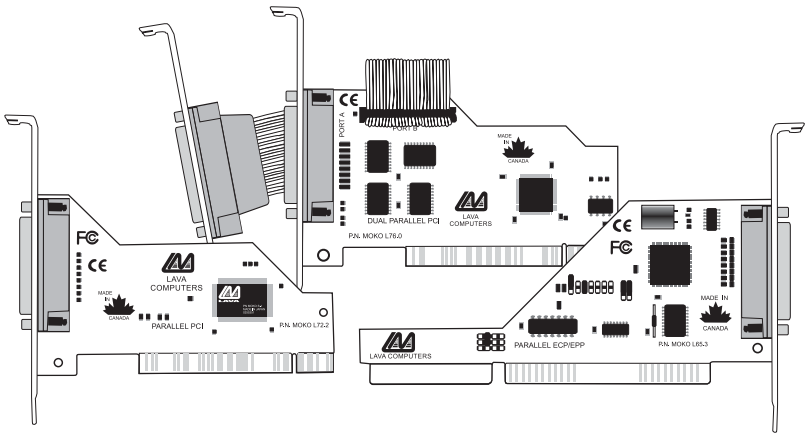
IEEE 1284: Parallel Ports



www.lavalink.com

Table of Contents

IEEE 1284: Parallel Ports

Lava Computer MFG Inc.

This white paper defines and discusses parallel port technologies in the context of Lava's parallel product offerings.

Introduction

Who knows more about parallel ports than Lava? The parallel port as it exists on PCs today has a long history and Lava, founded in 1984, has been there all the way.

The history of the PC's parallel port has been one of increasing speeds. This is a good thing, because the parallel port originally used on PCs was intended exclusively as a printer port. It worked well for sending data to the typical printers of the time. That data, usually ASCII text, didn't place great demands on throughput: printers were neither fast nor sophisticated. Eventually, however, the parallel port had to evolve. Printers needed more control, faster speeds, more advanced font handling, and increased graphical capability. As well, other uses for the parallel port, such as interfacing with external storage devices, created new demands for things like the ability to better handle streams of data moving in two directions.

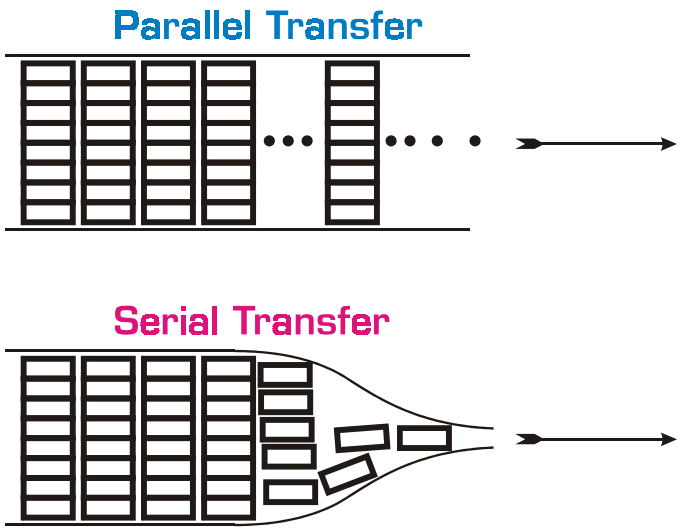
To meet these needs, new modes of parallel port operation were added. Each increased the speed and capability of the parallel port. The modes each are a type of parallel port data transfer as well as a handshaking protocol for that transfer. The more sophisticated modes include other features as well.

The following white paper describes what parallel ports are and covers the basic types and characteristics of parallel ports.

What are parallel ports?

Parallel ports are personal computer interfaces that transfer data (generally) a byte at a time. This contrasts with serial ports, that transfer data one bit at a time.

FIGURE 1. Parallel versus serial data transfer



Parallel ports conform to the specifications of a document of the Institute of Electrical and Electronic Engineers (IEEE) called *IEEE Std 1284-1994: Standard Signaling Method for a Bi-directional Parallel Peripheral Interface for Personal Computers*, or IEEE 1284 for short. This document defines most of the essentials of parallel ports as we have them today.

Basically, IEEE-1284 gathers and harmonizes standards for parallel transmissions that were already in use. It describes a system of asynchronous, fully interlocked, bi-directional communications between a peripheral (often a printer) and a host (usually a PC).

Asynchronous communications are communications where the timing of one communication event occurs in response to another. While individual events in the flow of communications may have a specific duration that is keyed to a clock, the overall timing of communications is not dictated by a clock.

Fully interlocked communications are communications where each control signal sent receives an acknowledging control signal. Devices only send data when the recipient device is ready to receive it, and each transmission is acknowledged as received.

Finally, *bi-directional communications* are simply data transfers in both directions; that is, from a host to a peripheral or from a peripheral to a host.

For parallel communications to operate as asynchronous, fully interlocked, and bi-directional, established modes of communications and protocols for negotiating those modes are necessary. IEEE-1284 covers five parallel port communication modes, which are described below.

Parallel port modes

The IEEE 1284 parallel port modes are called Compatibility, Nibble, Byte, Enhanced Parallel Port (EPP), and Extended Capabilities Port (ECP) modes. Every PC parallel port uses at least a couple of these modes, and some of these modes encompass others.

COMPATIBILITY MODE

Compatibility mode is a "forward channel" mode, or a mode that moves data from the PC to the peripheral. This mode moves bytes (8-bit chunks of data) one at a time to a peripheral, and is the default mode of parallel port operation. To be termed "IEEE 1284 Compatible" a port must be at least capable of this type of operation.

NIBBLE MODE

Nibble mode is a "reverse channel" mode, or one that moves data from the peripheral to the PC. This mode moves data one "nibble" (4 bits) at a time to the PC. Once the nibbles arrive they must be reassembled into bytes, an operation that slows the overall rate at which data can be transmitted. "IEEE Compliant" parallel ports must include at least compatibility and nibble modes, and an approved mechanism for negotiating transfers.

BYTE MODE

Byte mode is another reverse channel mode. This mode improves on nibble mode by enabling data to move to the PC one full byte at a time.

ENHANCED PARALLEL PORT (EPP) MODE

EPP mode works in both forward and reverse directions, moving full bytes each way. The primary advantage of EPP mode over compatibility, nibble, and byte modes is its speed. It can read or write one byte in a single ISA bus cycle, including the time required for protocol handshaking. Earlier modes required four cycles to transfer a byte. The EPP mode can also very quickly switch its direction of data movement, making it a good choice of mode for parallel port data storage devices. EPP mode includes within it compatibility, nibble, and byte modes that can be used if needed.

EXTENDED CAPABILITIES PORT (ECP) MODE

ECP mode adds another layer of sophistication to parallel data transfer. Like EPP mode, ECP moves full bytes on both forward and reverse channels. Again, each byte is moved in one ISA clock cycle. Additionally, ECP ports have data buffering, DMA support (the ability to write data directly to the system's memory without placing demands on the system's CPU), and data com-

"Well done! Our researches have evidently been running on parallel lines, and when we unite our results I expect we shall have a fairly full knowledge of the case."

Sherlock Holmes, *The Hound of the Baskervilles*

pression for more efficient data transfers. ECP mode includes all the other modes within it.

Byte, EPP, and ECP modes each require supporting hardware and software, and so are not always as easily implemented as are the more basic modes. On the other hand, the advantage of the advanced modes is speed.

Parallel port resource requirements

Parallel ports use a variety of system resources from the computers in which they are installed. Each parallel port needs a reserved range of addresses. The specific number of addresses and the address range a port demands depend on the mode of the port. As well, many parallel ports need an assigned interrupt request level, or IRQ. In addition, ECP mode, to use its DMA support, needs a DMA channel assigned.

Resource setting is automatic on systems that support Microsoft's Plug and Play standard. This applies to Lava's PCI parallel port cards. ISA cards have their resource settings configured by jumper switches on the card.

The resources used by a parallel port cannot conflict with the other components in the system, including other parallel ports.

ADDRESSES

The first category of resource used by parallel ports is addresses. Addresses are assigned to parallel ports in numerical order. When speaking of multiple parallel ports in a system, they are conventionally referred to as *LPT* ports. This naming arises from the original use of parallel ports as printer ports, and "LPT" stands for "line printer." When a system has multiple ports they are referred to in numerical order of address as LPT1, LPT2, and so on.

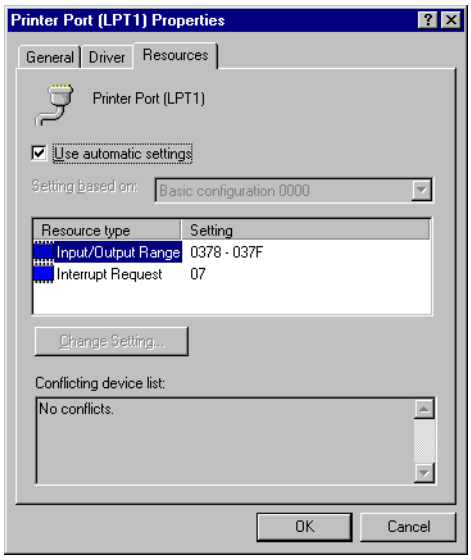
SPP Port addressing. The standard parallel port uses three consecutive addresses. The first is the base address or Data register, the second is the port's Status register, and the third is the port's Control register. These addresses are usually in one of the following ranges:

TABLE 1. Standard parallel port addresses

	Base Address (Data Register)	Status Register	Control Register
Range 1	3BCh	3BDh	3BEh
Range 2	378h	379h	37Ah
Range 3	278h	279h	27Ah

Parallel ports occupy these ranges in numerical order. The first parallel port (LPT1) in current systems usually has its base address at 378h, although the other two base addresses are available. LPT2 can occupy base address 378h or 278h, and LPT3 will be limited to 278h. These assignments can be modified in Windows®.

FIGURE 2. SPP port resource usage



EPP port addressing. EPP ports add five registers to the three used by the SPP. These registers are in the range from *base address + 3* to *base address + 7*. For example, an EPP port with base address 378h will have its EPP registers at 37Bh to 37Fh.

ECP port addressing. ECP ports add three registers to the three used by the SPP. These registers are in the range from *base address + 400h* to *base address + 402h*. For example, an ECP port with base address 378h will have its EPP registers at 37Bh to 37Fh and its ECP registers at 778h to 77Ah (Remember that an ECP port will also have EPP registers).

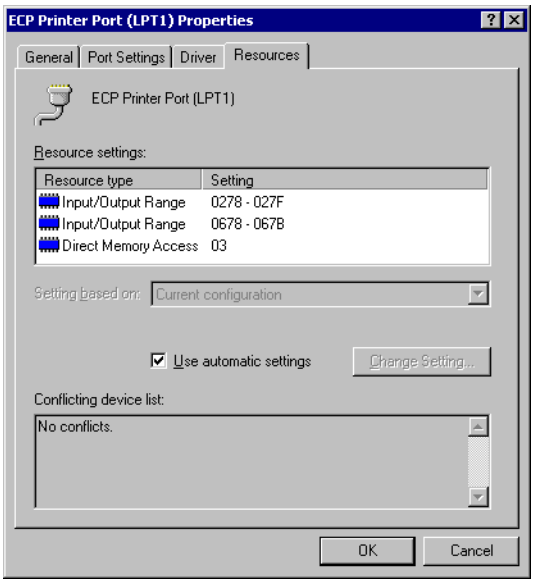
INTERRUPT REQUEST LEVELS

To send an interrupt request to the computer's CPU, a parallel port needs to have an assigned interrupt request level, or IRQ. Usually LPT1 uses IRQ7 and LPT2 uses IRQ5. When these IRQ assignments conflict with other devices such as sound cards, problems might result, and IRQs may need to be reassigned. Also, some peripherals may not require interrupts for the parallel port; in this case the port can operate without an assigned interrupt level, although perhaps less efficiently.

DMA CHANNELS

For an ECP to use its DMA (Direct Memory Access) capability, it needs a DMA channel available for its use. The DMA channel is in the range from 0 to 3.

FIGURE 3. ECP port resource usage showing DMA channel setting



Parallel port types and products

STANDARD PARALLEL PORT (SPP)

The original PC parallel printer port operated in compatibility and nibble modes. Today, this type of parallel port is called the "*Standard Parallel Port,"* or SPP.

SIMPLE BI-DIRECTIONAL OR PS/2 PORT

The next version of the parallel port arrived with the IBM PS/2. It added byte mode, and is generally called a *"Simple bi-directional"* or "*PS/2-type*" port. Lava's Parallel Bi-directional board is an example of this type of parallel port.

ENHANCED PARALLEL PORT (EPP)

The next type of parallel port takes its name from its fastest mode, and is called an "*Enhanced Parallel Port*" or EPP. Lava's Parallel-PCI, Parallel-PCI/LP, and Dual Parallel-PCI boards are all EPPs. They set the standard for ease of use among parallel ports. As PCI cards, they are fast Plug and Play devices that install easily and with unrivaled flexibility.

EXTENDED CAPABILITIES PORT (ECP)

Finally, the Extended Capabilities Port mode lends its name to ports called "*Extended Capabilities Ports*," or ECPs. Lava's ECP port board, the Lava Parallel-ECP/EPP, is a sophisticated parallel port board for the ISA bus.

How fast are parallel ports?

MEASURING THROUGHPUT: UNDERSTANDING THE NUMBERS

When describing data throughput rates to hard drives or across I/O interfaces, the terminology and abbreviations can be confusing. To start, quantities of data are measured in "bits" or "bytes." A byte is simply eight bits. When describing data storage capacity, the industry typically speaks of *bytes* of data: a floppy disk has 1440 kilobytes of storage capacity, a hard disk has 20 gigabytes of storage capacity, and so on. When it comes to data flows, however, we tend to speak of *bits* of data: a hard disk has a throughput of 25 megabits per second, a 16550 UART serial port has a speed of 115.2 kilobits per second, a Fast Ethernet connection is a 100 megabit per second connection, etc.

Throughput measurements specify a given quantity of data transferred in a given length of time, usually a minute or a second. When comparing products' relative speeds, it is important to be aware of inconsistent and overlapping abbreviations for "megabits per second" and "megabytes per second" to avoid misconceptions. The table below lists some of the standard throughput measures and their abbreviations, and highlights a possible area of confusion.

TABLE 2. Throughput terminology and abbreviations

Full Term	Abbreviation	Other Abbreviations Used
kilobits per second	kbps	Kbps
megabits per second	Mbps	mbps; Mb/s; MBps, MB/s
megabytes per second	MB/s	Mb/s; MBps
gigabits per second	Gbps	

For example, a conventional EPP parallel port has a maximum throughput of about 1.5 Mbps. A competing product might claim a maximum throughput of 1.5 MB/s. Do not be misled; the two ports transfer data at the same speed.

PARALLEL MODE SPEEDS

The speed of any parallel port is largely determined by the parallel mode it is using. The three most typical types of ports—a bi-directional or PS/2 port, an EPP port, and an ECP port—have theoretical maximum speeds of 150 kbps, 1.5 Mbps, and 2.5 Mbps respectively when implemented on the ISA bus.

The advanced modes—EPP and ECP—offer substantial speed advantages over the earlier modes. However, for a parallel connection to take advantage of an advanced mode, several conditions are necessary:

1. The port itself must be capable of that mode.
2. There must be driver software present that is designed to handle that mode.
3. The port's mode must be properly configured in the BIOS and operating system. For example, for an ECP port to take full advantage of its Direct Memory Access capabilities, a DMA channel needs to be configured.
4. Last but not least, the peripheral involved must be able to operate in that mode.

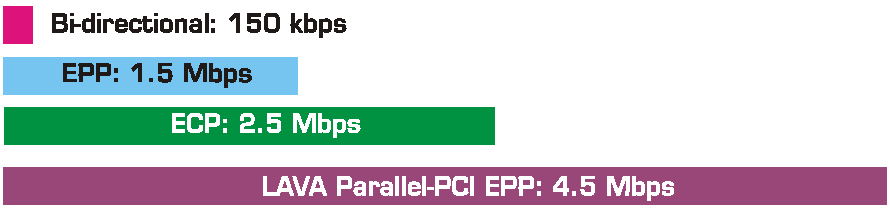
If these conditions are not met, a parallel connection will fall back to slower modes of operation, until it reaches a mode that is fully supported.

THE INFLUENCE OF BUS SPEED

The speed of the bus on which the parallel port is implemented is a major factor on overall parallel port speed. The traditional parallel port has been implemented on the ISA bus, which has a clock speed of 8 MHz. Since each cycle of a parallel port data transfer has a time allocated for its execution and completion, bus speed does make a difference.

In contrast to the ISA bus, the PCI bus has a clock speed of 33 Mhz. In practice, this means that the process of executing each instruction of a parallel port data transfer happens approximately three times more quickly than on the ISA bus, leading to a major speed boost, regardless of mode. The chart below compares parallel port speeds for different modes, and illustrates the increased speed of a PCI-bus EPP parallel port. The speed of the PCI bus, and its effective use by Lava's own Application-Specific Integrated Circuit (ASIC), give the Lava Parallel-PCI its speed.

FIGURE 4. Parallel port speeds: theoretical maximums



OTHER SPEED FACTORS

Beyond mode and bus, many other factors influence port speed. In other words, significant speed differences can result in actual use. On the hardware side these factors include the speed and architecture of the computer's CPU, and peripheral support for EPP or ECP. On the software side, the type of code and operating system will affect port speed. For example, other things being equal, DOS will have faster parallel port transfers than Windows.

Parallel port daisy chains

USB and IEEE 1394 (FireWire®) are often touted as having daisy-chaining. A less well-known fact is that parallel peripherals can daisy-chain as well. A supplement to the IEEE 1284 (IEEE 1284.3) specification defines a daisy-chaining protocol for up to nine peripheral devices on a single parallel port. The PC and the last device in the chain form the ends of the chain, and all the devices between them are connected in series.

The intermediate devices all need to understand the daisy-chaining protocol, although the final device does not. When the PC wishes to communicate with a device in the chain, the other peripherals adopt a "transparent" or "pass-through" mode. A common example of parallel port daisy-chaining is a ZIP™ drive attached to a computer, with a printer attached in turn to the ZIP™ drive.

Parallel port cables and connectors

The IEEE 1284 specification lists three cable connectors and their pinouts. All are seen on PCs and peripherals. The traditional D-sub 25-pin connector is called the IEEE 1284-A, the traditional Centronics connector is called the IEEE 1284-B, and a smaller new connector similar to the Centronics connector is called the IEEE 1284-C.

IEEE 1284-A

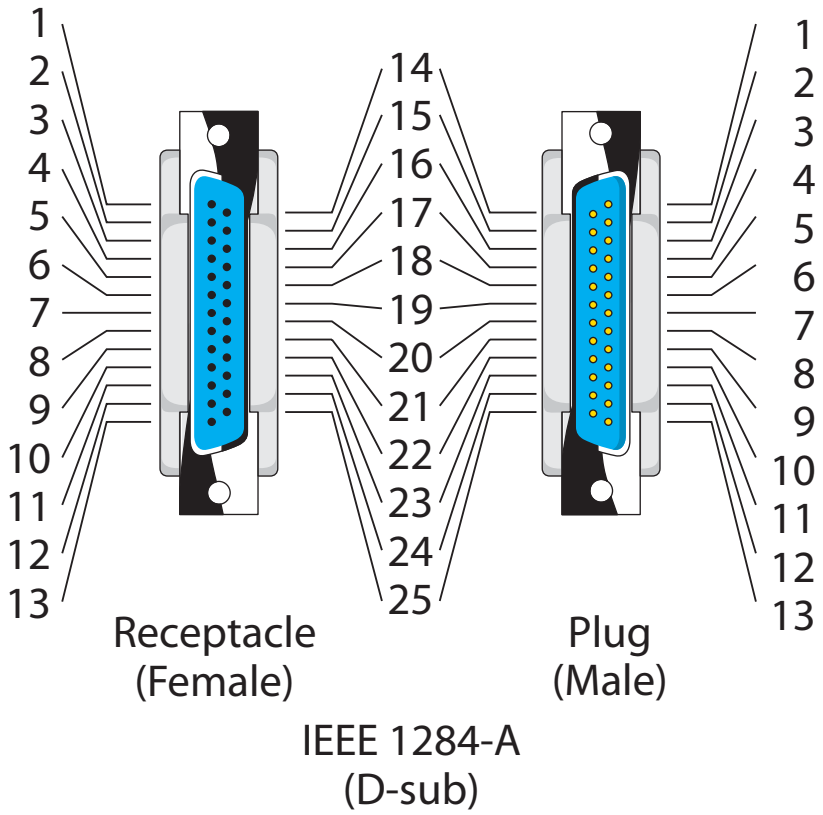


TABLE 3. IEEE 1284-A pin assignments

	Signal Name (Active State)[a]					
Pin	**Compatibility Mode**	**Nibble/Byte Modes**	**ECP Mode**	**EPP Mode**	**Signal Function**	**Signal Source**
1	nStrobe (L)	HostClk (V)	HostClk (C)	nWrite (L)	Control	Host
2	Data 1 (V)[b]	Data 1 (V)[b]	Data 1 (V)[b]	AD1 (V)[b]	Data	Either[c]
3	Data 2 (V)[b]	Data 2 (V)[b]	Data 2 (V)[b]	AD2 (V)[b]	Data	Either[c]
4	Data 3 (V)[b]	Data 3 (V)[b]	Data 3 (V)[b]	AD3 (V)[b]	Data	Either[c]
5	Data 4 (V)[b]	Data 4 (V)[b]	Data 4 (V)[b]	AD4 (V)[b]	Data	Either[c]

TABLE 3. IEEE 1284-A pin assignments

	Signal Name (Active State)[a]					
Pin	**Compatibility Mode**	**Nibble/Byte Modes**	**ECP Mode**	**EPP Mode**	**Signal Function**	**Signal Source**
6	Data 5 (V)[b]	Data 5 (V)[b]	Data 5 (V)[b]	AD5 (V)[b]	Data	Either[c]
7	Data 6 (V)[b]	Data 6 (V)[b]	Data 6 (V)[b]	AD6 (V)[b]	Data	Either[c]
8	Data 7 (V)[b]	Data 7 (V)[b]	Data 7 (V)[b]	AD7 (V)[b]	Data	Either[c]
9	Data 8 (V)[b]	Data 8 (V)[b]	Data 8 (V)[b]	AD8 (V)[b]	Data	Either[c]
10	nAck (L)	PtrClk (V)	PeriphClk (C)	Intr (H)	Control	Peripheral
11	Busy (H)	PtrBusy (V)	PeriphAck (V)	nWait (L)	Control	Peripheral
12	PError (H)	AckData (V)	nAckReverse (L)	User defined 1 (M)	Control	Peripheral
13	Select (H)	Xflag (V)	Xflag (V)	User defined 3 (M)	Control	Peripheral
14	nAutoFd (L or V[d])	HostBusy (V)	HostAck (V or C[e])	nDStrb (L)	Control	Host
15	nFault (L)	nDataAvail (V)	nPeriphRequest (L)	User defined 2 (M)	Control	Peripheral
16	nInit (L)	nInit (H)	nInit (L)	nReverseRequest (L)	Control	Host
17	nSelectIn (L)	IEEE1284 Active (V)	IEEE1284 Active (V)	nAStrb (L)	Control	Host
18	Signal ground for Pin 1[f]				Ground	
19	Signal ground for Pins 2 and 3[f]				Ground	
20	Signal ground for Pins 4 and 5[f]				Ground	
21	Signal ground for Pins 6 and 7[f]				Ground	
22	Signal ground for Pins 8 and 9[f]				Ground	
23	Signal ground for Pins 11 and 15[f]				Ground	
24	Signal ground for Pins 10, 12, and 13[f]				Ground	
25	Signal ground for Pins 14, 16, and 17[f]				Ground	

a. The active state of a signal is indicated in brackets after its name: L=low, H=high, V=varies low or high, C=closed loop or handshaking, M=manufacturer-specific.

b. Data bits are in order from 1 (least significant bit) to 8 (most significant bit).

c. “Either” because data signals are bi-directional; from host on forward path, from peripheral on reverse path.

d. Interpreted differently by different peripherals. On some printers, this bit may be set low by the host to indicate an automatic line feed. Sometimes this bit is used as a ninth data bit or a parity or command bit.

e. In forward path, this bit indicates whether the signals represent data or commands. In the reverse path, this bit is used on a handshake with PeriphClk (Pin 10).

f. These grounds have the same name in all modes.

IEEE 1284-B

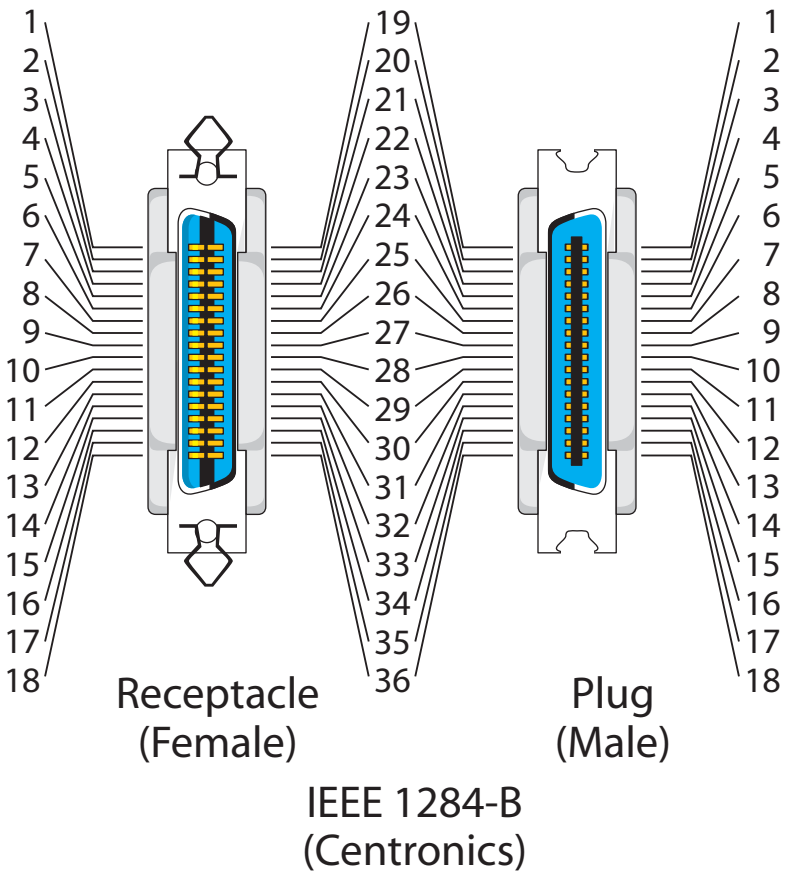


TABLE 4. IEEE 1284-B pin assignments

	Signal Name (Active State)[a]					
Pin	**Compatibility Mode**	**Nibble/Byte Modes**	**ECP Mode**	**EPP Mode**	**Signal Function**	**Signal Source**
1	nStrobe (L)	HostClk (V)	HostClk (C)	nWrite (L)	Control	Host
2	Data 1 (V)[b]	Data 1 (V)[b]	Data 1 (V)[b]	AD1 (V)[b]	Data	Either[c]
3	Data 2 (V)[b]	Data 2 (V)[b]	Data 2 (V)[b]	AD2 (V)[b]	Data	Either[c]
4	Data 3 (V)[b]	Data 3 (V)[b]	Data 3 (V)[b]	AD3 (V)[b]	Data	Either[c]
5	Data 4 (V)[b]	Data 4 (V)[b]	Data 4 (V)[b]	AD4 (V)[b]	Data	Either[c]
6	Data 5 (V)[b]	Data 5 (V)[b]	Data 5 (V)[b]	AD5 (V)[b]	Data	Either[c]
7	Data 6 (V)[b]	Data 6 (V)[b]	Data 6 (V)[b]	AD6 (V)[b]	Data	Either[c]
8	Data 7 (V)[b]	Data 7 (V)[b]	Data 7 (V)[b]	AD7 (V)[b]	Data	Either[c]
9	Data 8 (V)[b]	Data 8 (V)[b]	Data 8 (V)[b]	AD8 (V)[b]	Data	Either[c]

TABLE 4. IEEE 1284-B pin assignments

	Signal Name (Active State)[a]					
Pin	**Compatibility Mode**	**Nibble/Byte Modes**	**ECP Mode**	**EPP Mode**	**Signal Function**	**Signal Source**
10	nAck (L)	PtrClk (V)	PeriphClk (C)	Intr (H)	Control	Peripheral
11	Busy (H)	PtrBusy (V)	PeriphAck (V)	nWait (L)	Control	Peripheral
12	PError (H)	AckData (V)	nAckReverse (L)	User defined 1 (M)	Control	Peripheral
13	Select (H)	Xflag (V)	Xflag (V)	User defined 3 (M)	Control	Peripheral
14	nAutoFd (L or V[d])	HostBusy (V)	HostAck (V or C[e])	nDStrb (L)	Control	Host
15	Not defined					
16	Logic Ground[f]				Ground	
17	Chassis Ground[f]				Ground	
18	Peripheral Logic High[f] (V[g])				Control and/or Voltage[h]	Peripheral
19	Signal ground for Pin 1[f]				Ground	
20	Signal ground for Pin 2[f]				Ground	
21	Signal ground for Pin 3[f]				Ground	
22	Signal ground for Pin 4[f]				Ground	
23	Signal ground for Pin 5[f]				Ground	
24	Signal ground for Pin 6[f]				Ground	
25	Signal ground for Pin 7[f]				Ground	
26	Signal ground for Pin 8[f]				Ground	
27	Signal ground for Pin 9[f]				Ground	
28	Signal ground for Pins 10, 12, and 13[f]				Ground	
29	Signal ground for Pins 11 and 32[f]				Ground	
30	Signal ground for Pins 14, 31, and 36[f]				Ground	
31	nInit (L)	nInit (H)	nInit (L)	nReverseRequest (L)	Control	Host
32	nFault (L)	nDataAvail (V)	nPeriphRequest (L)	User defined 2 (M)	Control	Peripheral
33	Not defined					
34	Not defined					
35	Not defined					
36	nSelectIn (L)	IEEE1284 Active (V)	IEEE1284 Active (V)	nAStrb (L)	Control	Host

a. The active state of a signal is indicated in brackets after its name: L=low, H=high, V=varies low or high, C=closed loop or handshaking, M=manufacturer-specific.

b. Data bits are in order from 1 (least significant bit) to 8 (most significant bit).

c. “Either” because data signals are bi-directional; from host on forward path, from peripheral on reverse path.

d. Interpreted differently by different peripherals. On some printers, this bit may be set low by the host to indicate an automatic line feed. Sometimes this bit is used as a ninth data bit or a parity or command bit.

e. In forward path, this bit indicates whether the signals represent data or commands. In the reverse path, this bit is used on a handshake with PeriphClk (Pin 10).

f. These signals and grounds have the same name in all modes.

g. The peripheral sets this signal high to indicate the signals it drives are valid. It sets this signal low to indicate it is powered off or a dignal it drives is invalid.

h. Manufacturers can use this pin for its defined control function or to supply +5 VDC to a device.

IEEE 1284-C

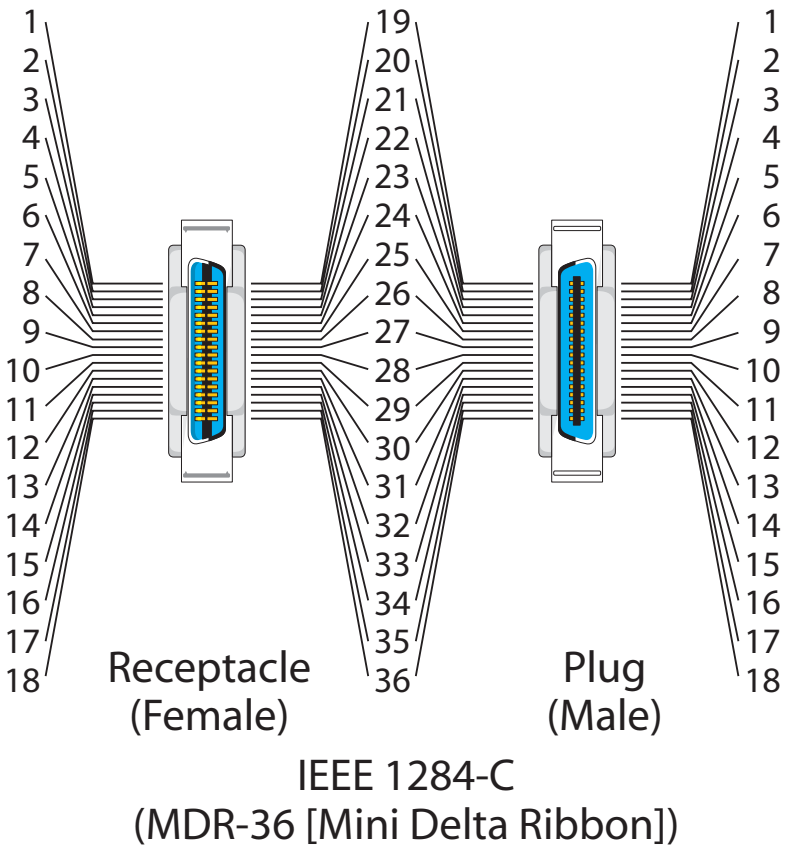


TABLE 5. IEEE 1284-C pin assignments

	Signal Name (Active State)[a]					
Pin	**Compatibility Mode**	**Nibble/Byte Modes**	**ECP Mode**	**EPP Mode**	**Signal Function**	**Signal Source**
1	Busy (H)	PtrBusy (V)	PeriphAck (V)	nWait (L)	Control	Peripheral
2	Select (H)	Xflag (V)	Xflag (V)	User defined 3 (M)	Control	Peripheral
3	nAck (L)	PtrClk (V)	PeriphClk (C)	Intr (H)	Control	Peripheral

TABLE 5. IEEE 1284-C pin assignments

	Signal Name (Active State)[a]					
Pin	**Compatibility Mode**	**Nibble/Byte Modes**	**ECP Mode**	**EPP Mode**	**Signal Function**	**Signal Source**
4	nFault (L)	nDataAvail (V)	nPeriphRequest (L)	User defined 2 (M)	Control	Peripheral
5	PError (H)	AckData (V)	nAckReverse (L)	User defined 1 (M)	Control	Peripheral
6	Data 1 (V)[b]	Data 1 (V)[b]	Data 1 (V)[b]	AD1 (V)[b]	Data	Either[c]
7	Data 2 (V)[b]	Data 2 (V)[b]	Data 2 (V)[b]	AD2 (V)[b]	Data	Either[c]
8	Data 3 (V)[b]	Data 3 (V)[b]	Data 3 (V)[b]	AD3 (V)[b]	Data	Either[c]
9	Data 4 (V)[b]	Data 4 (V)[b]	Data 4 (V)[b]	AD4 (V)[b]	Data	Either[c]
10	Data 5 (V)[b]	Data 5 (V)[b]	Data 5 (V)[b]	AD5 (V)[b]	Data	Either[c]
11	Data 6 (V)[b]	Data 6 (V)[b]	Data 6 (V)[b]	AD6 (V)[b]	Data	Either[c]
12	Data 7 (V)[b]	Data 7 (V)[b]	Data 7 (V)[b]	AD7 (V)[b]	Data	Either[c]
13	Data 8 (V)[b]	Data 8 (V)[b]	Data 8 (V)[b]	AD8 (V)[b]	Data	Either[c]
14	nInit (L)	nInit (H)	nInit (L)	nReverseRequest (L)	Control	Host
15	nStrobe (L)	HostClk (V)	HostClk (C)	nWrite (L)	Control	Host
16	nSelectIn (L)	IEEE1284 Active (V)	IEEE1284 Active (V)	nAStrb (L)	Control	Host
17	nAutoFd (L or V[d])	HostBusy (V)	HostAck (V or C[e])	nDStrb (L)	Control	Host
18	Host Logic High[f] (V[g])				Control and/or Voltage[h]	Host
	Logic Ground[f]				Ground	
	Chassis Ground[f]				Ground	
19	Signal ground for Pin 1[f]				Ground	
20	Signal ground for Pin 2[f]				Ground	
21	Signal ground for Pin 3[f]				Ground	
22	Signal ground for Pin 4[f]				Ground	
23	Signal ground for Pin 5[f]				Ground	
24	Signal ground for Pin 6[f]				Ground	
25	Signal ground for Pin 7[f]				Ground	
26	Signal ground for Pin 8[f]				Ground	
27	Signal ground for Pin 9[f]				Ground	
28	Signal ground for Pin 10[f]				Ground	
29	Signal ground for Pin 11[f]				Ground	
30	Signal ground for Pin 12[f]				Ground	
31	Signal ground for Pin 13[f]				Ground	

TABLE 5. IEEE 1284-C pin assignments

Pin	Signal Name (Active State)[a] Compatibility Mode	Nibble/Byte Modes	ECP Mode	EPP Mode	Signal Function	Signal Source
32	Signal ground for Pin 14[f]				Ground	
33	Signal ground for Pin 15[f]				Ground	
34	Signal ground for Pin 16[f]				Ground	
35	Signal ground for Pin 17[f]				Ground	
36	Peripheral Logic High[f] (V[g])				Control and/or Voltage[h]	Peripheral

a. The active state of a signal is indicated in brackets after its name: L=low, H=high, V=varies low or high, C=closed loop or handshaking, M=manufacturer-specific.

b. Data bits are in order from 1 (least significant bit) to 8 (most significant bit).

c. “Either” because data signals are bi-directional; from host on forward path, from peripheral on reverse path.

d. Interpreted differently by different peripherals. On some printers, this bit may be set low by the host to indicate an automatic line feed. Sometimes this bit is used as a ninth data bit or a parity or command bit.

e. In forward path, this bit indicates whether the signals represent data or commands. In the reverse path, this bit is used on a handshake with PeriphClk (Pin 10).

f. These signals and grounds have the same name in all modes.

g. The peripheral sets this signal high to indicate the signals it drives are valid. It sets this signal low to indicate it is powered off or a dignal it drives is invalid.

h. Manufacturers can use this pin for its defined control function or to supply +5 VDC to a device.

Operating system support

Lava’s parallel port boards are tested and compatible with DOS, Windows 3.1/3.11, Windows 95/98/98SE/NT4/2000/Me/XP, and Linux kernel 2.4+. They have also been implemented successfully on a number of UNIX platforms.

The following tables summarize the configuration of parallel ports in the Windows environment. As can be seen, setting the various configuration parameters is handled differently from one version of Windows to another.

TABLE 6. Changing parallel port settings: PCI cards

TO CHANGE	Operating System: DOS	95	98	98SE	NT4	Me	2000	XP
address	address change not possible	Device Manager/ Safe Mode Device Manager	Device Manager/ Safe Mode Device Manager	Device Manager/ Safe Mode Device Manager	LavaPort Applet in Control Panel	Device Manager/ Safe Mode Device Manager	address change not possible	address change not possible
IRQ	IRQ change not possible	Device Manager/ Safe Mode Device Manager	Device Manager/ Safe Mode Device Manager	Device Manager/ Safe Mode Device Manager	IRQ change not possible	Device Manager/ Safe Mode Device Manager	Win 2000/XP do not assign an IRQ by default. An IRQ can be enabled, but will be assigned by Win 2000/XP and cannot be changed.	
LPT	LPT change not possible	Registry	Registry	Registry	LPT change not possible	Registry	Device Manager/ Safe Mode Device Manager	Device Manager/ Safe Mode Device Manager

TABLE 7. Changing parallel port settings: ISA cards

TO CHANGE	Operating System: DOS	95	98	98SE	NT4	ME	2000	XP
address	Jumper	Jumper + Device Manager[1]	Jumper + Device Manager[1]	Jumper + Device Manager[1]	Jumper + Device Manager[1]	Jumper + Device Manager[1]	Jumper + Device Manager[1]	Jumper + Device Manager[1]
IRQ	Jumper	Jumper + Device Manager[1]	Jumper + Device Manager[1]	Jumper + Device Manager[1]	Jumper + Device Manager[1]	Jumper + Device Manager[1]	Jumper + Device Manager[1]	Jumper + Device Manager[1]
DMA	Jumper	Jumper + Device Manager[1]	Jumper + Device Manager[1]	Jumper + Device Manager[1]	Jumper + Device Manager[1]	Jumper + Device Manager[1]	Jumper + Device Manager[1]	Jumper + Device Manager[1]
LPT	Jumper	Jumper + Device Manager[1]	Jumper + Device Manager[1]	Jumper + Device Manager[1]	Jumper + Device Manager[1]	Jumper + Device Manager[1]	Jumper + Device Manager[1]	Jumper + Device Manager[1]

[1] Changes made to jumper settings need to be updated in the Device Manager. Changes made in Device Manager need to matched to jumpers by physically setting the jumpers.

Lava: where parallels converge

Lava has a range of products with parallel ports: five parallel-only cards, four parallel/serial combo cards, and the Lava SPH-USB 1.1 hub.

TABLE 8. Lava parallel port products

Product	Ports; Interface
Parallel-PCI	1 EPP; PCI bus
Parallel-PCI/LP	1 EPP; PCI bus
Dual Parallel-PCI	2 EPP; PCI bus
Parallel Bi-directional	1 bi-directional; ISA bus
Parallel-ECP/EPP	1 ECP-EPP; ISA bus
2SP-550	1 bi-directional + 2 16550 serial; ISA bus
SP-PCI	1 bi-directional + 1 16550 serial; PCI bus
2SP-PCI	1 EPP + 2 16550 serial; PCI bus
LavaPort-Plus	1 EPP + 2 16650 serial; PCI bus
SPH-USB 1.1 Hub	1 bi-directional + 1 16550 serial + 3 USB 1.1; USB

In some cases, the speed or mode limitations of a bi-directional parallel port will limit the potential of a peripheral. The table below shows some situations where this may be the case.

TABLE 9. Product-to-peripheral matchup

Legend ✪ Optimal ● Possible limitation or incompatibility	Parallel-PCI	Parallel-PCI/LP	Dual Parallel-PCI	Parallel Bi-directional	Parallel-ECP/EPP	SP-PCI	2SP-PCI	LavaPort-Plus	2SP-ISA	SPH-USB 1.1 Hub
Parallel scanner	✪	✪	✪	●	✪	●	✪	✪	●	●
Parallel printer	✪	✪	✪	✪	✪	✪	✪	✪	✪	✪
Parallel ZIP™ drive	✪	✪	✪	●	✪	●	✪	✪	●	●
Parallel tape drive	✪	✪	✪	✪	✪	✪	✪	✪	✪	✪
Parallel Point-of-Sale device	✪	✪	✪	✪	✪	✪	✪	✪	✪	✪

A closer look: Lava's Parallel-PCI

Here is a great card. The Lava Parallel-PCI epitomizes what Lava products are all about: filling a need with the most simple-to-use and reliable product that can be built. Lava's Parallel-PCI, when it came to market, was the first PCI-bus parallel port card in the world. It is fully PCI compliant and takes advantage of the speed and easy configuration

of the PCI bus. When it appeared, it almost tripled the speed benchmarks of ISA-bus parallel port cards, as well as offering greater flexibility of configuration. Users were no longer restricted to the three standard parallel port addresses: the Parallel-PCI would automatically select a free address and IRQ during installation, virtually eliminating user error and resource conflicts.

Lava didn't stop there. More recent versions of the Parallel-PCI incorporate the Moko S-1 ASIC (application-specific integrated circuit). This is Lava's own integrated circuit, specifically designed for the Parallel-PCI and other Lava products. The Moko S-1 is unique to Lava, and consolidates a number of separate components into one chip. This adds efficiency in manufacture, and more importantly, virtually eliminates any possibility of component failure—another reason Lava confidently offers the Lava Lifetime Warranty.

FIGURE 5. Parallel-PCI

The popularity of the Lava Parallel-PCI has led to another Lava first: the first low-profile parallel PCI card. Functionally the same as the Parallel-PCI, the Parallel-PCI/LP (Low Profile) makes the advantages of the Lava Parallel-PCI available to users with systems using slimline or low-profile (Flex-ATX motherboard format) cases.

FIGURE 6. Parallel-PCI/LP

The Parallel-PCI also comes in a two-port version, the Dual Parallel-PCI.

FIGURE 7. Dual Parallel-PCI

Lava Computer MFG Inc., 2 Vulcan Street, Toronto, Ontario, CANADA M9W 1L2
Tel: + 416-674-5942 • Fax: +416-674-8262 • www.lavalink.com