



Capítulo I

Estructuras de interconexión de un computador

Tema I: Estructuras de interconexión de un computador

- 1.1 Componentes de un computador
- 1.2 Función de un computador
 - 1.2.1 Ciclos de búsqueda y ejecución
 - 1.2.2 Ejemplo: Ejecución de una instrucción
 - 1.2.3 Diagrama de flujo del ciclo de instrucción
 - 1.2.4 El ciclo de interrupción
 - 1.2.5 Función de E/S
- 1.3 Estructuras de interconexión
- 1.4 Interconexión mediante bus
 - 1.4.1 Diagramas de temporización
 - 1.4.2 Estructura de bus
 - 1.4.3 Estructura jerárquica de buses
 - 1.4.4 Elementos de diseño del bus
 - 1.4.5 Consideraciones prácticas en la conexión mediante bus
 - 1.4.6 Ejemplo de estructura de bus: El Unibus

1.1 Componentes de un computador

Componentes de un computador:

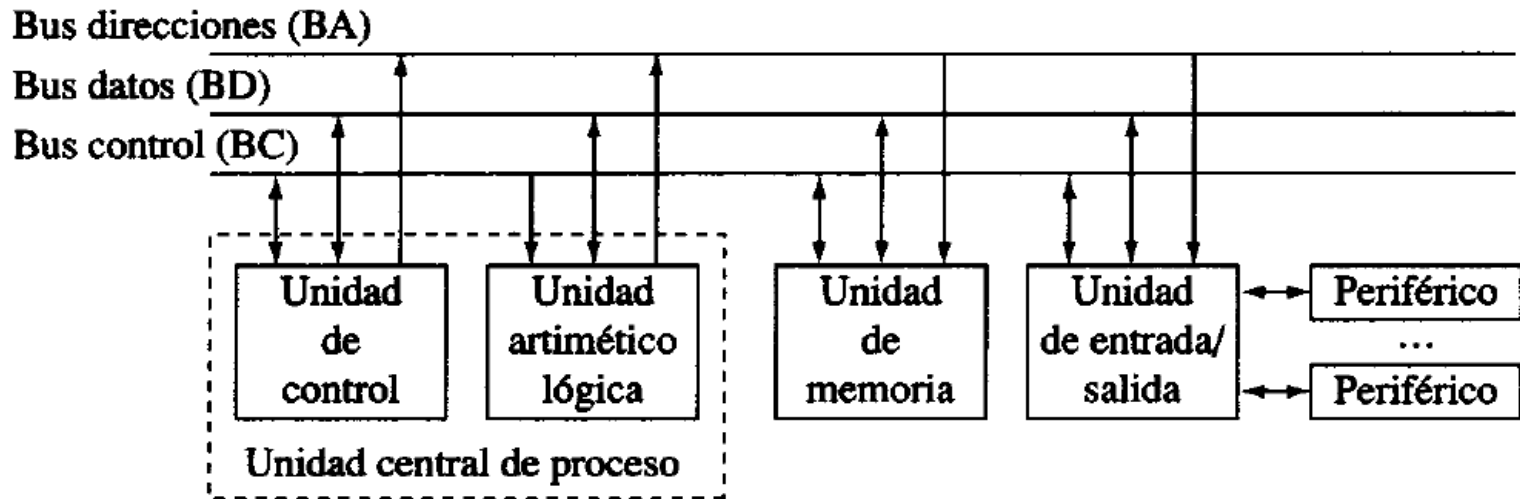


FIGURA 6.1. Estructura funcional de un computador.

Arquitectura Von Neumann

- Acceso a memoria aleatorio
- En memoria hay datos e instrucciones indistintamente
- Ejecución de programas es secuencial.
 - A no ser que se modifique la secuencia con una ruptura.

1.2 Función de un computador

- **Programa** = Conjunto de instrucciones y datos almacenados en memoria
- **Ciclo instrucción**
 - Fase de búsqueda
 - Fase de ejecución

Ejecución de una instrucción

- 1º. El PC apunta a la instrucción a ejecutar
- 2º. El código de la instrucción → R.I.
- 3º. Decodificación de la instrucción e incremento del PC
- 4º. Si se precisa buscar el operando nueva fase de búsqueda
- 5º. Si la instrucción está completa ejecutarla
- 6º. Ejecutar nueva instrucción

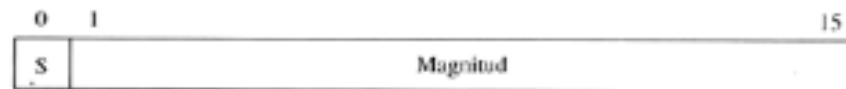
1.2.2 Ejemplo: Ejecución de una instrucción

Se considera una máquina hipotética que incluye las características especificadas en la Fig

a) Formato de instrucción:



b) Formato de los números enteros:



c) Registros internos de la CPU:

Contador de Programa (PC) = Dirección de la próxima instrucción

Registro de Instrucción (IR) = Instrucción que se está ejecutando

Acumulador (AC) = Almacenamiento temporal

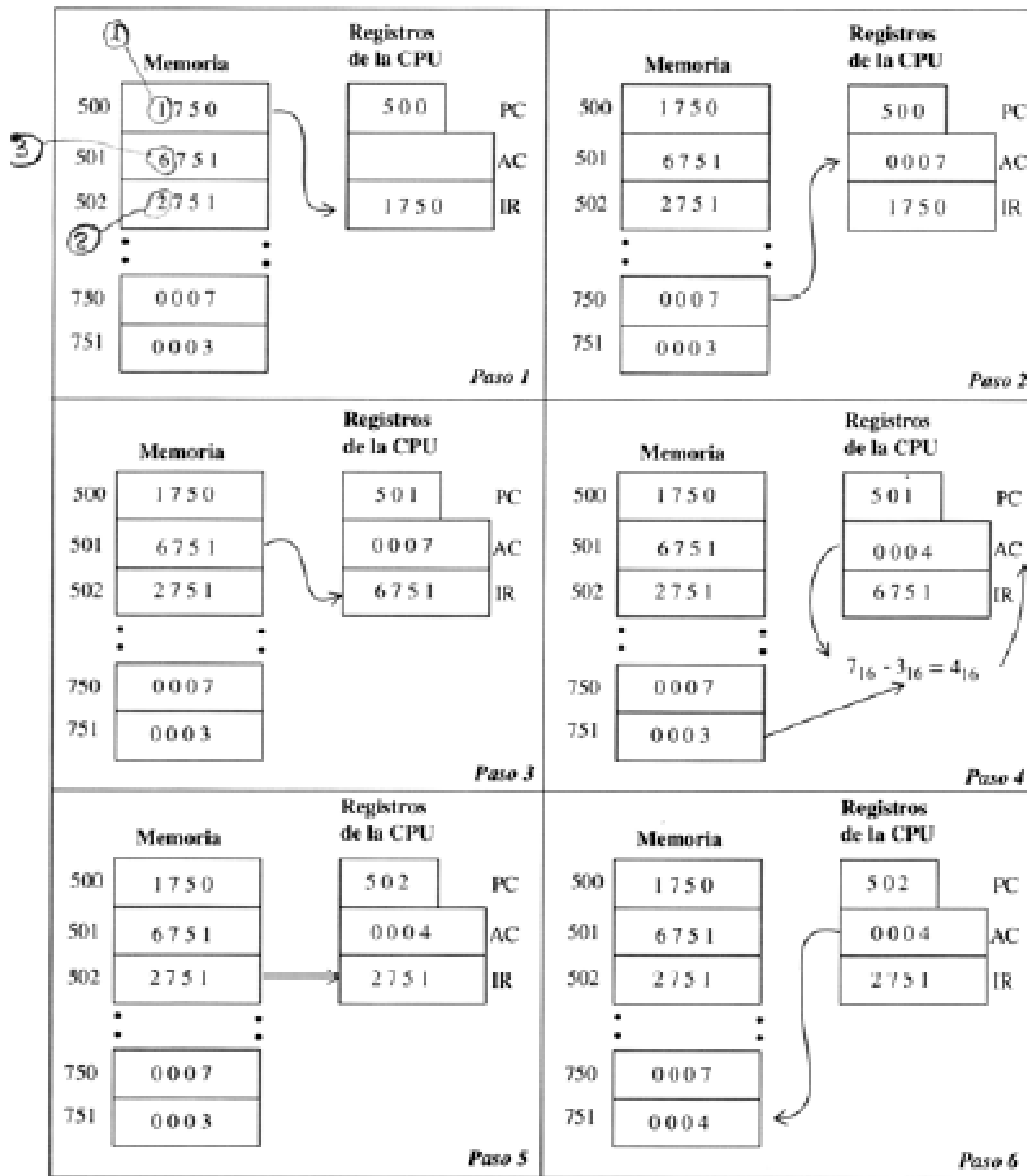
d) Lista parcial de códigos de operación:

- ① → 0001 = Cargar el AC desde memoria
- ② → 0010 = Almacenar el AC en memoria
- ③ → 0110 = Restar al AC de memoria

Figura 1.5: Características de un computador hipotético

La CPU contiene un acumulador (AC) que permite almacenar los datos de forma temporal. Las instrucciones como los datos tienen una longitud de 16 bits. El formato de la instrucción tiene haber $2^4 = 16$ códigos de operación diferentes y hasta $2^{12} = 4.096$ (4K) palabras de memoria con direccionamiento directo, ya que en el formato de instrucción se han reservado 4 bits para la operación y 12 bits para la dirección.

La Figura 1.6 ilustra la ejecución parcial de un programa, mostrando las partes relevante



1
1
2
3
4

Pasos de la ejecución del programa

- 1) El contenido del contador de programa (PC) es 500_{16} , que es la dirección de la primera instrucción. El contenido de esta dirección se carga en el registro de instrucción (IR). Debe observarse que este proceso requiere la utilización del registro de dirección de memoria (MAR) y un registro de datos de memoria (MBR). Para hacer el ejemplo de ejecución del programa lo más sencillo posible se ignoran, por el momento, los contenidos de estos registros internos a la CPU.
- 2) Los 4 primeros bits de IR (0001) indican que hay que cargar el AC. Los 12 bits restantes especifican la dirección, que es 750_{16} .
- 3) Se incrementa el PC y se lee la instrucción siguiente.
- 4) Al antiguo contenido del AC, se le resta el contenido de la posición 751_{16} y el resultado se almacena en el acumulador (código de operación = 0110).
- 5) Se incrementa el PC y se lee la instrucción siguiente.
- 6) Se almacena el contenido del AC en la posición 751_{16} (código de operación = 0010).

En este ejemplo, para restar al contenido de la posición 750_{16} el contenido de la posición 751_{16} se necesitan tres ciclos de instrucción, cada uno consistente en un ciclo de búsqueda y un ciclo de ejecución. Con un conjunto de instrucciones más complejas se habrían necesitado menos ciclos. ♦

Ciclo de interrupción

- Todos los computadores disponen de un mecanismo para que otros módulos puedan interrumpir a la UCP.
 - El objetivo fundamental de las interrupciones es mejorar el rendimiento de los computadores
- Ciclo con interrupción
 - Ciclo de búsqueda
 - Ciclo de ejecución
 - Ciclo de Interrupción
 - Comprobar y procesar la interrupción
 - Guarda el contexto del programa que está ejecutando.
 - Carga en el contador de programa la dirección de comienzo de un programa encargado de gestionar la interrupción.

1.3 Estructuras de interconexión

- El conjunto de caminos que conectan los diferentes módulos constituye la estructura de interconexión.
- E/S a través de la UCP:
 - La UCP y los módulos de E/S comparten la misma vía de acceso a memoria, por lo que la UCP debe parar sus cálculos al tener que controlar todos los intercambios.

Estructuras de interconexión (II)

- E/S a través de memoria:
 - Es posible el acceso directo a memoria principal de dos o más componentes de forma independiente.
- E/S mediante conmutador central:
 - Existe un mecanismo de distribución centralizado al cual se unen todos los componentes
- Bus de E/S:
 - Un conjunto de líneas que se comparten por todos los módulos

1.4 Interconexión mediante bus

■ Función del bus

- Soportar la información
- Garantizar comunicación

■ Estructura de bus

- Bus de Direcciones → anchura → capacidad direccionamiento
- Bus de Datos → anchura → rendimiento
- Bus de Control → específico en cada μP

Estructura Jerárquica de buses

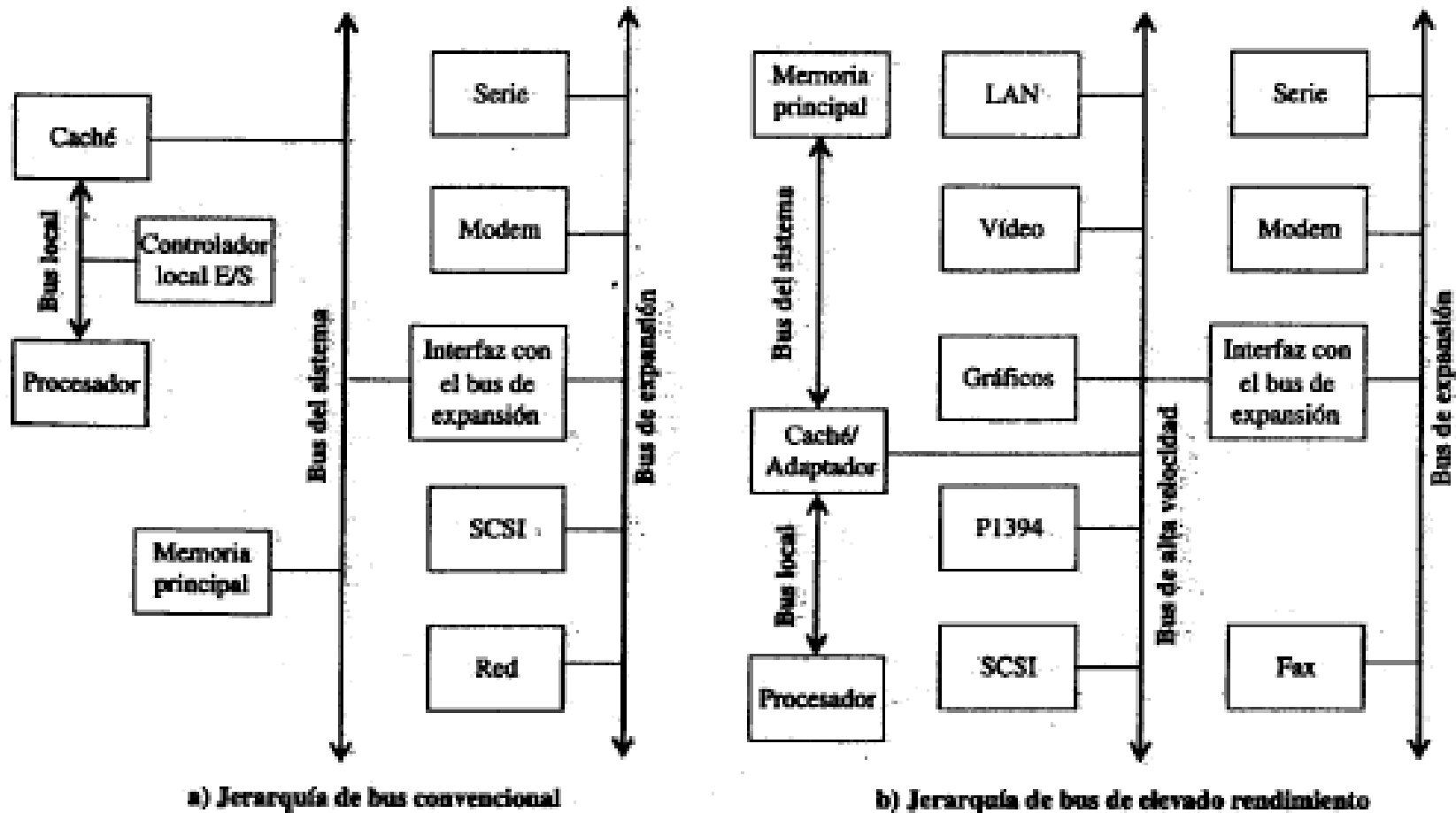


Figura 1.22: Ejemplos de jerarquías de buses múltiples

Elementos de diseño del bus

Tipo	Método de arbitraje	Temporización	Anchura del bus	Transferencia de datos
Dedicado	Centralizado	Síncrona	Dirección	Transferencia de escritura
No dedicado	Distribuido	Asíncrona	Datos	Transferencia de lectura

Tabla 1.1: Elementos de diseño del bus

Método de arbitraje Centralizado

- Existe un dispositivo controlador de bus encargado de controlar el bus. Estructura daisy-chaining

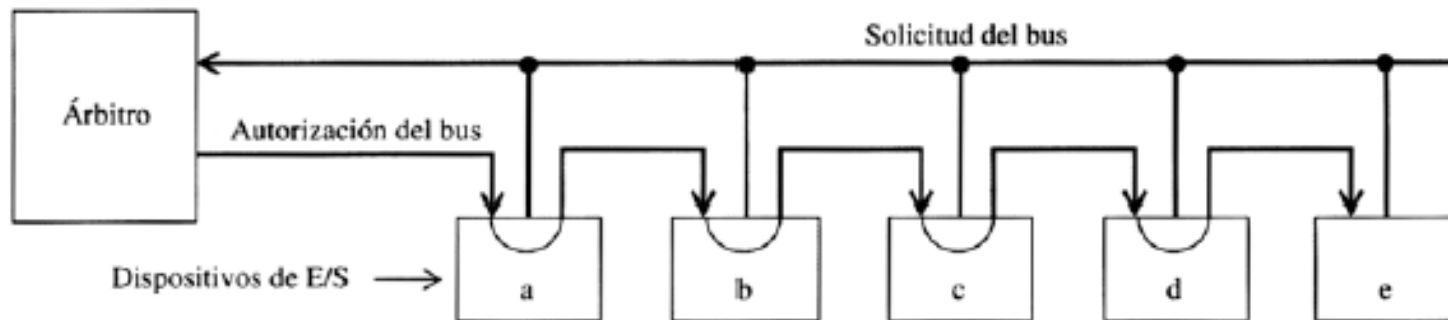


Figura 1.23: Bus con arbitraje centralizado por encadenamiento

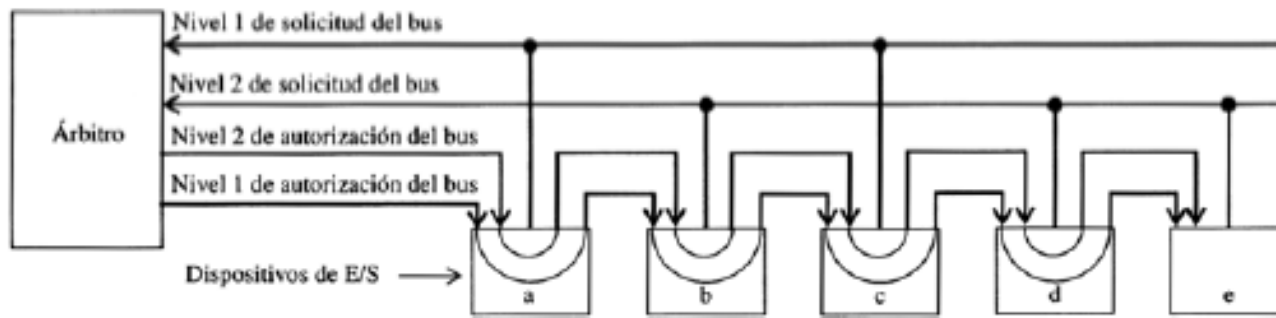


Figura 1.24: Bus con arbitraje centralizado y dos niveles de prioridad

Modo de arbitraje distribuido

- No hay controlador de bus y cada módulo tiene una lógica de control suficiente para poder acceder al bus

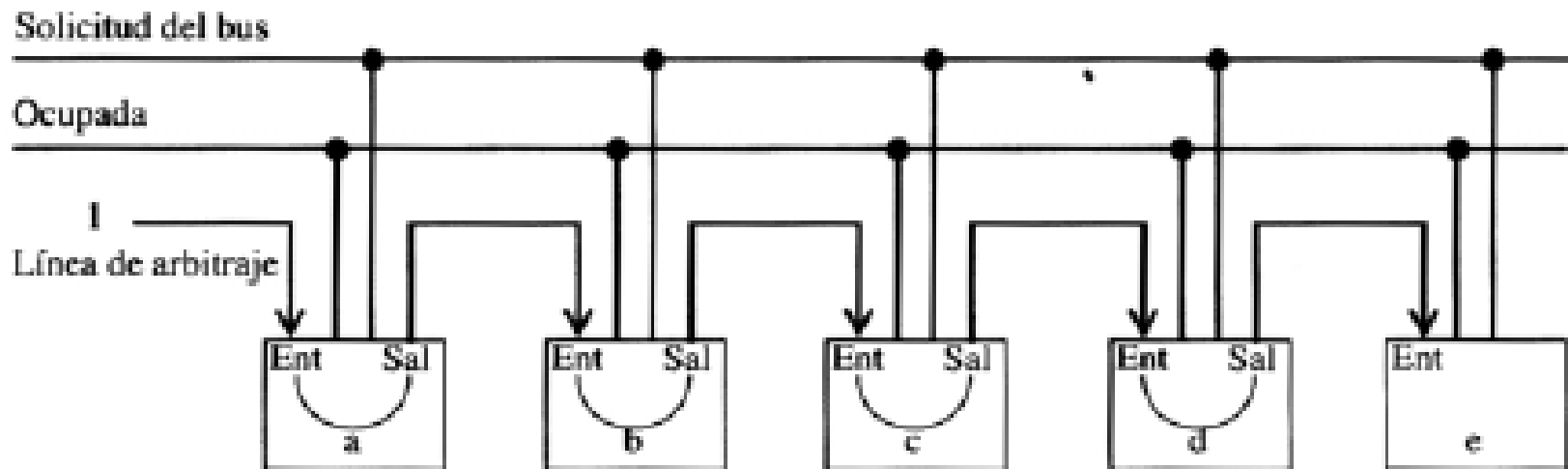


Figura 1.25: Bus con arbitraje distribuido

Sincronización síncrona

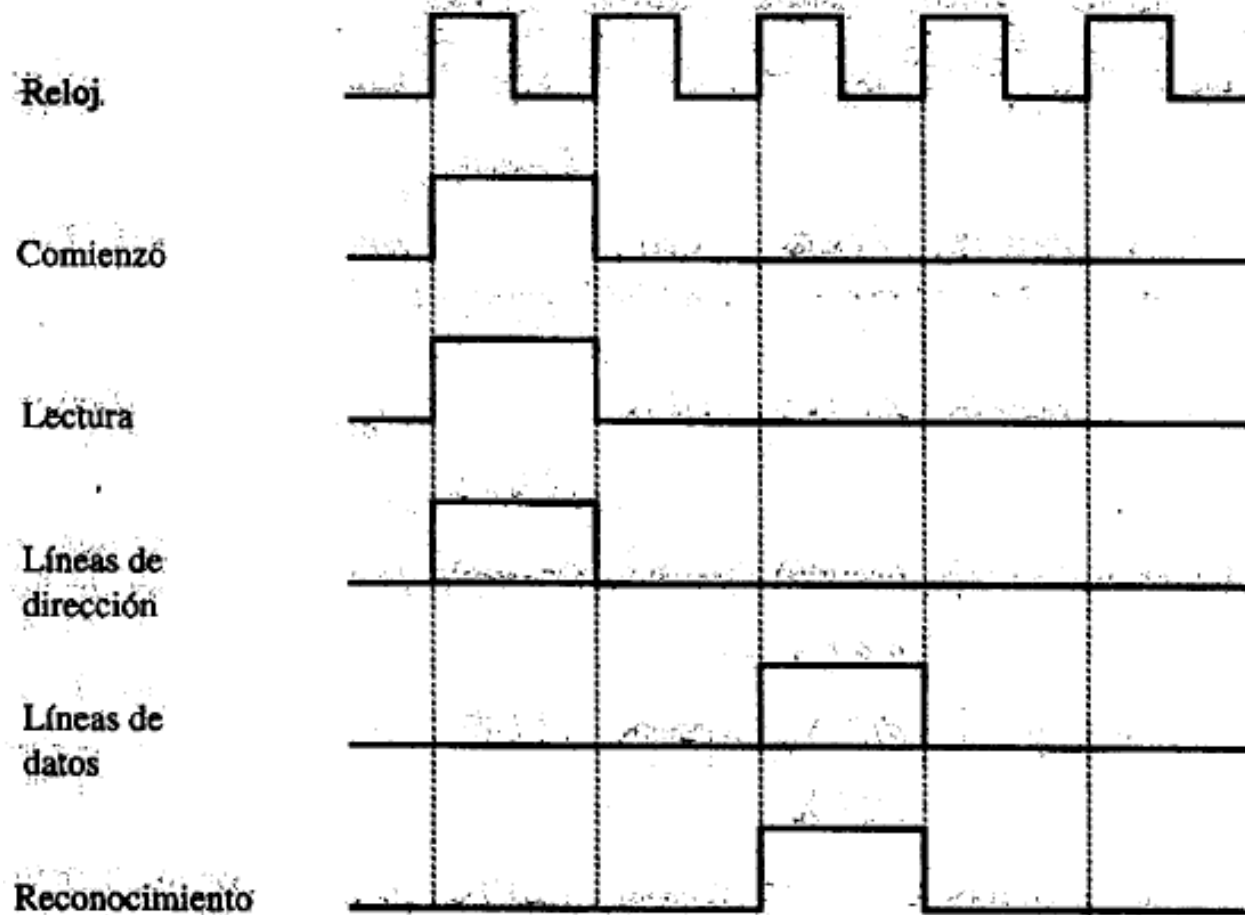


Figura 1.26: Temporización síncrona

Sincronización asíncrona

- 1) Se activa *MSIN*
- 2) Se activa *ESIN* en respuesta a *MSIN*
- 3) Se desactiva *MSIN* en respuesta a *ESIN*
- 4) Se desactiva *ESIN* en respuesta a la desactivación de *MSIN*

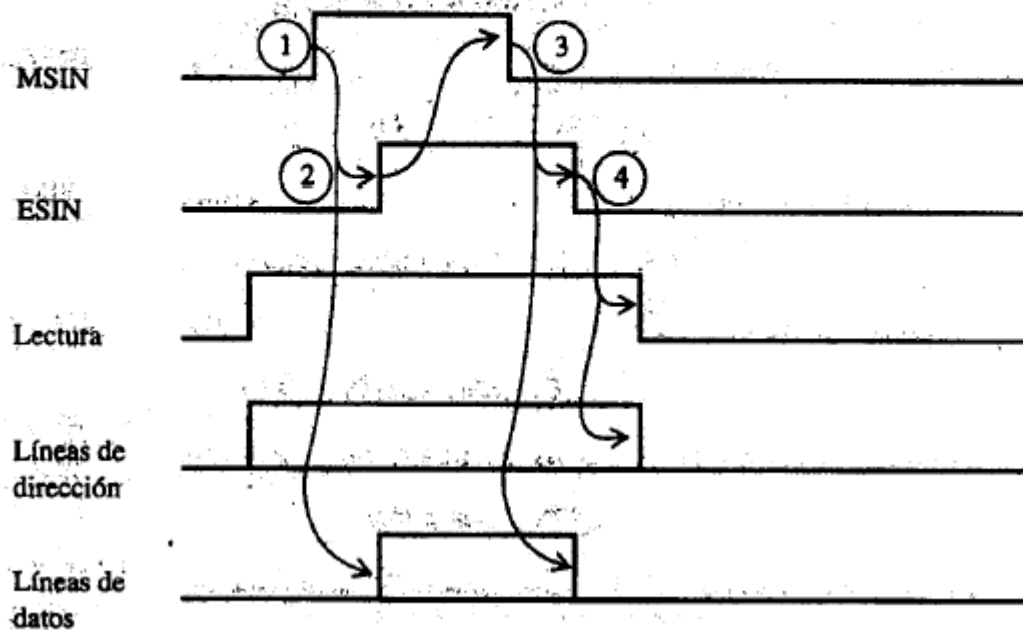


Figura 1.27: Temporización asíncrona

Puertas triestados (I)

- La utilización del bus compartido implica el empleo de circuitos de acoplo con el fin de garantizar el funcionamiento correcto del bus. Para evitar problemas de Fan Out

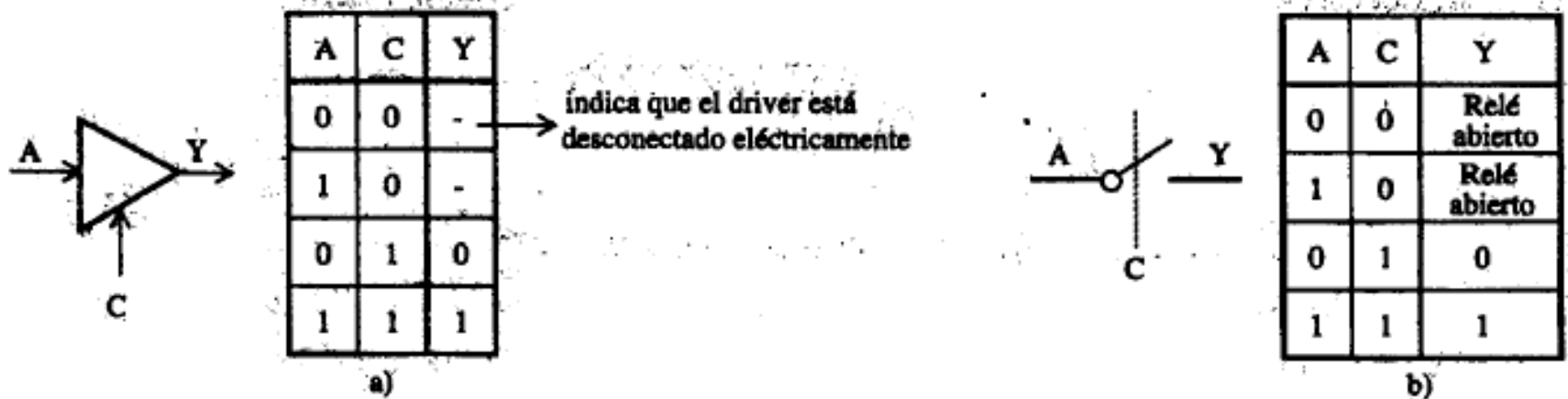


Figura 1.28: Funcionamiento de un driver triestado

Puertas triestado (II)

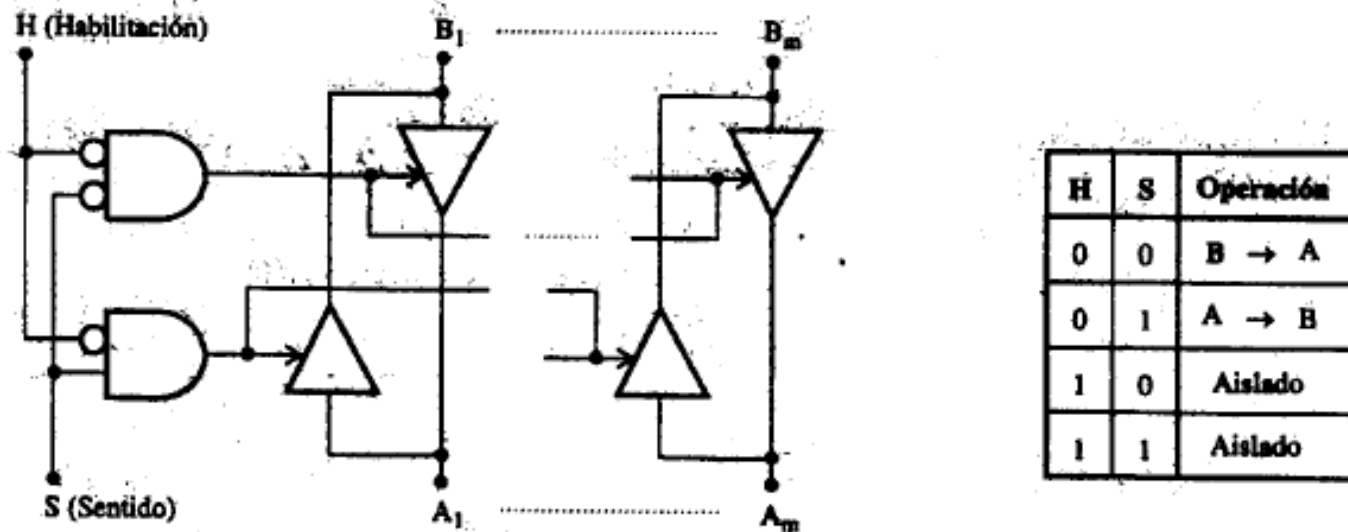


Figura 1.36: Circuito de conexión bidireccional con señal de control común

En la Figura 1.37 se representa de forma compacta el circuito de la Figura 1.36 con señales de control comunes: *S* (sentido de la comunicación) y *H* (habilitación del circuito).