

13 PARALLEL PORTS

ETRAX 100LX has two parallel ports for the connection of high-speed peripheral devices. These ports are designated p0 and p1 respectively and the characteristics and operating principles of both ports are similar.

The parallel ports can be used through register access or internal direct memory access (DMA). The ports can be configured to communicate with compatible peripherals by using various protocols, including:

- IBM XT/AT compatible Centronics;
- IBM PS/2 compatible Centronics;
- Hewlett Packard Fast Mode;
- Fastbyte protocol;
- IEEE 1284 Byte, Nibble, Extended Capability Port (ECP), and Enhanced Parallel Port (EPP) modes;
- ECP wide (16-bit) mode.

A port designated parallel port-W is available when ETRAX 100LX operates in the ECP wide (16-bit) mode. Parallel port-W uses the control and status signals of port p0, together with the 8-bit data lines of p0 and p1 to form a word-wide data bus.

Inputs and outputs to and from the parallel ports are multiplexed on to the same pins as some other interface applications (See chapter 19.10 *Multiplexed Interfaces*).

13.1 Parallel Port Registers

The parallel ports are served by two sets of dedicated registers, one for each port. The required mode of operation and all operational parameters are established, written and read in these registers. The table below summarises the purpose of each register.

Registers	Function
R_PAR0_CTRL_DATA R_PAR1_CTRL_DATA	32-bit write-only registers containing discrete control bits and data for the respective port.
R_PAR0_CTRL R_PAR1_CTRL	8-bit write-only registers for the selection of port control signals oe, seli, autofd, strb and init.
R_PAR0_STATUS_DATA R_PAR1_STATUS_DATA	32-bit read-only registers containing status bits and data for the respective port.
R_PAR0_STATUS R_PAR1_STATUS	16-bit read-only registers containing status bits for the respective port.
R_PAR0_CONFIG R_PAR1_CONFIG	32-bit write-only registers for configuration and control of each port.
R_PAR0_DELAY R_PAR1_DELAY	32-bit write-only registers in which data transfer timing periods are set individually for each port.
R_PAR_ECP16_DATA	In ECP wide (16-bit) mode, this read/write register contains the last data word sent or received at parallel port p0.

For more detailed information on the parallel port registers, please refer to chapter 18.10 *Parallel Port Registers*.

13.2 Modes of Operation

The modes of operation supported by ETRAX 100LX are:

- IEEE 1284 Compatibility (Centronics) mode;
- IBM Fastbyte;
- IEEE-1284 Nibble mode;
- IEEE-1284 Byte mode;
- IEEE-1284 ECP mode;
- ECP wide (16-bit) mode (parallel port-W);
- IEEE 1284 EPP mode;
- Manual mode - this is the default mode of ETRAX 100LX - see the note below.

Note: 1 To switch to any other mode within an IEEE 1284 transaction, a negotiation phase is necessary. The negotiation phase, as well as termination and host recovery, are not supported in ETRAX 100LX and therefore it must be set up by software in the Manual mode.

13.2.1 IEEE-1284 Compatibility (Centronics) Mode

The IEEE-1284 Compatibility (Centronics) mode is a simplex mode for forward data transfers (ETRAX 100LX to peripheral).

The significant signals used by the parallel ports in Compatibility (Centronics) mode, and the corresponding signal names at the multiplexed I/O interface, are listed in the table below.

Interface Pin Name		Signal	Description of Signal	Origin of Signal
Port_0	Port_1			
p0d7 - p0d0	p1d7 - p1d0	D7:D0	Data byte from port to peripheral.	ETRAX 100LX
$\overline{\text{p0strobe}}$	$\overline{\text{p1strobe}}$	nStrobe	Strobe signal. Set low to initiate a data transmission.	ETRAX 100LX
$\overline{\text{p0ack}}$	$\overline{\text{p1ack}}$	nAck	Handshake line. Set low to indicate that the peripheral is ready to receive data.	Peripheral
p0busy	p1busy	Busy	Handshake line. Set high to indicate that the peripheral is busy and thus unable to receive data.	Peripheral

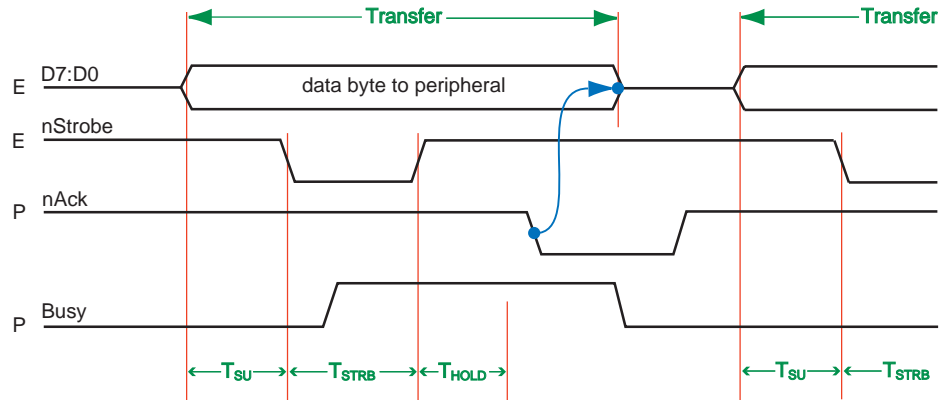
In the Compatibility (Centronics) mode only, ETRAX 100LX can be configured for different methods of handshaking with acknowledgment signal **nAck**. Each port's response to **nAck** is established by **oe_ack** (bit 3) or **ign_ack** (bit 4) in the port's configuration register R_PAR0_CONFIG or R_PAR1_CONFIG respectively.

Note: 2 The **oe_ack** bit and the **ign_ack** bit cannot both be asserted (set to 1), otherwise a conflict will occur.

Bit **oe_ack** asserted

When a data byte is ready, ETRAX 100LX checks that the **Busy** signal is not asserted, confirming that the peripheral can receive data. When setup time T_{SU} elapses, ETRAX 100LX asserts **nStrobe** and the transfer of the data byte commences. The **nStrobe** signal remains low for the duration of T_{STRB} , then the low/high transition of **nStrobe** initiates T_{HOLD} . The data transfer continues until T_{HOLD} elapses or until the peripheral asserts the **nAck** signal, whichever occurs last. Acknowledgment from the

peripheral is thus necessary before ETRAX 100LX can place a new data byte on the bus.



T_{SU} can be set between 10 ns and 5 μ s in steps of 20 ns.

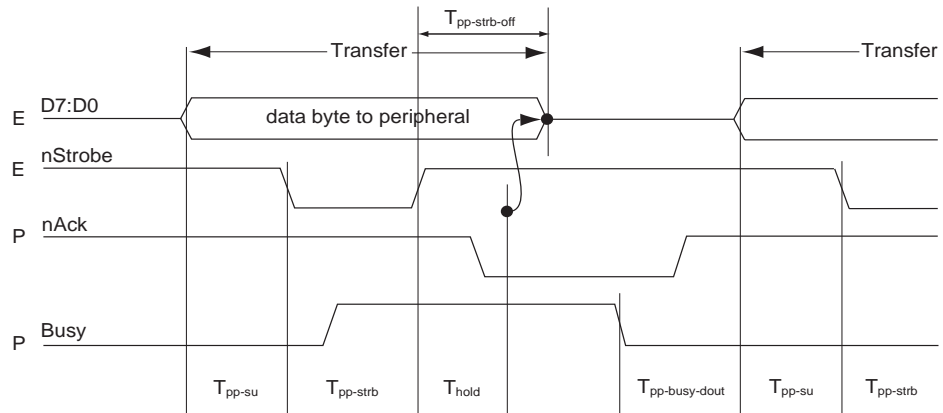
T_{STRB} and T_{HOLD} can be set between 20 ns and 5 μ s in steps of 20 ns.

The prefix n in a signal name represents a negative (active-low) signal in the IEEE 1284 Standard.

The curved arrow represents a sequential dependency.

E = ETRAX 100LX (host) P = peripheral

Figure 13-1 Compatibility (Centronics) Mode Timing with nAck Succeeding T_{HOLD}



T_{su} can be set between 10 ns and 5 μ s in steps of 20 ns.

T_{strb} and T_{hold} can be set between 20 ns and 5 μ s in steps of 20 ns.

The prefix n in a signal name represents a negative (active-low) signal in the IEEE 1284 Standard.

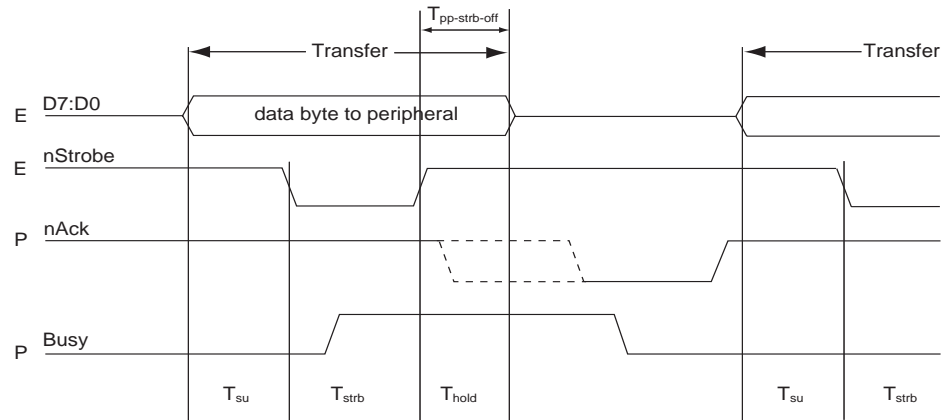
E = ETRAX 100LX (host) P = Peripheral

Figure 13-2 Compatibility (Centronics) Mode Timing with nAck Preceding T_{HOLD}

Bit oe_ack deasserted

When a data byte is ready, ETRAX 100LX checks the Busy signal to confirm that the peripheral can receive data. When T_{SU} elapses, signal nStrobe is asserted by ETRAX 100LX to start the transfer of the data byte. The nStrobe signal remains low for the duration of T_{STRB} , then the rising edge of nStrobe initiates T_{HOLD} . The data transfer continues until T_{HOLD} elapses, irrespective of the state of the nAck signal.

In this configuration, ETRAX 100LX waits for an acknowledgment from the peripheral before placing the next data byte on the bus. The low/high transition of the **nAck** signal is necessary to trigger the next assertion of **nStrobe**.



T_{su} can be set between 10 ns and 5 μ s in steps of 20 ns.

T_{strb} and T_{hold} can be set between 20 ns and 5 μ s in steps of 20 ns.

The prefix n in a signal name represents a negative (active-low) signal in the IEEE 1284 Standard.

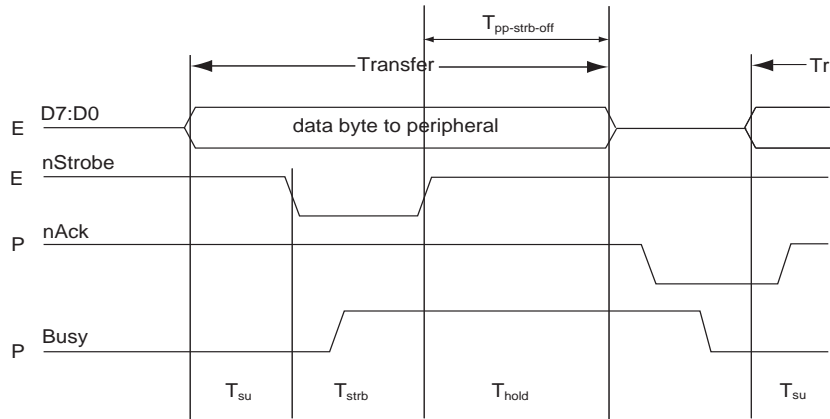
E = ETRAX 100LX (host) P = Peripheral

Figure 13-3 Compatibility (Centronics) Mode Timing with Byte Transfers Terminated by T_{HOLD}

For this example to work, the **ign_ack** bit must be set to 0 (wait).

Bit **ign_ack** asserted

If the port is configured to ignore the **nAck** signal, only the **Busy** signal is monitored by ETRAX 100LX. When a data byte is ready, ETRAX 100LX checks the **Busy** signal to confirm that the peripheral can receive data. When setup time T_{su} elapses, ETRAX 100LX asserts **nStrobe** and the data transfer commences. The **nStrobe** signal remains low for the duration of T_{strb} , then its rising edge initiates T_{hold} . The data transfer stops when T_{hold} elapses.



T_{su} can be set between 10 ns and 5 μ s in steps of 20 ns.

T_{strb} and T_{hold} can be set between 20 ns and 5 μ s in steps of 20 ns.

The prefix n in a signal name represents a negative (active-low) signal in the IEEE 1284 Standard.

E = ETRAX 100LX (host) P = Peripheral

Figure 13-4 Compatibility (Centronics) Mode Timing - Ignore nAck

Compatibility (Centronics) mode time periods

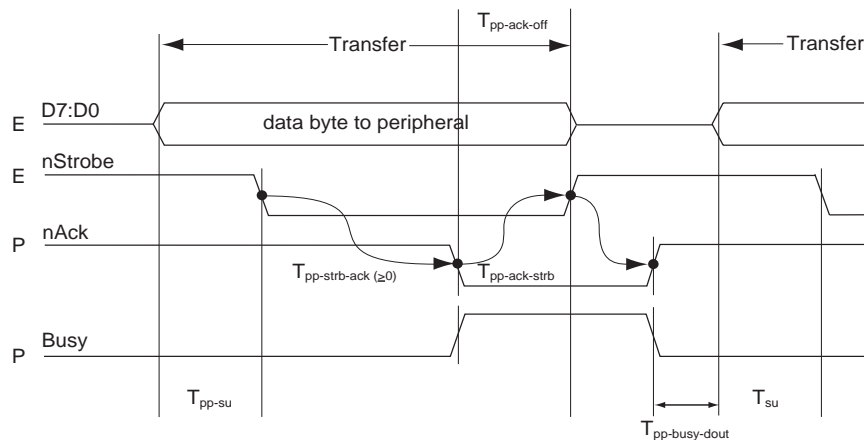
The data setup time (T_{su}), the strobe time (T_{strb}), and the data hold time (T_{hold}) are set individually for each port in register R_PAR0_DELAY or R_PAR1_DELAY respectively.

13.2.2 Fastbyte Mode

Fastbyte is a simplex mode for forward data transfer (ETRAX 100LX to peripheral) with four-phase handshaking. The significant signals used by the parallel ports in Fastbyte mode, and the corresponding signal names at the multiplexed I/O interface, are listed in the table below.

Interface Pin Name		Signal	Description of Signal	Origin of Signal
Port_0	Port_1			
p0d7 - p0d0	p1d7 - p1d0	D7:D0	Data byte from port to peripheral.	ETRAX 100LX
$\overline{\text{p0strobe}}$	$\overline{\text{p1strobe}}$	nStrobe	Strobe signal. Set low to initiate a data transmission.	ETRAX 100LX
$\overline{\text{p0ack}}$	$\overline{\text{p1ack}}$	nAck	Handshake line. Set low to indicate that the peripheral is ready to receive data.	Peripheral
p0busy	p1busy	Busy	Handshake line. Set high to indicate that the peripheral is busy and thus unable to receive data.	Peripheral

When data is ready ETRAX 100LX checks that the **Busy** signal is not asserted, confirming that the peripheral can receive data. After setup time T_{SU} has elapsed, ETRAX 100LX asserts the **nStrobe** signal. When the peripheral asserts **nAck**, ETRAX 100LX de-asserts **nStrobe** and the data transfer ceases.



T_{SU} can be set between 10 ns and 5 μs in steps of 20 ns.

The curved lines represent sequential dependencies.

The prefix n in a signal name represents a negative (active-low) signal in the IEEE 1284 Standard.

E = ETRAX 100LX (host) P = Peripheral

Figure 13-5 Fastbyte Mode Timing

Fastbyte mode time periods

For both ports, setup time T_{SU} is configurable in the respective internal register R_PAR0_DELAY or R_PAR1_DELAY (See chapter 18.10 *Parallel Port Registers*).

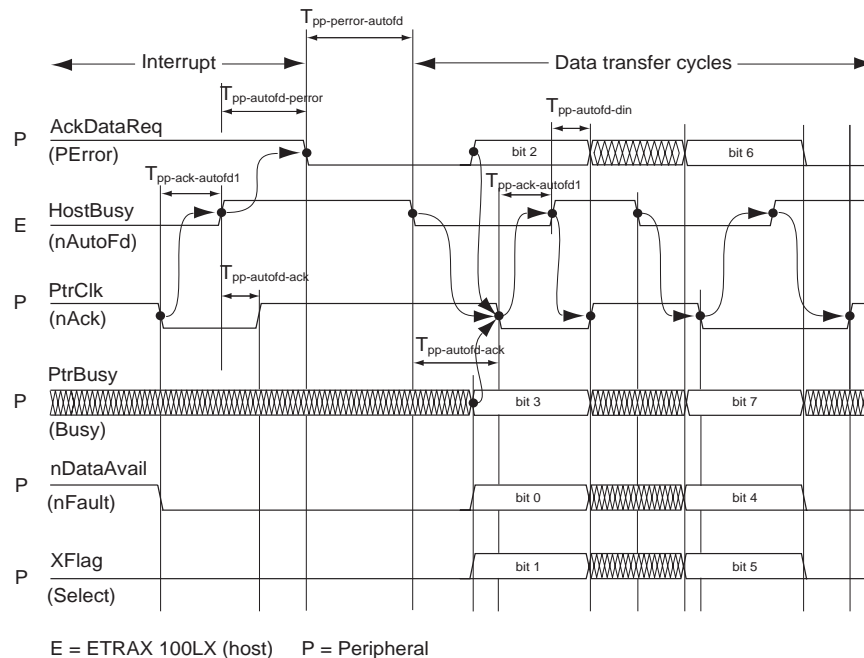
13.2.3 IEEE-1284 Nibble Mode

This is a simplex mode for reverse data transfers only (peripheral to ETRAX 100LX). The significant signals used by the parallel ports in Nibble mode, and the corresponding signal names at the multiplexed I/O interface, are listed in the table below.

Interface Pin Name		Signal	Description of Signal	Origin of Signal
Port_0	Port_1			
$\overline{p0autofd}$	$\overline{p1autofd}$	HostBusy	Handshake line. Set low to indicate that the respective port is ready to receive data, and high to indicate that the port has received a nibble.	ETRAX 100LX
$\overline{p0ack}$	$\overline{p1ack}$	PtrClk	Handshake line. Set low to indicate a valid nibble, and high to acknowledge that the respective port has received a nibble.	Peripheral
$\overline{p0fault}$	$\overline{p1fault}$	nDataAvail	Used for data bit 0, then 4.	Peripheral
p0select	p1select	XFlag	Used for data bit 1, then 5.	Peripheral
p0perror	p1perror	AckDataReq	Used for data bit 2, then 6.	Peripheral
p0busy	p1busy	PtrBusy	Used for data bit 3, then 7.	Peripheral

A peripheral requests attention from ETRAX 100LX with the initial handshaking between signals **PtrClk**, **HostBusy** and **AckDataReq**, which generates an interrupt to the respective parallel port (**par0_peri** for port p0, **par1_peri** for port p1). ETRAX 100LX responds to the interrupt when it is ready to receive data (See section 13.3 *Parallel Port Interrupts*).

The transfer of data takes place on the status signal lines in two cycles, one nibble at a time. Bits 0 to 3 are sent first and bits 4 to 7 follow after a handshake between **PtrClk** and **HostBusy**. The data bus **D7:D0** and the **nStrobe** signal are not used in IEEE 1284 Nibble mode.



The prefix n in a signal name represents a negative (active-low) signal in the IEEE 1284 Standard.

The curved lines represent sequential dependencies.

Figure 13-6 Nibble Mode Timing

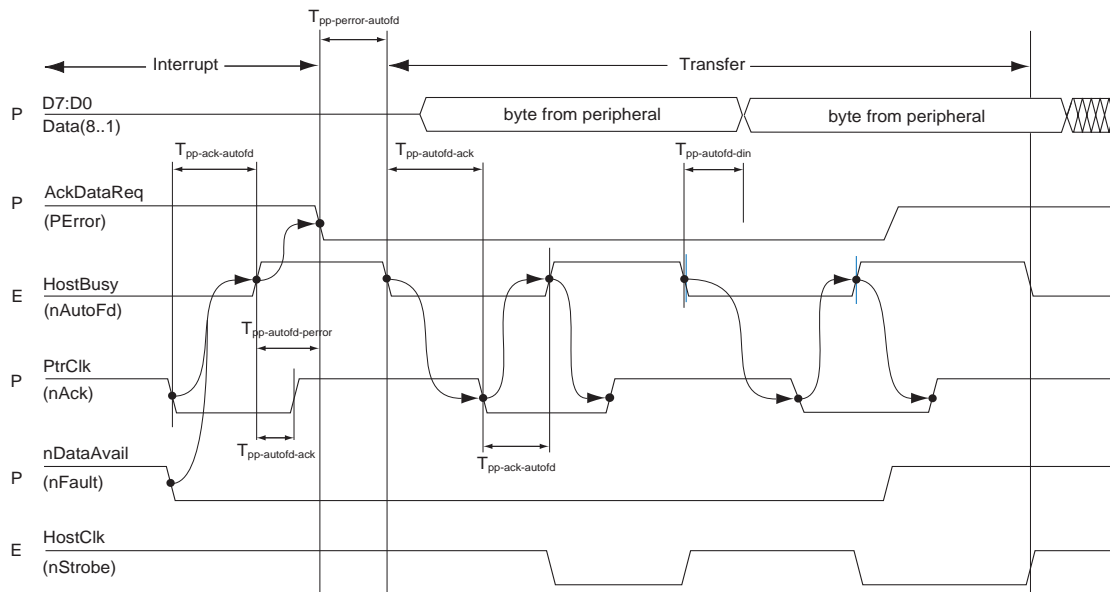
13.2.4 IEEE-1284 Byte Mode

This is a simplex mode for reverse data transfers (peripheral to ETRAX 100LX). The procedure is similar to the IEEE-1284 Nibble mode, the difference being that an entire byte is transferred simultaneously on the data bus.

The significant signals used by the parallel ports in this mode, and the corresponding signal names at the multiplexed I/O interface, are listed in the table below.

Interface Pin Name		Signal	Description of Signal	Origin of Signal
Port_0	Port_1			
p0d7 - p0d0	p1d7 - p1d0	D7:D0	Data bytes from peripheral to the respective port.	Peripheral
p0perror	p1perror	AckDataReq	Set low to acknowledge that the respective port is ready.	Peripheral
$\overline{\text{p0autofd}}$	$\overline{\text{p1autofd}}$	HostBusy	Handshake line. Set low to indicate that the respective port is ready to receive a byte, and high to indicate that the port has received the byte.	ETRAX 100LX
$\overline{\text{p0strobe}}$	$\overline{\text{p1strobe}}$	HostClk	Set low at the end of each byte transfer to indicate that the respective port has received a byte.	ETRAX 100LX
$\overline{\text{p0ack}}$	$\overline{\text{p1ack}}$	PtrClk	Handshake line. Set low to indicate a valid byte on the data lines, and high to acknowledge that the respective port has received a byte.	Peripheral
$\overline{\text{p0fault}}$	$\overline{\text{p1fault}}$	nDataAvail	Set low by peripheral to indicate that data is available.	Peripheral

Handshaking is realized by the $\overline{\text{PtrClk}}$ signal controlled by the peripheral, and the $\overline{\text{HostBusy}}$ signal controlled by ETRAX 100LX. Signal $\overline{\text{HostClk}}$ is pulsed low by ETRAX 100LX at the end of each byte transfer to indicate that the byte has been received.



The prefix n in a signal name represents a negative (active-low) signal in the IEEE 1284 Standard.

The curved lines represent sequential dependencies.

E = ETRAX 100LX (host) P = Peripheral

Figure 13-7 IEE-1284 Byte Mode Timing

13.2.5 IEEE-1284 ECP Mode (Forward and Reverse)

The ECP mode supports half-duplex (forward and reverse), parallel data exchange between ETRAX 100LX and a peripheral, at transfer rates of up to 6 Mbyte/s. The protocol provides for separate data and command cycles, and two types of command cycle:

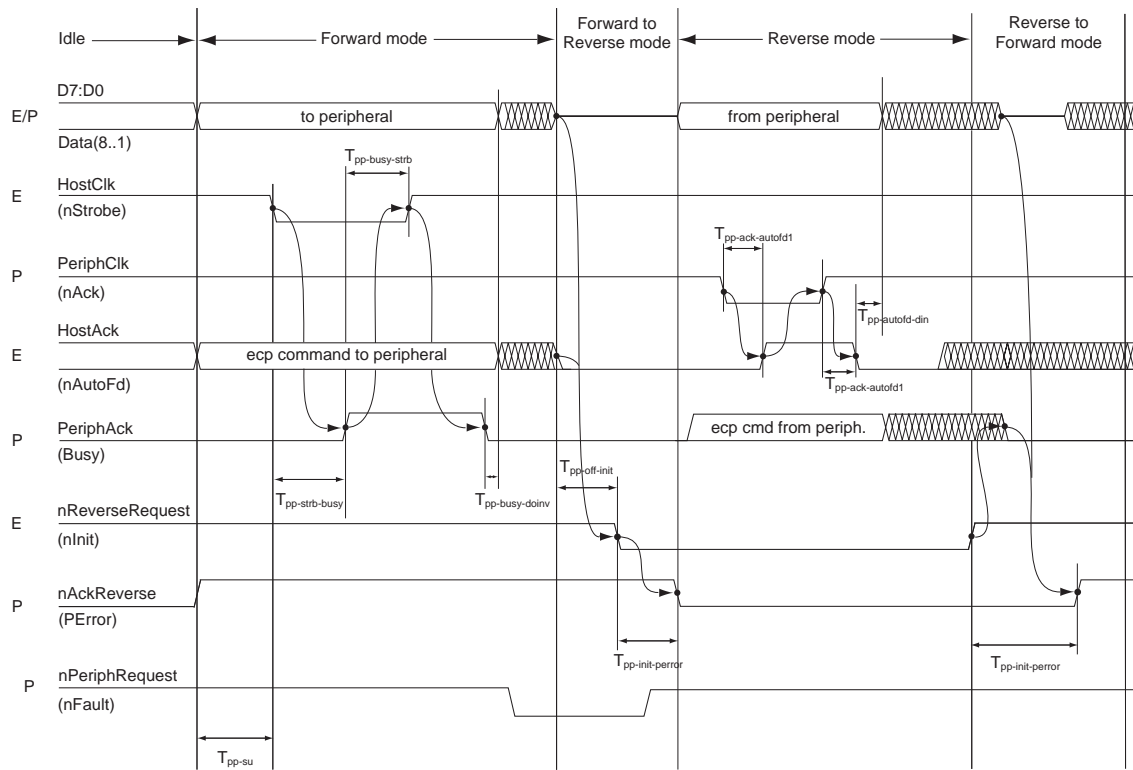
- data compression by Run Length Encoding (RLE) implemented in hardware;
- channel addressing, which must be handled in software.

The significant signals used by the parallel ports in the 1284 ECP mode, and the corresponding signal names at the multiplexed I/O interface, are listed in the table below.

Interface Pin Name		Signal	Description of Signal	Origin of Signal
Port_0	Port_1			
p0d7 - p0d0	p1d7 - p1d0	D7:D0	Carries data bi-directionally (half-duplex).	Both
$\overline{p0strobe}$	$\overline{p1strobe}$	HostClk	Handshakes with PeriphAck to transfer data or addresses in the forward direction. Set low to indicate valid data, and set high to clock data/addresses to the peripheral.	ETRAX 100LX
$\overline{p0ack}$	$\overline{p1ack}$	PeriphClk	Handshakes with HostAck to transfer data or addresses in the reverse direction.	Peripheral
$\overline{p0autofd}$	$\overline{p1autofd}$	HostAck	Provides command/data status in the forward direction. Handshakes with PeriphClk to transfer data in the reverse direction. Set high to indicate a data cycle.	ETRAX 100LX
p0busy	p1busy	PeriphAck	Handshakes with HostClk to transfer data or addresses in the forward direction. Provides command/data status in the reverse direction.	Peripheral
$\overline{p0init}$	$\overline{p1init}$	nReverseRequest	Set low to switch channel to reverse direction.	ETRAX 100LX
p0perror	p1perror	nAckReverse	Set low to acknowledge nReverseRequest .	Peripheral
$\overline{p0fault}$	$\overline{p1fault}$	nPeriphRequest	Set low to indicate that reverse data is available.	Peripheral

The **HostAck** signal is used to distinguish between data and command cycles. When **HostAck** is asserted, a data cycle is taking place: when **HostAck** is low a command cycle is taking place. During a command cycle, bit 7 of the data byte is used to indicate RLE or channel addressing. When bit 7 is set to 0, a run length count is being sent: when bit 7 is set to 1, a channel address is being sent.

The timing diagram below shows the transfer of data and commands in the forward direction, followed by a transfer of data and commands in the reverse direction.



T_{SU} can be set between 10 ns and 5 μ s in steps of 20 ns.

The prefix n in a signal name represents a negative (active-low) signal in the IEEE 1284 Standard.

The curved lines represent sequential dependencies.

E = ETRAX 100LX (host) P = Peripheral

Figure 13-8 ECP Mode Timing

ECP mode time periods

For both ports, setup time T_{SU} is configurable in the respective internal register `R_PAR0_DELAY` or `R_PAR1_DELAY`.

Stall detection

In ECP mode, a stall condition occurs if the peripheral is unable to accept a data byte sent by ETRAX 100LX. In this event ETRAX 100LX can abort the data transfer by asserting signal `nReverseRequest`. Regardless of whether the peripheral has already accepted the byte, it discards the byte and acknowledges ETRAX 100LX by asserting `nAckReverse`. ETRAX 100LX responds by de-asserting `nReverseRequest`.

Stall detection can be implemented in software by means of the `tr_rdy` bit (17), in register `R_PAR0_STATUS_DATA` or `R_PAR1_STATUS_DATA`. This bit is controlled exclusively by the respective parallel port. It is set to 1 when the port reads a new data byte, and to 0 when the software writes to `R_PAR0_CTRL_DATA` (7:0) or `R_PAR1_CTRL_DATA` (7:0). The stall condition can therefore be detected by writing to the control and data register, wait for a period to elapse (e.g. one second), and then read the status of `tr_rdy`. If this bit is set to 0, then data has not been sent.

An example of the code necessary to implement stall detection in ECP mode when using DMA is:

```
while (dma transfer is not ready) {
    WRITE to R_PARn_CTRL_DATA.data
    WAIT (e.g. 1s)
    READ R_PARn_STATUS_DATA.tr_rdy
    if (tr_rdy==0) {
        stall
    }
}
```

A similar work-around can be devised to implement stall detection in ECP mode when using registers.

13.2.6 ECP Wide (16-Bit) Mode

The ECP wide (16-bit) mode is not an IEEE 1284 mode. It is a feature of ETRAX 100LX that is almost identical to the IEEE 1284 ECP mode, except that data words are transferred instead of bytes. ECP wide (16-bit) mode is implemented in parallel port p0 only. Run Length Encoding (RLE) is not supported in this mode. The significant signals used by parallel port-W, and the corresponding signal names at the multiplexed I/O interface, are listed below.

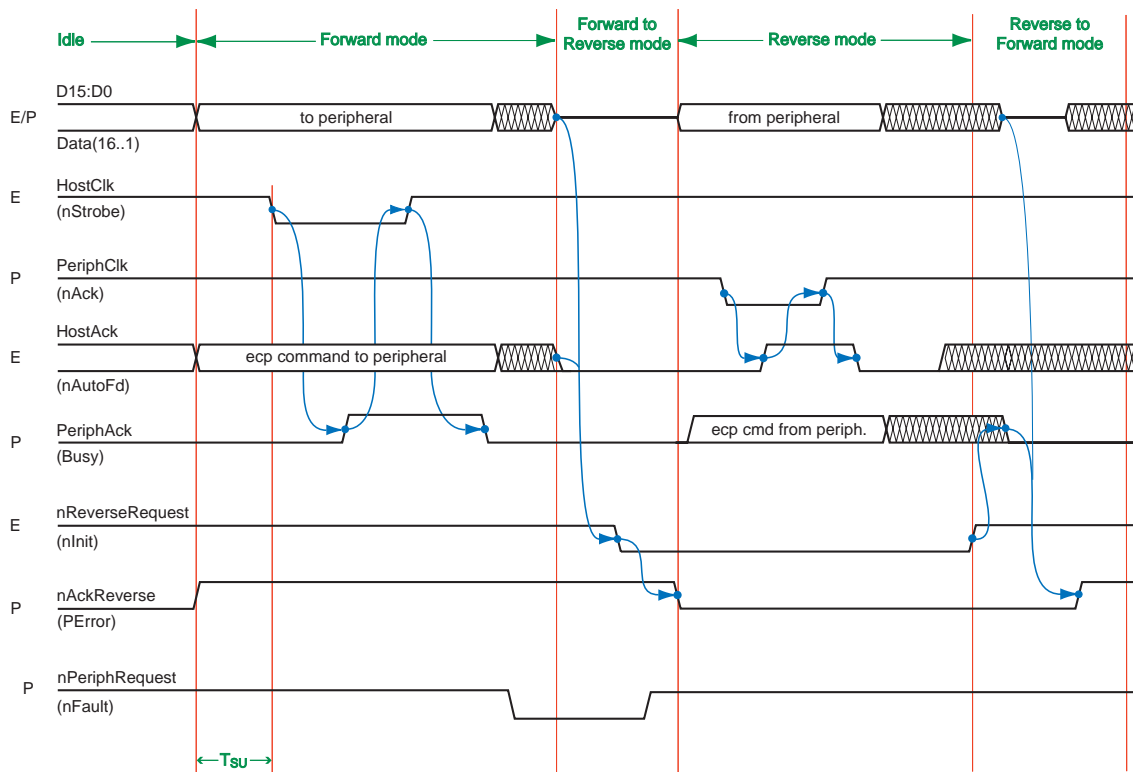
Interface Pin Parallel Port-W	Signal	Description of Signal	Origin of Signal
p0d15 - p0d0	D15:D0	Carries 16-bit data bi-directionally (half-duplex).	Both
p0perror	nAckReverse	Asserted (0) to acknowledge nReverseRequest ($\overline{p0init}$).	Peripheral
$\overline{p0autofd}$	HostAck	Provides command/data status in the forward direction. Handshakes with PeriphClk to transfer data in the reverse direction. Set high to indicate a data cycle.	ETRAX 100LX
$\overline{p0strobe}$	HostClk	Handshakes with PeriphAck to transfer data in the forward direction. Set low to indicate valid data, and set high to clock data to the peripheral.	ETRAX 100LX
$\overline{p0ack}$	PeriphClk	Handshakes with HostAck to transfer data in the reverse direction.	Peripheral
p0busy	PeriphAck	Handshakes with HostClk to transfer data in the forward direction. Provides command/data status in the reverse direction.	Peripheral
$\overline{p0fault}$	nPeriphRequest	Set low to indicate that reverse data is available.	Peripheral
$\overline{p0init}$	nReverseRequest	Set low to switch channel to reverse direction.	ETRAX 100LX

ECP wide (16-bit) register configuration

Only parallel port p0 controls the ECP wide (16-bit) mode. The mode is enabled by bit 10 (**wide**) in port p0 configuration register R_PAR0_CONFIG and bit 2 (**par0**), bit 7 (**par1**) and bit 31 (**par_w**) in general configuration register R_GEN_CONFIG. The following truth table shows the required bit settings.

R_GEN_CONFIG<2> par0	R_GEN_CONFIG<7> par1	R_GEN_CONFIG<31> par_w	R_PAR0_CONFIG<10> wide	Mode
0	X	0	X	
1	X	0	0	ECP
1	0	1	1	ECP wide

All bit combinations other than those shown above are forbidden.



T_{SU} can be set between 10 ns and 5 μ s in steps of 20 ns.

The prefix n in a signal name represents a negative (active-low) signal in the IEEE 1284 Standard.

The curved lines represent sequential dependencies.

E = ETRAX 100LX (host) P = Peripheral

Figure 13-9 ECP Wide (16-bit) Mode Timing

The data lines of both parallel ports are used for data transfers in ECP wide (16-bit) mode. A multiplexer, controlled by the **wide** signal, places the upper byte (**D15:D8**) of the data word on to the data lines of port p1. A dedicated read/write register **R_PAR_ECP_16_DATA** stores the last 16-bit data word sent or received.

ECP wide (16-bit) mode time periods

Setup time T_{SU} is configurable in register **R_PAR0_DELAY**.

13.2.7 EPP Mode

The IEEE Enhanced Parallel Port (EPP) protocol is for high-speed, half-duplex (forward and reverse), parallel data exchange between ETRAX 100LX and a peripheral. All activities are controlled by ETRAX 100LX, which first selects a register within a peripheral, then performs a series of asynchronous read/write operations with the selected register.

The significant signals used by the parallel ports in the EPP mode, and the corresponding signal names at the multiplexed I/O interface, are summarised in the table below.

Interface Pin Name		Signal	Description of Signal	Origin of Signal
Port_0	Port_1			
p0d7:p0d0	p1d7:p1d0	D7:D0	Carries 8-bit data/addresses bi-directionally.	Both
$\overline{\text{p0selectin}}$	$\overline{\text{p1selectin}}$	nAStrb	Address strobe. Set low when address bytes are being written or read. Set high when data bytes are being written or read.	ETRAX 100LX
$\overline{\text{p0autofd}}$	$\overline{\text{p1autofd}}$	nDStrb	Data strobe. Set low when data bytes are being written or read. Set high when address bytes are being written or read.	ETRAX 100LX
$\overline{\text{p0strobe}}$	$\overline{\text{p1strobe}}$	nWrite	Selects write cycles. Set low when address/data bytes are being written. Set high when address/data bytes are being read.	ETRAX 100LX
p0busy	p1busy	nWait	Handshake signal. Set low to indicate that a read or write cycle can commence. Set high to indicate that a read or write cycle can terminate.	Peripheral

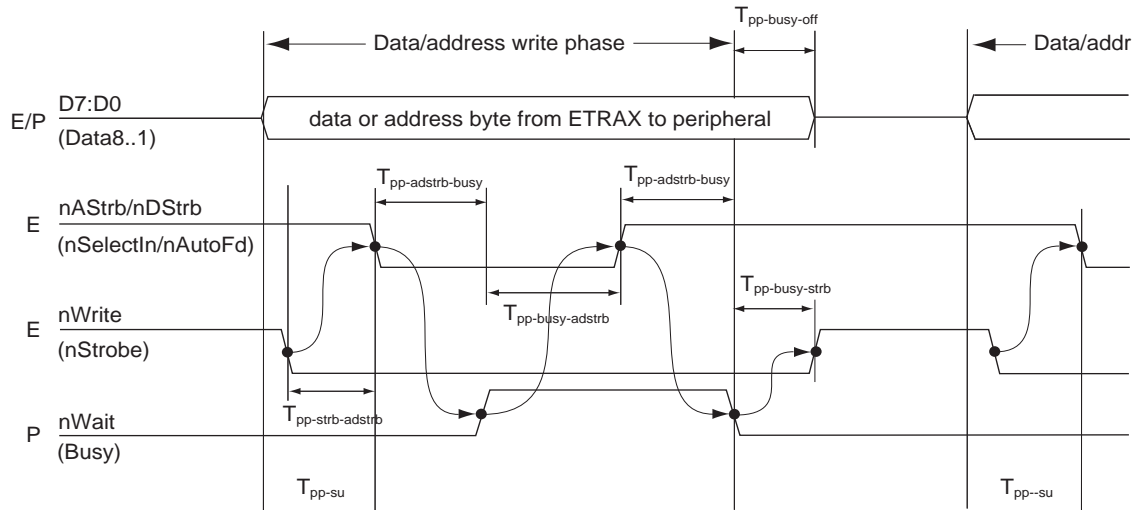
EPP write modes

To meet the requirements of the IEEE 1284 EPP protocol, ETRAX 100LX offers three different EPP write modes (1, 2 and 3). The required mode is established in registers R_PAR0_CONFIG and R_PAR1_CONFIG.

In the EPP write modes, ETRAX 100LX asserts the nWrite signal, places a data or address byte on the data bus, and asserts the corresponding strobe signal (nDStrb for data: nAStrb for an address). Within a response time, the peripheral signifies that it is ready to receive the data/address byte by de-asserting the nWait signal. ETRAX 100LX responds by de-asserting nDStrb/nAStrb to latch the data/address into the peripheral device. The peripheral acknowledges the end of the cycle and signifies that it is ready for the next cycle by asserting the nWait signal.

Timing diagrams of EPP write modes 1 to 3 are given in Figures 14-10 to 14-12 that follow.

13 Parallel Ports



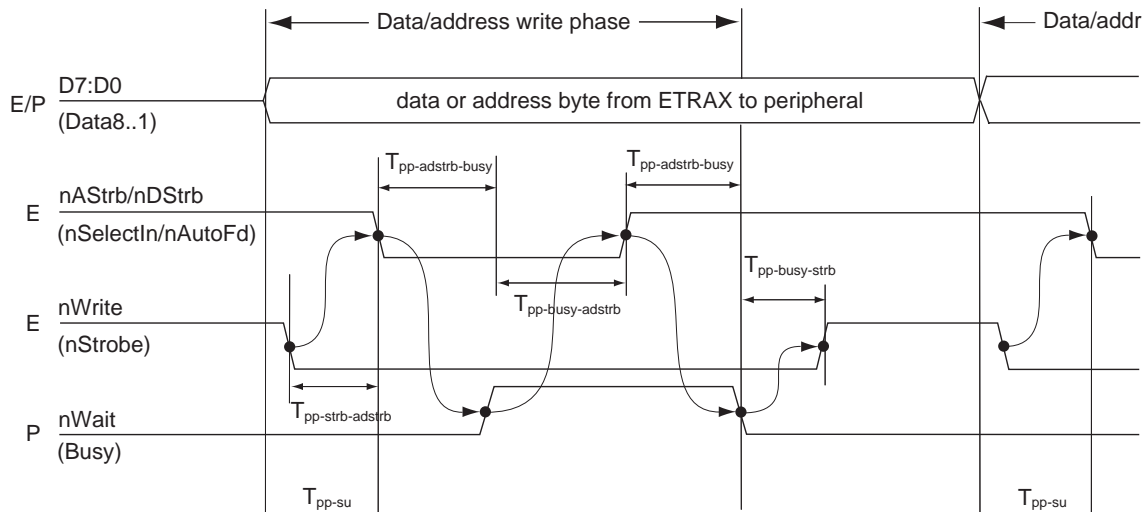
T_{su} can be set between 10 ns and 5 μ s in steps of 20 ns.

The prefix n in a signal name represents a negative (active-low) signal in the IEEE 1284 Standard.

The curved lines represent sequential dependencies.

E = ETRAX 100LX (host) P = Peripheral

Figure 13-10 EPP Address/Data Write Mode 1 Timing



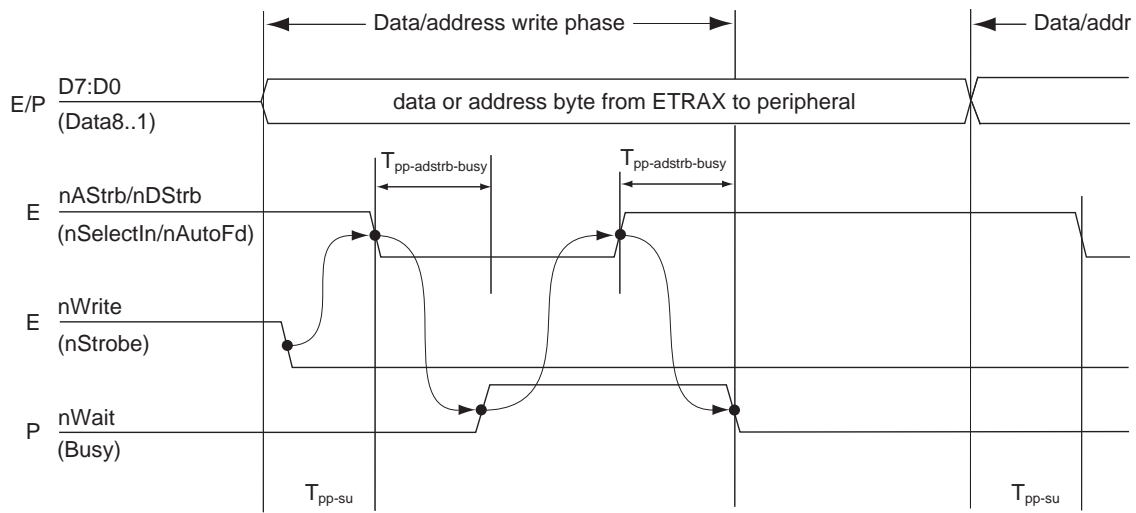
T_{su} can be set between 10 ns and 5 μ s in steps of 20 ns.

The prefix n in a signal name represents a negative (active-low) signal in the IEEE 1284 Standard.

The curved lines represent sequential dependencies.

E = ETRAX 100LX (host) P = Peripheral

Figure 13-11 EPP Address/Data Write Mode 2 Timing



T_{su} can be set between 10 ns and 5 μ s in steps of 20 ns.

The prefix n in a signal name represents a negative (active-low) signal in the IEEE 1284 Standard.

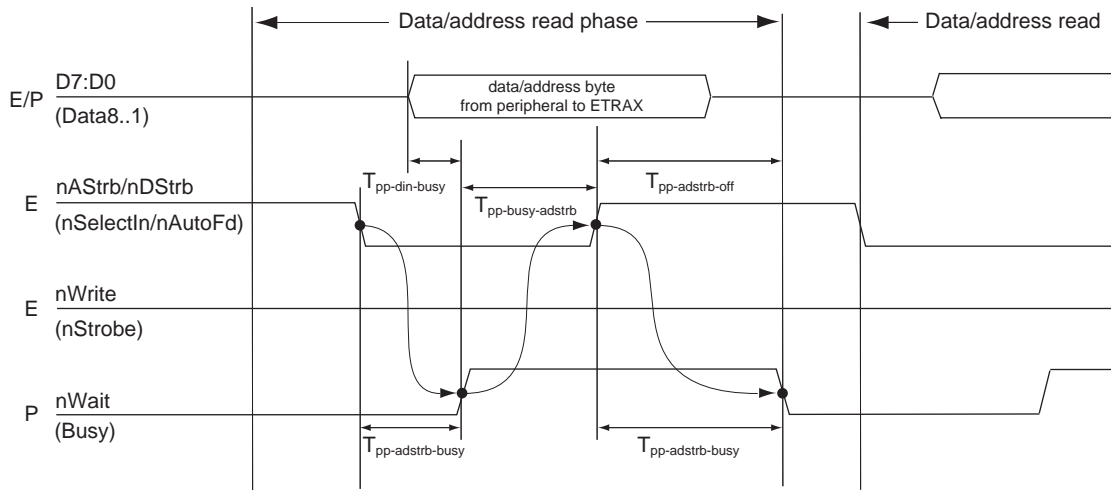
The curved lines represent sequential dependencies.

E = ETRAX 100LX (host) P = Peripheral

Figure 13-12 EPP Address/Data Write Mode 3 Timing

EPP address/data read cycles

In the EPP address or data write mode, ETRAX 100LX uses the data lines to read an address or data byte from the peripheral. ETRAX 100LX signifies a read cycle by asserting the appropriate strobe signal (**nAStrob** for an address: **nDStrobe** for data), and de-asserting the **nWrite** signal. When the peripheral responds, the strobe (**nAStrobe** or **nDStrobe**) is used to latch the address or data byte from the peripheral. The peripheral uses **nWait** for handshaking at the beginning and end of the write cycle.



The prefix n in a signal name represents a negative (active-low) signal in the IEEE 1284 Standard.

The curved lines represent sequential dependencies.

E = ETRAX 100LX (host) P = Peripheral

Figure 13-13 EPP Address/Data Read Cycle Timing

EPP configuration in the mode registers

The read/write functionality of the EPP mode is implemented by a configuration of the parallel port registers. Bits 11 and 2 to 0 in register R_PAR0_CONFIG or R_PAR1_CONFIG respectively are used to choose the read-write mode, where bits 2 to 0 are the mode field and bit 11 acts as the extended mode selector. The choice of an address or data transfer cycle is realized by bit 3 in the respective configuration register.

13.2.8 Manual Mode

In the context of ETRAX 100LX, Manual mode is the configuration of the parallel ports by means of customised software. This feature enhances the versatility of ETRAX 100LX by permitting the implementation of parallel data transfer protocols not otherwise supported within the IEEE 1284 standard.

To facilitate the Manual mode, all signals to and from the parallel ports are programmable. When Manual mode is operational these discrete signals, and the data buses, are read/written by software through the mode registers.

The software-programmable signals are:

Port p0	Port p1	Port-W
p0perror	p1perror	p0perror
$\overline{p0ack}$	$\overline{p1ack}$	$\overline{p0ack}$
p0busy	p1busy	p0busy
$\overline{p0fault}$	$\overline{p1fault}$	$\overline{p0fault}$
p0select	p1select	p0select
p0data_oe	p0data_oe	p0data_oe
$\overline{p0selectin}$	$\overline{p1selectin}$	$\overline{p0selectin}$
$\overline{p0autofd}$	$\overline{p1autofd}$	$\overline{p0autofd}$
$\overline{p0strobe}$	$\overline{p1strobe}$	$\overline{p0strobe}$
$\overline{p0init}$	$\overline{p1init}$	$\overline{p0init}$
p0d7 - p0d0	p1d7 - p1d0	p0d7 - p0d0
		p0d15 - p0d8

Table 13-1 Software-Controlled Signals in Manual Mode

Please refer to chapter 19.10 *Multiplexed Interfaces* for details of the multiplexed I/O interface that includes these signals.

13.3 Parallel Port Interrupts

Both parallel ports can generate a number of different interrupts, if configured to do so in the Interrupt Mask registers.

13.3.1 Peripheral Interrupt

The parallel ports can be configured so that in the Nibble, Byte, ECP and ECP wide (16-bit) modes, a peripheral can attract the attention of ETRAX 100LX through signal sequences. The peripheral requests attention by means of handshaking between the **PtrClk**, **HostBusy** and **AckDataReq** signals, which generates the interrupt. ETRAX 100LX acknowledges the interrupt when it is ready to receive data.

The peripheral interrupt for parallel port p0 is enabled in the **par0_peri** field (bit 10) of the **R_IRQ_MASK0_SET** register. It is cleared by reading the **peri_int** field (bit 24) in register **R_PAR0_CTRL_DATA**.

The peripheral interrupt for parallel port p1 is enabled in the **par1_peri** field (bit 18) of the **R_IRQ_MASK1_SET** register. It is cleared by reading the **peri_int** field (bit 24) in register **R_PAR1_CTRL_DATA**.

13.3.2 ECP Command Interrupt

The parallel ports can be configured to generate an interrupt in ECP reverse mode in response to the start of a command cycle from the peripheral.

The ECP command interrupt for parallel port p0 is enabled in the **par0_ecp_cmd** field (bit 11) of the **R_IRQ_MASK0_SET** register. It is cleared by reading the **ecp_cmd** field (bit 8) in register **R_PAR0_STATUS_DATA**.

The ECP command interrupt for parallel port p1 is enabled in the **par1_ecp_cmd** field (bit 19) of the **R_IRQ_MASK1_SET** register. It is cleared by reading the **ecp_cmd** field (bit 8) in register **R_PAR1_STATUS_DATA**.

13.3.3 Data Available Interrupt

The parallel ports can be configured to generate an interrupt when a port has input data available. When DMA is used for the data transfer, this interrupt indicates that at least one byte has been received since the interrupt was last cleared.

The data available interrupt for parallel port p0 is enabled in the **par0_data** field (bit 9) of the **R_IRQ_MASK0_SET** register. It is cleared by reading the **data** field (bits 7:0) in register **R_PAR0_CTRL_DATA**.

The data available interrupt for parallel port p1 is enabled in the **par1_data** field (bit 17) of the **R_IRQ_MASK1_SET** register. It is cleared by reading the **data** field (bits 7:0) in register **R_PAR1_CTRL_DATA**.

13.3.4 Ready Interrupt

The parallel ports can be configured to generate an interrupt when a port is ready to get new data for transmission.

The ready interrupt for parallel port p0 is enabled in the **par0_ready** field (bit 8) of the **R_IRQ_MASK0_SET** register. It is cleared by writing the **data** field (bits 7:0) in register **R_PAR0_CTRL_DATA**.

The ready interrupt for parallel port p1 is enabled in the **par1_ready** field (bit 16) of the **R_IRQ_MASK1_SET** register. It is cleared by writing the **data** field (bits 7:0) in register **R_PAR1_CTRL_DATA**.

Note: The **par0_ready/par1_ready** bits should be masked off when the DMA is used for data transfers.

13.3.5 EPP Interrupts

A peripheral gains the attention of the parallel port by means of a single interrupt request on the port's peripheral acknowledgement line (**p0ack** and **p1ack**). To generate an interrupt, the acknowledgement signal is pulsed low for period T_p (0.5 ms). Although an interrupt can be asserted at any time it is independent of, and does not interfere with, the EPP cycles.

The status of the EPP interrupt for parallel port p0 is indicated by the **ack** field (bit 27) of register **R_PAR0_STATUS_DATA** and the corresponding field (bit 11) of register **R_PAR0_STATUS**.

The status of the EPP interrupt for parallel port p1 is indicated by the **ack** field (bit 27) of register **R_PAR1_STATUS_DATA** and the corresponding field (bit 11) of register **R_PAR1_STATUS**.

