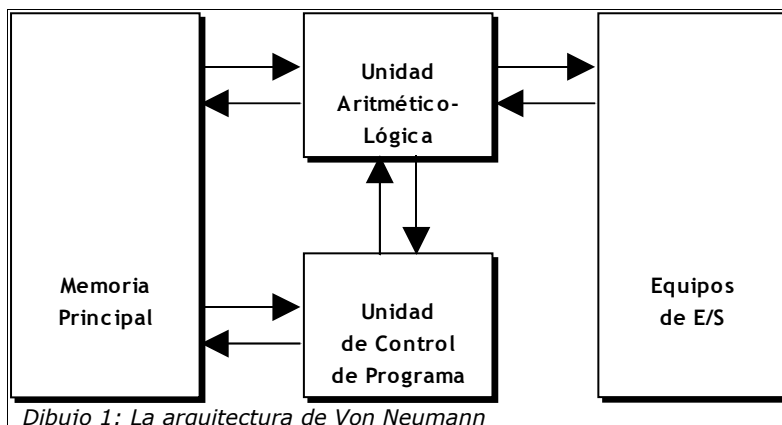



Apunte de **Organización de Computadoras**

El contenido de este apunte debe ser utilizado como una GUÍA para el estudio para la cursada, o incluso el final de esta asignatura (Organización de Computadoras), ya sea para el plan 2003 o 2007. Sin embargo, se recomienda asistir a las CLASES DE CONSULTA de los docentes a cargo, para COMPLEMENTAR las nociones vertidas en este documento, y comprender los requisitos de los docentes para con los alumnos, a la hora de tomar una respuesta como CORRECTA O NO, en un EXAMEN.

La Máquina De Von Neumann - IAS



Dibujo 1: La arquitectura de Von Neumann

 Fue diseñada en 1945 por Von Neumann y estableció la arquitectura de las computadoras que, con algunas modificaciones, es empleada hoy en día por la mayoría de los fabricantes.

Dicha arquitectura está constituida por:

- Una **Unidad Aritmética Lógica** encargada de las operaciones aritméticas y lógicas.
- Una **Unidad de Control** encargada de interpretar las instrucciones almacenadas en memoria para su ejecución.
- **Dispositivos de E/S** encargados de la comunicación al exterior, dirigidos por la CU.
- Una **memoria principal** en donde se almacenan los datos y las instrucciones, a los cuales se accede por la dirección (posición) dentro de la misma, sin considerar el tipo.

En este modelo, y aún se mantiene, la ejecución de las instrucciones se realiza siguiendo una secuencia de las mismas, una tras otra y si no se producen modificaciones explícitas que alteren su orden.

Estas instrucciones se dividen en cinco grupos: *Transferencia de datos*; *Salto incondicional*; *Salto condicional*; *Aritméticas* y *Modificación de direcciones*.

Ciclo de Instrucción

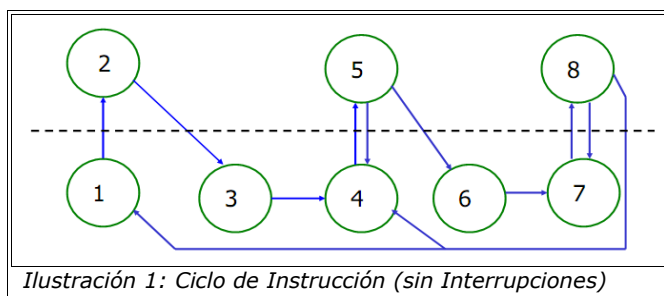


Ilustración 1: Ciclo de Instrucción (sin Interrupciones)

- 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).

Interrupciones

Una Interrupción es precisamente eso, una interrupción en la secuencia normal de funcionamiento.

Cuando el procesamiento de la interrupción se completa, la ejecución prosigue. Así, el programa de usuario no tiene que incluir ningún código especial para posibilitar las interrupciones; el procesador y el Sistema Operativo son los responsables de detener el programa de usuario y después permitir que prosiga en el mismo punto.

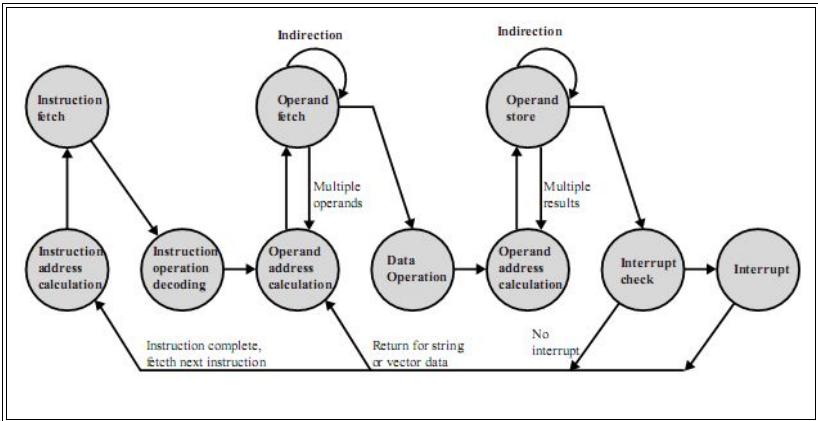


Ilustración 2: Ciclo de instrucción extendido (con interrupciones)

Tipo de Interrupción	Descripción
Por programa	Generada por una condición que se produce como resultado de la ejecución de una instrucción. Ejemplos: División por Cero, Overflow, Intento de ejecutar una instrucción inexistente, intento de acceder fuera del espacio de memoria permitido, etc.
Temporizador	Generada por un temporizador interno al procesador.
Entrada/Salida	Generada por un controlador de E/S. Para indicar la finalización sin problemas de una operación o para avisar de ciertas condiciones de error.
Fallo de Hardware	Generada por un fallo tal como falta de potencia de un generador de alimentación o un error de paridad en la memoria.

Tabla 1: Tipos de Interrupciones

Diseño del conjunto de instrucciones


El conjunto de instrucciones es el medio que tiene el programador para controlar la CPU.

Hay que tener en cuenta:

- Tipos de operaciones
 - Cuántas y cuáles operaciones se implementan para la organización del conjunto de instrucciones.
 - Para un mayor número de operaciones implementadas el número de bits disponibles para direccionamiento se reduce.
- Tipos de datos
 - Cuáles tipos de datos se pueden representar dentro del conjunto de instrucciones. El tipo de datos está determinado por la operación que se esté realizando con la secuencia de bits con la que trabajamos y, en general, se utiliza la siguiente clasificación :
 - Direcciones:
 - Números: enteros, coma fija, flotante, decimal.


- Caracteres: ASCII, EBCDIC, ASCII extendido
- Datos Lógicos: una secuencia de N bits, donde cada uno puede representar un valor de verdad (verdadero o falso).
- **Formato de instrucciones**
 - longitud de la instrucción en bits, número de direcciones, tamaño de cada campo
- **Registros**
 - Número de registros de la CPU que pueden ser referenciados por las instrucciones, y su uso.
- **Direccionamiento**
 - la manera de especificar la dirección de un operando o una instrucción (la próxima).


Formato de instrucción

 Dentro de la computadora cada instrucción está representada mediante una secuencia de bits. La secuencia se divide en campos en correspondencia a los elementos que la componen. A este esquema se lo conoce como **formato de la instrucción**.

Componente	Función
Código de operación.	Especifica la operación a realizar (ej. suma). Es un código binario
Referencia del operando fuente.	Establece dónde se encuentra el operando. La operación puede involucrar más de un operando fuente (o de entrada).
Referencia del operando resultado.	Establece dónde almacenar el resultado.
Referencia de la siguiente instrucción	Le dice a la CPU donde buscar la siguiente instrucción después de la ejecución de la instrucción anterior. En la mayoría de los casos se ubica a continuación de la instrucción actual.

Tabla 2: Componentes de la Estructura del formato de instrucción

 Cabe destacar que los operandos fuente y resultado pueden estar en tres lugares: Memoria, Registro de la CPU o Dispositivo de E/S.

 Es difícil para el programador tratar con las representaciones binarias de las instrucciones de máquina. Por lo tanto, se usa una representación simbólica. Por ello, los códigos de operación se representan por medio de abreviaturas, llamadas **mnemónicos** que indican la operación. Los ejemplos más comunes son ADD, SUB, MOV, AND, por ejemplo.

Categoría	Descripción	Ejemplos
Procesamiento de datos.	Operaciones aritméticas.	ADD, ADC, SUB, SBB, CMP, NEG, INC, DEC.
	Operaciones lógicas.	AND, NOT, XOR, OR.
Almacenamiento de datos.	Transferencias dentro del sistema.	MOV, PUSH, POP, PUSHF, POPF.
De E/S.	Transferencia de datos entre la computadora y los mecanismos externos.	IN, OUT.
Transferencia de Control.	Control de flujo de programa.	CALL, RET, JZ, JNZ, JS, JNS, JC, JNC, JO, JNO, JMP.
Manejo de Interrupciones	Operaciones para el control de interrupciones.	INT (N), IRET, CLI, STI.

Categoría	Descripción	Ejemplos
De Control	Control de ejecución del programa.	<i>NOP, HLT</i>

Tabla 3: Categorías de los codops

Memoria

Tipo de Memoria	Clase	Borrado	Mecanismos de escritura	Volatilidad
De acceso aleatorio (RAM)	Memoria de lectura/escritura	Eléctricamente, por bytes	Eléctricamente	Volátil
De solo lectura (ROM)	Memoria de sólo lectura	No es posible!	Mediante máscaras	No Volátil
ROM programable (PROM)			Eléctricamente	
PROM borrable (EPROM)	Luz ultravioleta, chip completo			
Memoria FLASH	Eléctricamente por bloques			
PROM borrable eléctricamente (EEPROM)	Eléctricamente por bytes			

Tabla 4: Tipos de Memoria

Jerarquía	Capacidad en bytes	Tiempo de acceso	Método de Acceso	Modo de Acceso
Registros	6 – 200	2 – 20 ns	A. Asociativo	Palabra
Memoria Caché	4k - 64k	20 – 100 ns	A. Asociativo	Bloque
Memoria Principal	100k - 50M	100 – 300 ns	A. Aleatorio	Palabra
Memoria secundaria disco	1M - 1G	5 – 50 ms	A. Directo	Bloque
Memoria auxiliar disco	50M - 10G	100 ms	A. Directo	Bloque
Memoria auxiliar cinta	ilimitada	? minutos	A. Secuencial	Registro

Tabla 5: Jerarquía de Memoria

✓ **La RAM Dinámica** es un tipo de memoria cuyas celdas se implementan utilizando un transistor y un condensador. El condensador mantiene el bit de información (un 0 o un 1). El transistor actúa como un conmutador que permite a los circuitos del chip leer el condensador o cambiar su estado. Una RAM dinámica, debido al uso de condensador, pierde gradualmente sus datos salvo si se refresca periódicamente.

✓ **La RAM Estática** se implementa con biestables (flip-flops). Una RAM estática mantendrá sus datos mientras se le suministre corriente; no se necesita un refresco periódico.

Direccionamiento de Memorias

Este mecanismo de direccionamiento presenta tres formas típicas llamadas 2D, 2D½ y 3D.

➔ **Direccionamiento 2D (Memoria Estática):**

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 mas 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 pueden 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), mas 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.



Direccionamiento 2D½ (Memoria Dinámica):

Dadas las dificultades a las que nos enfrentamos en el modo de direccionamiento 2D, y teniendo en cuenta el coste (económico, y técnico) que tiene su implementación, se ha optado por otro tipo de modo de direccionamiento que satisfagan requerimientos diferentes a los previamente satisfechos.

De esta manera, surge el modo de direccionamiento 2D½, el cual se organiza de la siguiente manera:

Una memoria que utilice este tipo de direccionamiento, tendría una estructura compuesta por varios chips, las cuales tendrían la responsabilidad de retornar el valor de un bit, el cual se utilizaría para formar el valor de la palabra de N bits que se quiera leer, o escribir en la memoria.

Para ello, sería necesario proveerle al chip, la posición de ella que quiero acceder.

En consecuencia, cuando a la memoria se le pide acceder a una dirección, la misma se divide en dos partes (a través de decodificadores), una que le indica a la memoria, que “renglón” se quiere leer (con la cual se ubica a todas los chips necesarios para la reconstrucción de ese “renglón”) y otra, que indica en que parte de ese chip se debe buscar el bit correspondiente a una fracción del “renglón” a devolver. Una vez identificado cada bit de la palabra, se reconstruye la misma, y se la ubica en la interfaz de salida de la estructura.

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 (mas 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).

Ahora bien, a consecuencia de este gran incremento en la capacidad de almacenamiento de una memoria con este modo de direccionamiento, y debido al hecho de que por una cuestión de costes, estas se construyen con transistores, estas memorias, si bien son mucho mas baratas que las 2D, también son considerablemente mas lentas. Sobre todo teniendo en cuenta el procesamiento extra que conlleva decodificar una dirección de memoria en particular.



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.



Direccionamiento 3D (Modelo Teórico):

Este es un modelo teórico, debido a que es, en realidad, un modelo que tiene implementado toda la estructura para hacer funcionar el modelo 2D½. Esto es planteado de forma de que, cada uno de los chips de las que se obtiene un bit, reflejaría una capa de el modelo de direccionamiento 3D. Entonces obtendríamos el contenido de una dirección, si obtuviésemos el contenido de cada capa en la misma coordenada X e Y, siendo Z el plano que representa a todas las capas en la misma coordenada.

Aspecto	Memoria 2D	Memoria 2D½
Ubicación de los bits	Todos los bits están en el mismo chip.	Los bits de una misma palabra estarán en distintos chips.
Espacio físico	Es muy larga y estrecha, permitiendo	Tiende a tener una forma rectangular. Ocupa

Aspecto	Memoria 2D	Memoria 2D½
	guardar un N° grande de palabras de pocos bits. Cada línea de selección de palabra tiene que tener un manejador y conectarse al decodificador. Ocupa mucha superficie.	poca superficie en relación a la cantidad de información que puede almacenar.
Corrección de error	Dificulta el uso eficaz de los circuitos correctores de error.	Al estar los bits dispersos en distintos chips hay menor probabilidad de error.
Complejidad de los decodificadores	Mucha complejidad, debido a la cantidad de palabras que tienen que manejar.	Al utilizar decodificación separada de filas y columnas, reduce la complejidad de los decodificadores.

Tabla 6: Comparación entre memoria 2D y 2D½

Sistemas de Numeración



$$N^o = \sum_{i=-m}^n (digito)_i \times (base)^i$$

Nº es el valor decimal de una cantidad expresada en base B y con (n+1+m) dígitos en posiciones i.

$$\dots + X_4 \cdot B^4 + X_3 \cdot B^3 + X_2 \cdot B^2 + X_1 \cdot B^1 + X_0 \cdot B^0 + X_{-1} \cdot B^{-1} + X_{-2} \cdot B^{-2} + \dots$$

Capacidad de representación, resolución y rango en sistemas restringidos a n bits



Capacidad de representación: Es la cantidad de números que se pueden representar. Ya sea en punto fijo o no es b^n .

Por ejemplo, si tengo un sistema restringido a 5 bits, sería 25 números, es decir, 32 números.



Resolución: Es la mínima diferencia entre un número representable y el siguiente. Se podría decir que es la diferencia entre el 0 y el siguiente.

Por ejemplo, en binario con dos dígitos fraccionarios es 0.01.



El **rango** de un sistema está dado por el número mínimo representable y el número máximo representable.



Se pueden obtener los extremos de los rangos, tanto como para punto fijo como para punto flotante de la siguiente manera:

- **Mínimo número negativo:**
 - $(mantisa\ máxima) * base^{máximo\ exponente\ positivo}$
- **Máximo número negativo:**
 - $-(mantisa\ mínima) * base^{máximo\ exponente\ negativo}$
- **Mínimo número positivo:**
 - $(mantisa\ mínima) * base^{máximo\ exponente\ negativo}$
- **Máximo número positivo:**
 - $(mantisa\ máxima) * base^{máximo\ exponente\ positivo}$

Por ejemplo, en binario con cinco dígitos es [0, 31] (donde el 0 es 00000 y el 31 es 11111). **El número máximo representable en un sistema para la parte entera es $b^n - 1$.**

Sistema de Numeración	Capacidad de representación	Resolución	Rango
BSS	2^n	1	$[0, 2^n - 1]$
BCS (Signo - Magnitud)	$2^n - 1$	1	$[-(2^{n-1} - 1), 2^{n-1} - 1]$
Punto fijo (<i>N bits parte entera, M bits parte fraccionaria</i>)	$2^{\text{cantidad de bits}}$	Mínimo número positivo	[Mínimo número negativo , Máximo número positivo]
Ca1	$2^n - 1$	1	$[-(2^{n-1} - 1), (2^{n-1} - 1)]$
Ca2	2^n	1	$[-(2^{n-1}), (2^{n-1} - 1)]$
Exceso a M		1	$[-(2^{n-1}), (2^{n-1} - 1)]$
Punto flotante (<i>M bits mantisa, E bits exponente</i>)	$2^{\text{cantidad de bits}}$	Mínimo número positivo	[Mínimo número negativo , Máximo número positivo]

Tabla 7: Comparación de Sistemas de Numeración



Tanto **BCS** como **Ca1** poseen doble representación del 0.

Representación de números (Coma Fija)

Con una notación de **Coma Fija**, es posible representar un rango de números enteros, tanto positivos como negativos, y el cero. Asumiendo una coma fija, dicho formato permite también representar números con parte fraccionaria. Sin embargo, no se pueden representar número enteros muy grandes, ni fracciones muy pequeñas.

Representación de Números (Coma Flotante)

Con el formato de **Coma Flotante** se puede superar el principal inconveniente de representación de la coma fija, utilizando la notación científica: moviendo la coma hasta la posición del bit más significativo y aumentando el exponente, si se desplaza hacia la izquierda, o disminuyéndolo si se desplaza hacia la derecha.

Se pueden representar tanto números reales como enteros con un rango mayor al permitido por coma fija. Para lograr esto, se dividen a los n bits en tres campos: **Signo**, **Mantisa (parte significativa)**, y **exponente**. La base del número queda implícita y por lo tanto, no necesita espacio en memoria. En un formato típico de 32 bits de coma flotante:

- El bit más significativo contiene el signo del número (0 = positivo, 1 = negativo).
- El valor del exponente se almacena en las posiciones del 1 al 8 con un sesgo (un exceso, generalmente de $2^{k-1} - 1 = 127$, donde k es número de bits en el exponente binario: 8).
- Y la parte restante de bits es la mantisa, donde se almacena el número fraccionario en notación científica entera.


Dado que existen varias notaciones científicas equivalentes para un único número, se necesita utilizar una notación normalizada ($\pm 0,1\text{bbbb} \times 2^{\pm e}$). Donde b puede tomar valor cero o uno, e es el exponente, y el primer bit detrás de la coma siempre será un 1. Y dado que no es necesario almacenar dicho bit, se lo supone implícito y se utiliza esa posición para albergar un dígito más (**MANTISA FRACCIONARIA NORMALIZADA CON BIT IMPLÍCITO**).

Esta nomenclatura tiene el inconveniente de no poder representar al cero y a los números más cercanos, tanto positivos como negativos, debido al bit implícito.

El rango de representación para los negativos va desde $-(1 - 2^{-k}) \times 2^{128}$ a $(-0,5 \times 2^{-127})$ y para los positivos, desde $(0,5 \times 2^{-127})$ a $((1 - 2^{-k}) \times 2^{128})$; donde k es el número de bits de la mantisa.

Hay cuatro casos que quedan fuera de representación:

- Desbordamiento negativo:** número negativos menores que $-(1 - 2^{-k}) \times 2^{128}$.
- Desbordamiento a -0:** número negativos mayores que $-0,5 \times 2^{-127}$.
- Desbordamiento a +0:** número positivos menores que $0,5 \times 2^{-127}$.
- Desbordamiento positivo:** número positivos mayores que $(1 - 2^{-k}) \times 2^{128}$.

 Con la *coma flotante* se amplía el rango de números representables, pero se disminuye la precisión de los mismo. Debido a que al redondear o truncar los bits menos significativos de la mantisa, el número final se aproxima al deseado. Por esto, los números no están espaciados por igual a lo largo de la recta real, los posibles valores están más próximos cerca del origen y más separados a medida que alejan de él.

Error

Si incrementamos el número de bits del exponente, ampliamos el rango de números representables, pero reducimos la densidad y la precisión. Esta proporcionalidad inversa produce dos errores:

- **Error Absoluto:** Es el valor absoluto de la diferencia entre el número verdadero y el aproximado:


$$|\text{numeroARepresentar} - \text{numeroRepresentado}|$$

- **Error Relativo:** Es la razón entre el error absoluto y el número verdadero, en porcentaje:

$$\frac{|\text{numeroARepresentar} - \text{numeroRepresentado}|}{\text{numeroARepresentar}}$$

La única forma de incrementar, tanto el rango como la precisión, es utilizar más bits. Los formatos de 32 bits son conocidos como *Precisión Simple* y los de 64 bits como *Precisión Doble*.


Estándar IEEE 754

 Es la representación más importante en coma flotante de la forma 1.M, y fue desarrollado para facilitar la portabilidad de los programas de una CPU a otra y alentar al desarrollo de programas numéricos sofisticados. Utiliza una mantisa fraccionaria normalizada con bit implícito en BCS, y el exponente se representa en exceso a $2^{n-1}-1$.

En esta nomenclatura, no todas las combinaciones se interpretan de la misma manera. Algunas combinaciones se emplean para representar valores especiales:


- Para valores de exponente desde 1 hasta 254 (simple precisión) o desde 1 hasta 2046 (doble precisión), se representan números normalizados.
- Un exponente cero junto con una fracción cero representa cero positivo o negativo.
- Un exponente con todos unos con una fracción cero representa infinito positivo o negativo.
- Un exponente cero con una fracción distinta de cero representa un número desnormalizado, que permite representar los números antes excluidos con la forma de 0.M. Cuando el exponente del resultado es demasiado pequeño, el resultado se desnormaliza desplazando a la derecha la parte fraccionaria e incrementando el exponente hasta que dicho exponente este dentro de un rango representable.
- Un exponente de todos unos con un fracción distinta de cero, es un **NaN (Not a Number - No es un número)** y se emplea para señalar varias condiciones de excepción.

Representaciones Alfanuméricas

 Las **representaciones alfanuméricas** determinan la forma en la que se representará la información de tipo texto que genera la computadora. Donde cada atributo que una letra, un dígito o un símbolo, recibe el nombre de carácter. El cual es único dentro de su sistema, generándose de las distintas combinaciones de los bits que posea dicha representación, y cuyo orden dependerá de la misma.

Si bien existen distintos formatos en la actualidad, tales como ASCII, EBCDIC y ANSI; todos poseen el mismo grupo básico de caracteres:

- **Letras Minúsculas:** 'a', 'b', 'c',... 'x', 'y', 'z'.
- **Dígitos:** '0', '1', '2',... '8', '9'.
- **Letras Mayúsculas:** 'A', 'B', 'C',... 'X', 'Y', 'Z'.
- **Caracteres Especiales:** '@', '#', '¬', '¿', '?',...

 Además, cada sistema posee un grupo de caracteres adicionales conocidos como **“Caracteres de Control”**, los cuales se utilizan para controlar periféricos, entre otras cosas.

Modos De Direccionamientos

En una instrucción se utilizan bits para expresar el código de operación (**codop**). También se necesitan una “gran” cantidad de bits para especificar de dónde provienen los datos.

Los **MDD (Modos De Direccionamientos)** 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 más eficiente de datos (arreglos).

Notación	Significado
A	Contenido de un campo de dirección en la instrucción.
R	Contenido de un campo de dirección en la instrucción que referencia a un registro.
EA	Dirección real (efectiva) de la posición que contiene el operando que se referencia.
(X)	Contenido de la posición X.

Tabla 8: Notación a utilizar en la tabla de los tipos de modos de direccionamiento

Tipo de MDD		Algoritmo	Características
Inmediato		Operando = A	Se utiliza para definir constantes y para inicializar variables.
			Ventaja: No requiere una referencia extra a memoria de datos.
			Desventaja: tamaño del operando limitado por el tamaño del campo de direccionamiento.
Directo	Por memoria	EA = A	Ventaja: Es simple.
			Desventaja: Tiene un espacio limitado de direcciones por cantidad de bits del campo.
	Por registro	EA = R	Ventajas: La referencia a registro usa menos bits que la especificación de la dirección y no requiere acceso a memoria de datos.
			Desventaja: los registros no son muchos y es un recurso preciado.
Indirecto	Por memoria	EA = (A)	Ventajas: Con una dirección de menos bits en la instrucción, se apunta a una dirección de más bits y el espacio de direccionamiento mayor.
			Desventaja: múltiples accesos a memoria.
	Por registro	EA = (R)	Ventajas: Se necesitan menos bits para especificar el registro que la posición de memoria. Espacio de direccionamiento grande.
			Desventaja: Referencia extra a memoria
Por desplazamiento	Relativo	EA = PC + A	La dirección de la instrucción actual se suma al campo de dirección para producir la dirección efectiva.
			El campo de dirección se trata como un número en Ca_2 .
	De Registro Base	EA = A + (R)	El registro referenciado contiene una dirección de memoria y el campo de dirección tiene un desplazamiento.
			Proporciona un mecanismo eficiente para realizar operaciones iterativas.
Pila o Stack	Indexado	EA = (A) + R	Algunas máquinas incrementan o decrementan este registro como parte de la instrucción (Autoindexación).
			Es un arreglo lineal de localidades de memoria. Es una lista ó 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 Puntero de Pila , cuyo valor es la dirección tope de pila. También tiene dos punteros más los cuales son de control, el puntero a la Base de la Pila , y al Límite de la Pila .
			Los dos primeros elementos de la pila pueden residir en registros de la CPU, en cuyo caso, el puntero de pila hace referencia al tercer elemento de la pila.


Tabla 9: Tipos de Modos de Direccionamiento

Estructura y función de la CPU


Organización de los registros

Tipo de registro	Subtipo de Registro	Descripción
Visibles para el usuario (Permiten al programador del lenguaje máquina o ensamblador, minimizar las referencias a memoria principal cuando optimiza el uso de los registros)	Uso general	Pueden ser asignados por el programador a diversas funciones.
		Puede contener el operando para cualquier codop , para funciones de direccionamiento.
		Tiene restricciones, puede haber registros específicos de operación en coma flotante y operaciones de pila, por ejemplo.
	Datos	Pueden utilizarse únicamente para contener datos, y no se pueden emplear para el cálculo de una dirección del operando.
	Direcciones	Pueden ser de uso mas o menos particular, o a un modo de direccionamiento en particular.
		Punteros de segmento: En una máquina con direccionamiento segmentado, contiene la dirección de la base del segmento. Pueden haber varios registros (una para el OS y otra para el proceso actual).
		Registros índice: Se usan para direccionamiento indexado, y pueden ser auto indexados.
		Puntero de pila: Hay un registro apuntando a la cabecera de la pila.
	Códigos de condición (flags)	Son bits fijados por el hardware como resultado de alguna operación aritmético-lógica.
De control y estado	Contador de Programa (PC)	Contiene la dirección de la instrucción a captar.
	Registro de Instrucción (IR)	Contiene la instrucción captada más recientemente.
	Registro de Dirección de Memoria (MAR)	Contiene la dirección de una posición de memoria.
	Registro Intermedio de Memoria (MBR)	Contiene la palabra de datos a escribir en memoria, o la palabra leída más recientemente.


Tabla 10: Tipos de Registros

 La CPU actualiza el PC después de cada instrucción, con lo cual siempre tendrá la dirección de la siguiente instrucción a ejecutar. La instrucción captada se carga en el IR, donde son actualizados el **codop** y los campos de operandos. Se intercambian datos con la memoria por medio de **MAR** y **MBR**. El primero se conecta con el bus de direcciones y el segundo con el bus de datos. Los registros visibles para el usuario intercambian datos con **MBR**.

Assembly

 **Assembly** o lenguaje de ensamble es un lenguaje que, en principio, define un mapeo directo de **códigos** (**codops**, que en realidad son una secuencia de bits con un significado en particular para la máquina) a **instrucciones comprensibles por un humano**.

Y el programa encargado de tomar programas escritos en assembly y generar los códigos que la computadora entenderá se lo denomina **assembler o ensamblador**.

 No todas las instrucciones de assembly se corresponden directamente con instrucciones de lenguaje de máquina, sino que varias son instrucciones para el ensamblador mismo.

Lógica digital

Identidades del Álgebra Booleana

Función	Con Respecto a la Multiplicacion	Con Respecto a la Suma
Identidad	$1 \cdot A = A$	$0 + A = A$
Nula	$0 \cdot A = 0$	$1 + A = 1$
Idempotencia	$A \cdot A = A$	$A + A = A$
Inversa	$A \cdot \neg A = 0$	$A + \neg A = 1$
Conmutativa	$A \cdot B = B \cdot A$	$A + B = B + A$
Asociativa	$(AB) \cdot C = A \cdot (BC)$	$(A + B) + C = A + (B + C)$
Distributiva	$A + (B \cdot C) = (A + B) \cdot (A + C)$	$A \cdot (B + C) = (AB) + (AC)$
Absorción	$A \cdot (A + B) = A$	$A + (AB) = A$
De Morgan	$\neg(A \cdot B) = \neg A + \neg B$	$\neg(A + B) = \neg A \cdot \neg B$

Tabla 11: Identidades del Álgebra Booleana

Puertas Lógicas



Una puerta es un circuito electrónico que produce como señal de salida una operación booleana sencilla de las señales de entrada. Son el bloque fundamental de construcción de todos los circuitos lógicos digitales.

Las puertas básicas utilizadas son: AND, OR, NOT, NOR, NAND.

Cada puerta puede tener muchas entradas y una salida. Cuando los valores de entrada cambian, la señal de salida correcta aparece casi instantáneamente, retardada solamente por el tiempo de propagación de la señal a través de la puerta, conocido como retardo de puerta.

Formas de definir una Puerta	Descripción
Símbolo	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.

Tabla 12: Formas de definir una Puerta Lógica

Circuitos Combinacionales



Un circuito combinacional es un conjunto de puertas interconectadas, cuya salida, en un momento dado, es función solamente de la entrada en ese instante

Formas de definir un Circuito Combinacional	Descripción
Símbolo	Describe la organización de la interconexiones entre puertas.
Función algebraica	Cada señal de salida se expresa como una función booleana de las señales de salida
Tabla de verdad	Para cada una de las 2^n combinaciones posibles de las n señales de entrada se enumera el valor binario de cada una de las m señales de salida.

Tabla 13: Formas de definir un Circuito Combinacional

Multiplexores

- ✓ Un Multiplexor conecta varias entradas con una única salida. En un momento dado, se selecciona una de las entradas para que pase a la salida.

Los multiplexores se usan en circuitos digitales para controlar el flujo de señales y datos. Un ejemplo es la carga del contador del programa. El valor a cargar en el PC se puede venir de una o varias fuentes diferentes:

- **De un contador binario**, si el PC se va a incrementar para la siguiente instrucción.
- **Del registro de instrucción**, se acaba de ejecutar una instrucción de salto usando direccionamiento directo.
- **De la salida de la ALU**, si la instrucción de salto especifica la dirección usando modo de desplazamiento.

Las distintas entradas se pueden conectar a la líneas de entrada de un multiplexor con el PC conectado a la línea de salida. Las líneas seleccionadas determinan cual el el valor a cargar en el PC. Como el PC contiene varios bits se utilizan varios multiplexores, uno por bit.

Circuitos Secuenciales

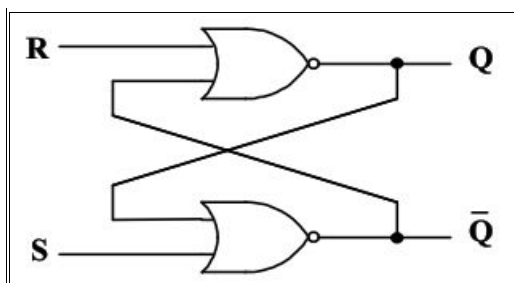
- ✓ La salida actual depende de la entrada actual y del estado actual del circuito, **proporcionando así memoria o información de estado.**

Biestable

La forma mas sencilla de un circuito secuencial es un biestable. Hay varios tipos de biestables, y todos comparten dos propiedades:

- **Es un dispositivo con dos estados.** En ausencia de entrada, recuerda el último estado. Por eso, puede funcionar como memoria de un bit.
- **Tiene dos salidas, que son siempre complementarias.** Normalmente son Q y $\sim Q$.

Biestable S-R



Dibujo 2: Implementación del biestable S-R con puertas NOR

- ✓ El circuito consta de dos entradas, S (set) y R (reset), y dos salidas, Q y $\sim Q$, y consiste en dos puertas NOR conectadas en retroalimentación.

Supongamos que S y R valen 0 y que Q es 0. Las entradas a la puerta NOR inferior son $Q = 0$ y $S = 0$, entonces $\sim Q = 1$, entonces $Q = 0$. Por lo tanto, el circuito permanece internamente consistente y estable mientras $R = S = 0$.

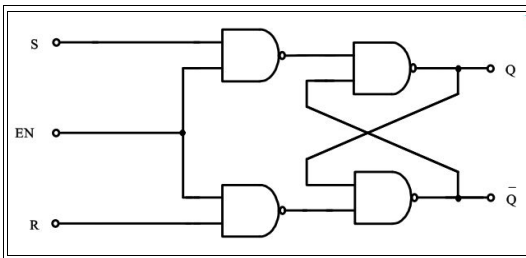
Entonces podemos tomar a Q como el “valor” del bit. Las entradas S y R sirven para escribir los valores, 1 y 0.

- i El S-R presenta error cuando las entradas $S = R = 1$. Estas entradas no están permitidas, ya que produciría una salida inconsistente: $Q = \sim Q = 0$.

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

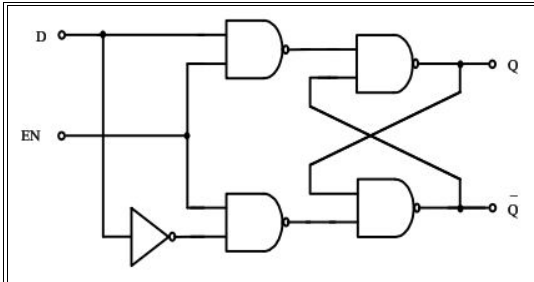
Tabla 14: Tabla característica de biestable S-R sincrónico y asincrónico

Biestable S-R Sincrónico



Dibujo 3: Implementación del biestable S-R con puertas NAND

✓ Es similar al Biestable S-R, con la diferencia de que las entradas S y R se aplican a las entradas de las puertas sólo durante el pulso de reloj.



Dibujo 4: Implementación de un biestable D con puertas NAND y NOT

Biestable D

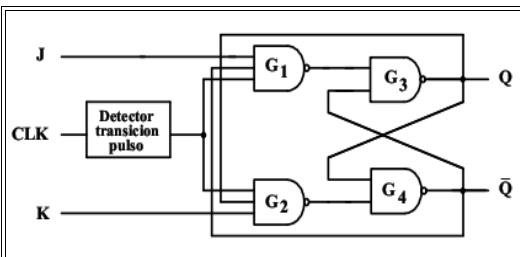
✓ El biestable D permite solucionar el problema de los biestables S-R usando un inversor, y garantizando que las entradas que no son del reloj de las puertas sean opuestas.

El biestable D se denomina a veces “**biestable de datos**” porque es, en efecto, almacén para un bit de datos.

La salida del biestable D es siempre igual al valor más reciente aplicado a la entrada. Por lo tanto, recuerda y produce la última entrada.

También se le llama “biestable de retardo”, porque retrasa un 0 o un 1 aplicado a la entrada durante un pulso de reloj.

Biestable J-K



Dibujo 5: Implementación de un biestable J-K con puertas NAND

✓ En el biestable J-K todos los valores de entrada son válidos.

La entrada J sola realiza la función de puesta a 1, causando que la salida sea 1; la entrada K sola, realiza la función de puesta a 0, provocando que la salida sea 0. Cuando J y K son 1, la función realizada se denomina función de conmutación: la salida se invierte. Si Q vale 1 y se aplica 1 a J y K entonces Q se hace 0.

Periféricos

✓ Se denominan periféricos tanto a las unidades o dispositivos a través de los cuales el computador se comunica con el mundo exterior, como a los sistemas que almacenan o archivan la información, sirviendo de memoria auxiliar de la memoria principal.

i La memoria auxiliar trata de suplir las deficiencias de la memoria principal como su baja capacidad y el hecho de que la información almacenada se borra al eliminar la alimentación de energía eléctrica.

Según la definición de periférico, éstos están constituidas por:

- **Unidad(es) de entrada.**
- **Unidad(es) de salida.**
- **Unidad(es) de Almacenamiento masivo o memoria auxiliar.**

Los **dispositivos de E/S transforman la información externa en señales codificadas**, permitiendo su transmisión, detección, interpretación, procesamiento y almacenamiento de forma automática. Los **dispositivos de Entrada transforman la información externa según alguno de los códigos de E/S**. En un **dispositivo de Salida se efectúa el proceso inverso**.

! **¡Hay que distinguir entre periféricos de un computador y máquinas auxiliares de un determinado servicio informático!**

✓ **Máquina auxiliar:** No está conectada físicamente al computador, sirven para preparar o ayudar en la confección o utilización de la información que se da a, o produce, el computador.

! **¡Tampoco confundir periférico con soporte de información!**

✓ **Soporte de información:** Se entiende aquellos medios físicos sobre los que va la información

Conexión de periféricos al computador

Las unidades funcionales del computador, se comunican por conjuntos o grupos de hilos denominados canales o buses de comunicación.

El bus del sistema, recorre toda la placa base de una computador de escritorio y los periféricos se interconectan al mismo directamente o a través de interfaces.

✓ Las **interfaces** son para **adaptar las características de los periféricos a las del bus del sistema**. Son las encargadas de controlar el funcionamiento de los periféricos y de proporcionar al sistema operativo, la información necesaria para manejar de forma efectiva, la actividad de E/S.

Las interfaces cubren tres objetivos:

- **Conversión de datos:** Adaptan la representación de datos del bus del sistema a la representación de datos del periférico (serie a paralelo y viceversa)
- **Sincronización:** La velocidad operativa del computador central suele ser mucho mayor que la de los periféricos. La interfaz regula el tráfico de información para que no se den problemas de sincronización o pérdidas de información. Los periféricos(o interfaces) incluyen una memoria intermedia o “**buffer**”, efectuándose el tráfico de datos entre el periférico y el bus a través de ella.
- **Selección de dispositivos:** se encargan de identificar la dirección del periférico que debe intervenir en tráfico de datos, generalmente , solo uno de ellos debe estar conectado lógicamente al bus de datos.

Ventajas:

- Independizar a los fabricantes de periféricos y placas controladoras de los fabricantes de procesadores y mainboards.
- Establecer normas generales para la comunicación entre cualquier clase de periféricos y los procesadores de modo que sean “conectables” a cualquier procesador.

Es interesante notar que estas placas de interfaz tienen muchas veces su propio procesador, memoria y su programa de control (“**driver**”) que les permite manejar una variedad de periféricos similares con prestaciones adaptables a lo que quiere el usuario.

Estas interfaces regulan las operaciones de E/S a través de tres técnicas:

- **E/S programable:** Los datos se intercambian entre la CPU y el módulo de E/S.

La CPU ejecuta el programa que controla las operaciones de E/S, envía una orden al modulo de E/S y debe esperar que este concluya.

La **desventaja** de esta técnica está el desaprovechamiento del tiempo de la CPU, ya que si la misma es más rápida que el modulo de E/S, esta se queda esperando a que termine la operación. Además, la CPU termina siendo un puente entre la memoria y el módulo, por ser la encargada de transferir los datos.

- **E/S mediante interrupciones:** Es la alternativa a la técnica anterior, dado que la CPU proporciona la orden de E/S y continúa trabajando con otras instrucciones, hasta que es interrumpida por el modulo de E/S, cuando este ha terminado su trabajo. Si bien se presenta una mejora al no desperdiciar el tiempo de la CPU, sigue siendo un puente en la memoria y el módulo de E/S.
- **DMA (Direct Memory Access - Acceso Directo a Memoria):** Es la alternativa a las dos técnica anteriores, debido a que no se utiliza a la CPU y a que la memoria y el módulo de E/S intercambian los datos a través del dispositivo de DMA. El cual es capaz de imitar a la CPU, y de recibir el control del sistema cedido por el

mismo.

Dicho control, lo necesita para transferir los datos a, y desde, la memoria a través del bus del sistema. Para lograr esto, el DMA utiliza el bus del sistema solo cuando la CPU no lo utiliza, o debe forzar a la misma a que suspenda temporalmente su funcionamiento. A esta técnica se la denomina **Robo de ciclo**.

Cuando la CPU desea leer o escribir un bloque de datos en un dispositivo de E/S, envía una orden al DMA, incluyendo la siguiente información:

- Si se solicita lectura o escritura, utiliza la línea de control de lectura o escritura entre la misma y el DMA.
- La dirección del dispositivo de E/S es indicada a través de las líneas de datos.
- La posición inicial de lectura o escritura se almacena en el registro de direcciones del DMA.
- El número de palabras a leer o escribir, también indicada por las líneas de datos.

La CPU ha delegado la operación de E/S al DMA y continúa con otro trabajo. El DMA transfiere los datos hacia la memoria y cuando termina, interrumpe a la CPU. Por lo tanto, la CPU solo interviene al principio y al final de la transferencia.

Tipo de Interfaces

Las interfaces pueden ser de dos tipos, dependiendo de cómo se transfiera la información:



Paralelas: Son aquellas en donde existen varias líneas que conectan al módulo de E/S con el periférico, y transfieren varios bits simultáneamente a través del bus de datos.



Estas interfaces se utilizan para dispositivos de alta velocidad como los discos rígidos.



Seriales: Son aquellas en donde solo existe una línea para transmitir los datos, y en donde los bits deben transmitirse uno a uno.



Estas interfaces se utilizan generalmente para impresoras y terminales.

Características Generales

Un periférico suele estar **formado** por **dos partes**: una **mecánica**(conmutadores manuales, motores, electro imanes, etc.) y otra **electrónica** (circuitos de la interfaz).

Cuando un periférico actúa sin intervención del computador central se dice que trabaja “**off line**”, caso contrario trabaja “**online**”.

Otra **características** de los periféricos y de los soportes de información son:

- **Fiabilidad:** probabilidad de que se profundiza un error en la E/S
- **Duración:** es la permanencia sin alteración de los datos a lo largo del tiempo.
- **Densidad:** cantidad de datos(bits o caracteres) contenidos por unidad de volumen, superficie o longitud ocupada.
- **Reutilización:** permitir o no guardar nueva información sobre datos que ya resultan obsoletos.
- **Tipo de acceso:** (vinculada o un lector/grabador) de tipo secuencial o directo.
- **Tiempo de acceso:** tiempo promedio que necesita un dispositivo para leer/grabar un dato.
- **Velocidad de Transferencia:** cantidad de información que el dispositivo es capaz de leer/grabar por unidad de tiempo.
- **Transportabilidad:** es transportable si es susceptible de ser trasladado de una unidad periférica a otra.
- **Ergonomía:** es ergonómico cuando su diseño físico externo se adapta al usuario, obteniéndose una buena integración hombre-máquina y una adecuada eficiencia en su utilización haciéndose cómodo su uso al hombre.

Clasificación

Tipo de clasificación	Subtipos
Por tipo de periférico	Unidades de entrada.
	Unidades de salida.
	Unidades de almacenamiento masivo.
	Unidades de E/S mixtas.
Por distancia a la que se encuentra	Locales: Se encuentran cerca de computador.
	Remotos: Se encuentran en situación lejana, la conexión hay que realizarla a través de líneas especiales de transmisión.


Periféricos Específicos

Terminales: Permiten al usuario interactuar directamente con el computador, se componen de dos elementos: teclado y monitor (a veces otros elementos auxiliares).

Teclado

Se utiliza para introducir órdenes e información al computador. Cuando una tecla es pulsada, se realiza la conversión de su posición, en una matriz de contactos, a un código alfanumérico. Seguidamente se envía el código al computador.

Monitor

 **Monitor:** Es un dispositivo de salida que, mediante una interfaz, muestra los resultados del procesamiento de una computadora. El más difundido es el Tubo de Rayos Catódicos (TRC) que puede ser monocromático o color, existen también de Cristal Líquido o Plasma.

Su construcción comprende de un tubo de vacío, semejante a una pirámide de base rectangular con un alargamiento en su vértice. En la parte mas estrecha lleva un filamento que permite producir un haz de electrones. En el cuello lleva dos bobinas que permiten modificar la trayectoria del haz en las dos direcciones X e Y hacia la base del tubo. Esta base, está recubierta por una capa fluorescente, que se ilumina al recibir el impacto del haz. La representación de letras o figuras se hace barriendo toda la superficie del tubo de una forma sistemática, y modulando la intensidad del haz de acuerdo a la información a representar.

Modo entrelazado

Un monitor entrelazado es uno en el cual los rayos de electrones no se dibujan en forma lineal, sino de forma entrelazada (línea de por medio), y cuando llega al final de la pantalla, regresa arriba para llenar las líneas anteriormente no refrescadas. Un monitor entrelazado con refresco de 100 Hz solamente refresca una línea cincuenta veces por segundo, dando un titileo obvio. Un monitor no entrelazado es el que dibuja todas las líneas en cada pasada antes de refrescar el cuadro siguiente, resultando una imagen más nítida.

Velocidad de Refresco

La velocidad de refresco o frecuencia vertical, es medida en Hertz y representa el número de cuadros mostrados en la pantalla por segundo. Si son pocos, el ojo notará los intervalos intermedios y verá que los objetos titilan en la pantalla. La velocidad de refresco aceptada en el mundo para una pantalla libre de titileos es de 70 Hz para arriba.

Los monitores se clasifican según:

- **Los colores que muestran:**
 - **Monocromáticos:** Pueden mostrar la información en blanco y negro, en ámbar o en verde.
 - **A color:** Pueden mostrar la información con las combinaciones del rojo, del azul y del verde.
- **Como representan la información:**
 - **De Caracteres:** Ya fuera de uso, son aquellos monitores que poseen dos memorias para representar únicamente caracteres: una del tipo RAM donde se guarda la información de cada carácter, con su atributo; y una memoria del tipo ROM, donde se almacena los patrones de cada carácter almacenado.
 - **Gráficas:** Son aquellos monitores que utilizan un trazado de líneas para mostrar la información.

Poseen solo una memoria del tipo ROM, donde se almacenan la intensidad, el color, y otros atributos posible para cada pixel representando en la resolución de pantalla.

- **Tamaño de pantalla** (en pulgadas).
- **Resolución de pantalla** (en pixeles):
 - **CGA** (640 * 200).
 - **VGA** (640 * 480).
 - **SVGA** (1024 * 768).

Elementos Auxiliares

Lápiz óptico: Permite marcar un punto de la pantalla de TRC. Esta señal permite determinar la micro línea y la posición de la misma en donde está el lápiz, lo que permite conocer su posición en la pantalla.

Impresoras

Su objetivo es producir texto. Sin embargo, algunos tipos tienen prestaciones gráficas. Según sus características se puede clasificar por:

- La forma de imprimir:
 - **De caracteres:** Imprimen carácter por carácter de manera unidireccional o bidireccionalmente, por lo que son muy lentas.
 - **De línea:** Imprimen una línea de caracteres simultáneamente, por lo que son muy rápidas.
 - **De página:** Imprimen toda una pagina en una sola pasada, aunque internamente imprimen línea por línea.
- El mecanismo de impresión:
 - **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, bola, cilindro, etc.
 - Cinta.
 - Tambor.
 - **Térmicas:** Son aquellas que imprimen mediante la transferencia de calor al cabezal, a través de una matriz de pequeñas resistencias, 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 mas rápidos).
 - **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.

Módem



Módem (MODulador - DEModulador): Es un dispositivo, externo o interno, utilizado para la comunicación entre computadoras a través de líneas telefónicas.

El módem convierte las señales digitales en analógicas (**modularización**), y las envía por la línea de teléfono a la que deben estar conectados el emisor y el receptor. Cuando la señal llega a su destino, otro módem se encarga de reconstruir la señal digital primitiva (**demodularización**), de cuyo proceso se encarga la computadora receptora.

Además, un módem está programado para ser tolerantes a errores de transmisión.

Índice de contenido

La Máquina De Von Neumann - IAS.....	3
Ciclo de Instrucción.....	3
Interrupciones.....	4
Diseño del conjunto de instrucciones.....	4
Formato de instrucción.....	5
Memoria.....	6
Direccionamiento 2D (Memoria Estática):.....	6
Direccionamiento 2D ^{1/2} (Memoria Dinámica):.....	7
Direccionamiento 3D (Modelo Teórico):.....	7
Sistemas de Numeración.....	8
Capacidad de representación, resolución y rango en sistemas restringidos a n bits	8
Representación de números (Coma Fija).....	9
Representación de Números (Coma Flotante).....	9
Error.....	10
Estándar IEEE 754.....	10
Representaciones Alfanuméricas.....	10
Modos De Direccionamientos.....	10
Estructura y función de la CPU.....	12
Organización de los registros.....	12
Assembly.....	12
Lógica digital.....	13
Identidades del Álgebra Booleana.....	13
Puertas Lógicas	13
Circuitos Combinacionales.....	13
Multiplexores.....	14
Circuitos Secuenciales.....	14
Biestable.....	14
Biestable S-R.....	14
Biestable S-R Sincrónico.....	14
Biestable D.....	15
Biestable J-K.....	15
Periféricos.....	15
Conexión de periféricos al computador.....	16

Tipo de Interfaces.....	17
Características Generales.....	17
Clasificación.....	18
Periféricos Específicos.....	18
Teclado.....	18
Monitor.....	18
Modo entrelazado.....	18
Velocidad de Refresco.....	18
Impresoras.....	19
Módem.....	19

Aprobé!!!!!!!!!!!!!!!!!!!!

