

## Intento de resolución de finales de organización de computadoras

Septiembre 2012:

2. ¿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 mas 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 operandos.

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.

### 3. Describa las características principales de la organización 2 ½ D de memoria semiconductora. Defina los valores de capacidad de almacenamiento y tiempo 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. Hoy en día se comercializan memorias de 1 a 8 GB, con tiempos de acceso entre 50 y 70 ns.

### 4. Enumere todos los elementos a tener en cuenta para el diseño del repertorio de instrucciones de un procesador. Describa dos de ellos.

Para el repertorio de instrucciones hay que tener en cuenta:

- Tipos de operaciones: cuantas y cuales
- Tipos de datos: cuales
- Formato de instrucciones: longitud (bits), N° de direcciones, tamaño de cada campo.
- Registros: cantidad que se pueden referenciar mediante instrucciones y su uso.
- Direccionamiento: la manera de especificar la dirección de un operando o una instrucción (la próxima)

Dos ejemplos podrían ser, las máquinas de dos direcciones y las de una dirección.

#### Tipos de operaciones

- Transferencia de datos: MOV (load/store)
- Aritméticas: ADD, SUB, INC, DEC, MUL, DIV.
- Lógicas: AND, OR, XOR, NOT.
- Conversión
- E/S: In, Out.
- Transferencia de control: salto, bifurcación.
- Control del sistema: usadas por el sistema operativo.

#### Tipos de datos

Los más importantes:

- Direcciones
- Números: enteros, punto fijo, punto flotante.
- Caracteres: ASCII, BCD.
- Datos lógicos

#### Modos de direccionamiento

En una instrucción se utilizan bits para expresar el código de operación: nos dicen qué hacer. También se necesitan una gran cantidad de bits para especificar de donde provienen los datos.

Los modos de direccionamiento tienen como objetivo:

- Disminuir la cantidad de bits en la instrucción.

- La dirección puede que no se conozca hasta el momento de ejecutar el programa.
- Manejo mas eficiente de los datos (arreglos).

Los modos de direccionamiento que existen son:

- Inmediato
- Directo
- Por registro
- Indirecto por memoria
- Indirecto por registro
- Por desplazamiento
- Del stack

#### Mdd 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.

#### Mdd Directo

El campo de direccion tiene la direccion efectiva del operando. Es simple pero tiene un espacio limitado de direcciones por cantidad de bits del campo.

#### Mdd 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.

#### Mdd 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 el espacio de direccionamiento es mayor. La principal desventaja es que realiza múltiples accesos a memoria.

#### Mdd 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.

#### Mdd 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 mas comunes son Relativo, De registro base e Indexado.

#### Mdd Del stack

El stack o pila es un arreglo lineal de localidades de memoria. Es una lista o cola donde el ultimo 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.

5. Se requiere mostrar en “True Color” de 1024x1024 píxeles, un vídeo que posee 20 imágenes por cada segundo ¿podría usar la memoria cuyos valores de capacidad y tiempo de acceso definió en el punto 3? justifiquen.

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

$$1 \text{ Byte} \underline{\hspace{2cm}} 50 \text{ ns}$$
$$62.914.560 \underline{\hspace{2cm}} \times \text{ns}$$

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

Por lo tanto la memoria especificada en el punto 3 no nos sirve para mostrar las 20 imágenes en un segundo.

### Mayo 2010

2. ¿Qué características determinan que un circuito lógico sea combinacional? Describa el método para implementar el circuito lógico de la función booleana F de cuatro entradas (a, b, c y d) cuya tabla de verdad posee valor de salida “1” cuando solo una de sus variables esta en “1”. realice el gráfico correspondiente de interconexión de compuertas resultantes.

Para que un circuito lógico sea combinacional se necesitan un conjunto de dos o mas puertas interconectadas. Responden a los valores logicos de las entradas, la salida esta determinada exclusivamente por los valores de las entradas en ese instante. Si cambia la entrada cambia la salida. Lo valores pasados de las entradas no influyen en los valores de las salidas.

3. ¿Qué es un ciclo de instrucción? 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?

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 una 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 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

4. ¿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 vaya a multiplicar muchos números 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.

5. 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 esta en que se necesita una menor cantidad de discos para almacenar la información. Y su desventaja recae en el costo.

Su uso esta 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 una 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 esta en que solo se puede recuperar un disco dañado por ves, si se dañaran dos al mismo tiempo, no podría reconstruir la información.

Su uso esta 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.

Febrero 2010:

2. Describa como se puede construir un sumador binario completo. Defina cuales son las funciones lógicas que se pueden utilizar, tabla de verdad y realice un esquema de interconexión de compuertas de dicho sumador.

Un sumador binario completo se puede construir armando la tabla de verdad con los valores booleanos de la suma binaria, y la tabla de verdad de la suma con acarreo. Una vez realizadas estas tablas, se puede hacer las funciones de salida.

Para que funcione un sumador de varios bits cada uno de los sumadores de un bit deben tener 3 entradas, incluyendo el acarreo del sumador inmediato inferior.

$$\text{Suma} = (-A \cdot -B \cdot C) + (-A \cdot B \cdot -C) + (A \cdot B \cdot C) + (A \cdot -B \cdot -C)$$

$$\text{Acarreo} = A \cdot B + A \cdot C + B \cdot C$$

3. ¿Qué mejoras podemos obtener en el funcionamiento de máquina que ejecuta instrucciones debido al principio de localidad de referencia?

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.

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.

4. ¿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.

5. ¿Qué objetivo persigue la existencia del registro puntero de pila (stack pointer) en el CPU? ¿Para qué serviría tener más de uno?

El objetivo del stack pointer es contener el valor de la dirección del tope de la pila. Tener más de un Stack Pointer serviría para poder tener una pila para el usuario y una para el sistema. Esto evitaría que el usuario tenga que estar pendiente de los cambios que realiza el sistema en la pila.

6. 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
```

7. Describa los componentes que definen el tiempo de acceso de un disco magnético. ¿Cómo se podría calcular un tiempo de acceso promedio?

Los componentes que definen el tiempo de acceso son: Por un lado, la cabeza de lectura, que tiene una cierta velocidad para alcanzar el cilindro buscado (tiempo de seek).

También los discos en si, que tienen una determinada velocidad de giro. Esto determina el tiempo de latencia, que ocurre desde que la cabeza de lectura se posiciona sobre el cilindro, hasta que el sector buscado pasa por debajo de la cabeza.

Para calcular el tiempo de acceso promedio debemos sumar el tiempo de seek y el tiempo de latencia promedio. Este último se puede calcular como el tiempo de giro de una pista dividido dos. Otra manera podría ser midiendo el tiempo de acceso de una gran cantidad de accesos y hacer un promedio.

Marzo 2009

2. ¿Qué es un 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-flop que tiene dos entradas al igual que es SR, pero todas las combinaciones posibles son validas. 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.

5) 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

Febrero 2013 2do llamado

3) 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.

5) ¿Que es un módem? 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 numero de bits enviados por segundo. Tasa de baudio es el numero 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.

5) Describa los mecanismos de impresión que conozca, compare calidad obtenible, cantidad de copias por vez, tamaño de papel utilizable, cantidad de colores y velocidad de impresión de los mecanismos que mencione.

- Impresora de impacto: Los caracteres son impresos golpeando los moldes de los caracteres o una cinta entintada sobre la hoja.  
En la disposición “**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.  
En cambio, las impresoras por **matriz de puntos**, marcan los caracteres sobre una cinta al igual que las de margarita, pero usando punzones manejados por solenoides. Tiene tantos punzones como sea el alto de la matriz de caracteres. Produce muy baja resolución.  
Ambos tipos de impresoras son muy lentas y por lo general ruidosas. Según el fabricante, pueden usar algunos colores, pero muy limitados y de baja calidad y resolución.

- Impresora láser: Es el tipo de impresora con mayor resolución, llegando hasta 1200dpi. Permite usar todos los colores y puede imprimir páginas completas de texto o gráfico a gran velocidad. Permite realizar muchas copias en menos tiempo que los demás sistemas. La desventaja es que son costosas.
- Impresora Ink-jet ó de inyección de tinta: Tienen un transductor ultrasónico que lanza pequeños chorros de tinta a los puntos correctos con la cabeza moviéndose sobre el papel. Permite tanto blanco y negro como color. El precio, resolución, y velocidad se encuentra entre las anteriores. Esto hace que sean las más comunes para uso cotidiano.

Con respecto a al tamaño utilizable, puede que las impresoras de impacto estén limitadas por el tamaño físico de los punzones o martillos. En los otros tipos, depende del tamaño de bandeja que incorpore el fabricante. Los tamaños más comunes son los estándares “A4”, “Oficio”, “carta”, etc.

La cantidad de copias por vez va depender de la velocidad de impresión del dispositivo, siendo las más rápidas las láser.

Describe 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 de el 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.

¿Qué define el Teorema fundamental de la numeración?

El teorema fundamental de la numeración establece la forma general de construir números en un sistema de numeración posicional.

$$N = \sum_{i=-k}^n d_i \cdot b^i$$

Donde:

- N es el número decimal
- d es un dígito que puede variar entre {0,1,..., (b-1)}
- b es la base de representación