Sistemas de Procesamiento de Datos

Índices:

UNIDAD 3: Sistemas Numéricos – Boole

Unidades de medida

Nibble: 4 bits.

Byte: 8 bits.

Kilobyte: 1024 bytes.

Word: 16 bits.

Los registros del procesador 8086 son de 16 bits y sigue la progresión estándar en la arquitectura x86, por lo tanto:

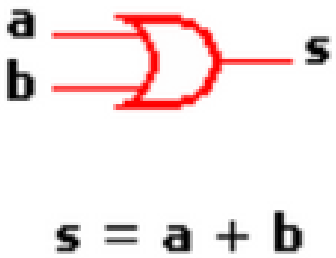
DoubleWord: 32 bits.

QuadWord: 64 bits.

Algebra de Boole y compuertas lógicas

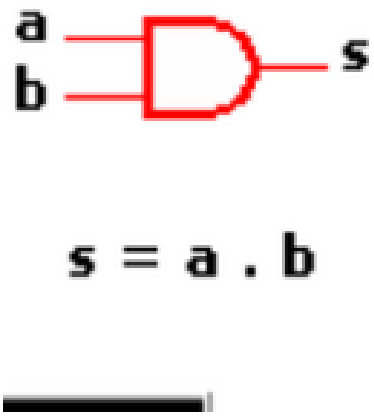
Sean A y B variables binarias:

- OR: A+1 = 1 y A+0 = A



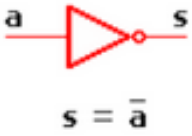
a	b	s
0	0	0
0	1	1
1	0	1
1	1	1

- AND: A*1 = A y A*0 = 0



a	b	s
0	0	0
0	1	0
1	0	0
1	1	1

- NOT: Si A = 1; ¬A = 0. Si A= 0; ¬A = 1



a	s
0	1
1	0

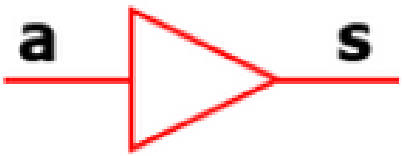
- LEY DE MORGAN: ¬(A+B) = ¬A*¬B y ¬(A*B) = ¬A+¬B

- XOR:

$s = a.\bar{b} + \bar{a}.b$

a	b	s
0	0	0
0	1	1
1	0	1
1	1	0

- BUFFER:

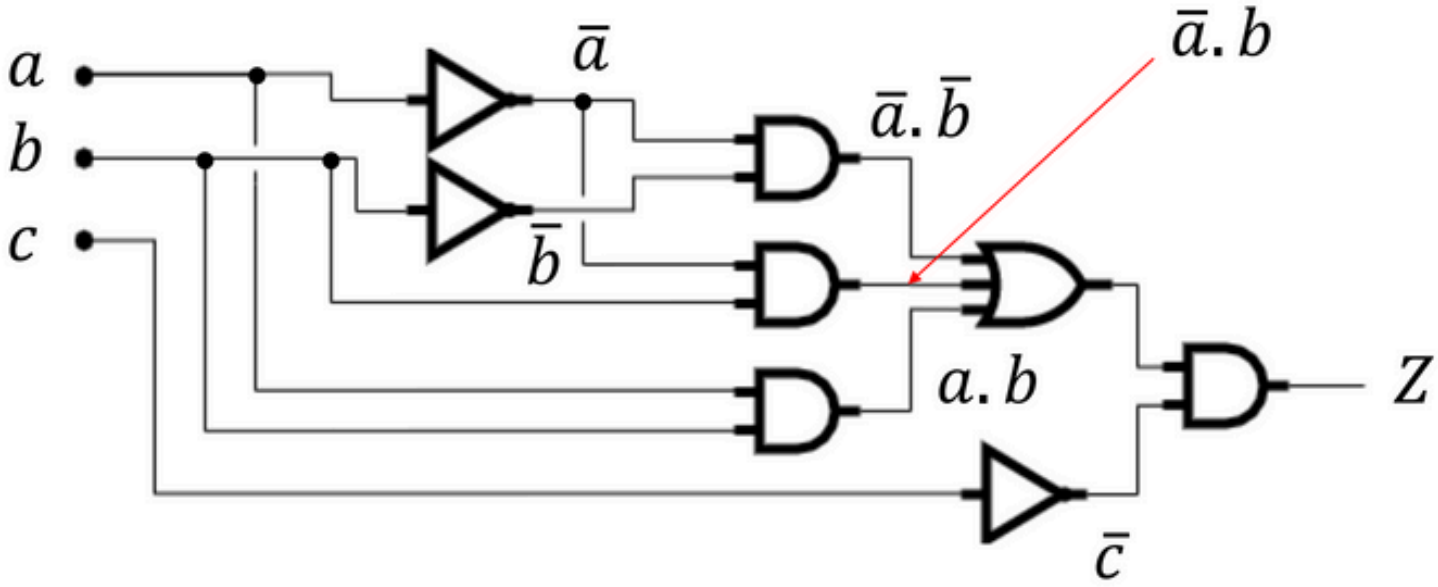


$s = a$

a	s
0	0
1	1

Si, el buffer tiene una tercera conexión, o patita, llamada “gate” y la señal de entrada de esta es cero, el estado de salida, sin importar el valor de a, será hz, que significa “alta impedancia” y este es el tercer estado de salida que puede existir.

$$Z = f(a,b,c) = \bar{c}.[\bar{a}.\bar{b} + \bar{a}.b + a.b]$$



Activación de compuertas/salidas

Por nivel: Se interpreta un cambio de estado en la compuerta cuando la señal en esta alcanza cierto valor.

Por flanco: Se interpreta un cambio de estado en la compuerta cuando la pendiente de su señal alcanza un determinado valor.

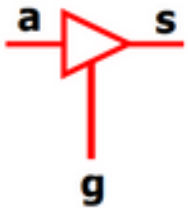
Schmitt-Trigger: Se interpreta un cambio de estado en la compuerta cuando la tensión en su entrada sobrepasa un determinado umbral “A”, provocando que este umbral se posicione en un valor “B” menor que “A”. Se vuelve al anterior estado cuando en la compuerta se atraviesa por debajo del umbral “B”, volviendo este a ser “A”. Se la utiliza para prevenir el ruido (señales eléctricas no deseadas o interferencias) que podría solaparse a la señal original y que causaría falsos cambios de estado si los niveles de referencia y entrada son parecidos.

A masa: Cada entrada digital es conectada a masa, potencial de referencia, o “a tierra” mediante una resistencia de alto valor previniendo una eventual desconexión u olvido y elimina la necesidad de conectar físicamente una resistencia en la plaqueta o circuito impreso en donde se encuentra el dispositivo conectado.

HZ: Es un tercer estado, diferente a 1 y 0 que significa “alta impedancia”; y se puede interpretar como que la salida de la compuerta está virtualmente desconectada.

g	a	b	s = a+b
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	X	X	HZ

X=(No importa)



La señal “g” (gate) coloca a la salida de la compuerta en el estado HZ.

ASCII

Es el American standard code for information; tiene siete bits para codificar cada carácter, formando 128 dígitos y se extiende a ocho bits o un byte poniendo el bit más significativo en 0.

Código de paridad

Cualquier combinación binaria puede convertirse en una combinación de paridad par o impar, agregando un bit más que cumpla con una condición de paridad deseada. Los códigos de paridad se los utiliza para detectar errores de transmisión de datos; si se transmite una combinación de paridad par y se recibe una

de paridad impar, se puede asegurar que hay error. Es posible recibir un error habiendo transmitido par y recibir par, pero la tasa de error en más de 1 bit es muy baja.

La detección de errores se ejecuta mediante una compuerta XOR.

Transcodificadores

Codificador: Tiene una entrada no codificada y una salida codificada.

Decodificador: Tiene una entrada codificada y una salida no codificada.

Conversor de código: Tiene una entrada codificada y una salida también codificada. (La entrada es a y sale B).

Multiplexores y demultiplexores

Multiplexor: Toma múltiples entradas y, a través de una llave rotativa controlada por un código binario, selecciona una sola para enviarla a la salida.

Demultiplexor: Toma una única entrada y, a través de una llave rotativa controlada por un código binario, la distribuye a una de varias salidas.

UNIDAD 4: Computador

Estructura de un computador

Los componentes o módulos generales de un computador son:

1. Computador: Acá se almacena la memoria principal, la unidad centrar del proceso (CPU/UPC), tiene entradas y salidas y tiene sistema de interconexión.

1. Memora principal: Tiene la memoria ROM que es únicamente de lectura y la RAM que es de lectura y escritura.

1. Estructura CPU: Es la interconexión interna de la CPU, tiene registros (REG), la unidad de control (CU) y la unidad aritmética lógica (ALU).

1. Estructura UC: Tiene lógica secuencial, unidad de control de registros y decodificadores y memoria de control.

Funciones principales de un computador

Una función es una operación que realiza cada uno de los componentes como parte de una estructura organizada.

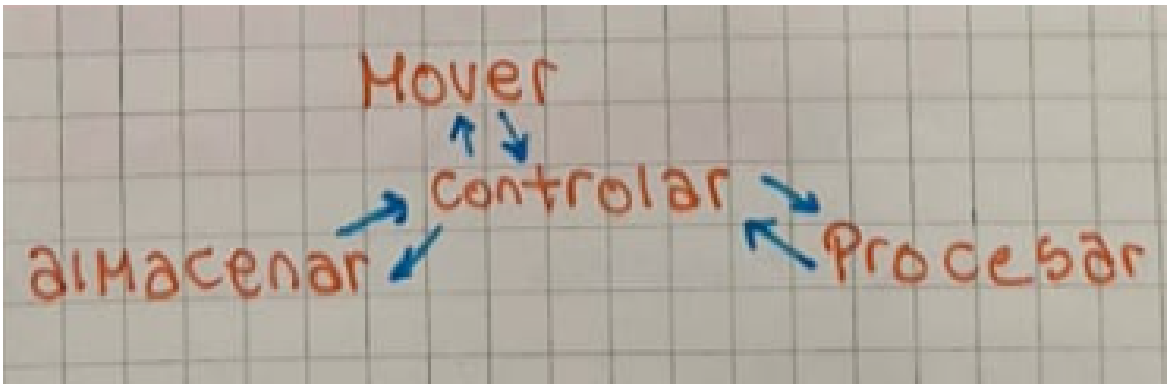
Procesamiento de datos: Se realiza a través de la Unidad Central de Procesamiento (CPU). La CPU interpreta y ejecuta instrucciones almacenadas en memoria.

Está compuesta por dos unidades principales: la Unidad de Control que dirige la ejecución de instrucciones y la Unidad Aritmético-Lógica (ALU) que ejecuta operaciones matemáticas y lógicas.

Almacenamiento: El computador almacena tanto instrucciones como datos. Se utilizan dos tipos principales de memoria: la RAM (memoria de acceso aleatorio), que es volátil, almacena datos temporales y programas en ejecución, y la ROM (memoria de solo lectura) que no es no volátil y contiene rutinas básicas para el arranque del sistema.

Mover: es la transferencia de datos entre diferentes partes del sistema: se la memoria a los registros internos del CPU, de un registro a otro, desde un dispositivo de entrada a la memoria, hacia un dispositivo de salida y de memoria a memoria. El procesamiento de datos no puede darse si esos datos no son primero movidos al lugar correcto.

Controlar: Es la coordinación de todas las operaciones internas mediante señales de control; estas señales viajan a través del bus de control y sincronizan los componentes del sistema.



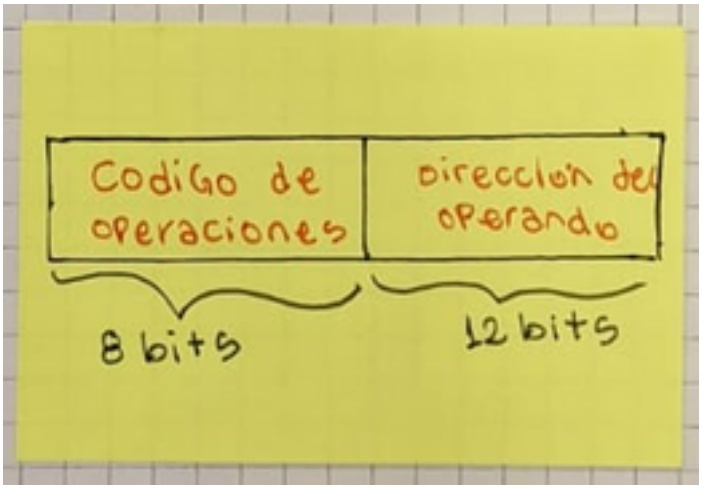
Arquitectura

Genéricamente, todos los computadores tendrán una estructura similar a la mencionada (CPU, memoria, sistema de interconexión) en una cantidad adecuada a la capacidad de procesamiento requerida.

Arquitectura Von Neumann:

- El único espacio de memoria es donde se alojan tanto las instrucciones de un programa como sus datos. La conexión entre la unidad control de proceso y el resto del ordenador se hace mediante un bus compuesto por tres grupos de señales o buses.

La longitud de palabra es de 40 bits; dos instrucciones 20 bits:



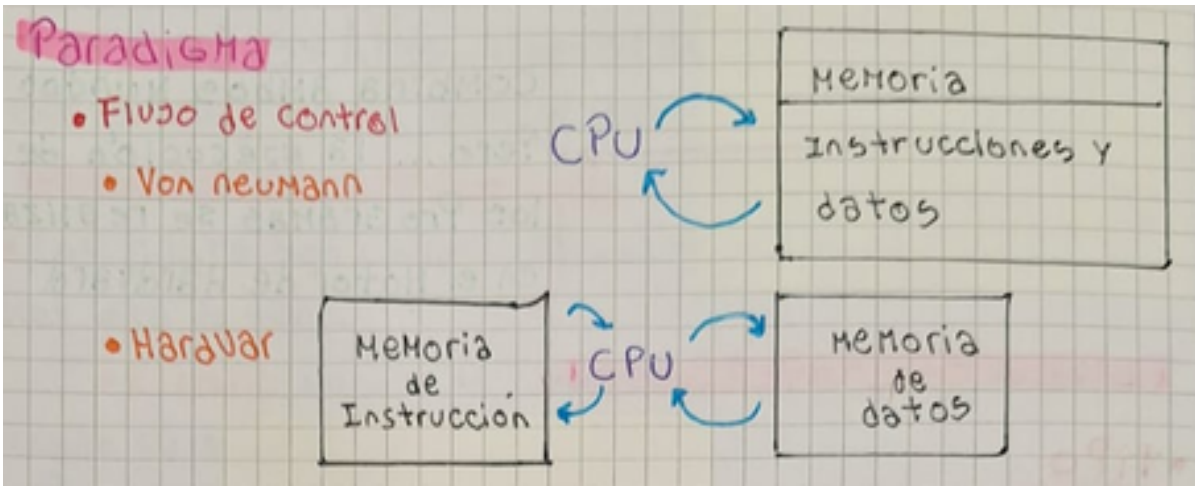
La arquitectura Von Neuman es hacia el exterior de la CPU, interactuando con un solo espacio de memoria.

Arquitectura Harvard:

- Posee dos espacios de memoria físicamente separados, una para los datos y otra para las instrucciones. En paralelo, accede a ambos espacios aumentando la tasa de transferencia de datos,

posee dos buses diferentes que pueden trabajar en simultaneo.

La arquitectura de Harvard es internamente, subdividiendo datos e instrucciones en dos espacios físicos separados que se alojan en las denominadas memorias caché.



Clasificación de Flynn

Clasifica los flujos de control y de datos; son los parámetros básicos para la clasificación:

1. SISD: La single Instruction Single Data, donde un único procesador ejecuta una única instrucción sobre un único conjunto de datos almacenados en una única memoria. Acá entra Von Newman y Harvard.

1. MIMD: La Multiple Instruction Multiple Data, donde se ejecuta un conjunto diferente de instrucciones con distintos conjuntos de datos.

Familia de computadores

1. Microcontroladores: Es una computadora de un solo chip. El CPU es simple, tiene reloj, un puerto de comunicación, memoria, etc.

1. Microcomputadoras: Se las puede definir de forma precisa; tienen RAM, discos, CPU, buses y periféricos, un solo chip de CPU, mono y multicore. Un ejemplo son las computadoras.

1. Minicomputadores: Son de mayor costo, más voluminosas, tienen una longitud de palabras más grande, un conjunto de instrucciones más valioso, tienen más CPU, tienen más potencia de procesamiento y son utilizadas tanto en negocios como para el campo científico.

1. Mainframes: Son de mayor capacidad de procesamiento que los minis, sensiblemente más costosas, ocupa grandes espacios y tienen grandes bases de datos, proceso intensivo de uso de CPU.

1. Supercomputadoras: Son exageradamente costosas, se los utiliza para resolver problemas matemáticos de alta complejidad; aerodinámica, sismología, metodología, física atómica, simulaciones, etc.

Arquitectura 2:

La arquitectura es cada uno de los atributos visibles al programador que tiene impacto directo en la ejecución de un programa:

- El conjunto de registros internos: cuatro generales que son AX, BX, CX y DX; cuatro segmentos que son DS, CS, ES y SS; tres punteros que son SP, BP y IP; dos índices que son DI y SI; y una bandera que es F.

- El conjunto de instrucciones: la transferencia de datos, la aritmética, la manipulación de bits, las cadenas, la transferencia de programa y el control del procesador.

- La longitud de palabra: 16 bits y en el bus externo 16 en el 8086 y 8 bit para el 8088.

- Cantidad de bit para representar los datos: para los enteros sin signo son 8 bits del 0 al 255 o de 16 bits del 0 al 65.535 y para los enteros con signo son 8 bits del -128 al 127 o 16 bits del -32.768 al 32.767.

- Mecanismos de direccionamiento de memoria: modos implícitos, inmediatos, directos, indirectos, indirectos de registro, relativo al puntero base e indexado al puntero base.

- Acceso a punteros periféricos: 224 interrupciones y mapeo de direcciones de dispositivos aislado de la memoria principal.

Unidad de Control (estructura interna)

Es una parte fundamental del CPU que se encarga de dirigir y coordinar todo lo que hace el procesador. Chat GPT dijo que es como el "director de orquesta" del sistema.

1. Memoria de control: Se refiere a la parte que guarda o accede a la instrucción actual que se va a decodificar. Puede estar implementada como una pequeña memoria de control o ROM con microinstrucciones, según la arquitectura.

1. Unidad de Control de Registros y Decodificadores: Encargada de interpretar la instrucción y mover/controlar los registros necesarios. Básicamente, decodifica y organiza la ejecución.

1. Lógica secuencial: Se refiere a la lógica que sigue una secuencia de pasos controlada por reloj, generando las señales correctas en el orden adecuado. Esto es lo que mantiene el ritmo del procesador.

Elementos estructurales de un computador

1. CPU (Unidad Central de Proceso): Ejecuta las instrucciones de los programas y coordina el funcionamiento del sistema. Dentro del CPU está la Unidad de Control, que interpreta y dirige la ejecución de instrucciones, la Unidad Aritmético-Lógica (ALU), que realiza operaciones matemáticas y lógicas, los Registros, internos que son pequeñas memorias rápidas usadas para almacenamiento temporal de datos e instrucciones, y el Contador de Programa (PC) que señala la dirección de la próxima instrucción a ejecutar.

1. Memoria principal: Es donde se almacenan datos y programas que se están ejecutando y se divide en RAM (memoria de acceso aleatorio), que es volátil, es de lectura y escritura y se borra al apagar, y la ROM (memoria de solo lectura) que no es volátil y contiene rutinas básicas como el arranque. "La RAM es usada por el sistema operativo y los programas del usuario" y "La ROM es usada para almacenar rutinas de bajo nivel".

1. Buses: Son los conductores que interconectan todos los componentes del sistema; bus de datos, que lleva los datos entre CPU, memoria y periféricos, bus de direcciones que indica dónde están los datos, y bus de control lleva señales de control que dicen qué hacer (leer, escribir, etc.).

1. Dispositivos de Entrada/Salida: Permiten al computador interactuar con el mundo exterior. "Son los encargados de enviar y/o recoger información del mundo externo a la computadora".

UNIDAD 5: ALU - Números Reales Con/Sin Signo, Punto Flotante

Celda sumadora completa de un bit

Es un circuito que suma dos bits, A y B, junto con un acarreo de entrada, Cy-1, proveniente de una celda anterior. Entrega dos salidas: S, que es el bit de resultado de la suma, y Cy, el acarreo de salida.

Clasificación de Flynn

Es una clasificación de arquitecturas de computadores basada en la cantidad de flujos de datos e instrucciones que pueden procesar al mismo tiempo. Los cuatro modelos son:

- SISD (Single Instruction, Single Data): Un solo procesador ejecuta una instrucción sobre un único dato. Un ejemplo es el CPU común porque Procesa una sola instrucción sobre un dato a la vez. Es el caso típico del 8086.

- SIMD (Single Instruction, Multiple Data): Una misma instrucción se ejecuta en varios datos simultáneamente (paralelismo de datos). Un ejemplo es el CPU vectorial porque ejecuta la misma instrucción sobre múltiples datos (usado en procesamiento gráfico o científico).

- MISD (Multiple Instruction, Single Data): Varias instrucciones actúan sobre un mismo dato (raro en la práctica).

- MIMD (Multiple Instruction, Multiple Data): Varios procesadores ejecutan instrucciones diferentes sobre datos distintos (muy común en sistemas distribuidos y multinúcleo). Un ejemplo son los procesadores multinúcleo, porque cada núcleo puede ejecutar su propia instrucción en sus propios datos, y los sistemas distribuidos, porque cada computador o nodo ejecuta tareas distintas sobre datos diferentes.

Registro de Estado (FLAGS)

Es un conjunto de banderas (bits) que reflejan el resultado de las operaciones realizadas por el microprocesador. Dos de las más importantes son:

- Carry Flag (CF): se activa, o almacena 1, cuando una suma genera un acarreo fuera del bit más significativo o cuando falta un bit al realizar una resta, es decir, ocurre un acarreo.
- Overflow Flag (OF): se activa, o almacena 1, cuando el resultado de una operación aritmética con números con signo es tan grande o pequeño que no cabe en el rango permitido.

El flag de acarreo CF no se debe considerar como indicador de overflow para operaciones de números con signo, el flag de acarreo CF es indicador de overflow para operaciones de números sin signo, la resta de dos binarios positivos nunca resultará en overflow y Una suma de un binario positivo más uno negativo nunca resultará en overflow.

En una operación de resta, existe un Borrow o "pedir prestado" en el bit más significativo. Esto se visualiza como un "1" en el flag de Carry del procesador. Si la resta se realiza con binarios enteros sin signo, ¿Cómo debe interpretarse? Como overflow.

Números enteros SIN/CON signo

Sin signo: Todos los bits son usados para la magnitud del número y no hay uno que sea de signo. El rango de representación para n bits es de {0;2^n-1}.

Con signo: Se pueden representar con cuatro convenios, “signo y magnitud”, “complemento a 1”, “complemento a 2” y “exceso 2^(n-1)”.

Signo y magnitud o SyM: El bit más significativo indica el signo; 0 para positivo y 1 para negativo. En este convenio hay un +0 y un –0 que duplican el valor de dos combinaciones distintas de bits y se pierde un dígito para la magnitud, causando que el rango de representación para n bits cambie; {-(2^(n-1)-1); + (2^(n-1)-1)}. Este convenio no sirve para las matemáticas.

C	B	A	#
0	0	0	+0
0	0	1	+1
0	1	0	+2
0	1	1	+3
1	0	0	-0
1	0	1	-1
1	1	0	-2
1	1	1	-3

Convenio a 1: Los números positivos son iguales que los de SyM y para representar uno de estos en negativo se invierte los 0 por 1 y viceversa. En este convenio también hay un +0 y un –0 que duplican el valor de dos combinaciones distintas de bits y el rango de representación para n bits es {-(2^(n-1)-1); +(2⊠(n-1)-1)}. Este convenio tampoco sirve para las matemáticas.

C	B	A	#
0	0	0	+0
0	0	1	+1
0	1	0	+2
0	1	1	+3
1	0	0	-3
1	0	1	-2
1	1	0	-1
1	1	1	-0

Convenio a 2: Los números positivos son iguales que los de SyM y para representar uno de estos en negativo se opera empezando con los mismos pasos que convenio a 1 y se sigue sumando 1 al bit menos significativo.

$+12_d = 00001100_b$

-1_d = Ca2 de (+1) → +1 = 00000001_b

Ca1 = 11111110_b

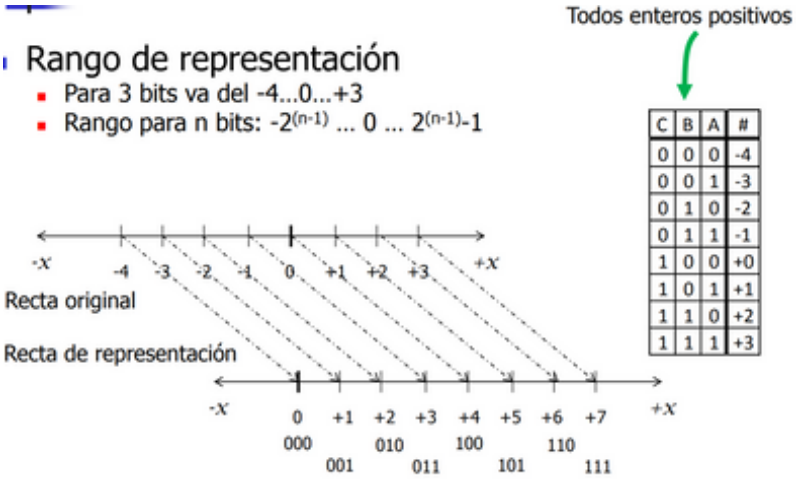
\+ 1_b

Ca2 = $11111111_b = -1_d$

El rango de representación para n bits es de {-(2^(n-1)); +(2^(n-1)-1)}. Este convenio sí sirve para las matemáticas.

C	B	A	#
0	0	0	+0
0	0	1	+1
0	1	0	+2
0	1	1	+3
1	0	0	-4
1	0	1	-3
1	1	0	-2
1	1	1	-1

Exceso 2^(n-1): El rango de representación {-2^(n−1); 2^(n−1) −1}. Los números en decimal no corresponden con sus binarios, esto se debe a que se desplaza la recta de representación en 2^(n-1), sumándole este resultado a cada punto a representar. Este convenio no sirve para hacer cuentas.



Punto fijo

UNIDAD 6: Memorias

Memorias

Son dispositivos utilizados para almacenar datos. Tienen capacidad de almacenamiento, tienen una determinada velocidad de acceso para leer o escribir, tienen permanencia de datos volátiles o permanentes y tienen presencia en circuitos integrados.

Jerarquía de memorias

De más rápidas, costosas y de baja calidad a más lentas, baratas y de alta capacidad.

Dentro del CPU:

- Registros, caché L1, caché L2 y caché L3.

En la placa madre:

- Memoria principal.

En el mismo gabinete:

- Discos estado sólido
- Discos rígidos
- Discos ópticos

En el gabinete central:

- Cintas magnéticas

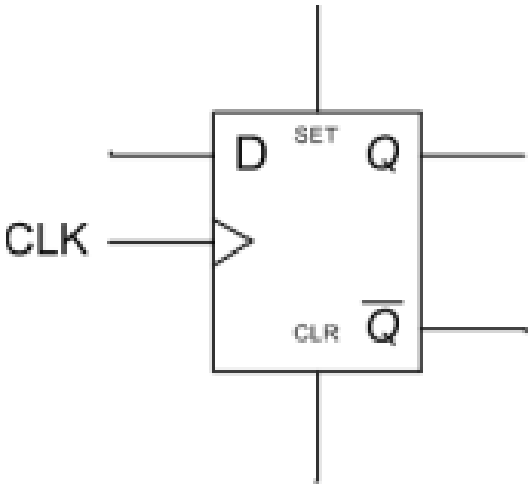
Memoria principal

Es uno de los elementos estructurales fundamentales de un computador, y se encarga de almacenar tanto los programas como los datos que están siendo utilizados en ese momento por el procesador.

- La Memoria Principal recibe/presenta los datos que se escribirán/entregarán en el bus de datos; este se encarga de transferir información entre el CPU, la memoria y los periféricos y es bidireccional, ya que la información puede fluir en ambos sentidos.
- La Memoria Principal recibe las órdenes de escribir o leer los datos en el bus de control; en este se encuentran las señales encargadas de la sincronización y control del sistema.
- La Memoria Principal recibe la dirección en donde se escribirán o de donde se leerán los datos en el bus de direcciones; este permite seleccionar la localidad de memoria, es unidireccional y siempre fluye desde el microprocesador.

Flip-flop

Es un dispositivo de almacenamiento de n bits o celdas de memoria; cada una de estas celdas almacena un solo bit y está compuesta por un flip-flops o biestable del tipo “D” que presentan el menor tiempo de acceso. Cada registro de n bits poseerá n flip-flops de este tipo:



D es la entrada sincrónica, las Q son salidas, CLK es la señal de clock, SET Coloca 1 en Q en forma asincrónica y CLR coloca 0 en Q de forma asincrónica. Para almacenar 0 o 1 de forma asincrónica hay

dos formas; de forma asincrónica se guarda un 1 en SET y un 0 en CLR o un 0 en SET y un 1 en CLR, de forma sincrónica se coloca un pulso en la entrada CLK.

Sea un registro de 4 bits; b3, b2, b1 y b0 siendo b3 el bit más significativo, cada uno de ellos con su correspondiente flip-flop, FF3, FF2, FF1 y FF0, el FF3 tendrá las señales D3, Q3 y !Q3. Entra el dato por D0, se conectan Q0 a D1, Q1 a D2 y Q2 a D3 y sale el dato por Q3.

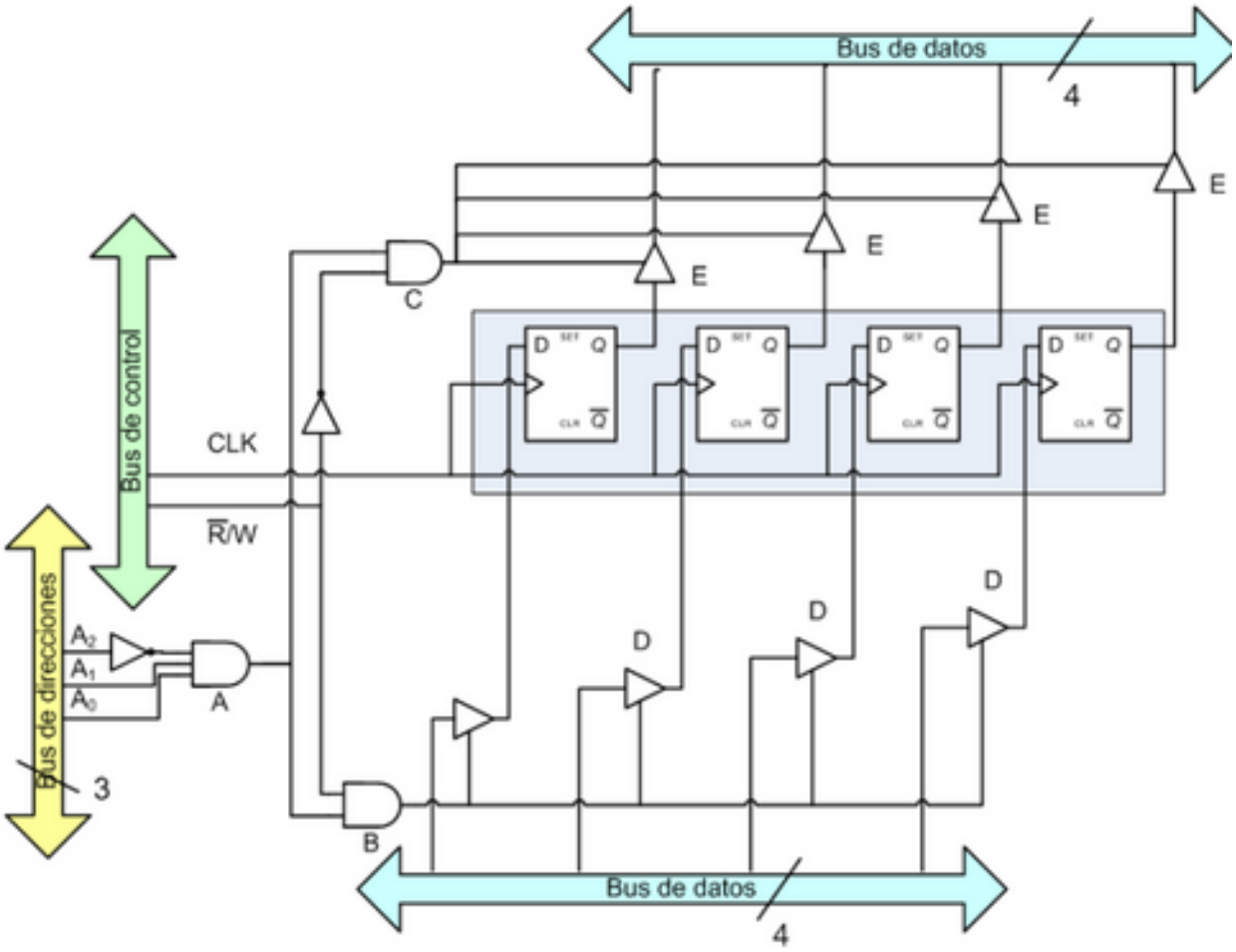
Para guardar un 0 asincrónicamente se coloca SET=0 y CLR=1, Para guardar un 1 sincrónicamente se coloca D=1 y se espera el pulso de CLK.

Lectura y escritura de registros

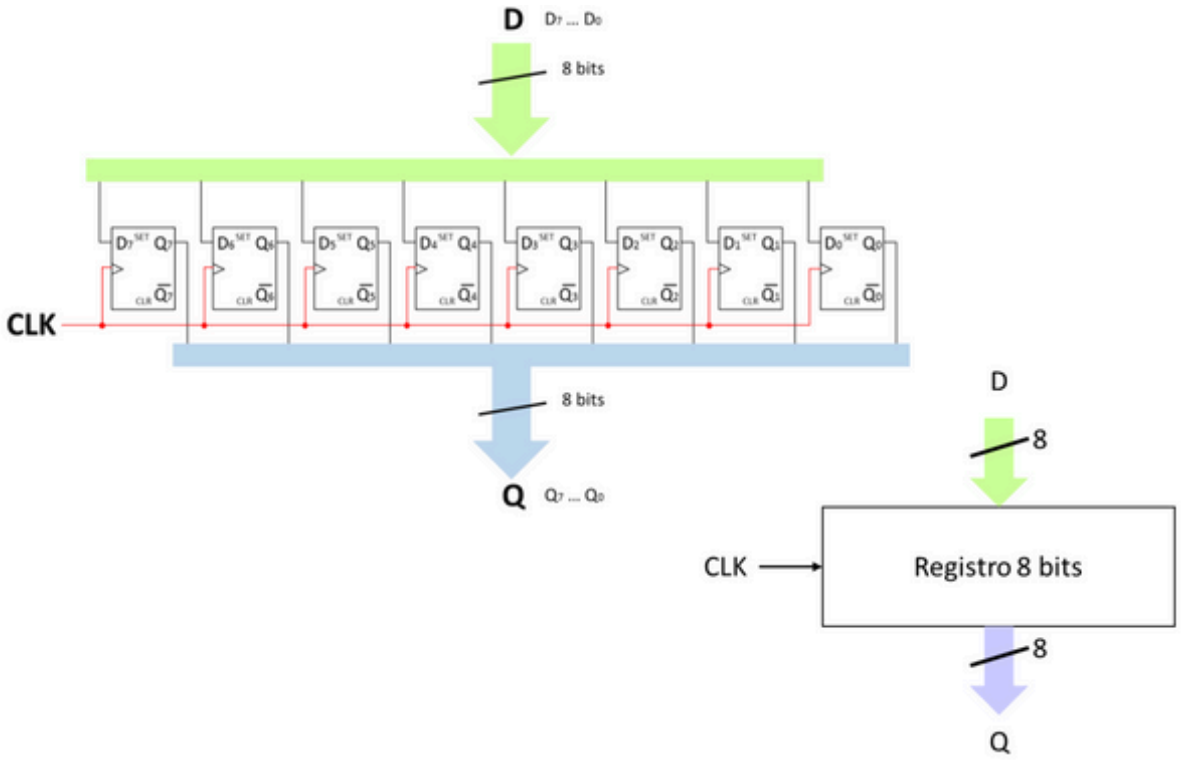
Para leer o escribir bits en las celdas de los registros; se coloca 011b en el bus de direcciones y la salida de la compuerta A es 1.

Para leer; la señal R/W se coloca en 0, haciendo que la salida de la compuerta C sea 1, habilitando los buffers 3-state E y disponiendo del dato del bus, y la compuerta B tiene un 0 en una de sus entradas para que no se escriba nada en el registro.

Para escribir; se presentan los datos en el bus de datos, la señal R/W se coloca en 1 para que la salida B sea 1, se habiliten los buffers 3-states y se disponga del dato en cada flip-flop, y la compuerta B tiene un 0 en una de sus entradas para que no se lea durante este3 ciclo.



El pulso de la señal de clock es el evento que marcará el momento de la escritura de un dato en un registro. Para seleccionar uno de los n registros disponibles, se debe decodificar el contenido del bus de direcciones. Al habilitarse los buffers 3-state conectados a las salidas Q del registro, el dato está disponible en el bus sin necesidad de un pulso de CLK. La señal !R/W es la que define si se habilitan los buffers 3-state conectados a las salidas Q o los conectados a las entradas D.

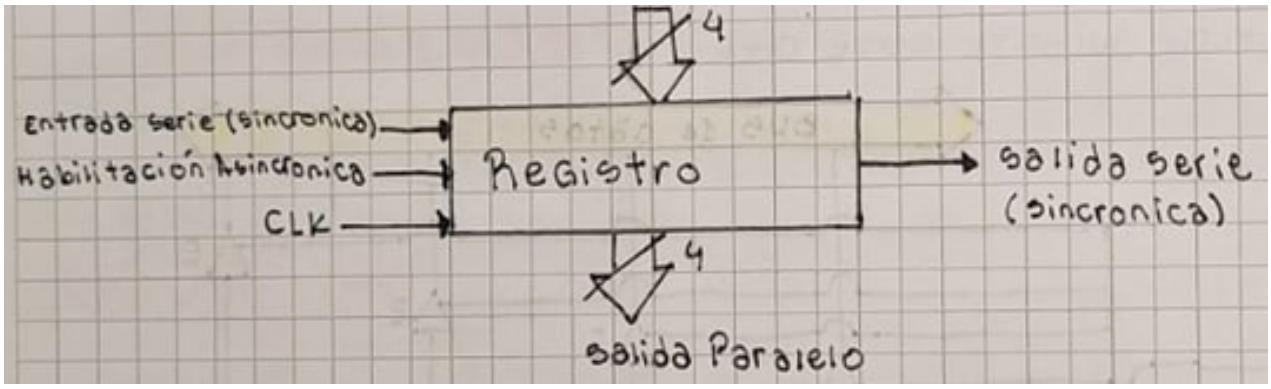


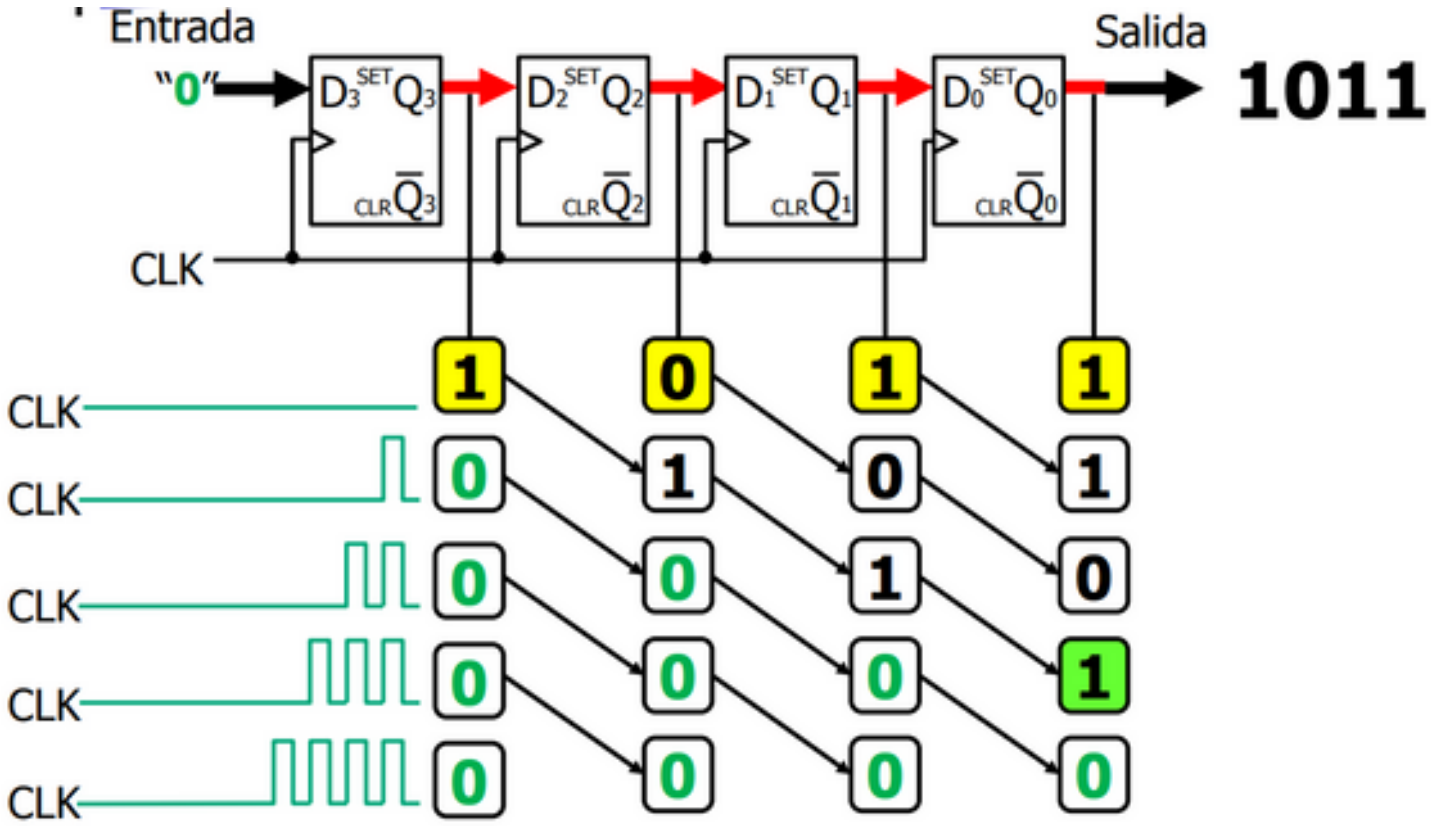
UART

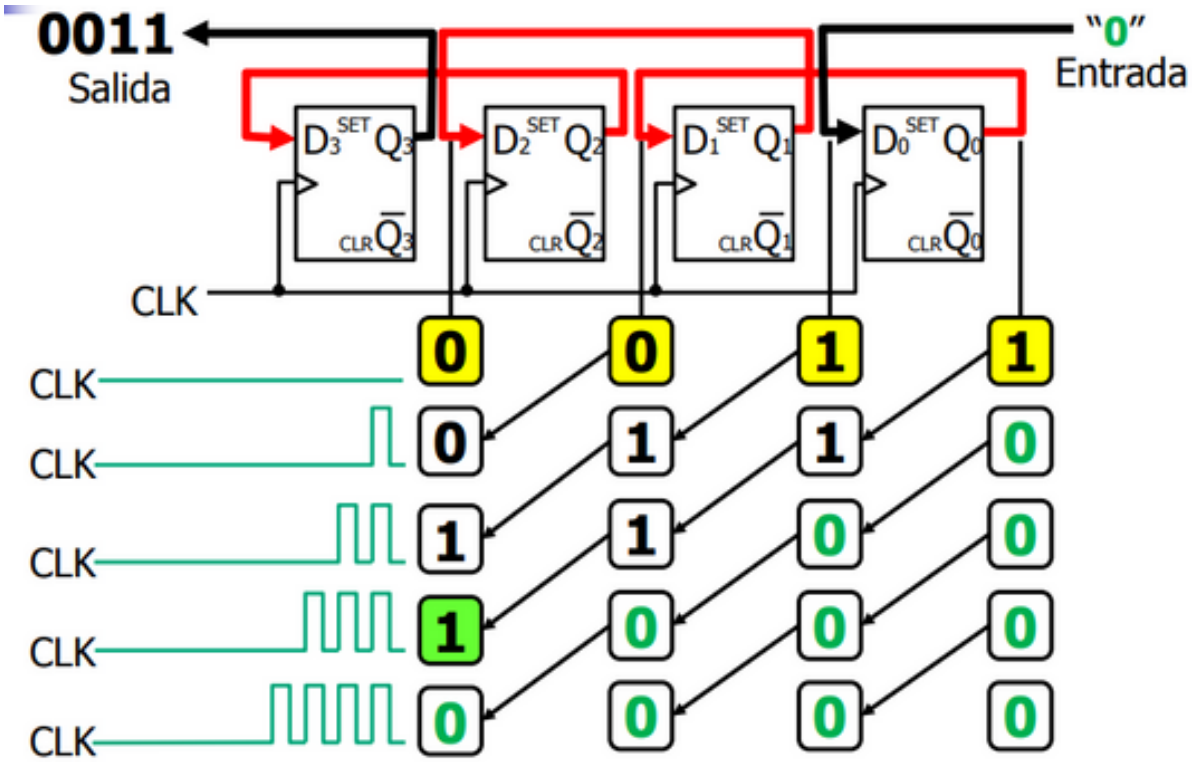
Es un dispositivo que permite la comunicación serial entre dos sistemas; convierte datos paralelos a serie y viceversa, permitiendo que se envíen por un solo cable, bit por bit, de forma asincrónica, es decir, sin compartir una señal de reloj.

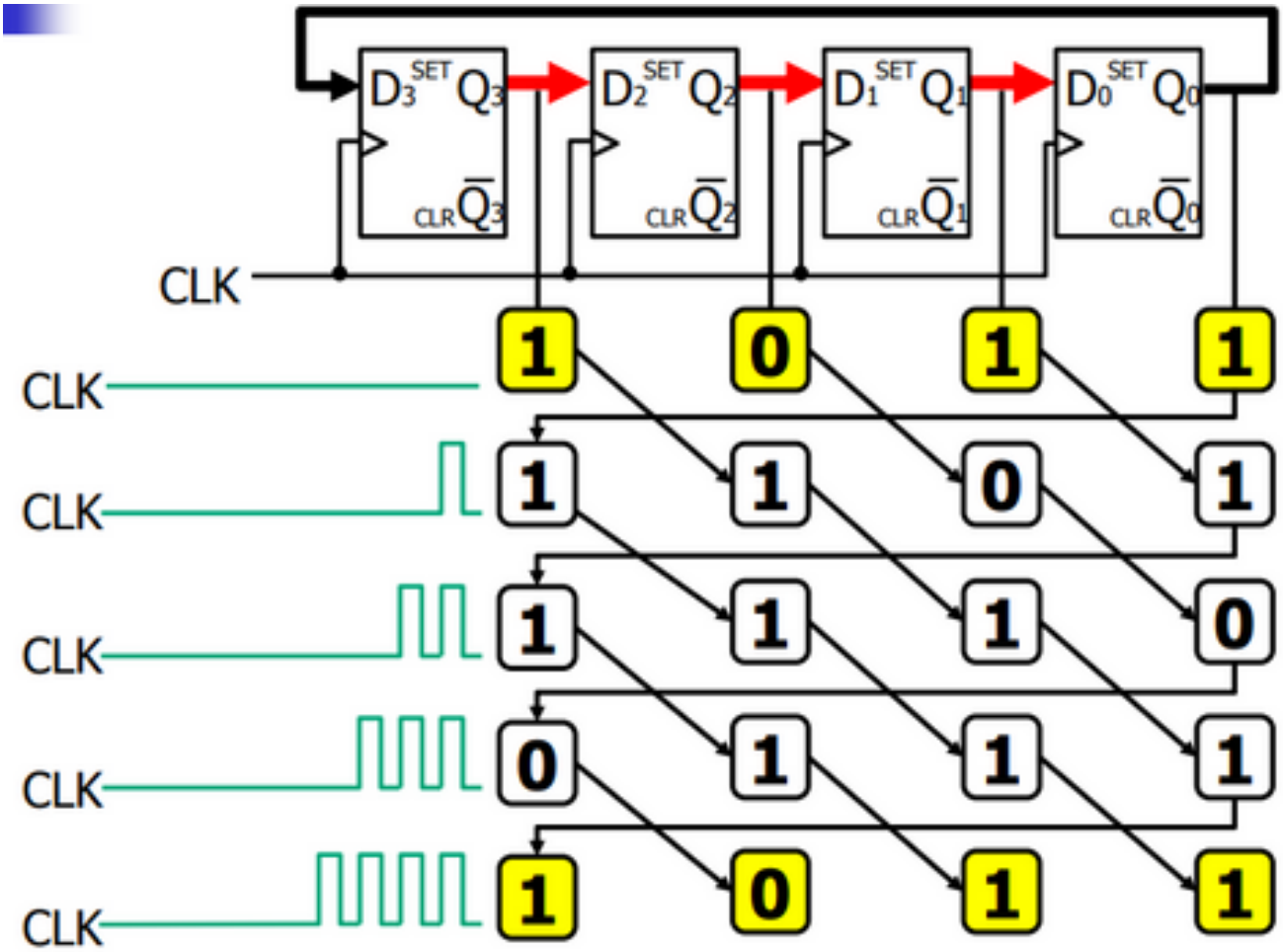
Los dispositivos que la UART contiene interiormente para poder realizar estas operaciones de conversión paralelo/serie son los registros de desplazamiento; circuitos digitales que permiten cargar datos en paralelo, desplazarlos bit a bit hacia la salida (serie) o al revés, recibir bits uno a uno por una línea serial y armar el byte completo en paralelo.

Debidamente interconectados, los registros vistos pueden desplazar sus bits, alojados en cada uno de los flip-flops, hacia la derecha o hacia la izquierda. Estos se encuentran en la ALU para hacer las operaciones producto y cociente y el movimiento de los bits se realizará con cada pulso del CLK o reloj. Se los puede cargar de forma asincrónica a través de SET y CLR o sincrónicamente a través de D y el CLK.


4
Entrada serie (sincronica)
Habilitación Asincronica
CLK
Registro
salida serie
(sincronica)
4
salida Paralelo


Entrada
"0"
Salida
1011
CLK
CLK
CLK
CLK
CLK
CLK





Memoria principal

Dispositivos rápidos conectados y controlados directamente por la CPU. Suele estar organizada en base a palabras de n bits y se accede a cada palabra de una por vez y a través de una única dirección. Los datos contenidos son simplemente 0 y 1, su interpretación dependerá de quien los lea o escriba.

El decodificador selecciona a cuál de los 2^n palabras se quiere acceder. La matriz de celdas de memoria almacena los 0 y 1. La lógica de control gestiona las operaciones de lectura y escritura del dispositivo. Los transductores adaptan los niveles de señal necesarios entre el bus de datos y las celdas de la matriz de la memoria.

Latencia

Access time: Desde que se realiza un pedido de lectura hasta que el mismo queda satisfecho.

Cycle time: Desde que se realiza un pedido hasta que se pueda realizar el siguiente.

Write time: Al escribir un dato, tanto el dato y la dirección donde se escribirá deben estar presentes antes de que llegue la señal de escritura y deben mantenerse luego de que se retire la señal de escritura (hold time) .

CAS (column access strobe): Indica el tiempo que tarda la memoria en colocarse sobre la columna.

RAS (row access strobe): Indica el tiempo que tarda la memoria en colocarse sobre una fila.

Active: Indica el tiempo que tarda la memoria en actuar sobre un tablero.

Precharge: Indica el tiempo que tarda la memoria en activar un tablero.

Tipos de memorias

SRAM (Static Random Access Memory): Son estáticas y son un tipo de memoria de lectura y escritura que mantiene sus datos siempre y cuando tenga alimentación. Son muy rápidas, son caras y consumen mucha energía. Son usadas en dispositivos que requieren alta velocidad de operación, como caché de procesadores y procesamiento digital de imágenes.

DRAM (Dynamic Random-Access Memory): Necesitan un pulso de energía periódico, o refresco, para mantener sus datos. Son más lentas y baratas que las SRAM y consumen menos energía. Son usadas como memoria principal en todo sistema con microprocesadores y se convirtieron en el cuello de botella del sistema moderno o PC.

SDRAM (synchronous DRAM): Permiten el acceso a un bloque de datos que estén en fila, haciendo la transferencia sincronizada. Permiten la transferencia en ráfagas, o burst, incorporan un circuito para hacer el refresco automáticamente, haciendo que su velocidad de transferencia de datos sea más lenta. Eran muy comunes en las PC hasta hace un par de años.

DDR RAM (double data rate random access memory): Fabricadas con tecnología SDRAM, pero transfieren los datos en ambos flancos de reloj, duplicando la tasa de transferencia. Trabajan con 2,5 V en lugar de 3,3 V con que trabajan las SDRAM. Adoptadas inicialmente por AMD, mientras Intel utilizaba las RAMBUS. Su Buffer interno es de 2 bits, velocidades del buffer desde 200 MHz, hasta 400 MHz, DIMM 184 contactos.

De acá, seguimos hasta las DDRS, cada vez se vuelven más eficientes, consumen menos energía, se aumentan las transferencias de datos y la capacidad de memoria.

XDR (extreme data rate): Una espacie de RAM-bus usada en la Nintendo 64.

MRAM (magneto resistive RAM): Básicamente, una RAM que utilizaba magnetismo.

Métodos de acceso

Aleatorias: Permiten direccionar cualquier posición de la memoria de forma directa e independientemente del lugar en donde se encuentre.

Semi - aleatorias: Discos floppy, discos duros, CD-ROM.

Secuenciales: Par acceder a una posición de memoria, debo recorrer la memoria desde el principio.

Clasificación de volatilidad

Volátiles: Al desconectarlas del suministro eléctrico, se pierden los datos. Pueden ser estáticas, construidas con flip-flops, o dinámicas, construidas con capacitores.

No volátiles: Los datos permanecen aún sin ser energizadas:

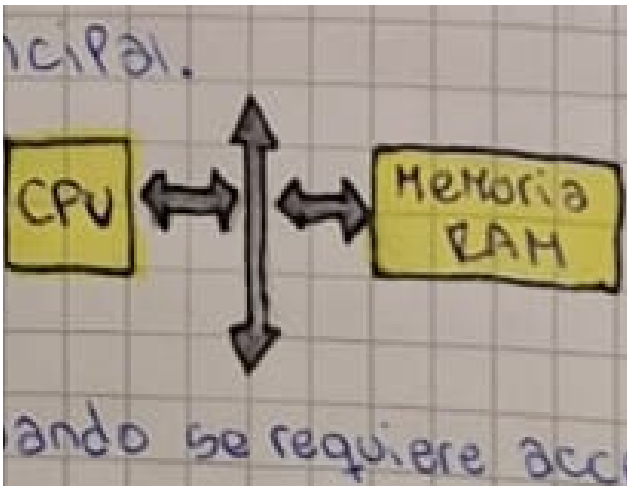
1. ROM (read only memory): Se graban los datos Durante el proceso de fabricación y no se pueden alterar, se coloca un diodo en donde se requiere guardar "ceros".

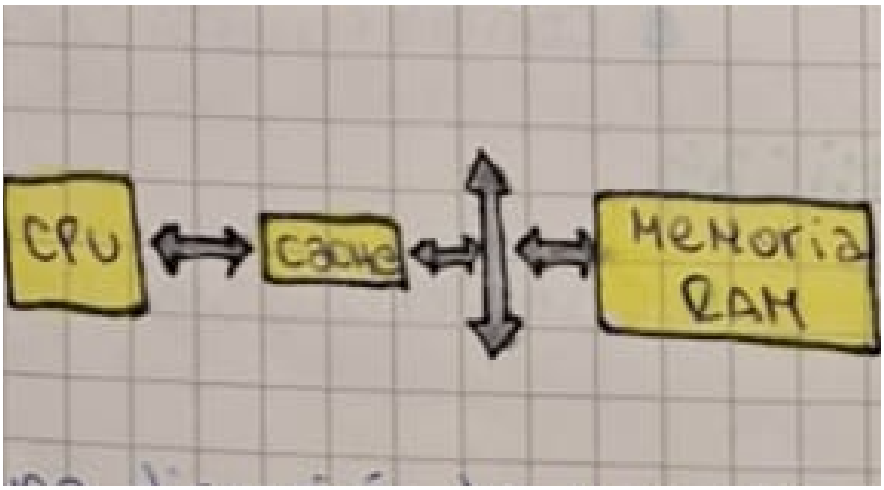
1. PROM (programmable read only memory): Se programa una sola vez, en donde, para guardar un "uno", hay que quemar el fusible; esto se realiza mediante un programa externo.

Caché

Es una memoria muy rápida que está ubicada dentro del procesador o cerca de él. Su función principal es guardar temporalmente los datos e instrucciones que se usan con más frecuencia, para que el procesador no tenga que acceder constantemente a la memoria RAM, que es más lenta.

El objetivo del caché es incrementar la performance del sistema; de CPU a memoria principal:





Cuando se quiere acceder a una dirección de memoria, con altísima probabilidad usaremos las que siguen, por ende se traen los bloques contiguos de memoria. Si el procesador encuentra el dato en el caché es un hit, si no es una miss.

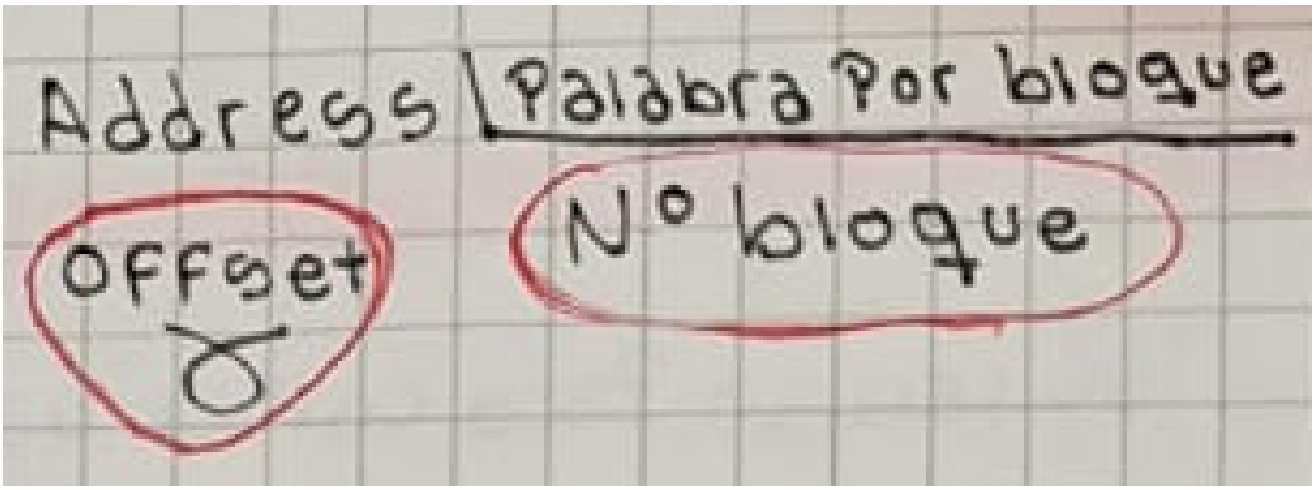
La organización de la memoria caché indica las reglas para copiar los datos de la memoria principal a la caché y las reglas para alojar nuevos datos cuando la caché esté llena. La memoria común tiene una capacidad de 2^n bytes, se divide en bloques consecutivos de b palabras, el tamaño de estos bloques es una potencia de 2 y se tendrá la cantidad de 2^n/b bloques.

Existen tres organizaciones:

Asociativa:

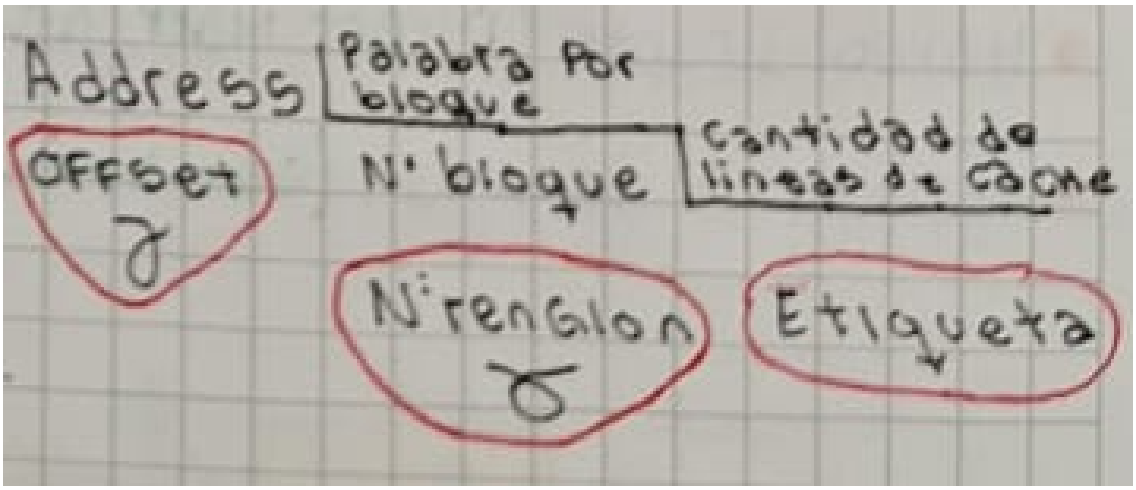
- M cantidad de líneas o renglones donde cada uno tiene una B (bit de validez), N (#bloque) y V (valores). Los datos de cualquier dirección de memoria pueden ser almacenados en cualquier línea de la caché, sin restricciones. Primero se calcula el bloque, luego se busca ese bloque en la caché y de ahí se saca el dato. Si no se encuentra se trae de la memoria principal.

Cualquier bloque de memoria puede almacenarse en cualquier línea de caché, no hay restricciones de ubicación porque cuando se busca un bloque se revisa toda la caché para ver si está y si no está puede colocarse en cualquier línea libre o reemplazar alguna existente. La ventaja es que causa menos conflictos que el mapeo directo.



Mapeo directo:

- Tiene m cantidad de renglones, b palabras por bloque, cada renglón tiene B (bit de validez), E (etiqueta) y V (valores), cada bloque tiene asignado un único renglón, El número de bloque es igual a dirección/b, la etiqueta es igual a #bloque/(cantidad de renglones) y el número de renglón es igual al resto de esta última división.



Cada bloque de la memoria principal sólo puede almacenarse en una única línea caché; la ventaja es que es simple y rápido de implementar y la desventaja es que si, dos bloques comparten el mismo índice, compiten entre sí y se reemplazan constantemente; a eso se le llama conflicto.

Políticas de escritura a memoria principal

Determinan cómo se administrará la escritura de la memoria RAM común con los datos contenidos en la memoria caché.

Escritura a memoria (WRITE – THROUGH CACHE):

- Cuando se modifica la caché, se modifica el mismo dato en la memoria principal, no tiene problemas de consistencia y es beneficioso para procesos de lectura intensiva.

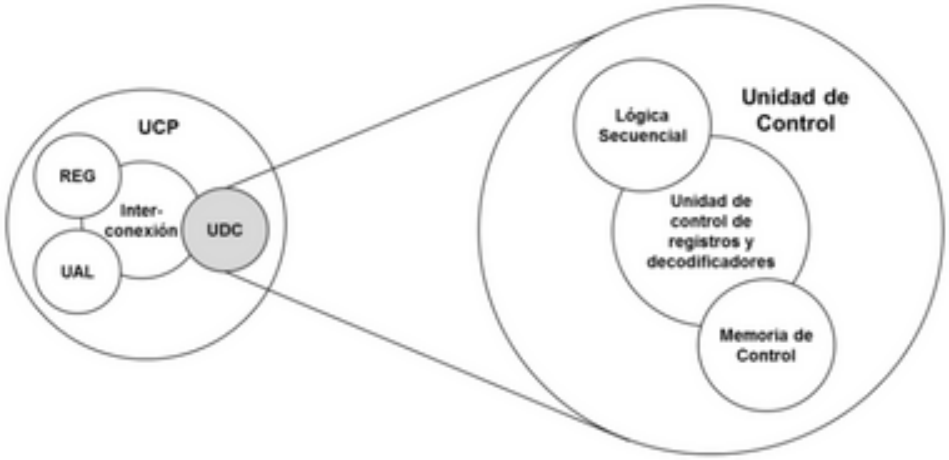
Retro grabado (WRITE – BACK CACHE):

- Se modifica solamente la caché, pero el dato en la memoria principal común se modifica cuando se necesita desocupa un bloque de la caché que tiene los datos modificados. Tiene más performance que el anterior, es beneficioso para procesos de escritura intensiva y es necesario usar un bit más para indicar si el renglón de la caché fue modificado o no (este es el DIRTY – OUT).

Unidad 7: Unidad de control, pipelines, RISC y CISC

Unidad de control (UDC o CU)

Es la encargada de la coordinación y orquestación de todas las tareas para la ejecución de las instrucciones en formato binario, es decir, bajo nivel. Durante la ejecución de una instrucción, la UDC generará todas las señales eléctricas internas necesarias para la conexión y desconexión de las diferentes unidades estructurales internas de la CPU y las señales eléctricas externas, que son señales de lectura, escritura, etc. La ALU realiza las operaciones aritméticas y lógicas ordenadas por la unidad de control y procesa los datos.



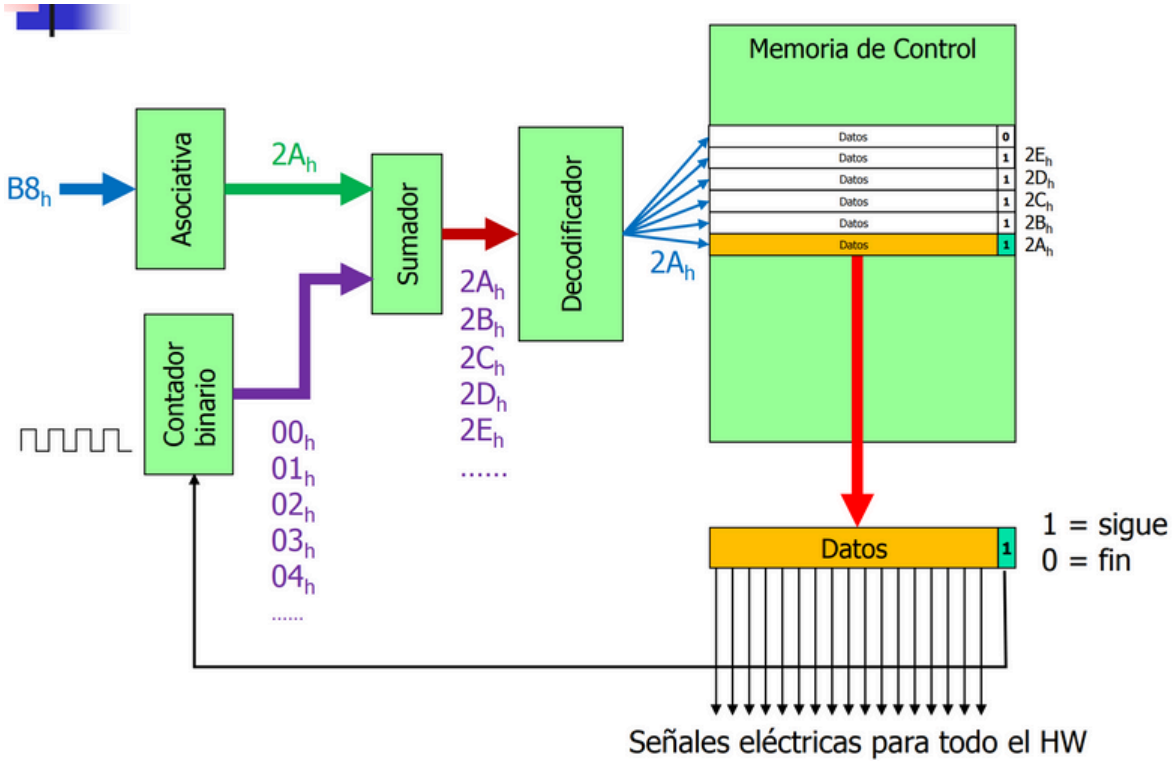
Las, inicialmente, eran cableadas porque las conexiones necesarias para la ejecución de cada instrucción se hacían con cables externos. Ahora son micro – programadas, porque los cables fueron reemplazados por circuitos que se conectaban o desconectaban con un 1 o 0; dichos 1 y 0 de Control están almacenados en

la UDC, haciendo que cada instrucción se ejecute en allí como un conjunto de microinstrucciones. A este conjunto se lo denomina microprograma y cada microinstrucción ocupada un ciclo de CLK.

Según como se ubiquen las microinstrucciones en la memoria de control se tendrán dos tipos de secuencia - miento de las mismas:

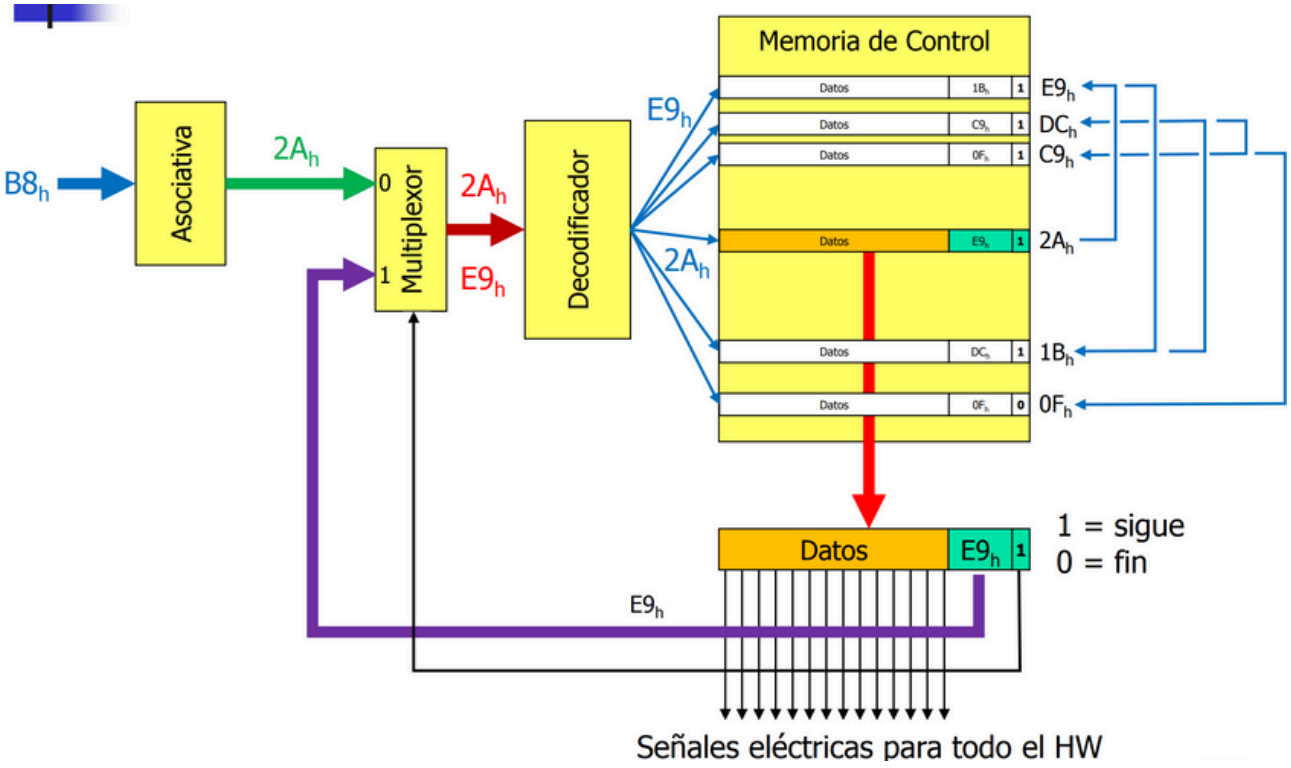
Secuencia - miento implícito

Las microinstrucciones se encuentran ordenadas secuencialmente en la memoria física en direcciones contiguas. Se emplea un incrementador al cual ingresa la dirección de la microinstrucción en curso y de esa manera se accede a las subsiguientes microinstrucciones.



Secuencia - miento explicito

Las microinstrucciones no se encuentran ordenadas secuencialmente en la memoria de control, por lo tanto, cada microinstrucción incorpora la dirección de la siguiente microinstrucción.



Formato de instrucciones

El formato consiste en un primer paquete de bits que define la operación (OPCODE o COP = Código de Operación) y luego una serie de paquetes de bits (OBJECTS) con la información complementaria para completar la operación.

Formato de instrucciones de longitud variable (VAX)

Operación y nro. de operandos	Especificador de la dirección 1	Dirección 1	...	Especificador de la dirección n	Dirección n

Formato de instrucciones de longitud fija (DLX, MIPS, PowerPC, HP-PA, SPARC)

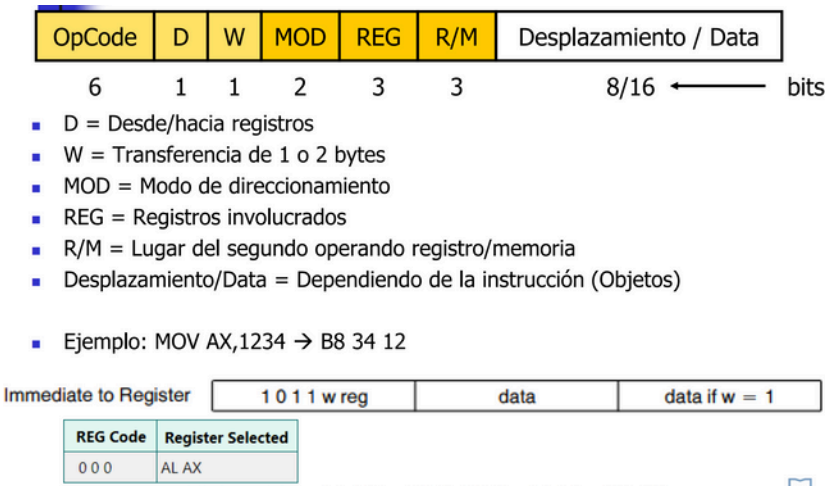
Operación	Dirección 1	Dirección 2	Dirección 3

Formato de instrucciones combinados (IBM 360, IBM 370, Intel 80x86)

Operación	Especificador de la dirección	Dirección

Operación	Especificador de la dirección 1	Especificador de la dirección 2	Dirección

Operación	Especificador de la dirección	Dirección 1	Dirección 2



Set de instrucciones

Es el conjunto de instrucciones disponible para el programador. Son códigos binarios que tienen una asociación a un lenguaje simbólico denominado assembler (o ensamblador); el lenguaje assembler que se utilice para programar atará la ejecución a un microprocesador determinado. Cada instrucción está caracterizada por un código de operación.

El código de operación de una instrucción provee en forma directa o indirecta, a través de un decodificador, el lugar en donde se encuentra de la primera microinstrucción de la microprograma a ejecutar. Cada instrucción del "set" indica los registros que lee o modifica, la longitud de cada instrucción depende del tipo de instrucción y cada instrucción puede demandar más o menos de ciclos de reloj para su ejecución completa.

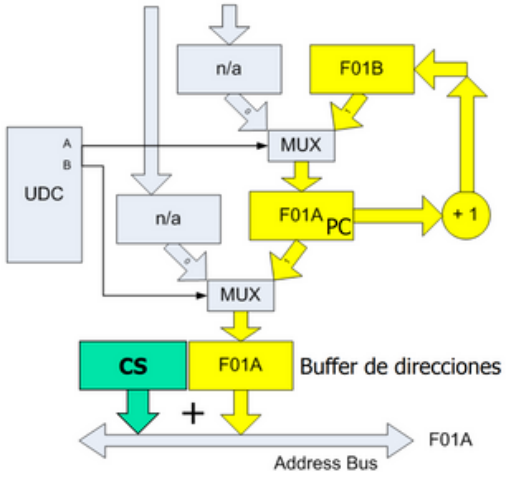
Contador de Programa (PC o PROGRAM COUNTER)

Contiene la dirección de la próxima instrucción a ejecutarse y es parte de la arquitectura del computador, por lo que es accesible por el programador. Cuando se inicia la ejecución de un programa nuevo, se carga este registro con la dirección de la primera instrucción del programa en cuestión.

Dependiendo del tipo de instrucción a ejecutar y de la lógica del programa, el PC se comportará de diferentes modos:

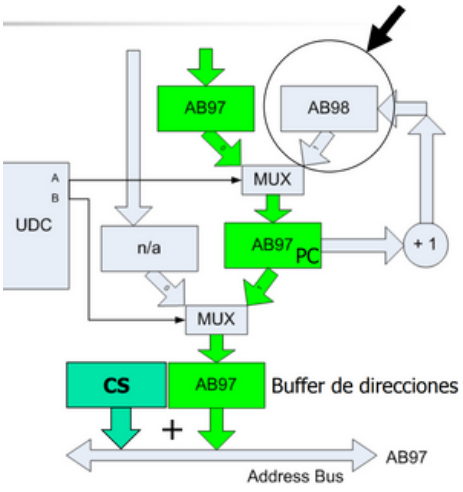
Modo habitual

El PC se carga con la dirección de la primera instrucción del programa, se lee y se incrementa en 1 para proseguir leyendo la próxima posición de memoria.



Modo salto

Se procesado salto en la ejecución del programa, por lo que se deberá continuar la ejecución en una posición distinta a la próxima, es decir, con funcionamiento habitual. Se carga el contador de programa en forma directa y luego se continua luego la ejecución en forma habitual.



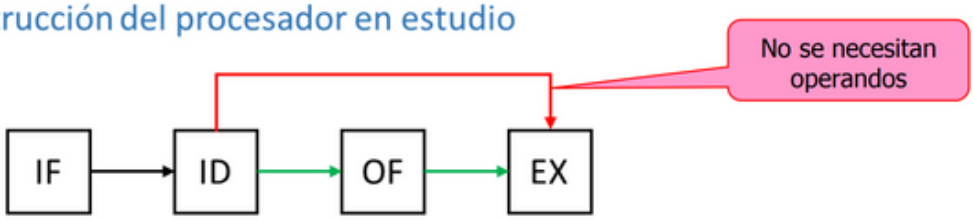
Modo indirección

El objeto requerido no se encuentra ubicado a continuación, por lo que se resuelve la indirección y se la carga en el buffer de direccionamiento. La ejecución del programa no se altera, por lo que el PC no cambia.

Ciclo de instrucción

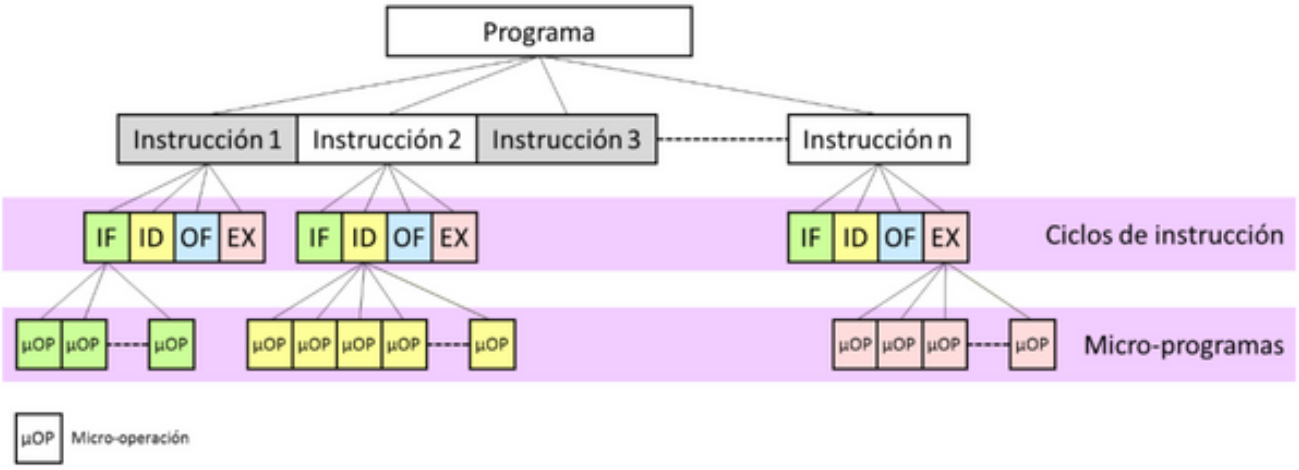
Es el ciclo repetitivo que sigue la UDC al ejecutar instrucciones. El ciclo de instrucción del procesador es el siguiente:

Ciclo de instrucción del procesador en estudio



IF = Instruction Fetch (Búsqueda del Código de Operación)
ID = Instrucion Decode (Decodificación del Código de Operación)
OF = Operand Fetch (Búsqueda de los operandos u objetos restantes)
EX = Execution (Ejecución de la operación)

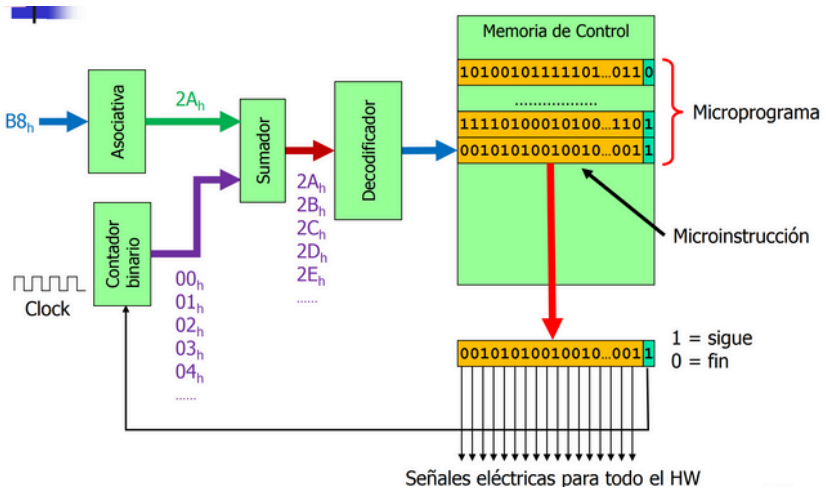
Un Microprograma es una secuencia de Microinstrucciones, denominando así al conjunto de señales eléctricas que serán necesarias para dicha ejecución.



Para la búsqueda de Código de Operación (INSTRUCTION FETCH), la UDC procederá a acceder a la MP a la dirección donde apunta el PC y leerá el COP. Esté COP vendrá por el bus de datos y será alojado en un registro interno llamado INSTRUCTION REGISTER (Registro de Instrucción).

Operación	Especificador de la dirección	Dirección	
Operación	Especificador de la dirección 1	Especificador de la dirección 2	Dirección
Operación	Especificador de la dirección	Dirección 1	Dirección 2

Para la Decodificación del Código de Operación (INSTRUCTION DECODE), el contenido del registro de instrucción será decodificado (mediante un DECODER o decodificador) entregando la posición de la Memoria de Control en donde se encuentra la primera microinstrucción a ejecutar. A partir de allí, la UDC por si sola recorrerá todas las microinstrucciones del micro - programa de ejecución de la instrucción.



Capacidad de direccionamiento

Indica cuántas direcciones, en binario, es capaz de generar un procesador para acceder a memorias físicas, a los puertos de entrada y salida y a la memoria virtual. La capacidad de direccionamiento para “n” bits es 2^n; dicha cantidad de bits está determinada por la cantidad de pines del microprocesador en el bus de direcciones. Para el acceso a los periféricos se usan los primeros 16 bits, es decir, desde la 0000h hasta la FFFFh que representa 65.536 direcciones.

Paralelismo

Es una técnica para reducir el tiempo y costo de recursos, es decir, mejorar la performance. Existen varios tipos de paralelismos con los que trabajan los componentes de las computadoras:

El paralelismo temporal

Es para solapar tiempos en un mismo dispositivo.

Paralelismo espacial

Es para agregar más dispositivos para trabajar al mismo tiempo.

Paralelismo de multiprocesamiento simétrico (SMP)

Compone dos o más procesadores donde todos son independientes entre sí, son de características similares, comparten un mismo espacio de memoria, comparten un mismo espacio de entrada y salida, poseen similares tiempos de acceso a la memoria principal, pueden ejecutar los mismos algoritmos o funciones y son controlados por un único sistema operativo integrado.

Anteriormente, la SMP tenía dos o más procesadores físicos y separados trabajando en paralelo, pero hoy en día puede tener un procesador con dos más núcleos donde cada uno trabaja en paralelo.

Paralelismo de multiprocesamiento en clústeres

Son un conjunto de computadoras que trabajan como si fuesen una sola supercomputadora, cuando cada una podría trabajar como un ordenador con todas sus funciones. Estas computadoras se enlazan o comunican a través de una red de área local o una red de área amplia y poseen discos de compartidos por cada una de ellas con configuración redundante de discos.

Acceso a memoria

Uniforme u homogéneo (UMA)

Todos los procesadores comparten un mismo espacio de memoria y los tiempos de memoria son independientes del procesador que acceda.

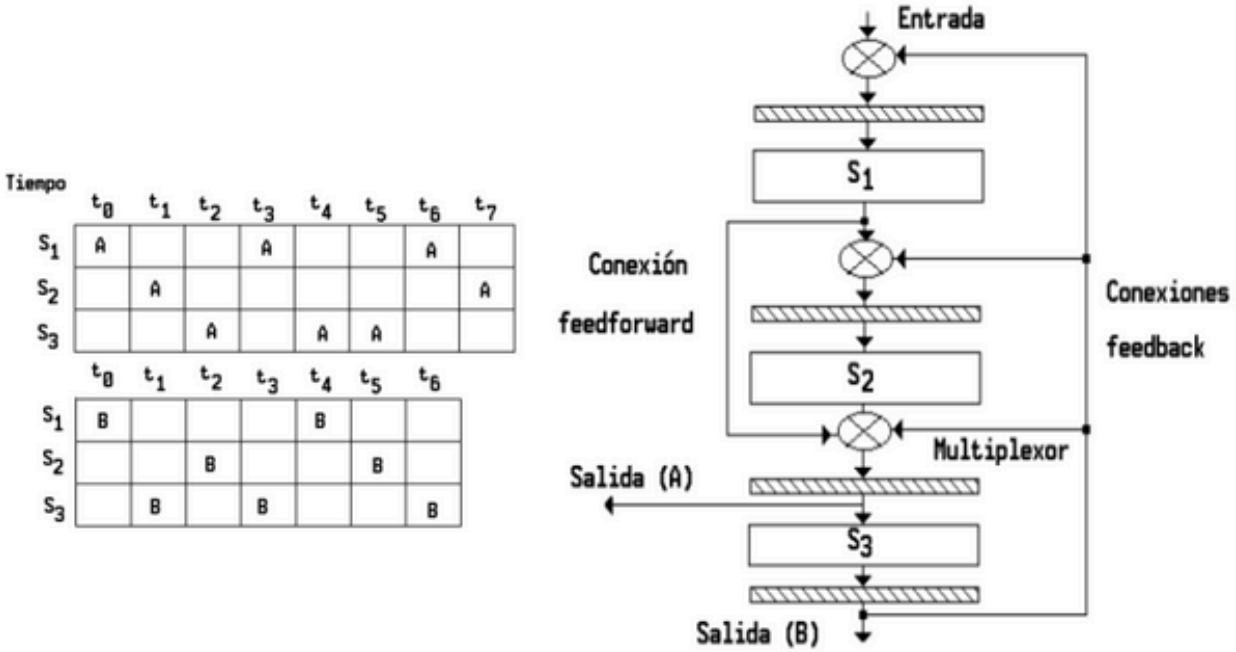
No uniforme o heterogéneo (NUMA)

Cada procesador tiene un espacio propio de memoria que no comparte con otros procesadores, hay coherencia de caché entre todos los niveles para que los procesadores no usen una versión vieja de un dato si uno de ellos lo modifica, se utiliza el caché L3 para compartir entre todos los procesadores y, para evitar que los datos compartidos circulen por los otros buses, los niveles L1 y L2 son propios.

Pipeline

Es una técnica de ejecución de instrucciones donde múltiples instrucciones se procesan al mismo tiempo, pero en diferentes etapas. En cada etapa se realiza una acción con una instrucción y, una vez que terminan, cada etapa recibe la instrucción que terminó de procesar la etapa anterior para seguir trabajando. Cada etapa o segmento de la cadena está especializada en una tarea específica de la "línea de ejecución" y lleva a cabo siempre la misma actividad. Esta tecnología es propia de procesadores eficientes.

En el siguiente ejemplo, hay dos instrucciones, A y B, en tres etapas, S1, S2 y S3; las instrucciones pasan en orden por la entrada y luego por las etapas que representan cada paso en el procesamiento de una instrucción. El feedforward permite adelantar resultados de una etapa a otra para evitar retrasos y no esperar a que todo termine, el feedback permite recircular resultados hacia etapas anteriores si se necesita reusar datos y el multiplexor elige qué datos o señales van a la siguiente etapa. Las salidas A y B representan los resultados finales de cada instrucción luego de pasar por el pipeline completo.



Los pipelines pueden tener errores cuando la CPU trata de ejecutar instrucciones en forma simultánea y entre ellas hay dependencia de datos; dichos problemas pueden ser de datos, de control y de estructura.

Problemas de datos:

RAW (READ-AFTER-WRITE): Un registro modificado, es leído inmediatamente luego. Como la modificación puede no haber finalizado, la lectura puede ser errónea.

- MOV AX, BX
- MOV [DS:0302], AX

WAR (WRITE-AFTER-READ): Similar al anterior, pero alternando la lectura por modificación y viceversa.

- MOV SI, AX
- MOV AX, [BX + 20_h]

WAW (WRITE-AFTER-WRITE): Se ejecutan dos instrucciones que escriben el mismo operando. La primera puede terminar luego de la segunda.

- MOV DX, 200_h
- MOV DX, offset DATONUEVO

Problemas de control: Ocurren cuando una instrucción de salto deberá ejecutarse, pero no se puede saber si la condición de salto será satisfactoria con lo cual la próxima instrucción a ejecutar no es la que físicamente está a continuación del salto.

Problemas de estructura: Ocurren cuando dos instrucciones requieren una misma unidad estructural, como la ALU.

- ADD AX, BX
- MUL CX

Solución al problema de datos (forwarding): Se envía el dato necesario de manera anticipada, sin esperar a que se actualice el registro final. Lo manda antes, desde la etapa donde la instrucción accede a la memoria

(MEM - MEMORY ACCESS) a donde se lo requiera.

Solución al problema de control (predicción de saltos): Se basa en poder determinar si una condición de salto (o división en dos ramales) en el flujo de instrucciones se ejecutará o no para permitirle a la CPU encontrar y ejecutar instrucciones sin tener que esperar que se resuelva la condición de salto. Estos predictores se utilizan en arquitecturas con pipeline, con lo cual evitan que el pipeline de instrucciones se vacíe. Existen más de un tipo de predictor de salto:

Estático: No se basa en la dinámica del código, sólo de la instrucción aislada, por lo que predice que el salto no va a ocurrir y continúa con la instrucción siguiente. Cuando condición de salto es evaluada y sí se da, continua la ejecución de la dirección de salto. Otros asumen que los saltos "hacia atrás" ocurrirán, mientras que los saltos "hacia delante" no lo harán; efectivos para Loop.

Dinámico: Se basa en una larga lista de historia de saltos para ser más preciso, guardando bits para indicar si en el pasado se ejecutó el salto o no; se basa en conceptos neurales. Se utiliza dos bits para historia de saltos; si una predicción se cumple, se mantiene la predicción; si no se cumple por primera vez, lo registra, pero mantiene la predicción; si una predicción no se cumple por segunda vez consecutiva, se cambia la predicción.

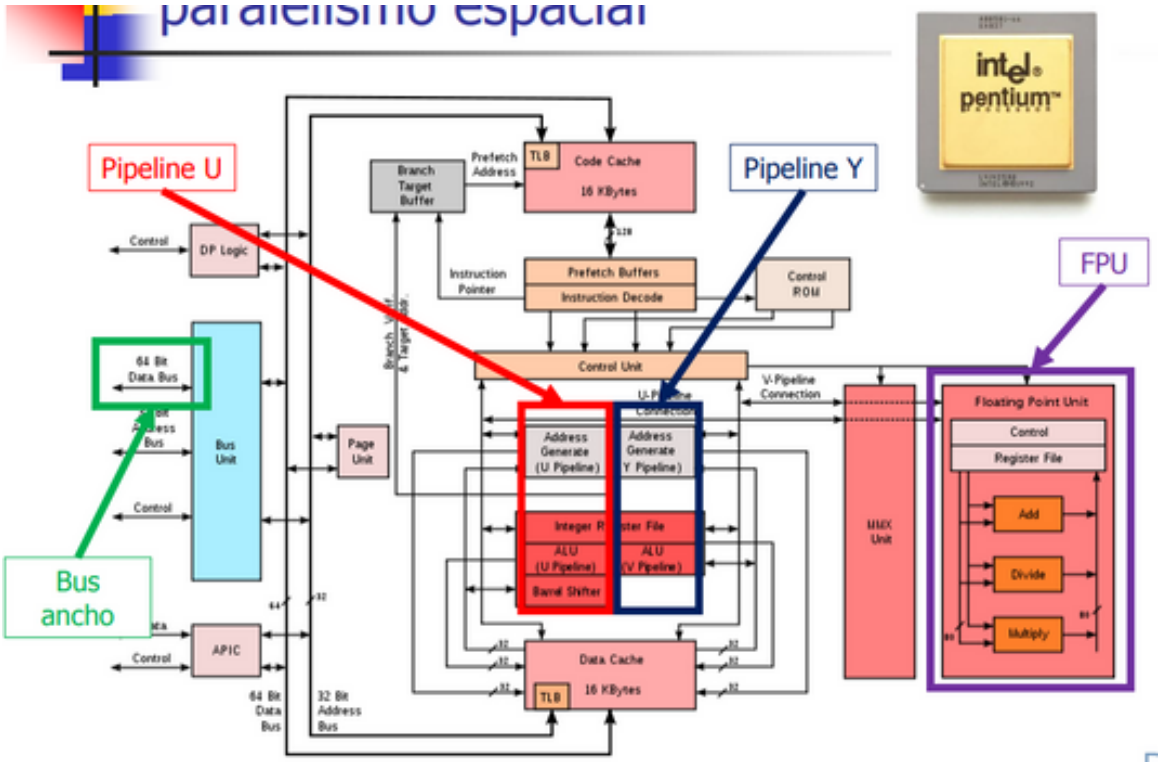
Solución al problema de estructura (más unidades): Se agrega más estructuras. Si, por ejemplo, tengo que realizar dos instrucciones en donde debe intervenir la ALU, debiera esperar a que finalice la primera para poder ingresar la segunda; por ello, se pueden integrar en lugar de una ALU, dos ALU.

Super pipeline

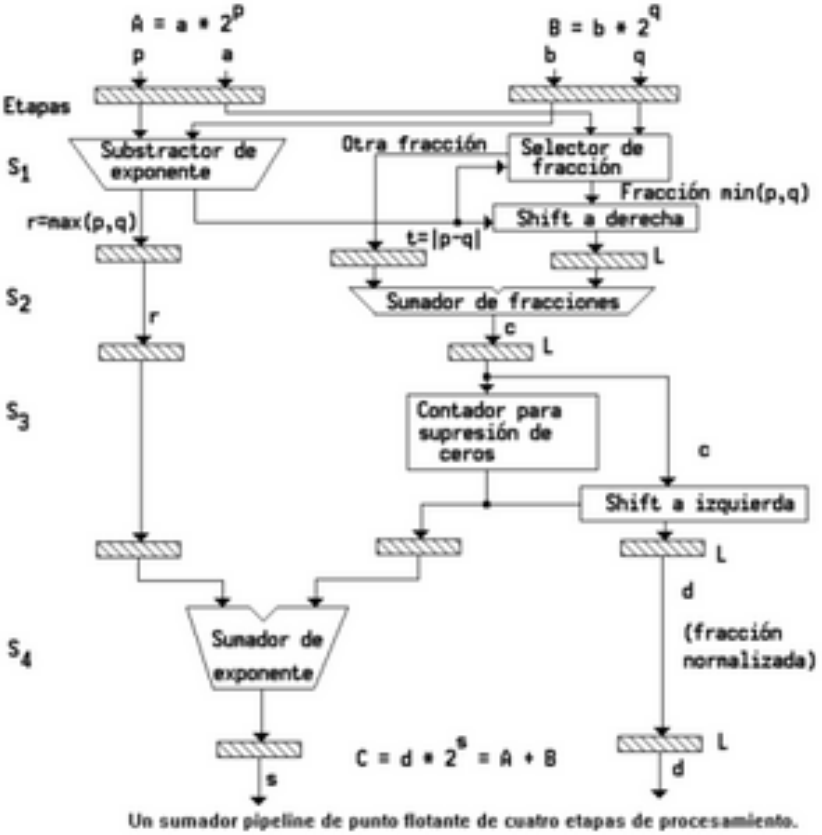
También llamados "pipeline super escalar", está presente en los PC modernos y es un pipeline con un gran número de pequeñas etapas para aumentar el paralelismo para mejorar la performance en la ejecución de instrucciones. Tiene mayor cantidad de stages o etapas, tiene múltiple despacho de instrucciones, es decir, que tiene más de una por ciclo de reloj, tiene ejecuciones fuera de orden, tiene ejecuciones especulativas, es decir, sin saber si se ejecutarán finalmente y tiene avanzados predictores de saltos.

Procesador super – escalar (y paralelismo especial)

Es un procesador con un único núcleo que contiene múltiples unidades de ejecución y múltiples Pipelines de instrucción; pudiendo ejecutar varias instrucciones simultáneamente en un mismo ciclo de reloj. Cada pipeline puede ser general o estar especializado en diferentes tipos de instrucciones, tienen ejecuciones fuera de orden, ejecuciones especulativas, etc. Múltiples pipelines en paralelo implican mantener un flujo de instrucciones continuo para alimentarlos.



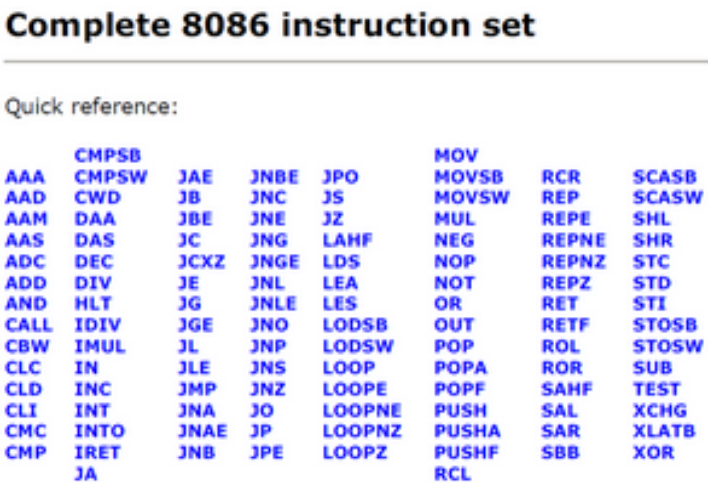
- Pipeline Sumador de Punto Flotante en 4 etapas



Un sumador pipeline de punto flotante de cuatro etapas de procesamiento.

Arquitectura CISC

Significa “COMPLEX INSTRUCTION SET COMPUTER”; tiene un set de instrucciones complejo donde cada una es un conjunto de más de una micro – instrucción que está en la memoria de control de la UDC, posee múltiples modos de direccionamiento, las instrucciones tienen distintos tamaños, tardan distinto ¿?, los registros son de propósito específico y los sistemas que pertenecen a esta arquitectura son IBM System/360, PDP-11, VAX, Motorola 68000 e Intel familia x86.



Complete 8086 instruction set

Quick reference:

	CMPSB				MOV		
AAA	CMPSW	JAE	JNBE	JPO	MOVSB	RCR	SCASB
AAD	CWD	JB	JNC	JS	MOVSW	REP	SCASW
AAM	DAA	JBE	JNE	JZ	MUL	REPE	SHL
AAS	DAS	JC	JNG	LAHF	NEG	REPNE	SHR
ADC	DEC	JCXZ	JNGE	LDS	NOP	REPNZ	STC
ADD	DIV	JE	JNL	LEA	NOT	REPZ	STD
AND	HLT	JG	JNLE	LES	OR	RET	STI
CALL	IDIV	JGE	JNO	LODSB	OUT	RETF	STOSB
CBW	IMUL	JL	JNP	LODSW	POP	ROL	STOSW
CLC	IN	JLE	JNS	LOOP	POPA	ROR	SUB
CLD	INC	JMP	JNZ	LOOPE	POPF	SAHF	TEST
CLI	INT	JNA	JO	LOOPNE	PUSH	SAL	XCHG
CMC	INTO	JNAE	JP	LOOPNZ	PUSHA	SAR	XLATB
CMP	IRET	JNB	JPE	LOOPZ	PUSHF	SBB	XOR
	JA				RCL		

Arquitectura RISC

Significa “REDUCED INSTRUCTION SET COMPUTER”; está basado en el modelo de Harvard, tiene mayor velocidad de ejecución y performance, tiene un set de instrucciones sencillo y con pocas instrucciones, cada instrucción realiza una micro – tarea, no hay memoria de control, por lo que no hay micro – programas, los registros suelen ser de propósito general y las instrucciones suelen demandar un ciclo de reloj, excepto a excepción del acceso a la memoria principal y la resolución de dependencias entre instrucciones. El ciclo de instrucciones; INST. FETCH, INST. DECODE, EXECUTION, MEM, ACCESS y WRITE BACK. Unos ejemplos de sus instrucciones sencillas son:

- LOAD / STORE memoria y registros
  - LD, ST, LDI, STI
- MOVIMIENTO DE DATOS
  - PUSH, POP, SWAP
- CALCULATE para ejecutar operaciones
  - ADD, SUB, MUL, DIV, MOD (módulo), CMP
- MANIPULACION DE BITS
  - , AND, OR, XOR, SHL, SHR
- BRANCH para control de ejecución
  - BEQ (saltar si es igual), BNE (distinto), BGT (mayor que), JMP, NOP
- ESPECIALES
  - Vectores, criptografía,

Hoy en día, las tecnologías CISC y RISC han convergido.

- Ejemplo de un programa RISC escrito en assembler:

```
LOAD R1, [num1_addr]  → Carga desde la memoria el dato num1 en R1
LOAD R2, [num2_addr]  → Carga desde la memoria el dato num2 en R2
ADD R3, R1, R2  → Suma el contenido de R1 y R2 y el resultado lo guarda en R3
STORE R3, [result_addr]  → Guarde el resulado R3 en la memoria
```

- Un programa más complejo: Factorial de un número

```
LOAD R1, 5  → Va a calcular el factorial de 5
LOAD R2, 1  → Inicializa el factorial en 1
Bucle:
    MUL R2, R2, R1  → Multiplica el resultado por el número
    SUB R1, R1, 1  → Decrementa el número
    CMP R1, 0  → ¿El número es cero?
    BEQ bucle  → Si es cero, salta a la etiqueta "done:"
    BR termino  → Si no es cero, salta a la etiqueta "termino:"
Termino:
    STORE R2, [result]  → Guarde el resulado R2 en la memoria
    HALT  → Fin
```

Para los copiladores, cada procesador tiene su set de instrucciones y, por lo general, no hay compatibilidad binaria entre los nuevos con sus predecesores dentro de una misma familia. No es necesario tener una versión diferente de compilador para cada chip RISC pero, se debe informar al compilador en que procesador se ejecutará el binario para que aplique las optimizaciones pertinentes. Las arquitecturas RISC avanzadas implementan Pipelines, ejecución en desorden, ejecución especulativa, etc.

Arquitecturas RISC y CISC

Característica	CISC	RISC
Cantidad de instrucciones en lenguaje máquina	Muchas	Pocas
Cantidad de modos de direccionamiento	Muchos	Pocos
Longitud de instrucciones	Varias	Mayormente fija
Cantidad de ciclos de reloj necesarios para ejecutar cada instrucción	Muchas, más de uno	Uno en casi todas
Instrucciones para acceder a la memoria y registros	Muchas	2, Load y Store
Registros de propósito específico	Si	En general no
Control microprogramado	Si	No

Diagrama en bloques de una CPU

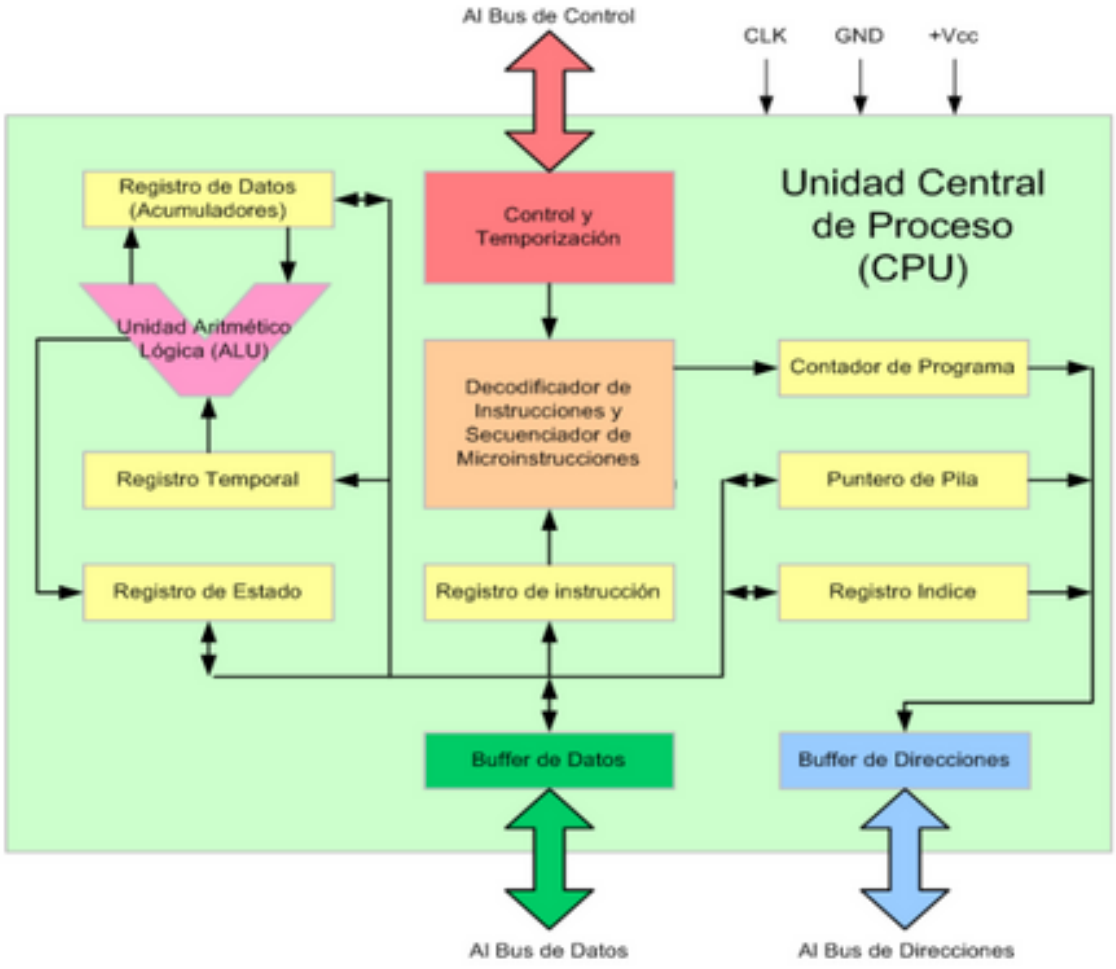
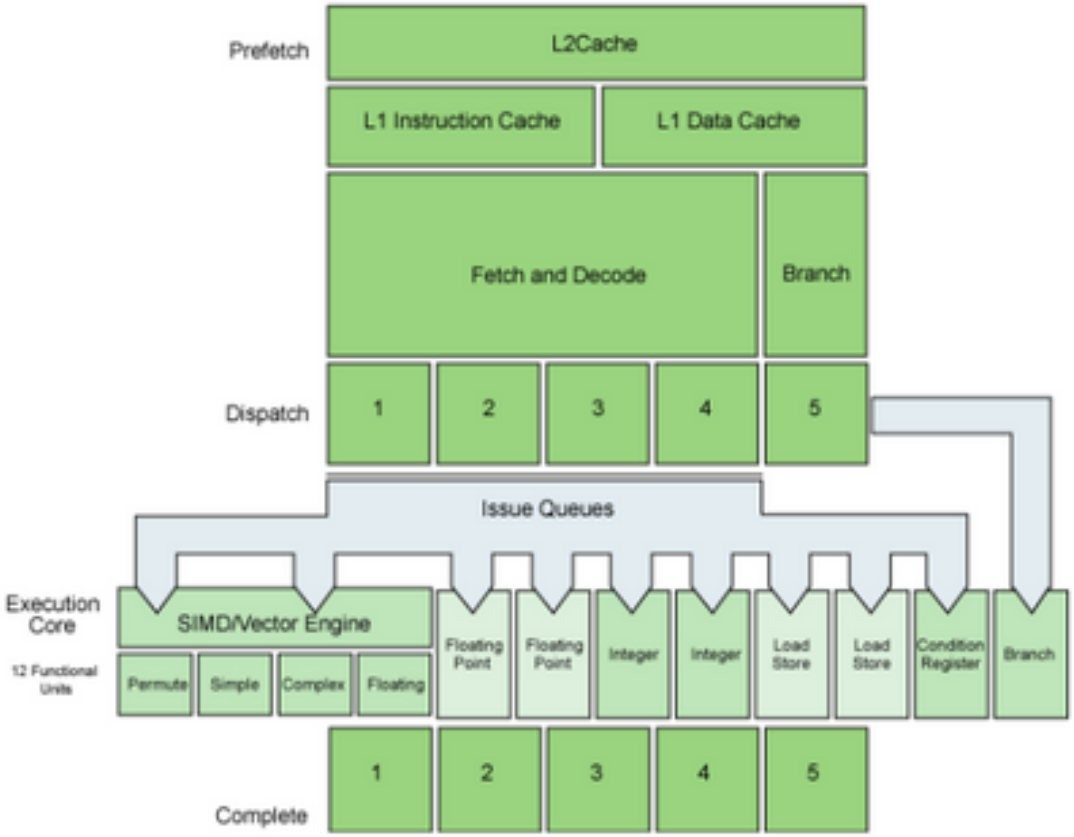


Diagrama PowerPC G5

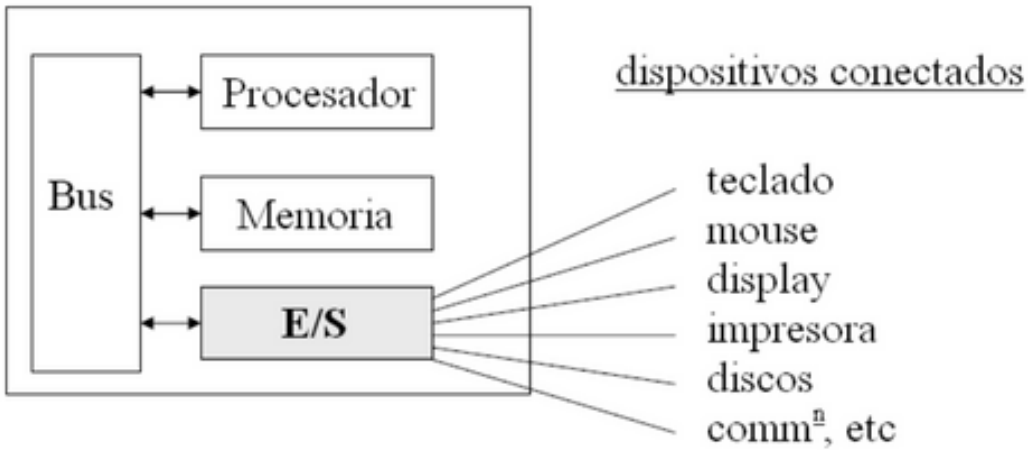


UNIDAD 8: Entrada/Salida, interrupciones, acceso directo a memoria

Entrada/Salida

Es el intercambio de datos entre los buses internos del ordenador, conectados a la CPU y

MP, y los dispositivos externos.



Las funciones de entrada/salida son:

1. Control y temporización: coordinar el tráfico entre los recursos internos y dispositivos externos.

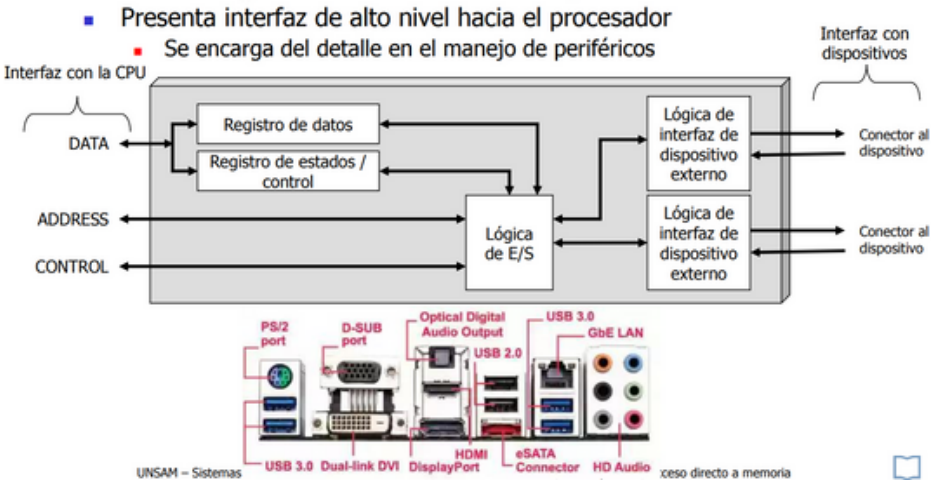
1. Comunicación con la CPU: decodificar órdenes, intercambiar datos, proveer señalización de estados y reconocer direcciones.

1. Comunicación con los dispositivos: similar al punto anterior, pero desde el otro lado: hablar con el dispositivo externo.

1. Almacenamiento temporal de datos: adaptación de velocidades de transferencia de información.

1. Detección de errores: informar errores al procesador usando métodos como bit de paridad o códigos HAMMING para verificar si los datos llegaron bien.

La estructura de un módulo E/S puede contener uno o más registros para intercambio de datos y cada registro tendrá una dirección única y unívoca, es decir, una dirección propia que no se comparte con ningún otro registro.



Clasificación general de periféricos

Tarjeta de expansión:

Es una unidad independiente y conectable que se instala en las ranuras de la placa base para añadir o mejorar funcionalidades:

- Placas gráficas avanzadas.
- Placas de sonido avanzadas.
- Placas de red.
- Placas de conectividad con almacenamiento.
- Placas de expansión USB.
- Placas WIFI.

Chip integrado o en la placa madre:

Es un circuito integrado incorporado, es decir, soldado o insertado, que se encarga de funciones esenciales como asistencia a la CPU o agregar funcionalidades:

- Gráficos integrados en la CPU.
- Sonido integrado.
- Red integrada.
- Almacenamiento integrado.
- BIOS/UEFI.
- Otros (PCH, PCI Express LANES, USB, WIFI/Bluetooth, etc.).

Interacción con periféricos

La programación de los dispositivos de E/S es distinta a la programación de memoria dado que estos operan asincrónicamente de la CPU y del programa en ejecución; para transferir información, el dispositivo y el procesador se deben sincronizar para el intercambio.

Existen dos formas para programar interacciones con dispositivos; la primera es sincrónicamente, esperando a que el dispositivo finalice la operación de E/S, y la segunda, que es más eficiente, es asincrónicamente, disparando la operación de E/S en el dispositivo para que la interfaz de E/S avise cuando se ha concretado.

Puerto E/S

Permite el intercambio de datos entre el bus, que está conectado a la CPU y a la MP, y el componente de E/S, que está conectado a los dispositivos. Un componente de E/S tiene 3 clase de puertos:

Puerto de CONTROL: Se escriben valores que controlan el comportamiento de componente/dispositivo.

Puerto de STATUS: Se leen valores que representan el estado actual del componente/dispositivo.

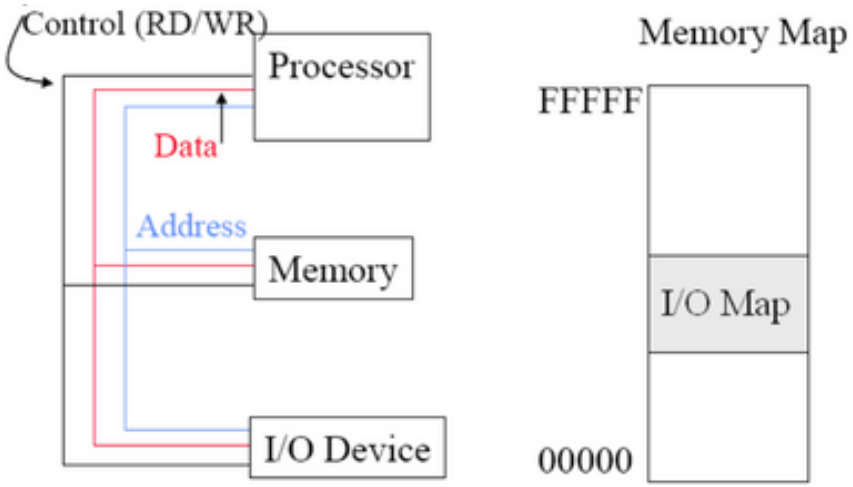
Puerto de DATA: Se leen y escriben valores que hacen al intercambio de la información con el periférico.

Cuando uno o más dispositivos se conectan a un sistema de computación se les identifica asignándole a cada puerto una dirección de E/S. Un dispositivo tiene uno o más puertos de E/S y cada uno de ellos es identificado por una única y unívoca dirección de E/S.

El direccionamiento de puertos E/S es la forma en que la CPU accede a los dispositivos de entrada/salida usando direcciones, al igual que accede a la memoria. En arquitecturas de microprocesadores, existen dos tipos de conexiones de direccionamiento de E/S: ISOLATED I/O y MEMORY-MAPPED I/O.

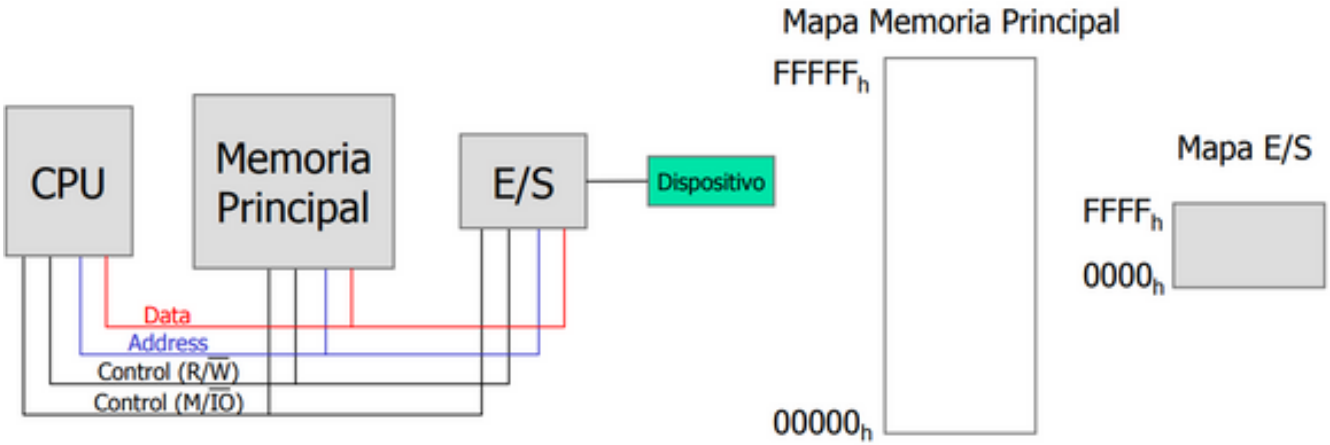
MEMORY-MAPPED I/O (Direccionamiento E/S Mapeada a Memoria):

El procesador utiliza el mismo set de instrucciones para accesos a memoria y para las operaciones de E/S; los dispositivos de E/S y la memoria se encuentran en el mismo espacio de memoria.



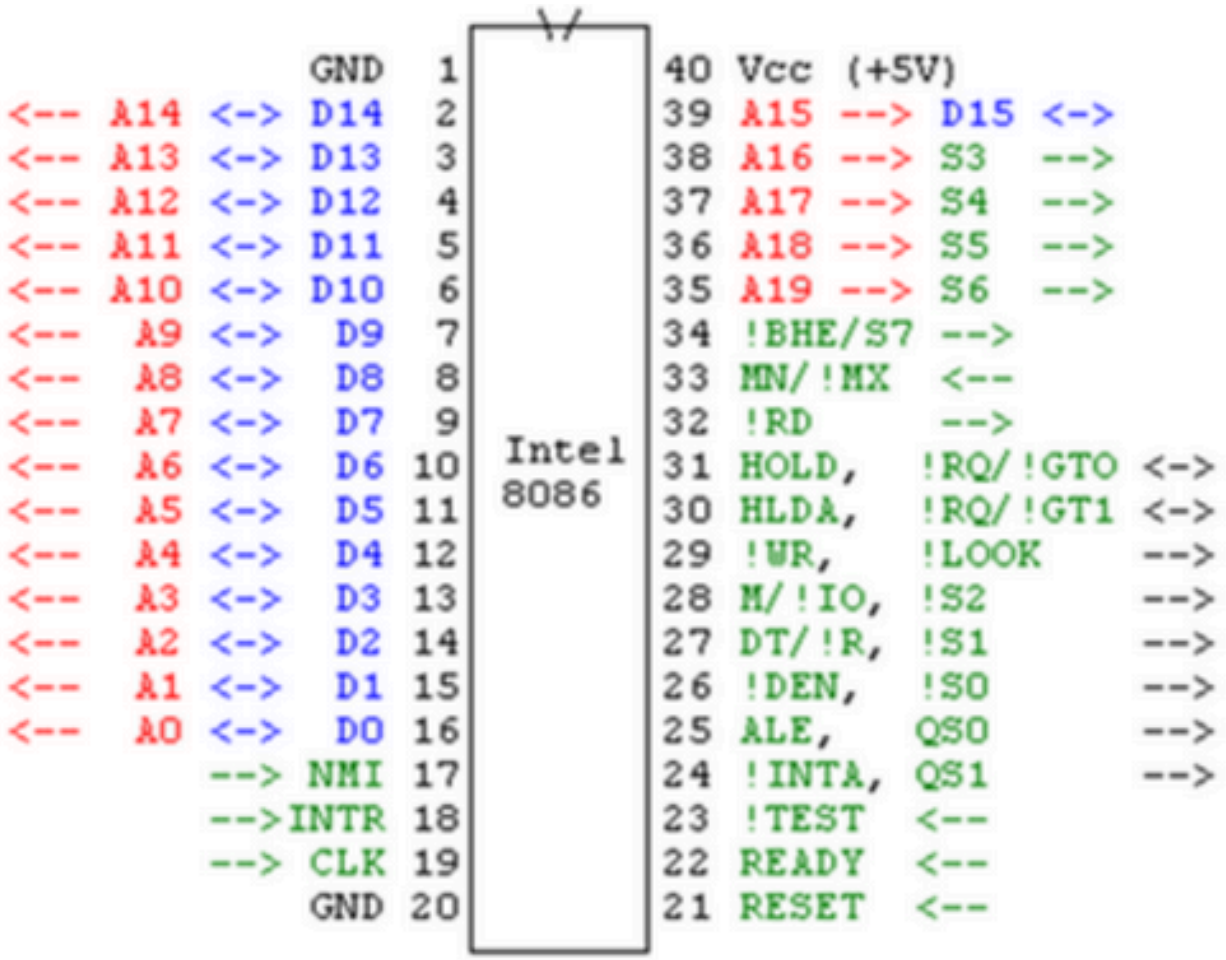
ISOLATED I/O (Direccionamiento E/S Mapeada Aislada):

El procesador tiene instrucciones dedicadas para operaciones de E/S; tiene un espacio de direcciones separado para dispositivos de E/S. En 8086: 20 bits para MP y 16 bits para E/S.



Direccionamiento de E/S por Mapeo Aislado de Intel

Intel usa ISOLATED I/O. Dentro de la familia 80x86, el rango de direcciones para E/S es de 0000h a FFFFh, que son 16 bits. En las PC, los dispositivos tienen asignados direcciones de E/S estándares y usados por todos los fabricantes; 60h para teclado, 61h para el parlante, 3BCh - 3BFh para LPT1.



Existen n instrucciones separadas que transfieren datos desde y hacia los puertos de E/S:

- **Transferencia de Memoria**

  MOV Reg, [Memory]
  MOV [Memory], Reg
  MOV Reg, Reg
  MOV Reg, immediate 1-2 bytes
  MOV Seg-Reg, [Memory]
  MOV [Memory], Seg-Reg
  MOV Seg-Reg, Reg
  MOV Reg, Seg-Reg

- **Transferencia de E/S**

  IN AL/AX, immediate byte
  IN AX/AX, DX
  OUT immediate byte, AL/AX
  OUT DX, AL/AX

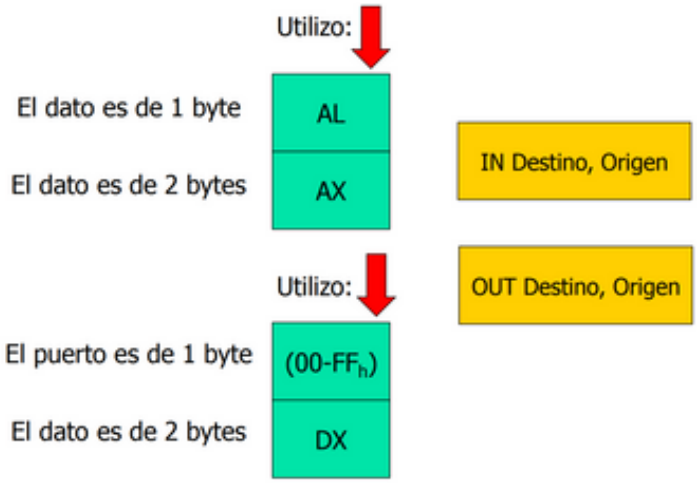
La instrucción IN lee desde un puerto de E/S y al dato lo guarda en AL o AX y la instrucción OUT escribe a un puerto de E/S con el dato contenido en AL o AX.

Sintaxis:

- `IN AL, imm8` → Dato de **8** bits, puerto de **8** bits
- `IN AX, imm8` → Dato de **16** bits , puerto de **8** bits
  - imm8 especifica una dirección de E/S de 8 bits en el rango 00-FF_h
- `IN AL, DX` → Dato de **8** bits, puerto de **16** bits
- `IN AX, DX` → Dato de **16** bits, puerto de **16** bits
  - DX contiene una dirección de E/S de 16 bits en el rango 0000-$FFFF_h$

Sintaxis:

- OUT imm8, AL → Dato de 8 bits, puerto de 8 bits
- OUT imm8, AX → Dato de 16 bits , puerto de 8 bits
- OUT DX, AL → Dato de 8 bits, puerto de 16 bits
- OUT DX, AX → Dato de 16 bits, puerto de 16 bits



POLLING de dispositivos de E/S - Espera Activa

Dentro del programa principal, con saltos condicionados la CPU pregunta, de a un periférico por vez, si requiere atención; si la necesita, ejecuta las acciones que le son solicitadas y si no, continúa preguntando al siguiente dispositivo. Mientras la CPU atiende un dispositivo, ningún otro puede ser atendido independientemente de su importancia; la desventaja es que no permite anidamiento, pero la ventaja es que no requiere hardware adicional ya que se implementa por software.

Interrupción

Cuando es por hardware, es un evento desencadenado por un evento externo que produce una transferencia impredecible del control de ejecución, desde el programa o rutina que es interrumpida hacia otra rutina.

La interrupción provecha los recursos de la CPU de manera más eficiente, se ejecuta en el main y se atiende a los periféricos si estos lo solicitan. Cuando la CPU recibe un pedido de instrucción, finaliza la

ejecución de la instrucción en curso y comienza la ejecución de la rutina de atención de la interrupción. Cuando esta rutina finaliza, el micro continúa donde había dejado.

Interrupción enmascarable

Es un bit que condiciona el reconocimiento de la interrupción, la IF o INTERRUPT FLAG, que si es 0 la solicitud es ignorada y si es 1 la solicitud es aceptada y se inicia el ciclo de reconocimiento.

La INTR o INTERRUPT REQUEST son una solicitud que los dispositivos ponen en 1 cuando quieren ser atendidos y, al final de ejecutar una instrucción, la CPU revisa si hay una señal de interrupción activa o INTR=1; dicho en otras palabras, la señal INTR se censa en el último ciclo de reloj de la instrucción en curso. Cuando INT=1, la CPU pone la señal !INTA o INTA=0 llamada INTERRUPT ACKNOWLEDGE y la IF pasa a 0 hasta que INTR esté en 0; esto último es así porque de lo contrario solo llegaría a ejecutarse la primera instrucción de esta rutina y la CPU comenzaría nuevamente con otro ciclo de reconocimiento. Como se leen 8 bits del tipo, hay 256 tipos distintos que pueden atenderse.

Una excepción es la instrucción WAIT que espera un “0” en la entrada !TEST para continuar con el programa. Esta instrucción muestra continuamente INTR durante su ejecución, permitiendo la atención de interrupciones en el interior de la espera. Cuando finaliza la rutina de atención, la CPU retorna a ejecutar la instrucción WAIT. Explicado con otras palabras (las de chat GPT); La instrucción WAIT hace que la CPU se detenga hasta que la señal !TEST sea 0, pero permite que se atiendan interrupciones mientras espera. Cuando la interrupción termina, la CPU retoma la espera donde la dejó.

Anidamiento

Una interrupción se anida en otra, cuando la primera interrumpe la ejecución de la ISR que de la segunda. Esto puede evitarse si se utiliza el flag IF según corresponda; si no se desea anidamiento, la CPU coloca el IF=0 antes de comenzar la ejecución de la ISR y no se modifica el flag IF y, si sí se desea anidamiento, la primera instrucción de atención a la interrupción tiene que ser la habilitación de atención de nuevas interrupciones colocando el IF=1 (instrucción STI). En ambos casos, al ejecutarse la instrucción IRET presente en la ISR, se restaurarán los flags tal como estaban antes de la interrupción. La atención de una interrupción (ejecución de su ISR) se “anida” dentro de otra cuando ésta última interrumpe la ISR que está en curso.

Interrupción NO enmascarable

Se usa para situaciones graves que requieran atención incondicional de la CPU como errores de paridad o energía baja; su atención es inevitable e incondicional y es siempre la interrupción más prioritaria ya que puede interrumpir la ejecución de cualquier servicio generado por INTR. La solicitud se realiza a través de

la entrada NMI (NON MASKABLE INTERRUPT) y es una entrada asincrónica activa por flanco ascendente. No es necesaria la señal de INTA.

Tabla de vectores de interrupción (IVT)

Es un sector del mapa de memoria, es decir, de todas las direcciones de memoria accesibles para el procesador, donde se almacenan las direcciones donde se encuentran las rutinas de las interrupciones. La IVT se encuentra al principio del mapa y va desde la 00000h a la 003FFh, ocupando 1024 bytes que se dividen en 256 vectores para darle 4 bytes a cada interrupción; dos para el CS (el segmento de memoria donde están las instrucciones del programa o rutina a ejecutar) y otros dos para la IP (el desplazamiento u offset dentro de ese segmento). El primer byte es el LSB del IP, el segundo MSB del IP, el tercer byte es el LSB del CS y el cuarto byte es el MSB del CS. Si un vector no se usa, se deberá apuntar a una ISR que sólo contenga la instrucción IRET para retornar sin hacer nada.

Cuando la señal INTA se activa, se lee un byte del bus de datos que contiene el tipo de la interrupción solicitada, que a su vez es un puntero de uno de los punteros, dentro de la tabla de vectores de interrupción, a donde se encuentra la rutina de atención.

Etapas de una instrucción

En total son cuatro:

- Solicitud: a través de una señal asincrónica de entrada a la CPU; esta última recibe un flanco ascendente en el pin INTR.

- Reconocimiento: se evalúa si la máscara de interrupciones está activada, es decir, si IF=1, y si la microprograma se encuentra en la última microinstrucción, se desactiva la máscara de interrupción poniendo IF=0, se coloca un flanco descendente en el pin !INTA, el contador de programa se encuentra en la próxima instrucción a ejecutarse en el programa interrumpido, guarda el CS, IP y Flags en la pila, Efectúa un ciclo de lectura sobre el bus de datos para capturar el INTERRUPT TYPE, calcula el puntero al vector de interrupciones (IVT), es decir, multiplica la INTERRUPT TYPE por 4, lee en MP la dirección del puntero obtenido y las siguientes 3 y finalmente se carga el CS y el IP en con el contenido del IVT. Dicho de otra manera; se identifica al periférico solicitante a través de su "tipo=n" y se obtiene la dirección de memoria donde está la rutina de atención del periférico. (ISR n = INTERRUPT SERVICE ROUTINE periférico n).

- Atención: comienza la ejecución de la INTERRUPT SERVICE ROUTINE (ISR) en la dirección apuntada por el IVT y cargada en CS e IP.

- Retorno: al ejecutar la instrucción INTERRUPT RETURN (IRET) se restablecen de la pila los valores de los flags, IP y CS guardados previamente, continúa la ejecución de la instrucción siguiente del programa interrumpido.

Pseudo - interrupción:

Por software:

Es una instrucción, es decir que no es impredecible, que provoca que la CPU ejecute una rutina de atención de una interrupción que se indique. Finalizada la ejecución de la rutina de atención, la CPU continúa ejecutando la instrucción posterior a la instrucción de la pseudo - interrupción. Al ser una instrucción, no respeta ningún esquema de prioridades y se anida siempre que se incluyan dentro de una rutina de atención de interrupción. Esta instrucción puede usarse para transferir el control a una rutina que sea reubicable en memoria, es decir, en las que el programa que las llama no conoce la dirección de inicio de las mismas, guarda en la pila las banderas y el PROGRAM COUNTER (CS: IP) y no ejecuta el ciclo de INTA, pero si inhibe Single Step y IF durante su ejecución.

Instrucción INT 3h: se utiliza para insertar puntos de ruptura en rutinas que están siendo depuradas, la instrucción ocupa un byte de longitud y es de tipo=03h.

Instrucción INT XXh: ejecuta el servicio de la interrupción del tipo=XXh, se usa para simular cualquier tipo de interrupción y para depuración y ocupa dos bytes, uno para el código de operación y otro para el para el dato del tipo. Para manejar más de un servicio de una rutina ISR, se usan a los registros para pasar los parámetros, indicando el tipo del servicio, y el valor del retorno depende del servicio invocado.

Instrucción INTO (INTERRUPT ON OVERFLOW): interrumpe el programa si existe un overflow luego de una operación indicado por la activación del flag OF, permitiendo corregir errores en el caso de overflow. Si OF=0, la instrucción no hace nada porque se comporta como un NOP y retorna a la próxima instrucción del INTO, si OF=1, ejecuta la rutina que apunta el IVT en el vector 4. Ocupa un byte, y es del tipo=04h.

Funciones instaladas por el DOS (INT 21h): imprimir un mensaje (AH=09h), salir (AH=4Ch), leer un carácter por teclado (AH=01h), mostrar un carácter por pantalla (AH=02h y DL=carácter) e imprimir carácter (AH=05h y DL=carácter).

Servicios del BIOS: Display (INT 10h), E/S a discos (INT13h), teclado (INT 16h y lectura=0h) e impresoras (INT 17h para funciones de impresoras).

Por error:

Se producen, por lo general, por un error de software y no respetan esquema de prioridades. En el 8086 hay solo del tipo=0 que es el error al dividir; cuando se intenta dividir por 0, la dirección de retorno de la rutina de atención de este tipo apunta a la misma instrucción que se terminó de ejecutar antes de producirse este error, permitiendo corregir el error que provocó la interrupción en la rutina de atención y volver a empezar. No es enmascarable.

Por trampa:

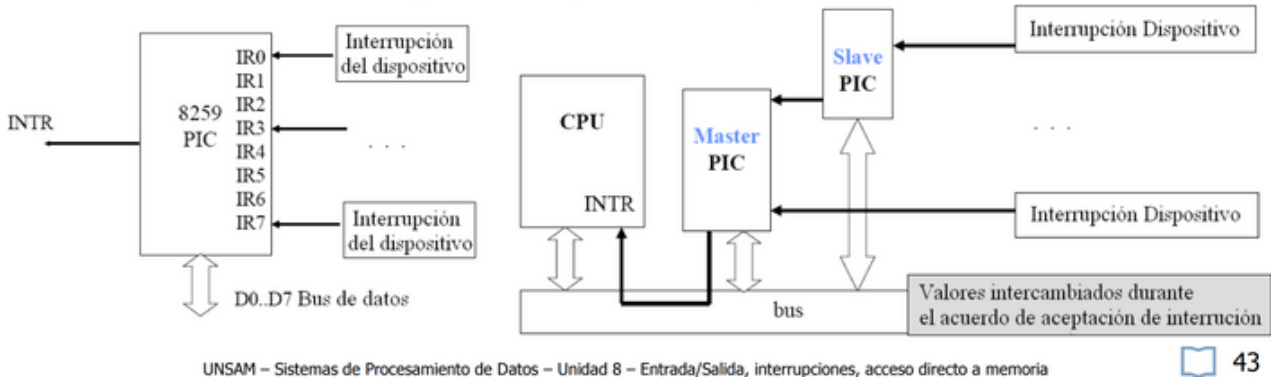
Se usa para implementar una rutina para el monitoreo durante la depuración de programas. Si el bit TF (TRAP FLAG) está activo, después de la ejecución de cada instrucción se produce una interrupción tipo=1; al aceptar esta interrupción, se borra el TF para poder ejecutar la rutina de atención en forma normal. No es enmascarable.

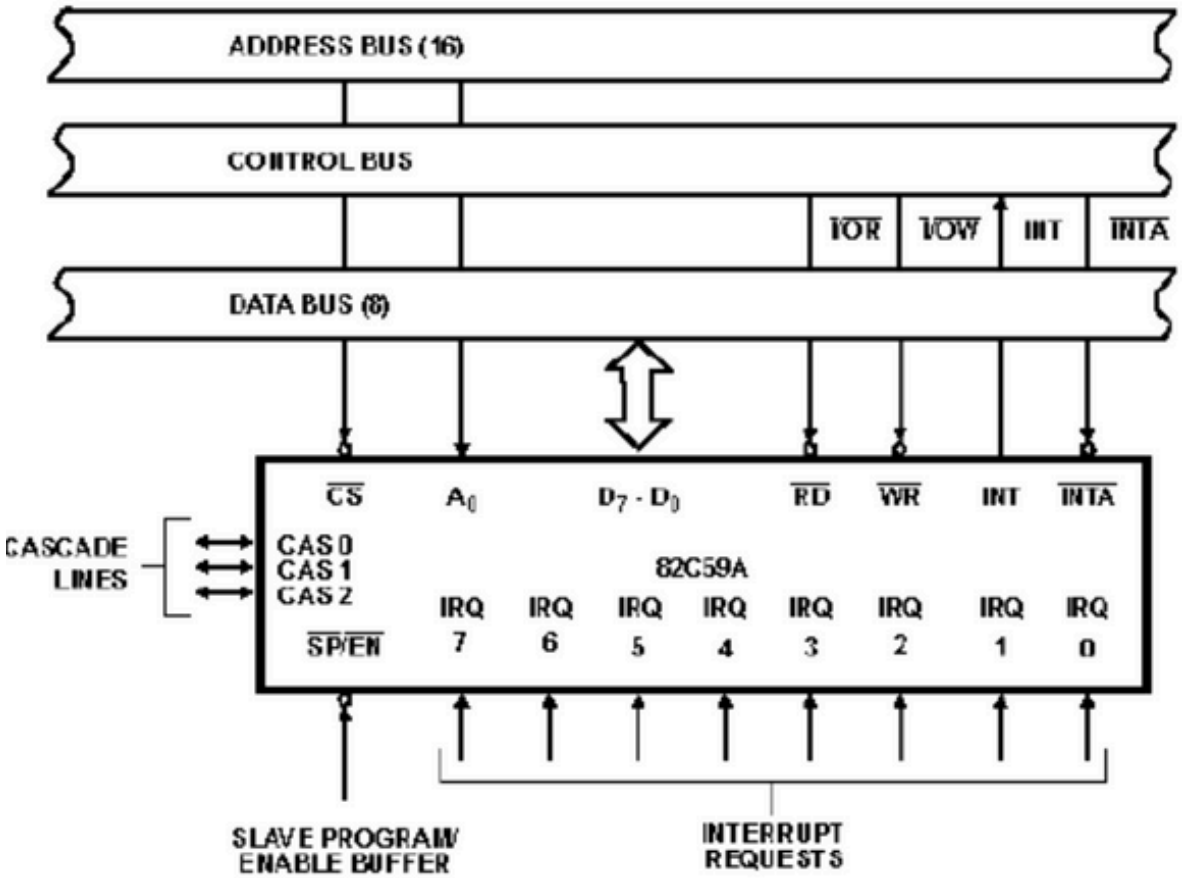
Intel 8259

(PIC = PROGRAMMABLE INTERRUPT CONTROLLER)

Amplía la estructura de interrupciones del 8086, es configurable y programable, asociando tipo a dispositivos y prioridades, y se lo considera como un dispositivo más del chipset. Conectándolo 8 8259A

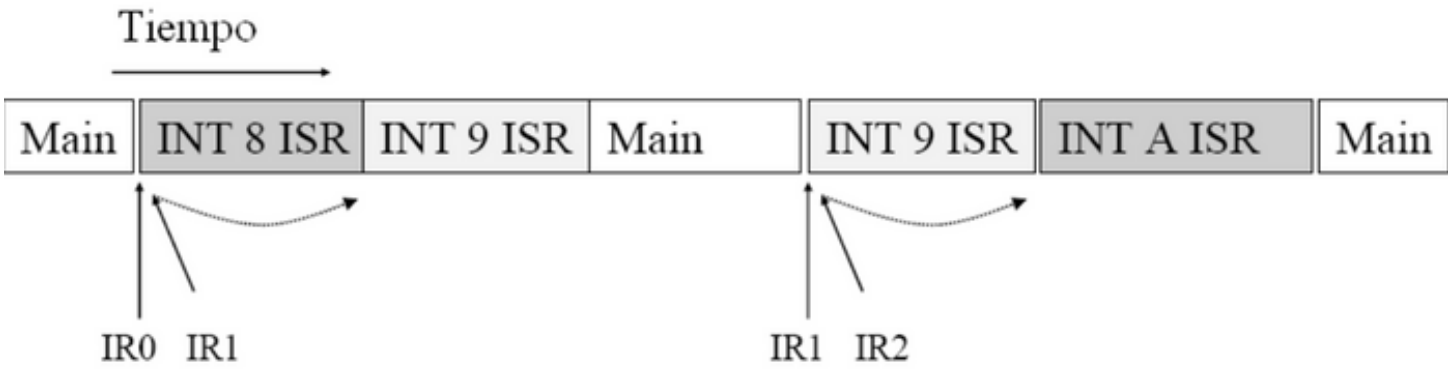
en cascada (8+1) se maneja 64 interrupciones. Durante el booteo, la BIOS programa el PIC maestro y desde IR0 a IR7 se mapean a los tipos de interrupciones 08h a 0Fh.



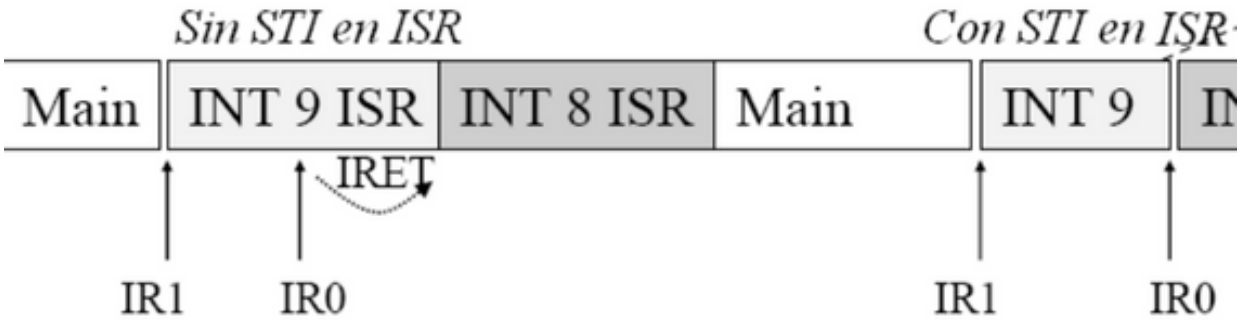


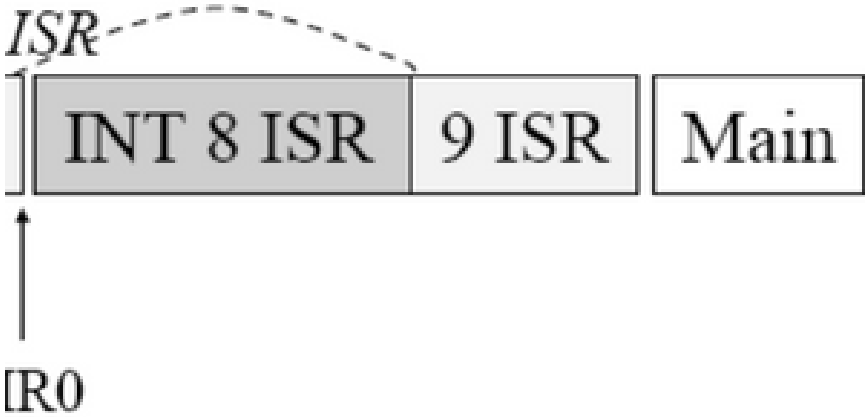
Los dispositivos tienen asignados prioridades que se tienen en cuenta cuando varias interrupciones suceden en el mismo tiempo o nuevas interrupciones ocurren mientras se está procesando la rutina ISR de interrupciones previas; quien hace cumplir estas prioridades es el controlador de interrupciones, teniendo en cuenta que, desde la IR0 hasta la IR7, el número más bajo es la prioridad más alta y el número más bajo la prioridad más baja. En PC los dispositivos tienen conexiones preconfiguradas hacia el controlador y en el DOS se programa al 8259A para asignar prioridades basadas en las conexiones de los dispositivos.

La prioridad de interrupción establece qué rutina ISR se ejecuta si dos interrupciones ocurren al mismo tiempo.

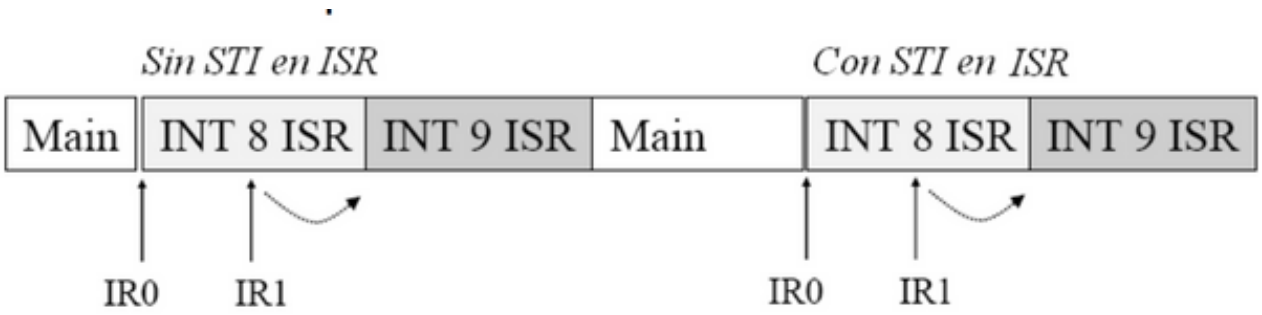


Establece qué ISR sigue ejecutando, si la CPU ya está ejecutando una rutina ISR y un segundo dispositivo de mayor prioridad interrumpe. El controlador intentará permitir a una interrupción de mayor prioridad interrumpir otra de menor. La segunda interrupción no será reconocida por el procesador sino hasta que se habiliten las interrupciones, es decir, IF=1.





Si una interrupción de baja prioridad sucede a otra de alta prioridad, el controlador mantiene la prioridad.



El controlador recuerda la interrupción de baja prioridad hasta que la de alta prioridad haya terminado. Cuando terminó, el controlador genera otra interrupción a favor del dispositivo pendiente. Se llama pendiente a una señal de interrupción que es almacenada en un LATCH, pero no ha sido reconocida por el procesador Las interrupciones pueden estar pendientes en el controlador o en el dispositivo.

Para recordar una interrupción pendiente se usa un registro interno del Intel 8259A de ocho bits donde cada uno de estos representa una entrada IR, cuando la interrupción es reconocida el bit asociado se apaga.

Después de enviar una interrupción al procesador, el controlador necesita saber cuándo es seguro generar una interrupción de menor prioridad, por lo que se necesita de un feedback desde la CPU indicando el fin de la interrupción en curso; se trata de un comando software, que el programa debe enviar (ICW, OCW), no es parte del ciclo INTA, y no es hecho por el hardware.

El modelo del programador o controlador de interrupciones es un dispositivo de E/S, dos puertos de 8 bits cada uno y tiene un INTERRUPT MASK REGISTER (INT 21h) de lectura/escritura que permite habilitar o deshabilitar interrupciones individuales en el controlador. Si el bit que representa el IR es 1, se ignora la interrupción porque está enmascarado, pero si IR=0, entonces es habilitado; es al revés del IF (INTERRUPT FLAG).

El COMMAND REGISTER (Puerto 20h) es de escritura solamente; se escribe en 20h para informar al controlador que terminó la interrupción.

El teclado del PC usa interrupciones; usa el POLLING del puerto de estado para detectar si hay una tecla presionada pendiente de ser enviada y está conectado al IR1 del controlador, a través del chip 8255.

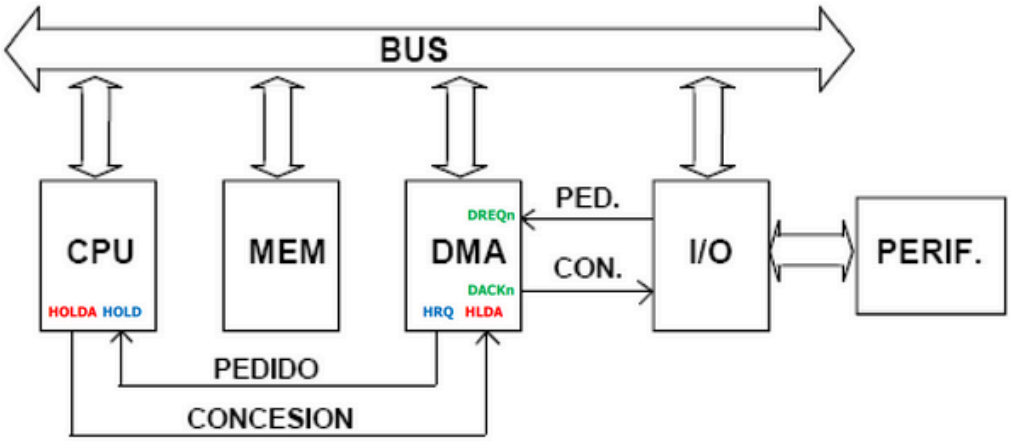
El 8255 PORGRAMABLE PERIPHERAL INTERFACE tiene dos puertos; el Data Port (Port PA) cuya dirección de E/S=60h y la lectura=tecla presionada y el Control Port (Port PB) cuya dirección de E/S=64h

y la lectura=(Buffer con teclas presionadas).

Acceso directo a memoria (DMA)

Es una técnica que permite a los periféricos realizar transferencias sobre la memoria sin la intervención de la CPU, la cual está con otras tareas. En el momento en que el DMA realiza las transferencias, la CPU se desconecta de los buses (3-state) y cede el control. La "petición de buses" está controlada por la CPU en el pin HOLD y concede la operación a través del pin HOLDA (HOLD ACKNOWLEDGE) una vez que terminó el ciclo de bus en curso. El dispositivo que controla los buses se lo denomina "Master" y el que es controlado, "Slave"; en este caso, la CPU deja de ser el Master de los buses.

La DMAC 8237A es el Controlador de Acceso Directo a Memoria Intel 8237ª, posee cuatro canales programables en tres modos diferentes y cada modo en tres tipos de transferencia, permite transferencias de memoria a memoria, realiza direccionamiento por incrementos o decrementos y es expandible a n canales DMA. Las señales principales son; DREQ que es la solicitud de acceso DMA del dispositivo, HRQ CPU que es la solicitud de buses a la CPU (HOLD), HLDA que es la aceptación de la CPU (HOLDA) y DACK que es el otorgamiento del servicio DMA al dispositivo.



La DMAC tiene diferentes modos de operación:

Modo Simple (Single Transfer Mode):

Se realiza una única transferencia y se asegura que la CPU complete un ciclo antes de realizar una nueva transferencia. El "ROLLOVER" del Terminal Count (de 0000h a FFFFh) indica el fin de la transferencia. En este momento, se genera un pulso en End-OFPROCESS (!EOP), pin bidireccional.

Modo En Bloques (Block Transfer Mode):

Se realizan transferencias de bloques hasta el ROLLOVER del TC o hasta que se detecte un EOP externo.

Modo A Demanda (DEMAND TRANSFER MODE):

La transferencia se realiza mientras DREQ permanece activo o hasta el ROLLOVER del TC o un EOP externo. Se pueden realizar todas las transferencias necesarias hasta agotar las posibilidades del dispositivo.

La conexión en Cascada (CASCADE MODE) permite la interconexión de más de un 8237A, permitiendo la propagación de señales de petición y su correspondiente prioridad.

Los tipos de transferencias son de lectura (READ) que es un movimiento de datos desde la memoria a la E/S, de escritura (WRITE) que es un movimiento de datos desde la E/S a la memoria y de verificación (VERIFY) que se generan direcciones, pero no se utilizan las señales de control, por lo que son pseudo – transferencias.

En las transferencias de memoria – memoria se emplean los canales 0 y 1; se realiza un pedido por DMA normal, por canal 0 y se lee dato desde la memoria y se guarda en el 8237A y por canal 1 se escribe en la memoria el dato guardado en el 8237A.

En la auto inicialización se restablecen los registros internos a los valores predeterminados.

Las prioridades son establecidas de manera fija o rotativa; en fija el (canal 0) = máxima y (canal 3) = mínima, de manera rotativa el último canal servido es el de menor prioridad para evitar monopolizar la atención.

El controlador DMA 8237A tiene multiplexada la parte alta del bus de direcciones y los 8 bits más significativos de la dirección son guardados en un latch externo a través del bus de datos (señal AEN,

ADDRESS ENABLE).

Los registros que usa el DMA 8237A son; CURRENT ADDRESS, CURRENT WORD, BASE ADDRESS Y BASE WORD COUNT, COMMAND, MODE, REQUEST, MASK, STATUS Y TEMPORARY.

El control de transferencias en el bus del 8086 tiene dos modos:

Modo mínimo:

Este modo se indica con la señal MN/!MX en “1” (Minimum Mode). Usa las señales HOLD y HOLDA, usa M/!IO para indicar direccionamiento a MP o E/S, DT/!R para indicar transmisión o recepción de datos y !DEN para habilitar el buffer de datos 8286/7.

Modo máximo:

Este modo se indica con la señal MN/!MX en “0” (MAXIMUM MODE). Se usa para conectar un coprocesador Intel 8087 y se utiliza el controlador de buses Intel I8288.