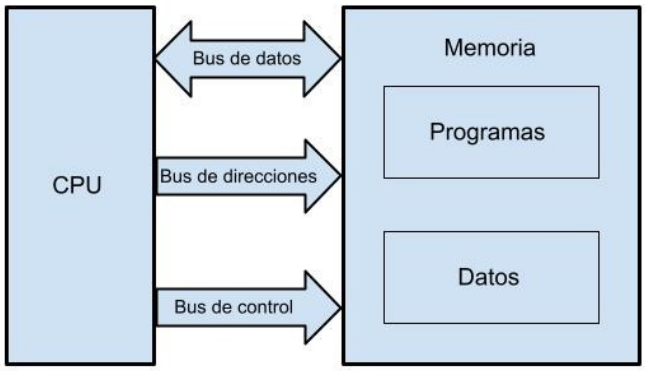
Apuntes

Tema 1: *Esquema funcional de un computador*

1. Arquitectura de von Neumann

- La mayoría de los computadores digitales actuales están basados en la arquitectura de von Neumann, es decir, los datos e instrucciones se almacenan en binario en la memoria sin existir distinciones entre estas y poseen 3 partes(procesador(CPU), memoria y unidad de E/S).
- Internamente, la CPU funciona como un pequeño ordenador:
  - Los datos se almacenan en registros, las instrucciones se leen de la memoria y las interpreta la unidad de control
  - La unidad de control establece secuencialmente caminos de datos según la instrucción que se esté ejecutando.
  - El proceso se repite indefinidamente.
- El ritmo de ejecución lo marca un reloj(circuito síncrono) por lo que el ordenador funciona como un autómata.



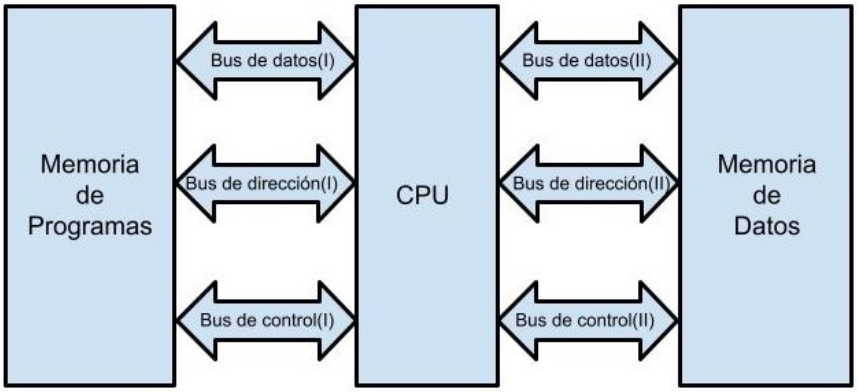
Arquitectura de von Neumann

- Si la CPU quiere escribir un dato en memoria:
  - En el bus de direcciones pone la dirección de memoria donde quiere escribir
  - En el bus de datos pone el byte que quiere escribir

- Activa en el bus de control las líneas necesarias para que se establezca el camino que lleve el dato hasta la memoria.

2. Arquitectura de Harvard:

- Tiene dos tipos de memoria, memoria de instrucciones y memoria de datos
- Ventajas
  - Aumenta la velocidad de acceso a la memoria (se puede hacer dos transferencias simultaneas)
- Inconvenientes
  - Puede haber memoria de sobra de un tipo y faltar del otro



Arquitectura Harvard

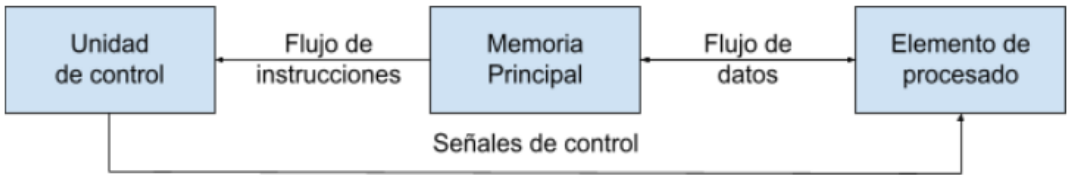
3. Evolución histórica del rendimiento

- Primera Generación: **Tubos de vacío**
  - Las mismas personas diseñaban, construían, programaban, operaban y mantenían las máquinas (no son ordenadores comerciales).
  - No había lenguajes de programación.
  - La programación podía llegar a ser electrónica.
  - La salida podían ser luces en un panel.
  - Los fallos de válvula eran habituales.
- Segunda Generación: **Transistores**
  - El transistor es más barato, más pequeño, más fiable y consume menos energía que las válvulas.

    - Se usa memoria principal de ferritas
    - La memoria no es lo suficientemente rápida para el procesador: Procesadores CISC.
    - Se crean las protecciones de memoria, interrupciones y temporizadores y las instrucciones privilegiadas.
- Tercera Generación: **Circuitos integrados**
    - Dos mundos: el mundo de la banca y el mundo científico.
    - Se empieza a hablar de compatibilidad ascendente. Los circuitos integrados abaratan los costes.
- Cuarta Generación: **Comienza la carrera de la integración, ordenadores personales**
    - El microprocesador incluye todos los componentes de la CPU en un único chip.
    - El ordenador se tiene que hacer amigable.
    - Comienza la era del PC de IBM, los MACs...
    - Se popularizan las redes de ordenadores.
    - Las generaciones a partir de esta gozan de menor consenso.
- Quinta Generación: **Alta integración (VLSI)**
    - La miniaturización hace procesadores más rápidos, pero se calientan más.
    - La tecnología del procesador evoluciona más rápidamente que la de la memoria y el disco duro.
    - Se deben usar técnicas para mantener el procesador ocupado al máximo.
- Sexta Generación: **Ultra alta integración (ULSI)**
    - Se agudiza el desequilibrio entre rapidez del proceso y rapidez de memoria.
    - Se alcanza un límite en la velocidad del reloj del procesador y se tiende al multiproceso(multinúcleo)
    - Triunfo de las arquitecturas masivamente paralelas de las tarjetas gráficas, incluso para aplicaciones de cálculo

4. Clasificación en cuanto a paralelismo

- Se tiende por paralelismo la ejecución concurrente de varias instrucciones en un ordenador. El paralelismo se puede lograr por hardware, software o una combinación de ambos.
- Categorías:
  - SISD: Single Instruction, Single Data.
    - Se corresponde con la arquitectura de Von Neumann tradicional.
    - La unidad de control toma una instrucción cada vez.
    - La instrucción se ejecuta sobre un dato, a lo sumo.



  - SIMD: Single Instruction, Multiple Data.
    - La unidad de control toma una instrucción cada vez.
    - Las unidades de proceso, todas a la vez, realizan la ejecución cada una sobre un dato diferente.
    - típico de ordenadores vectoriales/matriciales, GPUs.
    - Aumento espectacular de rendimiento para aplicaciones de naturaleza vectorial.
  - MISD: Multiple Instruction, Single Data.
  - MIMD: Multiple Instruction, Multiple Data.
    - Cada unidad toma y ejecuta sus propias instrucciones sobre sus propios datos.
    - Se sincronizan ocasionalmente.
    - Es lo que se conoce como multiprocesador.
    - Tres tipos:
      - Multiprocesadores de memoria compartida (SMP):
        - Espacio de direcciones de memoria común.
        - Suelen tener cachés individuales.

    - Hay un bus común.
- Multiprocesadores de memoria compartida distribuida (DSM):
    - Aunque todos pueden acceder a toda la memoria a través de un bus común, cada uno tiene un módulo de acceso propio.
    - El tiempo de acceso varía según dónde se quiera acceder(arquitectura NUMA).
    - Espacio de direcciones de memoria común.
- Multiprocesadores de memoria distribuida:
    - Espacio de direcciones independiente
    - Se pueden comunicar pero por paso de mensaje.

Tema 2: *Procesadores CISC*

2.1 Registros

Los registros constituyen un almacenamiento interno del procesador accesible por el programador. Es la memoria más rápida de que se dispone.

///FOTO REGISTROS

Los registros pueden contener valores, según su tamaño. Los registros A y B son de 8 bits, pero se pueden combinar dando un registro de 16 bits, llamado D.

///FOTO CPU ELEMENTAL

2.1.1 El acumulador

El acumulador es el registro involucrado en la mayor parte de operaciones de un procesador CISC. Su tamaño en bits determina el tamaño en bits del procesador.

Especialmente interviene en operaciones de transferencia de datos, operaciones aritméticas enteras y operaciones lógicas.

Cuando la operación es unaria, el origen y destino suele ser el propio acumulador. Pero cuando la operación es binaria, el origen o el destino es el acumulador mientras que el otro se debe especificar a parte.

2.1.2 El contador de programa

El contador de programa pertenece a un conjunto de registros denominados apuntadores, pues su contenido hace referencia o apunta a una dirección de

memoria.

El contador de programa apunta a la dirección de memoria donde se aloja el código máquina de la instrucción que se va a ejecutar a continuación.

Cargar el PC con un valor supone que la ejecución del programa salta a la instrucción situada en la dirección de memoria correspondiente con el valor que hemos cargado.

También hay instrucciones que suman algo al PC, haciendo que, se omitan las instrucciones situadas entre la dirección actual del PC y el lugar donde se produce el salto.

En el momento de la ejecución de una instrucción, el PC ya se encuentra apuntando a la instrucción siguiente por lo que no hay que contar los bytes de la propia instrucción para calcular el salto.

2.1.3 El puntero pila

Una pila es una estructura de datos con dos operaciones, push (meter datos) y pop (sacar datos), que sigue un protocolo LIFO (Last In, First Out).

///FOTO ESTRUCTURA PILA

Funcionamiento del puntero pila:

- El apuntador apunta a la dirección del último dato que se metió.
- La pila crece hacia direcciones bajas: si se mete algo, el puntero decrementa, si se saca algo el puntero se incrementa.
- Uno de los usos de la pila es para guardar temporalmente el contenido de los registros mientras se usan para otra cosa. En el caso de que se guarden varios registros, hay que restaurarlos en el sentido inverso en que se metieron. Si no se hace así, podemos encontrarnos con que el contenido de los registros se ha intercambiado.

Llamadas a subrutinas:

- El PC apunta ahora el inicio del procedimiento. En la pila ha quedado apuntada la dirección de retorno para no olvidarla. El puntero de pila se ha decrementado para meter ahí nuevos valores.
- La CPU ha sido capaz de retornar del procedimiento gracias a la pila. EL puntero de la pila SP se ha incrementado al sacar la dirección de retorno. El resto de instrucciones se ejecutarían como antes.

Tampoco es extraño encontrar procesadores en los que la dirección de la subrutina se expresa como un desplazamiento relativo al PC.

2.1.4 El registro de flags

El registro de flags es un registro auxiliar del procesador. Su significado cobra sentido cuando se considera su valor en binario.

El resultado de ciertas instrucciones (sobre todo aritméticas o lógicas) afecta al valor de dichos bits (banderas o flags). En el 6809 es el registro CC.

Aunque el nombre y la cantidas de flags varian entre procesadores, existen algunos comunes:

- Acarreo (carry), C: se activa cuando hay acarreo en sumas y restas. También se usa como flag de apoyo en operaciones de desplazamiento de bits.
- Paridad (parity), P: paridad par.
- Medio acarreo (half carry), H: acarreo del bit 3 al 4. Usando en BCD.
- Cero (zero), Z: se activa si el resultado de una operación es cero.
- Signo (sign), N: activo cuando el resultado es negativo.
- Desbordamiento (overflow), V: se activa si hay desbordamiento.

No todas las operaciones afectan igual a los flags en todos los procesadores. Se debe consultar la hoja de datos.

- Comparaciones condicionales:

  La clave del uso de los flags consiste en poder realizar saltos condicionales, es decir, saltos que se realicen o no dependiendo del valor de un flag.

	Salto si flag activo	Salto si flag inactivo
N	bmi (*minus*)	bpl (*plus*)
Z	beq (*equal*)	bne (*not equal*)
V	bvs (*V set*)	bvc (*V clear*)
C	bcs (*C set*)	bcc (*C clear*)

- Comparaciones complejas:

  Son aquellas comparaciones basadas en más de un flag.

Comparación	con signo	sin signo

Comparación	con signo	sin signo
Mayor que	bgt (*greater than*)	bhi (*higher*)
Mayor o igual que	bge (*greater or equal*)	bhs (*higher or same*)
Igual que	beq (*equal*)	beq (*equal*)
Diferente que	bne (*not equal*)	bne (*not equal*)
Menor o igual que	ble (*less or equal*)	bls (*lower or same*)
Menor que	blt (*less than*)	blo (*lower*)

La siguiente tabla nos muestra qué operaciones lógicas se realizan sobre los flags para ver si se produce el salto o no en las distintas comparaciones:

Comparación	con signo	sin signo
Mayor que	bgt	bhi
Mayor o igual que	bge	bhs
Igual que	beq	beq
Diferente de	bne	bne
Menor o igual que	ble	bls
Menor que	blt	blo

- Control de Flujo

  If-then-else:

```
if(A>7) // considerado con signo
  {// Instrucciones si es verdad la condición
   }
else
  {// Instrucciones si es falsa la condición
   }

        cmpa #7
        ble el_else
        ;; Instrucciones si es verdad la condicion
        .ds 10 ; sumulamos las instrucciones
        ;; ...
        bra seguir
el_else:
            ;; Instrucciones si es falsa la condicion
        .ds 10 ; sumulamos las instrucciones
            ;; ...
seguir:     ;; ... lo que venga a continuacion
```

  If-then-else condición AND:

```
if(A>=100 and A<=200) // considerado sin signo
  {// Instrucciones si es verdad la condicion
   }
else
  {// Instrucciones si es falsa la condicion
   }

        cmpa #100
        blo p_else
        cmpa #200
        bhi p_else
        ;; Instrucciones si es verdad la condicion
        .ds 10 ; sumulamos las instrucciones
        ;; ...
        bra p_seguir
p_else:
            ;; Instrucciones si es falsa la condicion
        .ds 10 ; sumulamos las instrucciones
            ;; ...
p_seguir:     ;; ... lo que venga a continuacion
```

Switch:

```
switch(A)
  {case 'S':
     // Instrucciones para el caso A=='S'
     break;
   case 'N':
     // Instrucciones para el caso A=='N?
     break;
   default:
     // Instrucciones para cualquier otro caso
}

        cmpa #'S
        bne caso_n
        ;; Instrucciones si A es 'S'
        .ds 10 ; sumulamos las instrucciones
        ;; ...
        bra c_seguir
caso_n: cmpa #'N
        bne c_default
        ;; Instrucciones si A es 'N'
        .ds 10 ; sumulamos las instrucciones
        ;; ...
        bra c_seguir
c_default:
            ;; Instrucciones si es falsa la condicion
        .ds 10 ; sumulamos las instrucciones
            ;; ...
c_seguir:     ;; ... lo que venga a continuacion
```

Repeat (do...while)

```
do
  {// Instrucciones del cuerpo del bucle
    }
while(A!=0x20)

b_repeat:  ;; Instrucciones del cuerpo del bucle
           .ds 10 ; simulamos las instrucciones
           ;; ...
           cmpa #0x20
           bne b_repeat
```

While

```
while (A>137) // Comparación sin signo
  {// Instrucciones del cuerpo del bucle
   }

b_while:  cmpa #137
          bls w_seguir
          ;; Instrucciones del cuerpo del bucle
          .ds 10 ; simulamos las instrucciones
          ;; ...
          bra b_while
w_seguir: ;; ... lo que venga a continuación
```

For

```
for (A=1;A<10;A=A+2)
  {// Instrucciones del cuerpo del bucle
   }

          lda #1
b_for:    cmpa #10
          bge f_seguir
          ;; Instrucciones del cuerpo del bucle
          .ds 10 ; simulamos las instrucciones
          ;; ...
          adda #2
          bra b_for
f_seguir: ;; ... lo que venga a continuacion
```

2.1.5 Los registros índice. Direccionamiento

A las posibles maneras en que una instrucción de código máquina puede localizar los datos que necesita se les denomina modos de direccionamientos.

- Direccionamiento inherente o implícito:

  Los datos que necesita la instrucción no están especificados en la propia instrucción, bien porque no hay datos, bien porque la localización es única.

- Direccionamiento inmediato:

  Los datos aparecen en el propio código máquina de la instrucción.

- Direccionamiento directo:

  En la instrucción aparece el modo de localizar los datos directamente, bien dando una dirección de memoria, bien especificando uno de los registros.

- Direccionamiento indirecto:

  Funciona como el direccionamiento directo, pero el dato que se obtiene es, en realidad, una dirección de memoria donde se encuentra el dato definitivo.

- Direccionamiento indizado directo:

  Su objetivo principal es manejar arrays en ensamblador. Cuando queremos acceder a un elemento en concreto del array a la dirección donde comienza en array se le denomina dirección base y al número de bytes que hay desde el inicio hasta el elemento que queremos acceder, se le denomina desplazamiento.

- Direccionamiento indizado indirecto:

  Funciona como combinación de los direccionamientos indizados vistos en el apartado anterior y el direccionamiento indirecto.

- Direccionamiento indizado al pc:

  Hacer indización sobre el PC tiene un curioso efecto secundario: el código máquina se vuelve directamente ***reubicable***, es decir, se puede cargar en cualquier lugar sin cambiarlo.

2.2 Esquema de bloques

Un esquema de bloques representa de modo somero las unidades funcionales de un sistema.

//ESQUEMA BLOQUES DE UN PC GENÉRICO

2.3 Fases de ejecución de una instrucción

2.3.1 Interrupciones

Mecanismos asíncronos por los que se interrumpe la ejecución normal de un programa para llamar a un procedimiento de servicio a un suceso.

Cuando una interrupción ha sido solicitada pero no se atiende se dice que está enmascarada. La petición se guarda y se atiende cuando la interrupción se desenmascara.

Pasos que se produce cuando e recibe y se atiende una interrupción:

- La CPU acaba la ejecución de la instrucción en curso.
- Se salta automáticamente a la dirección del procedimiento de servicio de la interrupción como si de otro procedimiento se tratase.

Como no se sabe cuando se va a producir la interrupción es importante que la rutina de tratamiento deje al final todos los registros del procesador como se los encontró.

No todas las interrupciones son enmascarables. También puede haber prioridades.

Una interrupción de software, al contrario de las interrupciones habituales, es provocada por la ejecución de una instrucción concreta del código máquina. Sus rutinas de servicio suelen apuntar a código del sistema operativo.

Se denomina vectores de interrupción a una tabla donde están las direcciones de memoria donde se va a saltar cuando se produzca cada clase de interrupción.

2.3.2 Formato de las instrucciones

El byte o bytes que determina con qué instrucción estamos trabajando se denomina código de operación.

2.3.3 Fases de ejecución

El procesador es un circuito secuencial síncrono. El ritmo de su funcionamiento viene marcado por un reloj.

Algunos procesadores solamente necesitan una señal de reloj simple de determinada frecuencia. Otros como el 6809 necesitan más de una de la misma frecuencia pero desfasadas. En este caso, se podrán hacer más operaciones por ciclo de reloj.

Se denomina ciclo máquina al periodo empleado por el procesador para realizar una operación elemental, particularmente, un acceso a memoria, una operación

aritmética o de entrada/salida.

Una instrucción en ensamblador tardará, dependiendo de su complejidad, uno o varios ciclos máquina.

//IMAGEN 6809 DIRECCION DE LECTURA Y DATOS CON CICLOS

Podemos considerar genéricamente las siguientes fases en la ejecución de una instrucción de código máquina:

- Toma de instrucción (IF): se hace uno o varios ciclos máquina de lectura de memoria, dependiendo de la longitud del código de operación de la instrucción. El PC sale al bus de direcciones y se va incrementando.
- Decodificación (ID): el procesador identifica la instrucción leída y se dispone a efectuar los pasos para ejecutarla.
- Toma de operandos (OF): dependiendo de la instrucción puede ser necesario ejecutar uno o varios ciclos maquina para buscar operandos.
- Ejecución (EX): al principio de esta fase, el PC ya está apuntando a la siguiente instrucción. Si el cálculo es complicado, puede llevar varios ciclos máquina su conclusión.
- Escritura del resultado (WB): si la instrucción calcula o mueve un dato, se escribe el resultado en el destino.
- Interrupciones (II): si se ha activado una señal de interrupción, es ahora cuando se atiende.

2.4 Niveles de ejecución

Las CPUs modernas tienen al menos dos modos de funcionamiento: modo usuario y modo supervisor.

Desde el modo usuario no se puede efectuar operaciones de entrada/salida y algunas de las instrucciones de la CPU no están permitidas.

Existirá una instrucción del modo supervisor para pasar al modo usuario y de este al modo supervisor se pasará automáticamente si se produce una interrupción.

Normalmente, será el código del sistema operativo el que se encuentre en los vectores de interrupción y el que se ejecute en modo supervisor con todos los privilegios.

2.5 Frecuencia de instrucciones de uso y ortogonalidad

Para estudiar la frecuencia de uso de las instrucciones podemos realizar dos tipos de estudios:

- Estáticos: se cuentan las instrucciones que aparecen en los listados de los programas
- Dinámicos: se cuentan las instrucciones que el procesador ejecuta en una ejecución típica.

Se obtienen las siguientes conclusiones:

- La mitad de las instrucciones usadas son de transferencia de datos.
- Las instrucciones de salto son las siguientes más usadas.
- La mitad de las instrucciones de un procesador CISC se usan menos de un 2%

ORTOGONALIDAD

Se dice que un juego de instrucciones es ortogonal si cada operación se puede hacer con cualquier tipo de operando y en cualquier modo de direccionamiento.

Tema 3: *Organización de la memoria*

3.1 Mapa de memoria

El mapa de memoria indica como está distribuida la memoria física y como será direccionada por el microprocesador.

Los microprocesadores van a tener un bus de direcciones (AB), uno de datos (DB) y control (CB).

Desde el punto de vista del procesador la memoria no es más que un array de bytes en la que cada uno de ellos ocupa una dirección concreta direccionable.

El valor que el microprocesador coloca en el AB se denomina dirección física.

Cada byte en la memoria se distingue de otro porque ocupa o tiene asignado una dirección lógica, que es con lo que trabaja el microprocesador. El conjunto de todas las direcciones lógicas determina el espacio de direccionamiento.

El espacio de direccionamiento y su organización dependen del número de líneas de los buses AB, DB y CB.

Un bus de direcciones de n líneas identifica 2^n direcciones físicas diferentes.

Con un bus de control de m líneas se podría asignar m direcciones lógicas a una física.

Con un bus de datos de 8 bits se accedería a un byte de cada acceso.

Esto daría un espacio de direccionamiento de $2^n \times m$ bytes.

La capacidad máxima de memoria con la que va a poder trabajar un microprocesador viene dado por el espacio de direccionamiento lógico. La capacidad física va a venir dada por la suma de las capacidades de los chips que forman la memoria.

Cada chip de memoria va a tener asignado un rango de direcciones lógicas, que corresponderá a su capacidad. Si se accede a una dirección lógica que esté en el rango ese chip estará activo mientras los restantes permanecerán inactivos.

Los mapas de memoria son representaciones gráficas de los rangos de direcciones lógicos que tienen asignados cada uno de los chips que constituyen la memoria física.

Los microprocesadores necesitan de un sistema de decodificación que permita habilitar las memorias según la dirección física que se haya colocado en el bus.

Los chips de memoria reciben como dirección los bits menos significativos del AB. Los más significativos los recibe un decodificador que, conjunto a unos operadores lógicos de comparación determinan que chip debe estar seleccionado.

3.2 Expansión de memoria

La ampliación del mapa de memoria exige ampliar el número de bits de direcciones:

Se puede rediseñar el computador para que sea capaz de generar direcciones de mayor número de bits.

Se puede concatenar a las direcciones generadas por el computador unos bits adicionales. Estos bits generalmente se pueden modificar por programa y van a permitir seleccionar uno de varios mapas de memoria. Con 2 bits se multiplica por cuatro el tamaño de la memoria principal.

- Es una solución sencilla y económica, pero complica la forma de trabajar con la memoria, dado que hay que indicar el mapa con el que se trabaja de forma explícita.
- Debe de tener una parte de memoria común para tener la parte del programa que tiene que ejecutar a continuación.

Se puede emplear conmutación de bancos de memoria. Se divide la dirección original en dos partes:

- La parte superior es la entrada de una tabla, formada por un banco de registros, que dada una dirección d1 proporciona una nueva dirección d1’ con un número diferente de bits. Uniendo d1’ con d2 se tendrá la dirección efectiva.
- La partición de la dirección divide el mapa de memoria original en 2^{r1} bloques de 2^{r2} posiciones cada uno.
- El mapa de memoria ampliado en $2^{r1'}$ bloques de 2^{r2} posiciones cada uno.

3.3 Jerarquía de memoria

Nivel 0: Registros. Es una memoria de alta velocidad y poca capacidad integrada en el microprocesador La capacidad de los registros viene dado por la arquitectura del procesador.

Nivel 1: Memoria caché. Es una memoria intermedia entre los registros y la memoria principal que se emplea para almacenar información (datos + programas) de frecuente uso reduciendo los tiempos de acceso. Suele estar realizada con tecnología SRAM(RAM estática), más rápida y cara que la DRAM. Según su velocidad y capacidad están organizadas en L1, L2 y L3.

Nivel 2: Memoria principal. Esta memoria suele ser memoria RAM (DRAM o SDRAM) y su tamaño es del orden de los Gigabytes.

Nivel 3: Almacenamiento secundario. Cuando se acaba la memoria RAM es posible emplear el disco duro como memoria virtual. Además permite guardar información de forma permanente. Como almacenamiento secundario podemos tener discos duros SSD, HDD, memorias flash, etc. Su tamaño es del orden de Gigabytes a Terabytes.

Tema 4: *Unidad de control*

Tema 5: *Buses*

5.1 Introducción

Un bus es una vía de comunicación que conecta dos o más dispositivos. La principal característica de los buses es que son medios de transmisión compartidos.

Los buses constan de un camino que permite comunicar de manera selectiva un cierto número de dispositivos de acuerdo a unas normas de comunicación.

- Enlace: Elemento que permite transmitir información entre dos o más dispositivos.
- Conmutador: Es un elemento que permite encaminar la información entre varios enlaces.

En las transferencias de información que se realizan en los buses existen dos agentes involucrados.

- Maestro: El que origina la transferencia (no todos los elementos conectados pueden serlo).
- Esclavo: El que responde a la transferencia.

La operación más básica para la realización de una transferencia elemental de un dato entre dos dispositivos se llama ciclo o transacción de bus.

Principales características de los buses:

- Funcionamiento en banda base.
- Paralelismo: bus serie, bus paralelo y bus multiplexado.
- Direccionamiento: como se define el dispositivo esclavo.
- Temporización:
  - Según uso:
    - Ciclo completo: el bus es ocupado durante todo el tiempo que duren las fases de direccionamiento y transferencia.
    - Ciclo partido: se divide el tiempo del bus en ranuras(time slots)(slots de petición y de respuesta)
  - Síncronos o asíncronos.
- Modo de operación:
  - Síncrono, temporización síncrona.
  - Asíncrono, con temporización síncrona o asíncrona.
- Estrategia de control: El bus es un elemento pasivo y en cada momento solo debe existir un maestro, pero pueden existir varios maestros potenciales.

Función: de uso general o específico.

Longitud: máxima longitud soportada.

Capacidad de conexión: número de dispositivos que se pueden conectar al bus.

5.2 Estado de alta impedancia

Las puertas tri-estado, además de valores 0 y 1, permiten un estado de alta impedancia. Esta propiedad va a permitir compartir las líneas de los buses entre varios dispositivos.

5.3 Multiplexión y demultiplexión

El bus puede ser paralelo, multiplexado o serie. El paralelo tiene un ancho de palabras que coincide con el ancho de la información a transmitir.

En los multiplexados se emplean los mismos hilos para enviar distintos tipos de información, añadiendo señales de control que permiten identificar el tipo de información que circula por el bus. La multiplexión de las líneas de datos o dirección permite aumentar el espacio de direccionamiento y el ancho de banda.

5.4 Bus de ciclo completo y partido

Ciclo completo:

El bus está ocupado durante todo el tiempo que duren las fases de direccionamiento y transferencia.

El bus puede estar ocupado durante mucho tiempo esperando a que un dispositivo responda.

//IMAGEN

Ciclo partido:

También llamado de conmutación de paquetes.

Se divide el tiempo del bus en ranuras (time slots) de esta manera el bus puede compartir varias transacciones entre diferentes dispositivos.

Un maestro manda un comando a un dispositivo y en vez de esperar a que este responda en el siguiente slot manda un comando para otro.

Los slots suelen ser de tamaño fijo, establecidos de forma síncrona, de ahí la necesidad de una señal de reloj. La duración viene dada por las características del bus.

Es complejo de gestionar.

//IMAGEN

5.5 Transferencias síncronas y asíncronas

Para que la transferencia de datos sea correcta debe establecerse un protocolo que coordine los eventos que suceden en le bus. Puede realizarse:

Bus síncrono:

De fácil implementación pero con poca flexibilidad a la hora de conectar dispositivos con varias velocidades.

Todo ocurre comandado por una señal de reloj.

Debe evitarse el clock skew(desfase del ciclo de reloj) mediante el uso de buses cortos.

Bus asíncrono:

No existe señal de reloj, con lo que no hay clock skew.

Más flexible en cuanto a las velocidades de los dispositivos que puede conectar.

Más lento y complejo de implementar debido a la necesidad de un protocolo de sincronización entre maestro y esclavo.

Bus semisíncrono:

Como un bus síncrono pero añade la señal WAIT.

Esta nueva señal permite ajustar el protocolo a la velocidad de cada dispositivo, añadiendo ciclos de espera.

Puede combinar transferencias síncronas y asíncronas, tomando las ventajas de ambos, velocidad y flexibilidad.

5.6 Arbitraje del bus

Cuando no se tiene un bus de uso exclusivo hay que garantizar que en todo momento solamente una unidad acceda al bus, para ello se emplean los protocolos de arbitraje.

Los protocolos de arbitraje organizan el uso compartido del bus en orden al estable

Tema 6: *Sistemas de E/S*

1. Introducción

La comunicación entre la CPU y el dispositivo se hace a través de un controlador. Este tiene una parte mecánica, electrónica o eléctrica con la que gobierna el dispositivo. También tiene un parte electrónica con la que se comunica con la CPU mediante unos registros(de control, datos y estado). La conexión puede darse también a través de un bus estándar. En ese caso hacen falta dos controladores: el del dispositivo y un adaptador de bus.

Para comunicarse con los controladores, la CPU tiene ciclos máquina especiales de entrada/salida, muy similares a los de lectura/escritura en memoria.

Para direccionar un dispositivo de los varios que pueden estar conectadores a los buses de la CPU por el bus de direcciones sale la dirección de puerto.

Cada controlador de dispositivo conectado al bus del sistema se diseña para que responda a determinados puertos, generalmente conectados a sus registros internos, formando el mapa de puertos del ordenador.

Las técnicas usadas para realizar la entrada/salida deben poder tratar con las características especiales de los dispositivos:

- Suelen ser más lentos que la CPU
- Pueden requerir atención inmediata
- Presentan un comportamiento asíncrono

2. Entrada/salida programada

Se basa en la técnica del sondeo (polling).

Para dispositivos de entrada, la CPU pregunta continuamente al dispositivo si tiene datos para ser leídos.

Para dispositivos de salida, la CPU pregunta continuamente al dispositivo si está listo para recibir nuevos datos.

Desventajas de la E/S programada:

- La CPU está ocupada sin hacer nada productivo (estado de espera ocupada).
- Si hay muchos dispositivos, hay que preguntar a todos en orden y puede tardar en atenderse las peticiones.

3. Entrada/salida dirigida por interrupciones

Cuando el dispositivo está listo, se genera una interrupción.

En las operaciones de entrada, el programa realiza la petición de entrada y, en caso de que el dispositivo no pueda satisfacerla, se cede la CPU a otro programa.

En las operaciones de salida, el programa realiza la petición de salida de datos, que queda almacenada en la memoria.

- El programa u otros programas, pueden continuar su ejecución.
- Cuando el dispositivo esté listo, se produce una interrupción y se escribe en él.

Ventajas:

- Permiten a la CPU realizar otras tareas mientras el dispositivo está ocupado.
- Permiten a los dispositivos requerir atención de un modo asíncrono
- En las CPUs modernas, los accesos a los dispositivos solo se pueden hacer en modo supervisión y se pasa a él a petición del dispositivo mediante una interrupción.

4. Acceso directo a memoria

Si queremos leer dato desde un dispositivo y almacenarlo en la memoria, el modo tradicional consisten leer los datos al acumulador y, posteriormente, escribir el contenido del acumulador en la memoria.

El DMA es un mecanismo que saca partido de las actuales arquitecturas de ordenadores para aumentar el rendimiento. Una línea no puede ser compartida para transmitir varias señales a la vez. Es por ello que, mientras la CPU usa los buses, no es posible cualquier otra transmisión.

//CPU leyendo o escribiendo en memoria

Sin embargo, cuando la CPU no usa los buses (deja sus conexiones a ellos en altas impedancias), un dispositivo podría aprovechar para comunicarse directamente con la memoria.

//CPU haciendo cálculos

Claves:

- Electrónica complicada
- Aumento de velocidad
- Permite posibilidades adicionales: transferencia memoria-memoria, búsquedas, etc.
- La CPU ordena al controlador de DMA una operación y el controlador genera una interrupción cuando está realizada.
- El controlador de DMA debe ser capaz de tomar el control de los buses y realizar las transferencias cuando la CPU los libera.

Se usa para dispositivos que requieren velocidades de transferencia alta y gran volumen de datos.

Existen dos maneras básicas de hace DMA:

- Memoria multipuerta: si la memoria tiene varias puertas, se dedica una a la CPU y el resto a DMA. Es caro.
- Robo de ciclo: es el caso habitual en que los buses son únicos y la CPU y el controlador de DMA compiten por ellos. Es más barato, pero hace que la duración de las instrucciones de la CPU ya no sea exacta.

Tipos de DMA:

- Simple: cada vez que se roba un ciclo se transfiere un único byte
- En ráfagas: al obtener los buses, se puede transferir un bloque completo
- Disperso: los bloques de datos no tienen por qué estar contiguos en la memoria.
- Gobernado por memoria: las órdenes de la CPU al controlador no se mandan por puertos de entrada/salida, sino que se escriben en la memoria.

5. Memoria mapeada

El 6809 no dispone de instrucciones específicas de entrada/salida. Para leer/escribir en los registros de un controlador, el 6809 realiza un acceso a memoria normal a una determinada dirección.

El dispositivo, si ve que el acceso se realiza a la dirección que tiene asignada, actualiza sus registros.

Así como un ordenador con E/S normal tiene un mapa de memoria y un mapa de puertos, uno con E/S por memoria mapeada dispone de un único mapa de memoria.

Posibles preguntas

TEMA-1

▼ Arquitectura von Neumann vs Harvard

En la arquitectura de von Neumann los datos e instrucciones se almacenan en binario en memoria y no se hace distinciones entre estos. Nos encontramos al procesador, memoria y unidad de E/S como sus partes. La arquitectura de Harvard tiene dos tipos de memoria, memoria de instrucciones y memoria de datos, esto tiene como ventaja el aumento de la velocidad de acceso a la memoria pero como inconveniente que puede haber memoria de sobra de un tipo y faltar del otro.

▼ Evolución del rendimiento

- Primera generación: Tubos de vacío
  - Los mismos que las diseñaban las construían, programaban, usaban y mantenían.
  - No había lenguajes de programación, esta podía a llegar a ser electrónica.
  - La salida podían ser luces en un panel.
  - Los fallos de válvula eran habituales.
- Segunda generación: Transistores
  - El transistor es más barato, pequeño, fiable y consume menos energía.
  - Usa memorias de ferrita. Aun así no son suficientemente rápidas para los procesadores CISC.
- Tercera generación: Circuitos integrados
  - Mundo de la banca y mundo de la ciencia.
  - Se empieza a hablar de compatibilidad ascendente. Los circuitos integrados abaratan los costes.
- Cuarta generación: Ordenadores personales

    - El microprocesador incluye todos los componentes de la CPU en un único chip.
    - El ordenador tiene que hacerse amigable.
    - Comienza la era del PC de IBM, los MACs.
    - Se popularizan las redes de ordenadores.
    - Las generaciones a partir de esta gozan de menor consenso.
- Quinta generación: Alta integración
    - La miniaturización hace procesadores más rápidos, pero se calientan más.
    - La tecnología del procesador evoluciona más rápidamente que la de la memoria y disco duro.
    - Se deben usar técnicas para mantener al procesador ocupado al máximo.
- Sexta generación: Ultra Alta Integración
    - Se agudiza el desequilibrio entre rapidez de proceso y rapidez de memoria.
    - Se alcanza un límite en la velocidad del reloj del procesador y se tiende al multiprocesador.
    - Triunfo de las arquitecturas masivamente paralelas de las tarjetas gráficas, incluso para aplicaciones de cálculo.

▼ Clasificación paralelismo

- SISD: Single Instruction, Single Data
    - Se corresponde con la arquitectura de von Neumann
    - La unidad de control toma una instrucción cada vez
    - La instrucción se ejecuta sobre un dato, a lo sumo
- SIMD: Single Instruction, Multiple Data
    - La unidad de control toma una instrucción cada vez
    - Las unidades de proceso, todas a la vez, realizan la ejecución cada una sobre un dato diferente.

    - Típico de GPUs.
    - Aumento espectacular del rendimiento para aplicaciones de naturaleza vectorial.
- MISD: Multiple Instruction, Single Data
- MIMD: Multiple Instruction, Multiple Data
    - Cada unidad toma y ejecuta sus propias instrucciones sobre sus propios datos
    - Se sincronizan ocasionalmente
    - Son los multiprocesadores

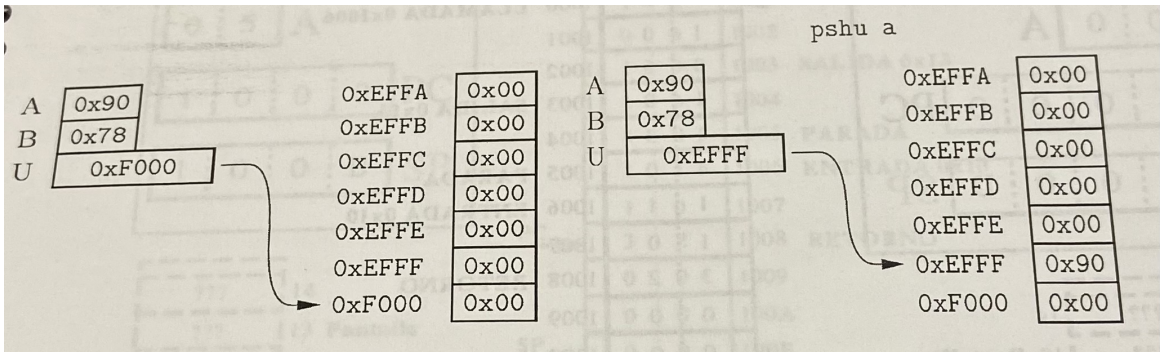
TEMA-2

▼ Puntero pila

Una pila es una estructura de datos con dos operaciones *push* (meter datos) y *pop* (sacar datos), que sigue un protocolo LIFO(*Last In, First Out*).

El funcionamiento es el siguiente: El puntero apunta a la dirección del ultimo dato que metió, la pila crece hacia direcciones bajas, es decir, si se mete algo el puntero decrementa, si se saca algo, el puntero se incrementa.

Uno de los usos de la pila es guardar temporalmente el contenido de los registros mientras se usan para otra cosa. En el caso de que se guarden varios registros, hay que restaurarlos en el sentido inverso en que se metieron.



▼ Registro flags y código

Comparación	con signo	sin signo
Mayor que	bgt	bhi
Mayor o igual que	bge	bhs

Igual que	beq	beq
Diferente de	bne	bne
Menor o igual que	ble	bls
Menor que	blt	blo

▼ Direccionamiento

- Direccionamiento inherente o implícito:

  Los datos que necesita la instrucción no están especificados en la propia instrucción, bien porque no hay datos, bien porque la localización es única.

- Direccionamiento inmediato:

  Los datos aparecen en el propio código máquina de la instrucción.

- Direccionamiento directo:

  En la instrucción aparece el modo de localizar los datos directamente, bien dando una dirección de memoria, bien especificando uno de los registros.

- Direccionamiento indirecto:

  Funciona como el direccionamiento directo, pero el dato que se obtiene es, en realidad, una dirección de memoria donde se encuentra el dato definitivo.

- Direccionamiento indizado directo:

  Su objetivo principal es manejar arrays en ensamblador. Cuando queremos acceder a un elemento en concreto del array a la dirección donde comienza en array se le denomina dirección base y al número de bytes que hay desde el inicio hasta el elemento que queremos acceder, se le denomina desplazamiento.

- Direccionamiento indizado indirecto:

  Funciona como combinación de los direccionamientos indizados vistos en el apartado anterior y el direccionamiento indirecto.

- Direccionamiento indizado al pc:

  Hacer indización sobre el PC tiene un curioso efecto secundario: el código máquina se vuelve directamente ***reubicable***, es decir, se puede cargar en cualquier lugar sin cambiarlo.

▼ Interrupciones

Las interrupciones son mecanismos asíncronos por los que se interrumpe la ejecución normal de un programa para llamar a un procedimiento de servicio a un suceso. Cuando una interrupción ha sido solicitada pero no se atiende se dice que está enmascarada, la petición se guarda y se atiende cuando la interrupción se desenmascara.

Pasos que se produce cuando se recibe y se atiende una interrupción: La CPU acaba la ejecución de la instrucción en curso. Luego se salta automáticamente a la dirección del procedimiento de servicio de la interrupción como si de otro procedimiento se tratase.

Una interrupción de software, al contrario de las interrupciones habituales, es provocada por la ejecución de una instrucción concreta del código máquina. Sus rutinas de servicio suelen apuntar a código del so.

▼ Fases de ejecución

Fases en la ejecución de una instrucción de código máquina:

- Toma de instrucción (IF): se hace uno o varios ciclos máquina de lectura de memoria, dependiendo de la longitud del código de operación de la instrucción. El PC sale al bus de direcciones y se va incrementando.
- Decodificación (ID): el procesador identifica la instrucción leída y se dispone a efectuar los pasos para ejecutarla.
- Toma de operandos (OF): dependiendo de la instrucción puede ser necesario ejecutar uno o varios ciclos maquina para buscar operandos.
- Ejecución (EX): al principio de esta fase, el PC ya está apuntando a la siguiente instrucción. Si el cálculo es complicado, puede llevar varios ciclos máquina su conclusión.
- Escritura del resultado (WB): si la instrucción calcula o mueve un dato, se escribe el resultado en el destino.
- Interrupciones (II): si se ha activado una señal de interrupción, es ahora cuando se atiende.

TEMA-3

▼ Mapa de memoria

El mapa de memoria indica como está distribuida la memoria física y como será direccionada por el microprocesador. Los microprocesadores tiene bus de

direcciones (AB), datos (DB) y control (CB). El espacio de direccionamiento y su organización depende del número de AB,DB y CB.

- Un bus de direcciones de n líneas identifica 2^n direcciones físicas diferentes.
- Con un bus de control de m líneas se podría asignar m direcciones lógicas a una física.
- Con un bus de datos de 8 bits se accedería a un byte de cada acceso.
- Esto da como resultado: $2^n \times m$ dicho de otra forma $2^{\mathrm{AB}} \times \mathrm{CB}$

▼ Jerarquía de memoria

Nivel 0: Registros. Es una memoria de alta velocidad y poca capacidad integrada en el microprocesador La capacidad de los registros viene dado por la arquitectura del procesador.

Nivel 1: Memoria caché. Es una memoria intermedia entre los registros y la memoria principal que se emplea para almacenar información (datos + programas) de frecuente uso reduciendo los tiempos de acceso. Suele estar realizada con tecnología SRAM(RAM estática), más rápida y cara que la DRAM. Según su velocidad y capacidad están organizadas en L1, L2 y L3.

Nivel 2: Memoria principal. Esta memoria suele ser memoria RAM (DRAM o SDRAM) y su tamaño es del orden de los Gigabytes.

Nivel 3: Almacenamiento secundario. Cuando se acaba la memoria RAM es posible emplear el disco duro como memoria virtual. Además permite guardar información de forma permanente. Como almacenamiento secundario podemos tener discos duros SSD, HDD, memorias flash, etc. Su tamaño es del orden de Gigabytes a Terabytes.

TEMA-4

▼ Unidad de control funciones

- Control de la ejecución de los programas:
  - Lectura de memoria principal de la instrucción apuntada por PC.
  - Incremento del PC.
  - Decodificación de la instrucción leída.

    - Ejecución de la instrucción.
- Debe resolver situaciones anómalas o de conflicto.
- Debe controlar la comunicación con los periféricos.

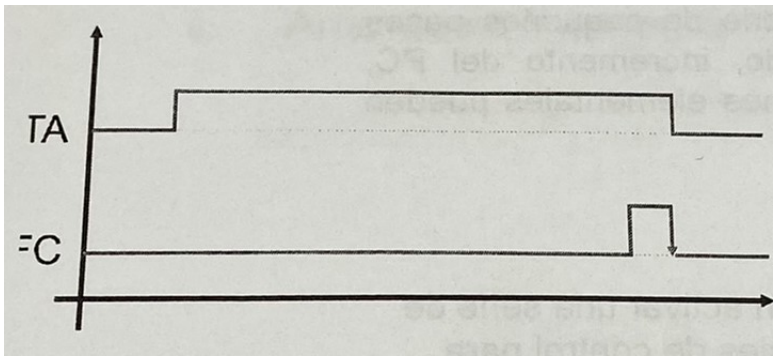
▼ Unidad de control operaciones elementales

Operaciones de transferencia

Necesitan dos elementos de almacenamiento, origen y destino. Entre estos debe establecerse un camino físico. Una vez establecidos los elementos de almacenamiento y su interconexión física se deben de activar alguna señal que indique al destino que tome lo que tiene en su entrada, el origen queda inalterado.

Transferencia de $\mathrm{A} \rightarrow \mathrm{C}$

- Se selecciona el camino físico.
    - Se activa TA para activar la salida del registro A
    - Es un bus triestado, TA estará activa todo el tiempo que queramos que el contenido de A esté en el bus.
- Después de un tiempo de estabilización el contenido de A está en la entrada de los registros.
- Se activa la señal FC, flanco de bajada para carga del registro C.



Operaciones de proceso

La información de origen se transforma al pasar por un operador combinacional en su camino al destino.

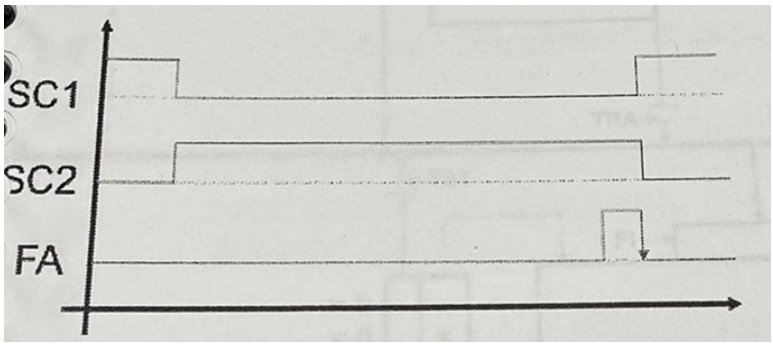
Las operaciones de proceso pueden ser:

- Con dos operadores, diádicas, con dos orígenes y un destino (suma).

- Con un operador, monódica, un solo origen y un destino (desplazamiento).
- El operador puede generar además del resultado una serie de información de estado(flags).

Proceso: $\mathrm{A} \leftarrow \mathrm{A}+\mathrm{D}$

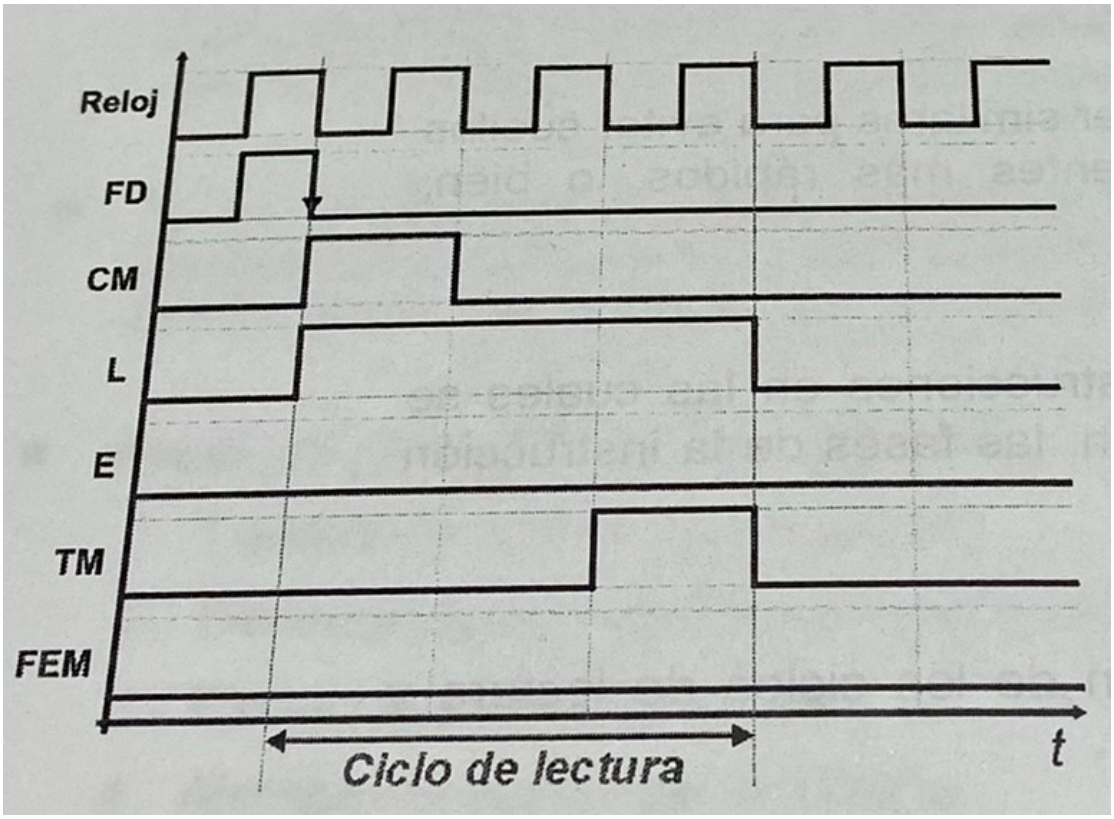
- Se activan las señales SC1 y SC2 para seleccionar los operandos (reg A y D).
- Se espera un tiempo a que se realice la operación.
- Se activa la señal FA, flanco de bajada para carga del registro A.



▼ Ciclos de lectura y escritura

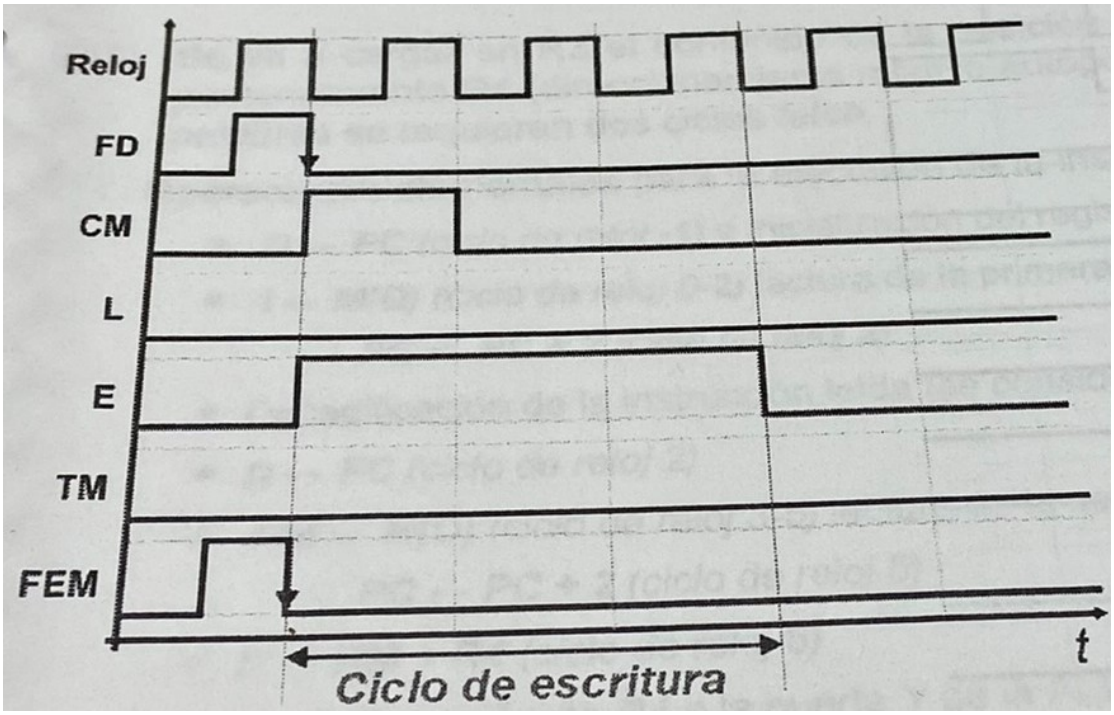
Ciclo de lectura

- Antes de iniciarse el ciclo de lectura se carga la dirección en el registro *D*, con el contenido del bus de direcciones, mediante el flanco de bajada de la señal *FD*.
- Mediante la activación de la señal *CM* se indica a la memoria que empieza un *ciclo de memoria*. En el mismo periodo se activa la señal *L* para indicar a la memoria que se va a realizar una *lectura*.
- La memoria localiza el dato y lo pone en el buffer de salida.
- En el último periodo la memoria activa la señal *TM*, que hará que mientras esté activa la memoria a través del triestado vuelque el dato solicitado en el bus de datos.



Ciclo de escritura

- Antes de iniciarse el ciclo de lectura se carga la dirección en el registro D, con el contenido del bus de direcciones, mediante el flanco de bajada de la señal *FD.* También se carga el dato a escribir en el registro *RM*, con el contenido del bus de datos, mediante el flanco de bajada de la señal *FEM.*
- Mediante la activación de la señal *CM* se indica a la memoria que empieza un *ciclo de memoria*. En el mismo periodo se activa la señal *E* para indicar a la memoria que se va a realizar una *escritura*.
- La memoria recoge el dato del registro *RM* y lo almacena en la dirección indicada en el registro D.



TEMA-5

▼ Buses

Un bus es una vía de comunicación que conecta dos o más dispositivos. La principal característica de los buses es que son medios de comunicación compartidos.

En las transferencias de información que se realizan en los buses existen dos agentes involucrados:

- Maestro: el que origina la transferencia.
- Esclavo: el que responde a la transferencia.

Características:

- Funcionamiento en banda base.
- Paralelismo: bus serie, paralelo y multiplexado.
- Direccionamiento: como se define el dispositivo esclavo.
- Temporización:
  - Según uso:
    - Ciclo completo: el bus es ocupado durante todo el tiempo que duren las fases de direccionamiento y transferencia.

        - Ciclo partido: se divide el tiempo del bus en ranuras(para petición y para respuesta).
    - Síncrono (reloj único que determina el fin de los ciclos o ranuras) o asíncrono, establecida por los dispositivos.
- Modo de operación:
    - Síncrono, con temporización síncrona.
    - Asíncrono, con temporización síncrona (buses semi-síncronos) o asíncrona.
- Estrategia de control: el bus es un elemento pasivo y en cada momento solo debe existir un maestro, pero pueden existir varios maestros potenciales.
- Función: de uso general o específico.
- Capacidad de conexión.
- Tipo de ciclo.
- Tipo de lógica

▼ Tipos de buses

Un bus puede ser paralelo, multiplexado o serie:

- Paralelo: tiene un ancho de palabras que coincide con el ancho de información a transmitir.
- Multiplexado: Se emplean los mismos hilos para enviar distintos tipos de información, añadiendo señales de control que permiten identificar el tipo de información que circula por el bus. La multiplexión de las líneas de direcciones permite aumentar el espacio de direccionamiento(si se multiplexan las líneas de dirección), y aumentar el ancho de banda (si se multiplexan las líneas de datos).
- Serie: Variante del multiplexado en el cual no se emplean señales para demultiplexar los bits, se codifica la información en forma de onda en el emisor y en el receptor se decodifica.

▼ Ciclos de buses

Desde el punto de vista de temporización el bus puede ser:

- Ciclo completo: El bus es ocupado durante todo el tiempo que duren las fases de direccionamiento y transferencia. Puede estar ocupado mucho tiempo esperando a que el dispositivo responda.
- Ciclo partido: Se divide el bus en slots, de esta manera el bus puede compartir varias transacciones entre diferentes dispositivos.
    - Un maestro manda un comando a un dispositivo y en vez de esperar a que este responda en el siguiente slot manda un comando para otro.
    - Los slots suelen ser de tamaño fijo, establecidos de forma síncrona, de ahí la necesidad de una señal de reloj. La duración viene dada por las características del bus.
    - Complejo de gestionar.

▼ Daisy chaining

- De dos señales:

  Emplea dos líneas, una para petición (*REQUEST*) y otra para la concesión (*GRANT*).

    - Funcionamiento:
        - El maestro que desea el control activa *REQUEST*.
        - El resto de maestros lo propagan a no ser que estén utilizando el bus. Si es así bloquean la propagación hasta finalizar su uso.
        - El árbitro activa el *GRANT*.
        - Si un maestro no ha solicitado el bus propaga el *GRANT*.
        - Si tiene una petición pendiente toma el control del bus.
    - De esta manera la prioridad viene determinada por la proximidad al árbitro
- De tres señales:

  Añade otra línea para indicar el estado de bus ocupado (BUS BUSY). El funcionamiento difiere en que cuándo un maestro toma el control activa el BUS BUSY, de tal manera que el árbitro sólo activa GRANT si BUS BUSY está desactivado, y un máster solo toma el control si esta línea está inactiva.
- De cuatro señales:

Añade una señal BUS ACK, que es activada por el máster que solicitó el bus en respuesta al GRANT, cuando el bus está ocupado. Cuando está activada el árbitro queda inhibido.

▼ Señales de control

- FRAME: La maneja el maestro para indicar el inicio y la duración de una transacción. Cuando se activa indica el inicio, y cuando se desactiva que la transacción ha finalizado.
- IRDY: Indica que el maestro está preparado, durante una lectura indica que el maestro está preparado para recibir datos y durante una escritura que hay datos en el AD.
- TRDY: Indica que el esclavo está preparado.
- STOP: El esclavo indica al maestro que desea parar la transacción en curso.
- LOCK: Indica una operación atómica que puede durar varios ciclos de reloj para completarse.
- IDSEL: Empleado como chip select en transacciones de configuración.
- DEVSEL: Activada por el esclavo cuando reconoce su dirección.

TEMA-6

▼ DMA

El DMA es un mecanismo por el cuál la transferencia de datos entre memoria y dispositivos se realiza sin pasar por los registros de la CPU, usando los buses cuando la CPU no los esté usando.

Existen dos maneras básicas de hacer DMA:

- Memoria multipuerta: si la memoria tiene varias puertas, se dedica una a la CPU y el resto a DMA. Es caro.
- Robo de ciclo: es el caso habitual en que los buses son únicos y la CPU y el controlador DMA compiten por ellos. Es más barato, pero hace que la duración de las instrucciones de la CPU ya no sea exacta.

Tipos de DMA:

- Simple: cada vez que se roba un ciclo se transfiere un único byte.
- En ráfagas: al obtener los buses, se puede transferir un bloque completo.

- Disperso: los bloques de datos no tienen por qué estar contiguos en la memoria.
- Gobernado por memoria: las órdenes de la CPU al controlador no se mandan por puertos de E/S, sino que se escriben en memoria.

Opciones:

- Autoincremento o fijo: la dirección origen o destino de la transferencia se incrementa o no automáticamente.
- Reinicio automático: al finalizar la transferencia se inicia otra igual automáticamente.

▼ Memoria mapeada

El 6809 no dispone de instrucciones específicas de entrada/salida. Para leer/escribir en los registros de un controlador realiza un acceso a memoria normal a una determinada dirección.

El dispositivo, si ve que el acceso se realiza a la dirección que tiene asignada, actualiza sus registros.

El programador siente que escribe o lee de la memoria cuando se comunica con los dispositivos.

Así como un ordenador con entrada/salida normal tiene un mapa de memoria y un mapa de puertos, uno con entrada/salida por memoria mapeada dispone de un único mapa de memoria.