

UNIDADES FUNCIONALES DEL COMPUTADOR

Tema 5

Principios de Computadoras

Temas

Introducción

1. Bus
2. Memoria
3. Entrada/Salida
4. Unidad aritmética-lógica
5. Unidad de Control

Introducción

- Un computador está formado por la CPU, la memoria y las unidades E/S, con uno o varios módulos de cada tipo.
- Estos componentes se interconectan (con el bus) para llevar a cabo la función básica del computador (ejecutar programas).
- Los ordenadores actuales siguen el modelo de Von Neumann:
 1. Datos e instrucciones se almacenan en la misma memoria
 2. Los contenidos en memoria se direccionan indicando su posición.
 3. La ejecución del programa se realiza instrucción a instrucción.

1. BUS

Tema 5

Principios de Computadoras

1. Bus

- Introducción
- Tipos de buses
- Evolución de los buses
- Problemas de comunicación
- Buses síncronos y asíncronos
- Partes del bus
- Triestado
- Arbitraje del bus
- Diagramas de tiempo
- Ejemplos de temporización

Introducción

- El bus permite la transferencia de información entre las distintas unidades funcionales del computador.
- Para que todo funcione correctamente es necesario:
 - Un **bus** de comunicación **interno de la CPU**
 - Un **bus** de comunicación de la **CPU con la memoria**
 - Un **bus** que comunique los dispositivos de **E/S con la CPU**
- La dificultad en la transferencia de información entre los componentes de un ordenador depende de las distancias físicas que separan los componentes.

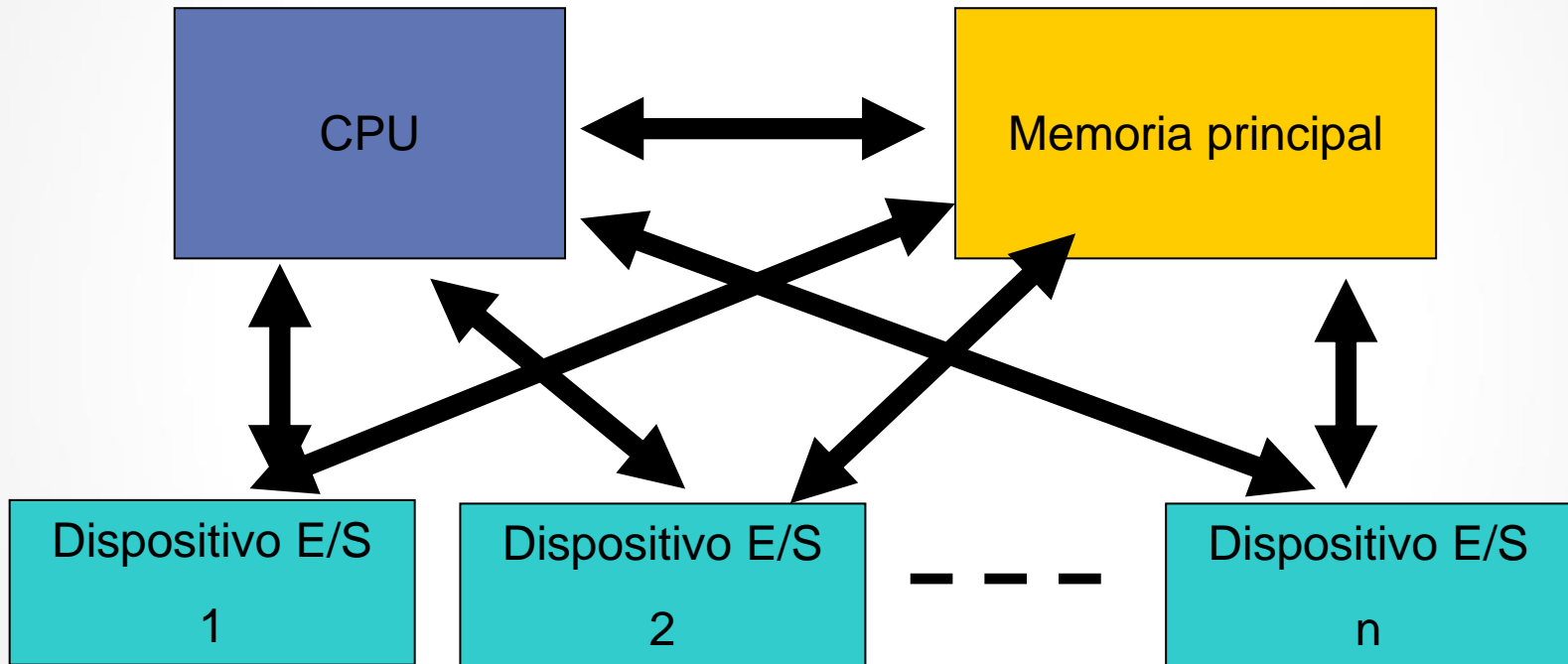
¿Cómo se comunica el ordenador con el mundo exterior?

- **Comunicación Externa:** Implica comunicaciones en distancias del orden de un metro. Se utiliza más la **comunicación serie** que la paralela por ser más económica y simple de controlar.
- **Redes:** Implica comunicaciones no entre partes del ordenador entre sí, sino entre ordenadores muy distantes.
 - LAN (Local Area Network): Ordenadores situados dentro del mismo edificio.
 - MAN (Metropolitan Area Network): Ordenadores situados dentro de la misma ciudad.
 - WAN (Wide Area Network): Ordenadores situados en distintas partes del mundo.

¿Cómo se comunican las distintas partes del ordenador entre sí?

- **Comunicación interna:** Es la que ocurre dentro del computador e involucra transmisiones de información no superiores a un metro.
- La comunicación interna se implementa mediante grupos de cables denominados **buses**, que permiten la **transferencia de información en paralelo** (palabra a palabra). Un bus no está constituido sólo por las líneas de comunicación, sino también por los mecanismos necesarios para controlar el acceso a éstas y supervisar el intercambio de señales.

¿Por qué es necesario el bus?



¿Por qué es necesario el bus?

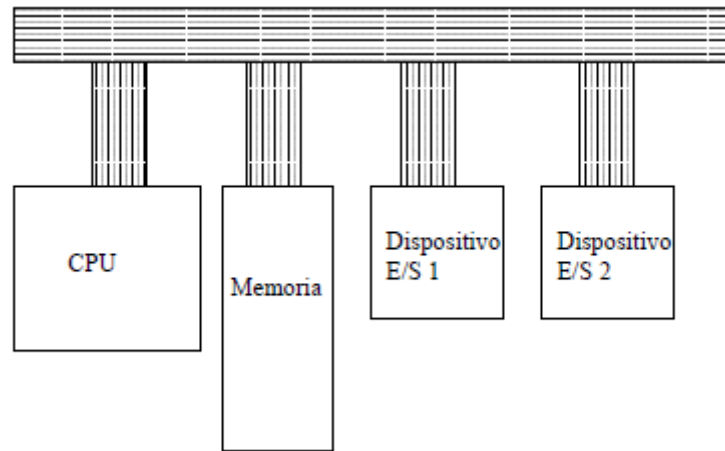
- **Organización punto a punto:** existe un camino de comunicación para cada transferencia posible.
- Hay un cable para cada par de componentes que necesitan comunicarse. Sólo es necesario enviar el dato y no hace falta la dirección del dispositivo al que va dirigido.
- Es una organización muy eficiente porque permite realizar muchos intercambios de información de forma paralela a lo largo de distintos caminos.

¿Problema?

Tipos de buses

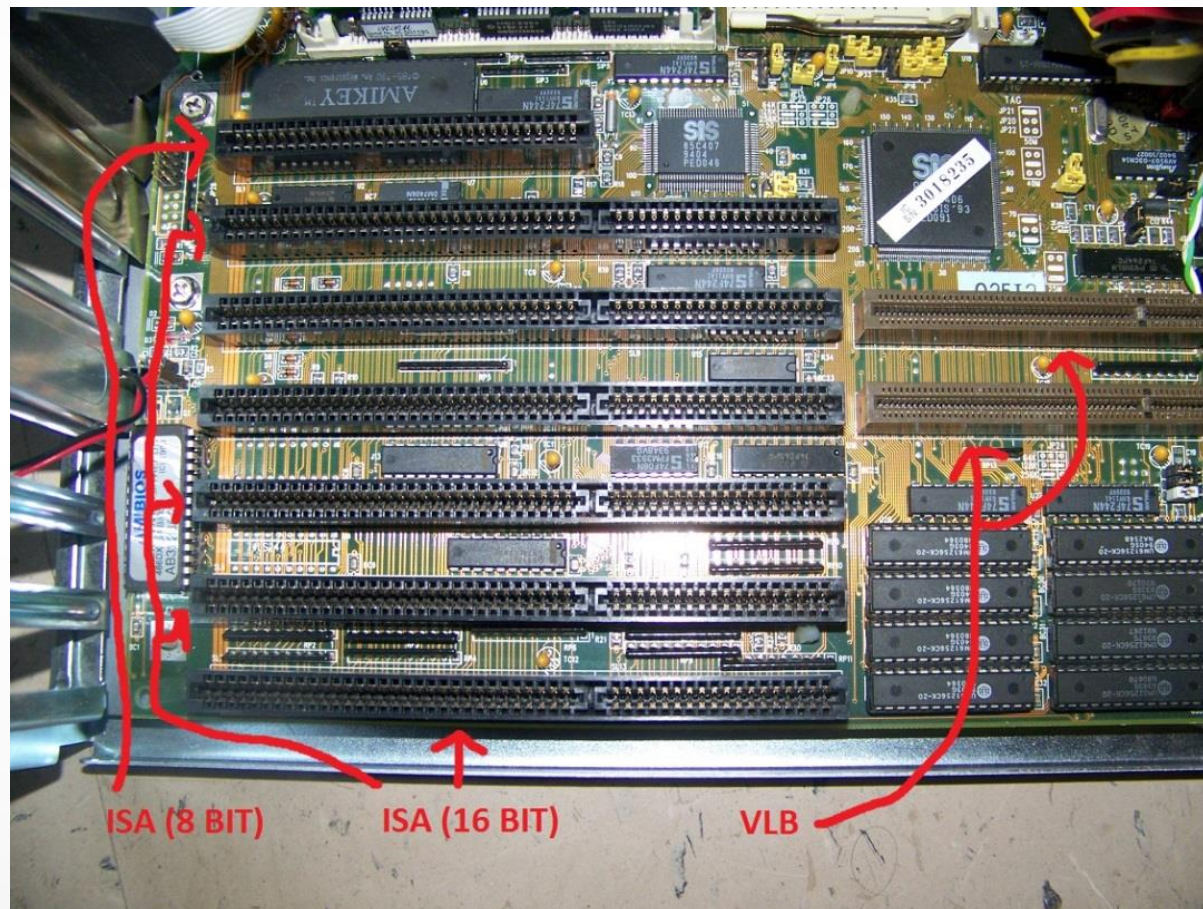
1. Bus simple compartido (Backplane):

El bus del sistema está compartido por todos los componentes. Sólo dos unidades pueden comunicarse por medio del bus del sistema en un determinado momento.

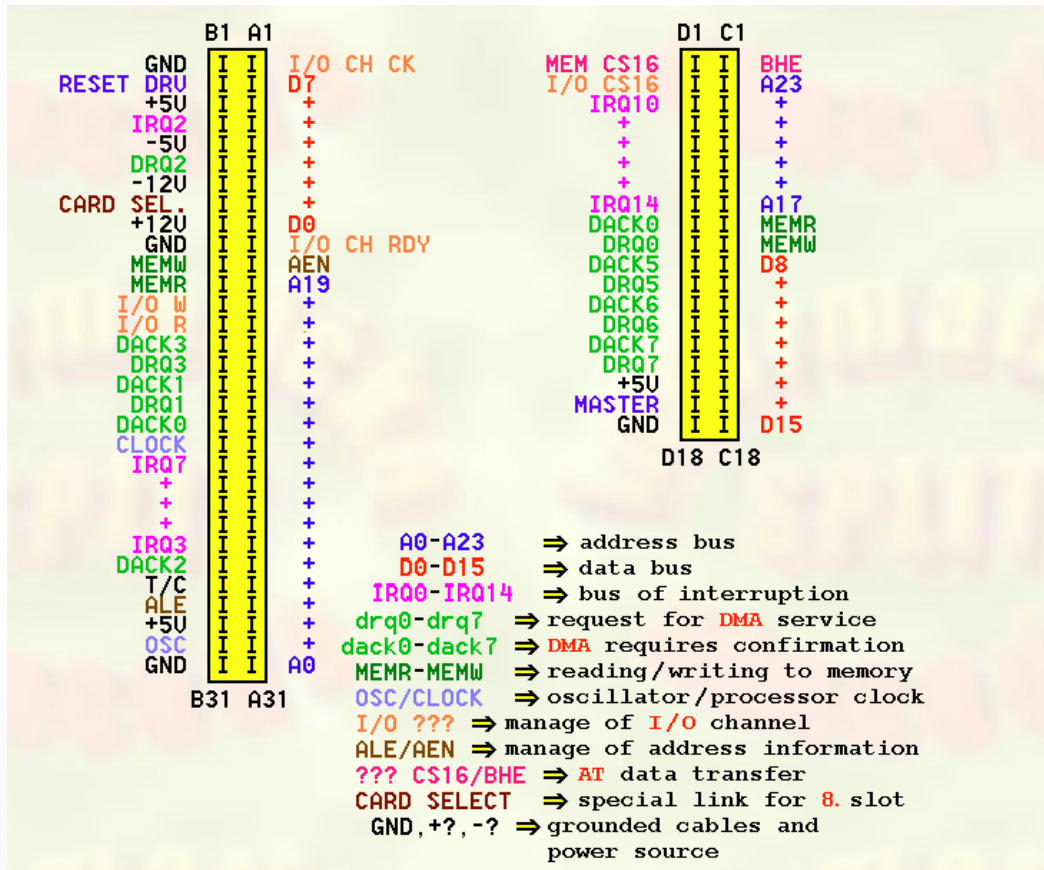


Recibe el nombre de Backplane porque normalmente se construyen en el plano posterior (sobre la placa base). Las placas con el procesador, la memoria y los dispositivos de E/S se conectan en el backplane.

ISA BUS & VESA BUS



Contactos de bus ISA



Tipos de buses

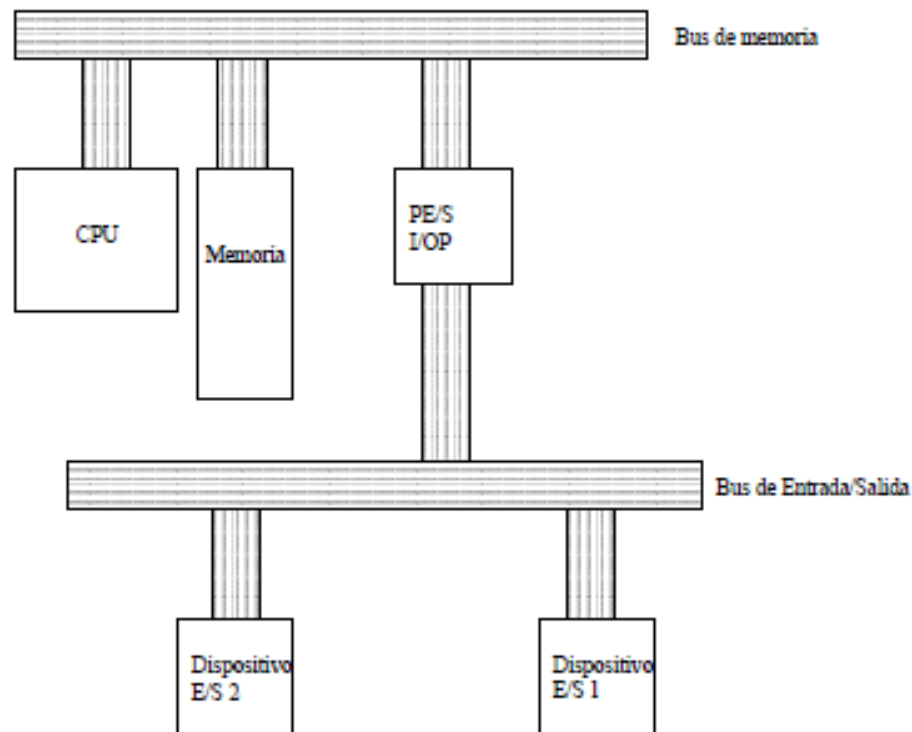
2. Bus de memoria y E/S separados

Dos buses: uno es para la comunicación entre CPU y memoria, y otro para la comunicación entre CPU y E/S. Ej: grandes computadores

- Bus procesador-memoria: de longitud pequeña, alta velocidad y adaptados a los sistemas de memoria.
- Bus procesador-E/S: pueden tener longitudes grandes y muchos tipos de dispositivos conectados entre sí

Tipos de buses

2. Bus de memoria y E/S separados



Evolución de los buses

PRIMERA GENERACIÓN

Los primeros ordenadores se basaban en la conexión de varias tarjetas a un bus tipo backplane.

ISA (Industry Standard Architecture). Fue creado por IBM en 1980 para ser empleado en los IBM PC (20 bits direcciones, 8 bits datos, 4.77 MHz) y se extendió en 1983 como "Bus XT". En 1984 se introduce un nuevo estándar, el "Bus AT" (16 bits datos, 24 bits direcciones, 8 MHz).

MCA (Micro Channel Architecture). Introducido por IBM en 1987 para reemplazar el bus ISA. Es mucho más avanzado que el ISA (32 bits, mejor calidad de señal, otro conector para tarjeta gráfica) pero incompatible.

EISA (Extended Industry Standard Architecture). Fue desarrollado en 1989 por el grupo de los nueve (AST, Compaq, Epson, Hewlett Packard, NEC, Olivetti, Tandy, Wyse y Zenith Data Systems). Amplía la arquitectura del bus ISA a 32 bits y permite que más de una CPU comparta el bus. Es compatible con ISA (se puede conectar un dispositivo ISA).

IBM PC MCA Connector Pin Out				
Pin	J1		J2	
#	A Row	B Row	C Row	D Row
1	VSYNC	ESYNC	+5V	D8
2	HSYNC	GND	D10	D9
3	BLANC	P5	D11	GND
4	GND	P4	D13	D12
5	P6	P3	+12V	D14
6	EDCLK	GND	Reserved	D15
7	DCLK	P2	SBHE	GND
8	GND	P1	CD DS 16	IRQ10
9	P7	P0	+5V	IRQ11
10	EVIDEO	GND	IRQ14	IRQ12
11	CD SETUP	Audio GND	IRQ15	GND
12	MADE24	Audio		
13	GND	GND		
14	A11	Oscillator		
15	A10	GND		
16	A9	A23		
17	+5V	A22		
18	A8	A21		
19	A7	GND		
20	A6	A20		
21	+5V	A19		
22	A5	A18		
23	A4	GND		
24	A3	A17		
25	+5V	A16		
26	A2	A15		
27	A1	GND		
28	A0	A14		
29	+12V	A13		
30	ADL	A12		
31	PREEMPT	GND		
32	BURST	IRQ9		
33	+12V	IRQ3	Pins Not Used	Pins Not Used
34	ARB0	IRQ4		
35	ARB1	GND		
36	ARB2	IRQ5		
37	+12V	IRQ6		
38	ARB3	IRQ7		
39	ARB GNT	GND		
40	TC	Reserved		
41	+5V	Reserved		
42	S0	CHCK		
43	S1	GND		
44	M10	CMD		
45	+12V	CHROYRTN		
46	CD CHRDY	CD SFDBK		
47	D0	GND		
48	D2	D1		
49	+5V	D3		
50	D5	D4		
51	D6	GND		
52	D7	CHRESET		
53	GND	Reserved		
54	DS 16 RIN	Reserved		
55	REFRESH	GND		

IBM Microchannel Bus

SCSI Bus

SCSI P Connector Single-Ended			
Pin	Signal Name	Pin	Signal Name
1	GND	35	D12-
2	GND	36	D13-
3	GND	37	D14-
4	GND	38	D15-
5	GND	39	DPARH-
6	GND	40	D0-
7	GND	41	D1-
8	GND	42	D2-
9	GND	43	D3-
10	GND	44	D4-
11	GND	45	D5-
12	GND	46	D6-
13	GND	47	D7-
14	GND	48	DPARITY-
15	GND	49	GND
16	GND	50	GND
17	TRMPWR	51	TRMPWR
18	TRMPWR	52	TRMPWR
19	RESERVED	53	RESERVED
20	GND	54	GND
21	GND	55	ATN-
22	GND	56	GND
23	GND	57	BSY-
24	GND	58	ACK-
25	GND	59	RST-
26	GND	60	MSG-
27	GND	61	SEL-
28	GND	62	C/D-
29	GND	63	REQ-
30	GND	64	I/O-
31	GND	65	D8-
32	GND	66	D9-
33	GND	67	D10-
34	GND	68	D11-

www.interfacebus.com

Evolución de los buses

SEGUNDA GENERACIÓN

Aparece el “bus del sistema” (CPU-memoria) y el “bus de expansión” (CPU-E/S)

VESA (Video Electronics Standards Association). Fue desarrollado en 1992 para equipos basados en el 80486 y permitía conectar la tarjeta gráfica directamente al procesador, lo que aumentaba el rendimiento en gráficos. Era compatible con el bus ISA, pero sus tarjetas de expansión eran muy grandes, lo que motivó que fuera sustituido por el PCI.

SCSI (Small Computer System Interface). Es un interfaz estándar que fue aprobado por el ANSI en 1986. Comenzó su uso en el Commodore Amiga, Apple Macintosh y Servidores Sun. Nunca fue muy utilizado en los IBM PC por el menor precio y buen rendimiento de los discos ATA (interface de transferencia de datos entre CPU y memoria).

Evolución de los buses

SEGUNDA GENERACIÓN

Aparecen los buses “plug and play”, donde ya no era necesario estar configurando el equipo cada vez que se conectaba un dispositivo.

PCI (Peripheral Component Interconnect) Es un bus backplane de propósito general. PCI 2.0 fue lanzado en 1993 y PCI 2.1 en 1995. Reemplazó rápidamente el bus en los servidores, pero tardó cierto tiempo en reemplazar al VESA local bus de los PC (después de 1994). **El PCI permite la configuración dinámica de un dispositivo periférico (Plug and play).**

USB Fue desarrollado en 1996 por siete empresas (IBM, Intel, Northern Telecom, Compaq, Microsoft, Digital Equipment Corporation y NEC). **Mejora las capacidades plug and play, eliminando la necesidad de reiniciar el ordenador al conectar o desconectar un dispositivo.** Para dispositivos multimedia como escáneres y cámaras multimedia se ha convertido en el estándar de conexión. Dado que **es más lento que el resto de buses**, es muy difícil que reemplace a los actuales buses para discos duros.

PCI express (derecha)

Table 7. Connector Pinout

Pin #	Signal Name	Pin #	Signal Name	Pin #	Signal Name	Pin #	Signal Name
61	PWR_SRC	82	GND	73	PEX_TX5#	74	GND
1	PWR_SRC	7	1VBUN	75	PEX_TX6#	76	PEX_TX5
3	PWR_SRC	4	1VBUN	77	GND	78	PEX_TX5
5	PWR_SRC	6	1VBUN	79	PEX_TX6#	80	GND
7	PWR_SRC	8	1VBUN	81	PEX_TX6#	82	PEX_TX6#
9	PWR_SRC	10	1VBUN	83	GND	84	PEX_TX6#
11	PWR_SRC	12	1VBUN	85	PEX_TX6#	86	GND
13	PWR_SRC	14	1VBUN	87	PEX_TX6#	88	PEX_TX6#
15	PWR_SRC	16	1VBUN	89	GND	90	PEX_TX6#
17	GND	18	1VBUN	91	PEX_TX6#	92	GND
19	GND	20	GND	93	PEX_TX6#	94	PEX_TX6#
21	GND	22	GND	95	GND	96	PEX_TX6#
23	GND	24	GND	97	PEX_TX6#	98	GND
25	KEY	26	KEY	99	PEX_TX6#	100	PEX_TX6#
27	KEY	28	KEY	101	GND	102	PEX_TX6#
29	KEY	30	KEY	103	PEX_TX6#	104	GND
31	KEY	32	KEY	105	PEX_TX6#	106	PEX_TX6#
33	KEY	34	KEY	107	GND	108	PEX_TX6#
35	KEY	36	KEY	109	PEX_TX6#	110	GND
37	PEX_TX6#	38	PEX_TX6#	111	PEX_TX6#	112	PEX_TX6#
39	PEX_TX6#	40	PEX_TX6#	113	GND	114	PEX_TX6#
41	PEX_TX6#	42	PEX_TX6#	115	PEX_TX6#	116	GND
43	PEX_TX6#	44	GND	117	PEX_TX6#	118	PEX_TX6#
45	PEX_TX6#	46	PEX_TX6#	119	GND	120	PEX_TX6#
47	GND	48	PEX_TX6#	121	PEX_TX6#	122	PEX_TX6#
49	PEX_TX6#	50	GND	123	PEX_TX6#	124	PEX_TX6#
51	PEX_TX6#	52	PEX_TX6#	125	GND	126	PEX_TX6#
53	GND	54	PEX_TX6#	127	PEX_TX6#	128	GND
55	PEX_TX6#	56	GND	129	PEX_TX6#	130	PEX_TX6#
57	PEX_TX6#	58	PEX_TX6#	131	GND	132	PEX_TX6#
59	GND	60	PEX_TX6#	133	PEX_TX6#	134	PEX_TX6#
61	PEX_TX6#	62	GND	135	PEX_TX6#	136	PEX_TX6#
63	PEX_TX6#	64	PEX_TX6#	137	CLK_REQ#	138	GND
65	GND	66	PEX_TX6#	139	PEX_TX6#	140	PEX_TX6#
67	PEX_TX6#	68	GND	141	HDA_SYNC	142	GND
69	PEX_TX6#	70	PEX_TX6#	143	HDA_CLK	144	PEX_TX6#
71	GND	72	PEX_TX6#				

Pin #	Signal Name	Pin #	Signal Name	Pin #	Signal Name	Pin #	Signal Name
145	SMB_DAT	146	HDA_RST#	185	DP_B_L0#	186	LVDS_LTX3#
147	SMB_CLK	148	VGA_RSD	187	DP_B_L0	188	LVDS_LTX3
149	Thermal#	150	GND	189	GND	190	DP_B_HPD
151	VGA_HVINC	152	VGA_GRN	201	DVI_B_TX2# / DP_A_L0#	202	LVDS_LTX2# / DVI_C_TX2#
153	VGA_VINC	154	GND	203	DVI_B_TX2 / DP_A_L1	204	LVDS_LTX2 / DVI_C_TX2
155	DOCA_CLK	156	VGA_BLD	205	GND	206	GND
157	DOCA_DAT	158	GND	207	DVI_B_TX1# / DP_A_L1#	208	LVDS_LTX1# / DVI_C_TX1#
159	HDA_SDI	160	LVDS_LCLK#	209	DVI_B_TX1 / DP_A_L1	210	LVDS_LTX1 / DVI_C_TX1
161	HDA_SDO	162	LVDS_LCLK	211	GND	212	GND
163	GND	164	GND	213	DVI_B_TX0# / DP_A_L2#	214	LVDS_LTX0# / DVI_C_TX0#
165	DP_B_L3#	166	LVDS_LTX0#	215	DVI_B_TX0 / DP_A_L2	216	LVDS_LTX0 / DVI_C_TX0
167	DP_B_L3	168	LVDS_LTX0	217	DVI_B_HPD	218	GND
169	AC/BATT#	170	SPDIF	219	DVI_A_CLK#	220	DOCK_DAT
171	DP_A_AUX#	172	LVDS_LTX2# / DVI_C_TX2#	221	DVI_A_CLK	222	DOCK_CLK
173	DP_A_AUX	174	LVDS_LTX2 / DVI_C_TX2	223	GND	224	LVDS_PPEN
175	GND	176	GND	225	DVI_A_TX2#	226	LVDS_BL_BPHI
177	DP_B_L2#	178	LVDS_LTX0# / DVI_C_TX0#	227	DVI_A_TX2	228	LVDS_BLEN
179	DP_B_L2	180	LVDS_LTX0 / DVI_C_TX0	229	GND	230	DOCK_DAT
181	GND	182	GND	231	DVI_A_TX0#	232	DOCK_CLK
183	DP_B_L1#	184	LVDS_LTX0# / DVI_C_TX0#	233	DVI_A_TX1	234	DVI_RUN
185	DP_B_L1	186	LVDS_LTX0 / DVI_C_TX0	235	GND	236	GND
187	GND	188	GND	237	DVI_A_TX0#	238	DVI_RUN
189	DVI_B_CLK# / DP_A_L3#	190	LVDS_LCLK# / DVI_C_TXC#	239	DVI_A_TX0	240	DVI_RUN
191	DVI_B_CLK / DP_A_L3	192	LVDS_LCLK / DVI_C_TXC	241	GND	242	DVI_RUN
193	DVI_B_HPD / DVI_C_HPD / DP_A_HPD	194	GND				

PCI-Express 16x Connector Pin-Out

Pin #	Name	Side B Connector	Description	Name	Side A Connector	Description
1	+12v		+12 volt power	PRST#		Hot plug present detect
2	+12v		+12 volt power	+12v		+12 volt power
3	RSVD		Reserved	+12v		+12 volt power
4	GND		Ground	GND		Ground
5	SMCLK		SMBus clock	JTAG2		TCK
6	SMDAT		SMBus data	JTAG3		TDI
7	GND		Ground	JTAG4		TDO
8	+3.3v		+3.3 volt power	JTAG5		TMS
9	JTAG1		+TRST#	+3.3v		+3.3 volt power
10	3.3Vaux		3.3v volt power	+3.3v		+3.3 volt power
11	WAKE#		Link Reactivation	PWRGD		Power Good
Mechanical key						
12	RSVD		Reserved	GND		Ground
13	GND		Ground	REFCLK#		Reference Clock
14	HS0[0:1]		Transmitter Lane 0, Differential pair	REFCLK#		Reference Clock
15	HS0[0:1]		Ground	GND		Ground
16	GND		Ground	HS1[0:1]		Receiver Lane 0, Differential pair
17	PRST#		Hotplug detect	HS1[0:1]		Receiver Lane 0, Differential pair
18	GND		Ground	RSVD		Reserved
19	HS0[0:1]		Transmitter Lane 1, Differential pair	RSVD		Reserved
20	HS0[1:1]		Ground	HS1[1:1]		Receiver Lane 1, Differential pair
21	GND		Ground	HS1[1:1]		Receiver Lane 1, Differential pair
22	GND		Ground	GND		Ground
23	HS0[0:1]		Transmitter Lane 2, Differential pair	GND		Ground
24	HS0[0:1]		Ground	GND		Ground
25	GND		Ground	HS1[0:1]		Receiver Lane 2, Differential pair
26	GND		Ground	HS1[0:1]		Receiver Lane 2, Differential pair
27	HS0[0:1]		Transmitter Lane 3, Differential pair	GND		Ground
28	HS0[0:1]		Ground	GND		Ground
29	GND		Ground	HS1[0:1]		Receiver Lane 3, Differential pair
30	RSVD		Reserved	HS1[0:1]		Receiver Lane 3, Differential pair
31	PRST#		Hot plug detect	GND		Ground
32	GND		Ground	RSVD		Reserved
33	HS0[0:1]		Transmitter Lane 4, Differential pair	RSVD		Reserved
34	HS0[0:1]		Ground	GND		Ground
35	GND		Ground	HS1[0:1]		Receiver Lane 4, Differential pair
36	GND		Ground	HS1[0:1]		Receiver Lane 4, Differential pair
37	HS0[0:1]		Transmitter Lane 5, Differential pair	GND		Ground
38	HS0[0:1]		Ground	GND		Ground
39	GND		Ground	HS1[0:1]		Receiver Lane 5, Differential pair
40	GND		Ground	HS1[0:1]		Receiver Lane 5, Differential pair
41	HS0[0:1]		Transmitter Lane 6, Differential pair	GND		Ground
42	HS0[0:1]		Ground	GND		Ground
43	GND		Ground	HS1[0:1]		Receiver Lane 6, Differential pair
44	GND		Ground	HS1[0:1]		Receiver Lane 6, Differential pair
45	HS0[0:1]		Transmitter Lane 7, Differential pair	GND		Ground
46	HS0[0:1]		Ground	GND		Ground
47	GND		Ground	HS1[0:1]		Receiver Lane 7, Differential pair
48	PRST#		Hot plug detect	HS1[0:1]		Receiver Lane 7, Differential pair
49	GND		Ground	GND		Ground
50	HS0[0:1]		Transmitter Lane 8, Differential pair	RSVD		Reserved
51	HS0[0:1]		Ground	GND		Ground
52	GND		Ground	HS1[0:1]		Receiver Lane 8, Differential pair
53	GND		Ground	HS1[0:1]		Receiver Lane 8, Differential pair
54	HS0[0:1]		Transmitter Lane 9, Differential pair	GND		Ground
55	HS0[0:1]		Ground	GND		Ground
56	GND		Ground	HS1[0:1]		Receiver Lane 9, Differential pair
57	GND		Ground	HS1[0:1]		Receiver Lane 9, Differential pair
58	HS0[0:1]		Transmitter Lane 10, Differential pair	GND		Ground
59	HS0[0:1]		Ground	GND		Ground
60	GND		Ground	HS1[0:1]		Receiver Lane 10, Differential pair
61	GND		Ground	HS1[0:1]		Receiver Lane 10, Differential pair
62	HS0[0:1]		Transmitter Lane 11, Differential pair	GND		Ground
63	HS0[0:1]		Ground	GND		Ground
64	GND		Ground	HS1[0:1]		Receiver Lane 11, Differential pair
65	GND		Ground	HS1[0:1]		Receiver Lane 11, Differential pair
66	HS0[0:1]		Transmitter Lane 12, Differential pair	GND		Ground
67	HS0[0:1]		Ground	GND		Ground
68	GND		Ground	HS1[0:1]		Receiver Lane 12, Differential pair
69	GND		Ground	HS1[0:1]		Receiver Lane 12, Differential pair
70	HS0[0:1]		Transmitter Lane 13, Differential pair	GND		Ground
71	HS0[0:1]		Ground	GND		Ground
72	GND		Ground	HS1[0:1]		Receiver Lane 13, Differential pair
73	GND		Ground	HS1[0:1]		Receiver Lane 13, Differential pair
74	HS0[0:1]		Transmitter Lane 14, Differential pair	GND		Ground
75	HS0[0:1]		Ground	GND		Ground
76	GND		Ground	HS1[0:1]		Receiver Lane 14, Differential pair
77	GND		Ground	HS1[0:1]		Receiver Lane 14, Differential pair
78	HS0[0:1]		Transmitter Lane 15, Differential pair	GND		Ground
79	HS0[0:1]		Ground	GND		Ground
80	GND		Ground	HS1[0:1]		Receiver Lane 15, Differential pair
81	PRST#		Hot plug present detect	HS1[0:1]		Receiver Lane 15, Differential pair
82	RSVD#		Hot Plug Detect	GND		Ground

Evolución de los buses

TERCERA GENERACIÓN

Buses muy adaptables. Se pueden usar tanto como buses internos como para conectar distintas máquinas entre sí. Esto origina problemas complejos de arbitraje de bus y hace necesario el uso de protocolos más parecidos a los protocolos de red.

Hypertransport Fue introducido en Abril de 2001. Es un enlace punto a punto de baja latencia bidireccional de gran ancho de banda serie/paralelo. Sus velocidades van de 200 MHz a 3 GHz (51.2 GB/s).

Infiniband Es un enlace serial punto a punto de altas prestaciones. Su ritmo de transferencia va de 2 GB/s a 300 GB/s.

Problemas de comunicación

- Retardos entre dos señales transmitidas simultáneamente pueden originar que lleguen fuera de fase al destino.
Ej: retardo de 10% implica que la señal llegará 33ns fuera de fase.
- Algunos buses utilizan más de un reloj para sincronizar dispositivos distantes. Uno en el controlador o master y otro en cada dispositivo.
- Como cada componente tiene su ritmo de transferencia distinto, sus circuitos de temporización deben ser independientes (comunicación asíncrona).
- Es necesario determinar el número y el tipo de señales a utilizar durante la comunicación interna del sistema.

Problemas de comunicación

- Un dispositivo puede ser seleccionado para la conexión al bus de dos modos distintos:
 - 1) El maestro lo selecciona (ej: siguiendo una instrucción de un programa)
 - 2) El propio esclavo puede pedir el acceso al bus enviando una señal de petición de bus (**BUS REQUEST**) al controlador del bus.

Si varias unidades quieren acceder al bus simultáneamente, el controlador del bus debe tener un método para seleccionar una de estas unidades. Este proceso se llama **ARBITRAJE DEL BUS**.

Buses síncronos y asíncronos

BUS SÍNCRONO

- Cada ítem es transferido durante un periodo de tiempo conocido con antelación tanto para la unidad fuente como para la unidad destino. Circuitos de interface de bus de ambas unidades están sincronizados mediante señales de sincronización emitidas periódicamente.
- La comunicación síncrona tiene la desventaja de que los periodos de tiempo utilizados para la transferencia de información están altamente determinados por las unidades más lentas del sistema. Los dispositivos rápidos no son capaces de comunicarse a su máxima velocidad.

Buses síncronos y asíncronos

BUSES SÍNCRONOS SERIALES

- El bus **RS-232** es un bus síncrono.
- El Bus **SPI** (del inglés Serial Peripheral Interface) es un estándar de comunicaciones, usado principalmente para la transferencia de información entre circuitos integrados en equipos electrónicos. *Los dispositivos clientes usan el reloj que envía el servidor, no necesitan por tanto su propio reloj.*
- Microwire, es un predecesor del bus SPI (National Semiconductor). Es un subconjunto estricto de SPI. Suele utilizar velocidades de reloj menores que SPI. Por ejemplo, 2 MHz vs. 20 MHz.
- El circuito inter-integrado (I2C) es un protocolo síncrono para conectar los dispositivos a la placa base.

Buses síncronos y asíncronos

BUSES SÍNCRONOS PARALELOS

- El bus ISA era un bus paralelo, síncrono con el reloj de la CPU.
- El bus PCI es un bus paralelo, síncrono con un solo reloj del bus.

Buses síncronos y asíncronos

BUS ASÍNCRONO

- Cada ítem transferido es acompañado por una señal de control separada para indicar su presencia a la unidad de destino. La unidad de destino puede responder con otra señal de control para reconocer la recepción de la información. Cada dispositivo puede generar las señales de control a su propio ritmo -> la velocidad de transmisión varía según dispositivo.
- Esta flexibilidad en los ritmos de transmisión se consigue a costa de una circuitería de control del bus más compleja.

Buses síncronos y asíncronos

BUSES ASÍNCRONOS

- Bus RS-232 puede ser síncrono o asíncrono.
- Bus SERIAL ATA
- Bus SERIAL ATTACHED SCSI
- Bus FIREWIRE (1394)
- Bus USB 2.0 (Universal Serial Bus)
- Bus PCI Express

Partes del bus

Líneas de datos

Para transmitir en paralelo todos los bits de una palabra de n bits. Tamaño múltiplo de 8 (8, 16, 32, 64).

Líneas de direcciones

Para identificar una unidad o una dirección de memoria para ser utilizada en la transferencia de datos, y por tanto, darle acceso al bus.

Líneas de control

Para transferir las señales de temporización y la información de estado de las unidades del sistema.

Partes del bus

- Es posible usar las líneas de datos para transferir tanto datos y direcciones: **MULTIPLEXACIÓN de datos/direcciones**, para disminuir el coste del bus.
- **Los buses que van a memoria** normalmente tienen líneas de direcciones separadas (cada palabra transferida a memoria debe ir acompañada de una dirección). Aquí no hay multiplexación.
- En cambio, **las transferencias E/S** ocurren normalmente en forma de grandes bloques de memoria, para lo que se requiere únicamente la dirección del comienzo del bloque. **Aquí si puede usarse la multiplexación.**

Triestado

- ¿Pueden estar accediendo al bus simultáneamente todos los dispositivos?
- ¿Pueden estar conectados al bus simultáneamente todos los dispositivos?

Si todos los dispositivos estuvieran conectados simultáneamente al bus puede ocurrir que dos dispositivos quisieran comunicar sus datos a la vez, y esto da lugar a un cortocircuito. En el mejor de los casos se perderían los datos de uno de ellos.

Para evitar esto, sólo se permite que un dispositivo esté conectado al bus en cada instante de tiempo. Esto requiere que tengamos que desconectar el resto de los dispositivos del bus durante ese tiempo.

Triestado

- Solución: puertas TRIESTADO entre los dispositivos y el bus.
- Las puertas Triestado tienen tres posibles salidas:

Entradas		Salida
X	E	Z
0	1	0
1	1	1
0	0	Z
1	0	Z

- Z es la salida de ALTA IMPEDANCIA

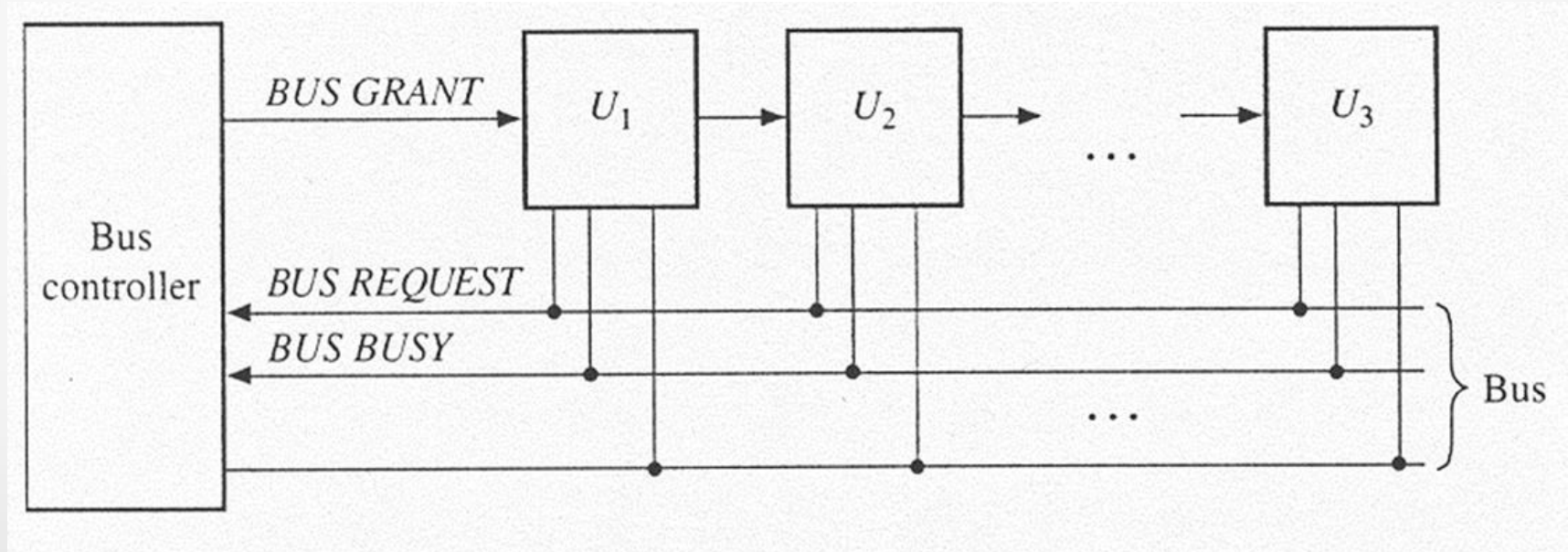
Triestado

- Las puertas Triestado permiten:
 - 1) Permitir arbitrar los accesos al bus
 - 2) Facilitar la transmisión de la señal bidireccional sobre una línea del bus permitiendo que la misma conexión al bus sirva tanto como puerto de entrada o salida.

Arbitraje del bus

1. Daisy Chaining (encadenamiento margarita):

- Hay tres líneas de control de bus: BUS REQUEST, BUS GRANT, BUS BUSY



Arbitraje del bus

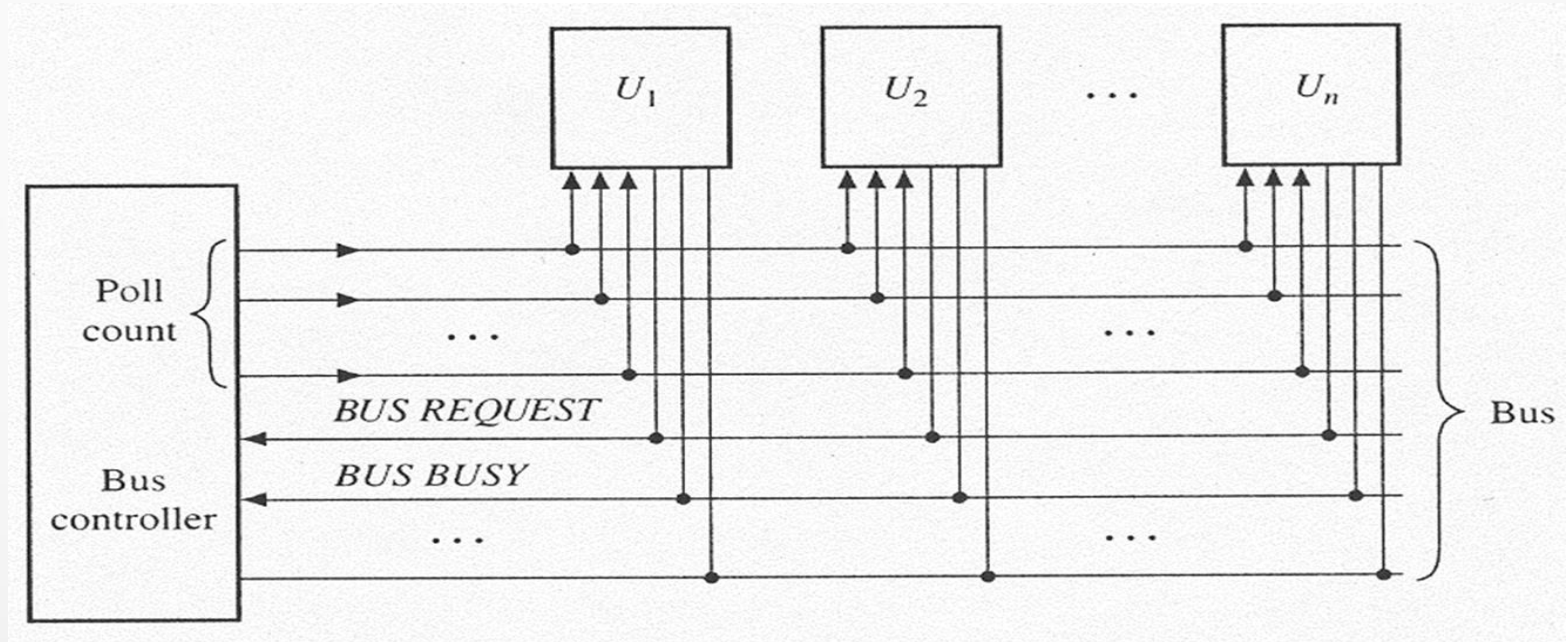
1. Daisy Chaining (encadenamiento margarita):

- Todas las líneas se conectan a BUS REQUEST. Si algún dispositivo pide acceso al bus, se activa esta línea.
- La unidad de control responde sólo si BUS BUSY está desactivada. Cuando responde activa BUS GRANT.
- Cuando el dispositivo recibe el BUS GRANT, activa el BUS BUSY.
- **La línea de BUS GRANT se conecta en serie:** si dos unidades solicitan el bus a la vez lo obtiene la más cercana. La prioridad se establece por el orden en que se conectan las unidades.
- **Inconveniente:** si falla una unidad el BUS GRANT no llegará al resto de las unidades.

Arbitraje del bus

2. Polling:

- Líneas de control de bus: BUS REQUEST, BUS BUSY, líneas de POLL COUNT.



Arbitraje del bus

2. Polling:

- BUS GRANT se reemplaza por una serie de líneas llamadas POLL COUNT que se conectan directamente a todas las unidades.
- Cada unidad tiene una dirección distinta (que es la que se indica en las líneas POLL COUNT).
- BUS REQUEST es una sola línea
- BUS BUSY es una sola línea.

Arbitraje del bus

2. Polling:

- **Consultar Entrada/Salida programada en el tema de Entrada/Salida.**
- El controlador de bus va testeando cada una de las unidades con una secuencia determinada (que puede ser modificada mediante programación). En definitiva les está preguntando si desean hacer uso del bus.
- Para ello escribe en las líneas de POLL COUNT la dirección de cada uno de los dispositivos en un orden prefijado.
- Cuando una unidad encuentra su dirección y desea hacer uso del bus, activa el BUS BUSY para que el controlador no siga testeando al resto de unidades.

Arbitraje del bus

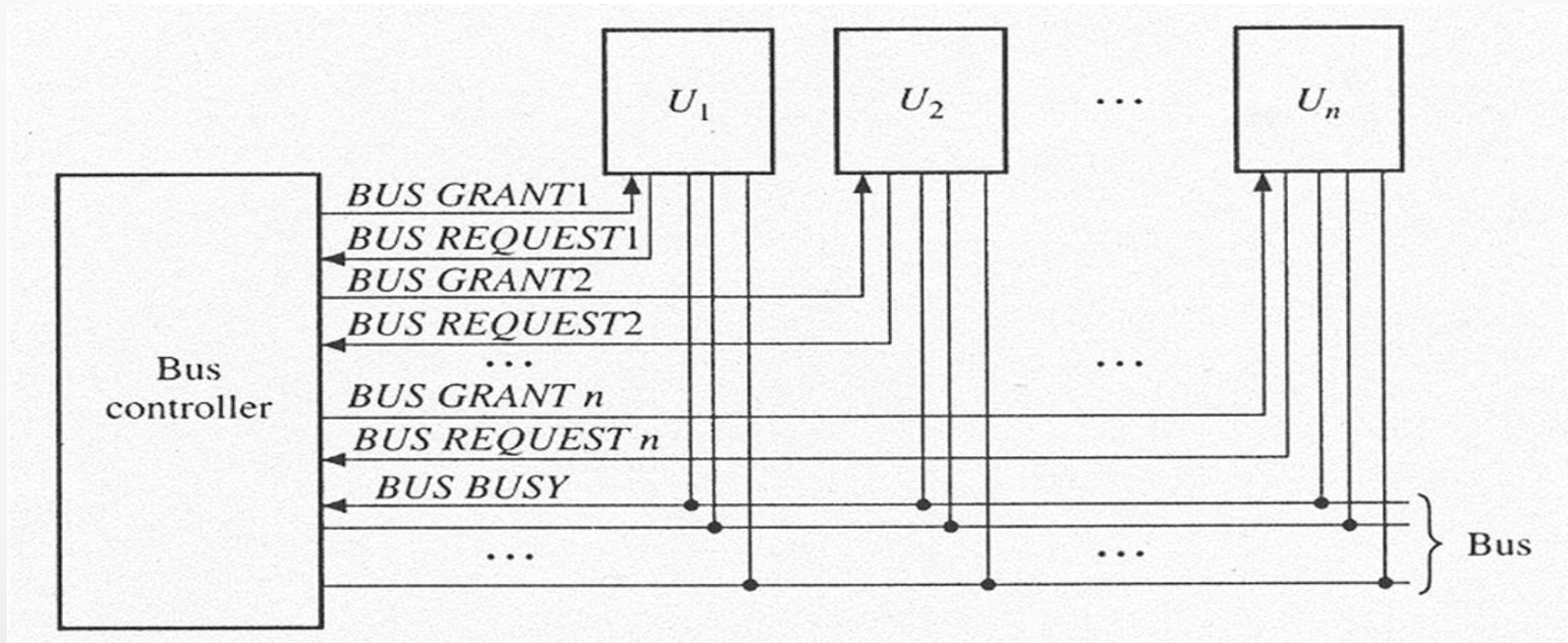
2. Polling:

- ***Consultar Entrada/Salida programada en el tema de Entrada/Salida.***
- Una vez activado el BUS BUSY puede transmitir sus datos.
- Si no se activa la línea de BUS BUSY el controlador pasa a testear la siguiente unidad.
- Ventaja: si una unidad falla, las demás no se ven afectadas. Esta flexibilidad se consigue a costa de aumentar el número de líneas de control.

Arbitraje del bus

3. Independent Request:

- Líneas de BUS GRANT y de BUS REQUEST separadas para cada unidad.



Arbitraje del bus

3. Independent Request:

- Existe una línea de BUS REQUEST por cada dispositivo
- Existe una línea de BUS GRANT por cada dispositivo
- Solo existe una línea BUS BUSY que comparten todos los dispositivos.

Arbitraje del bus

3. Independent Request:

- Cada dispositivo puede solicitar acceso al bus mediante su línea particular de BUS REQUEST, por lo que el controlador se libera de la tarea de ir preguntando a cada unidad.
- Si sólo un dispositivo ha activado la línea de BUS REQUEST, el controlador del bus activa la línea de BUS GRANT de ese dispositivo.
- Cuando el dispositivo ha recibido el acceso al bus activa la línea BUS BUSY para que ningún otro dispositivo pueda tener acceso al bus.

Arbitraje del bus

3. Independent Request:

- Si varios dispositivos solicitan acceso al bus el controlador tiene que decidir a quien se lo concede. Para ello tiene que tener priorizados a todos los dispositivos.
- La prioridad la decide el controlador del bus (es una prioridad programable).
- **Ventaja:** El controlador no pierde tiempo preguntando a cada unidad si necesita acceso al bus. Se gana en tiempo y en flexibilidad.
- **Inconveniente:** si tenemos n unidades conectadas al bus, necesitamos $2n$ líneas para BUS REQUEST Y BUS GRANT. Por tanto, este sistema es mas costoso económicamente que el POLLING.

Arbitraje del bus

3. Independent Request:

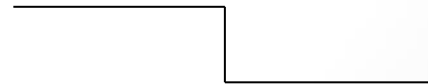
- El bus PCI utiliza esta técnica.
- *Consultar la Entrada/Salida por interrupciones en el tema de Entrada/Salida.*
- El sistema de arbitraje mediante independent request permite implementar las interrupciones. Por ejemplo:
 - Cuando pulsamos una tecla, el teclado envía una señal de bus request y cuando el controlador del bus le responde con un bus grant, procede a enviar la tecla que hemos pulsado.

Diagramas de tiempo

- Se usan para mostrar secuencias de eventos y dependencias entre eventos.
- La comunicación entre los dispositivos conectados al bus se produce a través de las líneas del bus. En cada línea se puede transmitir dos niveles diferentes de señal (niveles de tensión que representan el 0 y el 1). El diagrama de tiempo muestra el nivel de la señal de las líneas del bus en función del tiempo.



Flanco de subida



Flanco de bajada

Diagramas de tiempo

- El bus del sistema contiene una línea de reloj. Esta línea es muy importante porque los eventos se sincronizan con la señal de reloj.



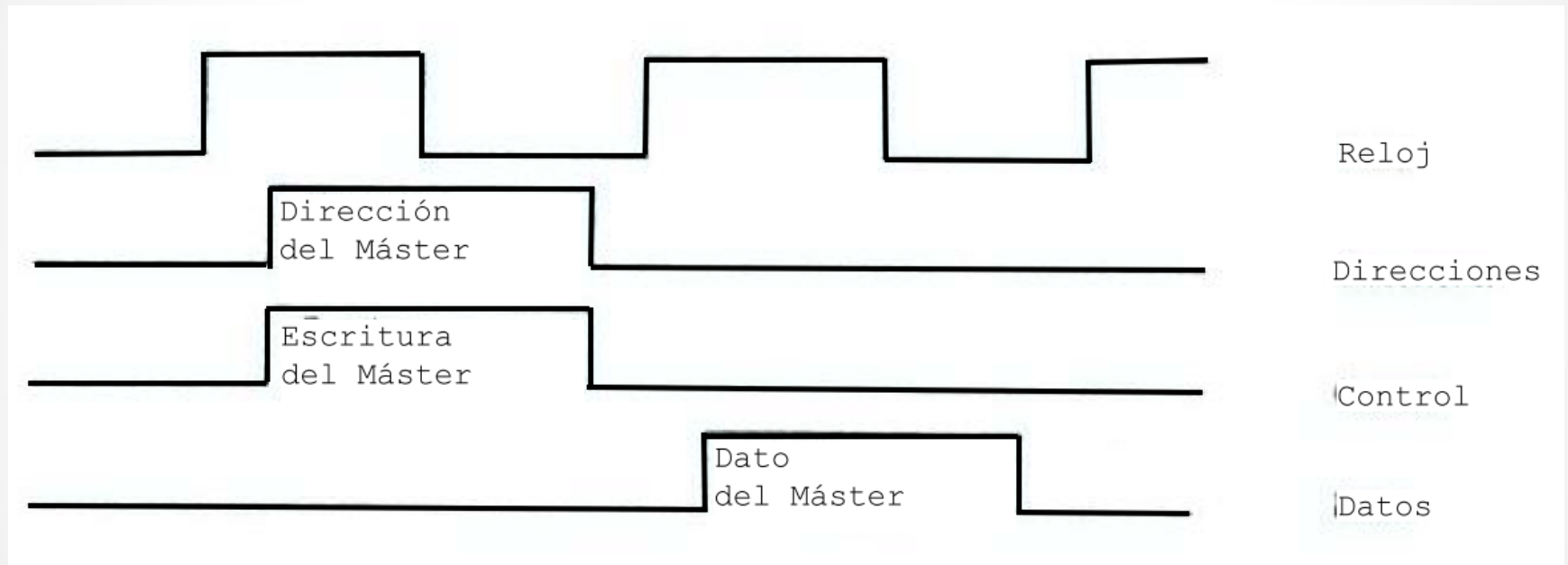
Ejemplos de temporización

1. Transferencia síncrona de lectura (CPU – memoria)



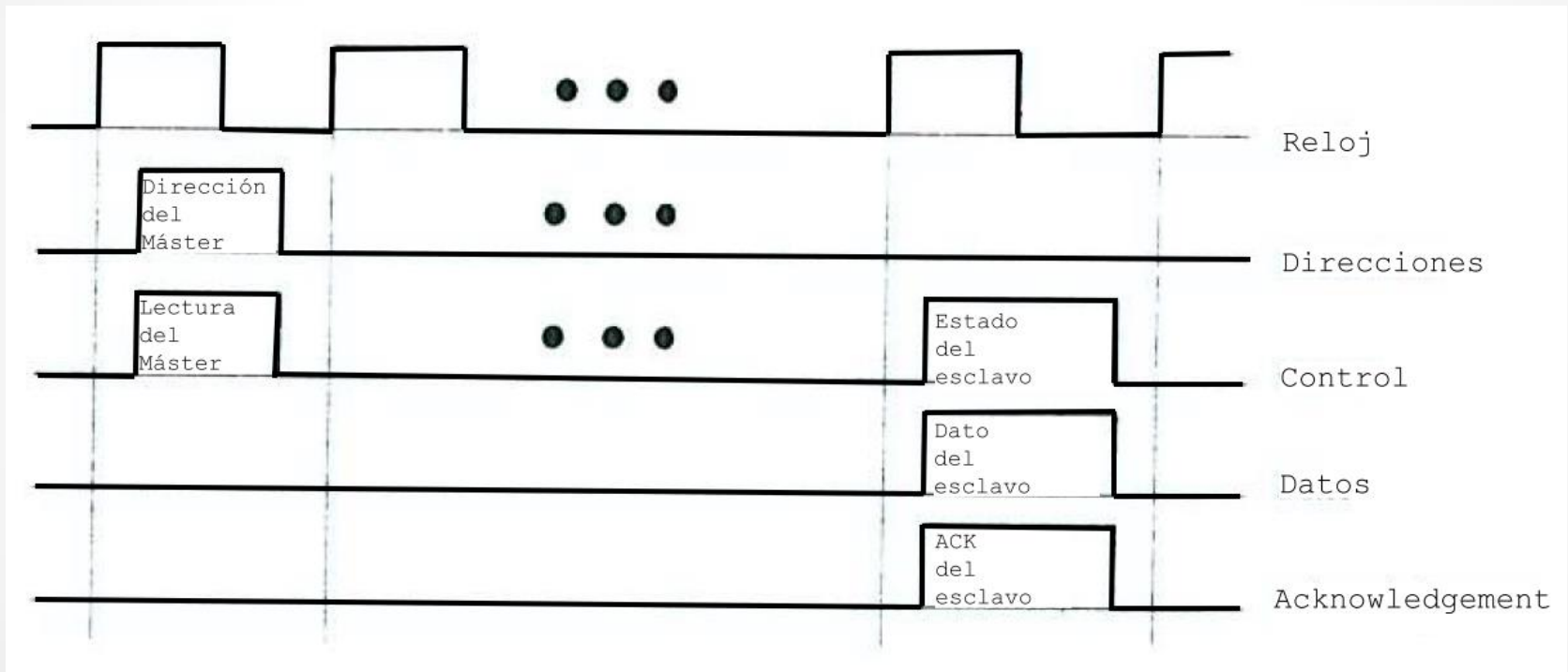
Ejemplos de temporización

2. Transferencia síncrona de escritura (CPU – memoria)



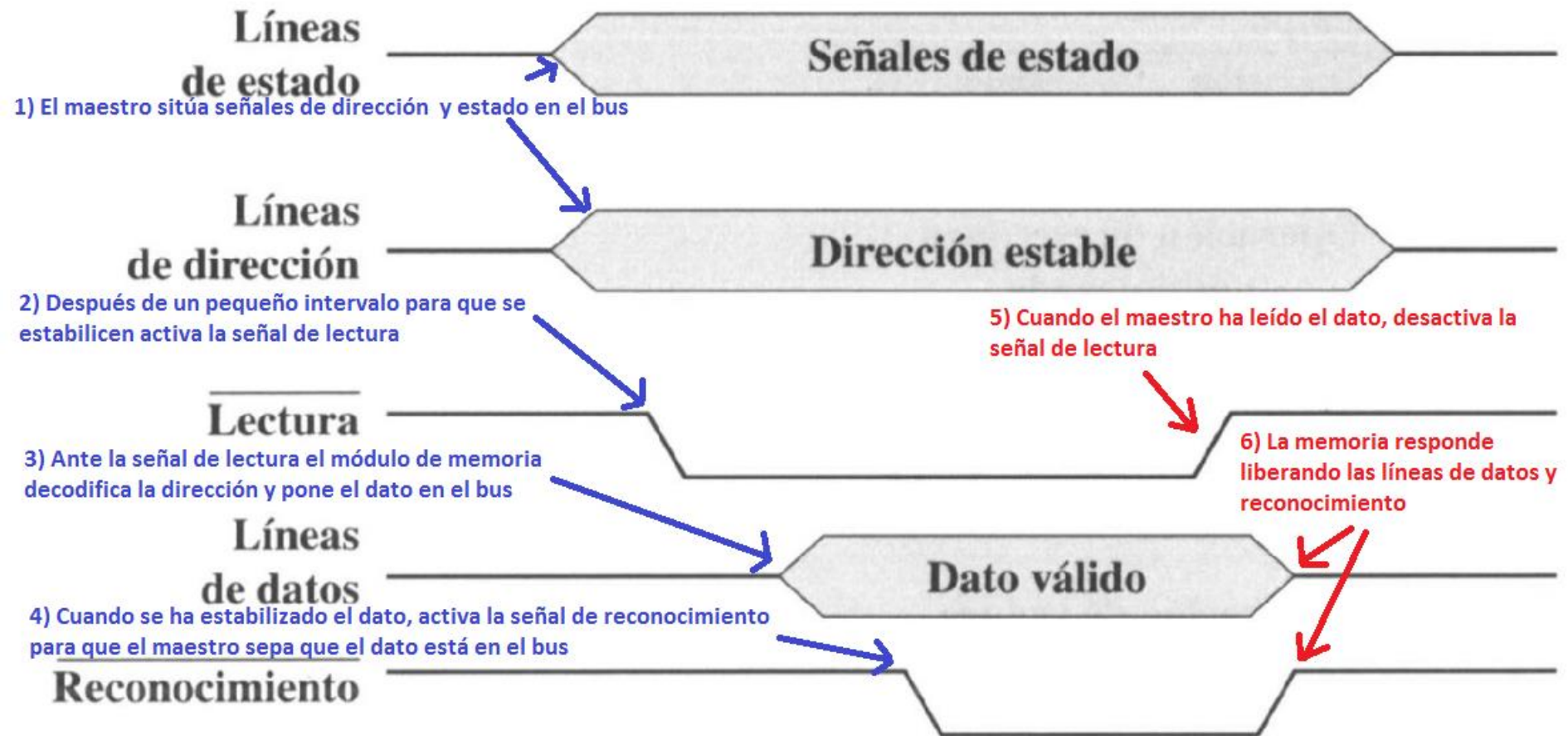
Ejemplos de temporización

3. Transferencia síncrona de datos con estado de espera (CPU – memoria)



Ejemplos de temporización

4. Transferencia asíncrona de datos - Lectura



Ejemplos de temporización

5. Transferencia asíncrona de datos - Escritura

