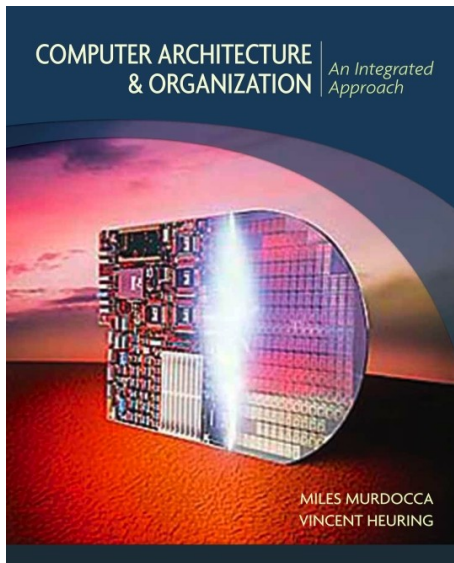


Organización de las Computadoras

Leonardo Giovanini



Gestión de periféricos

Consulta y Acceso Directo a Memoria

Contenidos

8.1 Buses del sistema

8.1.1 *Arquitectura del sistema*

8.2 Gestión de periféricos

8.3 Gestión programada

8.3.1 *Registros de estado*

8.4 Gestión por acceso directo

8.4.1 *Secuencia de operación*

8.4.2 *Controlador de DMA*

8.4.3 *DMA con cache*

Buses del sistema

Buses del sistema

Los sistemas informáticos generalmente constan de tres partes:

- i) la unidad central de procesamiento que procesa los datos;
- ii) la memoria que contiene los programas y los datos a procesar y
- iii) los dispositivos de entrada/salida que se comunican con el mundo exterior.

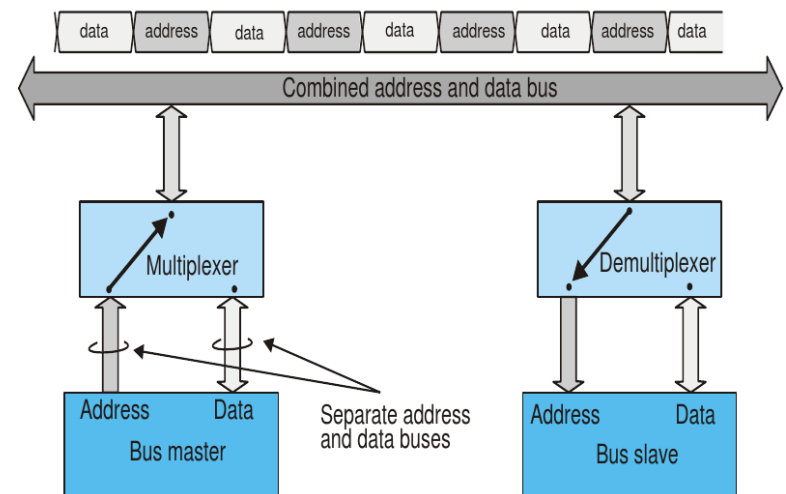


En arquitectura de computadoras, un **bus** (una contracción del Latín *omnibus*) es **un sistema de comunicación** que transfiere datos entre componentes dentro de una computadora o entre computadoras.

Los primeros buses de computadora eran un conjunto de cables paralelos con múltiples conexiones de hardware, pero el término ahora se usa para cualquier disposición física que proporcione la misma función lógica que un bus eléctrico paralelo.

Elementos que sirven clasificar y diferenciar buses:

- **Tipos de bus** - definido por el tipo de información que transporta;
- **Bus dedicado** - se asigna permanentemente a una función a un subconjunto de componentes de la computadora; y
- **Bus multiplexado** - la información se transmite a través de un único conjunto de líneas utilizando una línea de control.



Buses del sistema

La comunicación y coordinación entre los diferentes componentes es manejada un conjunto de líneas que transportan las señales conocidas como **buses del sistema**.

Los buses conectan los componentes de la computadora, los cuales deben operar a la velocidad del dispositivo más rápido conectado.

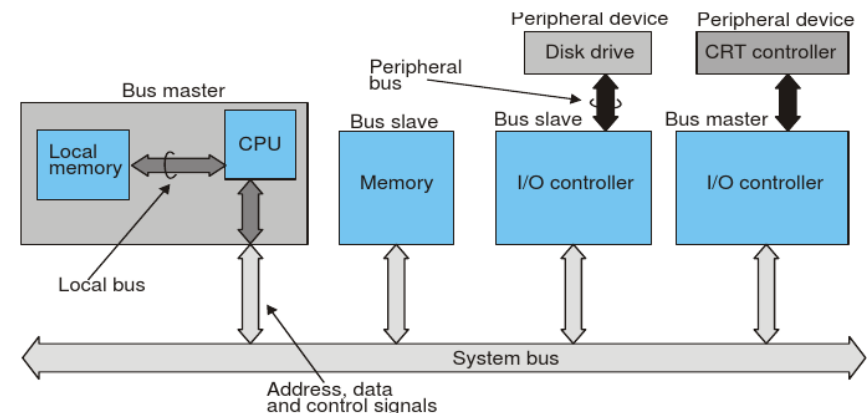
El bus del sistema transportan información de datos, direcciones y control, además pueden incluirse las líneas de distribución de energía.

Dependiendo de la información transportada los buses se clasifican en:

Bus de datos - proporcionan una ruta para mover datos entre los módulos del sistema. Es bidireccional y su ancho (número de líneas) determina el rendimiento general del sistema, en combinación con su ancho de banda y latencia;

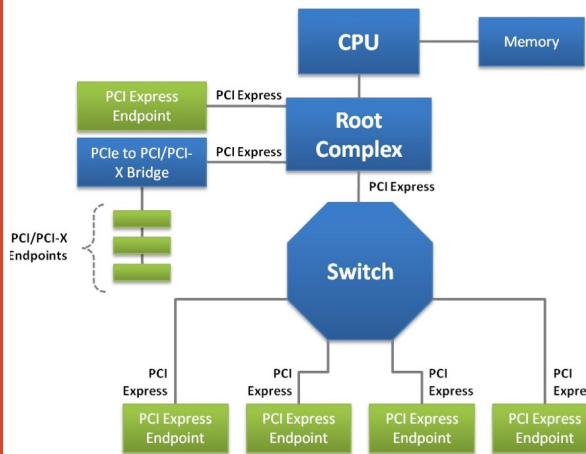
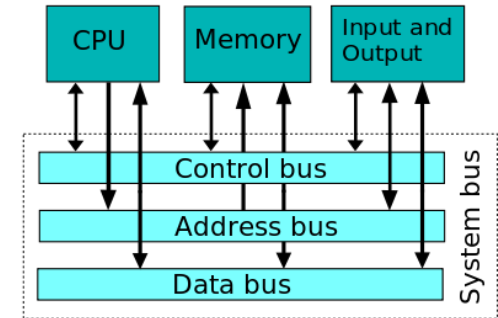
Bus de direcciones - designa el origen/destino de los datos en el bus de datos. Es unidireccional y su ancho determina la capacidad de memoria máxima posible del sistema. También se utiliza para direccionar los puertos de E / S. En este caso los bits superiores seleccionan un módulo en el bus, y los bits inferiores seleccionan la ubicación de la memoria o el puerto; y

Bus de control - controla el acceso/uso de los datos y direcciones, y tiene un bidireccional. Transmite información de comando y sincronización entre los módulos del sistema. Las señales de comando especifican las operaciones, mientras que las de sincronización indican la validez de los datos y la información de la dirección.



Buses del sistema

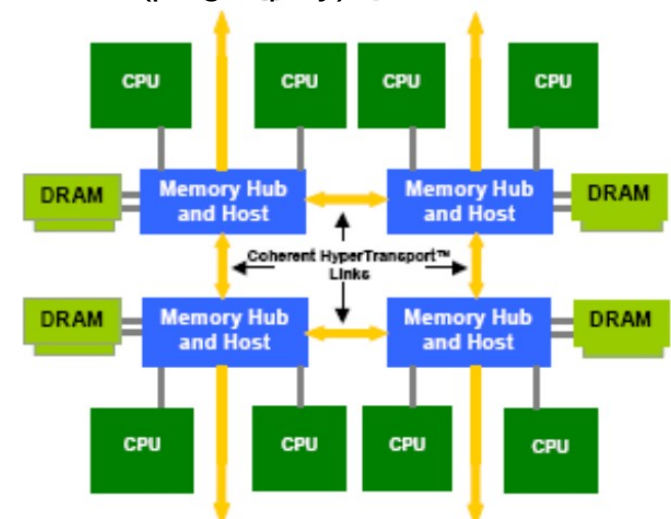
Primera generacion: Las primeras computadoras tenian buses separados para la memoria y los periféricos. Se accedía a los periféricos a través instrucciones separadas, con tiempos y protocolos diferentes. La CPU se usaba para controlar el sistema, generando problema para la ampliación y actualización del sistema. Además que la CPU utilizaba una parte considerable de su potencia en controlar el bus. Se modularizaron los buses combinandolos en un sistema unificado y estandarizado.



Segunda generacion: La arquitectura de la computadora es separada en grupos de acuerdo a su velocidad de operación e incluye un controlador de bus. Este dispositivo gestiona el flujo de datos desde la CPU hacia los periféricos, reduciendo la carga de trabajo de la CPU. La comunicación entre los dispositivos se gestiona automáticamente por el controlador, mejorando el desempeño global del sistema. Los problemas de velocidad se resolvieron aumentando el tamaño del bus y agregando configuración por software (plug & play).

Tercera generacion: La arquitectura de la computadora es reformulada como una red de alta velocidad con topología variable. Esta configuración es muy flexible en cuanto a sus conexiones físicas, lo que permite utilizarla para conectar buses internos como máquinas entre sí. Esto produce problemas complejos al tratar de atender solicitudes con requerimientos tan diferentes, ya que gran parte del trabajo en estos sistemas refiere al diseño del software en lugar del hardware.

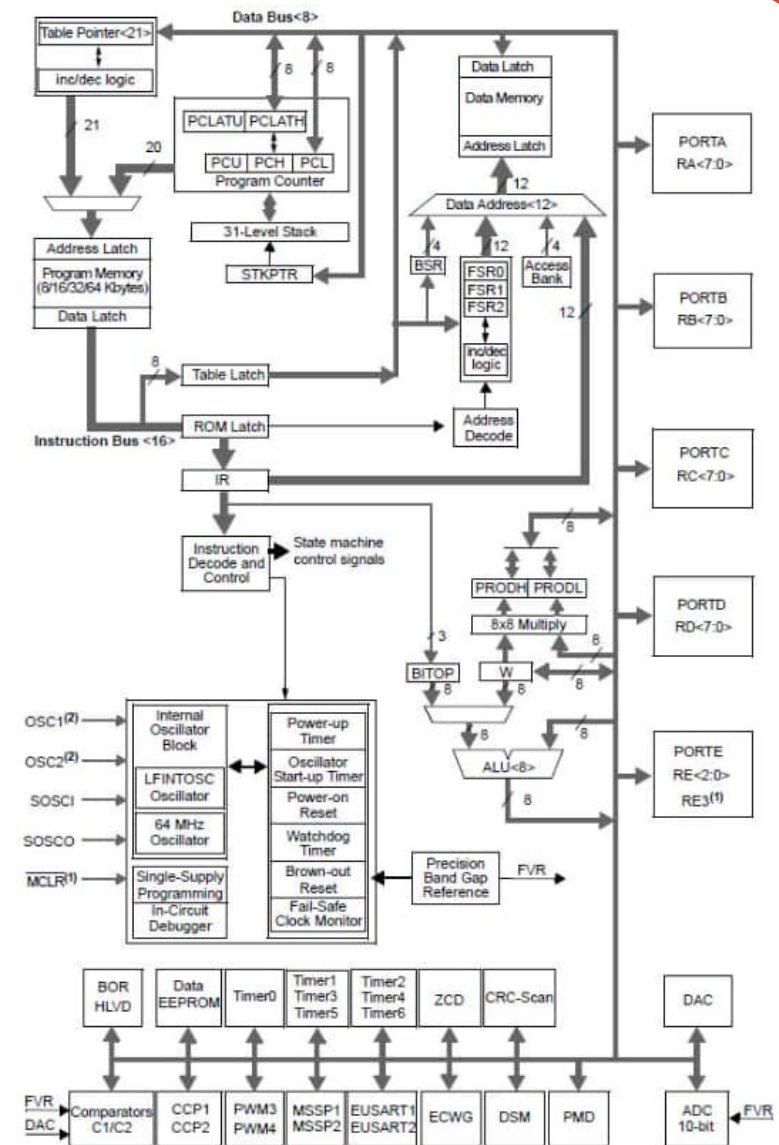
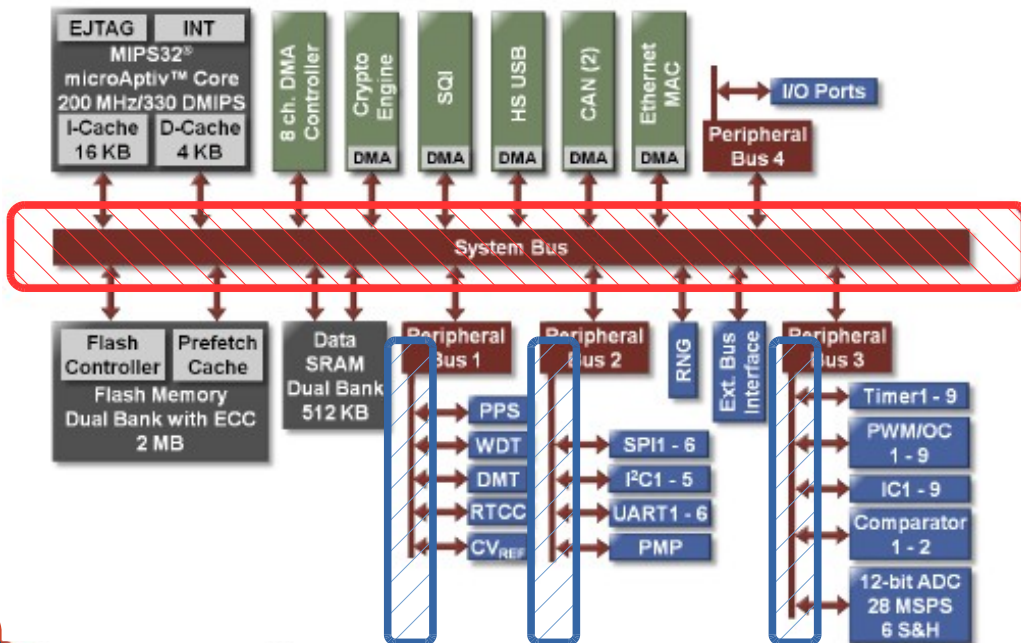
Ejemplos de buses de esta generación son HyperTransport, InfiniBand y Wishbone.



Buses del sistema

Arquitectura del sistema

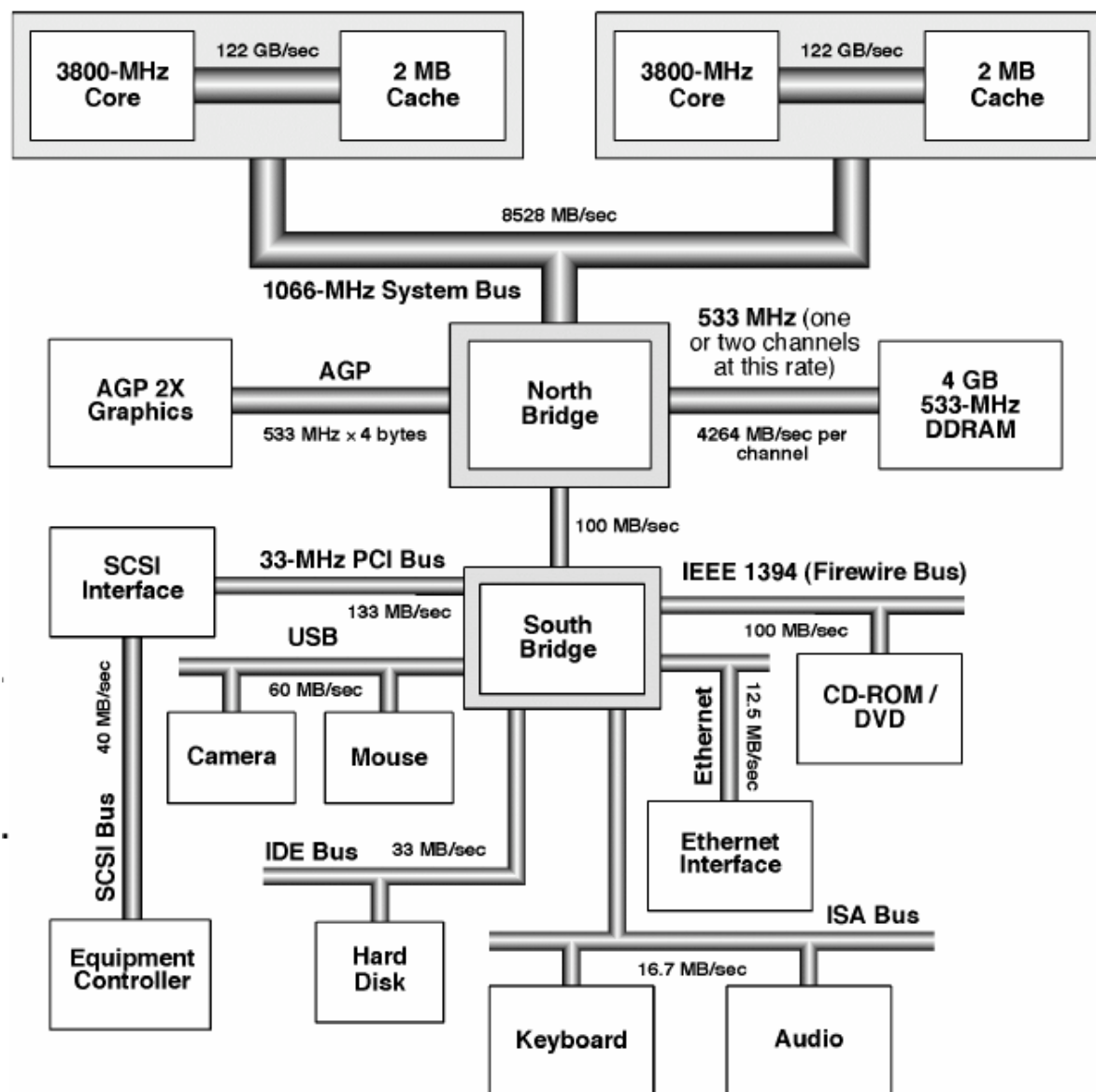
Las computadoras pueden implementarse utilizando multiples buses.



Las computadoras pueden implementarse utilizando un bus único

Buses del sistema

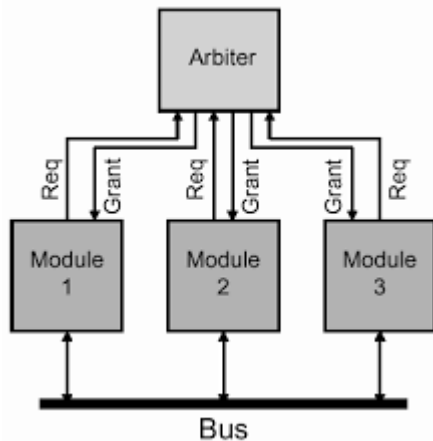
Arquitectura del sistema



Gestión de periféricos

El **arbitraje de bus** refiere a la gestión de los buses del sistema durante su operación, coordinando las actividades de los dispositivos que solicitan transferencias de datos.

El controlador que tiene acceso a un bus en una instancia se conoce como bus master y su selección debe tener en cuenta las necesidades del sistema.



Hay dos enfoques para el arbitraje de buses:

Centralizado - un único árbitro gestiona el bus; y

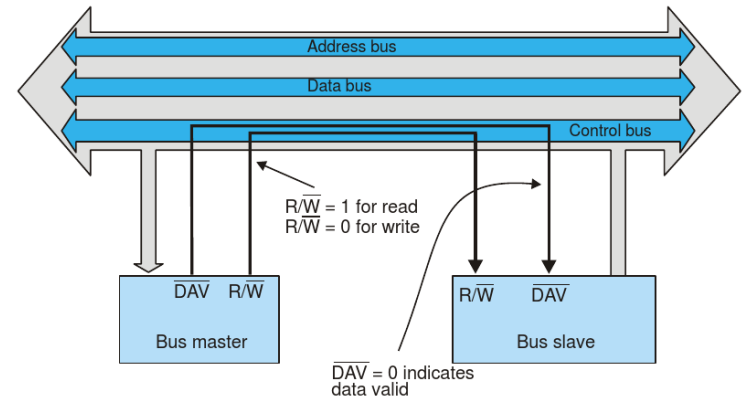
Distribuido - todos los dispositivos pueden gestionar el bus.

Hay tres tipos de transferencias que pueden realizarse:

- Memoria a periférico;
- Periférico a memoria;
- Entre bloques de memoria.

Hay tres mecanismos para gestionar las transferencias de datos:

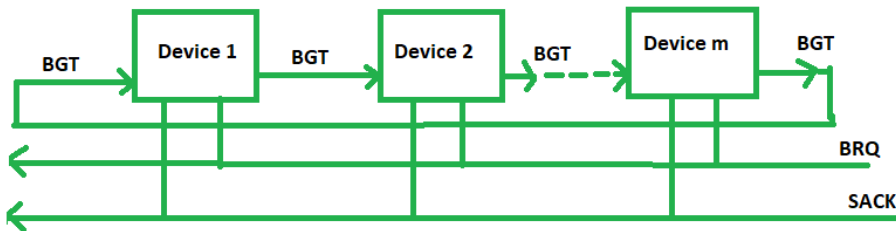
- Gestión programada;
- Gestión por interrupciones; y
- Gestión por acceso directo.



- **Encadenamiento** (daisy-chain) - es un método de arbitraje de bus centralizado. Durante cualquier ciclo el maestro puede ser cualquier dispositivo conectado al bus.

Ventajas - simplicidad y escalabilidad;

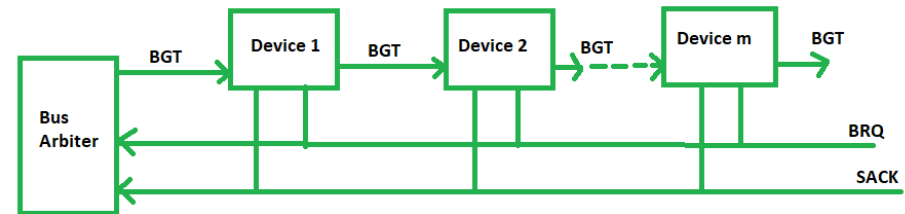
Desventajas - la prioridad depende de la posición; si un dispositivo falla todo el sistema falla.



- **Fija** (fix priority) - el control de bus pasa de un dispositivo a otro a través del árbitro centralizado.

Ventajas – velocidad de respuesta;

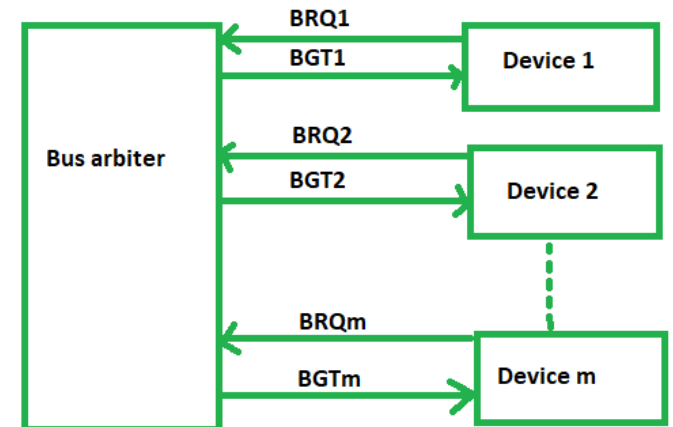
Desventajas - costo elevado; hardware complejo.



- **Rotación** (rotating priority) - la prioridad de los dispositivos se asigna cuando se accede al bus, pero puede cambiarse dinámicamente.

Ventajas - simplicidad; si un dispositivo falla todo el sistema deja de funcionar;

Desventajas – gestión compleja a medida que aumenta el tamaño del sistema.



Gestión programada

Gestión de periféricos

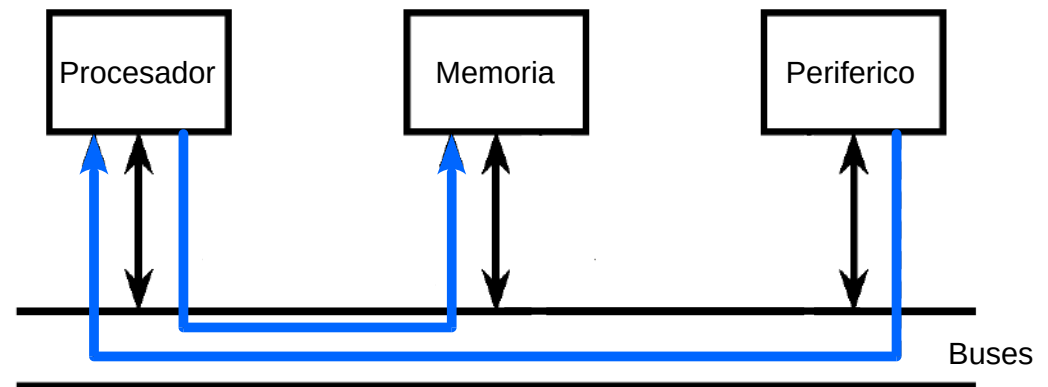
Gestión programada

Es un mecanismo por el cual las tareas necesarias para atender los periféricos se organiza en el programa. La rutina atiende cada uno de los periféricos hasta completar la lista de tareas.

El procesador tiene control directo sobre la operación de transferencia: i) Comprobación del estado del dispositivo, ii) envío de comandos de lectura/escritura, iii) transferencia de datos, y iv) la CPU espera que el dispositivo termine la transferencia.

Como el orden de atención es controlado por el programa, de modo que la prioridad de cada tarea puede ser asignada dinámicamente de acuerdo al estado general del sistema.

Esta asignación se realiza a través del cambio de orden de consulta.



Ventajas: se tiene control total sobre el proceso, tiene una alta repetibilidad en la ejecución, es robusto a los eventos externos sincrónicos.

Inconvenientes: tiene altos tiempos de latencia, es ineficiente en el uso del tiempo, no puede atender eventos asincrónicos y la carga de programación es elevada.

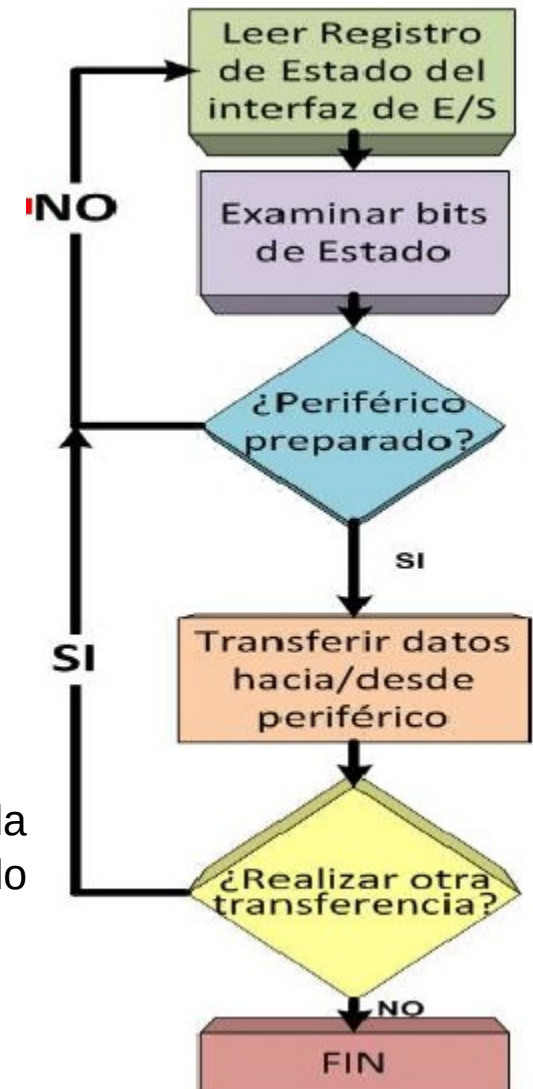
Gestión de periféricos

Gestión programada

Los pasos ejecutados durante una gestión programada son

1. El procesador direcciona un dispositivo y solicita una operación;
2. El dispositivo direccionado realiza la operación;
3. El dispositivo activa los bits de estado y espera a una nueva operación;
4. El procesador comprueba periódicamente el estado de esos bits, hasta que detecta que la operación fue completada;
5. En caso contrario procesador espera y vuelve a comprobarlo más tarde.

Como el procesador tiene que esperar a que el módulo termine la operación, el procesador permanece ocioso durante un período de tiempo.



Gestión de periféricos

Gestión programada – Registros de estados

Registros de estados asociados a la gestión de periféricos contiene información para su configuración, control y estado de la operación de cada periférico.

Ejemplo de registros de periféricos de un microcontrolador Microchip

REGISTER 10-4: PIR1: PERIPHERAL INTERRUPT REQUEST

R/W-0	R/W-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF
bit 7							bit 0

REGISTER 17-1: SSPSTAT: MSSP STATUS REGISTER

R/W-0	R/W-0	R-0	R-0	R-0	R-0	R-0	R-0
SMP	CKE ⁽¹⁾	D/ \bar{A}	P	S	R/ \bar{W}	UA	BF
bit 7							bit 0

REGISTER 18-1: TXSTA: TRANSMIT STATUS AND CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-1	R/W-0
CSRC	TX9	TXEN ⁽¹⁾	SYNC	SENDB	BRGH	TRMT	TX9D
bit 7							bit 0

REGISTER 19-1: ADCON0: A/D CONTROL REGISTER

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	CHS3	CHS2	CHS1	CHS0	GO/DONE	ADON
bit 7							bit 0

Gestión por acceso directo

Gestión de periféricos

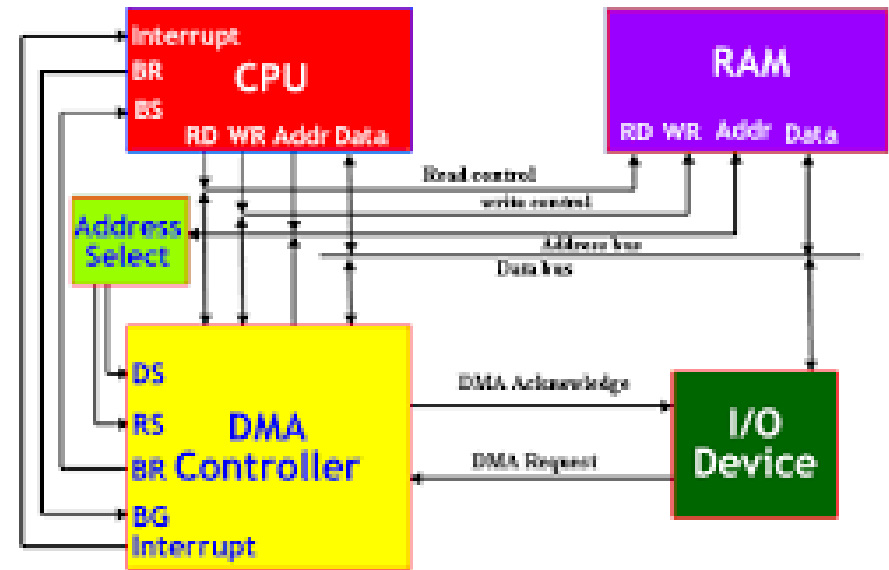
Gestión por acceso directo

El acceso directo a memoria (DMA) es mecanismo de gestión de buses que libera al procesador durante la transferencia de los datos, permitiendo a dispositivos de diferentes velocidades comunicarse sin someter a la CPU a una sobrecarga operativa.

Este mecanismo permite a los componentes de una computadora acceder a la memoria del sistema para leer o escribir independientemente de la CPU.

Una transferencia consiste en copiar bloques de memoria de un dispositivo a otro ejecutada por el controlador de DMA sin intervención de la CPU.

Las transferencias DMA son esenciales para aumentar el rendimiento de aplicaciones que requieran muchos recursos.

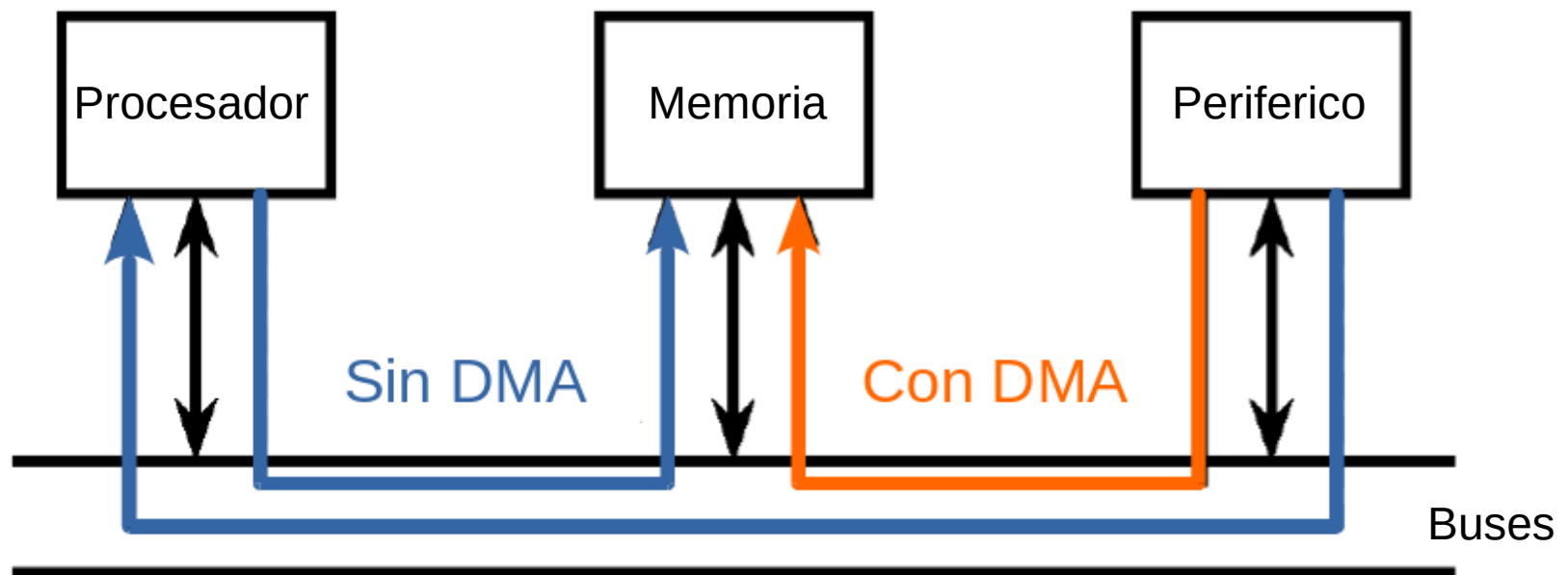


Aunque no se necesite a la CPU para la transacción de datos, se necesita el bus del sistema por lo que existen diferentes estrategias para regular su uso, permitiendo así que no quede totalmente acaparado por el controlador DMA.

El periférico avisa al controlador de DMA, la CPU programa al controlador, el cual toma control de los buses y transfiere los datos.

Gestión de periféricos

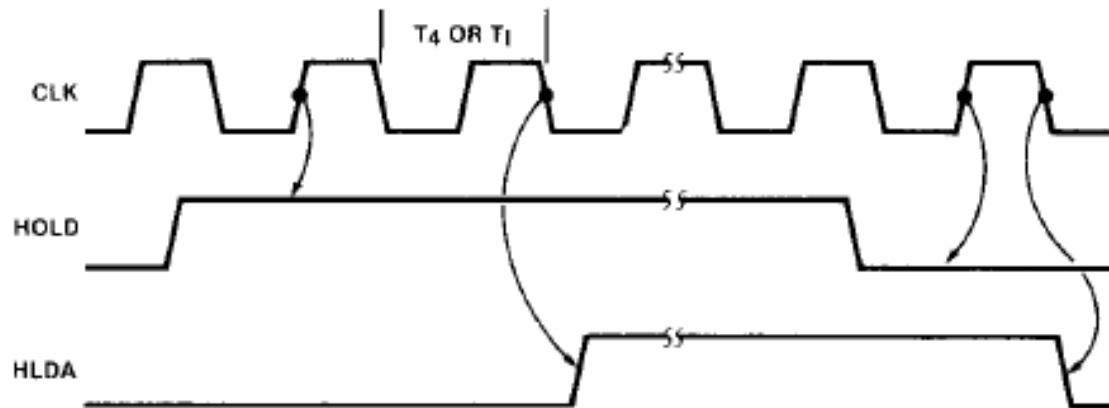
Gestión por acceso directo



Gestión de periféricos

Gestión por acceso directo

Secuencia de aceptación de DMA



Ventaja: Es rápido y eficiente para grandes bloques de datos y periféricos muy rápidos

Inconveniente: No es eficiente para bloques de datos pequeños y detiene al procesador.

Gestión de periféricos

Gestión por acceso directo

Durante las operaciones del DMA, el rendimiento del sistema puede verse afectado debido a que este dispositivo hace un uso intensivo del bus y por lo tanto la CPU no puede leer datos de memoria. Esto provoca que mientras el DMA está operando, la CPU deba esperar a que finalice dicha tarea sin ejecutar ninguna instrucción.

Para resolver esto, existe una memoria cache dentro de la CPU que permite a ésta seguir trabajando mientras el DMA mantiene ocupado el bus.

En computadores que no disponen de memoria cache, el DMA debe realizar su tarea evitando ocupar el bus de datos mientras la CPU realiza la fase de captación de la instrucción. A partir de estas posibilidades, existen dos tipos de transferencias:

Modo robo de ciclo: una vez que la CPU concede el bus al DMA, este lo vuelve a liberar al finalizar de transferir cada palabra. Teniendo que solicitar de nuevo el permiso de uso del bus a la CPU. Esta operación se repite hasta que el DMA finaliza la tarea. Este tipo de transferencia se suele usar en sistema que no disponen de memoria cache en la unidad de procesamiento, ya que de este modo, aunque la transferencia de datos tarda más en realizarse, la CPU puede seguir ejecutando instrucciones.

Modo transparente: el controlador de DMA solo transfiere datos cuando la CPU no utiliza los buses del sistema. Su principal ventaja es que nunca se detiene la ejecución del programa, pero su desventaja es que se necesita circuitería para determinar cuando la CPU no utiliza los buses.

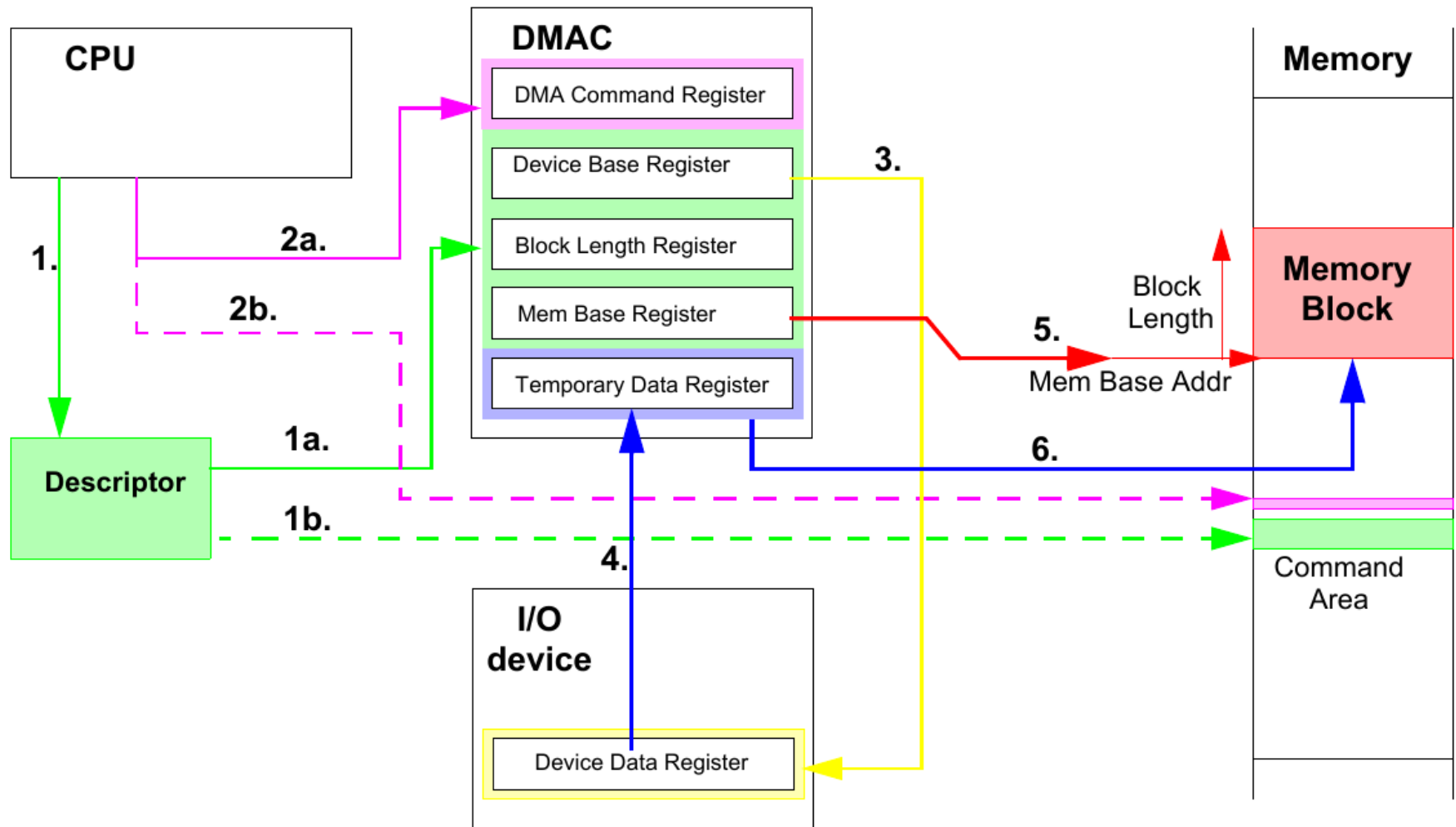
Gestión de periféricos

Gestión por acceso directo – Secuencia de operacion

1. El procesador inicializa el DMAC programando AR y WC;
2. El periférico realiza una petición de DMA al DMAC;
3. El DMAC le responde con una señal de aceptación;
4. El DMAC activa la línea de petición de DMA al procesador;
5. Al final del ciclo del bus en curso, el procesador pone las líneas del bus del sistema en alta impedancia y activa la sesión de DMA;
6. El DMAC asume el control de los buses de direcciones y control;
7. El periférico transmite una nueva palabra de datos al registro temporal del DMAC;
8. El DMAC ejecuta un ciclo de escritura en memoria para transferir el contenido del registro temporal a la posición M[AR];
9. El DMAC decrementa WC e incrementa AR;
10. El DMAC libera el bus y desactiva la línea de petición de DMA;
11. El DMAC compara WC con 0
 - 11.1 Si $WC > 0$, se repite desde el paso 7;
 - 11.2 Si $WC = 0$, el DMAC se detiene y envía una interrupción al procesador.

Gestión de periféricos

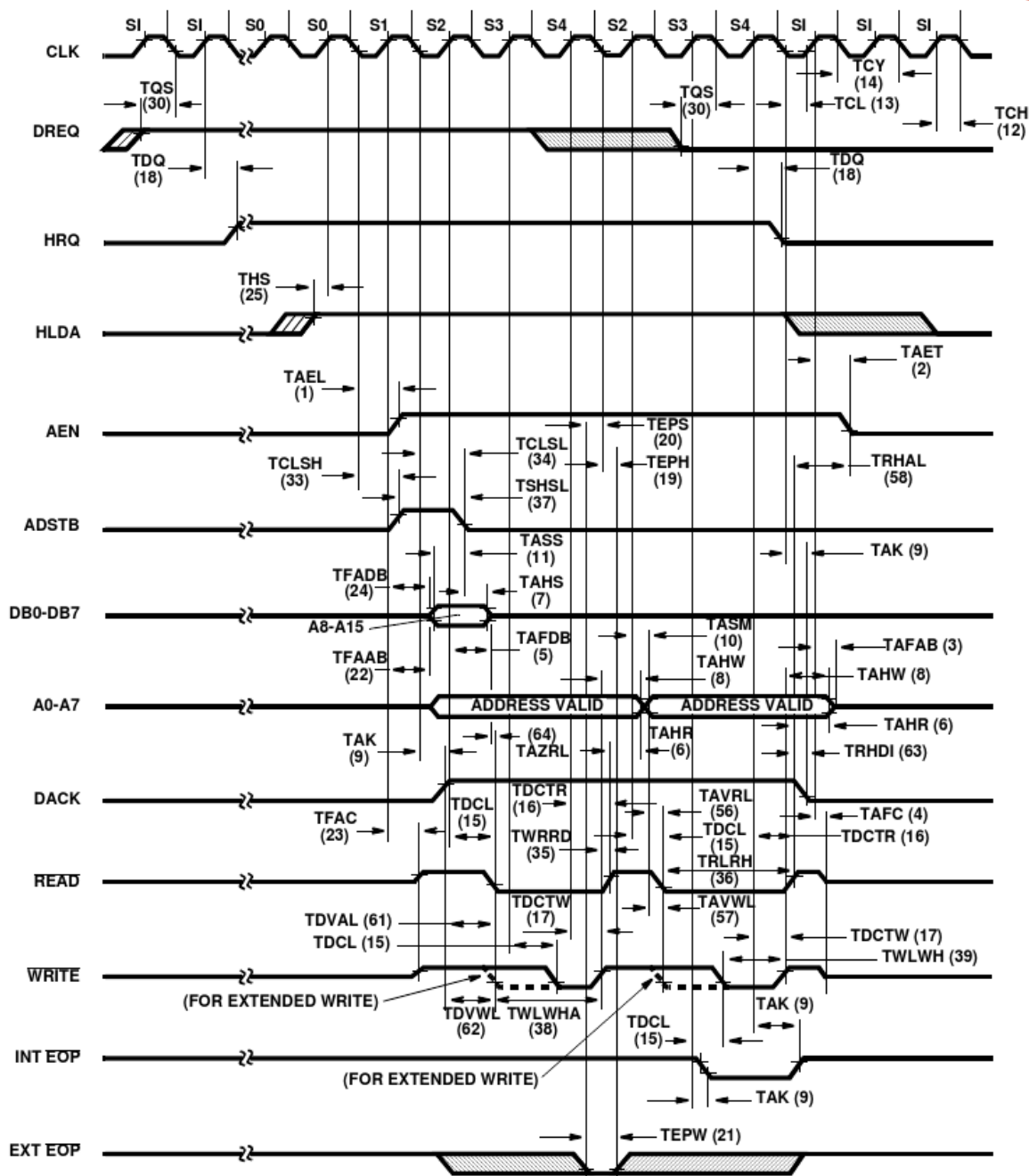
Gestión por acceso directo – Secuencia de operacion



Gestión de periféricos

Gestión por acceso directo

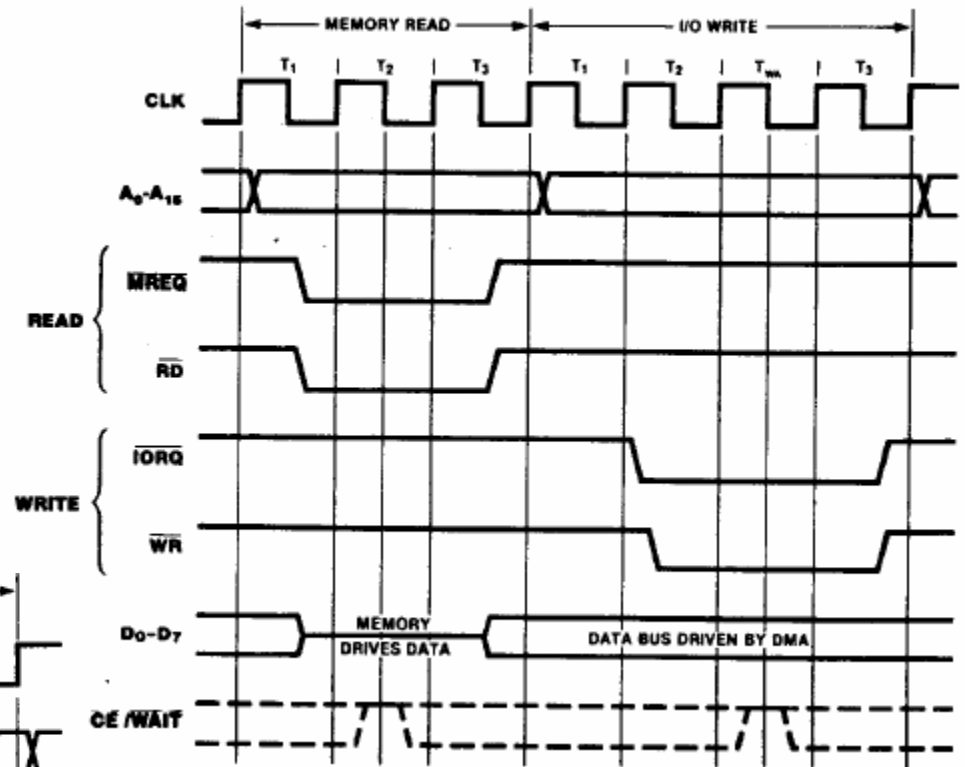
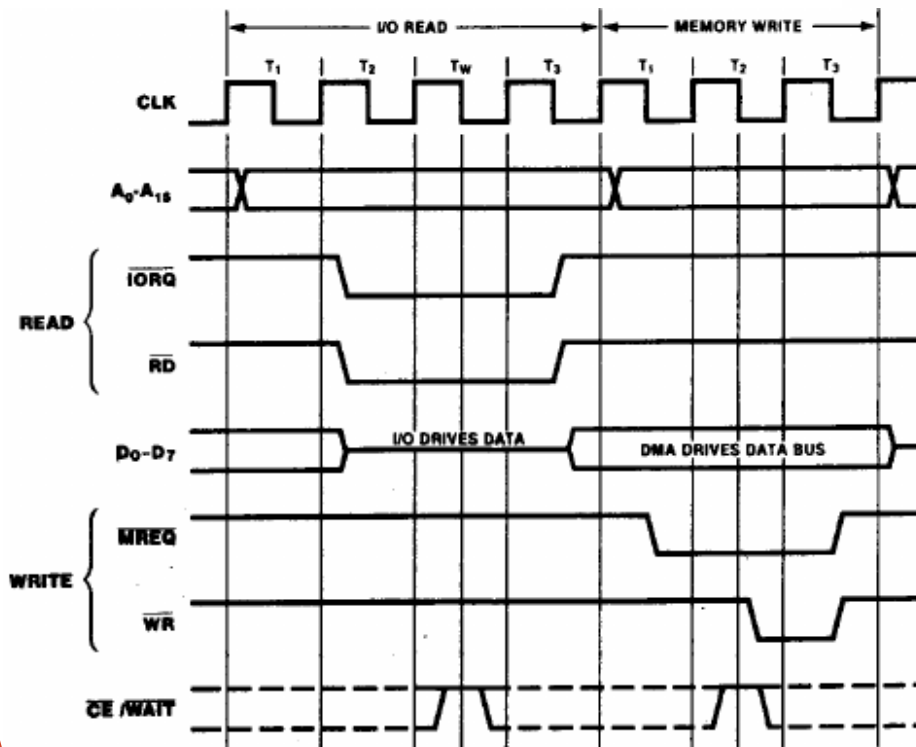
Secuencia de operacion



Gestión de periféricos

Gestión por acceso directo Secuencia de operacion

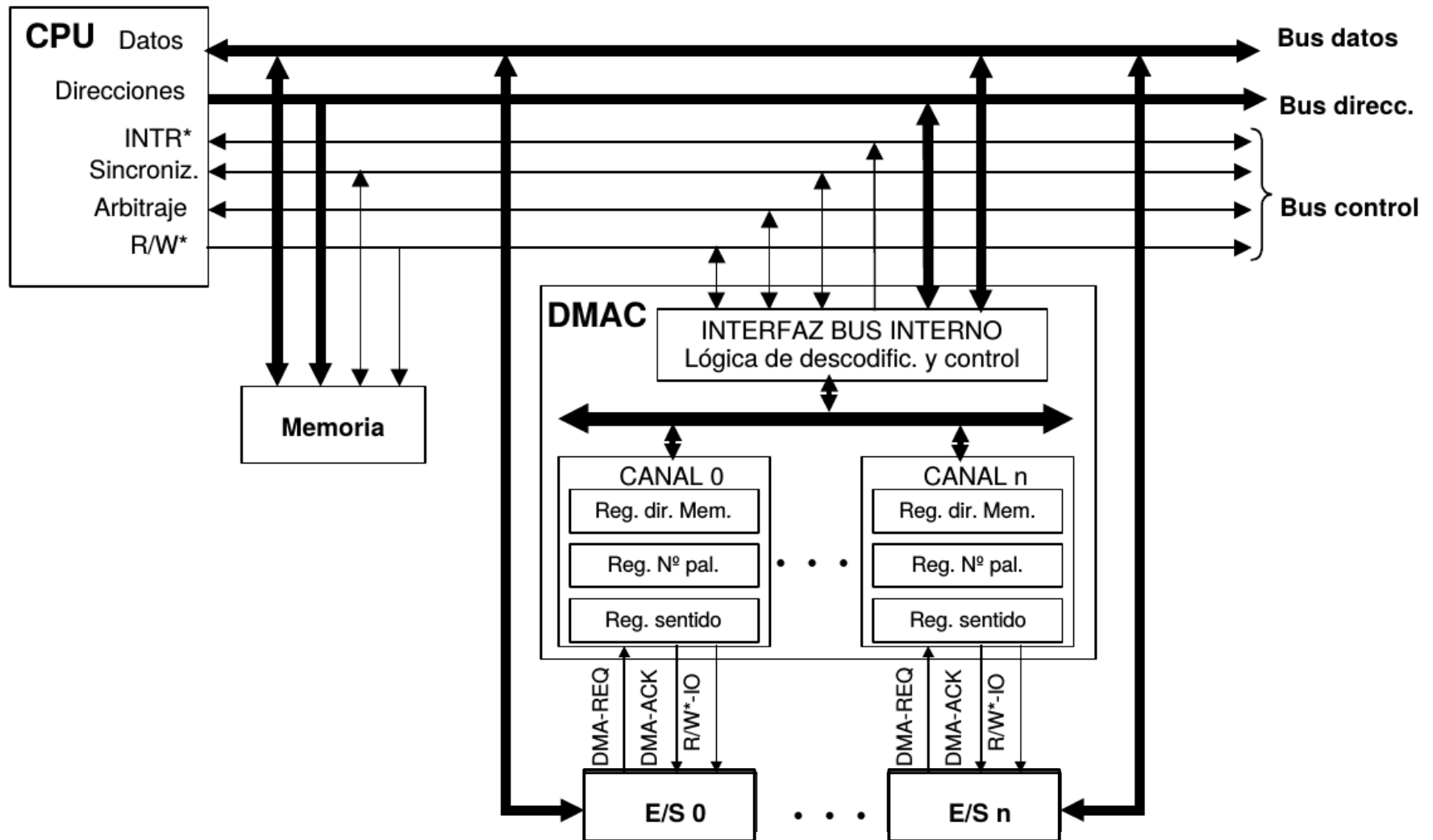
Transferencia dispositivo-memoria



Transferencia memoria-dispositivo

Gestión de periféricos

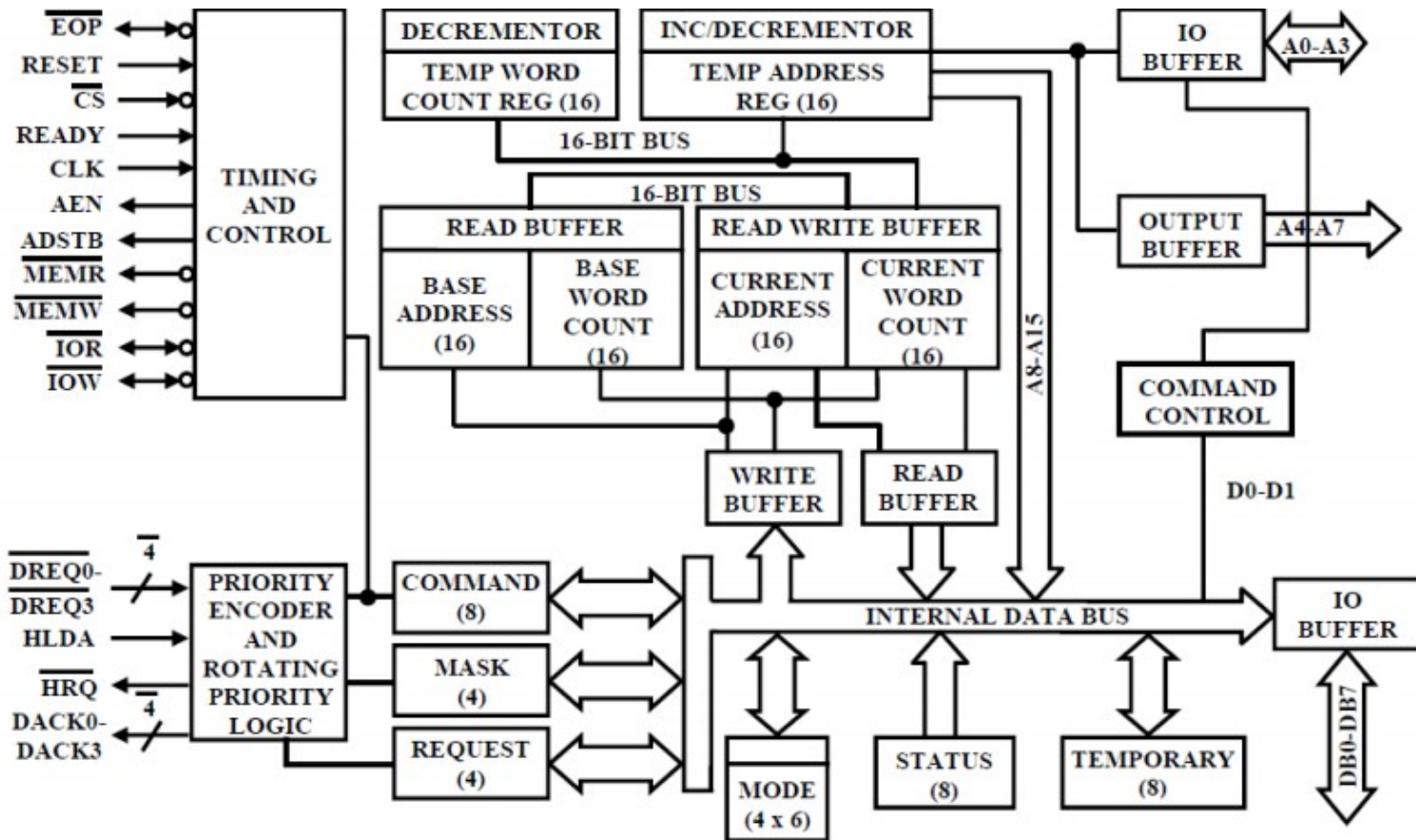
Gestión por acceso directo



Gestión de periféricos

Controlador 8237 – Estructura interna

El controlador 8237 es un generador de señales de control y dirección controlado por estado, que permite que los datos se transfieran directamente desde un dispositivo de E / S a la memoria o viceversa, sin que nunca se almacenen en un registro temporal.



Gestión de periféricos

Controlador 8237 – Registros

Current Address – este registro contiene el valor de la dirección utilizada durante las transferencias. La dirección se incrementa o disminuye automáticamente después de cada transferencia y sus valores se almacenan en este registro.

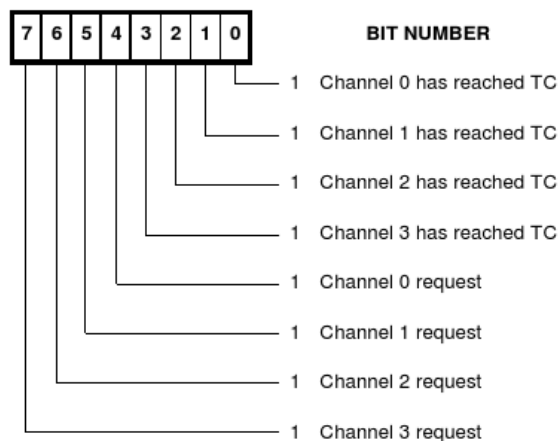
Current Word Count Register – este registro determina el número de transferencias a realizar. El número real de transferencias será uno más que el número programado.

Base Address and Base Word Count Registers – estos registros almacenan los valores originales de sus registros actuales asociados. Durante la autoinicialización, estos valores se utilizan para restaurar los registros actuales a sus valores originales.

Temporary Register – se utiliza para mantener los datos durante las transferencias de memoria a memoria. Siempre contiene el byte transferido en la operación anterior, a menos que se borre mediante un Reset.

NAME	SIZE	NUMBER
Base Address Registers	16-Bits	4
Base Word Count Registers	16-Bits	4
Current Address Registers	16-Bits	4
Current Word Count Registers	16-Bits	4
Temporary Address Register	16-Bits	1
Temporary Word Count Register	16-Bits	1
Status Register	8-Bits	1
Command Register	8-Bits	1
Temporary Register	8-Bits	1
Mode Registers	6-Bits	4
Mask Register	4-Bits	1
Request Register	4-Bits	1

Status Register



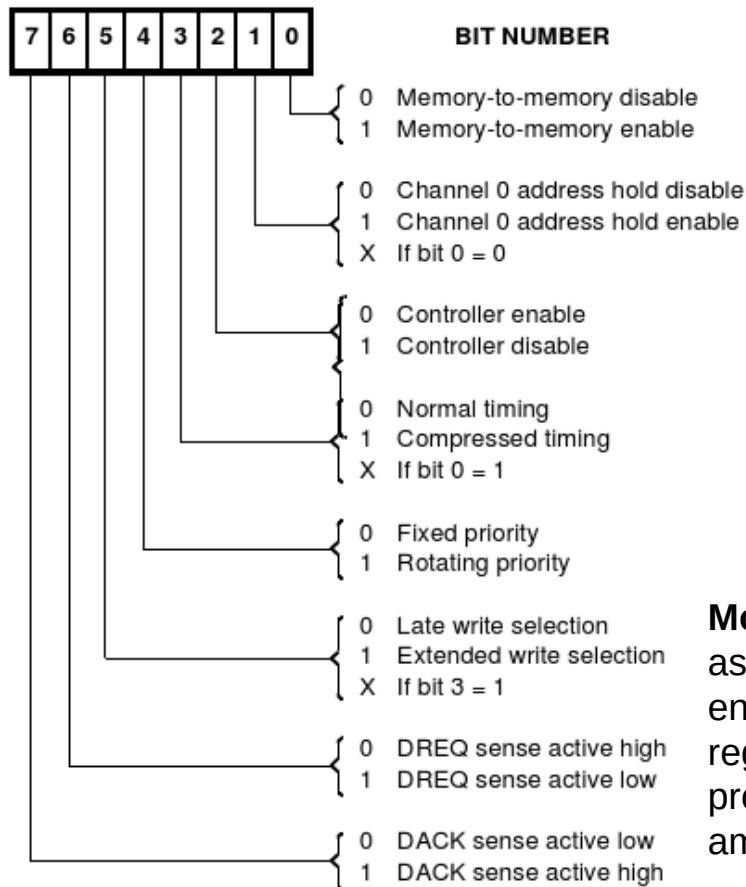
Status Register – es un registro de lectura que contiene información sobre el estado de los dispositivos. Esta información incluye qué canales han alcanzado un recuento de terminales y qué canales tienen solicitudes de DMA pendientes. Los bits 0-3 se configuran cada vez que un canal llega a un TC o se aplica un EOP externo. Estos bits se borran en RESET, Master Clear y en cada estado R0.

Gestión de periféricos

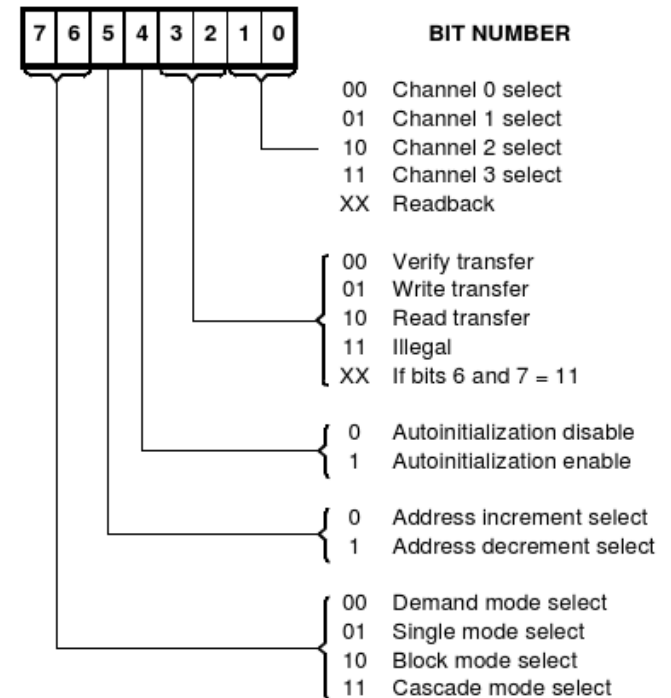
Controlador 8237 – Registros

Command Register – este registro controla el funcionamiento del 82C37A. Es programado por el procesador y se borra mediante RESET o una instrucción de borrado maestro.

Command Register



Mode Register



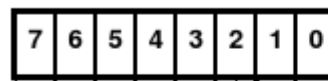
Mode Register – cada canal tiene un registro de modo asociado. Cuando el procesador está escribiendo en el registro en la condición de programa, los bits 0 y 1 determinan qué registro de Modo de canal se va a escribir. Cuando el procesador lee un registro de Modo, los bits 0 y 1 serán ambos.

Gestión de periféricos

Controlador 8237 – Registros

Mask Register – cada canal tiene asociado un bit de máscara que puede configurarse para deshabilitar un DREQ entrante. Cada bit de máscara se establece cuando su canal asociado produce un EOP, si el canal no está programado para autoinicializar. Cada bit del registro también se puede configurar o borrar por separado o simultáneamente bajo el control del software. Todo el registro también se configura mediante un restablecimiento o borrado maestro.

Mask Register



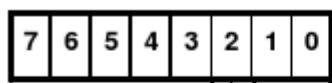
Don't Care

BIT NUMBER

- 00 Select Channel 0 mask bit
- 01 Select Channel 1 mask bit
- 10 Select Channel 2 mask bit
- 11 Select Channel 3 mask bit

- 0 Clear mask bit
- 1 Set mask bit

Request Register

Don't Care,
Write
Bits 4-7
All Ones,
Read

BIT NUMBER

- 00 Select Channel 0
- 01 Select Channel 1
- 10 Select Channel 2
- 11 Select Channel 3

- 0 Reset request bit
- 1 Set request bit

Request Register – el dispositivo puede responder a solicitudes de servicio DMA que se inician por software o con un DREQ. Cada canal tiene un bit de solicitud asociado en el registro de solicitud. Estos no son enmascarables y están sujetos a la priorización por parte de la red del Codificador de Prioridad. Cada bit de registro se establece o restablece por separado bajo el control del software. Todo el registro se borra con una instrucción de restablecimiento o borrado maestro. Para establecer o restablecer un bit, el software carga la forma correcta de la palabra de datos.

Gestión de periféricos

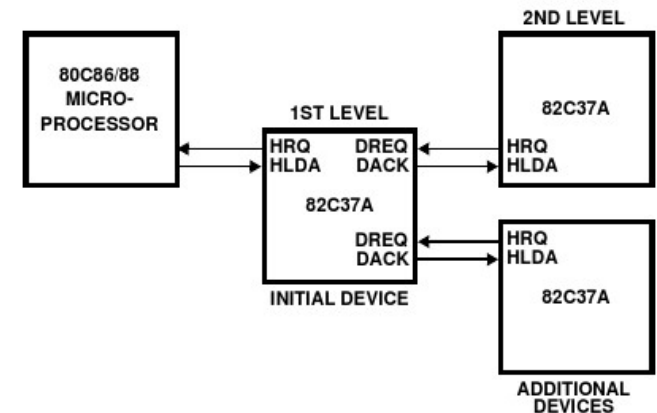
Controlador 8237 – Modos de transferencias

Single Transfer Mode – el dispositivo está programado para hacer solo una transferencia. El recuento de palabras y la dirección disminuirán o se incrementará después de cada transferencia. Cuando el recuento de palabras pasa de cero a FFFFH, se establece un bit de recuento de terminal en el registro de estado, se genera un EOP, y el canal se autoinicializará si se ha seleccionado esta opción;

Block Transfer Mode - el dispositivo se activa mediante DREQ o solicitud de software y continúa realizando transferencias durante el servicio hasta que se produce un TC debido a que el recuento de palabras va a FFFFH, o se encuentra un EOP externo. DREQ solo necesita mantenerse activo hasta que DACK se active;

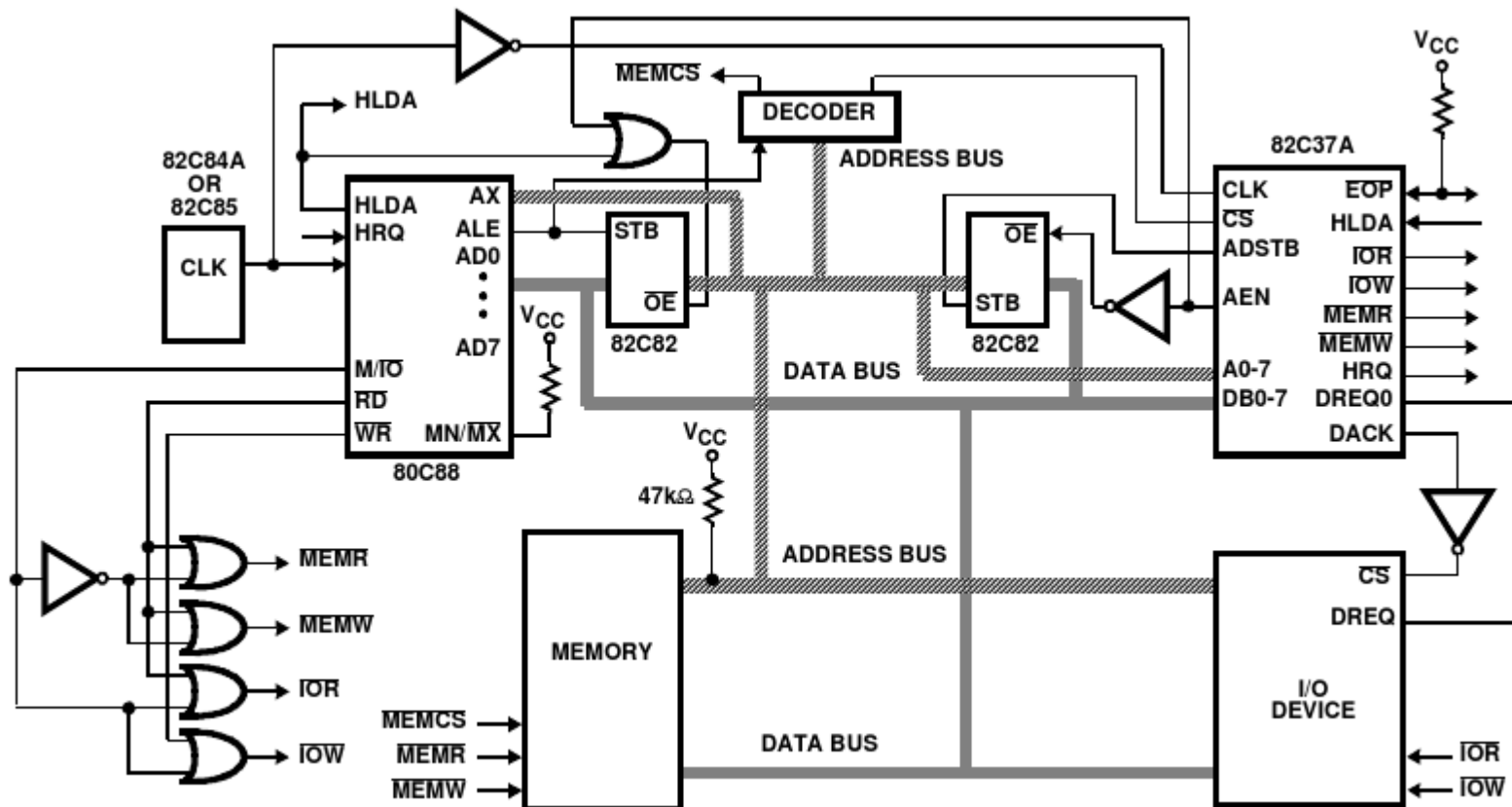
Demand Transfer Mode – el dispositivo continúa realizando transferencias hasta que encuentra un TC o un EOP externo, o hasta que DREQ se desactiva. Por lo tanto, la transferencia puede continuar hasta que el periférico haya agotado su capacidad de datos. Después de que el periférico haya tenido la oportunidad de ponerse al día, el servicio DMA se restablece mediante un DREQ. Durante el tiempo entre servicios cuando el procesador puede operar, los valores intermedios de dirección y conteo de palabras se almacenan en los registros Current Address y Current Word Count;

Cascade Mode – este modo se utiliza para conectar en cascada varios dispositivos para la expansión del sistema. Las señales HRQ y HLDA del dispositivo adicional se conectan a las señales DREQ y DACK respectivamente de un canal para el dispositivo inicial. Esto permite que las solicitudes DMA del dispositivo adicional se propaguen a través de los circuitos de red de prioridad del dispositivo anterior. La cadena de prioridad se conserva y el nuevo dispositivo debe esperar su turno para reconocer las solicitudes.



Gestión de periféricos

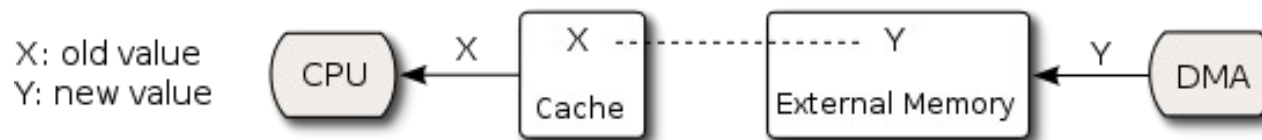
Controlador 8237 – Sistema con DMA



Gestión de periféricos

Gestión por acceso directo con Cache

La DMA puede llevar a problemas de **coherencia de cache** si tenemos una CPU equipada con memoria caché y una memoria externa que se pueda ser accedidas directamente por los dispositivos que utilizan DMA.



Cuando la CPU accede a la posición de memoria X, el valor actual se almacena en la caché. Si se realizan operaciones posteriores en X, se actualizará la copia en caché de X, pero no la versión de memoria externa de X. Si la caché no se vacía en la memoria antes de que otro dispositivo intente acceder a X, el dispositivo recibirá un valor caducado de X.

Del mismo modo, si la copia en caché de X no es invalidada cuando un dispositivo escribe un nuevo valor en la memoria, entonces la CPU utilizará un valor caducado de X.

Gestión de periféricos

Gestión por acceso directo con Cache

Este problema se resuelve de las siguientes formas

- **Sistemas de caché coherente:** implementan un método en el hardware externo mediante el cual se escribe una señal en el controlador de caché, la cual realiza una invalidación de la caché para escritura de DMA o caché de descarga para lectura de DMA;
- **Sistemas de caché no-coherente:** el sistema operativo debe asegurarse de que las líneas de caché se vacían antes de que una transferencia de salida de DMA sea iniciada y anulada antes de que una parte de la memoria sea afectada por una transferencia entrante de DMA que se haya requerido. El sistema operativo debe asegurarse de que esa parte de memoria no es accedida por cualquier subproceso que se ejecute en ese instante.

Este enfoque introduce **sobrecarga a la operación de DMA**, ya que la mayoría de hardware requiere un bucle para invalidar **cada línea de caché de forma individual**.

- **Sistemas de caché híbridos:** la caché secundaria L2 es coherente, mientras que en la caché L1 (generalmente la CPU) es gestionado por el software.