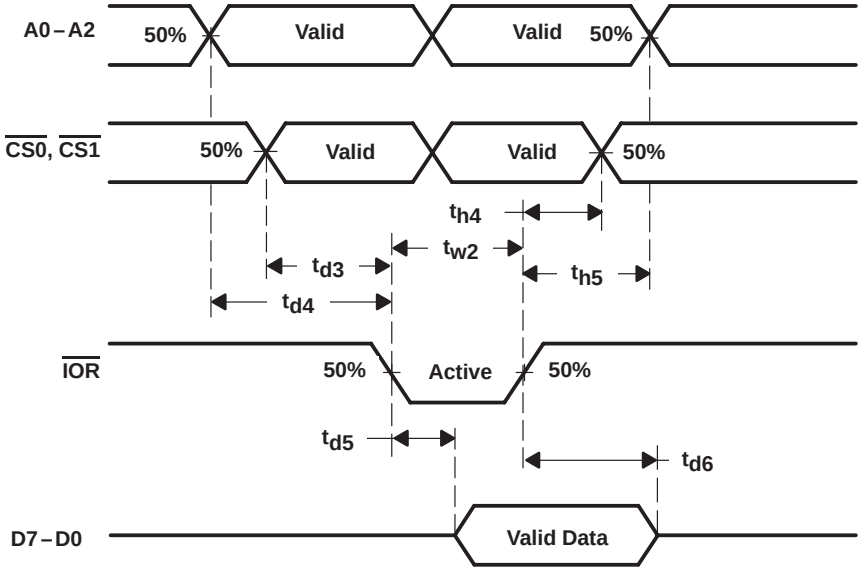


Figure 6. Read-Cycle Timing

PARAMETER MEASUREMENT INFORMATION

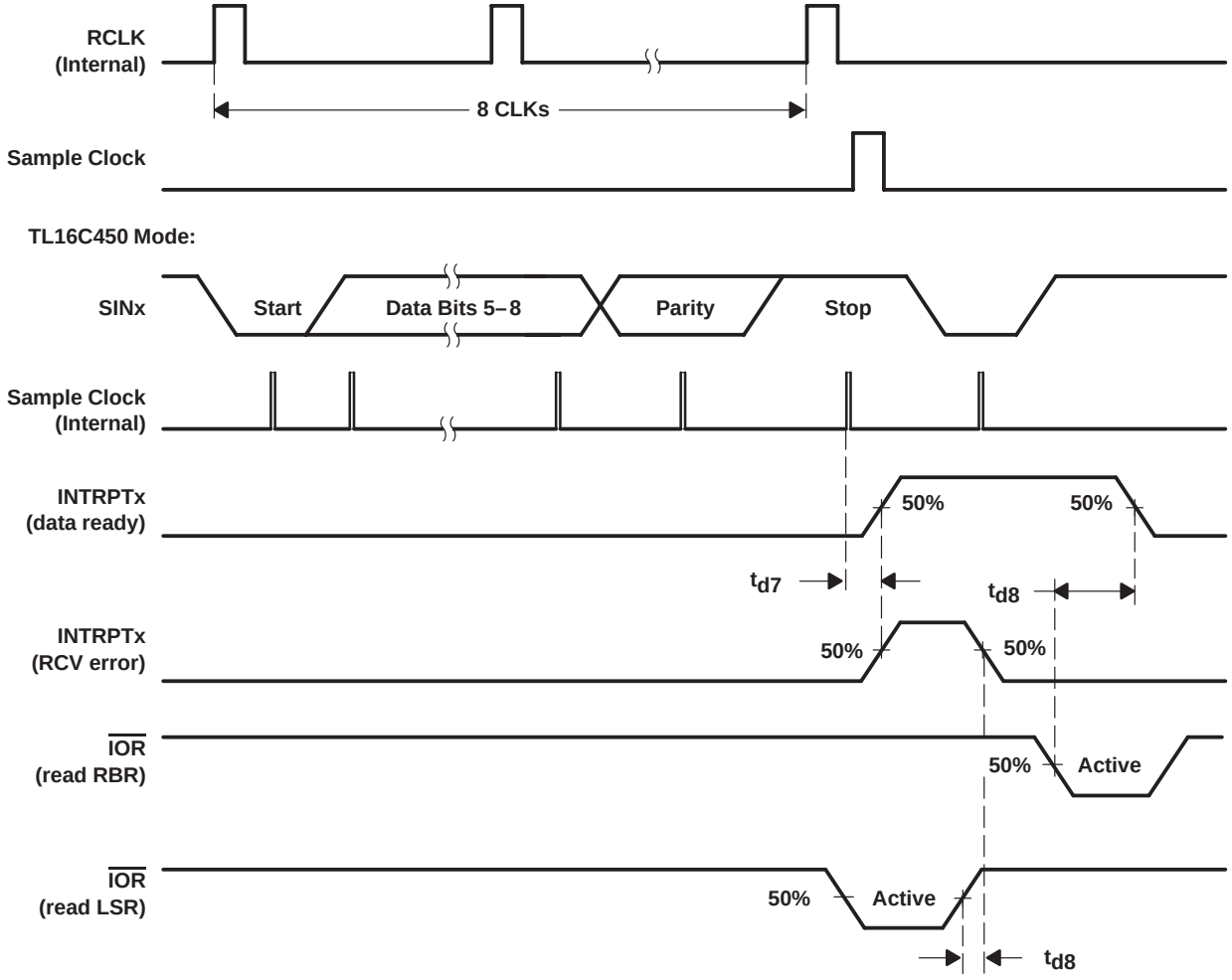
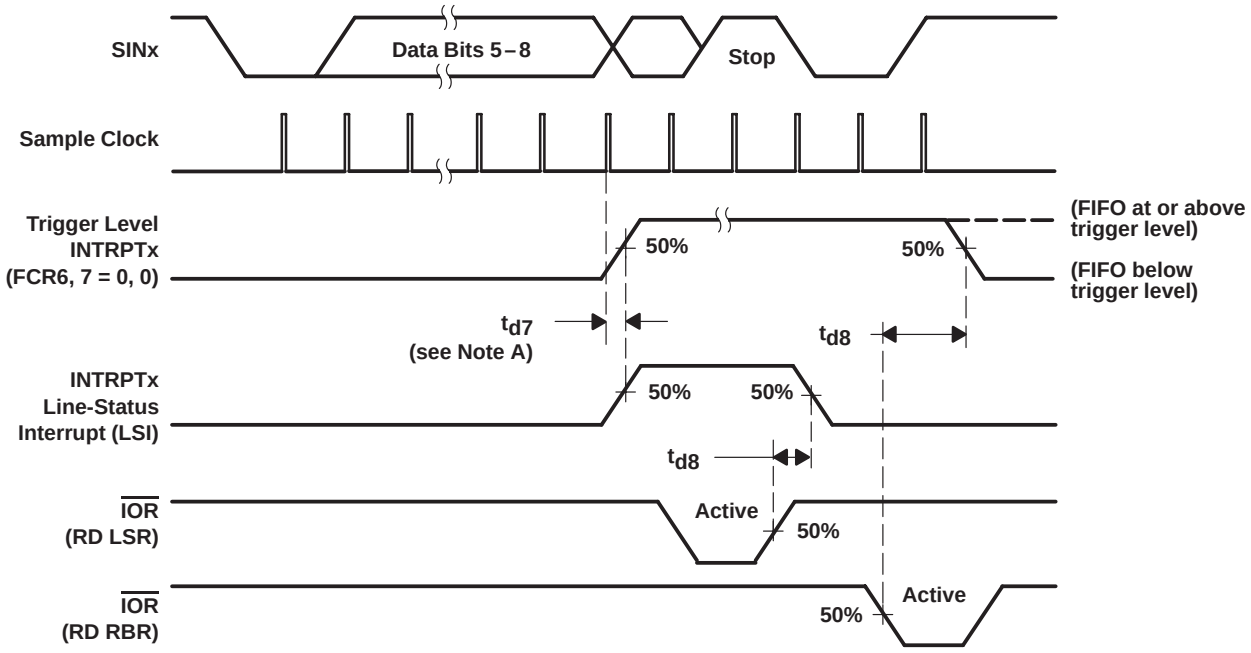


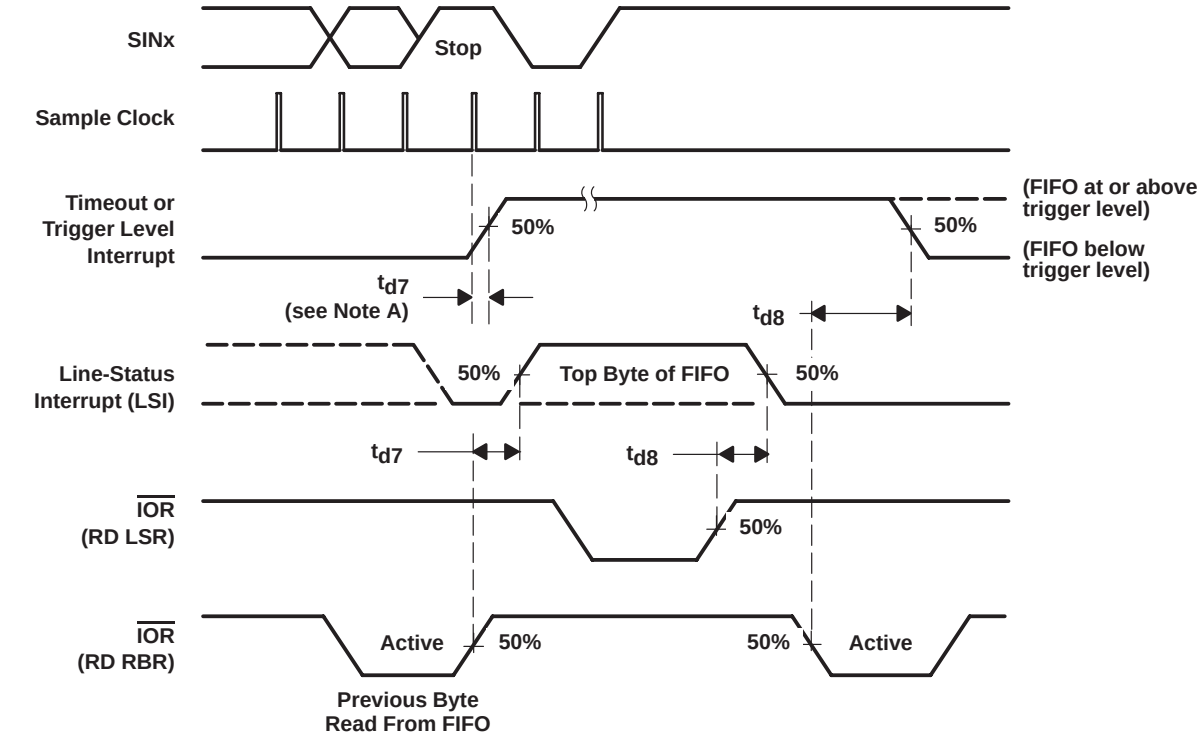
Figure 7. Receiver Timing

PARAMETER MEASUREMENT INFORMATION



NOTE A: For a timeout interrupt, t_{d7} = 9 RCLKs.

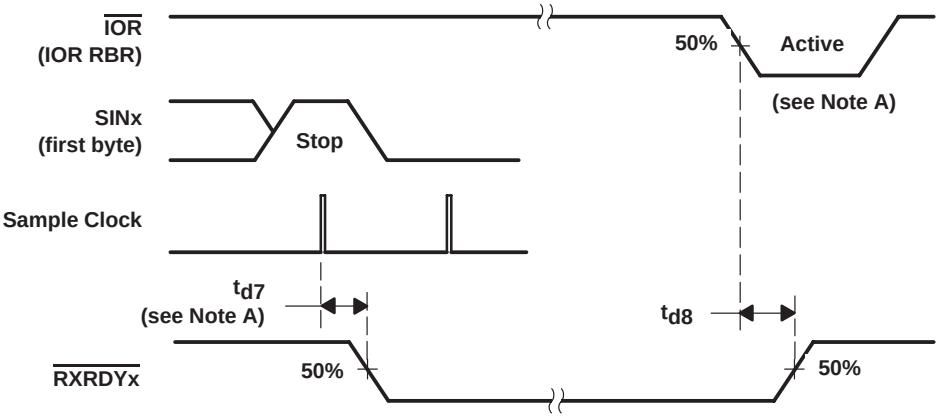
Figure 8. Receive FIFO First Byte (Sets DR Bit)



NOTE A: For a timeout interrupt, t_{d7} = 9 RCLKs.

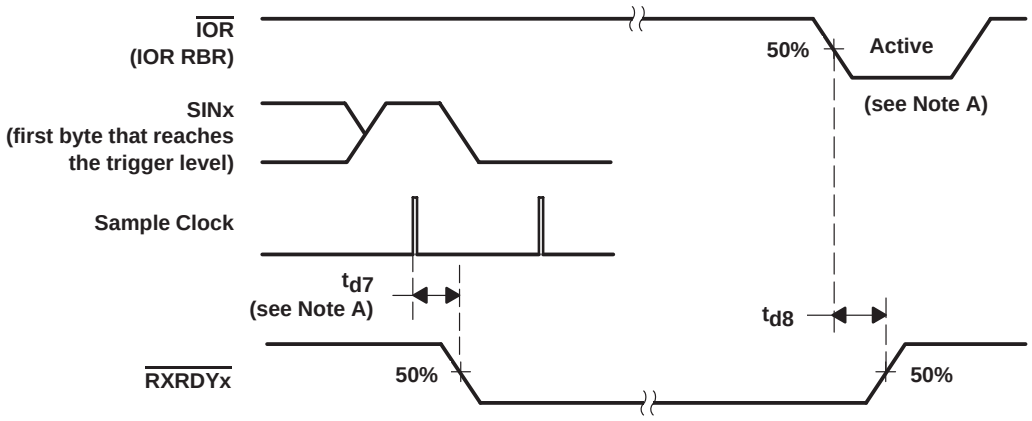
Figure 9. Receive FIFO Bytes Other Than the First Byte (DR Internal Bit Already Set)

PARAMETER MEASUREMENT INFORMATION



NOTE A: For a timeout interrupt, t_{d7} = 9 RCLKs.

Figure 10. Receiver Ready ($\overline{\text{RXRDY}}$), FCR0 = 0 or FCR0 = 1 and FCR3 = 0 (Mode 0)



NOTES: A. This is the reading of the last byte in the FIFO.
B. For a timeout interrupt, t_{d7} = 9 RCLKs.

Figure 11. Receiver Ready ($\overline{\text{RXRDY}}$), FCR0 = 1 or FCR3 = 1 (Mode 1)

PARAMETER MEASUREMENT INFORMATION

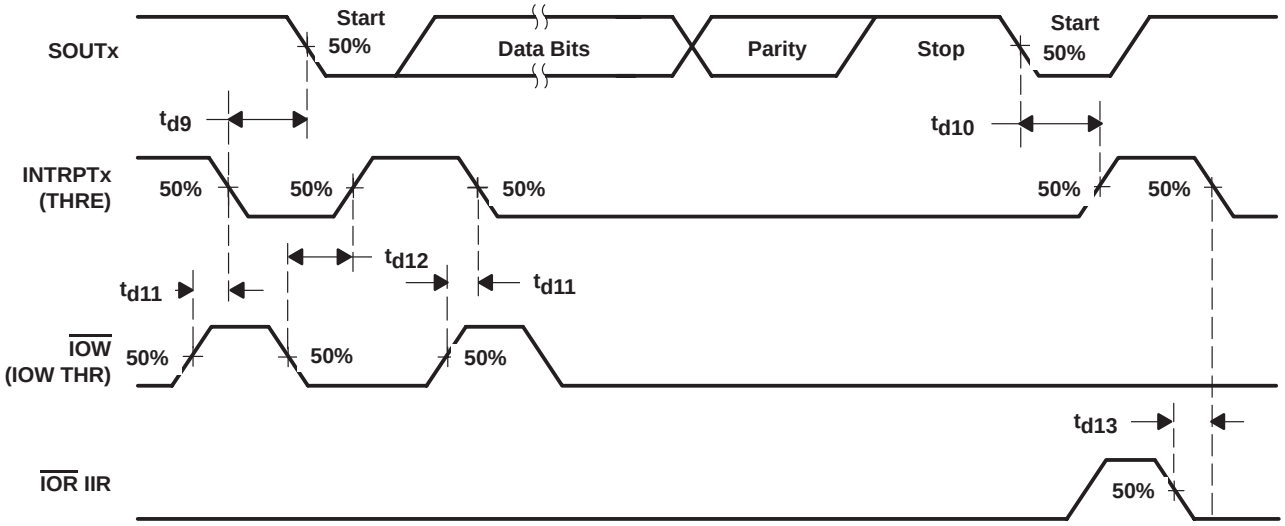


Figure 12. Transmitter Timing Waveforms

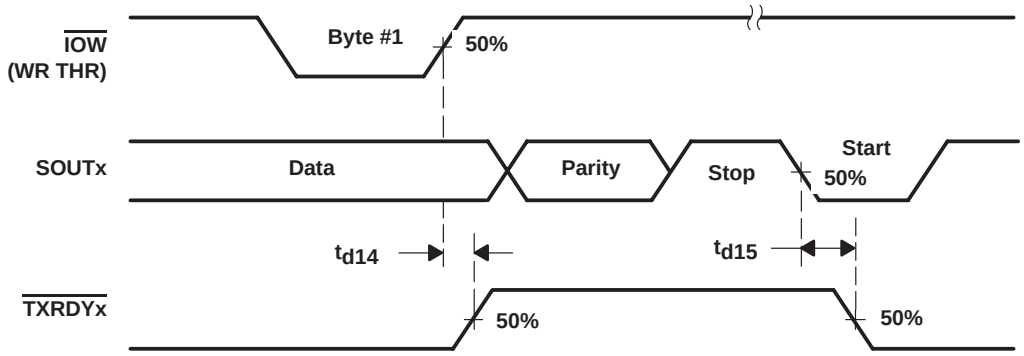


Figure 13. Transmitter Ready (TXRDYx), FCR0 = 0 or FCR0 = 1 and FCR3 = 0 (Mode 0)

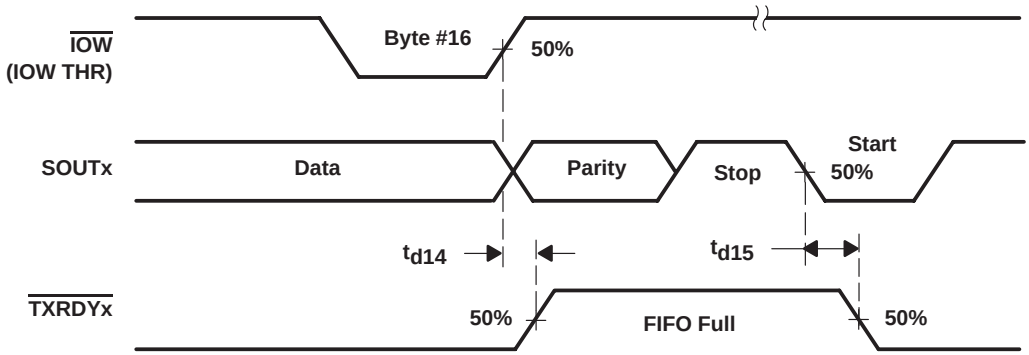


Figure 14. Transmitter Ready (TXRDYx), FCR0 = 1 and FCR3 = 1 (Mode 1)

PARAMETER MEASUREMENT INFORMATION

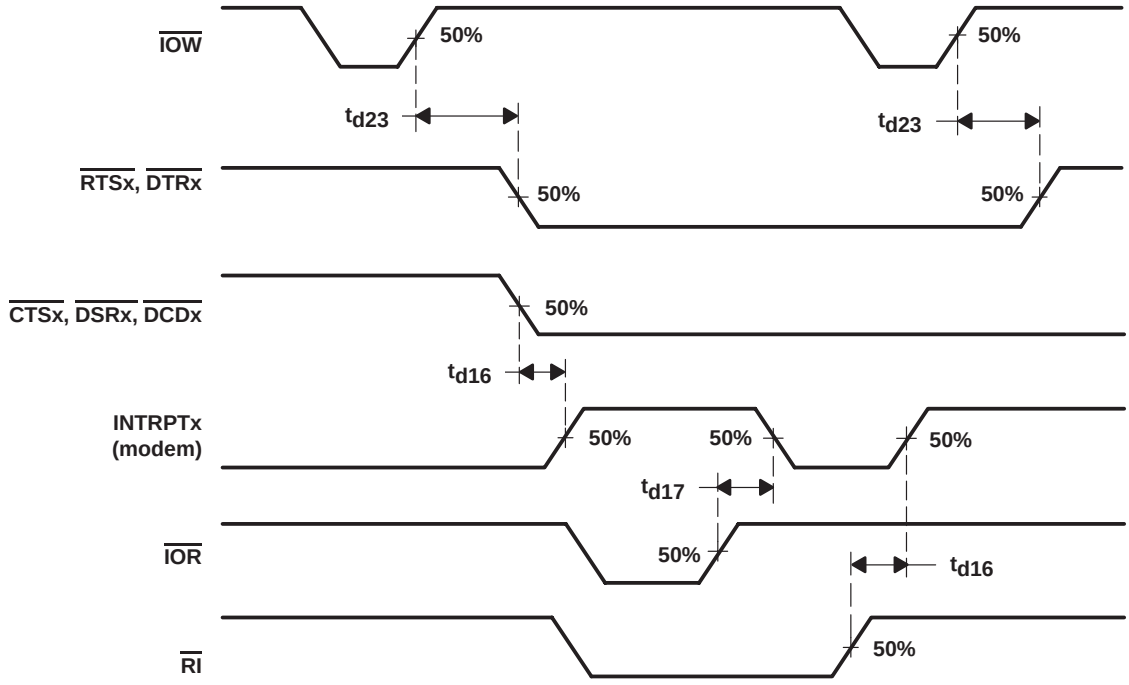


Figure 15. Modem-Control Timing

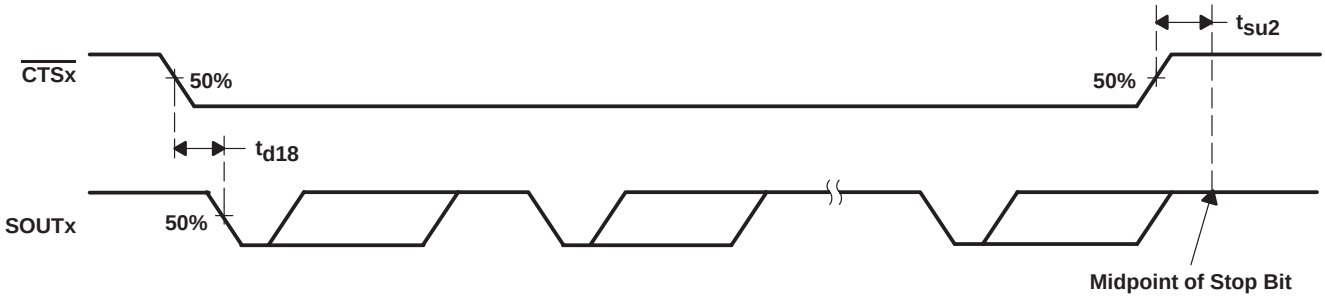


Figure 16. $\overline{\text{CTSx}}$ and SOUTx Auto Flow Control Timing (Start and Stop)

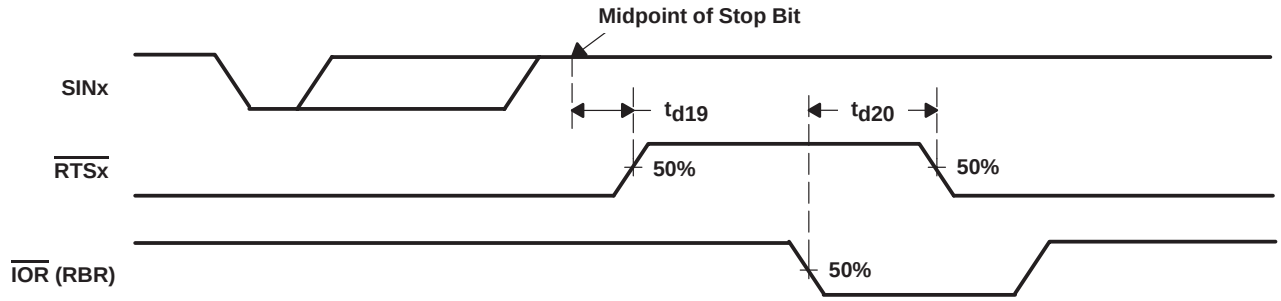


Figure 17. Auto-$\overline{\text{RTSx}}$ Timing for Receiver (RCV) Threshold of 1, 4, or 8

PARAMETER MEASUREMENT INFORMATION

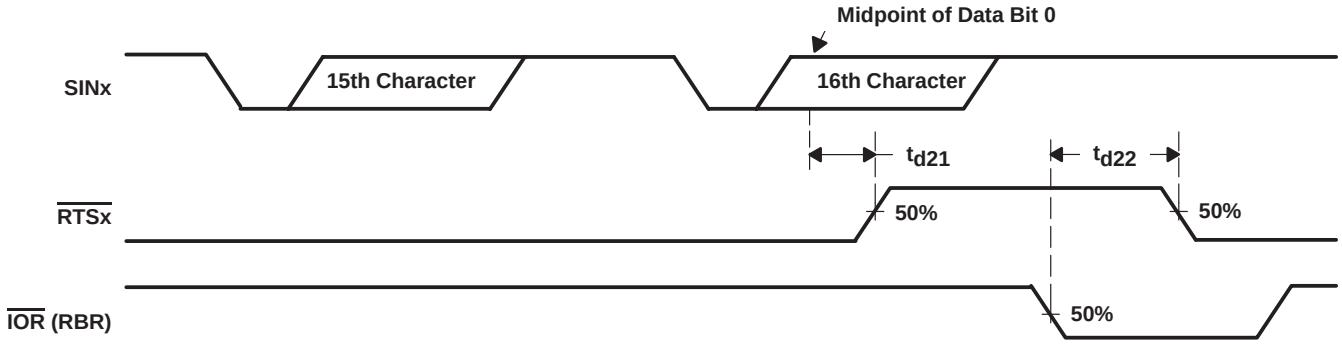


Figure 18. Auto-$\overline{\text{RTS}}$ Timing for Receiver Threshold of 14 Bytes

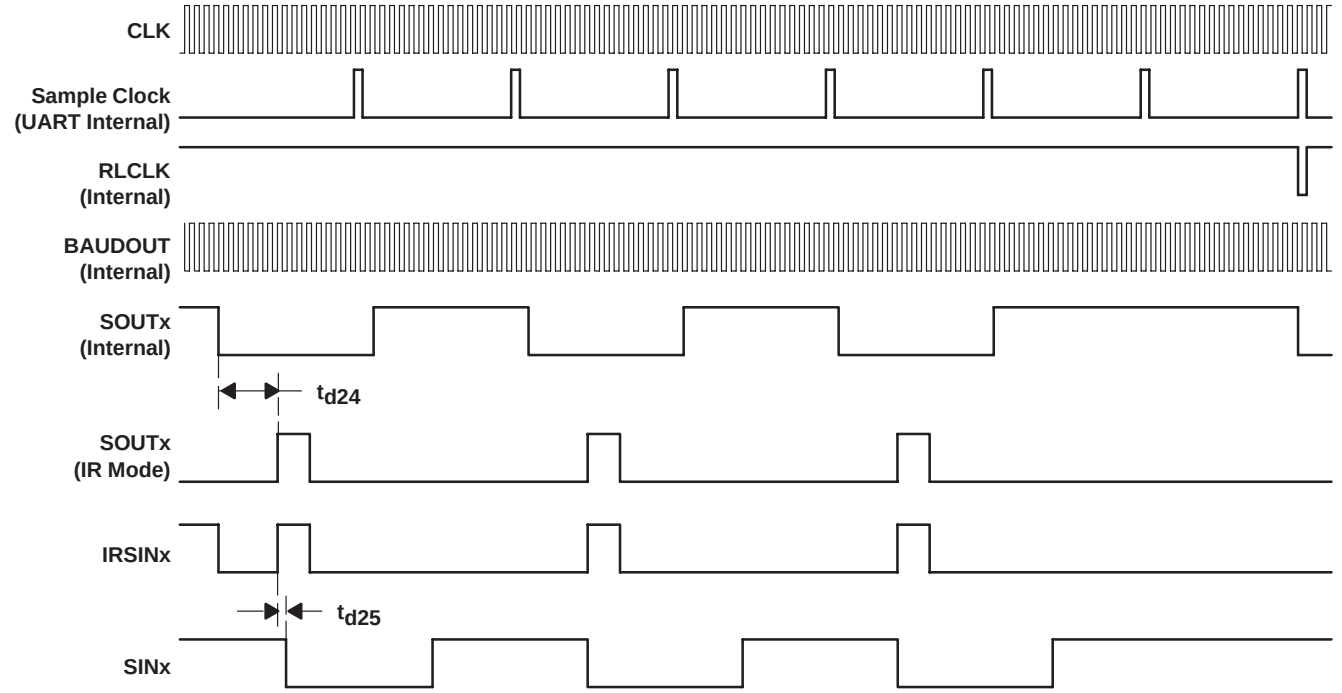


Figure 19. IR Signal Transfer

PARAMETER MEASUREMENT INFORMATION

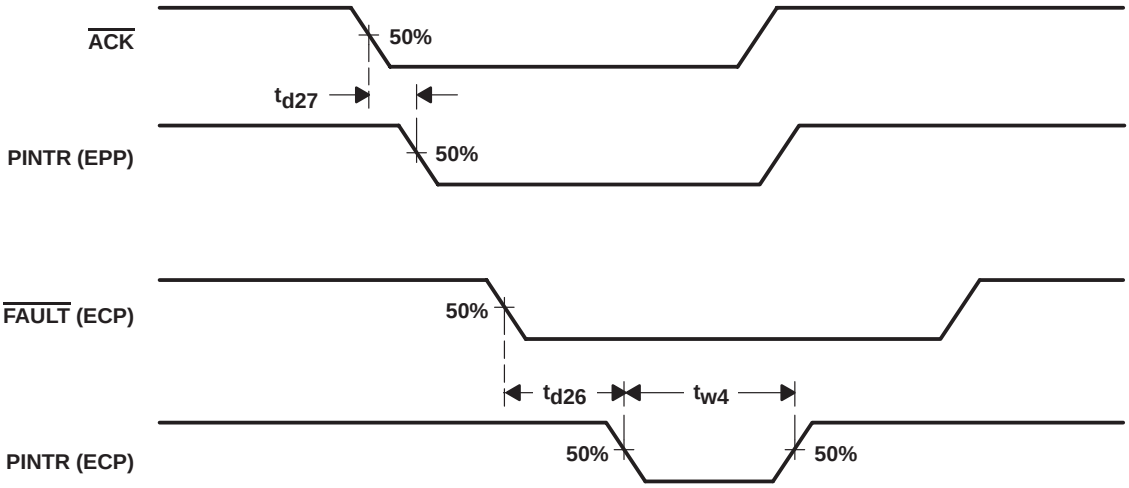


Figure 20. Interrupt Timing Diagram

PARAMETER MEASUREMENT INFORMATION

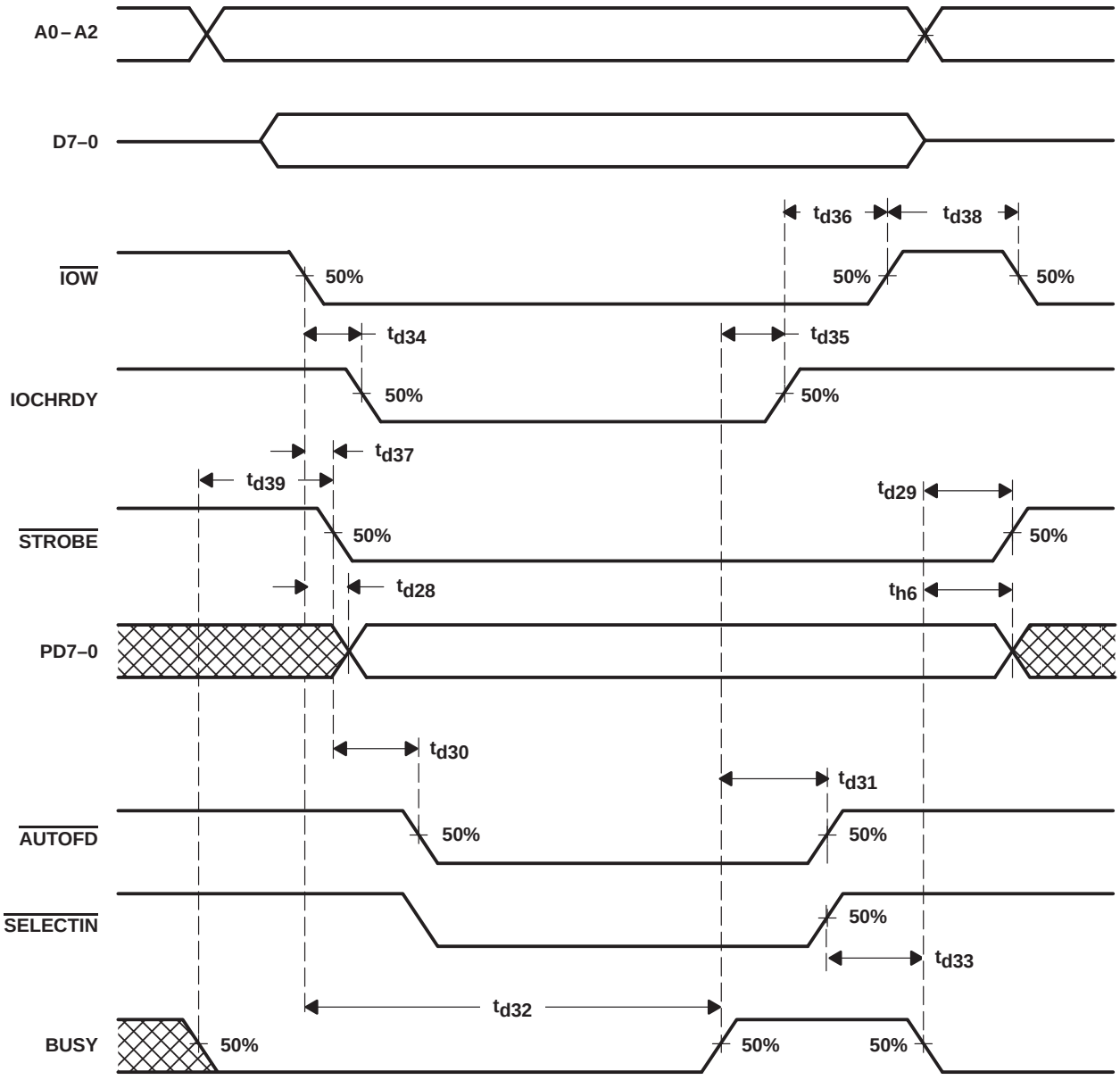


Figure 21. EPP Write Cycle

PRINCIPLES OF OPERATION

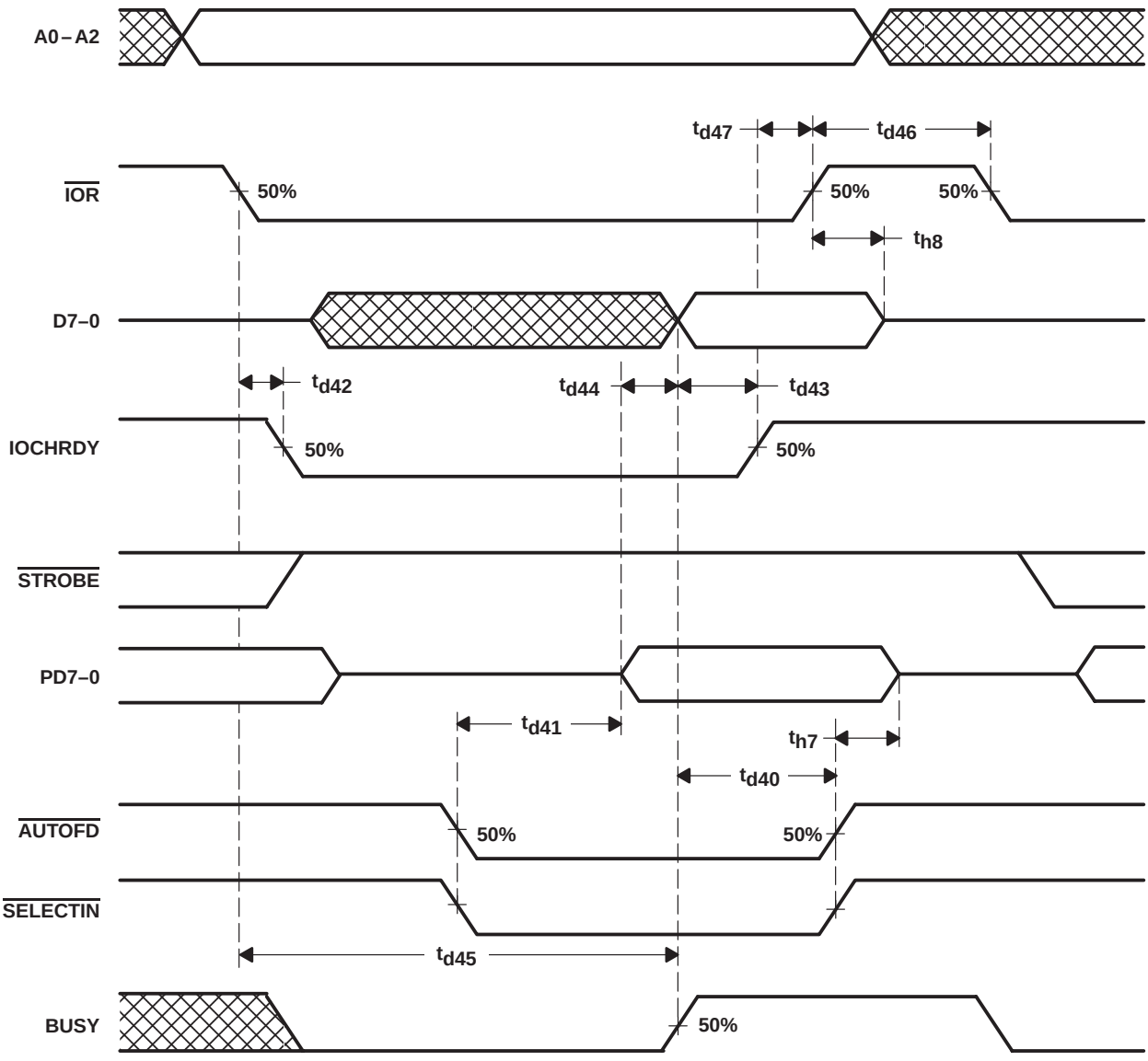


Figure 22. EPP Read Cycle Timing

PRINCIPLES OF OPERATION

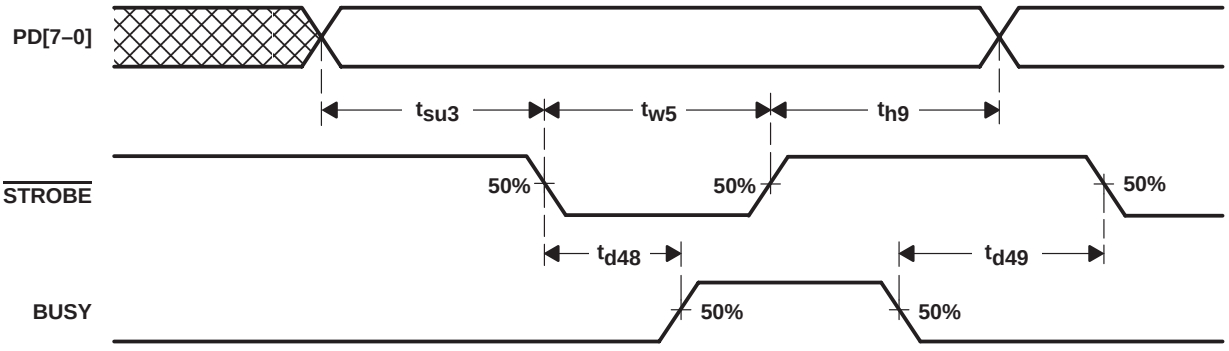


Figure 23. Parallel Port FIFO Timing

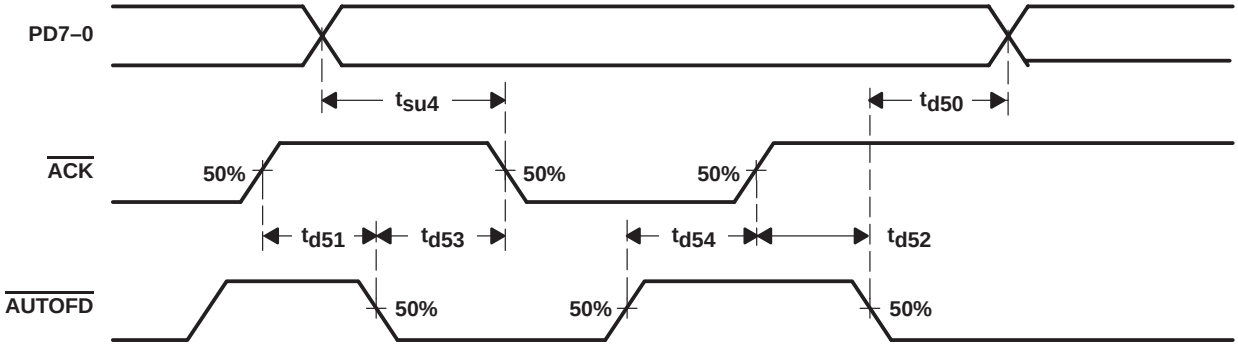


Figure 24. ECP Parallel Port Reverse Timing

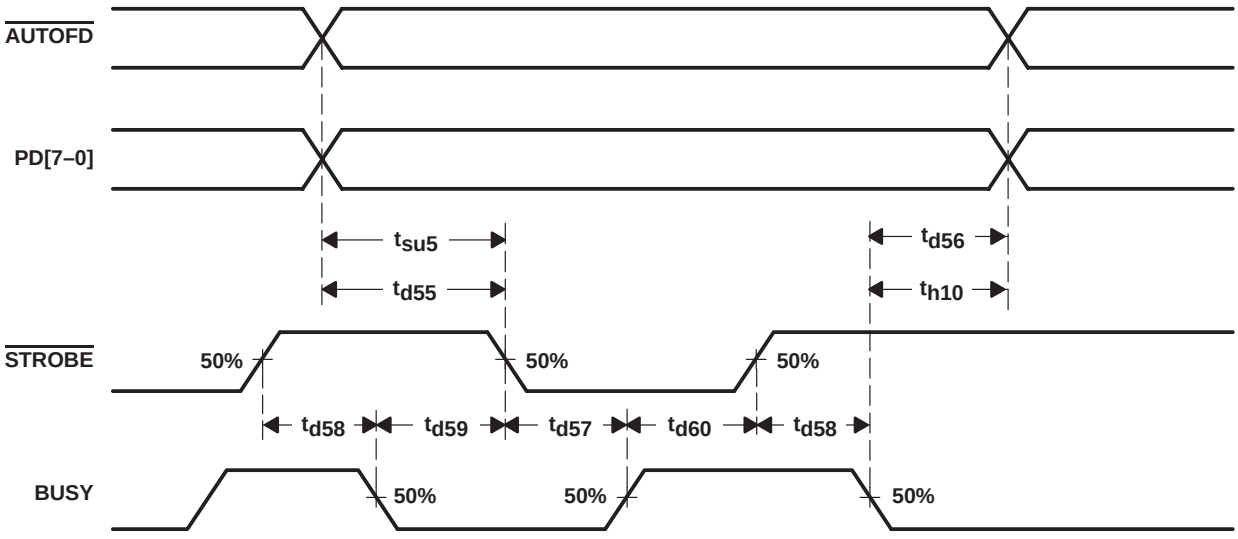


Figure 25. ECP Parallel Port Forward Timing

PRINCIPLES OF OPERATION

Table 1. Register Selection

DLAB†	A2	A1	A0	REGISTER
0	L	L	L	Receiver buffer (read), transmitter holding register (write)
0	L	L	H	Interrupt enable
X	L	H	L	Interrupt identification (read only)
X	L	H	L	FIFO control (write)
X	L	H	H	Line control
X	H	L	L	Modem control
X	H	L	H	Line status
X	H	H	L	Modem status
X	H	H	H	Scratch
1	L	L	L	Divisor latch (LSB)
1	L	L	H	Divisor latch (MSB)

† The divisor-latch access bit (DLAB) is the most significant bit of the line-control register. The DLAB signal is controlled by writing to this bit location (see Table 4).

accessible registers

The system programmer, using the CPU, has access to and control over any of the UART registers that are summarized in Table 2. These registers control UART operations, receive data, and transmit data. Descriptions of these registers follow Table 3.

Table 2. UART Reset Functions

REGISTER/SIGNAL	RESET CONTROL	RESET STATE
Interrupt Enable Register	Master Reset	All bits low (0–3 forced and 4–7 permanent)
Interrupt Identification Register	Master Reset	Bit 0 is high, bits 1, 2, 3, 6, and 7 are low, and bits 4–5 are permanently low
FIFO Control Register	Master Reset	All bits low
Line Control Register	Master Reset	All bits low
Modem Control Register	Master Reset	All bits low (6–7 permanent)
Line Status Register	Master Reset	Bits 5 and 6 are high; all other bits are low
Modem Status Register	Master Reset	Bits 0–3 are low; bits 4–7 are input signals
SOUT	Master Reset	High
INTRPT (receiver error flag)	Read LSR/MR	Low
INTRPT (received data available)	Read RBR/MR	Low
INTRPT (transmitter holding register empty)	Read IR/Write THR/MR	Low
INTRPT (modem-status changes)	Read MSR/MR	Low
$\overline{\text{RTS}}$	Master Reset	High
$\overline{\text{DTR}}$	Master Reset	High
Scratch Register	Master Reset	No effect
Divisor-Latch (LSB and MSB) Registers	Master Reset	No effect
Receiver Buffer Register	Master Reset	No effect
Transmitter Holding Register	Master Reset	No effect
RCVR FIFO	MR/FCR1–FCR0/ΔFCR0	All bits low
XMIT FIFO	MR/FCR2–FCR0/ΔFCR0	All bits low

PRINCIPLES OF OPERATION

Table 3. Summary of Accessible Registers

BIT NO.	REGISTER ADDRESS											
	0 DLAB = 0	0 DLAB = 0	1 DLAB = 0	2	2	3	4	5	6	7	0 DLAB = 1	1 DLAB = 1
	Receiver Buffer Register (Read Only)	Transmitter Holding Register (Write Only)	Interrupt-Enable Register	Interrupt-Ident. Register (Read Only)	FIFO Control Register (Write Only)	Line-Control Register	Modem-Control Register	Line-Status Register	Modem-Status Register	Scratch Register	Divisor Latch (LSB)	Divisor Latch (MSB)
	RBR	THR	IER	IIR	FCR	LCR	MCR	LSR	MSR	SCR	DLL	DLM
0	Data Bit 0†	Data Bit 0	Enable Received-Data-Available Interrupt (ERBI)	0 if Interrupt Pending	FIFO Enable	Word-Length Select Bit 0 (WLS0)	Data Terminal Ready (DTR)	Data Ready (DR)	Delta Clear to Send (ΔCTS)	Bit 0	Bit 0	Bit 8
1	Data Bit 1	Data Bit 1	Enable Transmitter Holding-Register-Empty Interrupt (ETBEI)	Interrupt ID Bit 1	Receiver FIFO Reset	Word-Length Select Bit 1 (WLS1)	Request to Send (RTS)	Overrun Error (OE)	Delta Data-Set Ready (ΔDSR)	Bit 1	Bit 1	Bit 9
2	Data Bit 2	Data Bit 2	Enable Receiver Line-Status Interrupt (ELSI)	Interrupt ID Bit 2	Transmitter FIFO Reset	Number of Stop Bits (STB)	OUT1 (an unused internal signal)	Parity Error (PE)	Trailing-Edge Ring Indicator (TERI)	Bit 2	Bit 2	Bit 10
3	Data Bit 3	Data Bit 3	Enable Modem-Status Interrupt (EDSSI)	Interrupt ID Bit 3 (see Note 6)	DMA Mode Select	Parity Enable (PEN)	OUT2 Enable external interrupt (INT0 or INT1)	Framing Error (FE)	Delta Data-Carrier Detect (ΔDCD)	Bit 3	Bit 3	Bit 11
4	Data Bit 4	Data Bit 4	0	0	Reserved	Even-Parity Select (EPS)	Loop	Break Interrupt (BI)	Clear to Send (CTS)	Bit 4	Bit 4	Bit 12
5	Data Bit 5	Data Bit 5	0	0	Reserved	Stick Parity	Autoflow Control Enable (AFE)	Transmitter Holding Register (THRE)	Data Set Ready (DSR)	Bit 5	Bit 5	Bit 13
6	Data Bit 6	Data Bit 6	0	FIFOs Enabled (see Note 6)	Receiver Trigger (LSB)	Break Control	0	Transmitter Empty (TEMT)	Ring Indicator (RI)	Bit 6	Bit 6	Bit 14
7	Data Bit 7	Data Bit 7	0	FIFOs Enabled (see Note 6)	Receiver Trigger (MSB)	Divisor-Latch Access Bit (DLAB)	0	Error in RCVR FIFO (see Note 4)	Data-Carrier Detect (DCD)	Bit 7	Bit 7	Bit 15

† Bit 0 is the least significant bit. It is the first bit serially transmitted or received.

NOTE 6: These bits are always 0 in the TL16C450 mode. When DLAB = 1, the six least significant bits of the scratch register contain the prescaler value. IR logic is selected if the seventh bit of the scratch register is set to 1. Otherwise UART output is selected.

PRINCIPLES OF OPERATION

receiver buffer register (RBR)

The UART receiver section of the TL16PIR552 consists of a receiver shift register (RSR) and a receiver buffer register (RBR). The RBR is actually a 16-byte FIFO. Timing is supplied by the 16× receiver clock (RCLK). Receiver-section control is a function of the UART line-control register.

The UART RSR receives serial data from SIN. The RSR then concatenates the data and moves it into the RBR FIFO. In the TL16C450 mode, when a character is placed in the receiver buffer register and the received-data-available interrupt is enabled (IER0 = 1), an interrupt is generated. This interrupt is cleared when the data is read out of the receiver buffer register. In the FIFO mode, the interrupts are generated based on the control setup in the FIFO-control register.

transmitter holding register (THR)

The UART transmitter section of the TL16PIR552 consists of a transmitter holding register (THR) and a transmitter shift register (TSR). The THR is actually a 16-byte FIFO. Transmitter-section control is a function of the UART line-control register.

The UART THR receives data off the internal data bus and when the shift register is idle, moves it into the TSR. The TSR serializes the data and outputs it at SOUT. In the TL16C450 mode, if the THR is empty and the transmitter holding-register-empty (THRE) interrupt is enabled (IER1 = 1), an interrupt is generated. This interrupt is cleared when a character is loaded into the register. In the FIFO mode, the interrupts are generated based on the control setup in the FIFO-control register.

interrupt enable register (IER)

The interrupt-enable register enables each of the five types of interrupts (refer to Table 4) and INTRPT in response to an interrupt generation. The interrupt-enable register can also be used to disable the interrupt system by setting bits 0 through 3 to logic 0. The contents of this register are summarized in Table 3 and are described below.

Bit 0: When set to 1, bit 0 enables the received-data-available interrupt

Bit 1: When set to 1, bit 1 enables the transmitter holding-register-empty interrupt

Bit 2: When set to 1, bit 2 enables the receiver line-status interrupt

Bit 3: When set to 1, bit 3 enables the modem-status interrupt

Bits 4 through 7: These bits are not used (always set to 0)

interrupt identification register (IIR)

The UART has an on-chip interrupt generation and prioritization capability that permits a flexible interface with most popular microprocessors.

The UART provides four prioritized levels of interrupts:

Priority 1 – Receiver line status (highest priority)

Priority 2 – Receiver data ready or receiver character timeout

Priority 3 – Transmitter holding-register empty

Priority 4 – Modem status (lowest priority)

PRINCIPLES OF OPERATION

interrupt identification register (IIR) (continued)

When an interrupt is generated, the interrupt identification register indicates that an interrupt is pending and encodes the type of interrupt in its three least significant bits (bits 0, 1, and 2). The contents of this register are summarized in Table 2 and described in Table 4. Detail on each bit is as follows:

Bit 0: Used either in a hardware-prioritized or polled-interrupt system. When this bit is a reset to 0, an interrupt is pending; for a 1, no interrupt is pending.

Bits 1 and 2: The bits are used to identify the highest priority interrupt pending as indicated in Table 3

Bit 3: This bit is always 0 in the TL16C450 mode. In FIFO mode, this bit is set along with bit 2 to indicate that a timeout interrupt is pending.

Bits 4 through 5: These bits are not used (always reset at 0).

Bits 6 and 7: These bits are always reset to 0 in the TL16C450 mode. They are set when bit 0 of the FIFO-control register is equal to 1.

Table 4. Interrupt-Control Functions

INTERRUPT-IDENTIFICATION REGISTER				PRIORITY LEVEL	INTERRUPT TYPE	INTERRUPT SOURCE	INTERRUPT RESET METHOD
BIT 3	BIT 2	BIT 1	BIT 0				
0	0	0	1	None	None	None	None
0	1	1	0	1	Receiver line status	Overrun error, parity error, framing error, or break interrupt	Read the line-status register
0	1	0	0	2	Received data available	Receiver data available in the TL16C450 mode or trigger level reached in the FIFO mode	Read the receiver buffer register
1	1	0	0	2	Character time-out indication	No characters have been removed from or input to the receiver FIFO during the last four character times, and there is at least one character in it during this time	Read the receiver buffer register
0	0	1	0	3	Transmitter holding-register empty	Transmitter holding-register empty	Read the interrupt-identification register (if source of interrupt) or writing into the transmitter holding register
0	0	0	0	4	Modem status	Clear to send, data-set ready, ring indicator, or data-carrier detect	Read the modem-status register

FIFO control register (FCR)

The FIFO control register (FCR) is a write-only register at the same location as the IIR, which is a read-only register. The FCR enables and clears the FIFOs, sets the receiver FIFO trigger level, and selects the type of DMA signaling (see Table 5).

Bit 0: When set to 1, bit 0 enables the transmitter and receiver FIFOs. This bit must be set to 1 when other FCR bits are written to or they are not programmed. Changing this bit clears the FIFOs.

PRINCIPLES OF OPERATION

FIFO control register (FCR) (continued)

Bit 1: When set to 1, bit 1 clears all bytes in the receiver FIFO and resets its counter logic to 0. The shift register is not cleared. The one that is written to this bit position is self clearing.

Bit 2: When set to 1, bit 2 clears all bytes in the transmit FIFO and resets its counter to 0. The shift register is not cleared. The one that is written to this bit position is self clearing.

Bit 3: When FCR0 is set to 1, setting FCR3 to a 1 causes $\overline{\text{RXRDY}}$ and $\overline{\text{TXRDY}}$ to change from a 0 to a 1.

Bits 4 and 5: These bits are reserved for future use.

Bits 6 and 7: These bits are used to set the trigger level for the receiver FIFO interrupt.

Table 5. Bits 6 and 7 FCR

BIT 7	BIT 6	RECEIVER FIFO TRIGGER LEVEL (BYTES)
0	0	01
0	1	04
1	0	08
1	1	14

line control register (LCR)

The system programmer controls the format of the asynchronous data-communication exchange through the line-control register. In addition, the programmer is able to retrieve, inspect, and modify the contents of the line-control register; this eliminates the need for separate storage of the line characteristics in system memory. The contents of this register are summarized in Table 3 and described in the following paragraphs.

Bits 0 and 1: These two bits specify the number of bits in each transmitted or received serial character. These bits are encoded as shown in Table 6.

Table 6. Bits 0 and 1 LCR

BIT 1	BIT 0	WORD LENGTH
0	0	5 bits
0	1	6 bits
1	0	7 bits
1	1	8 bits

PRINCIPLES OF OPERATION

line control register (LCR) (continued)

Bit 2: This bit specifies either one, one and one-half, or two stop bits in each transmitted character. When bit 2 is reset to 0, one stop bit is generated in the data. When bit 2 is set to 1, the number of stop bits generated is dependent on the word length selected with bits 0 and 1. The receiver clocks only the first stop bit regardless of the number of stop bits selected. The number of stop bits generated in relation to word length and bit 2 are shown in Table 7.

Table 7. Bit 2 LCR

BIT 2	WORD LENGTH SELECTED BY BITS 1 AND 2	NUMBER OF STOP BITS GENERATED
0	Any word length	1
1	5 bits	1 1/2
1	6 bits	2
1	7 bits	2
1	8 bits	2

Bit 3: Parity enable bit. When bit 3 is set to 1, a parity bit is generated in transmitted data between the last data-word bit and the first stop bit. In received data, when bit 3 is set to 1, parity is checked. When bit 3 is reset to 0, no parity is generated or checked.

Bit 4: Even parity select bit. When parity is enabled by bit 3, a 1 in bit 4 produces even parity (an even number of 1s in the data and parity bits) and a 0 in bit 4 produces odd parity (an odd number of 1s).

Bit 5: Stick parity bit. When bits 3, 4, and 5 are set to 1s, the parity bit is transmitted and checked as a 0. When bits 3 and 5 are 1s and bit 4 is a 0, the parity bit is transmitted and checked as 1. When bit 5 is a 0, stick parity is disabled.

Bit 6: Break control bit. Bit 6 is set to 1 to force a break condition; i.e., a condition where SOUT is forced to the spacing (0) state. When bit 6 is reset to 0, the break condition is disabled and has no effect on the transmitter logic; it only affects SOUT.

Bit 7: Divisor latch access bit (DLAB). Bit 7 must be set to 1 to access the divisor latches of the baud generator during a read or write. Bit 7 must be reset to 0 during a read or write to access the receiver buffer, the transmitter holding register, or the interrupt-enable register.

modem control register (MCR)

The modem control register is an 8-bit register that controls an interface with a modem, data set, or peripheral device that is emulating a modem. The contents of this register are summarized in Table 3 and are described in the following paragraphs.

Bit 0 (DTR) controls the $\overline{\text{DTR}}$ output.

Bit 1 (RTS) controls the $\overline{\text{RTS}}$ output.

Bit 2 Has no effect on operation.

Bit 3 When MCR3 is set, the external serial channel interrupt is enabled.

When any of bits 0-3 is set to 1, the associated output is forced low; a bit value of 0 forces the associated output high.

PRINCIPLES OF OPERATION

modem control register (MCR) (continued)

Bit 4 (LOOP) provides a local loop-back feature for diagnostic testing of the UART. When LOOP is set to 1, the following occurs:

- The transmitter SOUT is set high.
- The receiver SIN is disconnected.
- The output of the transmitter shift register is looped back into the receiver shift-register input.
- The four modem control inputs ($\overline{CTS}$, $\overline{DSR}$, $\overline{DCD}$, and $\overline{RI}$) are disconnected.
- The two modem control outputs ($\overline{DTR}$, $\overline{RTS}$) are internally connected to the four modem control inputs.
- The four modem control outputs are forced to the inactive (high) levels.

Bit 5 (AFE) is the autoflow control enable. When set high the autoflow control, as described in the detailed description, is enabled.

In the diagnostic mode, data that is transmitted is immediately received. This allows the processor to verify the transmit and receive data paths to the UART. The receiver and transmitter interrupts are fully operational. The modem control interrupts are also operational, but the modem control interrupt sources are now the lower four bits of the modem control register instead of the four modem control inputs. All interrupts are still controlled by the interrupt enable register.

The UART flow can be configured by programming bits 1 and 5 of the MCR as shown in Table 8.

Table 8. UART Flow

MCR BIT 5 (AFE)	MCR BIT 1 (RTS)	UART FLOW CONFIGURATION
1	1	Auto-$\overline{RTS}$ and auto-$\overline{CTS}$ enabled (autoflow control enabled)
1	0	Auto-$\overline{CTS}$ only enabled
0	X	Auto-$\overline{RTS}$ and auto-$\overline{CTS}$ disabled

line status register (LSR)

The line status register provides information to the CPU concerning the status of data transfers. The contents of this register are described below and summarized in Table 3. The line status register is intended for read operations only; writing to this register is not recommended outside of a factory testing environment. Bits 1–4 are the error conditions that produce a receiver line status interrupt.

Bit 0: Data ready (DR) indicator for the receiver. DR is set to 1 whenever a complete incoming character has been received and transferred into the receiver buffer register or the FIFO. DR is reset to 0 by reading all of the data in the receiver buffer register or the FIFO.

Bit 1: Overrun error (OE) indicator. When OE is set to 1, it indicates that before the character in the receiver buffer register was read, it was overwritten by the next character transferred into the register. OE is reset every time the CPU reads the contents of the line status register. If the FIFO mode data continues to fill the FIFO beyond the trigger level, an overrun error occurs only after the FIFO is full and the next character has been completely received in the shift register. An overrun error is indicated to the CPU as soon as it happens. The character in the shift register is overwritten, but it is not transferred to the FIFO.

PRINCIPLES OF OPERATION

line status register (LSR) (continued)

Bit 2: Parity error (PE) indicator. When PE is set to 1, it indicates that the parity of the received data character does not match the parity selected in the line control register (bit 4). PE is reset every time the CPU reads the contents of the line status register. In the FIFO mode, this error is associated with the particular character in the FIFO to which it applies. This error is revealed to the CPU when its associated character is at the top of the FIFO.

Bit 3: Framing error (FE) indicator. When FE is set to 1, it indicates that the received character did not have a valid (1) stop bit. FE is reset every time the CPU reads the contents of the line status register. In the FIFO mode, this error is associated with the particular character in the FIFO to which it applies. This error is revealed to the CPU when its associated character is at the top of the FIFO. The UART tries to resynchronize after a framing error. To accomplish this, it is assumed that the framing error is due to the next start bit. The UART samples this start bit twice and then accepts the input data.

Bit 4: Break interrupt (BI) indicator. When BI is set to 1, it indicates that the received data input was held in the logic low state for longer than a full-word transmission time. A full-word transmission time is defined as the total time to transmit the start, data, parity, and stop bits. BI is reset every time the CPU reads the contents of the line status register. In the FIFO mode, this error is associated with the particular character in the FIFO to which it applies. This error is revealed to the CPU when its associated character is at the top of the FIFO. When a break occurs, only one 0 character is loaded into the FIFO. The next character transfer is enabled after SIN goes to the marking state for at least two RCLK samples and then receives the next valid start bit.

Bit 5: Transmitter holding-register-empty (THRE) indicator. THRE is set to 1 when the transmitter holding register is empty, indicating that the UART is ready to accept a new character. If the THRE interrupt is enabled when THRE is set to 1, an interrupt is generated. THRE is set to 1 when the contents of the transmitter holding register are transferred to the transmitter shift register. THRE is reset to 0 concurrent with the loading of the transmitter holding register by the CPU. In the FIFO mode, THRE is set when the transmit FIFO is empty; it is cleared when at least one byte is written to the transmit FIFO.

Bit 6: Transmitter empty (TEMT) indicator. TEMT bit is set to 1 when the transmitter holding register and the transmitter shift register are both empty. When either the transmitter holding register or the transmitter shift register contains a data character, TEMT is reset to 0. In the FIFO mode, TEMT is set to 1 when the transmitter FIFO and shift register are both empty.

Bit 7: In the TL16PIR552, this bit is always reset to 0. In the TL16C450, this bit is always a 0. In the FIFO mode, LSR7 is set to 1 when there is at least one parity, framing, or break error in the FIFO. It is cleared when the microprocessor reads the LSR and there are no subsequent errors in the FIFO.

modem status register (MSR)

The modem status register is an 8-bit register that provides information about the current state of the control lines from the modem, data set, or peripheral device to the CPU. Additionally, four bits of this register provide change information; when a control input from the modem changes state, the appropriate bit is set to 1. All four bits are reset to 0 when the CPU reads the modem status register. The contents of this register are summarized in Table 3 and are described in the following paragraphs.

Bit 0: Change in clear to send (ΔCTS) indicator. ΔCTS indicates that the $\overline{\text{CTS}}$ input has changed state since the last time it was read by the CPU. When ΔCTS is set to 1 (autoflow control is not enabled and the modem status interrupt is enabled), a modem status interrupt is generated. When autoflow control is enabled, no interrupt is generated.

Bit 1: Change in data set ready (ΔDSR) indicator. ΔDSR indicates that the $\overline{\text{DSR}}$ input has changed state since the last time it was read by the CPU. When ΔDSR is set to 1 and the modem status interrupt is enabled, a modem status interrupt is generated.

PRINCIPLES OF OPERATION

modem status register (MSR) (continued)

Bit 2: Trailing edge-of-the-ring indicator (TERI) detector. TERI indicates that the $\overline{RI}$ input to the chip has changed from a low to a high level. When TERI is set to 1 and the modem status interrupt is enabled, a modem status interrupt is generated.

Bit 3: Change in data carrier detect (ΔDCD) indicator. ΔDCD indicates that the $\overline{DCD}$ input to the chip has changed state since the last time it was read by the CPU. When ΔDCD is set to 1 and the modem status interrupt is enabled, a modem status interrupt is generated.

Bit 4: Complement of the clear to send ($\overline{CTS}$) input. When the UART is in the diagnostic test mode (LOOP [MCR4] = 1), this bit is equal to the modem control register bit 1 (RTS).

Bit 5: Complement of the data set ready ($\overline{DSR}$) input. When the UART is in the diagnostic test mode (LOOP [MCR4] = 1), this bit is equal to the modem control register bit 0 (DTR).

Bit 6: Complement of the ring indicator ($\overline{RI}$) input. When the UART is in the diagnostic test mode (LOOP [MCR4] = 1), this bit is equal to the modem control register bit 2 (OUT1).

Bit 7: Complement of the data carrier detect ($\overline{DCD}$) input. When the UART is in the diagnostic test mode (LOOP [MCR4] = 1), this bit is equal to the modem control register bit 3 (OUT2).

scratch register (SCR)

The scratch register is an 8-bit register that is intended for programmer use as a scratchpad in the sense that it temporarily holds the programmer data without affecting any other UART operation.

prescaler

When DLAB = 1, the six least significant bits of the scratch register contain the prescaler value. IR logic is selected when the seventh bit is set to 1. Otherwise UART output is selected. After reset, UART0 is in IR mode and UART1 is in UART mode.

prescaler descriptions

The clock prescaler allows for the divisor from 0 to 31.5 in 0.5 increments (scr(0) is the half-bit divider). The divisor value is loaded from scratch register with DLAB = 1. The output of the divisor feeds the UART clock. A programmed divisor between 2 and 7.5 drives the UART clock low for one XIN clock cycle for integer divisor and 1.5 XIN clock cycles for integer-plus a half clock divisor. A programmed divisor of eight or greater drives the UART clock low for four XIN clock cycles for integer divisors and 4.5 XIN clock cycles for integer-plus-a-half divisor. Based on the above parameters, the acceptable XIN/divisor combinations can be derived. The precision of the programmable clock generator for integer-plus-a-half divisor depends on the closeness to a 50% duty cycle for the XIN input clock.

Example: When the oscillator frequency is 22 Mhz (see Table 9).

Table 9. Typical Prescaler Example

XIN	PRESCALER DIVISOR	UART CLOCK	UART DIVISOR	BAUD RATE
22.118 Mhz	5.5	4 Mhz	8	31.25k
22.118 Mhz	3	7.33 Mhz	1	458K

PRINCIPLES OF OPERATION

prescaler descriptions: (continued)

When DLAB = 1, the six least significant bits of the scratch register contain the prescaler value.

Table 10. SCR (0–5) Values

SCR(0-5) Value (Hex)	Result
0 (0)	No Clock (high)
0.5 (1)	divide-by-1
1 (2)	divide-by-1
1.5 (3)	divide-by-1
2 (4) to 31.5 (3F)	divide by 2 to 31.5

programmable baud generator

The UART contains a programmable baud generator that takes a clock input in the range between dc and 16 MHz and divides it by a divisor in the range between 1 and 2^{16}–1. The output frequency of the baud generator is 16 times (16×) the baud rate. The formula for the divisor is:

divisor = XIN frequency input ÷ (desired baud rate × 16)

Two 8-bit registers, called divisor latches, store the divisor in a 16-bit binary format. These divisor latches must be loaded during initialization of the UART in order to ensure desired operation of the baud generator. When either of the divisor latches is loaded, a 16-bit baud counter is also loaded to prevent long counts on initial load.

Tables 5 and 6 illustrate the use of the baud generator with crystal frequencies of 1.8432 MHz and 3.072 MHz, respectively. For baud rates of 38.4 kbits/s and below, the error obtained is very small. The accuracy of the selected baud rate is dependent on the selected crystal frequency (refer to Figure 16 for examples of typical clock circuits).

FIFO interrupt mode operation

When the receiver FIFO and receiver interrupts are enabled (FCR0 = 1, IER0 = 1, IER2 = 1), a receiver interrupt occurs as follows:

- The received-data-available interrupt issued to the microprocessor when the FIFO has reached its programmed trigger level. It is cleared when the FIFO drops below its programmed trigger level.
- The IIR receive-data-available indication also occurs when the FIFO-trigger level is reached, and like the interrupt, it is cleared when the FIFO drops below the trigger level.
- The receiver line-status interrupt (IIR = 06) has higher priority than the received-data-available (IIR = 04) interrupt.
- The data ready bit (LSR0) is set when a character is transferred from the shift register to the receiver FIFO. It is reset when the FIFO is empty.

PRINCIPLES OF OPERATION

FIFO interrupt mode operation (continued)

When the receiver FIFO and receiver interrupts are enabled:

- FIFO timeout interrupt occurs when the following conditions exist:
  - At least one character is in the FIFO.
  - The most recent serial character was received more than four continuous character times ago (if two stop bits are programmed, the second one is included in this time delay).
  - The most recent microprocessor read of the FIFO occurred more than four continuous character times ago. This causes a maximum character received to interrupt an issued delay of 160 ms at 300 baud with a 12-bit character.
- Character times are calculated by using the RCLK input for a clock signal that makes the delay proportional to the baud rate.
- When a timeout interrupt has occurred, it is cleared and the timer is reset when the microprocessor reads one character from the receiver FIFO.
- When a timeout interrupt has not occurred, the timeout timer is reset after a new character is received or after the microprocessor reads the receiver FIFO.

When the transmitter FIFO and THRE interrupt are enabled (FCR0 = 1, IER1 = 1), transmit interrupts occur as follows:

- The occurrence of transmitter holding-register-empty interrupt (IIR(3–0) = 2) is delayed one character time minus the last stop bit time when there have not been at least two bytes in the transmitter FIFO at the same time since the last time the transmitter FIFO was empty. It is cleared (IIR(3–0) = 1) as soon as the transmitter holding register is written to (1 to 16 characters may be written to the transmit FIFO while servicing this interrupt) or the IIR is read. The first transmitter interrupt after changing FCR is immediate if it is enabled.
- The transmitter empty indicator (LSR6 (TEMT) = 1) is delayed one character time when there has not been at least two bytes in the transmitter FIFO at the same time since the last time that TEMT = 1. TEMT is set after the stop-bit has been completely shifted out (finishes one complete bit time or 16 BAUDOUT cycles).
- The transmitter FIFO empty indicator (LSR5 (THRE) = 1) works the normal way in this mode and is not delayed.

Character timeout and receiver FIFO trigger-level interrupts have the same priority as the current received-data-available interrupt.

PRINCIPLES OF OPERATION

FIFO polled mode operation

With FCR0 = 1 (transmitter and receiver FIFOs enabled), resetting IER0, IER1, IER2, IER3, or all four to 0 puts the UART in the FIFO polled mode of operation. Since the receiver and transmitter are controlled separately, either one or both can be in the polled mode of operation.

In this mode, the user program checks receiver and transmitter status using the LSR. As stated previously:

- LSR0 is set as long as there is one byte in the receiver FIFO.
- LSR (1–4) specify which error(s) have occurred. Character error status is handled the same way as when in the interrupt mode; the IIR is not affected since IER2 = 0.
- LSR5 indicates when the transmitter holding register is empty.
- LSR6 indicates that both the transmitter holding register and transmitter shift register are empty.
- LSR7 indicates whether there are any errors in the receiver FIFO.

There is no trigger level reached or time-out condition indicated in the FIFO-polled mode. However, the receiver and transmitter FIFOs are still fully capable of holding characters (See Table 11 and 12).

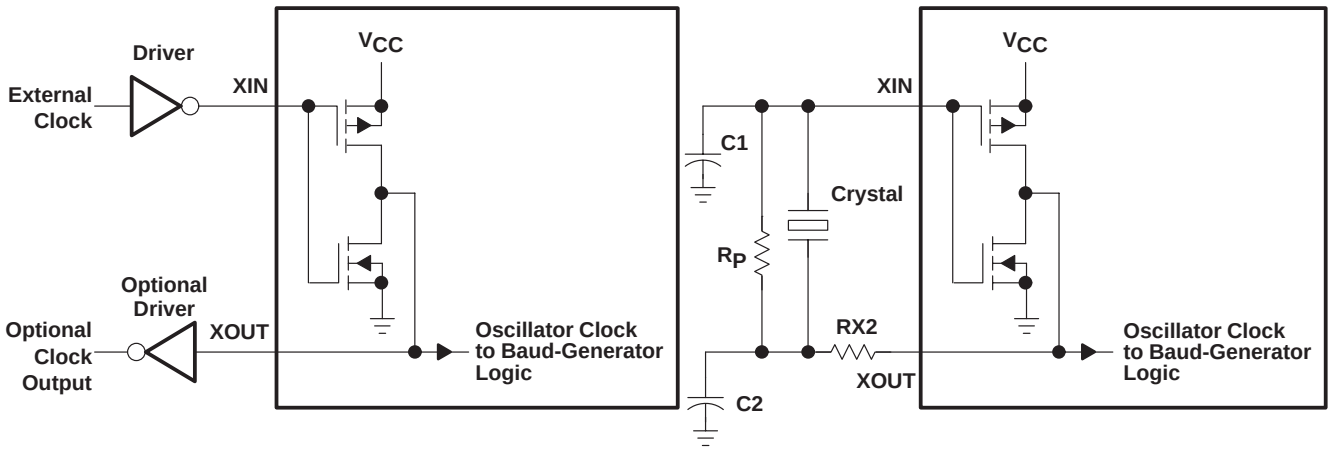
Table 11. Baud Rates Using a 1.8432-MHz Crystal

DESIRED BAUD RATE	DIVISOR USED TO GENERATE 16× CLOCK	PERCENT OF ERROR DIFFERENCE BETWEEN DESIRED AND ACTUAL BAUD RATES
50	2304	
75	1536	
110	1047	0.026
134.5	857	0.058
150	768	
300	384	
600	192	
1200	96	
1800	64	
2000	58	0.69
2400	48	
3600	32	
4800	24	
7200	16	
9600	12	
19200	6	
38400	3	
56000	2	2.86

PRINCIPLES OF OPERATION

Table 12. Baud Rates Using a 3.072-MHz Crystal

DESIRED BAUD RATE	DIVISOR USED TO GENERATE 16× CLOCK	PERCENT OF ERROR DIFFERENCE BETWEEN DESIRED AND ACTUAL BAUD RATES
50	3840	
75	2560	
110	1745	0.026
134.5	1428	0.034
150	1280	
300	640	
600	320	
1200	160	
1800	107	0.312
2000	96	
2400	80	
3600	53	0.628
4800	40	
7200	27	1.23
9600	20	
19200	10	
38400	5	



TYPICAL CRYSTAL OSCILLATOR NETWORK

CRYSTAL	R_P	RX2	C1	C2
3.072 MHz	1 MΩ	1.5 kΩ	10–30 pF	40–60 pF
1.8432 MHz	1 MΩ	1.5 kΩ	10–30 pF	40–60 pF

Figure 26. Typical Clock Circuits

PRINCIPLES OF OPERATION

IR encoder and decoder

The IR encoder and decoder circuitry work with the UART to change the serial bit stream into a series of pulses and back again. For every zero bit in the serial stream, a pulse is sent at the middle of the bit with a duration of 3/16 of a bit time. If a one or series of ones is sent, the encode does not send a pulse. The decoding process consists of receiving a pulse and sending a stretched version of the pulse to the UART. The stretched version must be at least three-fourths of a bit time to be correctly decoded by most UARTs. Because the serial stream can occur at any baud rate, some means of changing the encoding and decoding baud rate is needed. The easiest way to accomplish this is to clock the encoder and decoder circuits with the UART baud rate 16x clock. A block diagram of a design is shown in Figure 27.

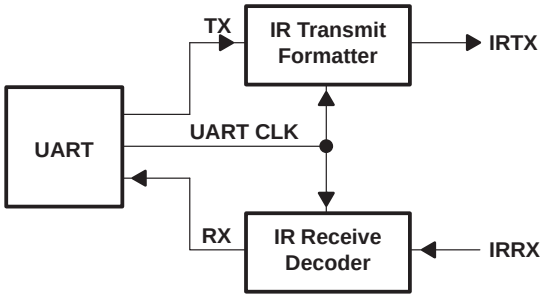


Figure 27. Conventional UART Connection to the External IR Encoder and Decoder

The encoder uses a 4-bit synchronous counter to transmit the pulses. It uses a delay of one-half of a bit time to insure that a true zero bit has been sent. It sends a pulse for three UART clocks in the middle of the zero bit. When two zeros are sent in a row, the counter simply overruns at the beginning of the next zero bit and starts the process over. When a 1 is sent, the counter is reset.

The decoder also uses a 4-bit synchronous counter including a synchronous reset. The rising edge of an incoming pulse sets the input flip-flop. This causes the counter to begin counting and a zero to be sent out at the RX signal output. When the counter gets to 16, the input flip-flop is reset to wait for another input pulse. If another pulse occurs before the counter expires, the counter is synchronously reset and the RX signal output remains 0. When the counter expires, the RX signal output returns to 1, and the circuit waits for another input pulse.

PRINCIPLES OF OPERATION

IR encoder and decoder (continued)

IR mode can be chosen through the use of SLR(6) with DLAB = 1. With DLAB = 1 and if SCR(6) = 1, IR mode is selected. Built in multiplexers are used to choose the correct signal to input to the UART. Refer to Figure 28 for more detail.

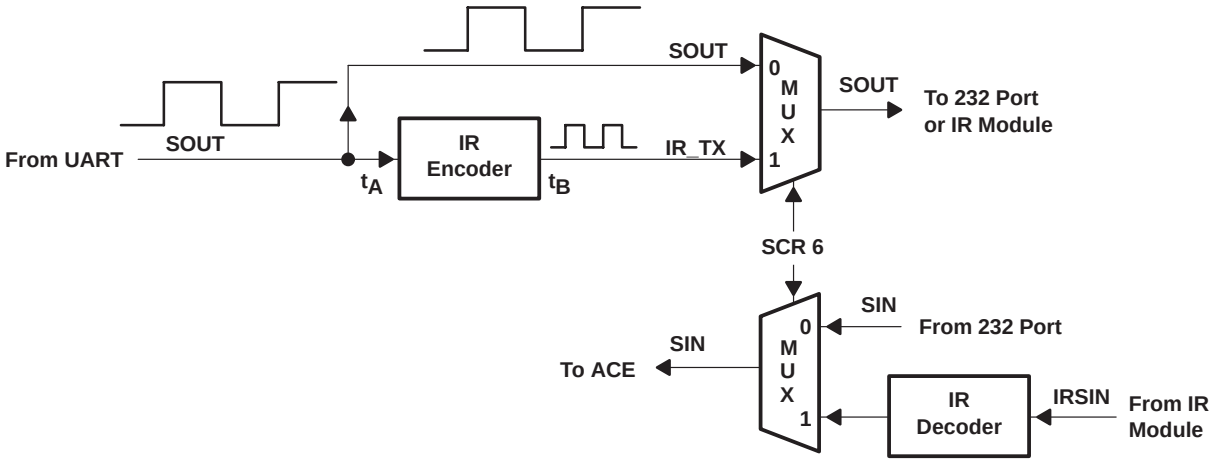


Figure 28. TL16PIR552 Mode Select

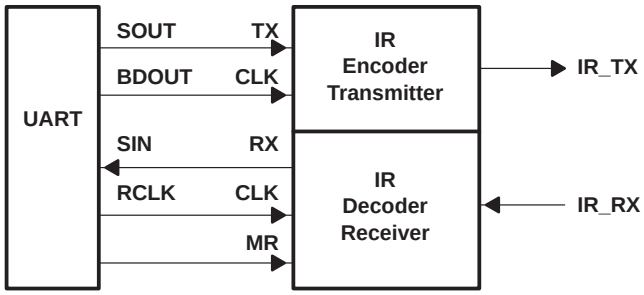


Figure 29. TL16PIR552 IR Encoder/Decoder

For encoding, the IR_TX is selected when SCR6 is set to 1. This sends IR_TX on the device SOUT terminal. This signal can be used as an input to any IR transceiver (see Figure 29).

For decoding, the IRSIN from the device (this can be connected to any IR transceiver) goes through the IR decode block and then it is the input to the UART (see Figure 29).

PRINCIPLES OF OPERATION

parallel port (see Table 13)

The parallel port is essentially an extended capabilities port with an additional enhanced parallel port mode, and includes the following features:

- Compatible with standard Centronics parallel interface
- Support for ECP and EPP
- Data path FIFO buffer
- Direct Memory Access (DMA) transfer
- Decompression of run length encoded data in ECP reverse mode

The parallel port is an extended capabilities parallel (ECP) port with additional enhanced parallel port (EPP) protocol support. Modes 000 and 001 are compatible with Centronics and bidirectional Centronics ports, and mode 100 is defined to be EPP mode. Thus, together with the ECP protocol modes, the parallel port supports three distinct transfer protocols which all share the standard parallel port signals.

The parallel port consists of an 8-bit host interface (including DMA support) which is connected to the fast-AT bus, a sequencer containing state-machines for the three different protocols, a 16-byte FIFO data path and a parallel interface.

transfer protocols

There are three parallel port transfer protocols in the TL16PIR552. Descriptions of these protocols follow and specifications with terminal numbers follow in Table 13.

standard Centronics protocol

In the standard Centronics modes, the parallel port is compatible with the Centronics unidirectional or bidirectional parallel port. It consists of a single-byte data port which writes and reads data to/from the port data lines and registers to control and reflect the status of the parallel port signals. Signaling protocol is handled by software, which must assert control strobes and poll for acknowledgement itself.

enhanced parallel port (EPP protocol)

In enhanced parallel port mode, $\overline{\text{SELECTIN}}$ and $\overline{\text{AUTOFD}}$ are automatically generated and are redefined to be address strobe and data strobe, respectively, while $\overline{\text{STROBE}}$ indicates a write or a read cycle. Additional I/O addresses are defined for data and address accesses and when these locations are used, handshaking is performed automatically by hardware.

extended capabilities port (ECP) protocol

The enhanced capability port specification is an enhancement to the IEEE 1284 standard; it defines new transfer protocols and timing which offer a reverse channel as fast as the forward channel. Software overhead is reduced by direct memory access (DMA) support, data buffering, and automatic strobe generation. ECP defines separate I/O locations for address and data accesses.

PRINCIPLES OF OPERATION

Table 13. Parallel Port Connector Specifications

Terminal Number	Standard	EPP	ECP
56	$\overline{\text{STROBE}}$	$\overline{\text{WRITE}}$	HOSTCLK
52-49, 44-47	PD0–7	PD0–7	PD0–7
27	$\overline{\text{ACK}}$	INTR	PERIPHCLK
26	BUSY	$\overline{\text{WAIT}}$	PERIPHACK
23	PERROR	USER DEFINED	MPERIPHREQUEST
24	SELECT	USER DEFINED	USER DEFINED
57	$\overline{\text{AUTOFD}}$	$\overline{\text{DSTRB}}$	HOSTACK
25	$\overline{\text{FAULT}}$	USER DEFINED	$\overline{\text{PERIPHREQUEST}}$
54	$\overline{\text{INIT}}$	$\overline{\text{INIT}}$	$\overline{\text{REVERSEREQUEST}}$
55	$\overline{\text{SELECTIN}}$	$\overline{\text{ASTRB}}$	$\overline{\text{SELECTIN}}$(1,3)

NOTE 7: For the cable interconnection required for ECP support and the slave connector terminal numbers, refer to the *Extended Capabilities Port Protocol and ISA Standard*, Rev. 1.09, Jan. 7, 1993. This document is available from Microsoft®.

parallel port modes of operation (see Table 14)

The seven parallel port operating modes are selected by bits 7-15 of the extended control register.

1. Standard Centronics mode (000)

This is the default mode in which the parallel port behavior is compatible with the Centronics standard port. The FIFO is reset and the direction bit in the device control register has no effect.

2. Bidirectional Centronics standard mode (001)

This is the same as mode 000 except that setting the direction bit puts the data line in a high-impedance state and reading the data register returns the value on the data lines.

3. Parallel port FIFO mode (010)

In this mode bytes written or DMA transferred to the FIFO are transmitted automatically using the Centronics standard protocol. Only the forward direction is useful.

4. ECP mode (011)

In the forward direction (direction = 0) data written to the ECPDFIFO and bytes written to ECPAFIFO addresses are placed in a single FIFO and transmitted automatically using ECP protocol. In the reverse direction (direction = 1) data bytes are transferred from the ECP parallel port and placed in the ECPDFIFO.

5. Enhanced parallel port mode (100)

In this mode, EPP read, write, or address/data cycles can be executed or, if no EPP cycle is pending, compatible Centronics standard access can be made (as in mode 001). Note that the software must ensure that the direction = 0 before attempting to perform an EPP write cycle.

6. FIFO test mode (110)

In this mode the FIFO can be written and read or DMA transferred in any direction, but no data will be transmitted on the parallel port. The FIFO does not stop accepting or sending data when full or empty conditions occur; FIFO read and write address counters will wrap.

7. Configuration mode (111)

In this mode the CONFGA and CONFGB registers are accessible.

PRINCIPLES OF OPERATION

Table 14. Mode Summary

MODE	DESCRIPTION
000	Centronics (default) mode, forward direction only
001	Bidirectional Centronics mode
010	Parallel port data FIFO mode
011	ECP Parallel port mode
100	EPP mode
110	Test mode
111	Configuration mode

mode switching

Mode switching is only allowed into and out of the modes 000 and 001. All 1284 negotiation takes place in these two modes. Setting the mode to 011 (ECP Parallel Port mode) causes the hardware to initiate data transfer. Switching out of modes 011 and 010 in the middle of a transfer or when data remains in the FIFO causes the transfer to be aborted and the data to be lost.

data compression

The TL16PIR552 supports run length encoded (RLE) decompression in hardware and can transfer compressed data to a peripheral. Run length encoded (RLE) compression in hardware is not supported. To transfer compressed data in ECP mode, the compression count is written to the ECPAFIFO and a data byte is written to the ECPDFIFO.

Compression is accomplished by counting identical bytes and transmitting an RLE byte that indicates how many times the next byte is to be repeated. Decompression simply intercepts the RLE byte and repeats the following byte the specified number of times. When a run-length count is received from a peripheral, the subsequent data byte is replicated that specified number of times. A run-length count of zero specifies that only one byte of data is represented by the next data byte. A run-length count of 127 indicates that the next byte should be expanded to 128 bytes. To prevent data expansion, however, run-length counts of zero should be avoided.

The RLE command byte is loaded to the FIFO. Then the logic offloads this command to the decompression counter to generate the correct number of decompressed data available. The logic asserts PDRQ only when there are data bytes in the FIFO. Although the command byte is loaded to the FIFO, the PDRQ is not asserted.

PRINCIPLES OF OPERATION

register definitions

Table 15 is a summary of the parallel port internal registers, assuming 378h is the base address.

Table 15. Parallel port internal register

Chip Select	A2–A0	Mode	Access	Name	Function
$\overline{\text{PPCS}}$	000	Std/Bidi (000–001)	R/W	DATA	Data port
	000	ECP (011)	W	ECPAFIFO	ECP address FIFO
	001	All	R	DSR	Status register
	010	All	R/W	DCR	Control register
	011	EPP (100)	R/W	EPPADDR	EPP address port
	100–111	EPP (100)	R/W	EPPDATA	EPP data Port
$\overline{\text{ECPCS}}$	000	ECP (011)	R/W	ECPDFIFO	ECP data FIFO
	000	CFIFO (010)	R/W	PP DATA FIFO	Parallel port data FIFO
	000	TFIFO (110)	R/W	TEST FIFO	
	000	Config (111)	R	CNFGA	Configuration register A
	001	Config (111)	R	CNFGB	Configuration register B
	010	All	R/W	ECR	Extended control register

data register (DATA)

This is the standard parallel port data register. In standard mode, writing to this register drives data onto the parallel port data lines. In all other modes, the drivers may be put into a high-impedance state by setting the direction bit in the DCR. Reads to this register return the value of the data lines.

ECP address FIFO register (ECPAFIFO)

A data byte written to this register is placed in the FIFO and tagged as an ECP address. Table 16 is a summary of the device status register bits and their descriptions. Table 17 is a summary of the device control register bits and their descriptions.

Table 16. Device status register (DSR)

Bit	Default	Name	Description
7	Default	$\overline{\text{BUSY}}$	Bit 7 corresponds to the inverse of a BUSY input
6	–	$\overline{\text{ACK}}$	Bit 6 corresponds to the $\overline{\text{ACK}}$ input
5	–	PE	Bit 5 corresponds to the PERROR input
4	–	SLCT	Bit 4 corresponds to the SELECT input
3	–	$\overline{\text{ERR}}$	Bit 3 corresponds to the $\overline{\text{FAULT}}$ input
2	–	$\overline{\text{PRINT}}$	Print interrupt. Bit 2 is reset to 0 by the rising transition of $\overline{\text{ACK}}$ and set to 1 by a read operation of this register.
1	0	–	Bit 1 is reserved
0	0	TIMEOUT	When enabled in EPP mode, bit 0 is set to 1 when 10-µs timeout occurs. Bit 0 is cleared by any write operation to the DSR register.

PRINCIPLES OF OPERATION

Table 17. Device Control Register (DCR)

Bit	Default	Name	Description
7	1	–	This bit is reserved
6	1	–	This bit is reserved
5	0	DIR	In standard and CFIFO modes this bit has no effect; in all other modes, when set to 1 this bit put the parallel port data lines into a high-impedance state, 0 = forward, 1 = reverse.
4	0	INT2EN	When high, this bit enables interrupts to access the host (on the rising edge of $\overline{\text{ACK}}$).
3	0	SLIN	1: $\overline{\text{SELECT}}$ output active; 0: $\overline{\text{SELECT}}$ output inactive.
2	0	$\overline{\text{INIT}}$	0: $\overline{\text{INIT}}$ output active; 1: $\overline{\text{INIT}}$ output inactive.
1	0	AFD	1: $\overline{\text{AUTOFD}}$ output active; 0 AUTOFD output inactive.
0	0	STB	1: $\overline{\text{STROBE}}$ output active; 0: $\overline{\text{STROBE}}$ output inactive.

EPP address register (EPPADDR)

This is the EPP address strobe register. In EPP mode, an address strobe is automatically generated when data is read from or written to this register. This register is only available in EPP mode.

EPP data register (EPPDATA)

This is the EPP data strobe register. In EPP mode, a data strobe is automatically generated when data is read or written to this register. This register is only available in EPP mode.

ECP Data FIFO register (ECPDFIFO)

A data byte wirtten or DMA transferred to this register is placed in the FIFO and tagged as ECP data in forward direction. Data bytes from peripherals are read under an automatic hardware handshake from ECP into this FIFO when the direction bit is set to 1. Table 18 is a description of the bits in the ECP configuration, register A. Table 19 is a description of the bits in the ECP configuration, register B. Table 20 is a description of the bits in the extended control register.

Table 18. ECP Configuration Register A (CNFGA)

Bit	Name	Description
7–4	IMPLD	Bits 7–4 are used for the implementation ID number. Always read as 0001, 8-bit implementation (PWord = 1 byte)
3–0	–	These bits are reserved; read as 1s

Table 19. Configuration Register B (CNFGB)

Bit	Name	Description
7	COMPRESS	Bit 7 always read as a 0: Compression is not supported
6	INTRVALUE	Bit 6 returns value on IRQ line to determine possible conflicts
5–3	INTRLINE	Bits 5–3 are always read as 001: IRQ7 selected
2–0	DMACHNL	Bits 2–0 are always read as 011: DMA channel 3 selected

PRINCIPLES OF OPERATION

Table 20. Extended Control Register (ECR)

Bit	Access	Name	Description
7–5	R/W	MODE	Bits 7–5 have the following selected modes of operation: 000: Standard Parallel Port mode (forward direction only) 001: Bidirectional Parallel Port mode 010: Parallel Port FIFO mode (forward direction only) 011: ECP Parallel Port mode 100: EPP mode 101: reserved 110: FIFO test mode 111: Configuration mode
4	R/W	$\overline{\text{ERRINTREN}}$	In ECP mode, when bit 4 is set to 0, an interrupt on the falling edge of $\overline{\text{FAULT}}$ is enabled. A 1 disables the interrupt.
3	R/W	DMAEN	When bit 3 is reset to 0, DMA is disabled. A bit setting of 1 enables the DMA. The DMA starts when SERVICEINTR is reset to 0.
2	R/W	SERVICEINTR	When bit 2 is set to 1, DMA and all service interrupts are disabled.
1	R	FULL	When bit 1 is set to 1, the FIFO is full.
0	R	EMPTY	When bit 0 is set to 1, the FIFO is empty.

CFIFO parallel port data FIFO

Bytes written or DMA transferred from the system to this FIFO are transmitted by a hardware handshake to the peripheral using the standard parallel port protocol. This mode is only defined for the forward direction.

TFIFO test mode

Data bytes can be read, written, or DMA transferred to or from the system to this FIFO in any direction. Data in the TFIFO register is not transmitted to the parallel port lines using a hardware protocol handshake. However, data in the TFIFO may be displayed on the parallel port data lines. The TFIFO does not stall when overwritten or underrun. Data is simply rewritten or overrun. The FULL and EMPTY bits must always keep track of the correct FIFO state. The TFIFO transfers data at the maximum ISA rate so that software may generate performance metrics.

The WRITEINTR threshold can be determined by starting with a full TFIFO, and emptying it one byte at a time until the SERVICEINTR bit is set. This may generate a spurious interrupt, but indicates that the threshold has been reached. Likewise, READINTR threshold can be determined by setting the direction bit to 1, and filling the empty TFIFO a PWord at a time until SERVICEINTR bit is set. Data bytes are always read from the head of TFIFO regardless of the value of the direction bit. For example, when a 0x4433, 0x2211, or a 0x00ff is written to the FIFO, then reading the TFIFO returns a 0x4433, a 0x2211,or a 0x0ff in the same order in which it was written. The FIFO size and interrupt threshold can be determined by writing PWords and checking the FULL and SERVICEINTR bits.

PRINCIPLES OF OPERATION

description of printer operation

ECP: command/data

The ECP mode supports two advanced features to improve the effectiveness of the protocol for some applications. The features are implemented by allowing the transfer of normal 8-bit data or 8-bit commands.

When in the forward direction, normal data is transferred when $\overline{\text{AUTOFD}}$ is high and an 8-bit command is transferred when $\overline{\text{AUTOFD}}$ is low.

The most significant bit of the command indicates whether it is a run-length count (for compression) or a channel address.

When in the reverse direction, normal data is transferred when BUSY is high and an 8-bit command is transferred when BUSY is low. The most significant bit of the command is always zero. Reverse channel addresses are seldom used and may not be supported in hardware.

Table 21. Bits 6 and 7 of the ECP Command

D7	D6
0	Run-length count (0–127)
1	Channel Address (0–127)

run length encoded data compression

The parallel port supports decompression of run length encoded (RLE) data in ECP DMA mode (011) reverse direction only. During reverse direction transfers, the peripheral indicates a command byte is to be transferred by setting PERIPHACK (BUSY) low. Bits 6-0 of the command byte indicate the number of times the next data byte should be replicated; bit 7 is always zero.

interrupts

The interrupts are enabled by the SERVICEINTR bit in the ECR register. When SERVICEINTR bit is a 1 the DMA and all of the service interrupts are disabled. When SERVICEINTR bit is a 0 the selected interrupt condition is enabled. When the interrupting condition is valid, then the interrupt is generated immediately when this bit is changed from a 1 to a 0. This can occur during the programmed I/O when the number of bytes removed or added from/to the FIFO does not cross the threshold.

The interrupt generated is ISA friendly in that it must pulse the interrupt line low, allowing for interrupt sharing. After a brief pulse low following the interrupt event, the interrupt line is put into a high-impedance state so that other interrupts may assert.

An interrupt is generated:

1. For DMA transfers: When SERVICEINTR is 0, DMAEN is 1, and the DMA TC is received.
2. For programmed I/O:
   a. When the SERVICEINTR bit is reset to 0, DMAEN is set to 0, DIRECTION is set 0, and there are 12 or more free bytes in the FIFO. Also, an interrupt is generated when SERVICEINTR bit is cleared to 0 whenever there are 12 or more free bytes in the FIFO.
   b. When the SERVICEINTR bit is reset to 0, DMAEN is set to 0, DIRECTION is set to1 and there are 12 or more bytes in the FIFO. Also, an interrupt is generated when SERVICEINTR bit is cleared to 0 whenever there are 12 or more byte in the FIFO.

PRINCIPLES OF OPERATION

3. When $\overline{\text{ERRINTREN}}$ is 0 and $\overline{\text{FAULT}}$ transitions from 1 to 0 or when $\overline{\text{ERRINTREN}}$ is reset from 1 to 0 and $\overline{\text{FAULT}}$ is asserted, an interrupt is generated.
4. When the ACKINTEN is 1 and the $\overline{\text{ACK}}$ signal transitions from 0 to 1, an interrupt is generated.

FIFO operation

When the FIFO threshold is set to 12 all data transfers to or from the parallel port can proceed in the DMA or programmed I/O (non-DMA) mode is indicated by the selected mode. The FIFO is used by selecting the parallel port FIFO mode or ECP parallel port mode. After a reset, the FIFO is disabled. Each data byte is transferred by a programmed I/O cycle or PDRQ depending on the selection of DMA or programmed I/O mode.

DMA transfers

DMA transfers are always to or from the ECPDFIFO, TFIFO or CFIFO registers. DMA utilizes the standard PC DMA transfers, the host first sets up the direction and state as in the programmed I/O case. It then programs the DMA controller in the host with the desired count and memory address. DMAEN is set to 1 and SERVICEINTR is reset to 0. The ECP requests the DMA transfers from the host by activating the PDRQ terminal. The DMA empties or fills the FIFO using the appropriate direction and mode. When the terminal count in the DMA controller is reached, an interrupt is generated and SERVICEINTR is asserted, disabling the DMA. The FIFO is enabled directly by asserting $\overline{\text{PDACK}}$ and addresses need not be valid. PINTR is generated when a TC is received. (Note: The only way to properly terminate DMA transfers is with a TC request.)

The DMA may be disabled in the middle of a transfer by first disabling the host DMA controller and setting SERVICEINTR to 1, followed by resetting DMAEN to 0, and waiting for the FIFO to become empty or full. Reasserting the DMA is accomplished by enabling DMA in the host, setting DMAEN to 1, followed by resetting SERVICEINTR to 0.

DMA mode - transfers from the FIFO to the host

(Note: In the reverse mode, the peripheral may not continue to fill the FIFO when it runs out of transfer data, even when the chip continues to request more data from the peripheral.)

The ECP activates the PDRQ terminal when there is data in the FIFO. The DMA controller must respond to the request by reading data from the FIFO. The ECP deactivates the PDRQ terminal when the FIFO is empty or when the TC becomes true (qualified by $\overline{\text{PDACK}}$), indicating that no more data is required. PDRQ goes inactive after $\overline{\text{PDACK}}$ goes active for the last byte of the data transfer (or on the active edge of $\overline{\text{IOR}}$, on the last byte, if no edge is present on $\overline{\text{PDACK}}$). When PDRQ goes inactive due to the FIFO going empty, then PDRQ is active again as soon as there is one byte in the FIFO. When PDRQ goes inactive due to the TC, then PDRQ is active again when there is one byte in the FIFO, and SERVICEINTR has been re-enabled. (Note: A data underrun may occur when PDRQ is not removed in time to prevent an unwanted cycle.)

interrupt programmed I/O mode or non-DMA mode

The ECP or parallel port FIFOs may also be operated using interrupt-driven programmed I/O.

Programmed I/O transfers are to the ECPDFIFO and ECPAFIFO or from the ECPDFIFO or to/from the TFIFO at 400h. To use the programmed I/O transfers, the host first sets up the direction and state, resets DMAEN to 0 and resets SERVICEINTR to 0.

The programmed I/O empties or fills the FIFO using the appropriate direction and mode.

PRINCIPLES OF OPERATION

programmed I/O - transfers from the FIFO to the host

In the reverse direction an interrupt occurs when the SERVICEINTR bit is 0 and 12 bytes are available in the FIFO. If at this time the FIFO is full it can be emptied completely in a single burst, otherwise 12 bytes may be read from the FIFO in a single burst.

An interrupt is generated when the SERVICEINTR bit is 0 and the number of bytes in the FIFO is greater than or equal to 12. The PINTR terminal can be used for interrupt-driven systems. The host must respond to the request by reading data from the FIFO. This process is repeated until the last byte is transferred out of the FIFO. If at this time the FIFO is full, it can be completely emptied in a single burst. Otherwise, a minimum of 12 bytes may be read from the FIFO in a single burst.

programmed I/O - transfers from the host to the FIFO

In the forward direction an interrupt occurs when SERVICEINTR = 0 and there are 12 or more byte spaces free in the FIFO. At this time if the FIFO is empty it can be filled with a single burst before the empty bit needs to be re-read. Otherwise, it may be filled with 12 bytes.

An interrupt is generated when the SERVICEINTR bit is 0 and the number of bytes in the FIFO is less than or equal to 4. The PINTR terminal can be used for interrupt-driven systems. The host must respond to the request by writing data to the FIFO. If at this time the FIFO is empty, it can be completely filled in a single burst. Otherwise, a minimum of 12 bytes may be written to the FIFO in a single burst. This process is repeated until the last byte is transferred into the FIFO.

PRINCIPLES OF OPERATION

EPP mode

When the EPP mode is selected in the configuration register, the standard and bidirectional modes are also available. If no EPP read, write, or address cycle is currently executing, then the PDx bus is in the standard or bidirectional mode. All output signals ($\overline{\text{STROBE}}$, $\overline{\text{AUTOFD}}$, $\overline{\text{INIT}}$) are set by the device control register and the direction is controlled by the DIR of the control port.

In EPP mode, the system timing is closely coupled to the EPP timing. For this reason, a timer is required to prevent system lockup. The timer indicates if more then 10 µs have elapsed from the start of the EPP cycle ($\overline{\text{IOR}}$ or $\overline{\text{IOW}}$ asserted) to BUSY being deasserted (after command). When a timeout occurs, the current EPP cycle is aborted and the timeout condition is indicated in DSR bit 0.

During an EPP cycle, if $\overline{\text{STROBE}}$ is active, it overrides the EPP write signal forcing the PDx bus to always be in a write mode and the write signal to always be asserted.

write operation timing

The timing for a write operation (address or data) is shown in the timing diagram EPP write data or address cycle. IOCHRDY is driven active low at the start of each EPP write and is released when it has been determined that the write cycle can complete. The write cycle can complete under the following circumstances:

1. If the EPP BUSY is not ready (BUSY is active low) when $\overline{\text{AUTOFD}}$ or $\overline{\text{SELECTIN}}$ goes active, then the write can complete when BUSY goes inactive high.
2. If the EPP BUSY is ready (BUSY is inactive high), then the chip must wait for it to go active low before changing the state of $\overline{\text{AUTOFD}}$, $\overline{\text{STROBE}}$ or $\overline{\text{SELECTIN}}$. The write can complete once BUSY is determined to be inactive.

write sequence of operation

The EPP mode write sequence of operations is as follows:

1. The host selects an EPP register, places data on the data bus, and drives $\overline{\text{IOW}}$ active.
2. The chip drives IOCHRDY inactive (low).
3. When BUSY is not asserted, the chip must wait until BUSY is asserted.
4. The chip places address or data on parallel data (PD) bus and asserts $\overline{\text{STROBE}}$.
5. The chip asserts $\overline{\text{AUTOFD}}$ or $\overline{\text{SELECTIN}}$ indicating that the PD bus contains valid information, and the $\overline{\text{STROBE}}$ signal is valid.
6. The peripheral device deasserts BUSY, indicating that any setup requirements have been satisfied and the chip may begin the termination phase of the cycle.
7. The chip deasserts $\overline{\text{AUTOFD}}$ or $\overline{\text{SELECTIN}}$ which marks the beginning of the termination phase. If it has not already done so, the peripheral should latch the information byte now. The chip latches the data from the data bus for the parallel data (PD) bus and releases IOCHRDY allowing the host to complete the write cycle.
8. The peripheral device asserts BUSY indicating to the host that any hold time requirements have been satisfied and is acknowledging the termination of the cycle.
9. The chip may modify the $\overline{\text{STROBE}}$ and PD signals in preparation for the next cycle.

PRINCIPLES OF OPERATION

read operation timing

The timing for a read operation (data) is shown in Figure 20. IOCHRDY is driven active low at the start of each EPP read and is released when it has been determined that the ready cycle can complete. The ready cycle can complete under the following circumstances:

1. When the EPP bus is not ready (BUSY is active low when $\overline{\text{AUTOFD}}$ goes active) the read can complete when BUSY goes inactive high.
2. When the EPP bus is ready (BUSY is inactive high) the chip must wait for it to go active low before changing the state of $\overline{\text{STROBE}}$ or before $\overline{\text{AUTOFD}}$ goes active. The read can complete once BUSY is determined inactive.

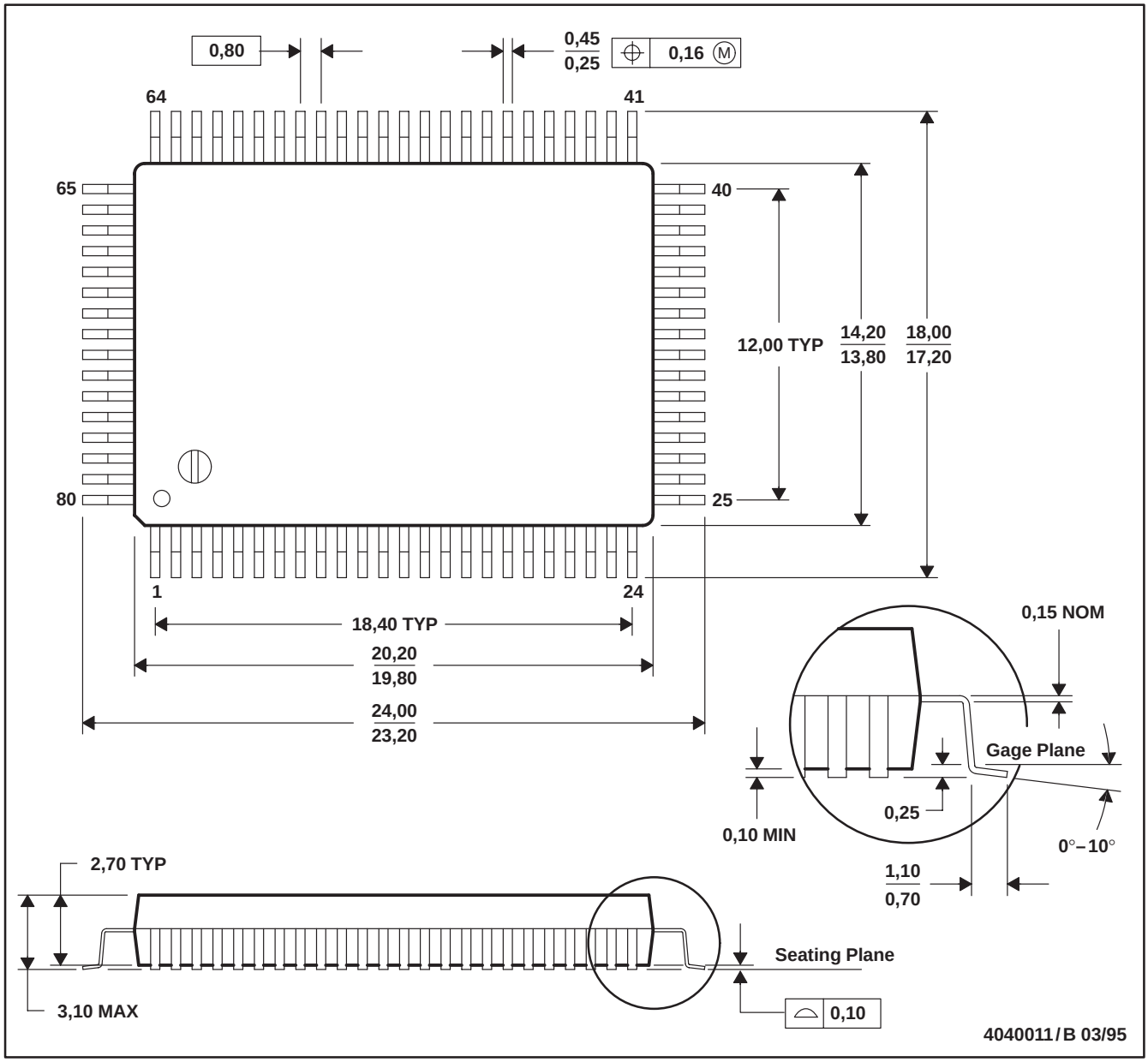
read sequence of operation

The read sequence is as follows:

1. The host selects an EPP register and drives $\overline{\text{IOR}}$ active.
2. The TL16PIR552 drives IOCHRDY inactive (low).
3. When BUSY is not asserted, the chip must wait until BUSY is asserted.
4. The PD bus is put into a high-impedence state and deasserts $\overline{\text{STROBE}}$.
5. The TL16PIR552 asserts $\overline{\text{AUTOFD}}$ or $\overline{\text{SELECTIN}}$ indicating that the PD bus is in a high-impedence state, DIR is set, and the $\overline{\text{STROBE}}$ signal is valid.
6. The peripheral device drives the PD bus valid.
7. A peripheral device deasserts BUSY, indicating that PD is valid and the TL16PIR552 may begin the termination phase of the cycle.
8. The TL16PIR552 latches the data from the PD bus for the data(D) bus, deasserts $\overline{\text{AUTOFD}}$ or $\overline{\text{SELECTIN}}$, and this marks the beginning of the termination phase. The valid data is then driven onto the D bus and asserts (releases) IOCHRDY allowing the host to complete the read cycle.
9. The peripheral device puts PD into a high-impedence state and asserts BUSY, indicating to the host that PD is in a 3-state condition.
10. The chip may modify $\overline{\text{STROBE}}$, DIR, and PD in preparation for the next cycle.

MECHANICAL DATA

PH (R-PQFP-G80) **PLASTIC QUAD FLATPACK**



NOTES: A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.

IMPORTANT NOTICE

Texas Instruments (TI) reserves the right to make changes to its products or to discontinue any semiconductor product or service without notice, and advises its customers to obtain the latest version of relevant information to verify, before placing orders, that the information being relied on is current and complete.

TI warrants performance of its semiconductor products and related software to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

Certain applications using semiconductor products may involve potential risks of death, personal injury, or severe property or environmental damage ("Critical Applications").

TI SEMICONDUCTOR PRODUCTS ARE NOT DESIGNED, INTENDED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT APPLICATIONS, DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS.

Inclusion of TI products in such applications is understood to be fully at the risk of the customer. Use of TI products in such applications requires the written approval of an appropriate TI officer. Questions concerning potential risk applications should be directed to TI through a local SC sales office.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards should be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance, customer product design, software performance, or infringement of patents or services described herein. Nor does TI warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used.