

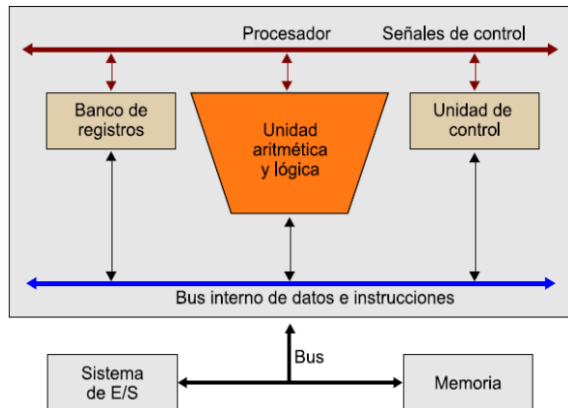
ORGANIZACIÓN DE COMPUTADORAS

ARQUITECTURA DE VON NEUMANN. Describir los elementos de la IAS. ¿Qué diferencia hay con las actuales?

La arquitectura de Von Neumann está compuesta por 4 elementos principales:

- Una *Unidad Central de Procesamiento* (CPU).
- Una *Memoria Principal* (MP).
- Un *Módulo de Entrada/Salida* (E/S).
- Un *Sistema de Interconexión* para estos procesos.

La característica principal es que la CPU está conectada a una única memoria donde se guardan conjuntamente, datos e instrucciones. La memoria es accesible por posición, independientemente si se trata de datos o instrucciones, y la ejecución de las instrucciones se realiza de forma secuencial.



ELEMENTOS DE LAS "IAS"

El computador IAS consta de una estructura de:

- *Unidad Aritmética Lógica* (ALU), capaz de hacer operaciones con datos binarios.
- *Unidad de Memoria*, donde se almacenan los datos e instrucciones.
- *Unidad de Control* (UC), que dirige las operaciones.
- *Un equipo de Entrada/Salida* (E/S), dirigido por la UC que provee las instrucciones y datos (entrada) y envía los resultados (salida).

DIFERENCIAS CON LAS COMPUTADORAS ACTUALES

Las diferencias principales con las computadoras actuales son que estas cuentan con una ALU para punto flotante, contienen más registros (algunos contienen direcciones y otros datos) y cuentan con memoria cache.

TEOREMA FUNDAMENTAL DE LA NUMERACION

El teorema fundamental de la numeración relaciona una cantidad expresada en cualquier sistema de numeración posicional, con la misma cantidad expresada en el sistema decimal, y establece que el valor decimal de una cantidad expresada en otro sistema de numeración, está dado por la fórmula:

$$N^{\circ} = \sum_{i=-m}^n (\text{digito})_{\text{sub } i} \times (\text{base}) \text{ elevado a } i$$

N° es el valor decimal de una cantidad expresada en base B y con $(n+1+m)$ dígitos en posiciones i .

- n : número de dígitos de la parte entera.
- m : número de dígitos de la parte fraccionaria.

- i: indica la posición del dígito respecto de la coma.
- base: es la base del sistema de numeración (2 para binario, 16 para hexadecimal, etc.)

CIRCUITO COMBINATORIO Y CIRCUITO SECUENCIAL

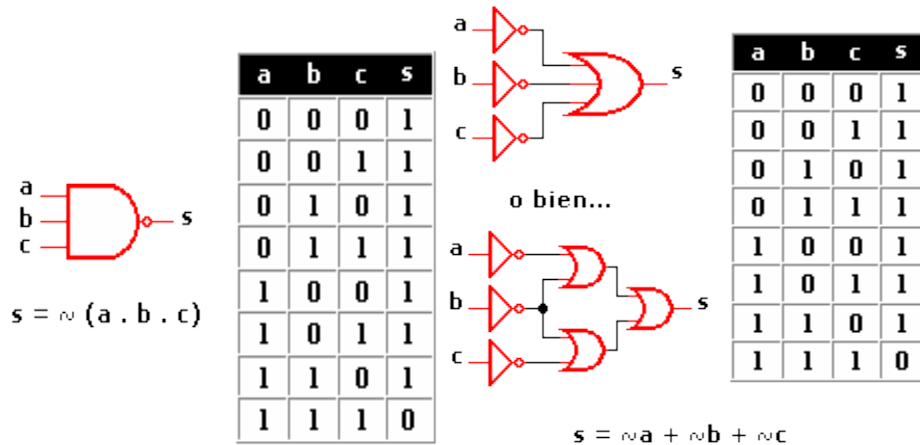
Un CIRCUITO COMBINATORIO es un conjunto de puertas lógicas interconectadas entre sí, cuya salida en un momento dado, es función solamente de la entrada en ese instante. La aparición de la entrada viene seguida casi inmediatamente por la aparición de la salida, con solo retardos de las puertas. En general, un circuito combinatorio consiste en n entradas y m salidas. Ejemplos son: Multiplexor, Demultiplexor, Codificador, Decodificador y Sumador.

En cambio, en un CIRCUITO SECUENCIAL la salida en cualquier instante de tiempo T depende de las entradas en ese instante y del valor de las entradas en $T-1$. Se las puede clasificar como asíncronos (con retardos asociados a las puertas lógicas) o síncronos (solamente se permiten cambios de estados en instantes marcados por el reloj). Ejemplos son: Biestables RS, JK, D, Registros y Contadores.

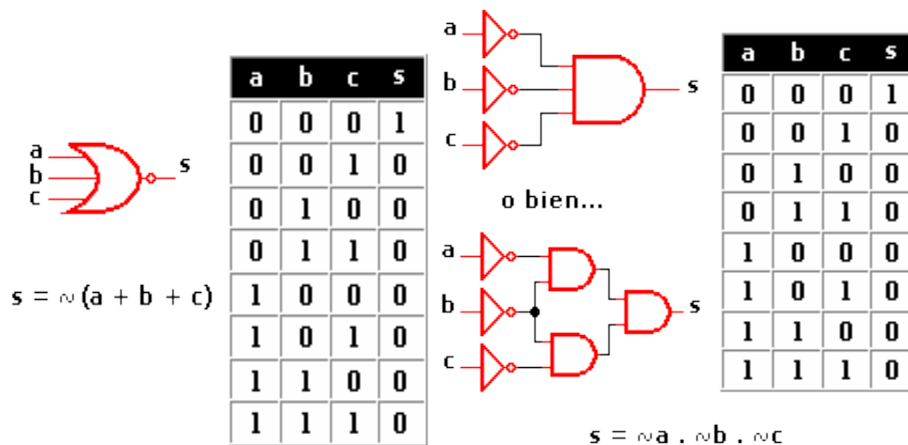
LEYES DE MORGAN. EXPRESARLAS UTILIZANDO COMPUERTAS NAND, NOR Y/O NOT

Las leyes de Morgan dicen que:

- 1) El producto lógico negado de varias variables lógicas es igual a la suma lógica de cada una de dichas variables negadas. $\rightarrow \sim(A \cdot B) = \sim A + \sim B$

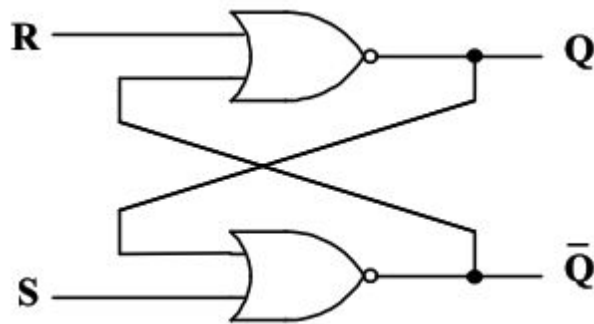


- 2) La suma lógica negada de varias variables lógicas es igual al producto de cada una de dichas variables negadas. $\rightarrow \sim(A+B) = \sim A \cdot \sim B$



PORQUE UN FLIP-FLOP SR NO DEBE RECIBIR VALORES DE ENTRADAS PARA R Y S CON VALOR CERO

No debe recibir valores de entrada iguales a cero porque de esa manera no se producirán cambios en la salida.



S	R	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	Error! (Inconsistencia)

FLIP-FLOP JK. DESCRIBA LAS CARACTERÍSTICAS DE FUNCIONAMIENTO CON TABLA DE COMPORTAMIENTO Y GRÁFICO DEL CIRCUITO LÓGICO.

Es un circuito secuencial, por lo tanto, la salida depende de la entrada como del estado interno del circuito. Es un tipo de flip-flops que tiene dos entradas al igual que el SR, pero todas las combinaciones posibles son válidas. La entrada J sola realiza la función de puesta a 1, causando que la salida sea 1. La entrada K solo realiza la función de puesta a 0. Cuando J y K son 1 la función realizada es de conmutación: la salida se invierte.

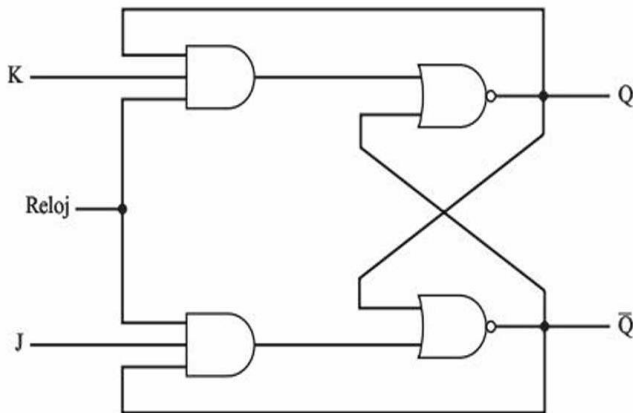
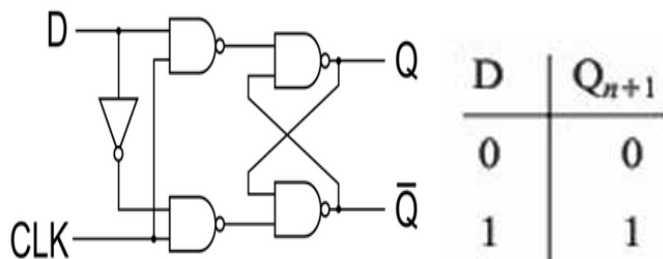


TABLA DE LA VERDAD FF-JK		
J	K	Q
0	0	EA
0	1	0
1	0	1
1	1	\neg EA

Donde: (EA) es estado anterior y (\neg EA) es negación de estado anterior.

FLIP-FLOP D.

Un Biestable D es un circuito digital, también denominado 'Biestable de datos' porque sirve en efecto para almacenar 1 bit de datos. Posee una sola entrada D (Data) y la salida Q, obtiene el valor de la entrada D cuando la señal del reloj se encuentra activada.



D	Q_{n+1}
0	0
1	1

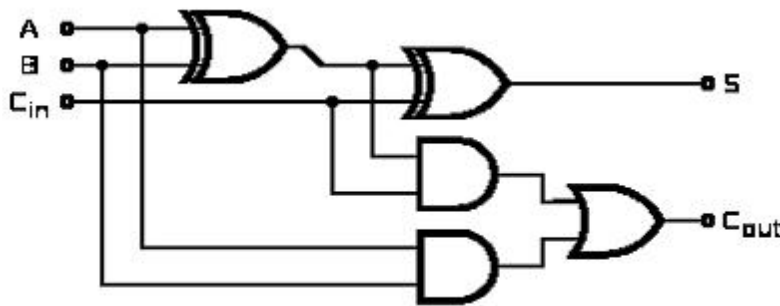
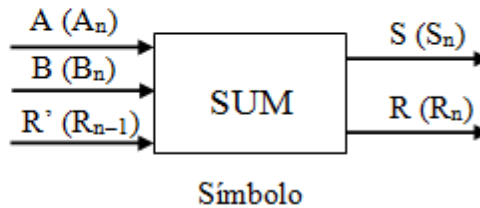
SUMADOR BINARIO COMPLETO. FUNCIONES LOGICAS. TABLA DE VERDAD.

El sumador completo es un circuito combinacional que consiste en 3 entradas y 2 salidas. A las 3 entradas las designamos A, B, R' y a las 2 salidas S y R. La salida S vale 0, si todos los bits de entrada son ceros, en cambio, S es igual a 1 solamente si una entrada es igual a 1 o cuando las 3 entradas son 1. Por otra parte, la salida R tiene un bit de arrastre de 1 si dos de las tres entradas son iguales a 1.

Entradas			Salidas	
A	B	R'	R	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

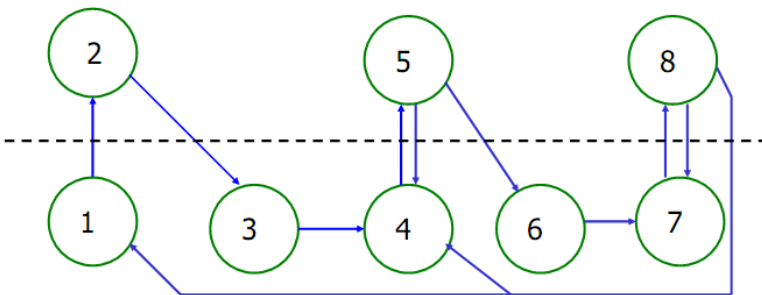
$$\text{Suma: } S = (A \oplus B) \oplus R'$$

$$\text{Arrastre: } R = (A \oplus B) R' + AB$$



CICLO DE INSTRUCCIÓN

El ciclo de instrucción es el procedimiento requerido para una sola instrucción. Se puede decir que un ciclo de instrucción incluye subciclos: captación o búsqueda, interpretación y ejecución e interrupción. La ejecución del programa se detiene sólo si la computadora se desconecta, se produce un error o se encuentra una instrucción que la detiene. La secuencia exacta de eventos que tiene lugar durante un ciclo de instrucción depende del diseño de la CPU.



- Los estados en la parte superior implican un intercambio entre la CPU y la memoria o Entrada/Salida.
- Los estados en la parte inferior implican sólo operaciones internas en la CPU.

1. Cálculo dirección instrucción (Instruction Address Calculation): determina la dirección de la siguiente instrucción a ejecutarse
2. Búsqueda instrucción (Instruction Fetch): lee la instrucción de su posición de memoria a la CPU.

3. Decodificación de la instrucción (Instruction Operation Decoding): analiza la instrucción para determinar el tipo de operación a realizar y los operandos que se usarán.
4. Cálculo dirección operando (Operand Address Calculation): si la operación implica la referencia a un operando en la memoria o E/S, entonces se determina la dirección.
5. Búsqueda del operando (Operand Fetch): busca el operando en la memoria o E/S.
6. Operación sobre los datos (Data Operation): ejecuta la instrucción.
7. Cálculo dirección resultado (Operand Address Calculation).
8. Almacenamiento resultado (Operand Store).

CARACTERISTICAS DE MAQUINAS QUE EJECUTAN 1 INSTRUCCION

- Existe solo un registro de uso general, el acumulador.
- Instrucciones más primarias, lo que requiere una CPU más compleja.
- Instrucciones más cortas.

CARACTERISTICAS DE MAQUINAS QUE EJECUTAN 2 O MAS INSTRUCCIONES

- Mayor tiempo de ejecución.
- Programas más largos y complejos
- Se dispone de múltiples registros de uso general.
- Flexibilidad y facilidad para utilizar varios registros.

CONJUNTO/REPERTORIO DE INSTRUCCIONES. ASPECTOS A TENER EN CUENTA PARA ARMARLO

Los más importantes entre dichos aspectos fundamentales de diseño son:

- El repertorio de operaciones: cuántas y qué operaciones considerar, y cuán complejas deben ser.
- Los tipos de datos: los distintos tipos de datos con los que se efectúan operaciones.
- Los formatos de instrucciones: longitud de la instrucción (en bits), número de direcciones, tamaño de los distintos campos, etc.
- Los registros: número de registros del procesador que pueden ser referenciados por las instrucciones, y su uso.
- El direccionamiento: el modo o modos de direccionamiento mediante los cuales puede especificarse la dirección de un operando.

DESCRIBA LOS DIFERENTES PASOS DEL CICLO SI LA INSTRUCCIÓN A REALIZAR ES XOR AL, 55H (OR EXCLUSIVO ENTRE REGISTRO Y OPERANDO INMEDIATO). SI AL CONTENÍA 5AH, ¿QUE RESULTADO QUEDA EN AL LUEGO DE LA EJECUCIÓN DE LA INSTRUCCIÓN?

Los pasos de la instrucción XOR AL, 55H son:

1. Busca el código de operación.
2. Incrementa el PC.
3. Decodifica la instrucción y detecta que tiene que ir a buscar la otra parte.
4. Busca la otra parte del código de operación.
5. Incrementa el PC.
6. Decodifica la instrucción y detecta que es una operación lógica XOR, por lo que tendrá que usar la ALU.
7. Busca el operando que en este caso es el 55H.
8. Incrementa el PC.
9. Realiza la operación XOR entre el valor almacenado en AL y el 55H.
10. Almacena el resultado en AL.

El resultado que quedaría en AL sería 00001111 = 0FH

DESCRIBA PASO A PASO EL CICLO DE INSTRUCCIÓN CORRESPONDIENTE A LA EJECUCIÓN JMP MEMO1. DICHA INSTRUCCIÓN OCUPA 3 BYTES EN MEMORIA. LA MEMORIA ALMACENA PALABRAS DE 8 BITS Y DIRECCIONA CON 16 BITS.

1. Busca el código de operación
2. Incrementa el Program Counter
3. Decodifica y detecta que se trata de un salto
4. Busca la parte baja de la dirección a la que tiene que saltar
5. Incrementa el PC
6. Busca la parte alta de la dirección a la que tiene que saltar
7. Incrementa el PC
8. Carga el PC con el valor de la dirección a la que tiene que saltar

¿QUE OPERACIONES DE TIPO ARITMÉTICO PODEMOS ENCONTRAR EN UN REPERTORIO DE INSTRUCCIONES? ¿PORQUE LOS DISTINTOS TIPOS DE DATOS QUE DESEEN UTILIZARSE EN ESAS OPERACIONES CONDICIONAN EL HARDWARE NECESARIO PARA OPERAR?

El conjunto de operaciones de tipo aritmético que podemos encontrar son la suma, resta, multiplicación, división, incremento, decremento, valor absoluto y negación.

Los tipos de datos condicionan al hardware por una serie de cosas:

- Si se está diseñando un sistema que valla a multiplicar muchos números sería conveniente sería conveniente incluir un multiplicador por hardware.
- Si se va a utilizar números muy grandes debe disponerse de una cantidad adecuada de bits para los buses en general, memoria, CPU, etc. Por ejemplo, una máquina de 4 bits no se podría utilizar para un software que trabaja con números de miles de millones.

¿EN QUÉ MOMENTO DEL CICLO DE INSTRUCCIONES SE FIJA LA CPU SI HAY PEDIDO DE INTERRUPCIONES? ¿POR QUÉ? DESCRIBA LOS PASOS QUE SE LLEVAN A CABO CUANDO SE ENCUENTRA EL PEDIDO.

Después de la ejecución de la instrucción.

Pasos

1. Suspende la ejecución del programa en curso y guarda su contexto. Esto significa almacenar la dirección de la siguiente instrucción a ejecutar y cualquier otro dato relacionado con la actividad en curso del procesador.
2. Carga el PC con la dirección de comienzo de una rutina de gestión de interrupción.

A continuación, el procesador prosigue con el ciclo de captación y accede a la primera instrucción del programa de gestión de interrupción, que dará servicio a la interrupción. Generalmente el programa de gestión de interrupción forma parte del sistema operativo. Normalmente, este programa determina el origen de la interrupción y realiza todas las acciones que sea necesarias. Cuando la rutina de gestión de interrupción se completa, el procesador puede proseguir la ejecución del programa de usuario en el punto en el que se interrumpió.

MODOS DE DIRECCIONAMIENTO. QUE SON Y PARA QUE SIRVEN

Los modos de direccionamiento son las diferentes maneras de especificar, referenciar o expresar la ubicación de los operandos dentro de una instrucción. Prácticamente todas las arquitecturas de computadoras ofrecen varios modos de direccionamiento. Los principales objetivos son: disminuir la cantidad de bits en la instrucción y el manejo más eficiente de datos. Los modos más comunes son inmediatos, directos, indirectos, relativos, implícitos (de pila).

Inmediato: El operando se obtiene automáticamente de la memoria al mismo tiempo que la instrucción. No requiere una referencia extra a memoria.

La desventaja es que el tamaño del operando está limitado por el tamaño del campo de direccionamiento.

Directo: El campo de dirección tiene la dirección efectiva del operando. Es simple, pero tiene un espacio limitado de direcciones por cantidad de bits del campo.

Por registro: Conceptualmente es igual al Directo, pero se especifica un registro en lugar de una posición de memoria. La referencia a registro usa menos bits que la especificación de la dirección y no requiere acceso a memoria de datos. La desventaja es que son pocos.

Indirecto por memoria: En la instrucción está la dirección de la dirección del operando. Trata de solucionar el problema del Directo. Así, con una dirección de menos bits en la instrucción, se apunta a una dirección de mas bits.

La ventaja es que es espacio de direccionamiento es mayor. La principal desventaja es que realiza múltiples accesos a memoria.

Indirecto por registro: En la instrucción se especifica el registro que tiene almacenada la dirección. La ventaja de esto es que necesita menos bits para especificar el registro que la posición de memoria.

Por desplazamiento: Combina capacidades de Indirecto y Directo. Requiere que la instrucción tenga dos campos de dirección. Estos dos campos se suman para producir la dirección efectiva. Los más comunes son Relativo, De registro base e Indexado.

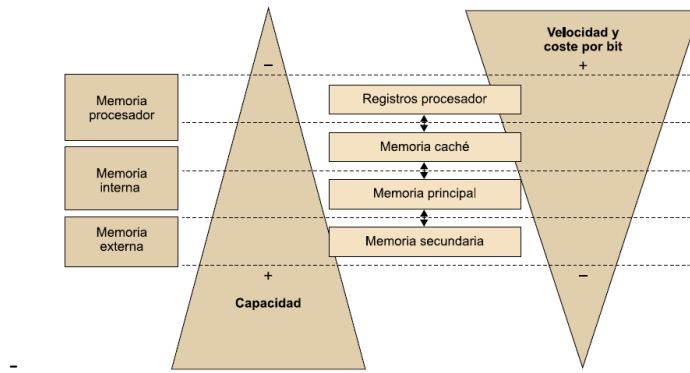
Del Stack o Pila: El stack o pila es un arreglo lineal de localidades de memoria. Es una lista o cola donde el último en entrar es el primero en salir. Es una zona de memoria reservada.

Asociado con la pila o stack hay un registro apuntador (SP) cuyo valor es la dirección tope de pila o stack.

JERARQUIA DE MEMORIA. PORQUE ES ADECUADO USARLO

El objetivo en el diseño del sistema de memoria, es obtener gran capacidad y un tiempo de acceso reducido con el precio más bajo posible. Como no existe ninguna tecnología que cumple simultáneamente estos requisitos, la memoria de una computadora se estructura en varios niveles con el objetivo de conseguir mejores prestaciones, y forma lo que se denomina Jerarquía de Memoria.

En una Jerarquía de Memoria se utilizan varios tipos de memorias, con distintas características de capacidad, velocidad y coste, divididas en diferentes niveles (Memoria del procesador, memoria interna y memoria externa). Cada nivel de la Jerarquía se caracteriza también por la distancia a la que se encuentra del procesador. Los primeros niveles son los más próximos y esto es así porque también son los niveles con velocidad más elevada.



PRINCIPIOS QUE SUSTENTAN EL FUNCIONAMIENTO DE UNA JERARQUIA DE MEMORIA

El éxito de esta organización se debe a la disminución de frecuencia del acceso. El cual es válido por el principio de “localidad de las referencias”. En el que se plantea que, durante el curso de la ejecución de un programa, la referencias a memoria tienden a estar agrupadas.

Entonces, es posible organizar los datos a través de la jerarquía, de manera que el porcentaje de accesos a cada nivel más bajo sea menor que el nivel anterior.

El objetivo final de la jerarquía de memorias es conseguir que, cuando el procesador acceda a un dato, este se encuentre en el nivel más rápido de la jerarquía. Obtenemos así una memoria a un coste moderado, con una velocidad próxima a la del nivel más rápido y la capacidad del nivel más alto.

El funcionamiento de una jerarquía de memoria tiene tres características claves (coste, capacidad y tiempo de acceso), en estas existe un compromiso:

- A menor tiempo de acceso, mayor coste por bit.
- A mayor capacidad, menor coste por bit.
- A mayor capacidad, mayor tiempo de acceso.

Cuando la jerarquía desciende, ocurre que:

- Disminuye el coste por bit.
- Aumenta la capacidad.
- Aumenta el tiempo de acceso.
- Disminuye la frecuencia de acceso a la memoria por parte del procesador.

LOCALIDAD DE REFERENCIA. MEJORAS EN EL FUNCIONAMIENTO DE MAQUINA.

El principio de localidad de referencia nos dice que, en el curso de la ejecución de un programa, las referencias a memoria por parte del procesador, tanto para instrucciones como para datos, tienden a estar agrupadas. Los programas normalmente contienen un numero de bucles iterativos y subrutinas. Cada vez que se entra en un bucle a una subrutina, hay repetidas referencias a un pequeño conjunto de instrucciones. Esto también se cumple al trabajar con matrices o arreglos.

La principal mejora que podemos obtener cuando se cumple el principio de localidad de referencia es la disminución de la frecuencia de acceso a los niveles inferiores de la jerarquía de memoria. Es gracias a este principio que dicha jerarquía puede implementarse.

MEMORIA CACHE

La Memoria Caché se sitúa entre la memoria principal y el procesador, puede estar formada por uno o varios niveles. Tiene un tiempo de acceso inferior al de la memoria principal con el objetivo de reducir el tiempo de acceso medio a los datos, pero también tiene un tamaño mucho más reducido que la memoria principal. Si un dato está en la memoria caché, es posible proporcionarlo al procesador sin acceder a la memoria principal, si no, primero se lleva el dato de la memoria principal a la memoria caché y después se proporciona el dato al procesador. Debido al fenómeno de localidad de las referencias, cuando un bloque de datos es captado por la cache para satisfacer una referencia a memoria simple, es probable que se hagan referencias futuras a otras palabras del mismo bloque.

MEMORIA PRINCIPAL

En la memoria principal se almacenan los programas que se deben ejecutar y sus datos, es la memoria visible para el programador mediante su espacio de direcciones. La memoria principal se implementa utilizando diferentes chips conectados a la placa principal del computador y tiene una capacidad mucho más elevada que la memoria caché (del orden de Gbytes o de Tbytes en supercomputadores). Utiliza tecnología DRAM (Dynamic RAM), que es más lenta que la SRAM, pero con una capacidad de integración mucho más elevada, hecho que permite obtener más capacidad en menos espacio.

Todos los tipos de memoria principal se implementan utilizando tecnología de semiconductores y tienen el transistor como elemento básico de su construcción. El elemento básico en toda memoria es la celda. Una celda permite almacenar un bit, un valor 0 o 1 definido por una diferencia de potencial eléctrico. La manera de construir una celda de memoria varía según la tecnología utilizada. La memoria interna es una memoria de acceso aleatorio; se puede acceder a cualquier palabra de memoria especificando una dirección de memoria. Una manera de clasificar la memoria interna según la perdurabilidad es la siguiente:

- Memoria Volátil: es la memoria que necesita una corriente eléctrica para mantener su estado, de manera genérica denominada RAM. Las memorias volátiles pueden ser de dos tipos: SRAM y DRAM.
- Memoria no volátil: mantiene el estado sin necesidad de corriente eléctrica. Las memorias no volátiles pueden ser de diferentes tipos: ROM, PROM, EPROM, EEPROM, memoria flash.

ORGANIZACIÓN 2D DE MEMORIA.

Se puede pensar en el direccionamiento 2D como en una lista de celdas apiladas, las cuales tienen un tamaño N (donde N puede ser el tamaño de palabra, por ejemplo). Un tipo de memoria que maneje este tipo de direccionamiento debe tener una conexión con alguna interfaz que le permita conseguir los datos a ingresar, y otra que le indique a que celda tiene que escribir.

También debe tener una conexión con el lugar a donde tiene que guardar los datos, si es que se realiza una lectura, aunque no siempre es necesario, porque se puede pensar en utilizar la misma conexión para escribir o leer, indicándole a la memoria con un flag que operación se debe hacer. Esto le agrega complejidad al programador para manejar esta memoria, pero en contrapartida se gana un diseño un poco más simple. Hay que recalcar que cuando se accede, ya sea leyendo o escribiendo, se lee una celda completa (una palabra, por ejemplo).

La problemática que tiene este tipo de disposición de memoria es que es físicamente muy difícil de implementar con una cantidad de registros grandes, generalmente estas memorias oscilan en tamaño entre los 512Kb-1Mb. Esto se debe a que, si pensamos en este tipo de organización como una lista de celdas, esta estructura es muy frágil (pensemos, por ejemplo, que, si queremos implementar una memoria de 15Mb con palabras de 32bits con celdas que almacenan palabras completas, entonces, tendrías 3.932.160 celdas de memorias acomodadas una tras de otra, con lo que esta disposición requeriría mucho espacio a lo largo del soporte de la memoria). Hay que

mencionar que, si bien puede haber muchas celdas, el tiempo de acceso a ellas con esta disposición es CASI constante.

Generalmente, este tipo de direccionamiento de memoria se aplica en las Memorias Estáticas (Registros de CPU, o Caché), más que nada porque se implementa con Flip-Flops que, a diferencia de los transistores, no necesitan refresco y son más rápidos a la hora de retribuir la información que almacenan.

ORGANIZACIÓN 2½D DE MEMORIA SEMICONDUCTORA. CAPACIDAD DE ALMACENAMIENTO Y TIEMPO DE ACCESO

Los bits se almacenan en una matriz cuadrada de N filas por M columnas. Se accede de un bit a la vez, que tiene asociada una dirección que puede ser leída o escrita según la señal de control.

Están hechas de transistores en lugar de flip-flops como la memoria 2D, lo que permite poder fabricarlas de mayor capacidad. El problema de los transistores es que pierden la carga y por lo tanto pierden el valor almacenado, por lo cual se debe usar una técnica de refresco. Esto puede provocar cuello de botella. Tiene dos decodificadores, la dirección se divide en dos.

El direccionamiento 2D½ mantiene una gran ventaja con respecto al direccionamiento 2D, y esta es que, a pesar de que obtener una palabra de esta memoria requiere más procesamiento, puede extender su capacidad inmensamente con respecto al direccionamiento 2D. De hecho, las memorias RAM que utilizamos en nuestras computadoras tienen capacidades que alcanzan los Gb (más allá del hecho de que cada una de estas plaquetas está compuesta de varios chips, esto habla del tamaño de la estructura que se puede armar con este tipo de direccionamiento).

Generalmente, este tipo de direccionamiento de memoria se aplica en las Memorias Dinámicas (DRAM), que se implementan con transistores, los cuales necesitan refrescos periódicos de la información que almacenan.

Hoy en día se comercializan memorias de 1 a 8 GB, con tiempos de acceso entre 50 y 70 ns.

MEMORIA SECUNDARIA

Corresponde a dispositivos de almacenamiento secundario: discos magnéticos, cintas magnéticas, discos ópticos, dispositivos de memoria flash, etc., y también se pueden considerar sistemas de almacenamiento en red. Estos dispositivos son gestionados por el sistema de ficheros del sistema operativo mediante el sistema de entrada/salida. Los dispositivos que forman la memoria externa se conectan al computador con algún tipo de bus (serie o paralelo). Estos dispositivos se pueden encontrar físicamente dentro del computador conectados por buses internos del computador (IDE, SATA, SCSI, etc.) o pueden estar fuera del computador conectados por buses externos (USB, Firewire, eSATA, Infiniband, etc.).

DISCOS MAGNETICOS. COMPONENTES QUE DEFINEN EL TIEMPO DE ACCESO.

Son dispositivos formados por un conjunto de platos con superficies magnéticas y un conjunto de cabezales de lectura y escritura que permiten grabar y recuperar datos en él. La información se graba en estas superficies.

Los componentes que definen el tiempo de acceso son:

- Tiempo de búsqueda: es el tiempo que tarda la cabeza en posicionarse en la pista deseada donde se escribirán o leerán los datos.
- Tiempo de latencia (retardo rotacional o latencia rotacional): es el retraso que se da mientras se espera la rotación del disco para que se tenga al alcance el sector deseado bajo el cabezal. Esto depende de la velocidad rotacional de un plato (o del motor de giro) medida en revoluciones por minuto (RPM)

El “tiempo de acceso” se define haciendo el “tiempo de búsqueda” + “el tiempo de latencia”.

TIEMPO DE ACCESO MEDIO

Por tanto, el tiempo de acceso medio total se puede expresar como

$$T_a = t_s + 1/2r + b/rN$$

donde T_s es el tiempo de búsqueda medio. Nótese que, en una unidad con zonas, el número de pistas

es variable, complicándose el cálculo.

donde:

T = tiempo de transferencia

b = número de bytes a transferir

N = número de bytes de una pista

r = velocidad de rotación en revoluciones por segundo

TIEMPO DE ACCESO PROMEDIO

El tiempo de acceso promedio se podría calcular sumando todos los tiempos de acceso y dividiéndolo por el número de accesos totales

REGISTRO DE PUNTERO DE PILA (STACK POINTER). OBJETIVO.

El objetivo del stack pointer es contener el valor de la dirección del tope de la pila.

La pila tiene asociado un puntero cuyo valor es la dirección de la cabecera o tope de la pila.

Alternativamente, los dos elementos de la cabecera de la pila pueden residir en registros del procesador, en cuyo caso el puntero de pila hace referencia al tercer elemento de la pila. El puntero de pila se mantiene en un registro. Así, las referencias a posiciones de la pila en memoria son direcciones de acceso indirecto con registro.

El modo de direccionamiento de pila es una forma de direccionamiento implícito. Las instrucciones máquina no necesitan incluir una referencia a memoria, sino que operan implícitamente con la cabecera de la pila.

PARA QUE SIRVE TENER MAS DE UN PUNTERO DE PILA.

La ventaja de tener otro puntero de pila es que se genera otra pila aparte. A esta la puedo manejar y administrar yo mismo (usuario). Es decir, una pila la maneja el sistema y la otra el usuario. Nos independizamos de la pila del sistema.

MOSTRAR EN "TRUE COLOR" DE 1024X1024 PÍXELES, UN VIDEO QUE POSEE 20 IMÁGENES POR CADA SEG.

$1024 \times 1024 \times 3 \text{ Bytes} \times 20 = 62.914.560 \text{ Bytes} = 60 \text{ MB}$

1 Byte _____ 50 ns

62.914.560 _____ x ns

$(62.914.560 \times 50) / 1 = 3,14 \text{ seg. aprox.}$

DESCRIBA LOS MÉTODOS DE ACCESO A LA INFORMACIÓN ALMACENADA EN MEMORIA SECUNDARIA. ¿QUE VENTAJA Y/O USOS PUEDE MENCIONAR DEL ALMACENAMIENTO EN RAID?

Los métodos de acceso son:

- Secuencial: la memoria se organiza en unidades (registros). Para acceder a uno se debe ir trasladando desde la posición actual a la deseada por todos los registros intermedios. Ej.: discos.
- Directo: se accede directamente a una vecindad dada de registros o bloques, luego una búsqueda secuencial contando o esperando alcanzar la posición. Ej.: discos.
- Aleatorio-Random: cada posición direccionable tiene un único mecanismo de acceso. Ej.: memoria principal y cache.
- Asociativa: de tipo aleatoria donde se hace una comparación de ciertos bits de una palabra buscando coincidencias de valores dados, y esto para todas las palabras simultáneamente. Una palabra accedida basándose en una parte de su contenido y no de su dirección o posición.

Las ventajas y/o usos de RAID depende de cada nivel:

- NIVEL 0: En este nivel los datos del usuario y del sistema están distribuidos a lo largo de todos los discos del conjunto, a través de la tira de datos (las cuales pueden ser bloques físicos, sectores o alguna otra entidad) y se proyectan cíclicamente en dispositivos consecutivos. Al conjunto de tiras lógicamente consecutivas, que se proyecta exactamente sobre una misma tira en cada miembro del conjunto, se las denomina franja.

- NIVEL 1: Consigue la redundancia duplicando todos los datos, proporcionando una copia de seguridad en tiempo real. Se hace una distribución de datos como en el nivel 0, pero cada franja lógica se proyecta en dos discos físicos separados. De esta forma cada disco tiene un espejo que tiene la misma información.

Esto trae tres ventajas:

1. Una petición de lectura puede ser servida por cualquier de los dos discos que contienen los datos pedidos.
2. La petición de escritura, que requiere que las dos tiras correspondientes se actualicen, se hace en paralelo.
3. Cuando una unidad falla, se puede acceder inmediatamente a los datos de la otra.

La principal desventaja es el costo de su implementación, ya que se requiere el doble del espacio de disco del disco lógico. Ej.: Para organizar un RAID 1 de 4 discos físicos se usarán 2 para el espacio lógico y 2 para el espejado.

- NIVEL 2: utiliza una técnica de acceso en paralelo, que consiste en que todos los discos miembros, participan en la ejecución de cada E/S. Dado que el giro de cada unidad esta sincronizado, para que cada cabezal este en la misma posición en cada disco.

En este nivel las tiras son muy pequeñas, y la redundancia y corrección de errores, se logran a través de una ecuación de paridad, que consiste en calcular los bits de cada posición de cada disco, y almacenarlos en las correspondientes posiciones en los discos de paridad.

Cuando se produce un error en un solo bit, el controlador del conjunto lo reconoce, realiza el calculo para saber el valor correspondiente y lo corrige instantáneamente.

En cuanto a las operaciones E/S, una petición de lectura no conlleva un tiempo adicional, sin embargo, la petición de escritura debe acceder a todos los discos de datos y de paridad.

La ventaja de este RAID está en que se necesita una menor cantidad de discos para almacenar la información. Y su desventaja recae en el costo.

Su uso está destinado en entornos en los que haya muchos errores de disco y la seguridad no sea alta.

- NIVEL 3: se organiza de manera similar que el de nivel 2 y utiliza un acceso en paralelo, pero se diferencia de él en que solo requiere de un disco redundante, sin importar lo grande que sea el conjunto de disco.

En este nivel la corrección es simple, se calcula un sencillo bit de paridad para el conjunto de bit individuales que están en la misma posición, en todos los discos de datos. En caso de que falle una unidad, se accede a disco de paridad y se reconstruye la información desde el resto de los dispositivos.

Las ventajas de este RAID se encuentran en el ahorro de espacio, ya que solo se necesita un disco de paridad y en las altas velocidades de transferencias de datos. Su desventaja está en que solo se puede recuperar un disco dañado por vez, si se dañaran dos al mismo tiempo, no podría reconstruir la información.

Su uso está destinado en entornos no orientados en transacciones.

- NIVEL 4: Este nivel, así como el 5 y el 6, usan una técnica de acceso independiente, por lo que las diferentes E/S se pueden atender en paralelo. (es más adecuado para sistemas que requieran velocidades de petición de E/S altas, que para los que necesiten velocidades de transferencia altas)

Las tiras de datos son relativamente grandes, y se realiza una paridad bit a bit que se guarda en la tira correspondiente del disco de paridad. Esto trae aparejado un cuello de botella sobre el disco de paridad, ya que debe actualizarse cada vez que se haga una escritura en cualquiera de los discos de datos.

- NIVEL 5: Este nivel es muy similar al nivel 4, pero las tiras de paridad se distribuyen a lo largo de todos los discos. Se distribuye de manera cíclica: para un conjunto de n discos, la tira de paridad está en diferentes discos para las primeras n tiras, y este patrón se repite. O sea que la tira de paridad siempre está en otro disco que su correspondiente tira de datos.

Esta distribución evita el cuello de botella que se produce el nivel 4, ya que la actualización de la paridad se distribuye sobre todos los discos.

- NIVEL 6: El nivel 6 agrega otro algoritmo de comprobación de datos, y se distribuyen de la misma manera que el nivel 5. Uno es el algoritmo de los niveles 3, 4 y 5 que calcula la XOR, y otro es un algoritmo independiente. Esto hace posible la regeneración de los datos incluso si dos de los discos que contienen los datos de los usuarios fallan.

Es una solución perfecta para aplicaciones con objetivos críticos, pero tiene un gran costo de cálculo y escritura de paridades.

ENUMERE LAS INSTRUCCIONES DE TIPO TRANSFERENCIA DE CONTROL QUE POSEE EL MSX-88 Y EXPLIQUE EL MODO DE DIRECCIONAMIENTO DE C/U DE ELLAS.

Las instrucciones de tipo transferencia de control que posee el MSX-88 son: Los saltos (JMP, JZ, JNZ, JS, JNS, JC, JNC, JO, JNO)

Las llamadas a subrutinas (Call, Ret)

También son de transferencia de control las de interrupciones.

Los modos de direccionamiento que usan son:

Relativos al PC, donde se tiene un número que debe ser sumado al contenido del PC.

Directo, cuando la etiqueta representa una dirección a la que se debe saltar.

¿QUÉ ES UNA PUERTA LÓGICA? DESCRIBE EL MÉTODO PARA LA OBTENCIÓN DEL CIRCUITO COMBINATORIO DE UNA TABLA DE VERDAD CON CUATRO VARIABLES DE ENTRADA QUE PRODUCE UNA SALIDA "1" SOLO CUANDO UNA DE LAS 4 ENTRADAS ESTA EN 1.

Una puerta lógica es un dispositivo electrónico que produce como señal de salida una operación booleana a partir de las señales de entrada. Las puertas básicas usadas en lógica digital son: AND OR NOT, y en base a estas se derivan XOR, NAND y NOR. Cada puerta tiene una o más entradas que cuando cambian, la señal de salida aparece casi instantáneamente, retrasada solo por el retardo de puerta. La puerta se define de tres formas:

- Simbólica: es el convenio gráfico conocido universalmente para representar la puerta
- Función algebraica: es la ecuación con la que se puede definir una puerta.
- Tabla de verdad: enumera el valor de una operación para cada combinación posible de los valores de los operando.

El método consiste en usar la técnica de suma de productos, primero se arma la tabla de verdad:

A	B	C	D	F
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0

0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

Tomar las filas en las que la salida es "1", y armar la función algebraica. Los "0" serán la variable negada.

$$F = [(\sim A).(\sim B).(\sim C).D] + [(\sim A).(\sim B).C.(\sim D)] + [(\sim A).B.(\sim C).(\sim D)] + [A.(\sim B).(\sim C).(\sim D)]$$

Así de esta forma queda dibujar el circuito lógico que van a ser tres AND unidas a un OR. Las negaciones se realizan agregando una NOT antes.

Se unen todas las salidas de las compuertas AND con una OR porque solo se va a cumplir una de las condiciones a la vez.

COMPLETADA LA INSTRUCCIÓN ADD AX, MENO1. ¿QUÉ INSTRUCCIÓN DEBERÁ EJECUTAR A CONTINUACIÓN PARA DETERMINAR SI EL RESULTADO OBTENIDO ES CORRECTO O NO? CONSIDERE LOS CASOS DE REPRESENTACIÓN DE NÚMEROS SIN SIGNOS Y EN CA2.

Para números sin signo debería ejecutar:

```

JNC correcto
MOV CX, 1
JMP fin
correcto: MOV CX, 0
fin:      HLT

```

Para números en ca2 debería ejecutar:

```

JNO correcto
MOV CX, 1
JMP fin
correcto: MOV CX, 0
fin:      HLT

```

PERIFERICOS

MODEM. ¿PARA QUE SIRVE, PORQUE SE UTILIZA? ¿CUALES SON LOS PARÁMETROS DE UN MÓDEM?

MODEM: MODulador, DEModulador. Convierte señales '0' y '1' en tono de audio y viceversa. La tasa de Bits/seg (bps) es el número de bits enviados por segundo. Tasa de baudio es el número de cambio de señal por segundo.

Su principal uso es para telecomunicaciones: convierte señales analógicas provenientes de un sistema telefónico a cadenas binarias. Es una de las tecnologías más usadas para conectarse a internet.

MECANISMOS DE IMPRESION

De impacto: Son aquellas que imprimen mediante el impacto del cabezal sobre la hoja:

- Matriz de puntos: Es un tipo de impresora muy lenta, que utiliza un conjunto de agujas que golpean una cinta entintada sobre el papel, imprimiendo punto a punto.
- Margarita: todos los caracteres sobre relieve están dispuestos en forma de flor e impactan de a uno contra la hoja, golpeando una cinta entintada que les permite marcar el carácter. La ventaja es que cada carácter individual se imprime con buena calidad, pero no puede imprimir nada que no sea cadenas de caracteres.
- Cinta
- Tambor

Térmicas: Son aquellas que imprimen mediante la transferencia de calor al cabezal, a través de una matriz de pequeña resistencia, a las que, al pasar corriente eléctrica por ellas, se calientan formando los puntos. Estas impresoras pueden ser de carácter (las líneas se imprimen con un cabezal móvil) o de línea (contiene tantas cabezas como caracteres a imprimir por línea, son más rápidas).

Chorro a tinta: Son aquellas que imprimen mediante la carga de gotas de tinta, por medio de electricidad estática. El carácter se forma con la tinta que cae al papel según la carga estática que posea la misma. Cuando finaliza, las gotas que sobran se desvían hacia un camino de retorno al cartucho.

Son modelos bidireccionales y muy utilizados por su velocidad y gran calidad de impresión.

Láser: Son aquellas que imprimen mediante la radiación de un láser sobre una superficie con propiedades electrostáticas, desde un tambor que tiene la imagen impregnada en tóner.

Poseen un sistema similar a las fotocopadoras, con una elevada velocidad y calidad de impresión

DESCRIBA LAS DIFERENCIAS QUE TENDREMOS ENTRE UNA COMPUTADORA CON TECLADO Y MONITOR COMO PERIFÉRICOS Y UNA QUE POSEE UN EQUIPO DENOMINADO TERMINAL COMO PERIFÉRICO. PUEDE CONSIDERAR CANTIDAD Y UBICACIÓN DE LA MEMORIA DE VIDEO, CANTIDAD Y TIPO DE PUERTAS DE E/S, TIPO DE COMUNICACIÓN ENTRE CPU Y PERIFÉRICO.

En una computadora tradicional con teclado y monitor, tendremos un monitor de video con memoria de visualización que está mapeada en la memoria principal del equipo. En cambio, en un esquema con terminales, cada terminal se compone de un teclado y un monitor con su propia memoria de visualización dedicada a ese terminal. Esto evita que se llene la memoria principal del computador que centraliza a los terminales. Cada terminal puede estar alejado físicamente del CPU, por ejemplo, se puede tener el computador principal en una habitación y poner un terminal en cada una de las demás habitaciones.

La terminal conecta sus periféricos con la computadora a través de una conexión en serie, que tiene poco ancho de banda, por lo que los monitores de los terminales suelen estar orientados a caracteres para ahorrar transferencia de datos. En cambio, las computadoras con video mapeado en memoria conectan los periféricos directamente al bus de memoria, que permite cambios rápidos. Esto hace que tenga solo 1 E/S para el monitor, a diferencia del computador con terminales, que va a tener 1 conexión de E/S para cada terminal, a pesar de que no necesite mapear la memoria de visualización en su memoria principal.