Microprocesadores. El MC68000

1. Introducción
2. Características principales del MC68000
3. Registros del modelo de programación
4. Organización de la memoria
5. Buses del Sistema
6. Formato de instrucción máquina
7. Modos de direccionamiento
8. Conjunto de instrucciones
9. Procesamiento de excepciones

Microprocesadores. El MC68000

Bibliografía:

Julio Septien y otros. “La Familia del MC68000” Editorial Sintesis.

Microprocesadores. El MC68000

Objetivos:

- Concretar los conceptos generales sobre computadores mediante el estudio de un microprocesador real.
- Describir el modelo de usuario.
- Describir los buses del sistema (Patillas del CI).
- Conocer los modos de direccionamiento y justificar su utilidad.
- Conocer sus instrucciones y realizar rogramas en ensamblador.
- Describir las situaciones en las que se pueden producir excepciones.
- Describir el funcionamiento del microprocesador durante el procesamiento de una excepción.
- Estudiar las interrupciones como caso particular de las excepciones.

1. Introducción

- Microprocesador: es un procesador en un sólo Circuito integrado incluyendo:
  - Unidad de control
  - Unidad aritmético-lógica y banco de registros
  - buses internos de datos direcciones y control
- Los computadores actuales se basan en microprocesadores.
- Características de los microprocesadores:
  - Repertorio de instrucciones.
  - Frecuencia de reloj.
  - Anchura de los buses internos.
  - Tamaño de los registros.
  - Número de patillas de encapsulamiento.

1. Introducción

Microprocesadores más conocidos

- Fabricantes: Intel, AMD, Motorola, Zilog, Texas Instruments, etc.
- Microprocesadores de 4 bits: 4004.
- Microprocesadores de 8 bits: 8008, 8080, 6800, Z-80. 8085.
- Microprocesadores de 16 bits: 8086, 68000, 80286.
- Microprocesadores de 32 bits: 68020, 68030, 68040, 80386, 80486, Pentium, PowerPC.
- Microprocesadores de 64 bits: Alpha, SPARC, R4000.
- Microprocesadores de 128??? bits: CELL IBM Sony y Toshiba

1. Introducción

Microprocesadores y procesadores de propósito específico

- Microcontroladores: Son computadores en una sola pastilla, incluyendo memoria e interfaz para entrada/salida.
- Procesadores digitales de señal (DSP): adaptados para realizar tratamiento de señales digitales. (por ejemplo en un modem ADSL).
- Otros procesadores de propósito especifico: E/S, controladores de red local.

1. Introducción

- Abordaremos el estudio de los microprocesadores eligiendo uno concreto y no de forma general.
- El microprocesador elegido es el 68000 de motorola. Aunque es antigüo tiene el nivel de complejidad adecuado para la asignatura y existen mucha y muy adecuada bibliografía para su estudio.
- Se realizará el estudio desde dos puntos de vista:
  - Estructural: registros internos accesibles al programador y descripción de las patillas.
  - Funcional: modos de direccionamiento, repertorio de instrucciones y tipos de instrucciones.

2. Características principales del MC68000

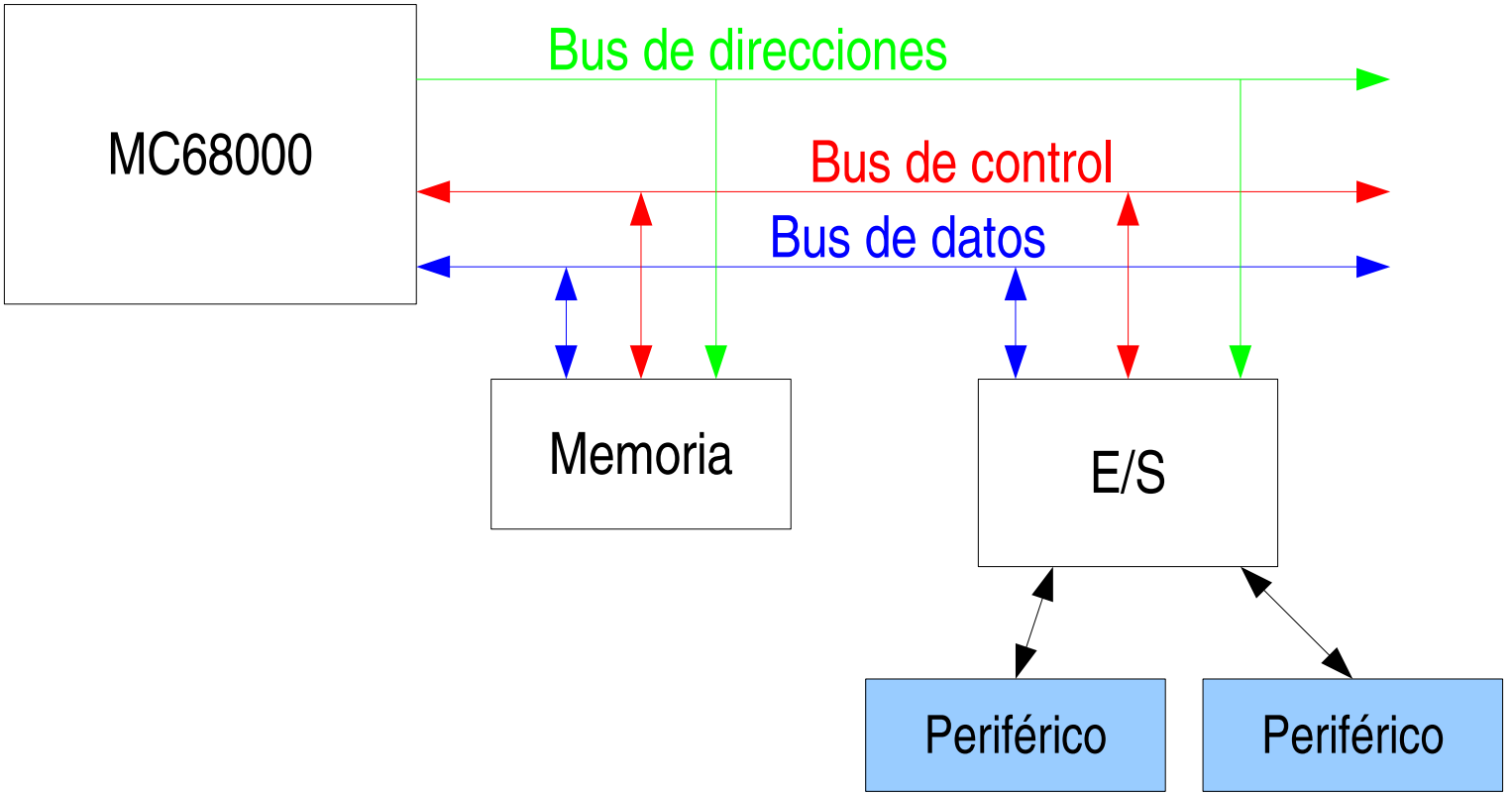
- Máquina CISC (computador con repertorio de instrucciones complejo).
- Tamaño de los buses:
  - Datos:16 bits.
  - Direcciones: 23 bits, internamente se trabaja con 24 bits de direcciones pero el bit menos significativo no sale aunque podemos saber su valor a través de las señales de control LDS y UDS.
- Tamaño de los datos:
  - Octeto (byte):.B
  - Palabra (word): .W
  - Palabra larga (loword):.L
  - Bits (agrupados de 8 en 8 en un octeto).
  - Dígitos BCD (empaquetados de dos en dos en un octeto).

2. Características principales del MC68000

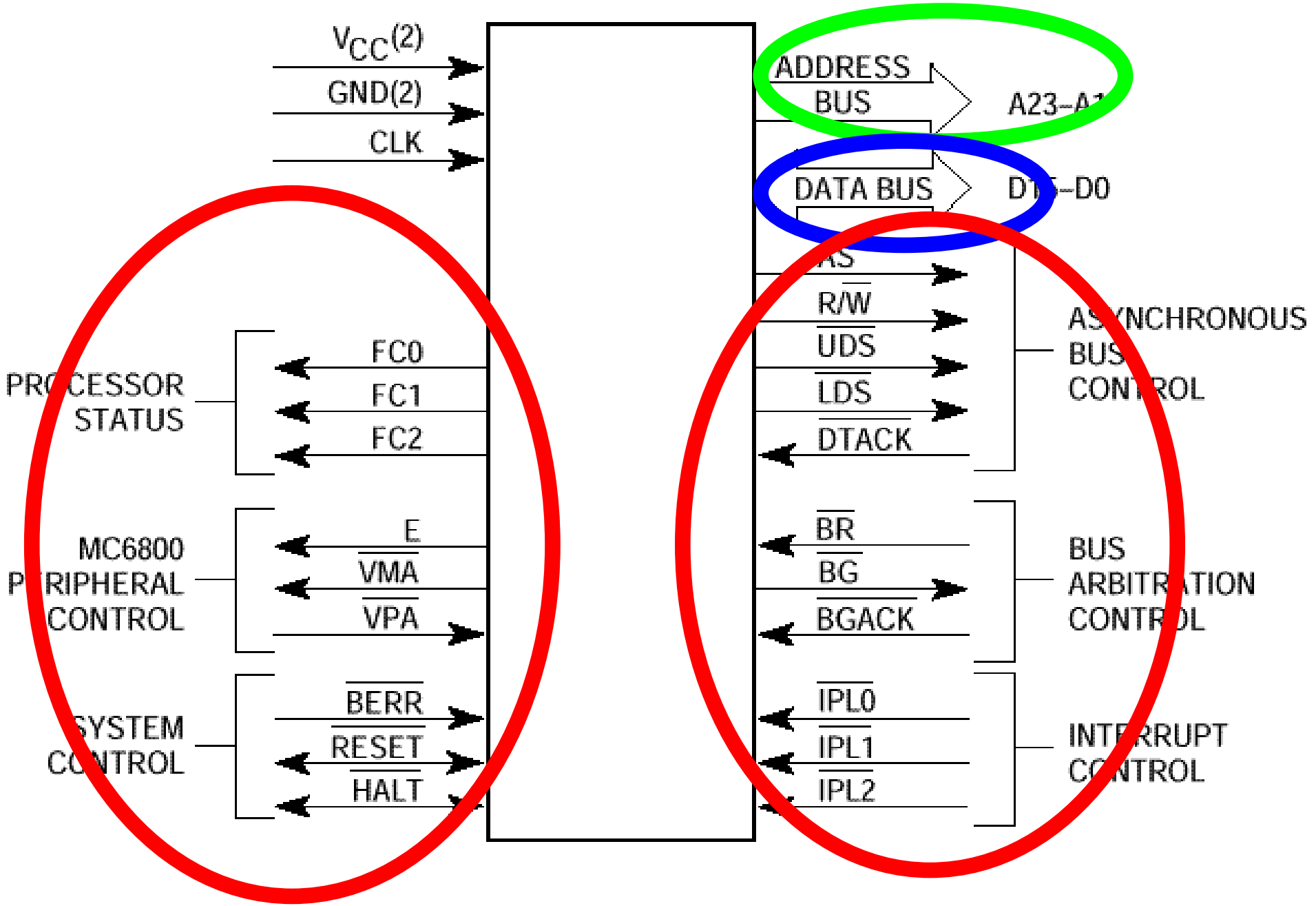
- Esquema de bus único para memoria y E/S.
- Modos de funcionamiento:
  - usuario: Los programas normales funcionan en este modo, existen instrucciones privilegiadas que no pueden utilizarse en este modo.
  - supervisor: se puede acceder a todos los recursos de la máquina sin restricciones.

2. Características principales del MC68000

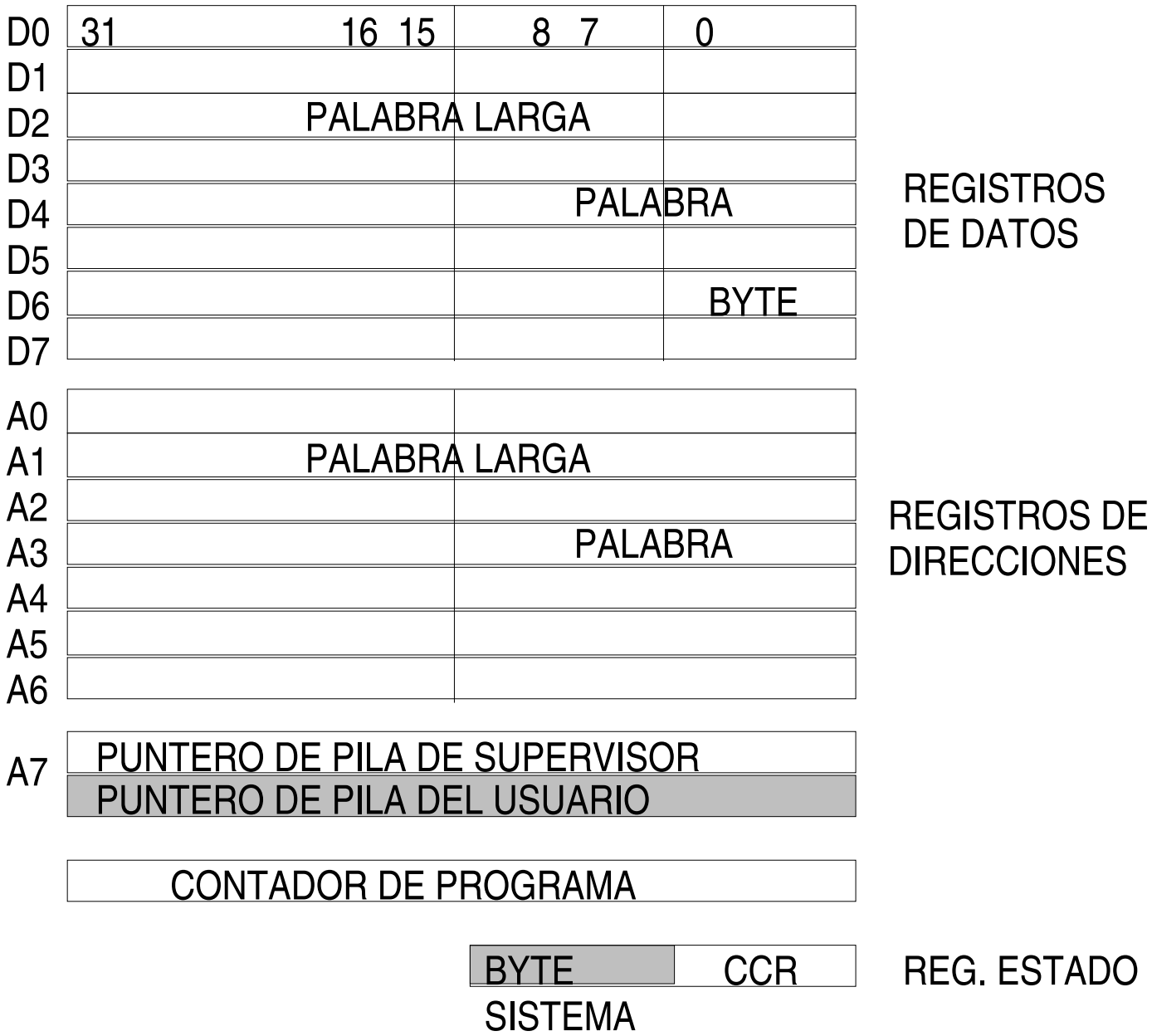
Un computador basado en el MC68000



2. Características principales del MC68000

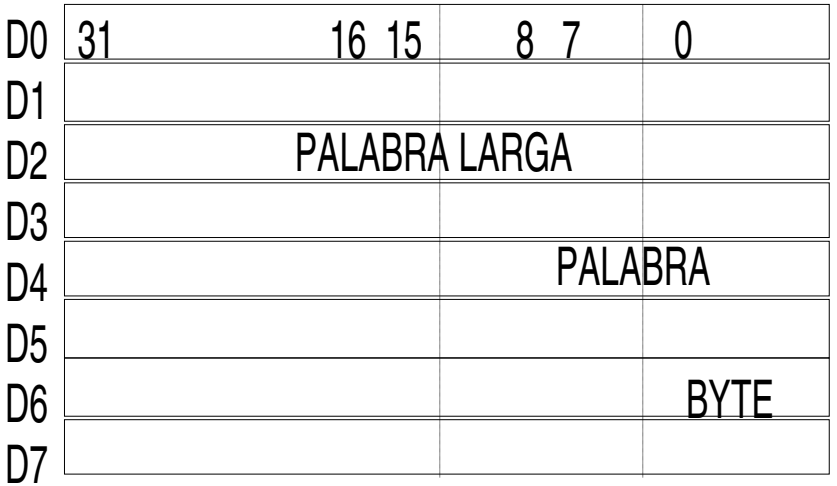


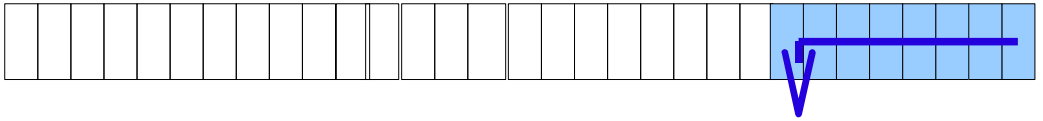
3. Registros del modelo de programación



3. Registros del modelo de programación

Registros de datos





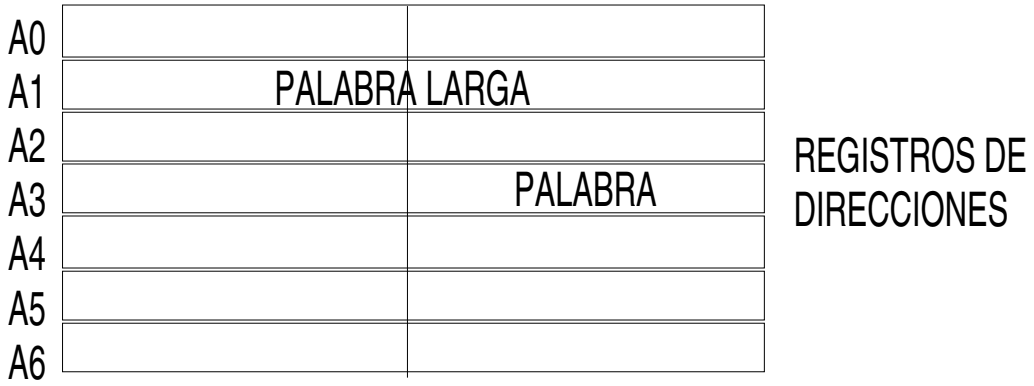
REGISTROS
DE DATOS

DEPLAZAMIENTO A LA IZQDA TAMAÑO BYTE

- Tamaño:32 bits
- Modos de acceso:
  - Octeto (.B): se accede sólo a los 8 bits menos significativos
  - Palabra (.W): se accede solo a los 16 bits menos significativos
  - Palabra Larga o doble (.L): se accede al registro completo.
- Uso: para almacenar datos temporales.

3. Registros del modelo de programación

Registros de direcciones



- Tamaño:32 bits
- Modos de acceso:
  - Palabra (.W): se accede solo a los 16 bits menos significativos
    *En escritura en modo palabra, se realiza la extensión del signo a 32 bits
  - Palabra Larga o doble (.L): se accede al registro completo.
  - Uso; Puntero a datos o estructura de datos en memoria
    * Como la memoria máxima tiene 2^{24} octetos , al usar un registro de direcciones como puntero se descarta su octeto más significativo
- Operaciones con estos registros no modifican los bits del CCR

3. Registros del modelo de programación

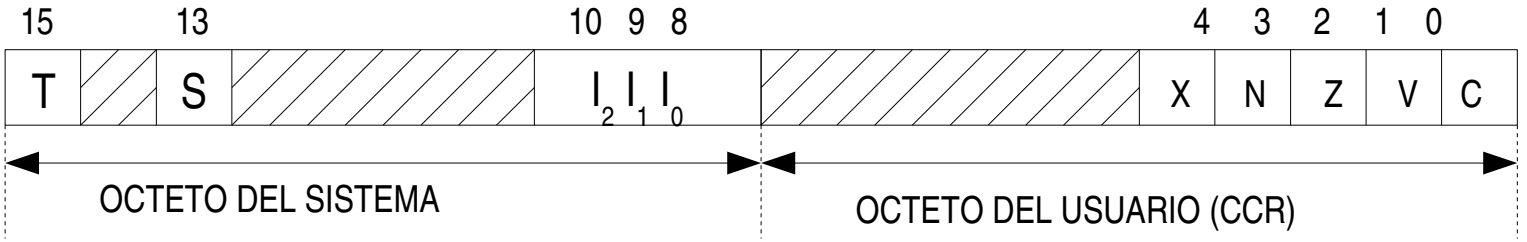
Punteros de propósito específico

- Contador del programa (PC)
  - Tiene 32 bits de los cuales sólo se usan los 24 menos significativos.
  - Actúa como puntero a la próxima instrucción que se va a ejecutar (entre instrucciones) o a la próxima palabra de una instrucción durante el ciclo de búsqueda.

- Punteros de pila, el MC68000 tiene dos punteros de pila:
  - Puntero de pila de usuario (USP): visible por el usuario como SP o A7
  - Puntero de pila de supervisor (SSP): accesible sólo por el supervisor como SP 0 A7.
  - Son de 32 bits.
  - Funcionan como registros de direcciones.
  - Tienen un uso específico en determinadas instrucciones

3. Registros del modelo de programación

El registro de estado del MC68000. SR (status register)



- CCR (condition code register o registro de códigos de condición):
  - C: acarreo
  - V: desbordamiento en números con signo.
  - Z: resultado igual a cero.
  - N: resultado negativo.
  - X: bit de extensión. utilizado en operaciones aritméticas de múltiple precisión y de desplazamiento.
- Octeto del sistema.
  - T: modo traza (1=se genera una excepción al ejecutar una instrucción; 0=se ejecutan las instrucciones normalmente)
  - S: modo de funcionamiento (1=supervisor; 0= usuario).
  - I_2 I_1 $I_{0:}$ nivel actual de prioridad de la interrupción.

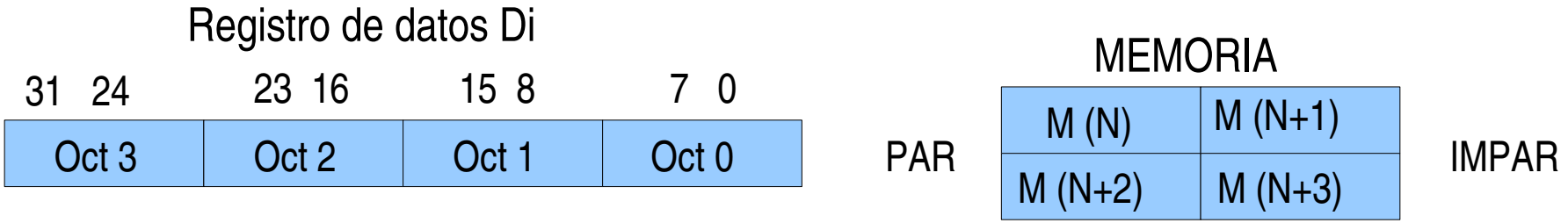
4. Organización de la memoria

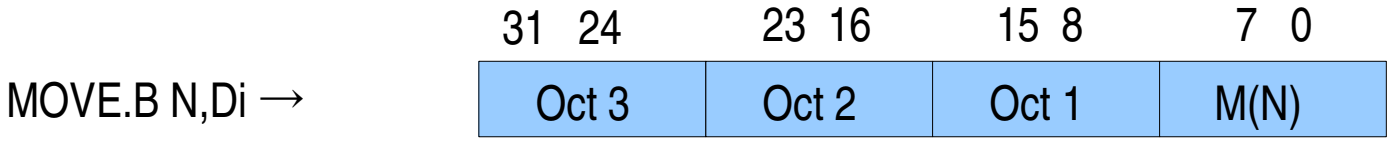
- Espacio de direcciones de 16 Mbytes:
  - $A_{23}A_{22.}..A_1A_0$ para dirección interna 2^{24}: 16Mposiciones: 1 posición=1byte.
  - En este espacio se puede incluir cualquier tipo de memoria (RAM o ROM), registro o dispositivo de E/S.
- Bus de direcciones de 23 bits y bus de datos de 16 bits (8Mword).
  - AB[23]: $A_{23}A_{22.}..A_1$ para la dirección exterior 2^{23}= 8 Mposiciones: 1 posición 1 Word=1palabra=2 Bytes. Con ello, la capacidad total es 8 Mword= (M .2Bytes=16MBytes.

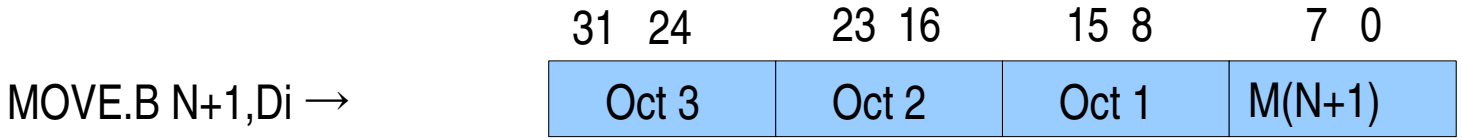
4. Organización de la memoria

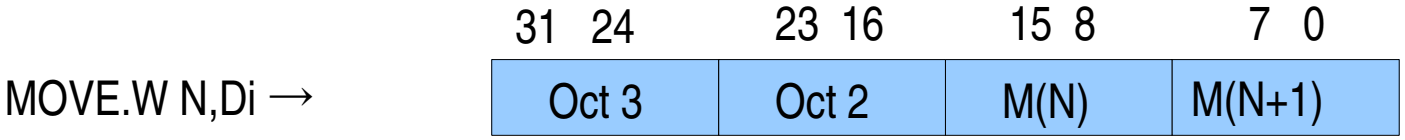
- **Acceso a un dato tamaño octeto:**
  - el octeto puede estar en direcciones pares (se activa UDS) o impares (se activa LDS)
- **Acceso a un dato tamaño palabra:**
  - La palabra siempre debe comenzar en una dirección par a la que llamaremos N (se activan UDS y LDS).
  - Big endian: el octeto más significativo es el de la dirección par (N), y el menos significativo es el de la dirección impar (N+1).
- **Acceso a palabra larga:**
  - La doble palabra comienza en una dirección par a la que llamaremos N (se activan UDS y LDS).
  - Se realizan dos accesos consecutivos a la memoria.
  - Big endian: el octeto más significativo se encuentra en la dirección N , el siguiente en la N+1 (impar), el siguiente en la dirección N+2 (par) y el menos significativo en la dirección N+3 (impar)

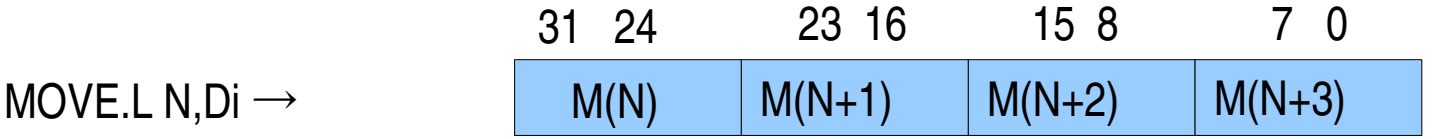
4. Organización de la memoria











4. Organización de la memoria

Representación de los datos

Ejemplo: Representar en memoria los números primos (1,2,3,5)
desde $001122

BCD Empaquetado

	PAR	IMPAR
$001122	00010010	00110101
$001124		
$001126		
$001128		
$00112A		

Binario sin signo tamaño byte

	PAR	IMPAR
$001122	00000001	00000010
$001124	00000011	00000101
$001126		
$001128		
$00112A		

Binario sin signo tamaño word

	PAR	IMPAR
$001122	00000000 00000001	
$001124	00000000 00000010	
$001126	00000000 00000011	
$001128	00000000 00000101	
$00112A		

Binario sin signo tamaño long word

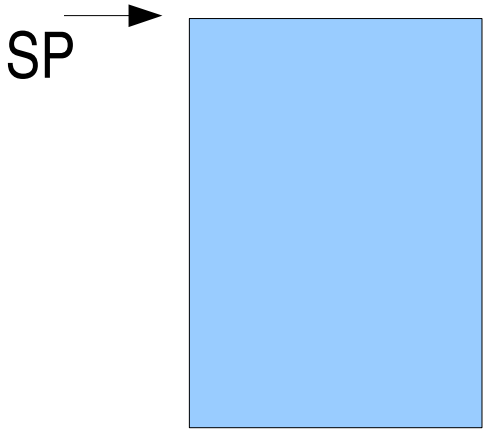
	PAR	IMPAR
$001122	00000000 00000000	
$001124	00000000 00000001	
$001126	00000000 00000000	
$001128	00000000 00000010	
$00112A	00000000 00000000	
$00112C	00000000 00000011	
$00112E	00000000 00000000	
$001130	00000000 00000101	

4. Organización de la memoria: Estructura de la Pila

- Estructura LIFO (último en entrar-primero en salir).
- Puntero de Pila: A7 o SP.
- Pila de usuario y de supervisor independientes.
  - USP: puntero de pila de usuario.
  - SSP: puntero de pila de supervisor.
- La pila crece hacia posiciones crecientes de la memoria, y el puntero apunta hacia el último elemento insertado en la misma (si no hay ninguno, apunta al fondo de pila).
  - Inserción: MOVE.{WL} DATO, -(SP)
  - Extracción: MOVE.{WL} (SP)+, DATO
- Es posible acceder a cualquier dato almacenado en la pila.
  - Direccionamientos relativos al puntero de pila.
  - Direccionamientos relativos al puntero de marco si está definido.

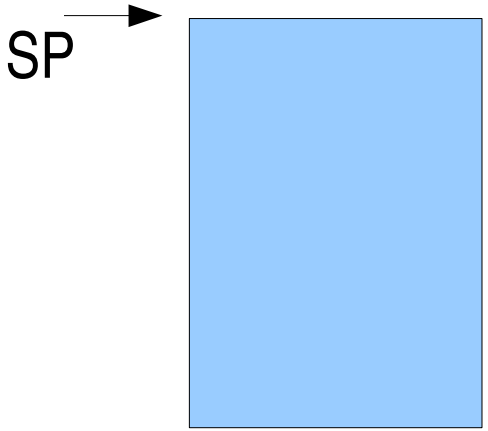
Organización de la memoria: Estructura de la Pila

ANTES



Organización de la memoria: Estructura de la Pila

ANTES

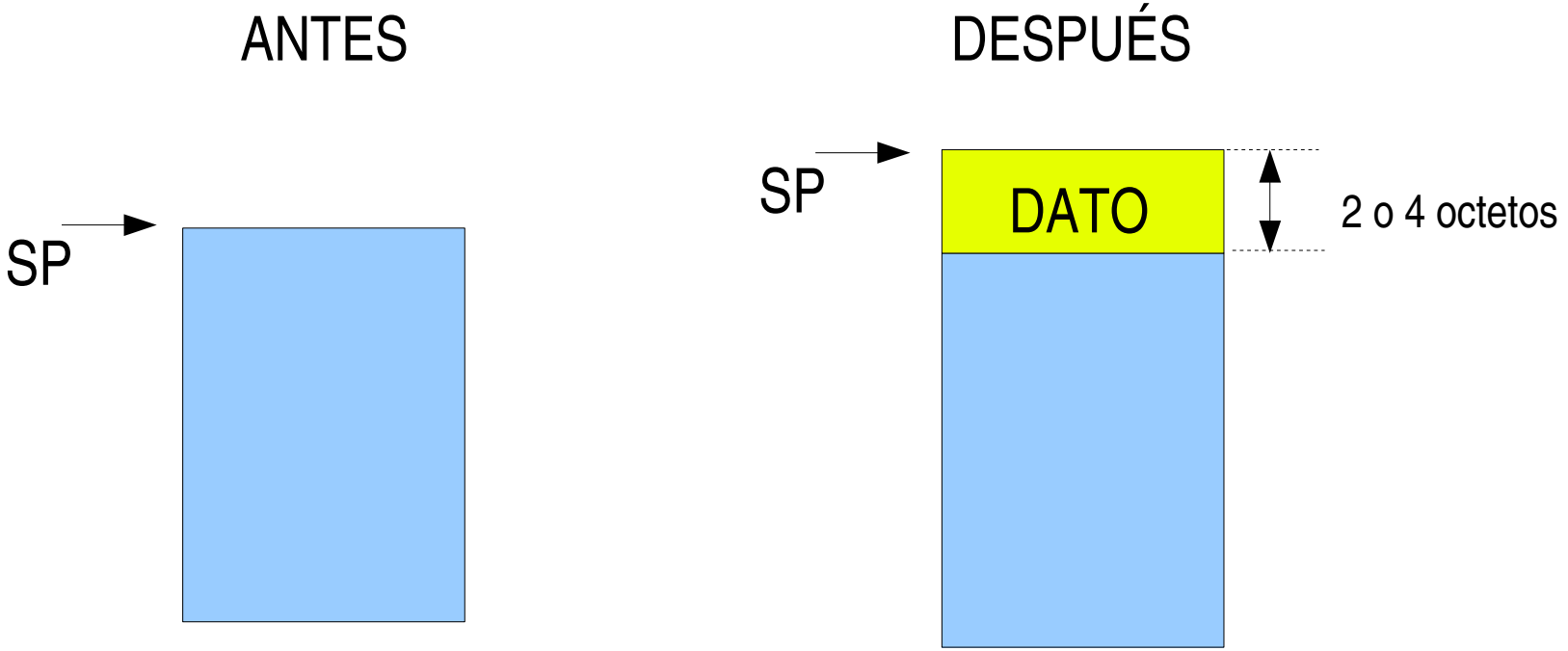


Inserción

MOVE.{WL} DATO, -(SP)

El contenido del dato se almacena en la cima de la pila

Organización de la memoria: Estructura de la Pila

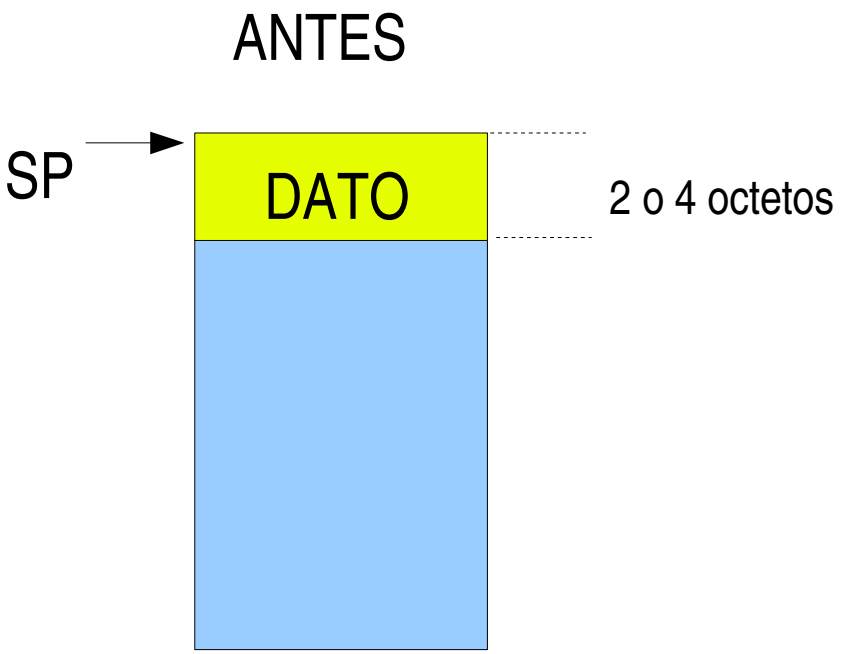


Inserción

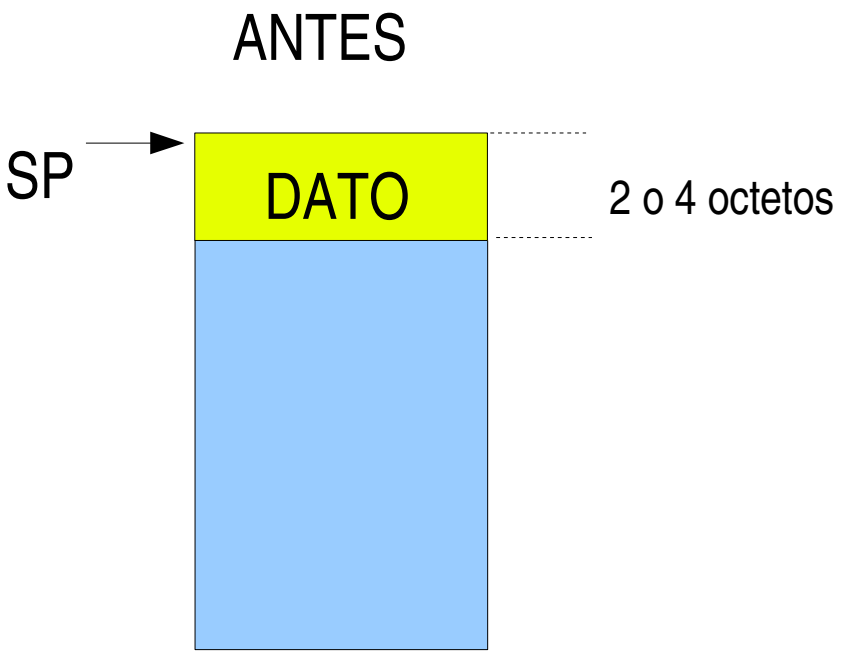
MOVE.{WL} DATO, -(SP)

El contenido del dato se almacena en la cima de la pila

Organización de la memoria: Estructura de la Pila



Organización de la memoria: Estructura de la Pila

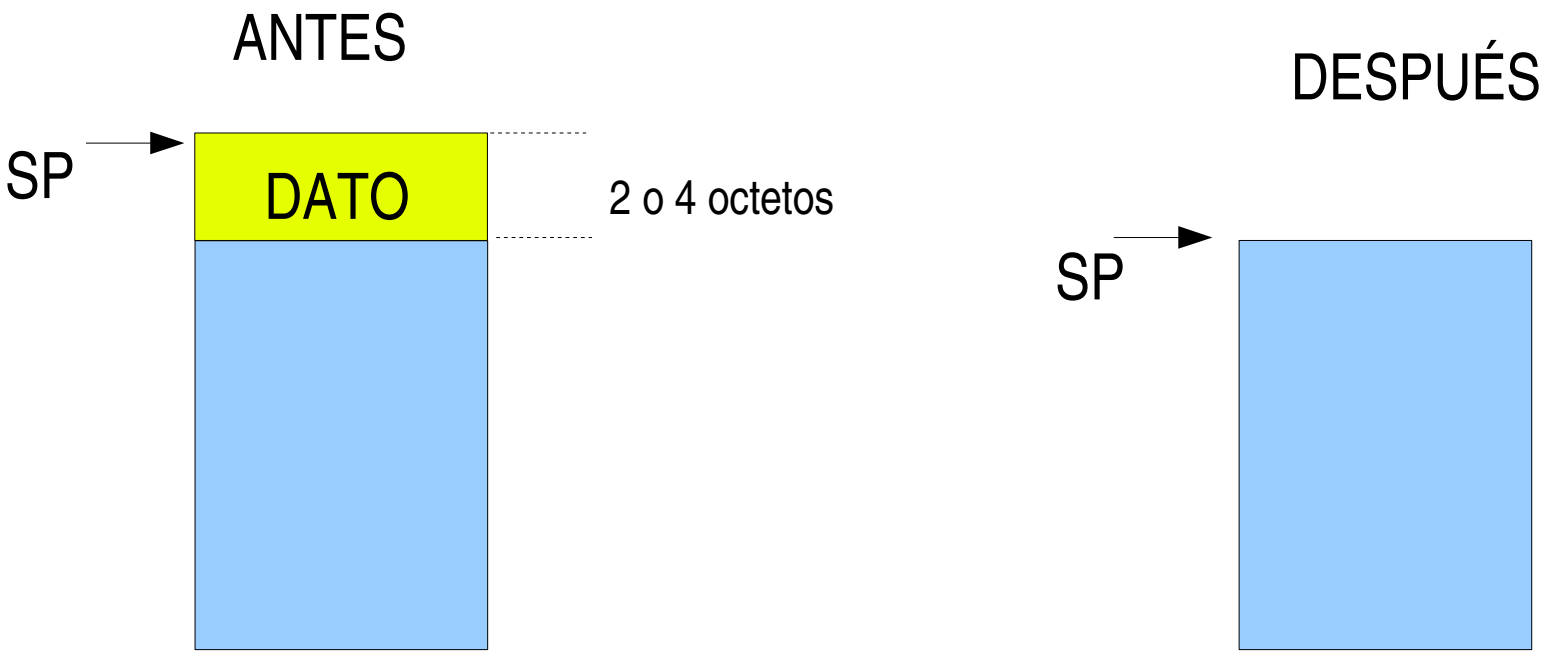


Extracción

MOVE.{WL} (SP)+,DATO

La información extraída se almacena en dato

Organización de la memoria: Estructura de la Pila



Extracción

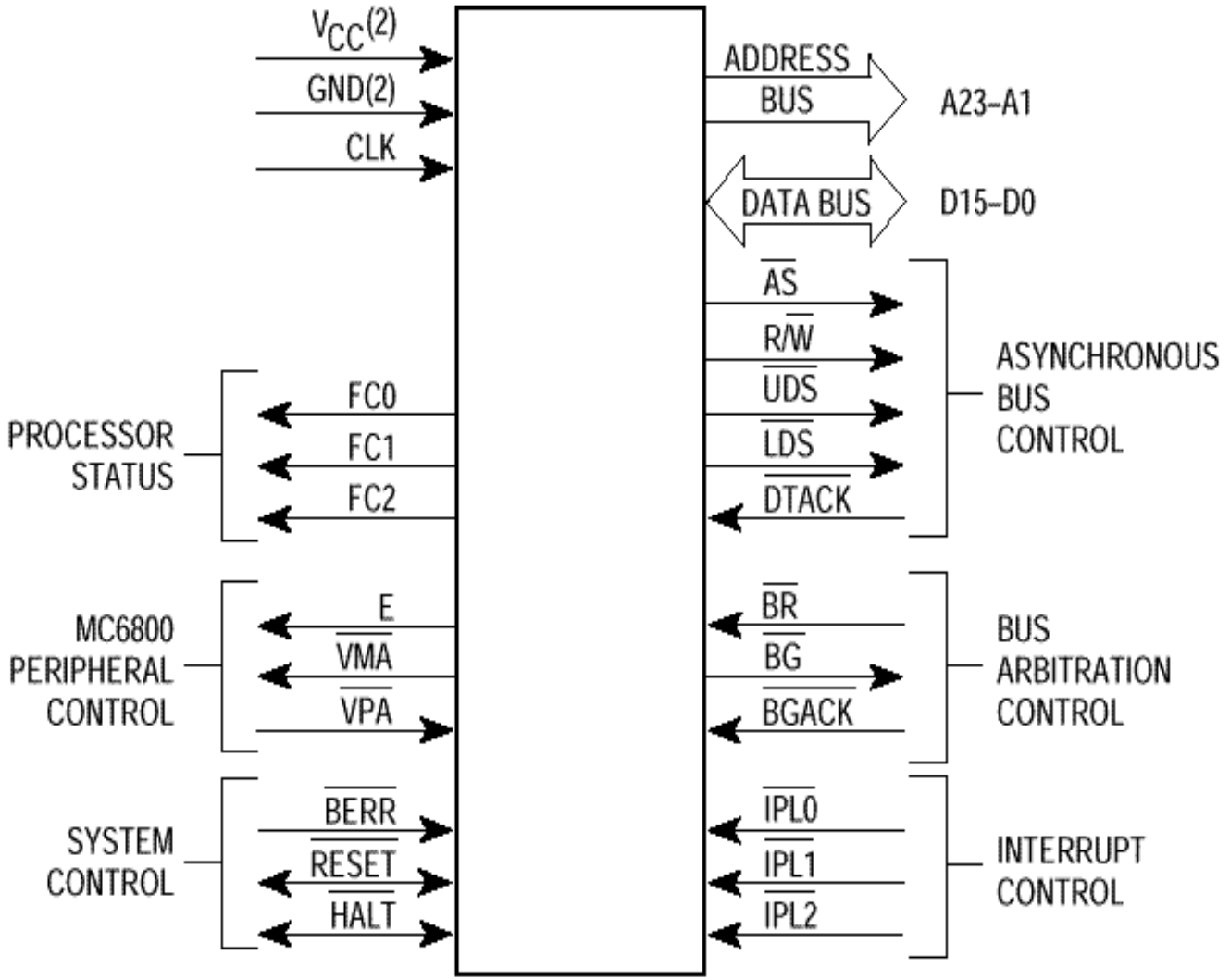
MOVE.{WL} (SP)+,DATO

La información extraída se almacena en dato

5. Buses del Sistema

- BUS DE DATOS (D15-D0) –
16bits. Triestado.
Bidireccional.

- BUS DE DIRECCIONES (A23-A1) - 23 bits = 8 M direcc. Físicas. Triestado. Unidireccional.



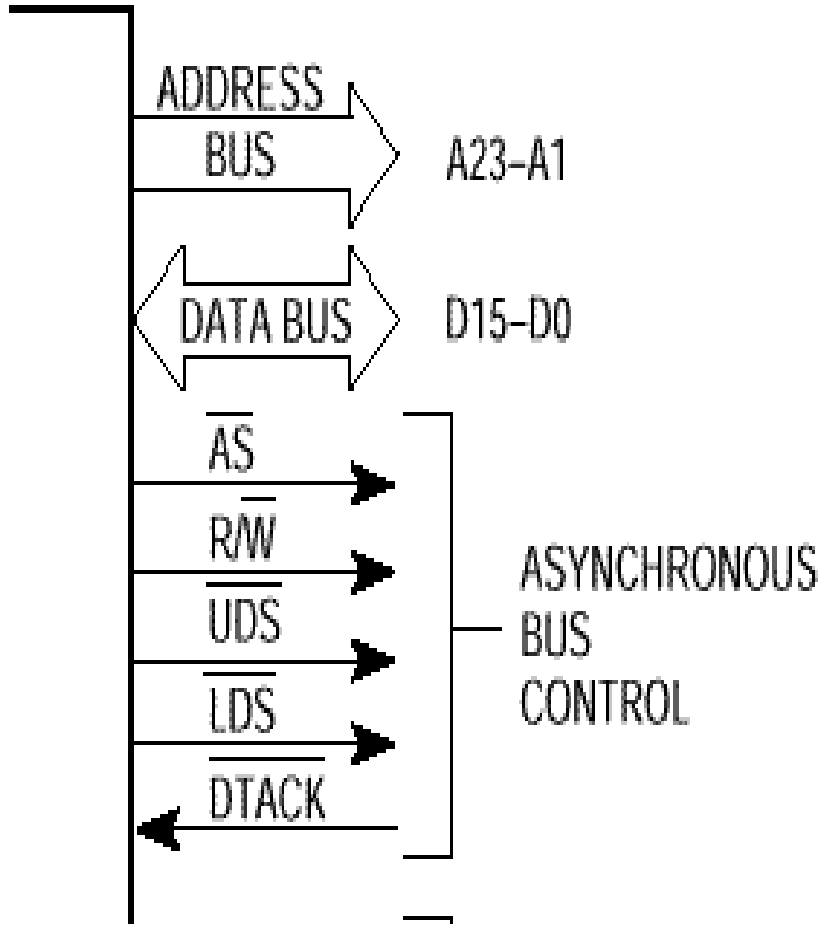
5. Buses del sistema: Control del bus asíncrono y estado del procesador

- #AS (Address Strobe): Salida. Triestado. Activa en bajo. Informa de que existe una dirección estable en el bus de direcciones.

- R/#W: Salida. Indica el tipo de transferencia (lectura o escritura)

- #UDS y #LDS (Up Data Strobe , Low Data Strobe). Salidas. Activas en bajo. Indican la parte del bus de datos (mitad superior D15-D8 o mitad inferior D7-D0) que se utiliza en los ciclos de lectura o escritura.

- #DTACK Data Transfer Acknowledge).. Entrada. Activa en bajo. Indica el fin de un ciclo de lectura/escritura.



5. Buses del sistema: Control del bus asíncrono y estado del procesador

$\overline{UDS}$	$\overline{LDS}$	R/$\overline{W}$	D8–D15	D0–D7
High	High	—	No Valid Data	No Valid Data
Low	Low	High	Valid Data Bits 15–8	Valid Data Bits 7–0
High	Low	High	No Valid Data	Valid Data Bits 7–0
Low	High	High	Valid Data Bus 15–8	No Valid Data
Low	Low	Low	Valid Data Bits 15–8	Valid Data Bits 7–0
High	Low	Low	Valid Data Bits 7–0*	Valid Data Bits 7–0
Low	High	Low	Valid Data Bits 15–8	Valid Data Bits 15–8*

- Accesos a bytes en direcciones impares utilizan D7-D0, por tanto, #UDS=1 y #LDS=0.
- Accesos a bytes en direcciones pares utilizan D15-D8, por tanto, #UDS=0 y #LDS=1.
- Accesos a words, utilizan D15-D0, por tanto, #UDS=#LDS=0.
- Accesos a long words, requieren dos ciclos de words.

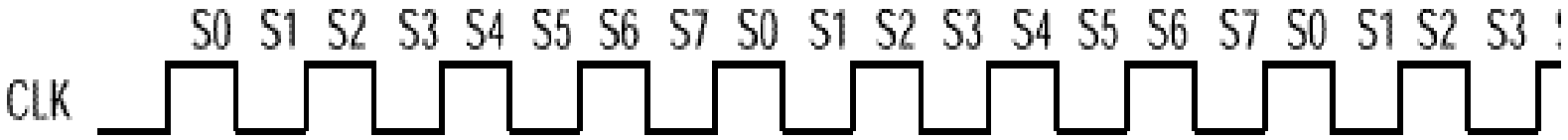
5. Buses del sistema: Control del bus asíncrono y estado del procesador

- Estado del procesador: F2,FC1,FC0. Salidas. Informan del tipo de ciclo de lectura/escritura que está realizando el microprocesador.
- Ciclo Supervisor/Usuario.
- Ciclo de Programa. Si accede a memoria en la fase de FETCH.
- Ciclo de Datos. Si accede a memoria en la fase de EXECUTE.
- Reconocimiento de interrupciones.

0	**0**	**0**	**Indefinido**
0	**0**	**1**	**Datos de usuario**
0	**1**	**0**	**Programa de usuario**
0	**1**	**1**	**Indefinido**
1	**0**	**0**	**Indefinido**
1	**0**	**1**	**Datos de supervisor**
1	**1**	**0**	**Programa de supervisor**
1	**1**	**1**	**Reconocimiento de interrupciones**

5. Buses del sistema: Control del bus asíncrono y estado del procesador

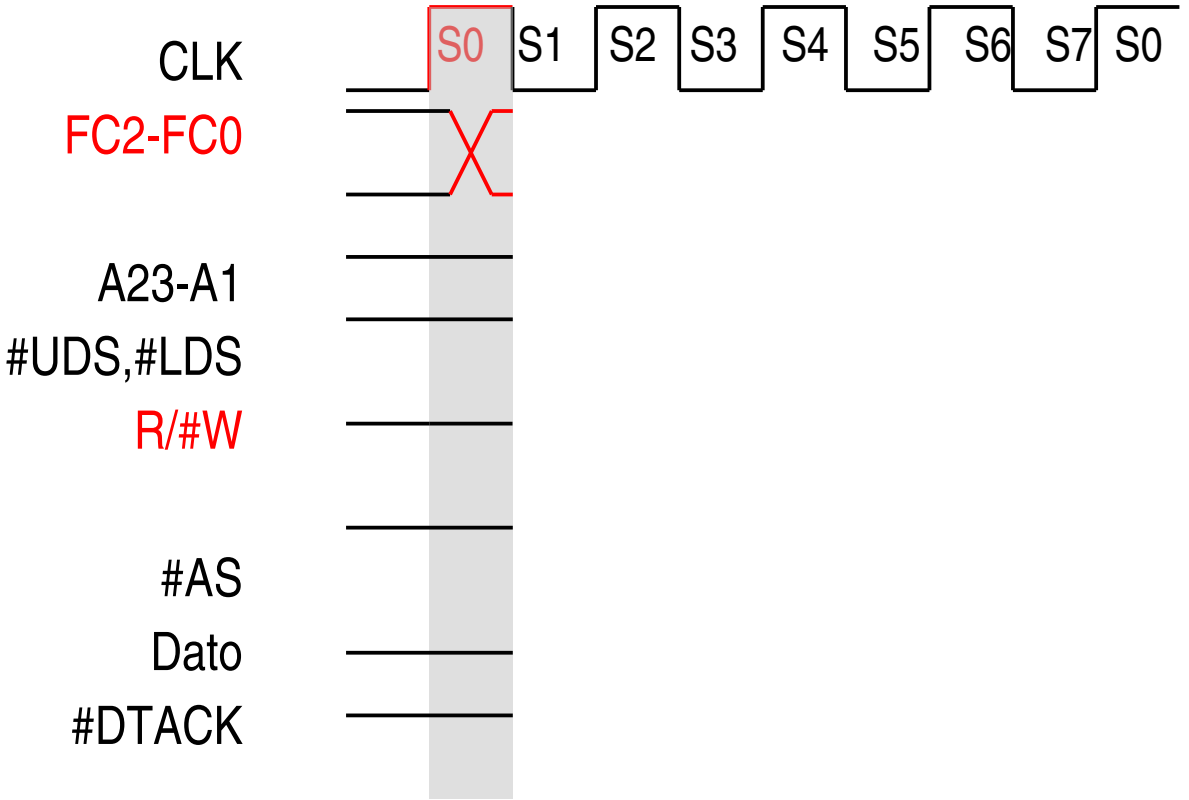
- Existen tres ciclos de bus:
  - Ciclo de LECTURA
  - Ciclo de ESCRITURA
  - Ciclo de LECTURA-MODIFICACION-ESCRITURA *
- Ciclo de LECTURA
  - Durante un ciclo de lectura, el procesador recibe 1 o 2 bytes de datos por el bus
  - #UDS y #LDS definen que mitad del bus se está utilizando.
  - Operandos de tamaño palabra larga se requieren dos ciclos de lectura
  - En un ciclo de lectura se definen ocho estados. Cada estado se corresponde con un nivel lógico de la señal de reloj CLK.



5. Buses del sistema: Control del bus asíncrono y estado del procesador

Temporización de lectura (sin estados de espera): Estado S0

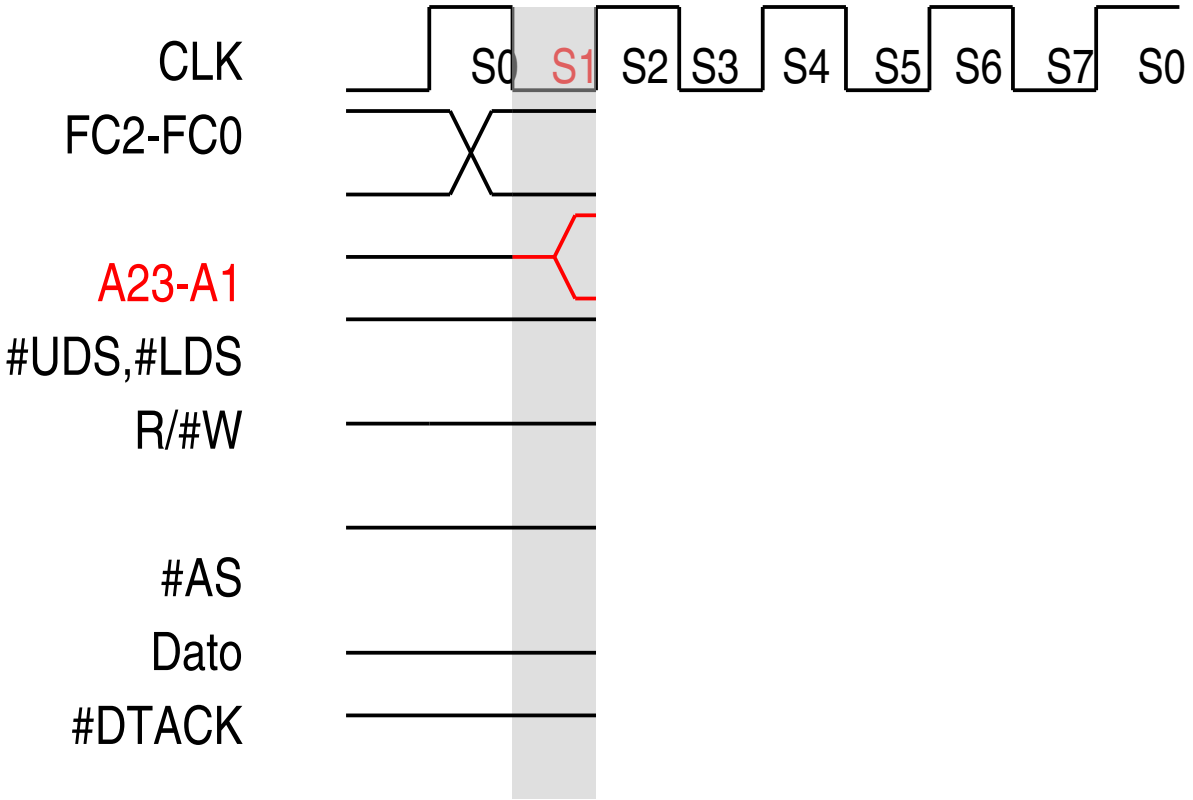
- Se actualizan FC2:FC0 y se pone a 1 la señal R/W#



5. Buses del sistema: Control del bus asíncrono y estado del procesador

Temporización de lectura (sin estados de espera): Estado S1

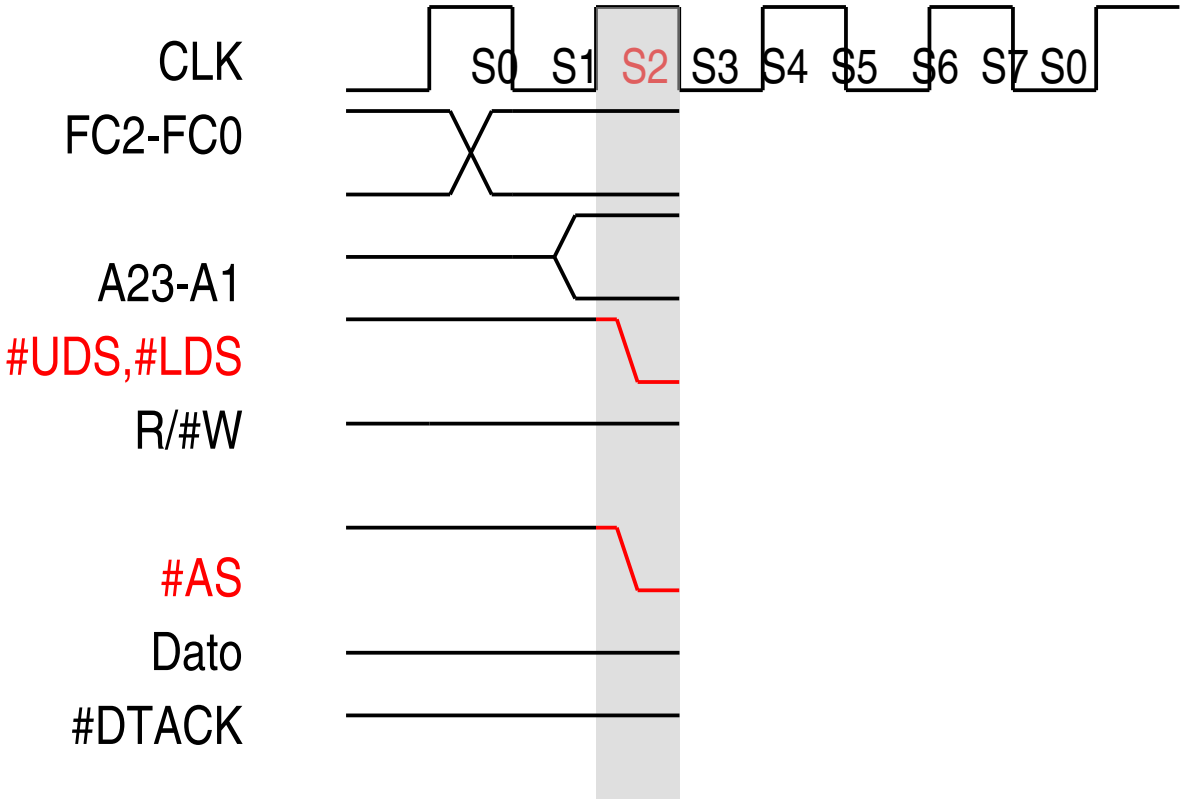
- Se colocan la dirección en el bus A23-A1.



5. Buses del sistema: Control del bus asíncrono y estado del procesador

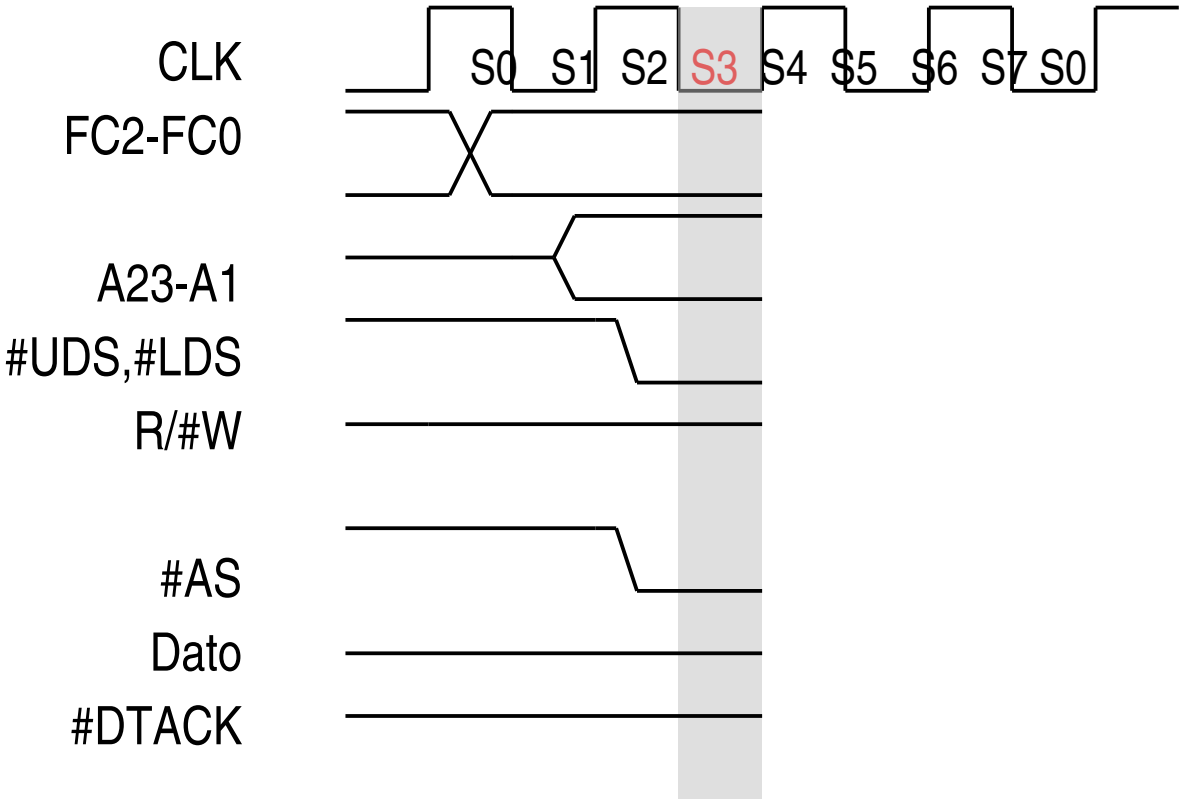
Temporización de lectura (sin estados de espera): Estado S2

- Se activan #AS y #UDS,#LDS según corresponda.



5. Buses del sistema: Control del bus asíncrono y estado del procesador

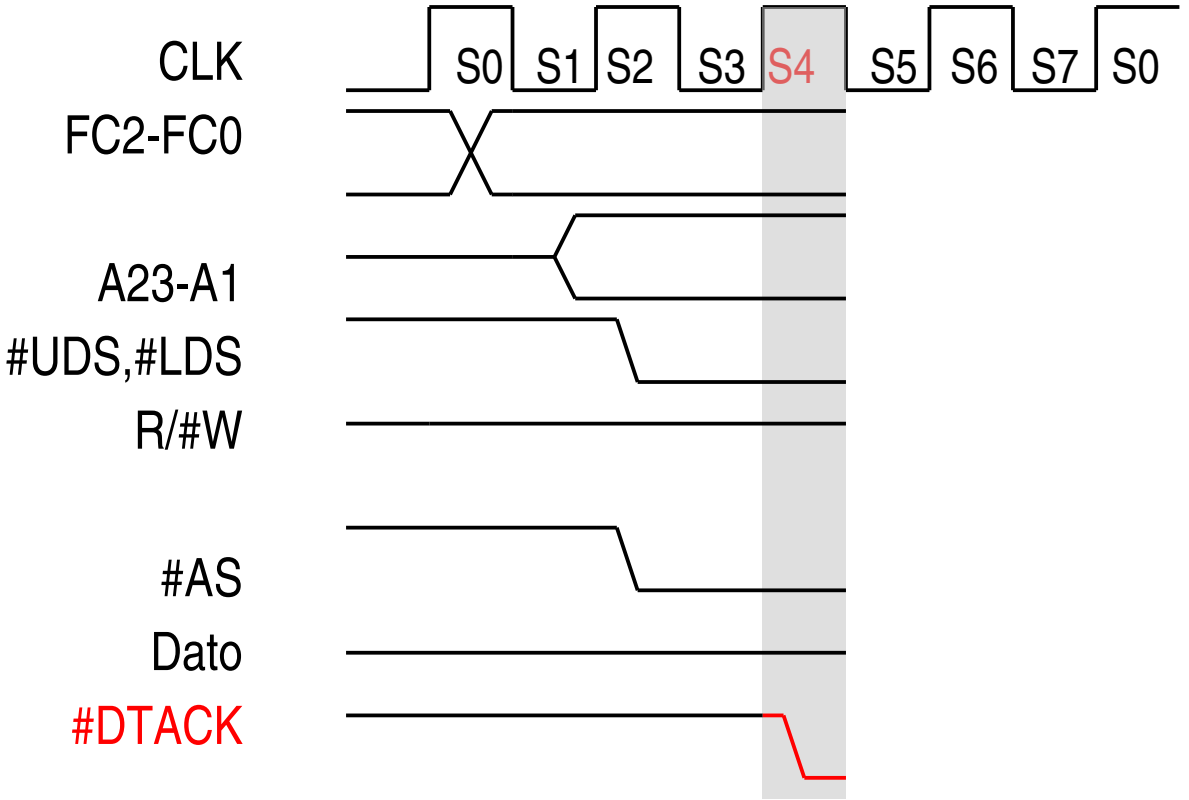
Temporización de lectura (sin estados de espera): Estado S3



5. Buses del sistema: Control del bus asíncrono y estado del procesador

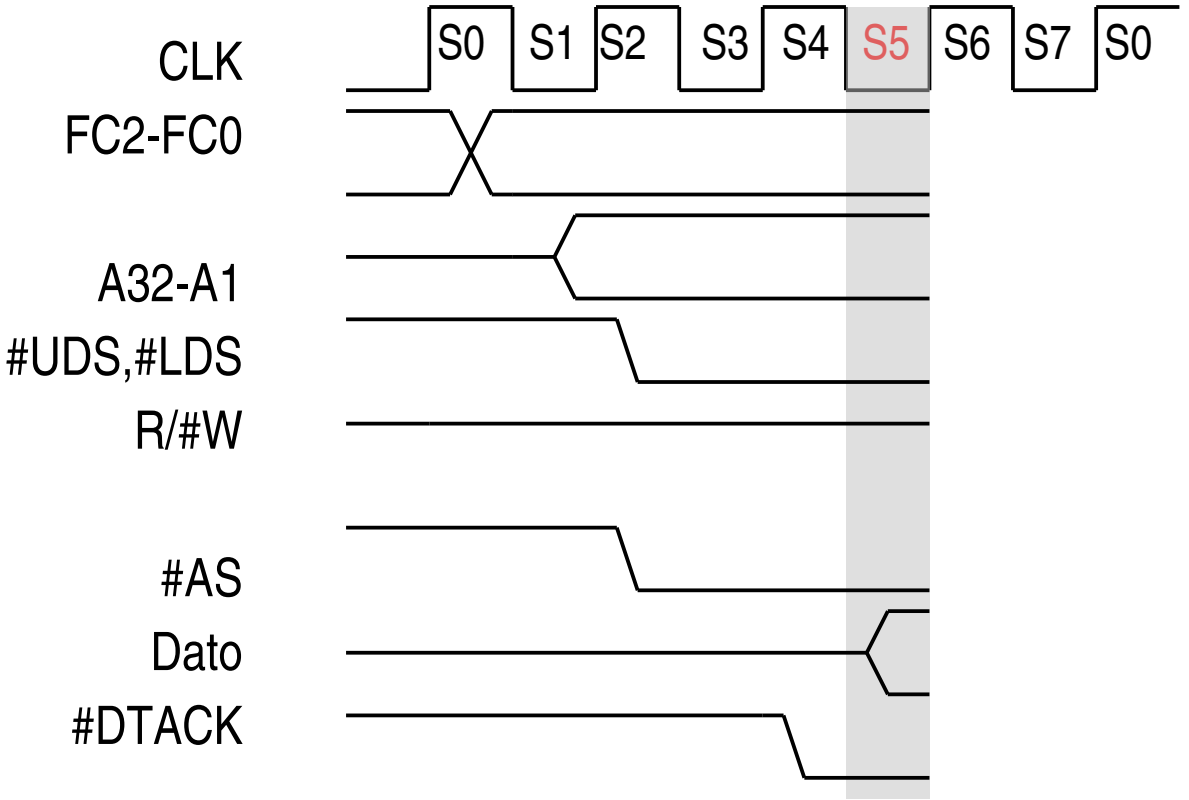
Temporización de lectura (sin estados de espera): Estado S4

- Si se recibe la activación de DTACK, se pasa al estado S5 y no se insertan estados de espera.



5. Buses del sistema: Control del bus asíncrono y estado del procesador

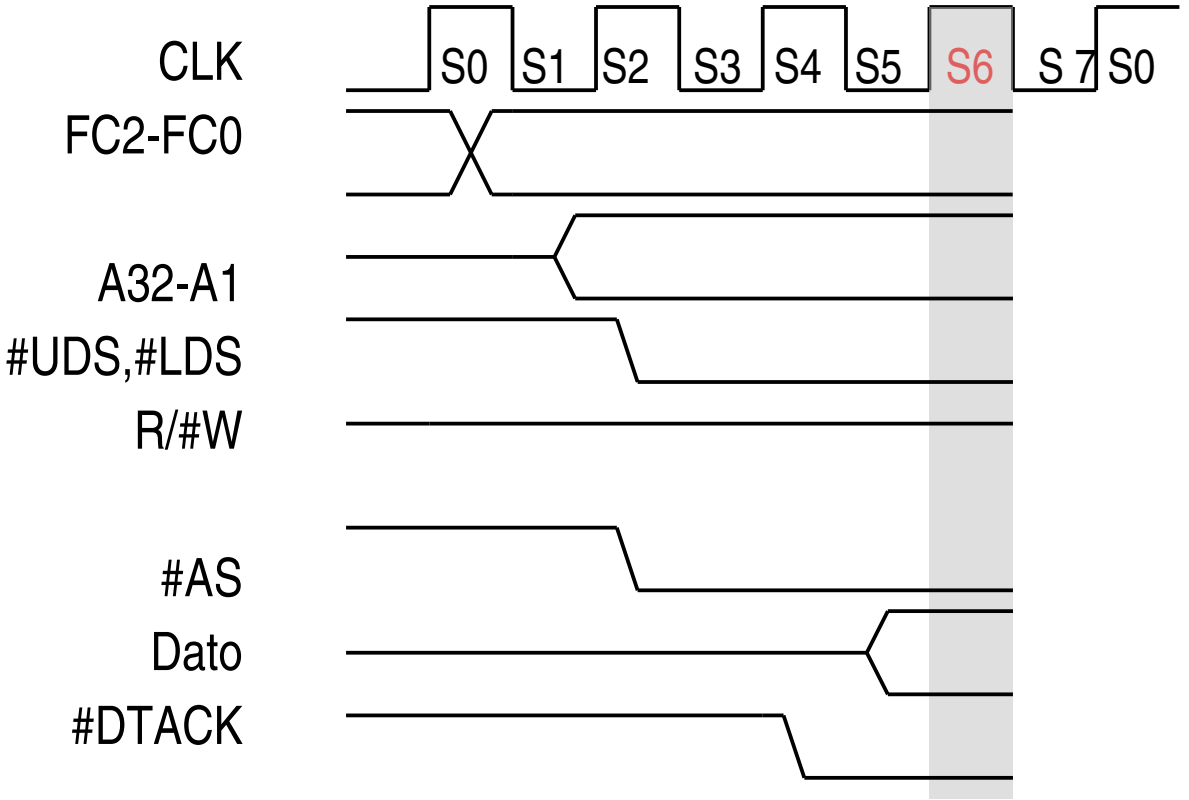
Temporización de lectura (sin estados de espera): Estado S5



5. Buses del sistema: Control del bus asíncrono y estado del procesador

Temporización de lectura (sin estados de espera): Estado S6

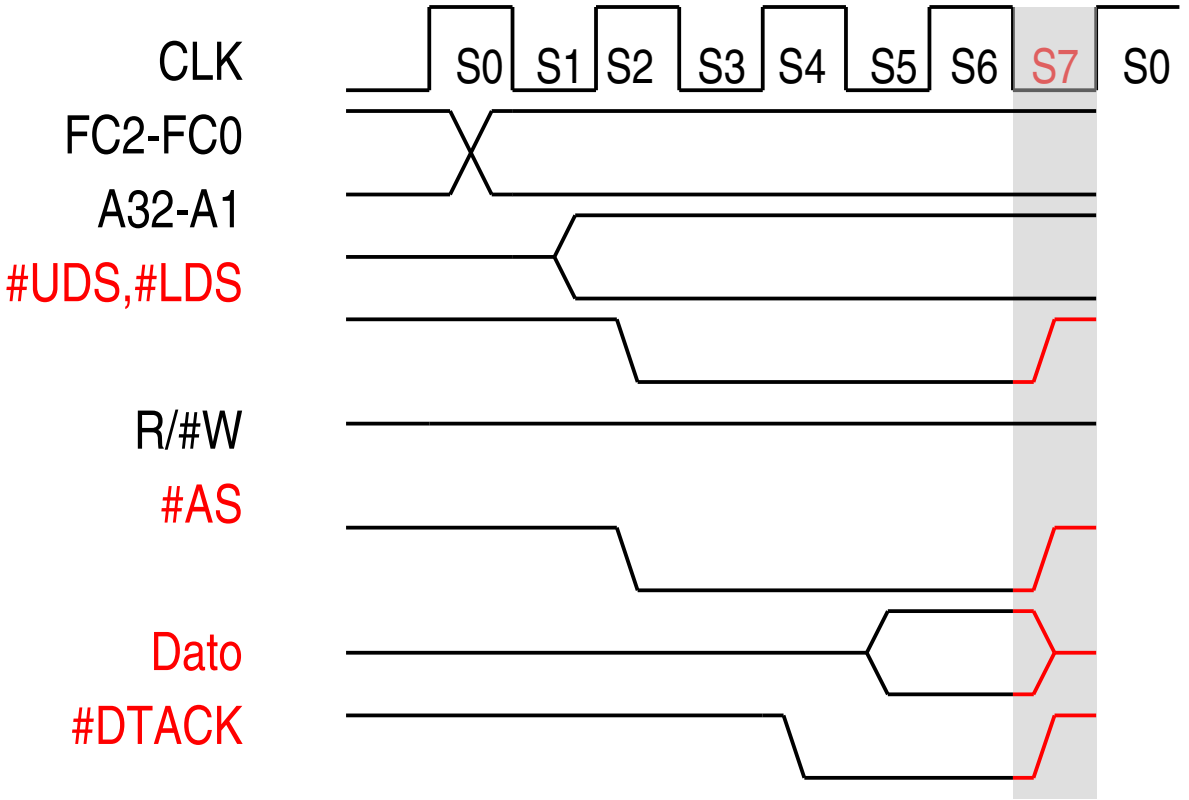
- Se captura el dato al final del ciclo (flanco descendente).



5. Buses del sistema: Control del bus asíncrono y estado del procesador

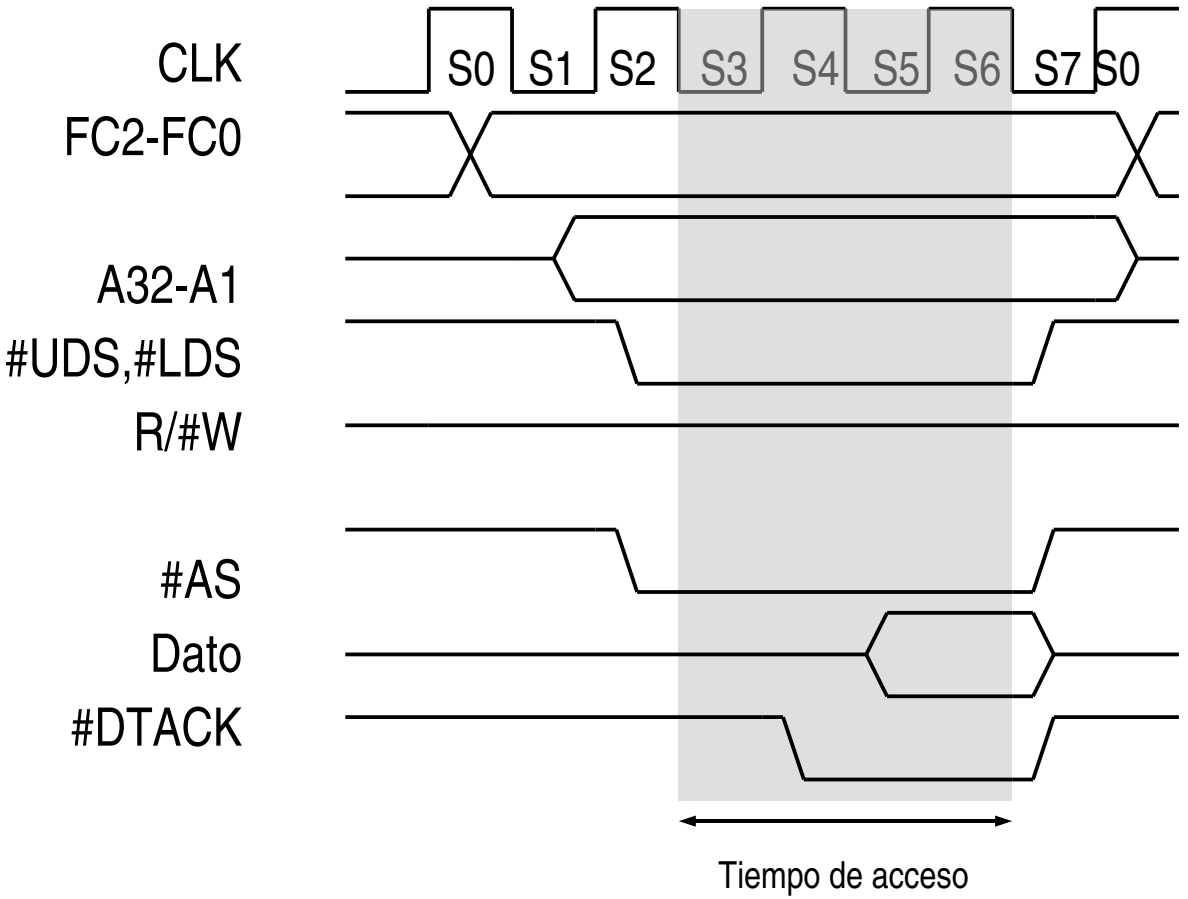
Temporización de lectura (sin estados de espera): Estado S7

- Se desactivan #UDS,#LDS,#AS, junto con #DATCK. El dato es retirado del bus.



5. Buses del sistema: Control del bus asíncrono y estado del procesador

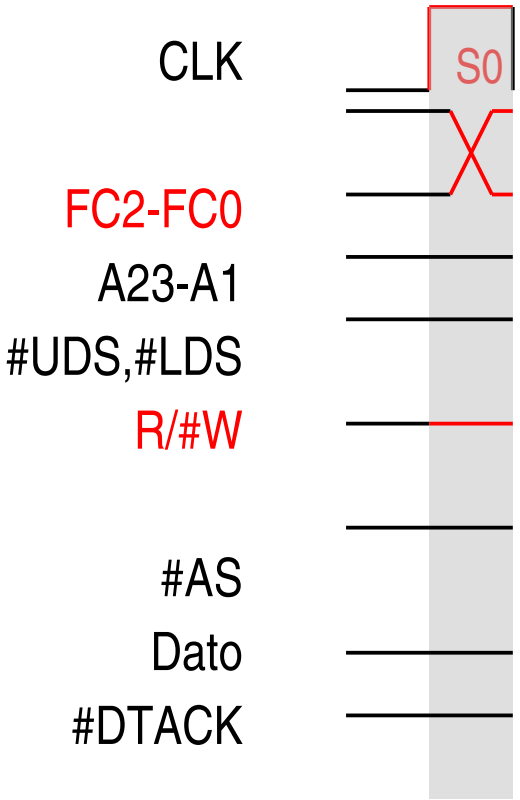
Temporización de lectura (sin estados de espera):



5. Buses del sistema: Control del bus asíncrono y estado del procesador

Temporización de lectura (con estados de espera): Estado S0

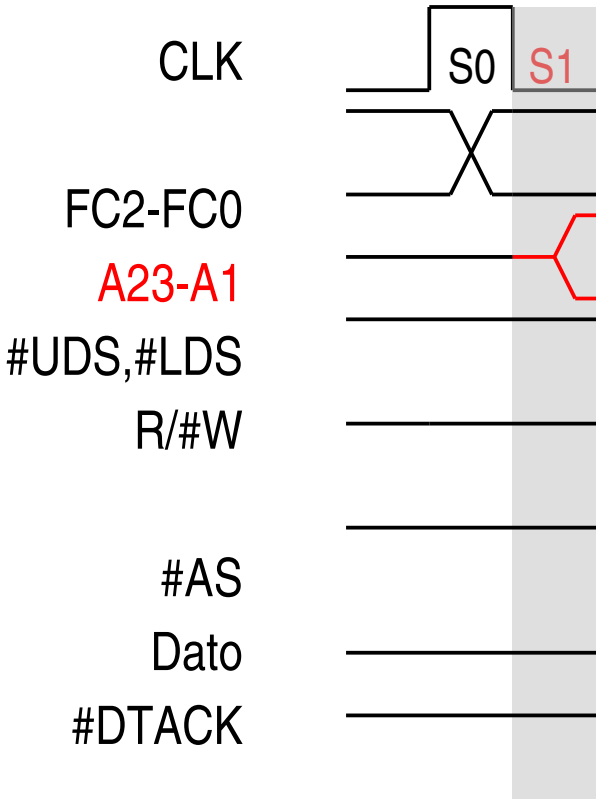
- Se actualizan FC2:FC0 y se pone a 1 la señal R/W#



5. Buses del sistema: Control del bus asíncrono y estado del procesador

Temporización de lectura (con estados de espera): Estado S1

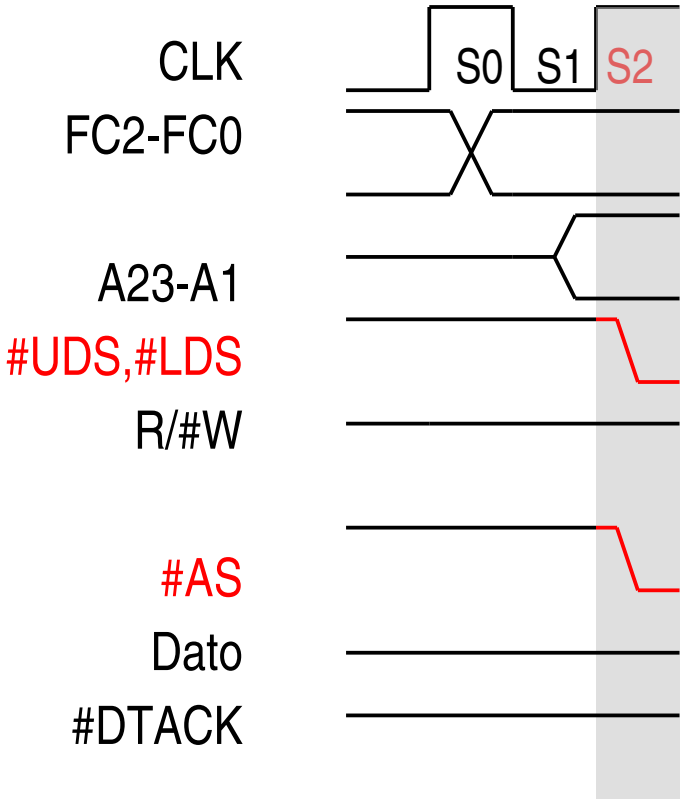
- Se colocan la dirección en el bus A23-A1.



5. Buses del sistema: Control del bus asíncrono y estado del procesador

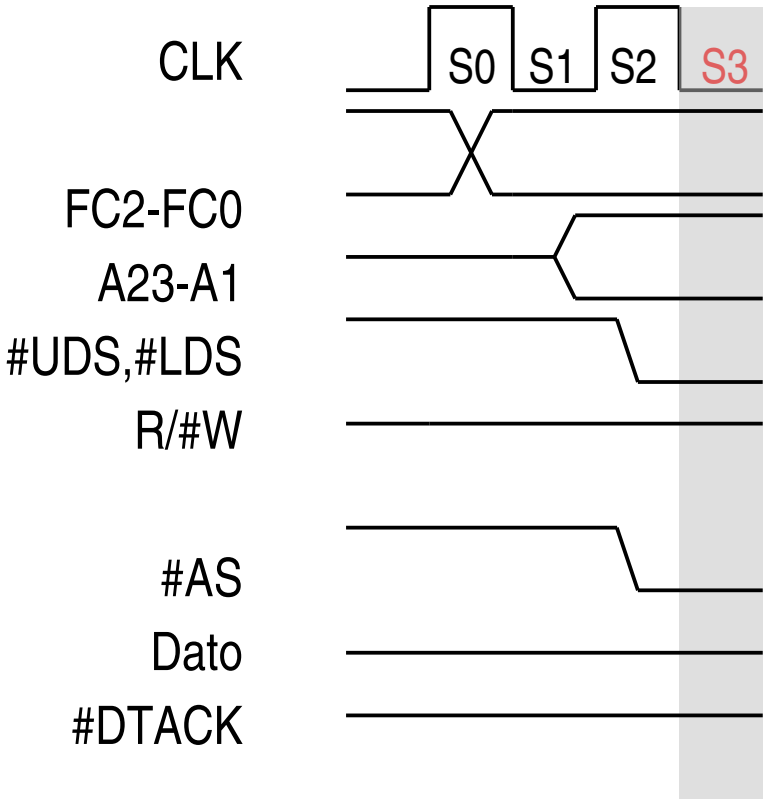
Temporización de lectura (con estados de espera): Estado S2

- Se activan #AS y #UDS,#LDS según corresponda.



5. Buses del sistema: Control del bus asíncrono y estado del procesador

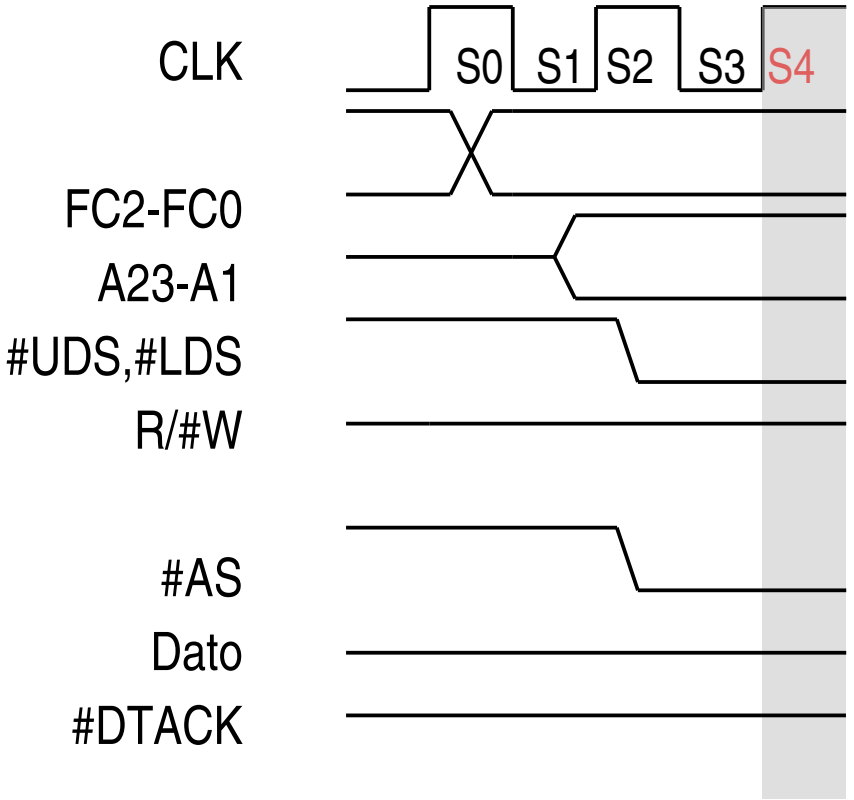
Temporización de lectura (con estados de espera): Estado S3



5. Buses del sistema: Control del bus asíncrono y estado del procesador

Temporización de lectura (sin estados de espera): Estado S4

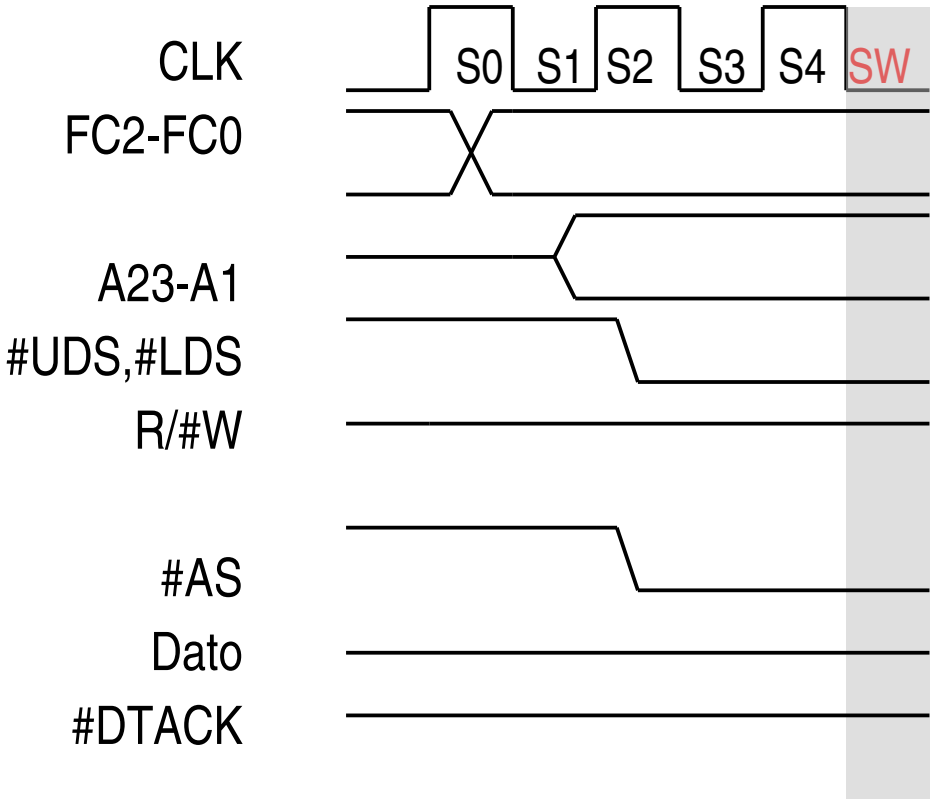
- No se recibe la activación de DTACK, se insertan estados de espera.



5. Buses del sistema: Control del bus asíncrono y estado del procesador

Temporización de lectura (con estados de espera): Estado SW

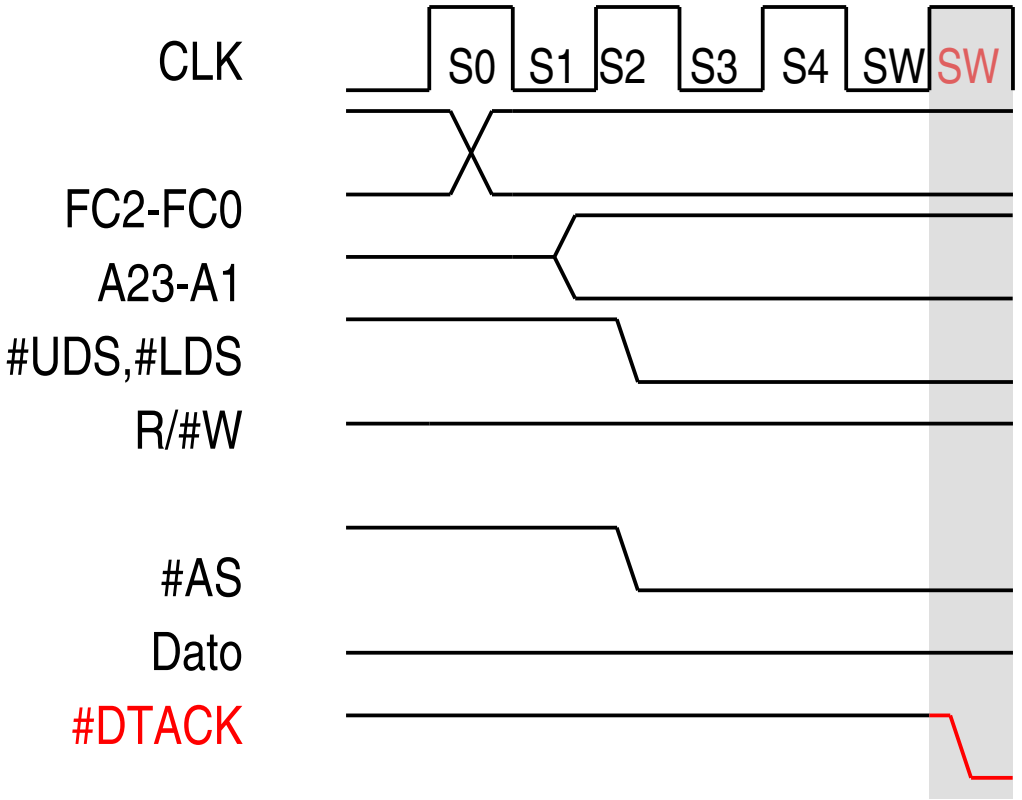
- Si espera a la activación de DTACK



5. Buses del sistema: Control del bus asíncrono y estado del procesador

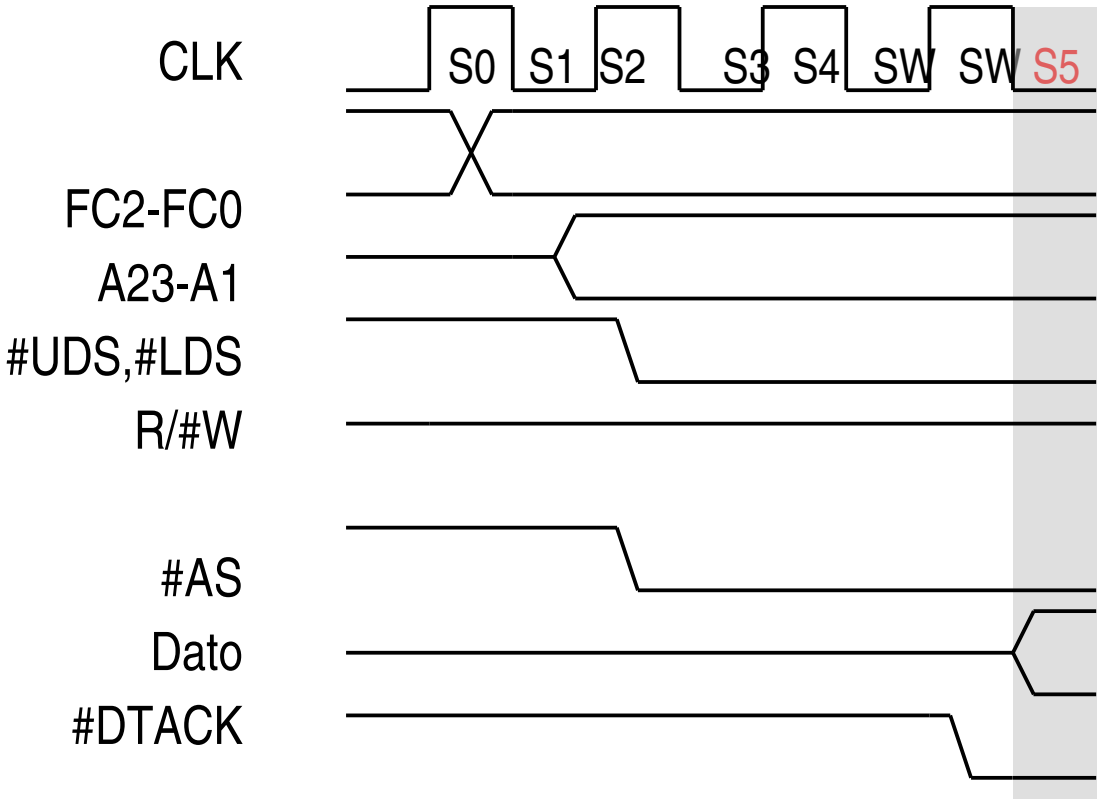
Temporización de lectura (con estados de espera): Estado SW

- Si espera a la activación de DTACK



5. Buses del sistema: Control del bus asíncrono y estado del procesador

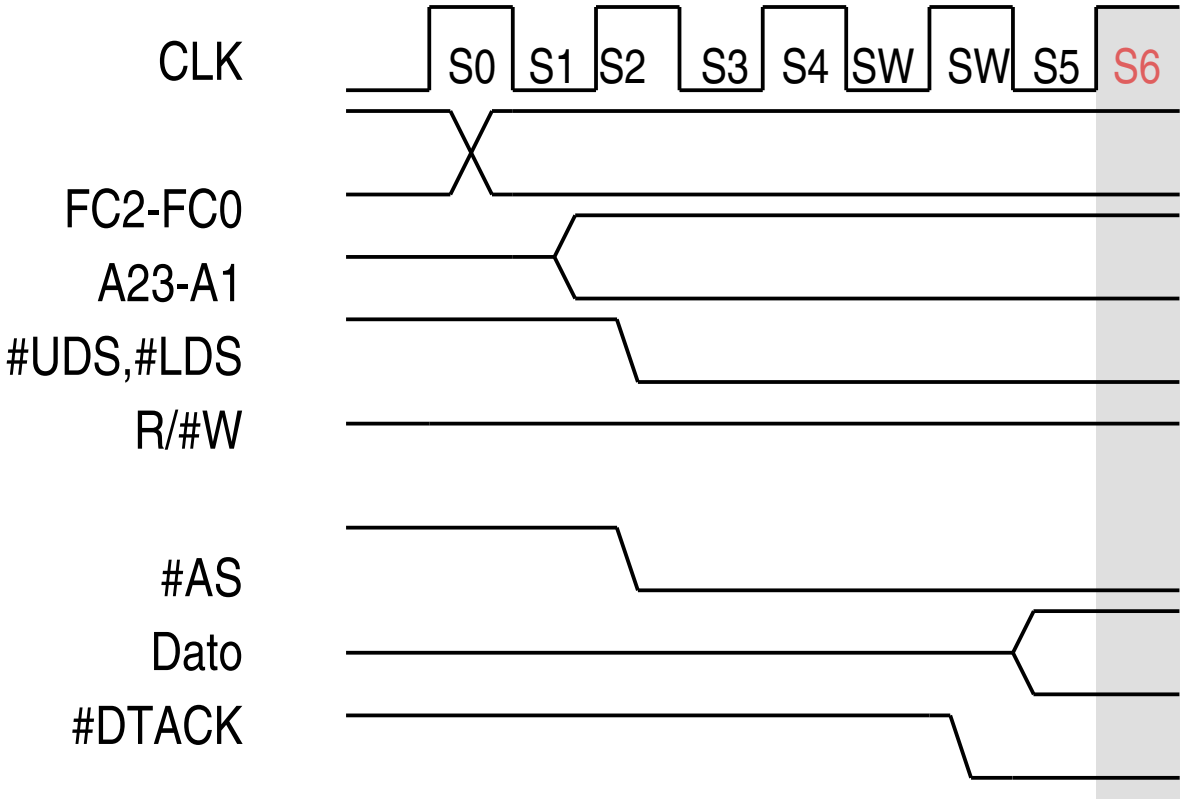
Temporización de lectura (con estados de espera): Estado S5



5. Buses del sistema: Control del bus asíncrono y estado del procesador

Temporización de lectura (con estados de espera): Estado S6

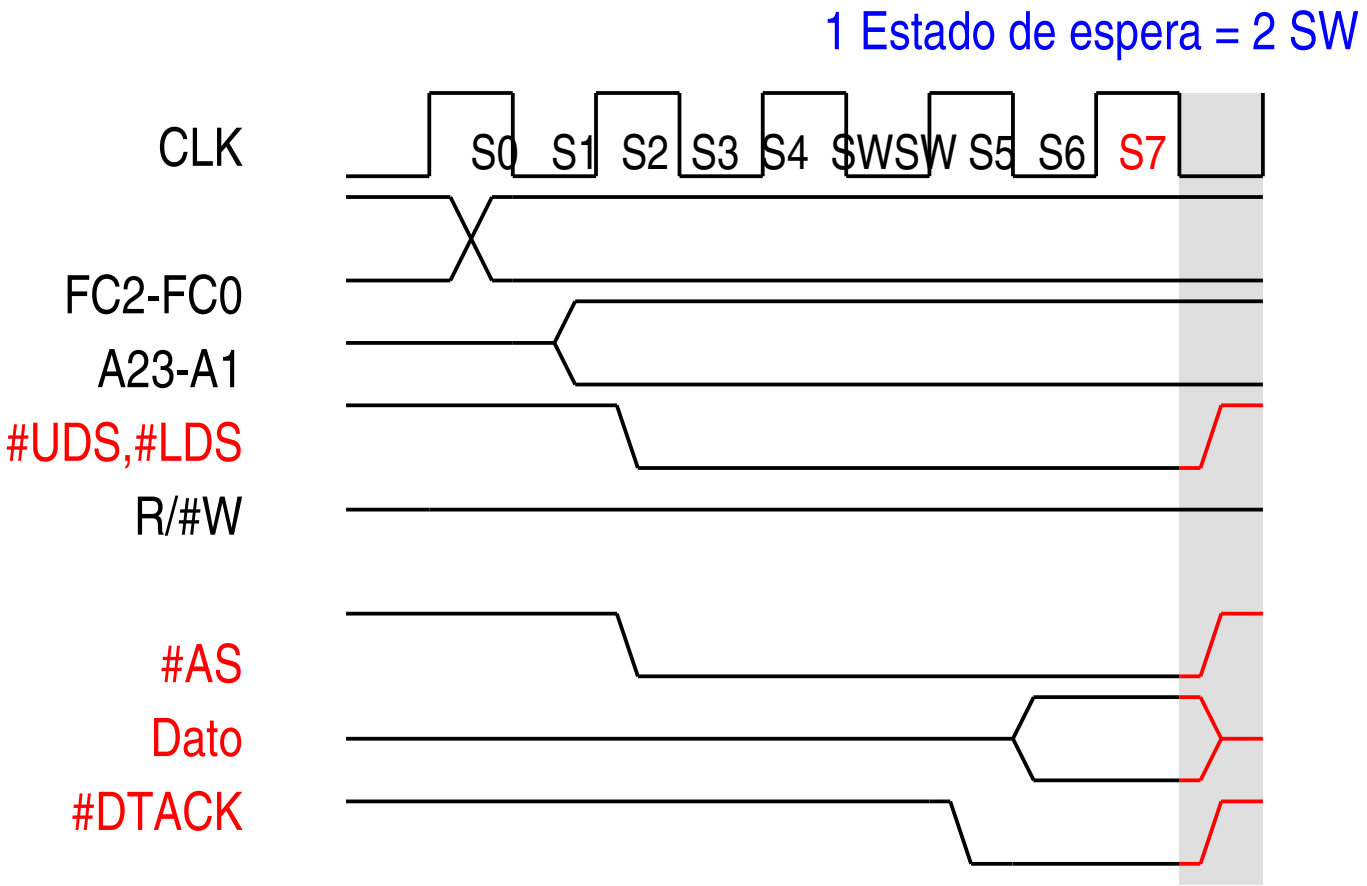
- Se captura el dato en el flanco descendente.



5. Buses del sistema: Control del bus asíncrono y estado del procesador

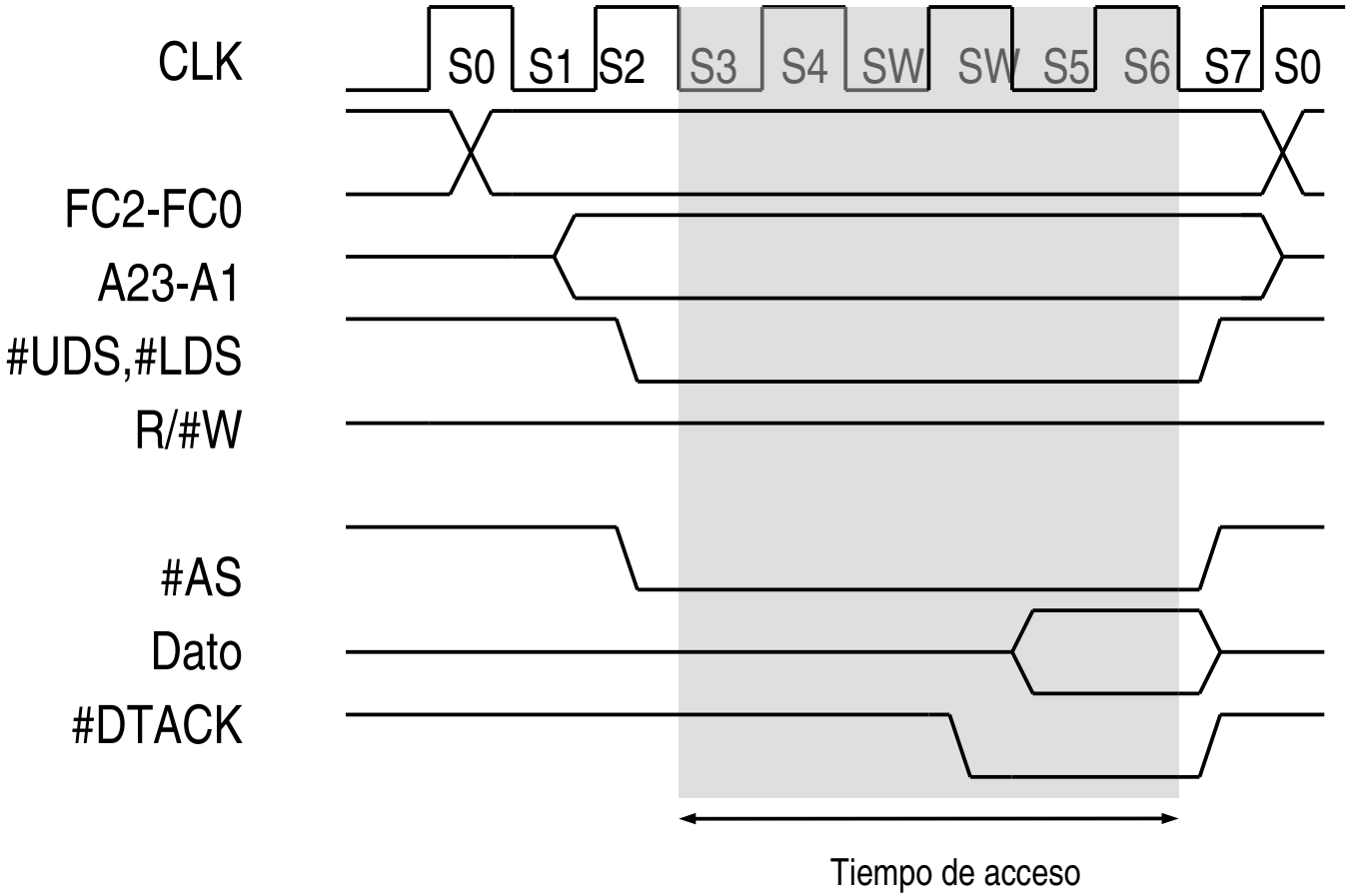
Temporización de lectura (con estados de espera): Estado S7

- Se desactivan #UDS,#LDS,#AS, junto con #DATCK. El dato es retirado del bus.



5. Buses del sistema: Control del bus asíncrono y estado del procesador

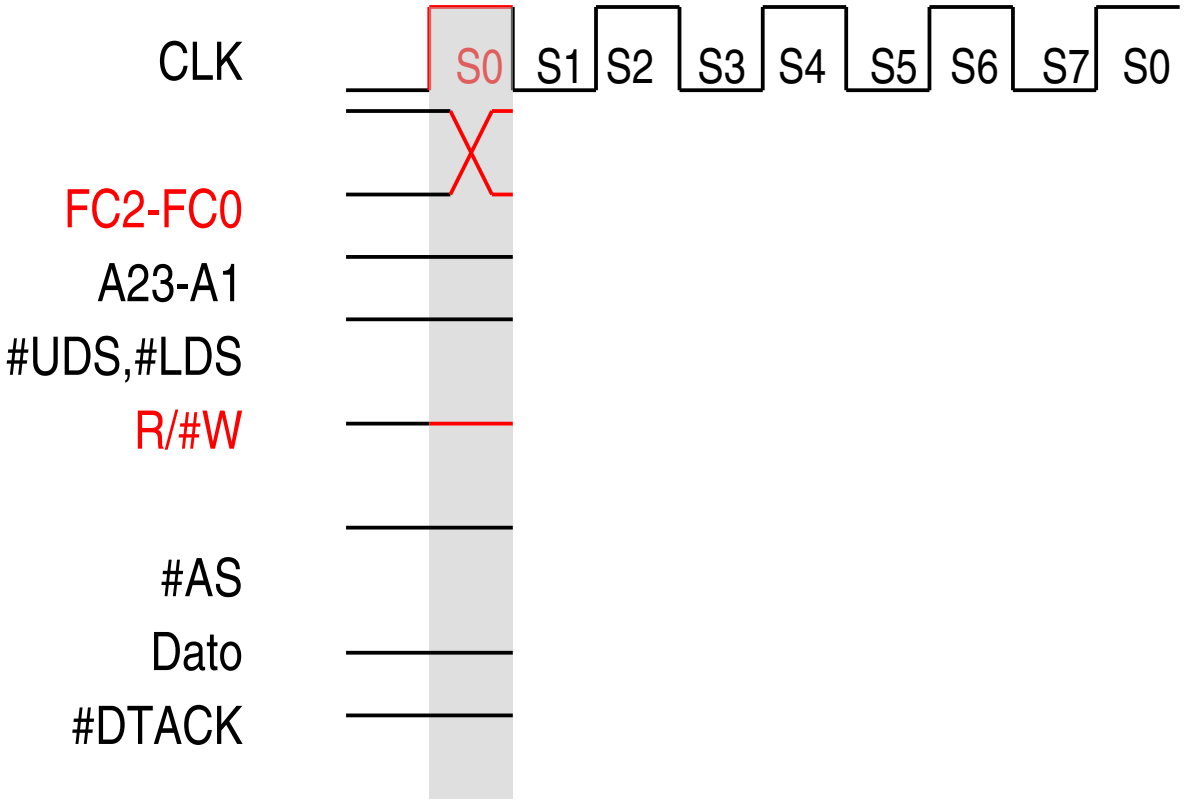
Temporización de lectura (con estados de espera):



5. Buses del sistema: Control del bus asíncrono y estado del procesador

Temporización de escritura (sin estados de espera): Estado S0

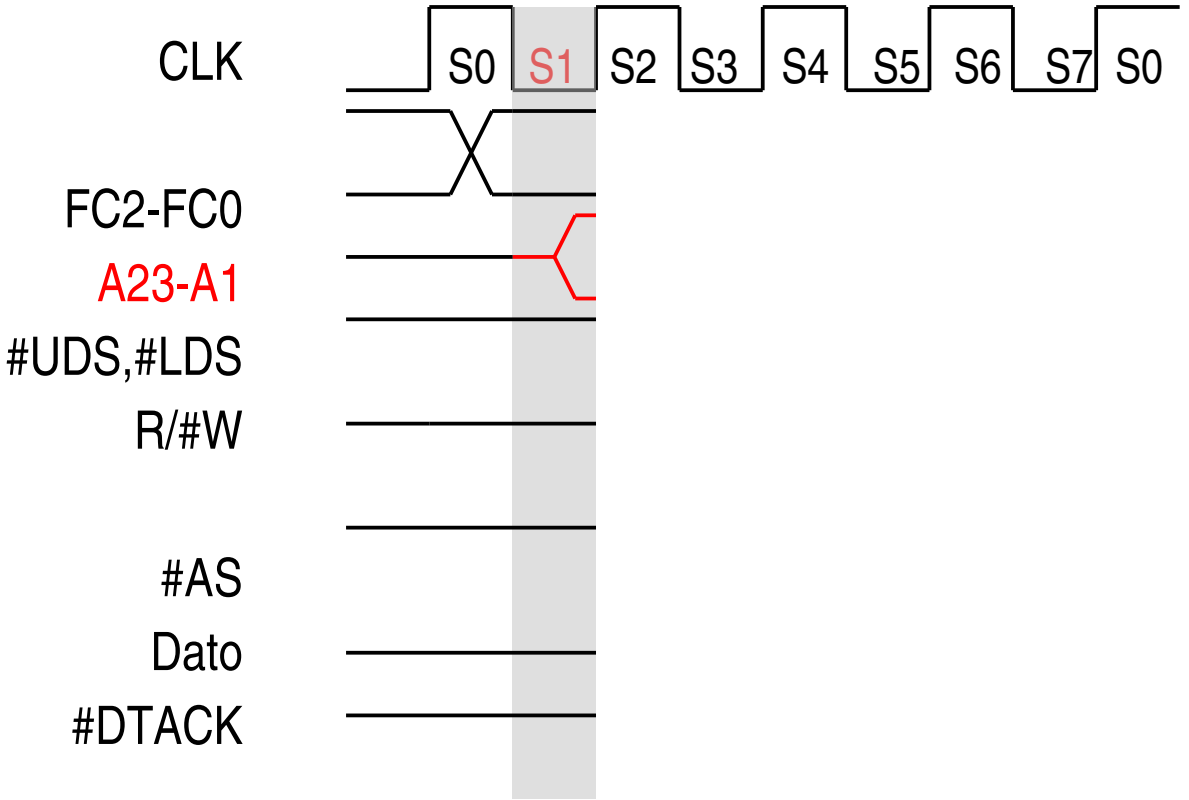
- Se actualizan FC2:FC0 y se pone a 1 la señal R/W#



5. Buses del sistema: Control del bus asíncrono y estado del procesador

Temporización de escritura (sin estados de espera): Estado S1

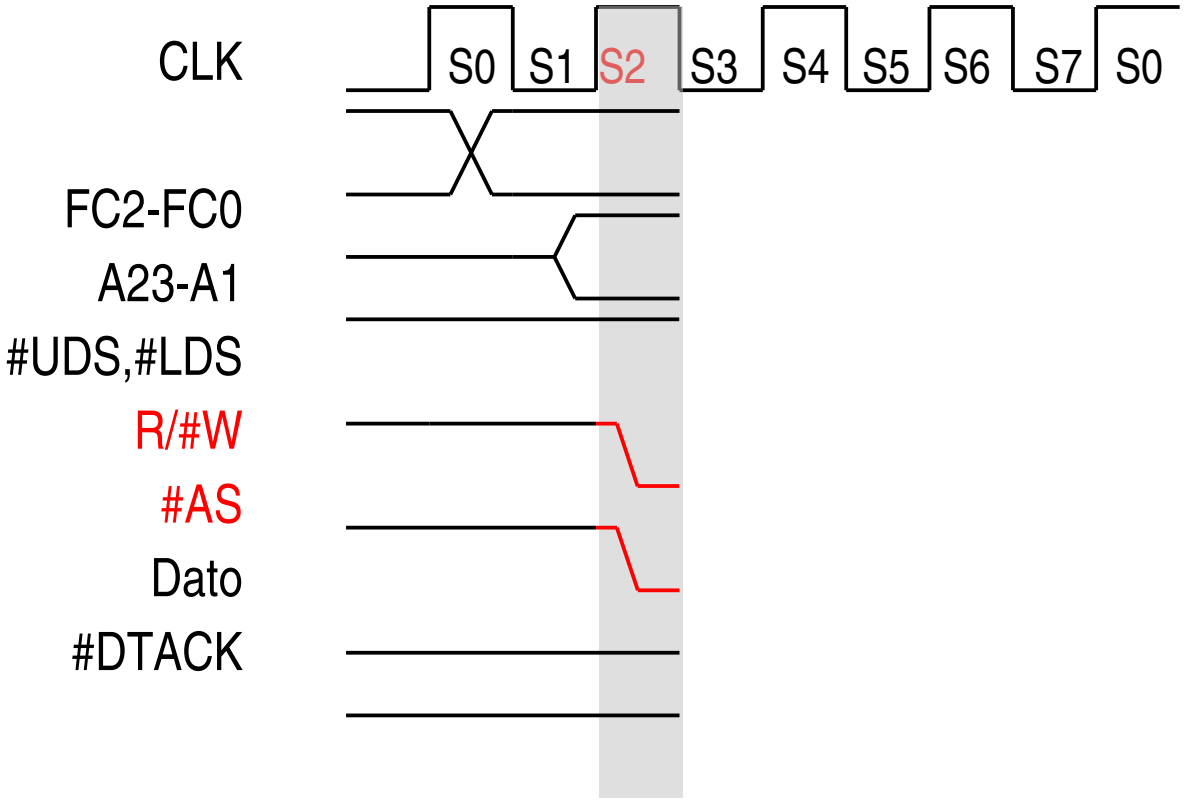
- Se colocan la dirección en el bus A23-A1.



5. Buses del sistema: Control del bus asíncrono y estado del procesador

Temporización de escritura (sin estados de espera): Estado S2

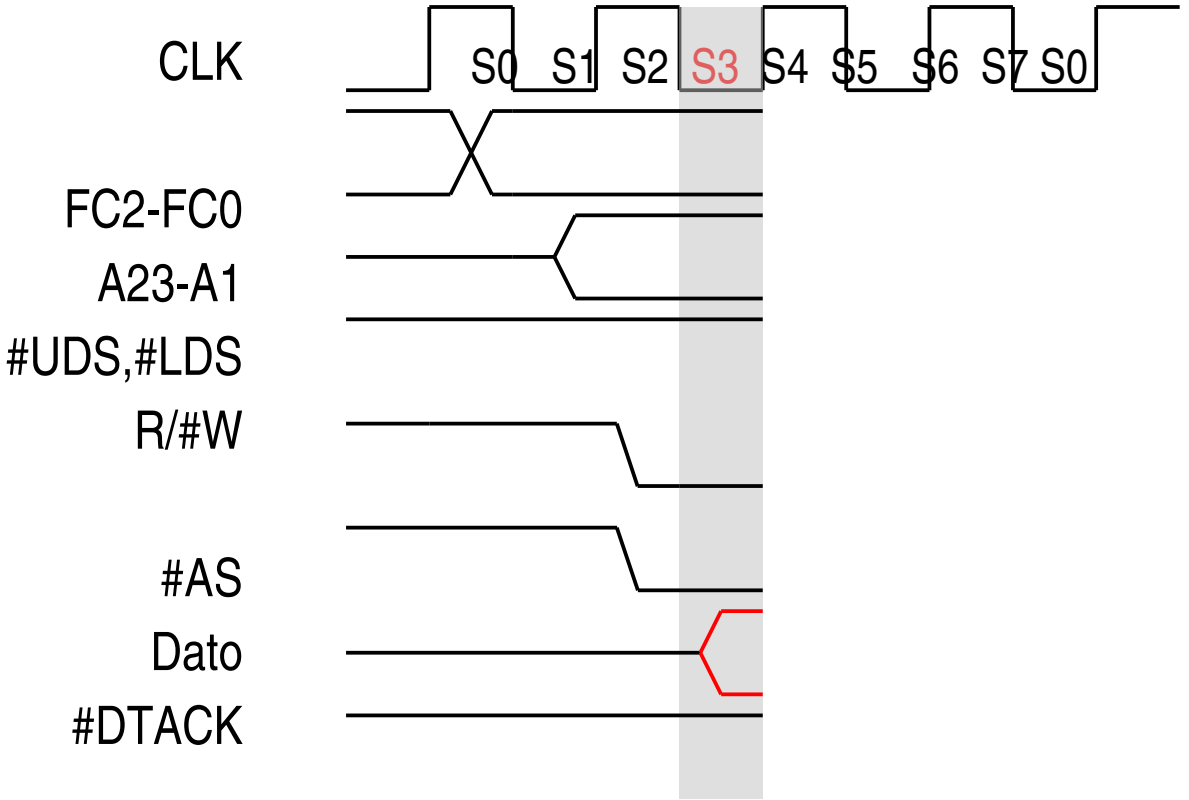
- Se activan #AS y se pone a 0 R/#W



5. Buses del sistema: Control del bus asíncrono y estado del procesador

Temporización de escritura (sin estados de espera): Estado S3

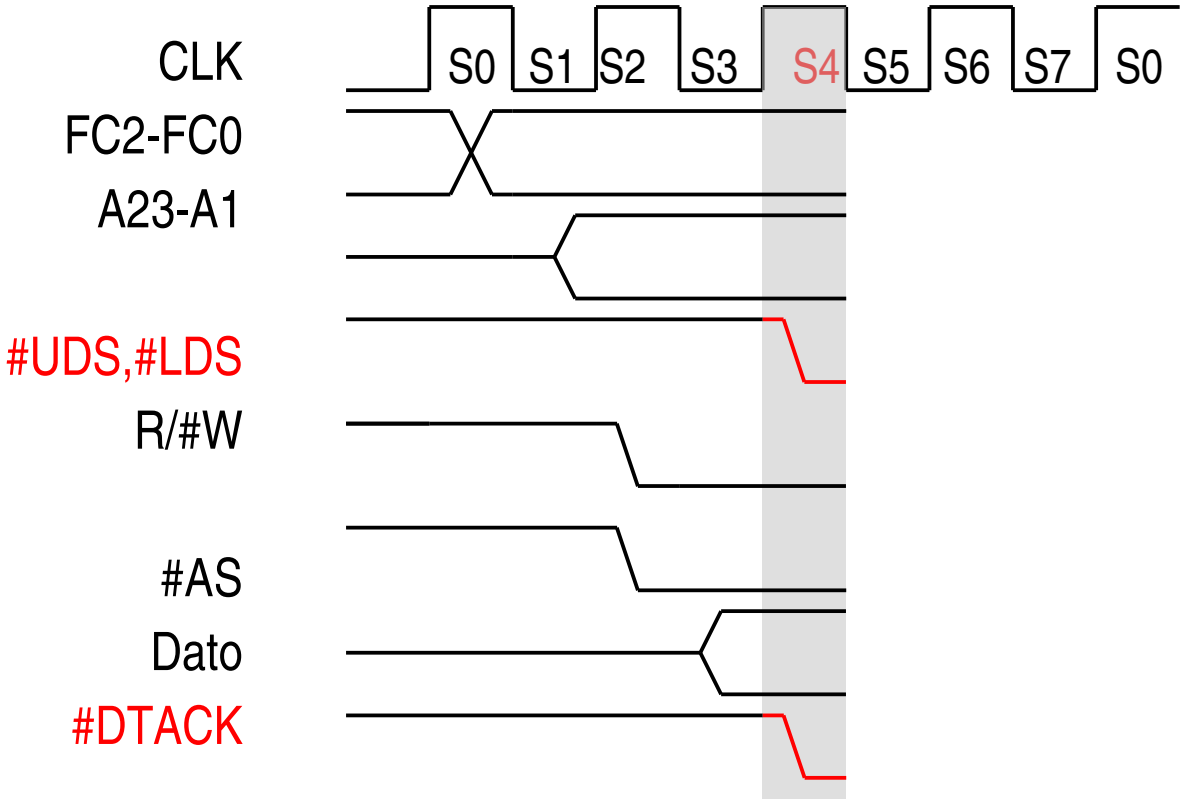
- Se sitúa el dato en el bus de datos.



5. Buses del sistema: Control del bus asíncrono y estado del procesador

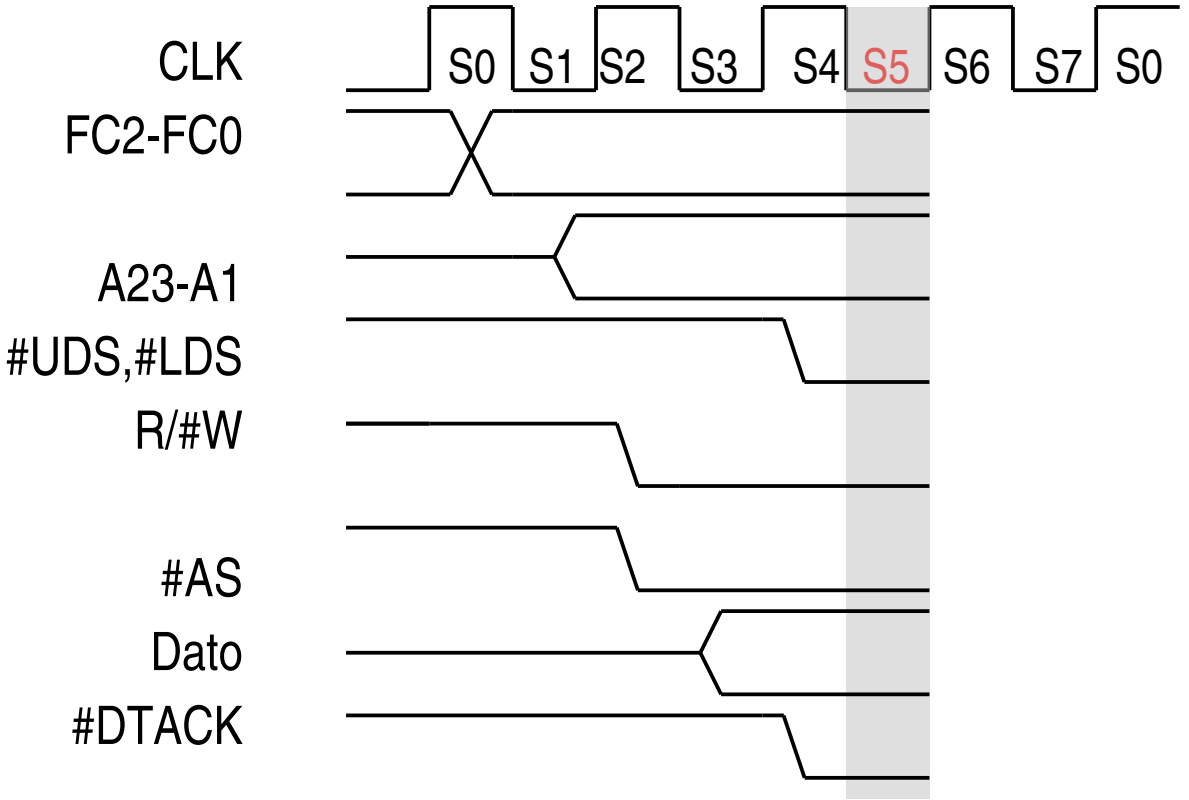
Temporización de escritura (sin estados de espera): Estado S4

- Se activan #UDS y #LDS según corresponda y se recibe la activación de DTACK, se pasa al estado S5 y no se insertan estados de espera.



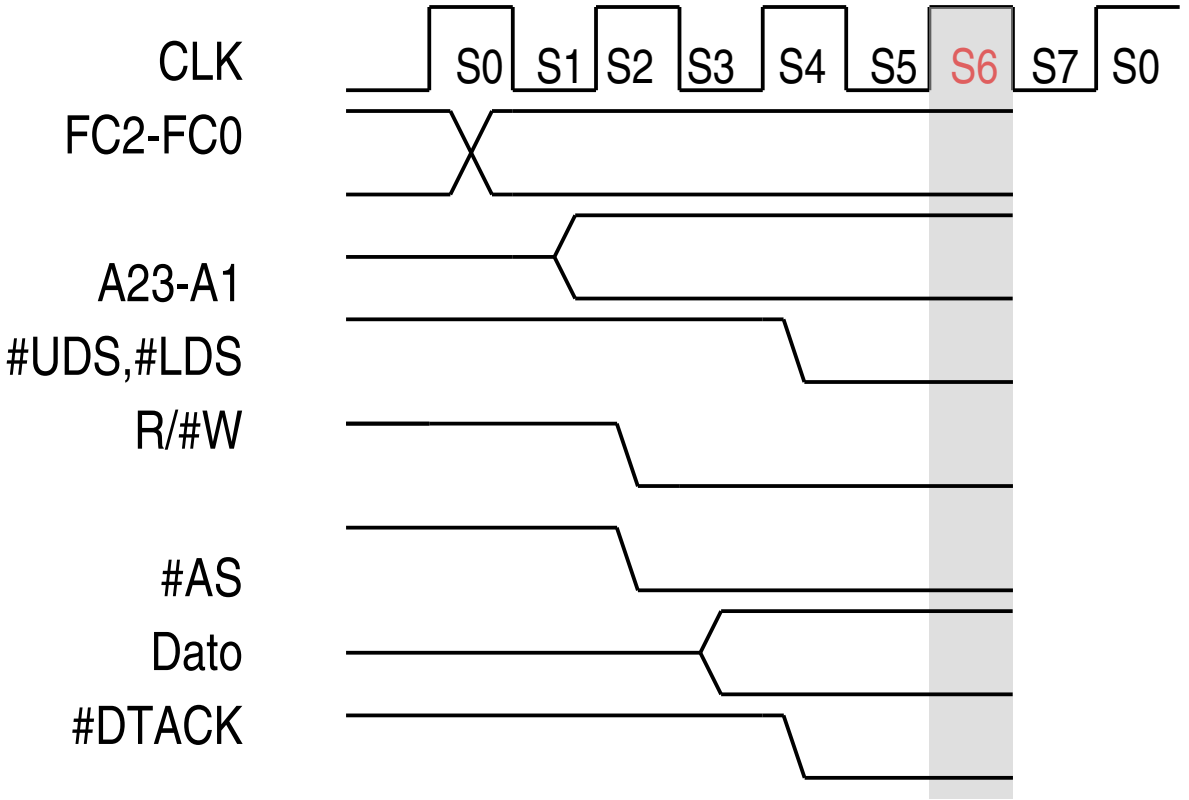
5. Buses del sistema: Control del bus asíncrono y estado del procesador

Temporización de escritura (sin estados de espera): Estado S5



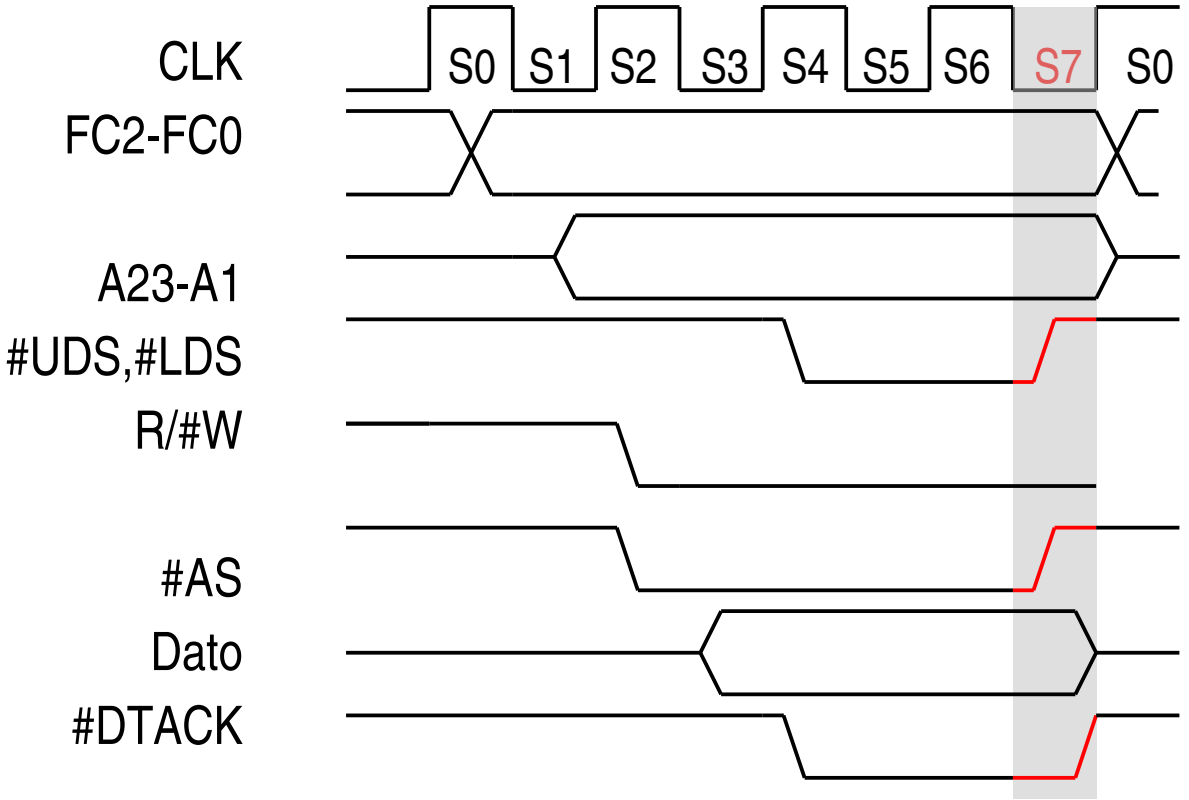
5. Buses del sistema: Control del bus asíncrono y estado del procesador

Temporización de escritura (sin estados de espera): Estado S6



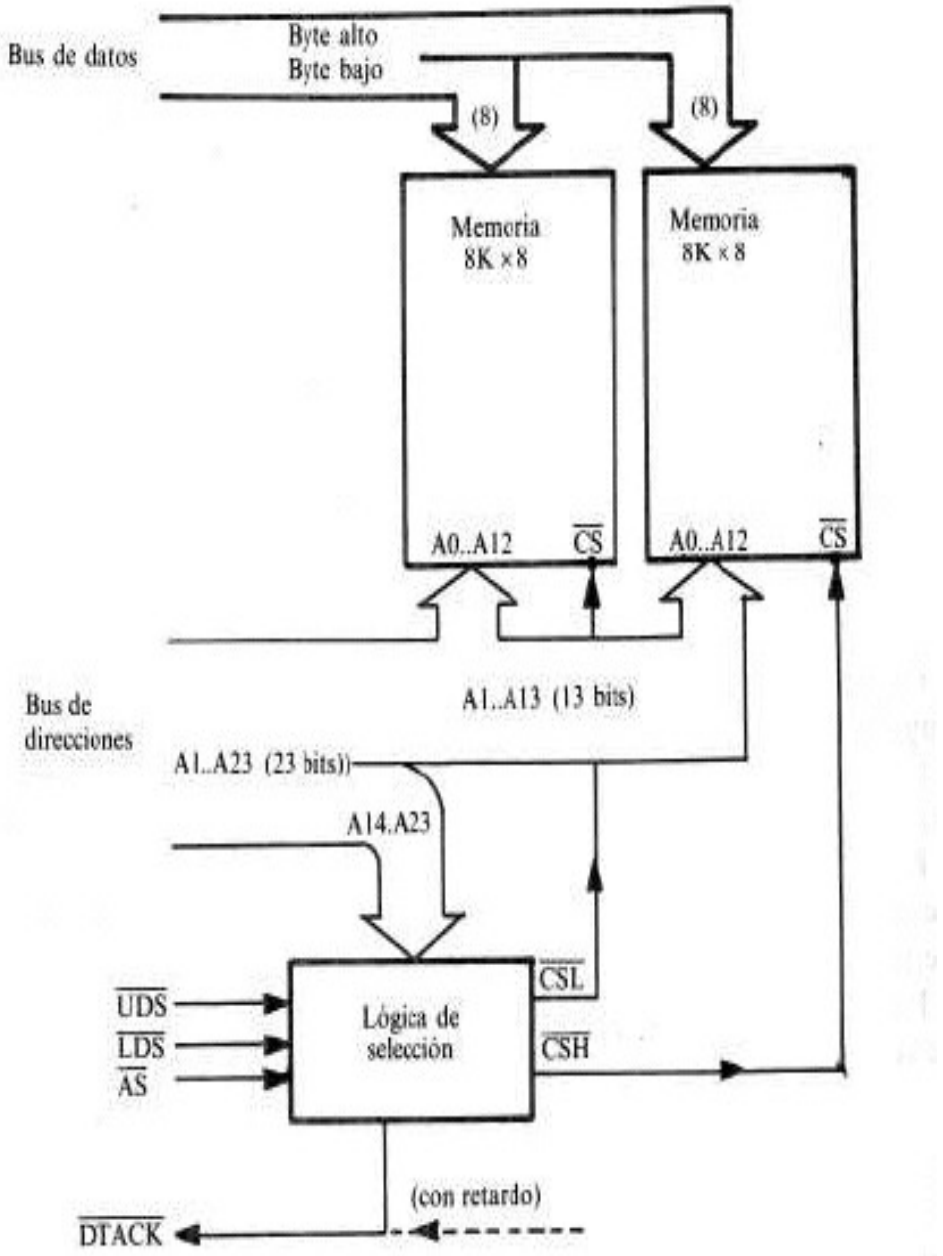
5. Buses del sistema: Control del bus asíncrono y estado del procesador

Temporización de escritura (sin estados de espera): Estado S7



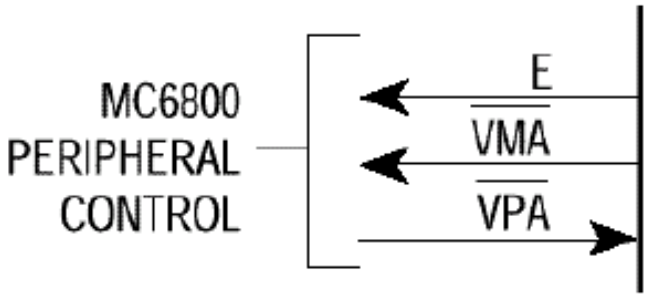
5. Buses del sistema: Conexión de memorias al MC68000

- Los chips de memoria deben venir en parejas, donde el bus de datos de uno se conecta a la mitad baja del bus y el del otro a la mitad alta.
- Las líneas menos significativas del bus de direcciones se utilizan para seleccionar la dirección interna de ambos chips.
- Las líneas más significativas, junto con #UDS, #LDS, #AS, habilitan uno o ambos chips de memoria mediante la lógica de selección que, a su vez, habilita la entrada #DTACK (con o sin retardo).

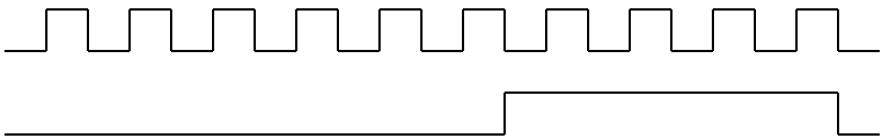


5. Buses del sistema: Control del bus síncrono

- E. Salida. Porta una señal de reloj que se conecta en las entradas de reloj de los periféricos síncronos. Un periodo de E equivales a 10 periodos de CLK (6 en bajo y en alto). Siempre está corriendo.



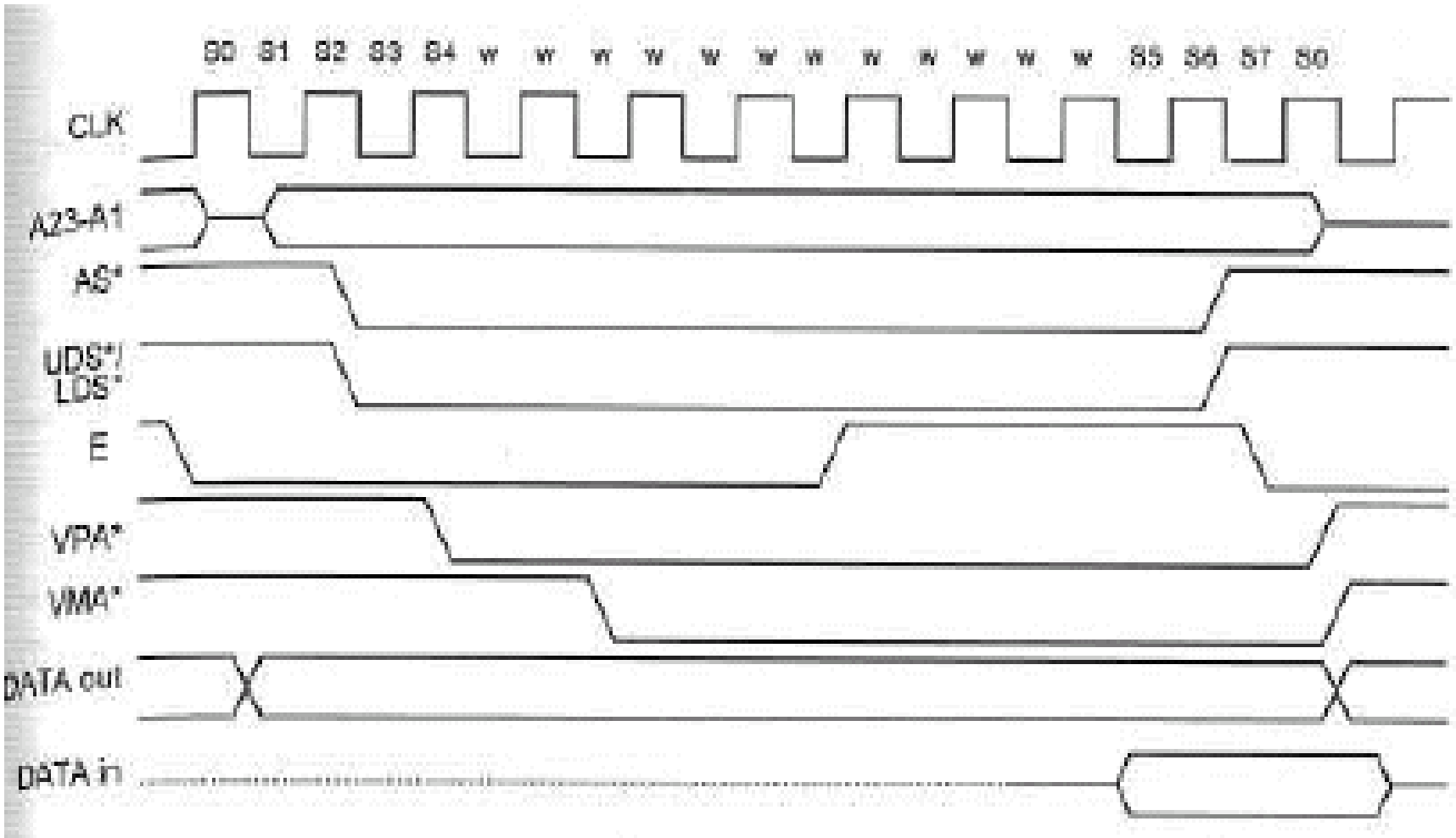
- Clk



- -#VPA. Entrada. Activa en bajo. Indica que:
  - 1) Se está direccionando periféricos síncronos y que el MC68000 debe sincronizarse con la señal E.
  - 2) Se debe utilizar vectorización automática.
- #VMA. Salida. Activa en bajo.
  - Su activación indica al periférico que dispone de una dirección válida y que el microprocesador está sincronizado con la señal E.

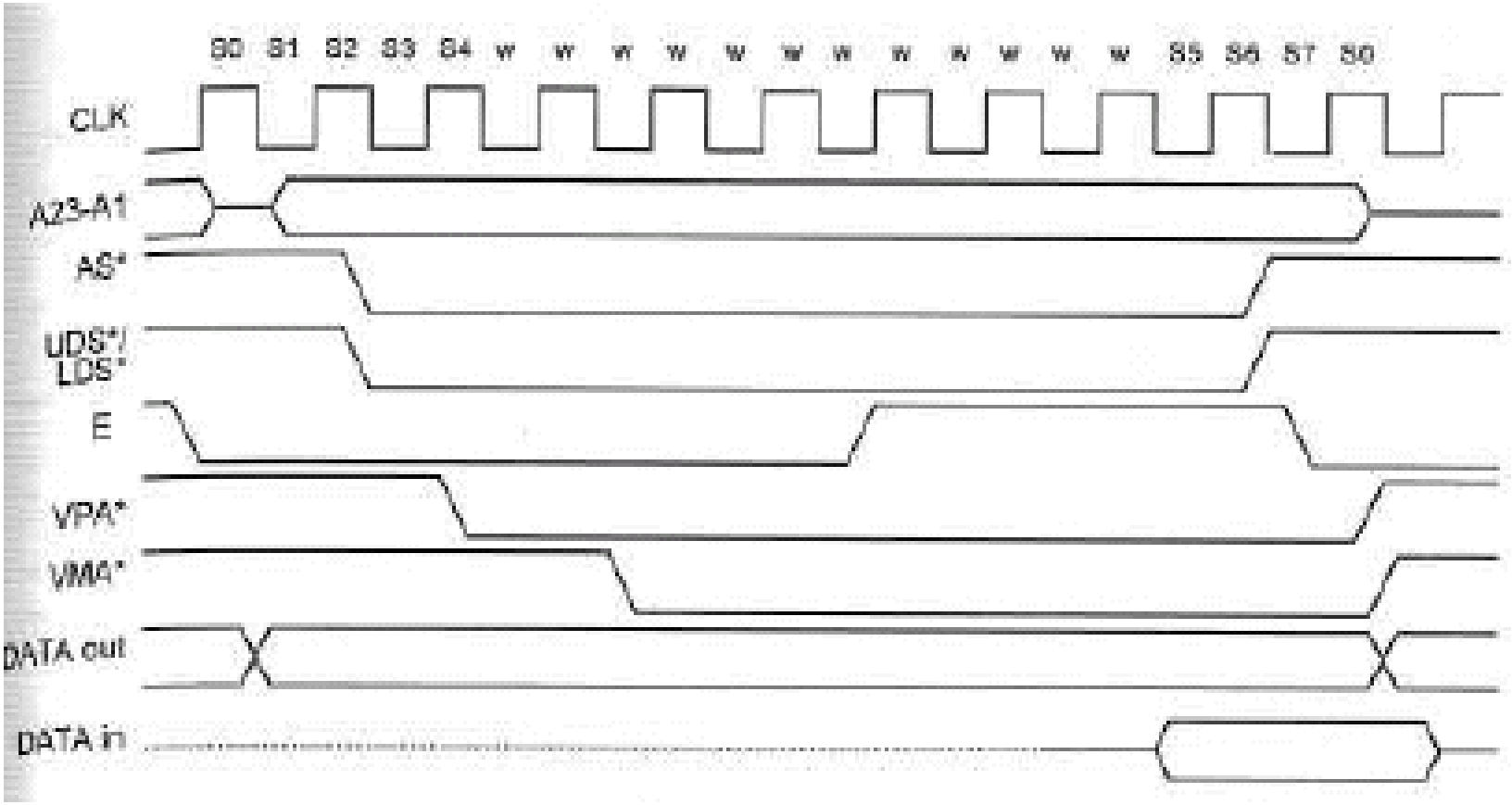
5. Buses del sistema: Control del bus síncrono

- El procesador inicia un ciclo normal de escritura o lectura
- Se activa /VPA indicando que se está accediendo a un periférico síncrono.
- El procesador espera hasta que la señal E pase a nivel bajo, entonces activa /VMA



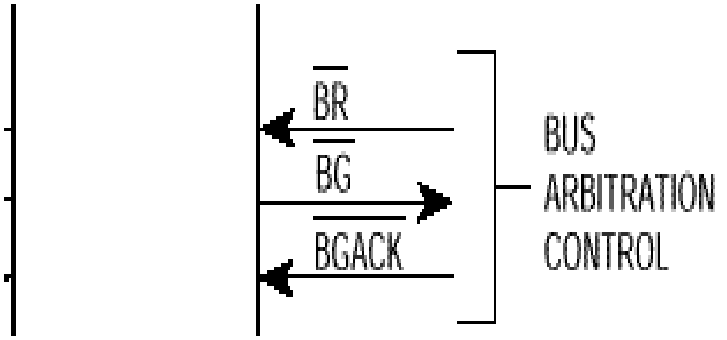
5. Buses del sistema: Control del bus síncrono

- El periférico espera hasta que E se active, entonces transfiere los datos.
- El procesador espera que E vuelva a 0 (en un ciclo de lectura los dato son cargados), entonces desactiva /VMA y /AS, /UDS y /LDS.



5. Buses del sistema: Protocolo de arbitraje

Son señales que permiten que otros dispositivos distintos de la CPU puedan utilizar el bus



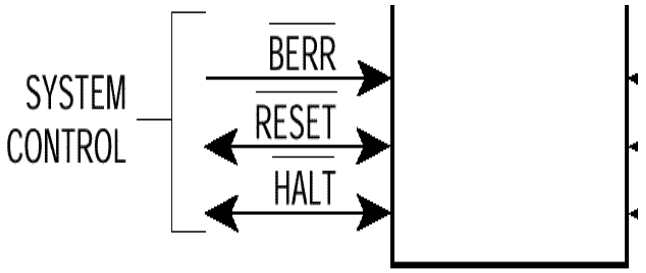
#BR: *Bus Request.* Esta línea de entrada indica al microprocesador que existe un controlador externo que solicita ser el controlador del bus para acceder a la memoria.

#BG: *Bus Grant*. Esta línea de salida indica a posibles controladores externos que el microprocesador cede el bus cuando termine el ciclo de bus actual.

#BGACK: *Bus Acknowledge*. Esta entrada indica al micoprocesador que un dispositivo externo controla ahora el bus. Cuando el controlador termine, desactiva esta entrada y el microprocesador se hace cargo nuevamente del bus.

5. Buses del sistema: Control del sistema

#BERR. *Bus error*. Entrada. Su activación avisa de la existencia de un problema en el ciclo de bus.
#RESET, #HALT. Son señales bidireccionales, se pueden utilizar de forma separada o conjunta entre ellas o con #BERR



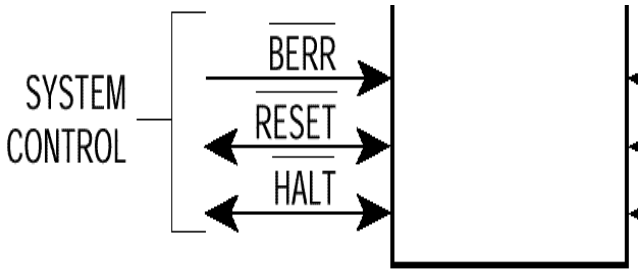
Utlización de #RESET y #HALT como salidas :

• Ejecución de la instrucción RESET: el procesador activa la señal #RESET para inicializar los controladores que tiene conectados.

•Si el procesador se encuentra con errores al tratar los propios errores (doble error de bus, explicado más adelante) y no puede continuar, se queda parado y activa la señal #HALT. Esta señal también se activará al ejecutar la instrucción STOP. Sólo se puede salir de esta situación reiniciando el procesador.

5. Buses del sistema: Control del sistema

Utlización de #BERR y #RESET y #HALT como entradas:



• Con laseñal #BERR se puede decir a la CPU que aborte el ciclo actual y pase a tratar un error de BUS (excepción de error de BUS), no debe estar activa al mismo tiempo que #HALT. Esta señal se deberá activar después de #AS y antes de #DTACK. Razones para su activación:

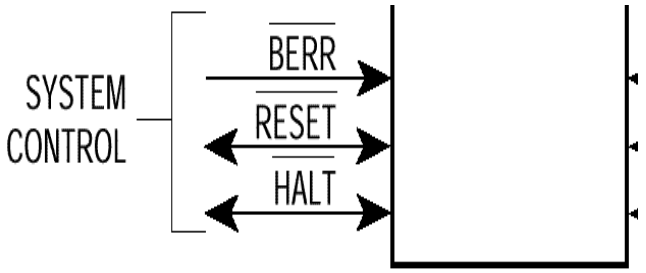
- Se intenta acceder a una dirección no implementada físicamente.
- Se intenta acceder en modo usuario a espacios a los que sólo se debe acceder en modo supervisor.
- Se intenta escribir sobre espacios en los que sólo se debe leer.
- la señal #DTACK no se ha activado tras agotar los estados de espera.

• Activando #HALT sin que este activa #BERR, el procesador terminará el ciclo actual y se quedará parado, manteniendo buses y datos en alta impedancia.

• Si se activan #HALT y BERR simultáneamente, se le indica al procesador que ha habido un error y que debe de volver a intentar desde el principio el acceso a memoria que estaba haciendo

5. Buses del sistema: Control del sistema

Utlización de #BERR y #RESET y #HALT como entradas:



- Activando #RESET y #HALT se entra en un ciclo de inicialización del procesador (excepción de RESET). Al encender un sistema basado en este micro hay que mantener activas esta dos señales un mínimo de 10ms para una inicialización correcta. Cuando veamos las excepciones veremos que durante un ciclo de RESET se inializan los valores del puntero de pila y del contador del programa y que todo esto se hace trabajando en modo supervisor.

5. Buses del sistema: Control del sistema

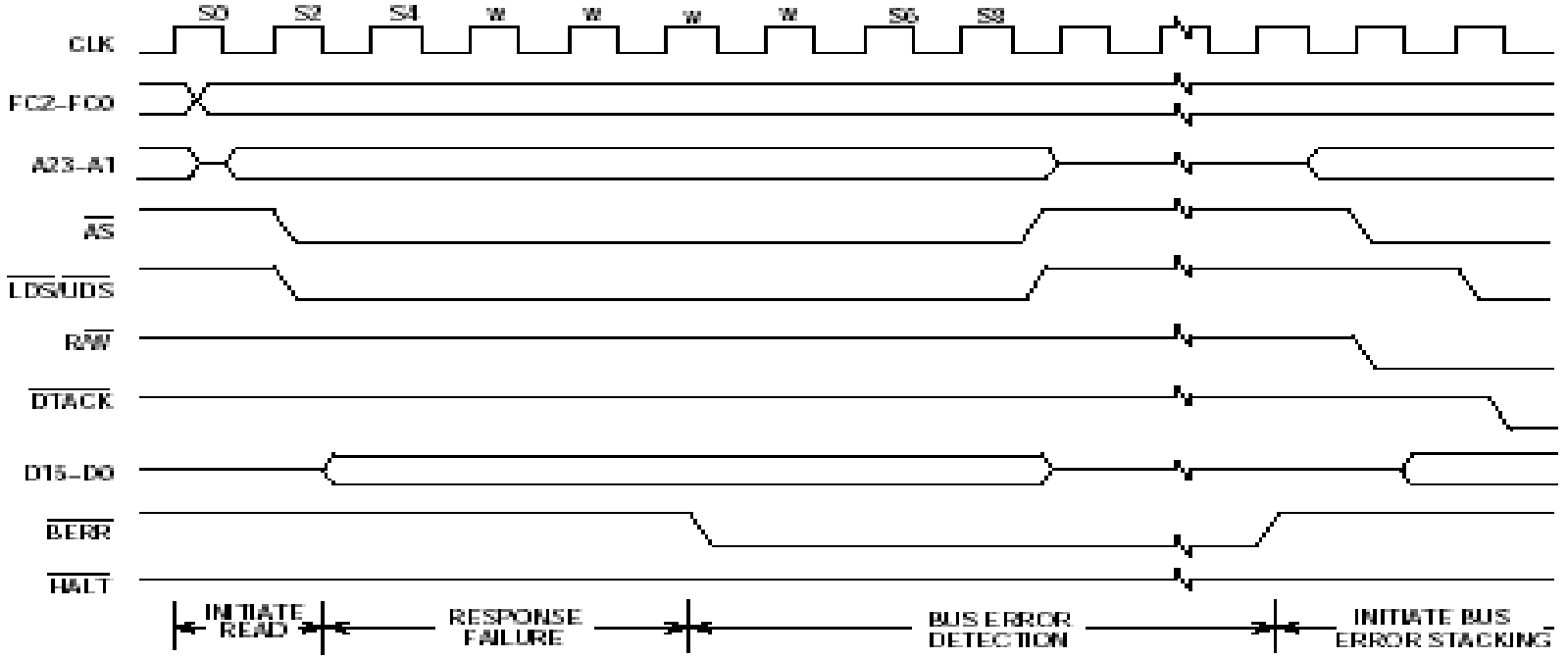


Diagrama temporal del reintento del ciclo de bus explicado anteriormente

Ocurre si se activó #BERR durante un ciclo en el cual estaba activo #HALT. Cuando se desactiva #HALT, el microprocesador reintenta el último ciclo de bus usando los mismos códigos de función, dirección y datos (para una operación de escritura).

5. Buses del sistema: Control del sistema

Doble error de bus

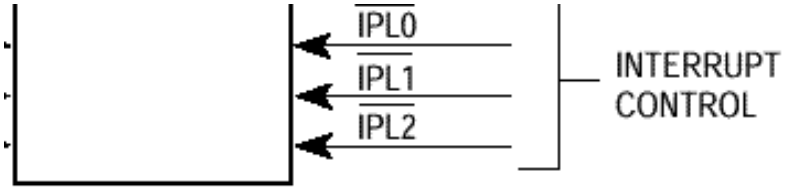
- Si se activa #BERR cuando #HALT está inactivo, el MC68000 inicia el procesamiento de una excepción.

- Si ocurre otro error de bus durante el proceso de excepción (antes de la ejecución de otra instrucción) el microprocesador para y coloca la línea de #HALT a 0. Esto se reconoce como doble error de bus. Únicamente mediante un reset externo se puede sacar el microprocesador de este estado de parada.

5. Buses del sistema: Terminales de petición de interrupciones

#IPL2-0. Entradas

• Codifican el nivel de petición de interrupción. El nivel 0, #IPL0#IPL1#IPL2=111 indica que no hay interrupción, y el nivel 7, #IPL0 #IPL1 #IPL2=000, indica la interrupción de mayor prioridad y que no es enmascarable.

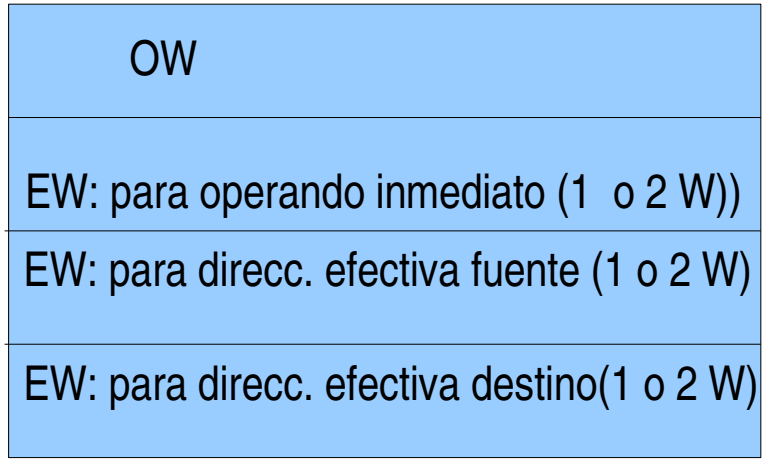


• Deben permanecer sin modificarse hasta que el microprocesador inicie el ciclo de reconocimiento de interrupciones.

• El nivel de interrupción que puede atender el microprocesador viene fijado por los bits I0,I1,e I2 del SR. Si el nivel de petición de interrupción es de mayor prioridad que el fijado por la máscara, se atenderá la petición salvo para la NMI (nivel 7) que se atenderá igualmente.

6. Formato de instrucción máquina

- Una instrucción máquina del MC68000 puede contener entre una y cinco palabras: la primera se denomina PALABRA DE OPERACIÓN (OW), las restantes, palabras de extensión (EW).
- El orden de escritura en memoria de las palabras se muestra a continuación:



6. Formato de instrucción máquina

La OW puede contener los siguientes campos:

* Código de operación (OPCODE)

* Tamaño del operando

* Dirección efectiva (EA) que tiene dos subcampos:

  - Modo

  - Registro

	EA	
MODO DE DIRECCIONAMIENTO	MODO	REGISTRO
Directo de registro de datos	000	Número de registro (Dn)
Directo de registro de direcciones	001	Número de registro (An)
Indirecto de registro	010	Número de registro (An)
Indirecto de registro con postincremento	011	Número de registro (An)
Indirecto de registro con predecremento	100	Número de registro (An)
Indirecto de registro con desplazamiento	101	Número de registro (An)
Indirecto de registro indexado con desplazamiento	110	Número de registro (An)
Absoluto corto	111	000
Absoluto largo	111	001
Relativo a PC con desplazamiento	111	010
Relativo a PC indexado con desplazamiento	111	011
Inmediato	111	100

6. Formato de instrucción máquina

- Las PALABRAS DE EXTENSIÓN (EW) contienen información de apoyo a los modos de direccionamiento (campo EA de OW).

Ejemplo:

Formato de la instrucción MOVE



MOVE.W $2000(A3),$010200

OW : 0011001111101011
EW1: 0010000000000000
EW2: 0000000000000001
EW2: 0000001000000000

	EA	
MODO DE DIRECCIONAMIENTO	MODO	REGISTRO
Directo de registro de datos	000	Número de registro (Dn)
Directo de registro de direcciones	001	Número de registro (An)
Indirecto de registro	010	Número de registro (An)
Indirecto de registro con postincremento	011	Número de registro (An)
Indirecto de registro con predecremento	100	Número de registro (An)
Indirecto de registro con desplazamiento	101	Número de registro (An)
Indirecto de registro indexado con desplazamiento	110	Número de registro (An)
Absoluto corto	111	000
Absoluto largo	111	001
Relativo a PC con desplazamiento	111	010
Relativo a PC indexado con desplazamiento	111	011
Inmediato	111	100

6. Formato de instrucción máquina

El formato de la OW varia según el número de operandos de la instrucción

INSTRUCCIONES SIN OPERANDOS.
SINTAXIS ENSAMBLADOR: **Código_op**

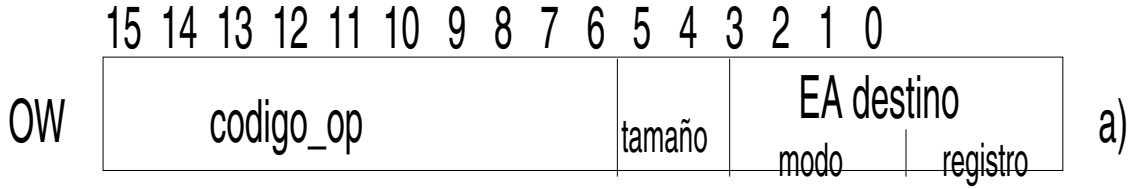


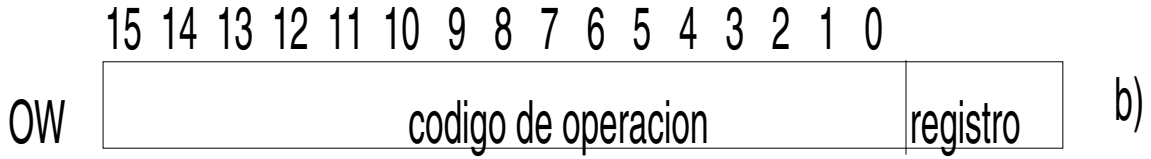
Ejemplos: RTS, NOP, RTE

6. Formato de instrucción máquina

INSTRUCCIONES CON UN OPERANDO.

SINTAXIS ENSAMBLADOR: **Código_op.TAM op_destino**





Formato a): Consta de tres campos-> (Cod.Ope ; Tam ; EA destino)

Ejemplos a): CLR.L D0, NOT.B D1, NEG.W D2

Formato b): Instrucciones que sólo usan un registro de datos como operando -> dos campos-> (Cod.Ope ; reg destino)

Ejemplos b): SWAP D3, EXT D1

6. Formato de instrucción máquina

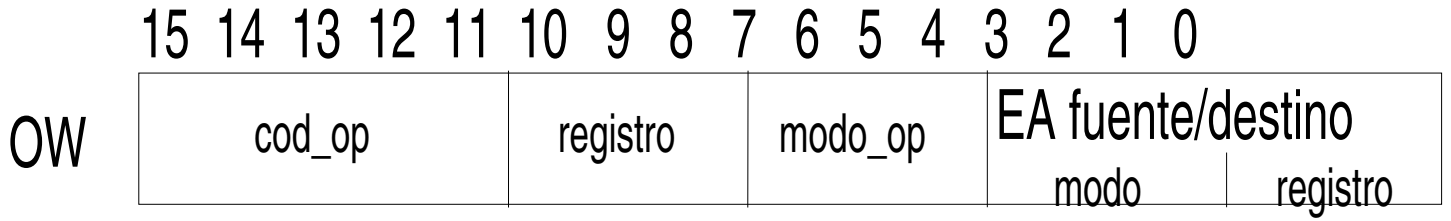
INSTRUCCIONES CON DOS OPERANDOS.
SINTAXIS ENSAMBLADOR: **Código_op.TAM op_fuente, op_destino**



Formato (a) Los dos operandos están especificados por EAs
Ejemplo: MOVE D3,D1

6. Formato de instrucción máquina

INSTRUCCIONES CON DOS OPERANDOS.
SINTAXIS ENSAMBLADOR: **Código_op.TAM op_fuente, op_destino**



Byte	Palabra	Doble pal.	Operación
000	001	010	<registro>OP<EA> → registro
100	101	110	<EA>OP<registro> → EA

Formato (b): Un operando es un registro de datos y el otro por una EA.
Ejemplo :ADD.B (A1),D0 ; CMP.W 8(A3),D0

6. Formato de instrucción máquina

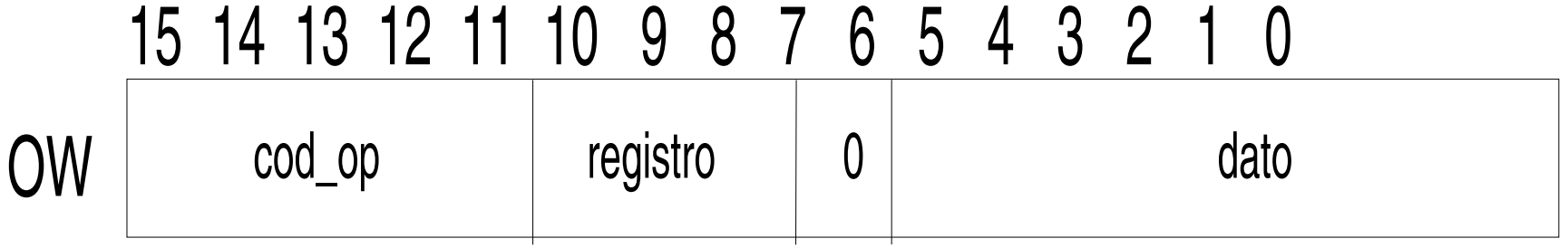
INSTRUCCIONES CON DOS OPERANDOS.
SINTAXIS ENSAMBLADOR: **Código_op.TAM op_fuente, op_destino**

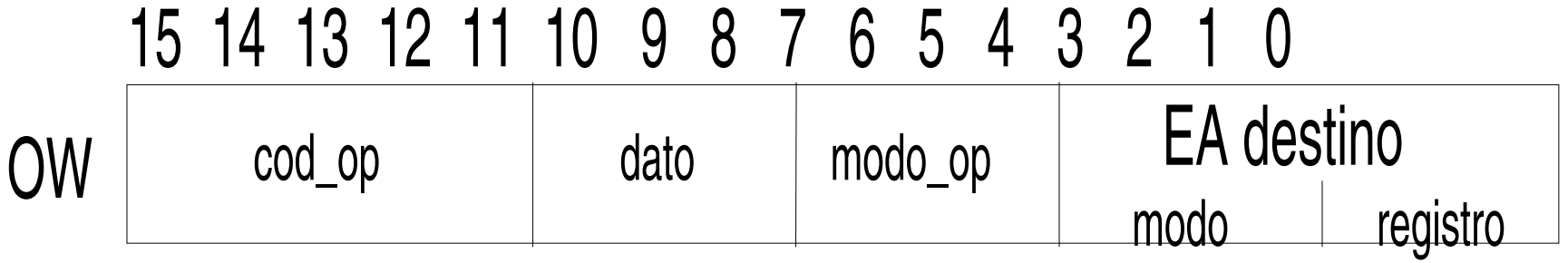


modo_op especifica el tamaño de la operación y el tipo de registros que intervienen (datos o direcciones)

6. Formato de instrucción máquina

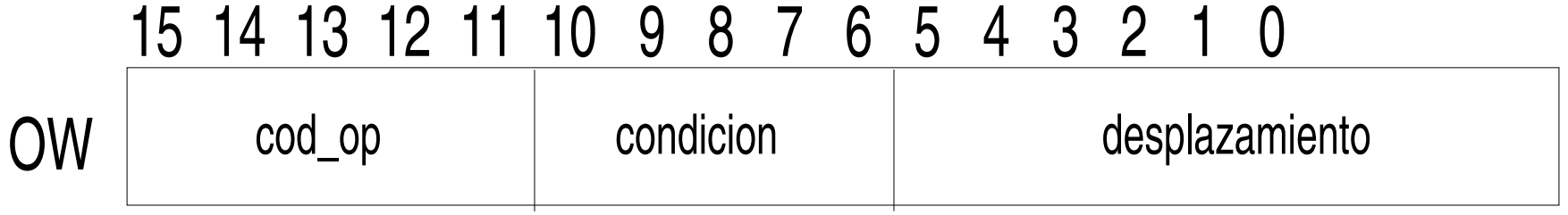
Existen formatos especiales (FORMATOS CORTOS – instrucciones ADDQ, SUBQ y MOVEQ) donde uno de los operandos es una pequeña constante que se codifica en la propia OW.





6. Formato de instrucción máquina

Las instrucciones de bifurcación condicional, Bcc, tienen un formato especial: la OW incluye un campo de desplazamiento y un campo de condición de la bifurcación.



7. Modos de direccionamiento

El MC68000 permite 12 modos de direccionamiento para localizar a los operandos.

Podemos dividir los modos de direccionamiento en 5 grupos:

- **Directo de registro :** el operando es el dato contenido en un registro de datos o de direcciones. Incluye: **directo de registro de datos y directo de registro de direcciones.**
- **Indirecto de registro:** El operando se encuentra en memoria y es necesario obtener su dirección a partir de un registro de direcciones. Incluye: **Indirecto de registro, indirecto de registro con postincremento, indirecto de registro con predecremento, indirecto de registro con desplazamiento e indirecto de registro indexado con desplazamiento.**
- **Absoluto:** el operando está en memoria y se da su dirección explicitamente. Incluye: **absoluto largo y absoluto corto.**
- **Relativo al contador del programa:** La dirección del operando es relativa al contenido del PC. Incluye: **relativo al PC con desplazamiento y relativo al PC indexado con desplazamiento.**
- **Inmediato:** El operando se expresa de forma explícita.

7. Modos de direccionamiento

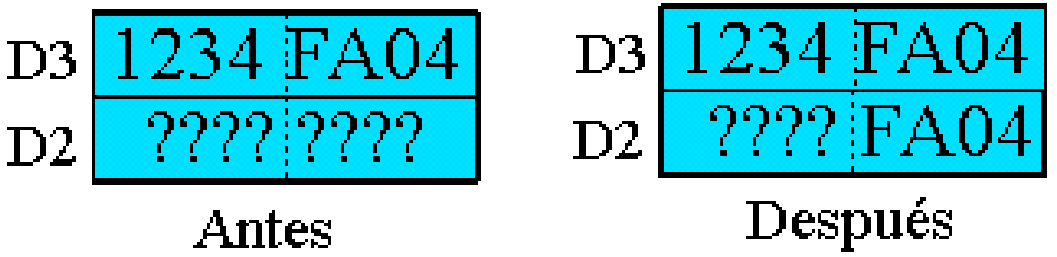
Directo de Registro Datos:

- El operando está en el registro de datos especificado por la dirección efectiva.

Sintáxis: ***MNEMÓNICO.N*** *Dn,OPERANDO_DESTINO*

Ejemplo:

MOVE.W D3,D2



7. Modos de direccionamiento

** Programa 4.1: las horas antigúas están en D1 y las que hay que*
**añadir en D2*

MOVE.L D1,D3 Las horas antigúas están en D3
ADD.L D2,D3 las horas totales en D3
BVS ERROR salta a etiqueta ERROR si hay desbordamiento
BEQ CERO Salta a CERO si el total es cero
.....

7. Modos de direccionamiento

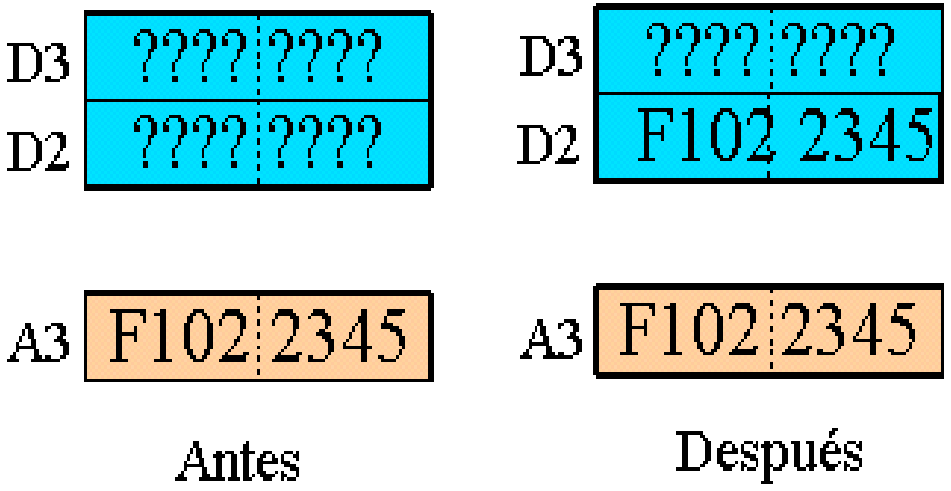
Directo de Registro Direcciones:

El operando está en el registro de direcciones especificado por la instrucción.

Sintáxis: ***MNEMÓNICO.N*** *An,OPERANDO_DESTINO*

Ejemplo:

MOVE.L A3,D2



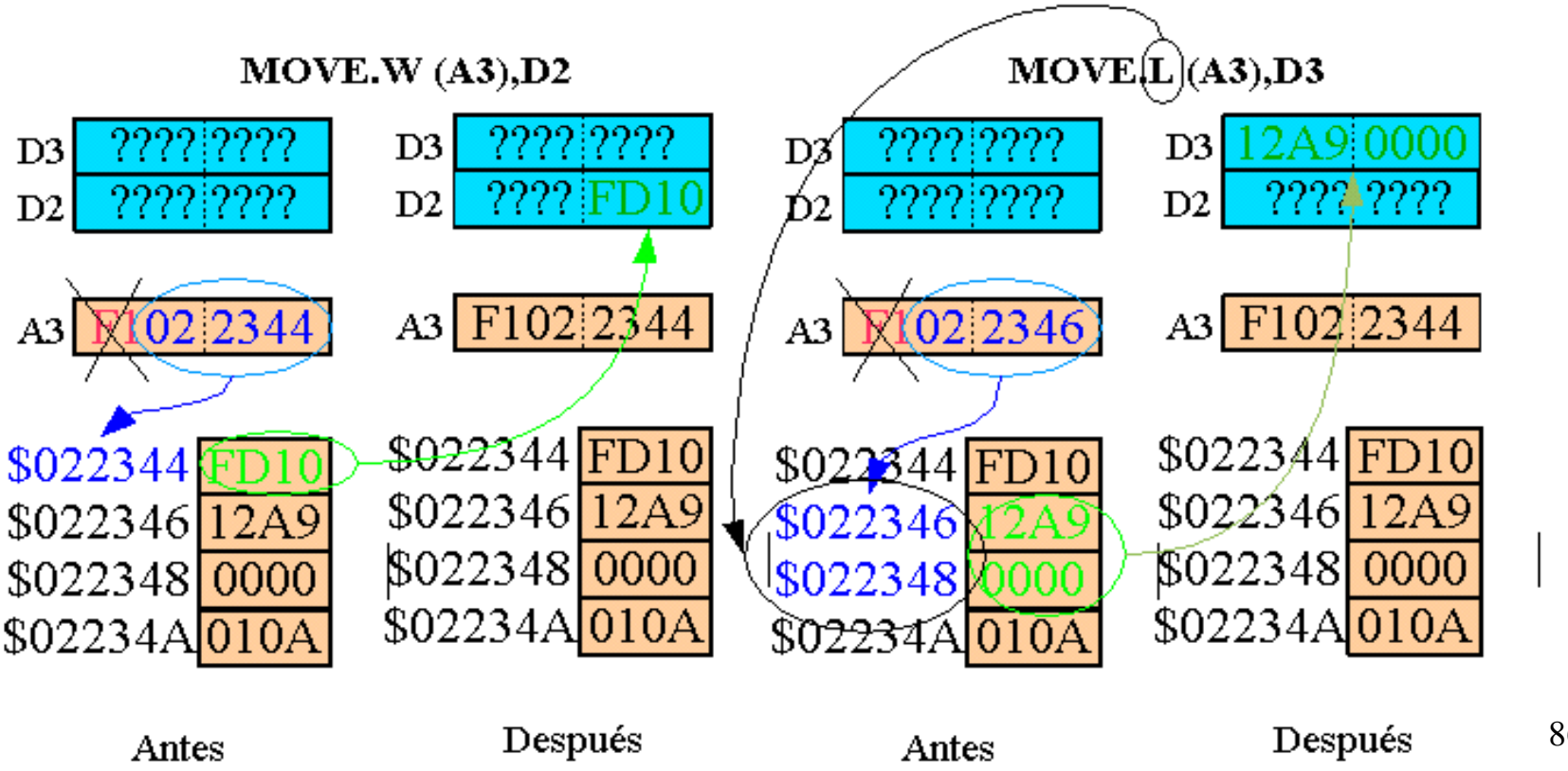
7. Modos de direccionamiento

Indirecto de Registro Direcciones:

El operando está en la posición de memoria indicada por el registro de direcciones especificado en la instrucción.

Sintáxis: ***MNEMÓNICO.N*** *(An),OPERANDO_DESTINO*

Ejemplos:



7. Modos de direccionamiento

** Programa 4.2: las horas antigúas están en una doble palabra en la dirección*
** $6000 y las que hay que añadir en $6004*

** Poner las direcciones en registros de dirección*
MOVEA.L #$6000, A1 A1 contiene $6000
MOVEA.L #$6004,A2 A2 contiene $6004
MOVEA.L #$6008,A3 A3 contiene $6008

MOVE.L (A1),D3   Las horas antigúas están en D3
ADD.L (A2),D3   las horas totales en D3
MOVE.L D3,(A3) Las horas totales están en la dirección $6008 de la memoria
BVS ERROR   salta a etiqueta ERROR si hay desbordamiento
BEQ CERO   Salta a CERO si el total es cero
.....

7. Modos de direccionamiento