

Informática II

Puerto paralelo

Gonzalo F. Pérez Paina



Universidad Tecnológica Nacional
Facultad Regional Córdoba
UTN-FRC

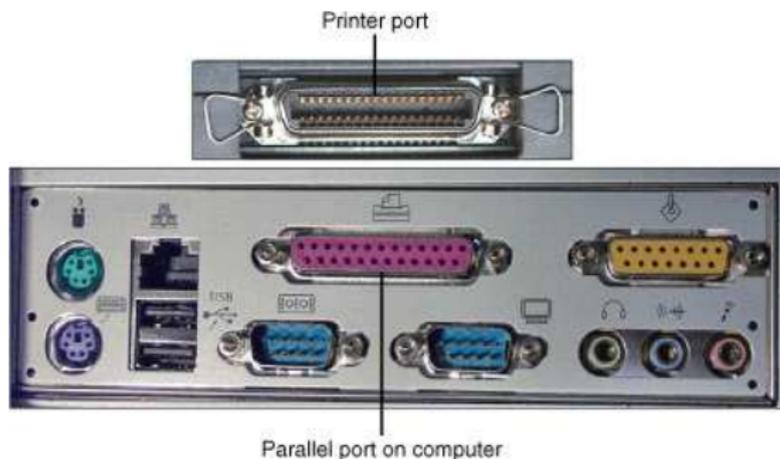
– 2018 –

Introducción - Historia y contexto

- ▶ Computadora personal (PC) de IBM en 1981 (puertos serie y paralelo)
- ▶ Las impresoras de entonces (matriz de puntos) se comunicaban por puerto serie
- ▶ Cuando las tecnologías avanzó surgió la necesidad mayor velocidad en la transferencia de datos
- ▶ En 1991 se arma una junta de fabricantes de impresoras (NPA, Network Printing Alliance)
- ▶ La NPA le solicita a IEEE formar un comité para desarrollar un nuevo estándar

Introducción - Historia y contexto

- ▶ Computadora personal (PC) de IBM en 1981 (puertos serie y paralelo)
- ▶ Las impresoras de entonces (matriz de puntos) se comunicaban por puerto serie
- ▶ Cuando las tecnologías avanzó surgió la necesidad mayor velocidad en la transferencia de datos
- ▶ En 1991 se arma una junta de fabricantes de impresoras (NPA, Network Printing Alliance)
- ▶ La NPA le solicita a IEEE formar un comité para desarrollar un nuevo estándar



Introducción - Historia y contexto

- ▶ Computadora personal (PC) de IBM en 1981 (puertos serie y paralelo)
- ▶ Las impresoras de entonces (matriz de puntos) se comunicaban por puerto serie
- ▶ Cuando las tecnologías avanzó surgió la necesidad mayor velocidad en la transferencia de datos
- ▶ En 1991 se arma una junta de fabricantes de impresoras (NPA, Network Printing Alliance)
- ▶ La NPA le solicita a IEEE formar un comité para desarrollar un nuevo estándar

El estándar **IEEE 1284**¹ define las diferentes versiones de **puertos paralelos**:

- ▶ SPP Mode (Standard Parallel Port), Compatibility/Centronix Mode
- ▶ Nibble Mode
- ▶ Byte Mode
- ▶ EPP Mode (Enhanced Parallel Port)
- ▶ ECP Mode (Extended Capabilities Port)

(todos compatibles hacia atrás)

¹Estándar del Método de Señalización para una Interfaz Paralela Bidireccional Periférica para Computadoras Personales (Marzo 1994)

Introducción - Historia y contexto

Conector tipo-D de 25 pines (17 de señal y 8 de GND)

- ▶ 8 líneas de datos
- ▶ 4 líneas de control
- ▶ 5 líneas de estado



(El estándar IEEE 1284 define tres conectores² –Tipo A, B, y C–)

²Female 25 pin D. Female 36 pin Centronics, mini-Centronics

Introducción - Historia y contexto

Conector tipo-D de 25 pines (17 de señal y 8 de GND)

- ▶ 8 líneas de datos
- ▶ 4 líneas de control
- ▶ 5 líneas de estado



(El estándar IEEE 1284 define tres conectores² –Tipo A, B, y C–)

USB/DB25, IEEE 1284



USB/DB36, IEEE 1284 (impresora)



²Female 25 pin D. Female 36 pin Centronics, mini-Centronics

Comunicación en paralelo

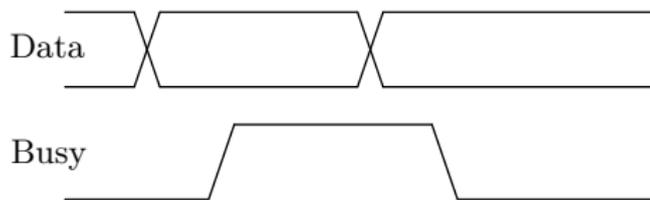
Ejemplo de comunicación en paralelo (handshake Centronics)



1. Escribir el byte en los bits de datos

Comunicación en paralelo

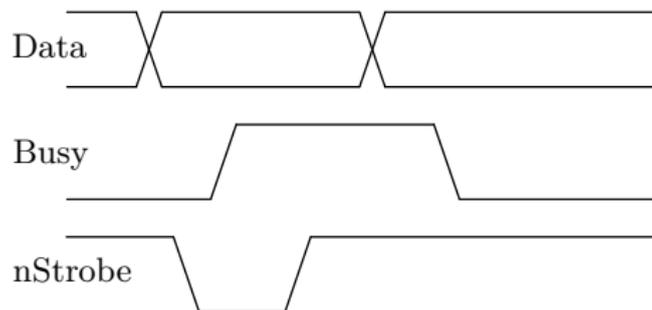
Ejemplo de comunicación en paralelo (handshake Centronics)



1. Escribir el byte en los bits de datos
2. Verificar si el dispositivo (impresora u otro) está ocupado (**Busy**)

Comunicación en paralelo

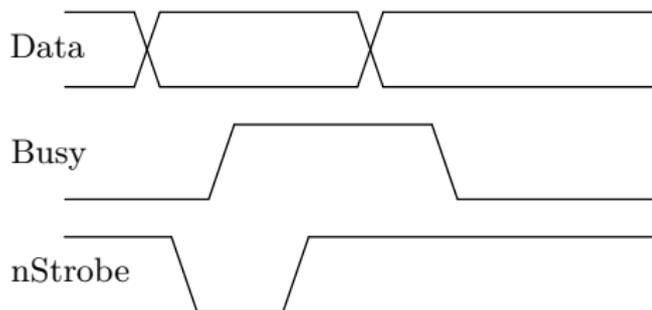
Ejemplo de comunicación en paralelo (handshake Centronics)



1. Escribir el byte en los bits de datos
2. Verificar si el dispositivo (impresora u otro) está ocupado (**Busy**)
3. Poner en nivel bajo la línea **Strobe**

Comunicación en paralelo

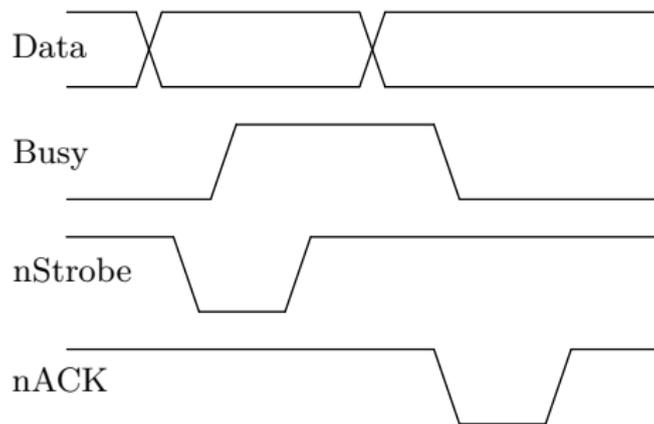
Ejemplo de comunicación en paralelo (handshake Centronics)



1. Escribir el byte en los bits de datos
2. Verificar si el dispositivo (impresora u otro) está ocupado (**Busy**)
3. Poner en nivel bajo la línea **Strobe**
4. Esperar $1\mu\text{s}$ y poner la línea **Strobe** en nivel alto

Comunicación en paralelo

Ejemplo de comunicación en paralelo (handshake Centronics)



1. Escribir el byte en los bits de datos
2. Verificar si el dispositivo (impresora u otro) está ocupado (**Busy**)
3. Poner en nivel bajo la línea **Strobe**
4. Esperar $1\mu\text{s}$ y poner la línea **Strobe** en nivel alto
5. La impresora acepta el dato (**ACK** $5\mu\text{s}$)

Comunicación en paralelo

Los puertos paralelos en modo EPP y ECP realizan el handshake por HW → mayor velocidad de transferencia

Comunicación en paralelo

Los puertos paralelos en modo EPP y ECP realizan el handshake por HW → mayor velocidad de transferencia

Velocidad de transferencia de datos

- ▶ SPP: entre 50 y 150KB/s
- ▶ EPP & ECP: entre 1 y 2 MB/s

Comunicación en paralelo

Los puertos paralelos en modo EPP y ECP realizan el handshake por HW → mayor velocidad de transferencia

Velocidad de transferencia de datos

- ▶ SPP: entre 50 y 150KB/s
- ▶ EPP & ECP: entre 1 y 2 MB/s

El ECP puede además utilizar DMA (Direct Memory Access) y buffers FIFO

Hardware – Conector y señales

Pin No (D-Type 25)	SPP Signal	Direc. I/O	Register
1	nStrobe	In/Out	Control
2	Data0	Out	Data
3	Data1	Out	Data
4	Data2	Out	Data
5	Data3	Out	Data
6	Data4	Out	Data
7	Data5	Out	Data
8	Data6	Out	Data
9	Data7	Out	Data
10	nAck	In	Status
11	Busy	In	Status
12	Paper-Out/End PaperEnd	In	Status
13	Select	In	Status
14	nAuto-Linefeed	In/Out	Control
15	nError/nFault	In	Status
16	nInitialize	In/Out	Control
17	nSelect-Printer nSelect-In	In/Out	Control
18-25	Ground	Gnd	

n: significa que la señal es activa en bajo (niveles TTL de tensión)

Direcciones de puertos y registros

Direcciones de los puertos

- ▶ 0x3BC-0x3BF: Antiguos puertos paralelos incluidos en placas de video (luego en motherboard).
- ▶ 0x378-0x37F: Dirección usual para LPT1^a
- ▶ 0x278-0x27F: Dirección usual para LPT2

^aLPT: Line PrinTing Device

Direcciones de puertos y registros

Direcciones de los puertos

- ▶ 0x3BC-0x3BF: Antiguos puertos paralelos incluidos en placas de video (luego en motherboard).
- ▶ 0x378-0x37F: Dirección usual para LPT1^a
- ▶ 0x278-0x27F: Dirección usual para LPT2

^aLPT: Line PrinTing Device

Registros

- ▶ Base+0: Registro de datos (Lectura/Escritura)
- ▶ Base+1: Registro de estado (Lectura)
- ▶ Base+2: Registro de control (Escritura)
- ▶ Base+3/7: Registros varios (para distintas implementaciones)

Registro de datos

Offset	Name	Read/Write	Bit No.	Properties
Base+0	Data Port	Write	Bit 7	Data 7 (pin 9)
			Bit 6	Data 6 (pin 8)
			Bit 5	Data 5 (pin 7)
			Bit 4	Data 4 (pin 6)
			Bit 3	Data 3 (pin 5)
			Bit 2	Data 2 (pin 4)
			Bit 1	Data 1 (pin 3)
			Bit 0	Data 0 (pin 2)

Registro de **escritura** (si se realiza una lectura se leerá el último byte escrito).

Registro de estado

Offset	Name	Read/Write	Bit No.	Properties
Base+1	Status Port	Read Only	Bit 7	Busy (pin 11)
			Bit 6	Ack (pin 10)
			Bit 5	Paper Out (pin 12)
			Bit 4	Select In (pin 13)
			Bit 3	Error (pin 15)
			Bit 2	IRQ (not)
			Bit 1	Reserved
			Bit 0	Reserved

Registro de **solo lectura**, cualquier dato que se escriba será ignorado.

Registro de control

Offset	Name	Read/Write	Bit No.	Properties
Base+2	Control Port	Read/Write	Bit 7	Unused
			Bit 6	Unused
			Bit 5	Enable bi-directional port
			Bit 4	Enable IRQ Via ACK line
			Bit 3	Select printer (pin 17)
			Bit 2	Initialize printer (pin 16)
			Bit 1	Auto Linefeed (pin 14)
			Bit 0	Strobe (pin 1)

Programación

Programación en GNU/Linux:

- ▶ Utilizando las funciones: `inb` y `outb` (permisos), o bien
- ▶ mediante manejo de archivos: `/dev/lp0`, `/dev/lp1`, `/dev/usb/lp1`, ...
(dispositivos de caracteres)

Programación

Programación en GNU/Linux:

- ▶ Utilizando las funciones: `inb` y `outb` (permisos), o bien
- ▶ mediante manejo de archivos: `/dev/lp0`, `/dev/lp1`, `/dev/usb/lp1`, ... (dispositivos de caracteres)

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <sys/io.h>
4
5 #define BASE 0x378 //LPT1
6
7 int main(void)
8 {
9     if(ioperm(BASE, 1, 1))
10         fprintf(stderr, "Couldn't open parallel port"), exit(1);
11
12     outb(255, BASE); //Set all pins hi
13     sleep(5);
14     outb(0, BASE); //Set all pins lo
15
16     return 0;
17 }
```

Programación

Programación en GNU/Linux:

- ▶ Utilizando las funciones: `inb` y `outb` (permisos), o bien
- ▶ mediante manejo de archivos: `/dev/lp0`, `/dev/lp1`, `/dev/usb/lp1`, ... (dispositivos de caracteres)

```
1 #include <fcntl.h>
2 #include <unistd.h>
3
4 int main(void)
5 {
6     char data = '0';
7     int fd = open("/dev/lp0", O_WRONLY);
8
9     if(fd < 0)
10         return -1;
11
12     /* Write data port lines */
13     write(fd, &data, 1);
14
15     close(fd);
16     return 0;
17 }
```

- ▶ *Parallel Port Complete. Programming, Interfacing, & Using the PC's Parallel Printer Port*, Jan Axelson (Book)
- ▶ *Interfacing the Standard Parallel Port –SPP–* (web)
- ▶ *Interfacing the Enhanced Parallel Port –EPP–* (web)
- ▶ *Interfacing the Extended Capabilities Parallel Port –ECP–* (web)

