

Preguntas teóricas - Final OC 🧚‍♀️ ✨ 💻

1. ¿Qué es 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 combinatorial (combinatorio) consiste en n entradas y m salidas. (Ejemplos son: Multiplexor, Demultiplexor, Codificador, Decodificador y Sumador)

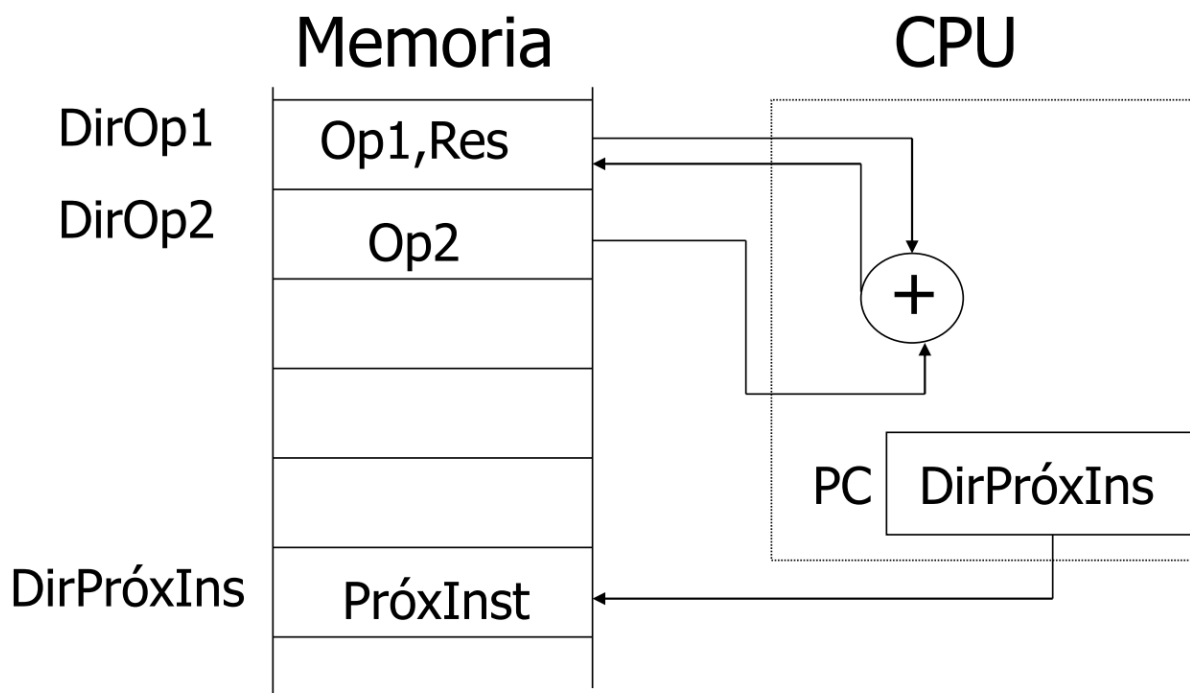
2. MSX88 representa una máquina de 2 direcciones?

Si, explicar máquina de 2 direcciones, por ej ADD AX, BX, aunque existan operaciones de 1 dirección, la mayoría son de 2 direcciones. Corrección por Runco: El MSX88 simula ser una máquina de dos direcciones, esto es así porque en uno de los operandos hay una dirección pero en el otro hay un registro.

Máquina de 2 direcciones:

Tiene un compromiso entre la versatilidad y la rapidez.

Hay que MOVER el operando 1 a un REGISTRO TEMPORAL, tengo menos elección donde guardar el resultado.



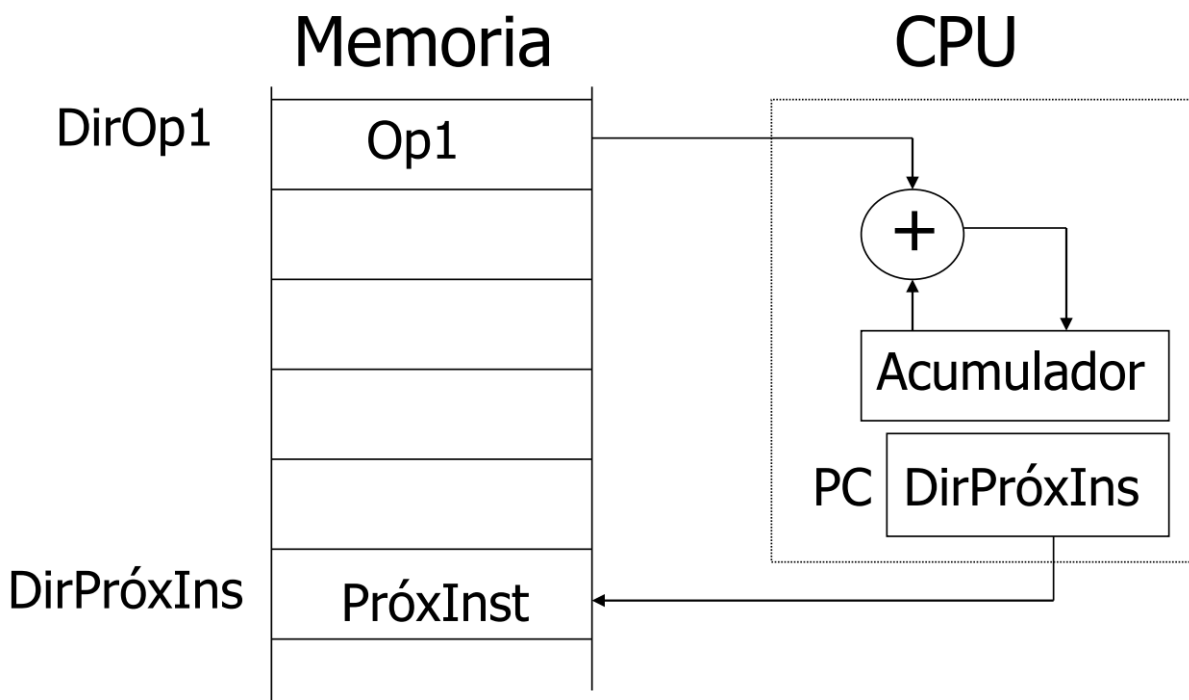
3. Explicar máquina de una dirección

Registros especiales en la CPU (**acumulador**)

Instrucciones para cargar(load) y descargar(store) el acumulador.

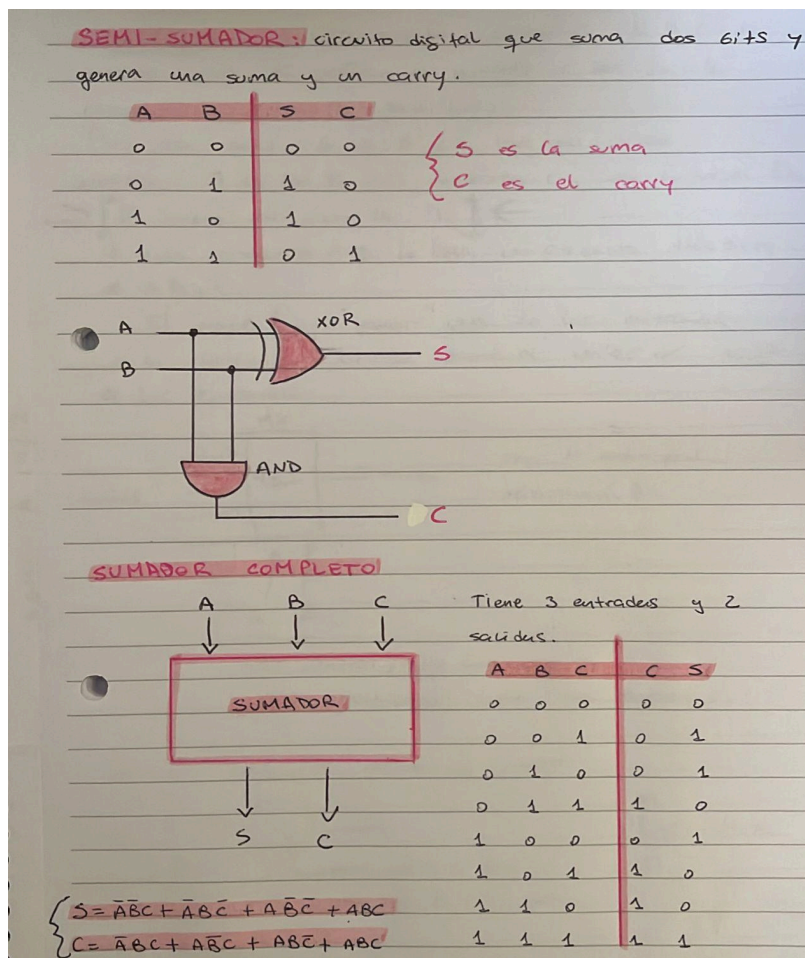
Un operando y resultado en lugar predefinido

Obs: poca versatilidad de dónde pongo los operandos



4. Explicar que es el "sumador completo". Realizar tabla de verdad y gráfico

Es un circuito digital que suma 3 entradas: A, B y C, y genera dos salidas: S (suma) y C (carry). A diferencia del semisumador que solo tiene 2 entradas con las mismas salidas.



5. ¿Qué es el cilindro? Nombrar las Diferentes pistas

Conjunto de pistas homólogas

¿Qué es el decodificador? (pdf pág 46 gráfico)

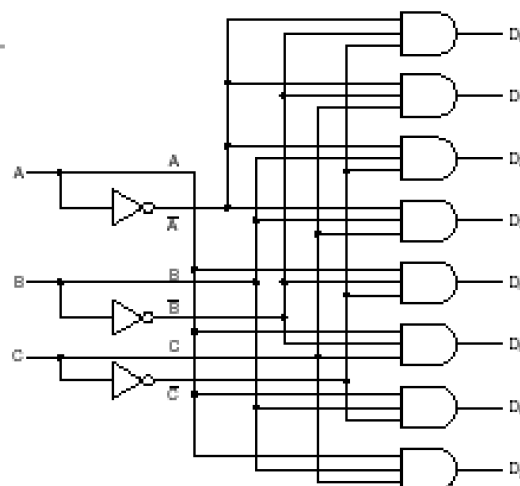
Un decodificador es un circuito lógico con entradas y salidas, tal que para cada combinación de entradas se activa al menos una salida. Puede servir para convertir una información codificada, como puede ser un número binario, en una información no codificada como puede ser un número decimal. Dispone de una entrada de habilitación (EN: Enable) que conecta o desconecta el dispositivo y puede ser activa a nivel bajo, ya que el dispositivo se activa cuando dicha entrada recibe un '0' lógico o activa a nivel alto, que se activa al recibir un '1' lógico.

Ejemplo 2

Para cada combinación de las entradas A, B y C

sólo UNA de las salidas D_x vale '1'

Decodificador 3 a 8



Formato simple y doble precisión IEEE 754 mantisa y exponente casos especiales

◊ ESTANDAR DEL IEEE754.

Este estándar se desarrolló para facilitar la portabilidad de los programas de un procesador a otro y para alentar el desarrollo de programas numéricos sofisticados. Ha sido ampliamente adoptado y se utiliza prácticamente en todos los procesadores actuales. El estándar del IEEE define el formato para precisión simple de 32 bits y para precisión doble de 64 bits.

• PRECISIÓN SIMPLE (32 bits).

SIGNO (1 bit)	EXPONENTE (8 bits)	MANTISA (23 bits)
---------------	--------------------	-------------------

• PRECISIÓN DOBLE (64 bits).

SIGNO (1 bit)	EXPONENTE (11 bits)	MANTISA (52 bits)
---------------	---------------------	-------------------

- EXPONENTE sin signo y en exceso $2^{n-1} - 1$ (127 para precisión simple y 1023 para precisión doble).

• CASOS ESPECIALES:

SIGNO	EXPONENTE	MANTISA	SIGNIFICADO
0	Todos 1	Todos 0	Más infinito ($+\infty$).
1	Todos 1	Todos 0	Menos infinito ($-\infty$).
0 ó 1	Todos 1	Distinta de 0	Not a Number (NaN).
0 ó 1	Todos 0	Todos 0	Ceros (Positivo y Negativo).
0 ó 1	Todos 0	Distinta de 0	Número muy pequeño (cerca al cero).

Fórmula de capacidad de discos

$$\frac{\text{BYTES}}{\text{SECTOR}} \times \frac{\text{SECTORES}}{\text{PISTA}} \times \frac{\text{PISTAS}}{\text{CARA}} \times \text{platos} \times 2 \text{ (ya que cada plato tiene 2 caras)}$$

EJEMPLO

Hallar capacidad de un disco que tiene 4 platos, 3600 pistas/caras, 580 sectores/pistas , 256 bytes/sector

SOLUCIÓN

$$256 \times 580 \times 3600 \times 4 \times 2 = 42762240000 \text{ BYTES}$$

¿Qué es la ALU?

La unidad aritmética y lógica o ALU, es un circuito combinacional capaz de realizar operaciones aritméticas y lógicas con números enteros y también con números reales. Las operaciones que puede efectuar vienen definidas por el conjunto de instrucciones aritméticas y lógicas de las que dispone el juego de instrucciones del computador. Las operaciones aritméticas habituales que puede hacer una ALU incluyen suma, resta, multiplicación y división. Además, se pueden incluir operaciones específicas de incremento positivo (+1) o negativo (-1). Dentro de las operaciones lógicas se incluyen operaciones AND, OR, NOT, XOR, operaciones de desplazamiento de bits a la izquierda y a la derecha y operaciones de rotación de bits.

BUS DE DATOS

Proporcionan un camino para transmitir datos entre los módulos del sistema. El conjunto constituido por estas líneas, se denomina bus de datos. Consta generalmente de 8, 16 o 32 líneas distintas (anchura del bus). Puesto que cada línea solo puede transportar un bit cada vez, el ancho del bus determina cuántos bits se pueden transferir al mismo tiempo.

(Ejemplo: Si tengo que transportar 5 bytes y mi bus es de 5 bits, debo pasar 5 veces porque traigo un byte, paso por la unidad de control y voy a decodificarlo)

(conviene leer la página 26 del pdf resumen de la materia, explica los buses)

(PERIFÉRICOS EN LA PÁGINA 40)

Monitor y monitor alfanumérico

Los monitores alfanuméricos en memoria se almacenan solo códigos de carácter ASCII, los códigos de carácter se convierten en pixels por una ROM de caracteres y por cada carácter se generan varios pixels sucesivos en varias líneas sucesivas

¿Cómo se capta una instrucción?

DESCRIPCIÓN DE CÓMO SE CAPTA UNA INSTRUCCIÓN

básicamente según Runco: traer una instrucción de memoria y decodificarla 😊

Cuando captó una instrucción, tengo la dirección cargada en la PC de la CPU lo pone en el MAR y de ahí lo presenta en el bus de direcciones, una vez que lo captó necesito señales de control (lectura y escritura) para saber que vamos a hacer con esa dirección. Supongamos que es de Lectura

La memoria dice que quiere leer en esta dirección de memoria que es mía, Por lo tanto voy a dejar el dato, la celda contenida en el bus de datos y de ahí pasa al MBR (registro que tienen relación entre el Bus de datos y la CPU) de ahí pasa por el IP, una vez que pasa por ahí, se decodifica la instrucción, se analiza por el CODOP, se sabe qué instrucción es, se sabe quién va a participar y se analiza los modos de direccionamiento de esa instrucción y la UC pasa a buscar los operandos, una vez que ya los tiene cargados, los pasa a la ejecución y ese resultado lo deja en un registro.

Relación entre partes del disco: copiar audio de wsp.

Cada PLATO tiene dos CARAS, el conjunto de platos es la UNIDAD DE DISCOS, dentro de cada cara tenemos cierta cantidad de PISTAS y en cada pista (usualmente) la misma cantidad de SECTORES, entonces, **FALTA TERMINAR!!!** las pistas están formadas por sectores, el conjunto de pistas de una cara va a tener la capacidad de esa CARA si yo multiplico eso por la cantidad de caras que va a ser 2 x disco me va a dar la cantidad total

Llamamos cilindro al conjunto de pistas homólogas en las distintas caras

El manejo a través de cilindros se hace para optimizar el tiempo de acceso porque estoy más tiempo en cada cilindro trabajando y me ahorro los tiempos de búsqueda

Notación en punto fijo y punto flotante

Punto Fijo: Todos los números a representar tienen exactamente la misma cantidad de dígitos y la coma fraccionaria siempre está ubicada en el mismo lugar y esta coma no se va a almacenar ya que se supone está en un lugar determinado

Punto Flotante: Estos surgen como manera para poder representar números muy grandes o muy pequeños que en punto fijo no se pueden representar porque había cierta limitación

Resolución que es y cómo se calcula en Punto flotante

La resolución en punto flotante no es constante a lo largo del intervalo como sí ocurre en la representación en BSS. Esta se calcula en el extremo superior con la diferencia entre el número mayor y anterior y la resolución en el extremo inferior es la diferencia entre el número más mínimo y su siguiente.

Resolución + grande más lejos del cero

Resolución - pequeña más cerca del cero

Modem que es Y que son ad/da, banda ancha y funcionalidades

Un módem es un dispositivo que permite la conexión de dispositivos a una red de datos, convirtiendo las señales digitales de un dispositivo en señales analógicas para transmitirlos a través de líneas telefónicas o redes de comunicación, y viceversa, convirtiendo las señales analógicas en digitales para que los dispositivos las puedan entender y procesar.

Funcionalidades del modem: Discado, rediscado (el modem puede contestar), cada modem acuerda la velocidad de transmisión, pueden comprimir datos.

Banda ancha: el módem modula frecuencias más altas.

Modem consta de dos partes principales:

Modulador que convierte señales digitales en señales analógicas (modulación) Ej: modulación de amplitud en cuadratura (QAM) y la modulación de frecuencia (FM)

Demodulador que convierte las señales analógicas en señales digitales (demodulación) Ej: demodulación de amplitud en cuadratura (QAM) y la demodulación de frecuencia (FM)

Los AD y DA son las siglas de "analógico a digital" y "digital a analógico". Estos términos se refieren a los procesos de conversión de señales analógicas en señales digitales y viceversa.

Modos de direccionamientos decir los tipos. Qué es y su objetivo.

- | | |
|--|---------------------------------|
| → Inmediato | → Indirecto por registro |
| → Directo | → Por desplazamiento |
| → Por registro | → Del stack (no lo vemos) |
| → Indirecto por memoria
(msx no lo tiene) | |

Palabras de Runco:

son distintas maneras de expresar dónde están los operandos en las instrucciones, no hay uno solo porque al tener varios modos tenemos los 3 beneficios de acá abajo:

Los mdd 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)

- **Salto condicional:** tienen direccionamiento relativo al program counter (PC)
- **Directo:** MOV AX, dirección
- **Indirecto:** MOV AX, [BX] (solo con BX podemos usar corchetes!!!)
- **Por registro:** MOV AX, BX
- **Indirecto por registro:** MOV AX, [BX]
- **Indirecto por registro con desplazamiento:** MOV AX, [BX + n]
 - n = desplazamiento, le damos algún valor

Rom de caracteres

¿Qué define el Teorema Fundamental de la Numeración?

Teorema Fundamental de la Numeración

$$N^{\circ} = \sum_{i=-m}^n (\text{dígito})_i \times (\text{base})^i$$

$$\dots + x_4 \times B^4 + x_3 \times B^3 + x_2 \times B^2 + x_1 \times B^1 + x_0 \times B^0 + x_{-1} \times B^{-1} + x_{-2} \times B^{-2} + \dots$$

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

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

La base me dice la cantidad de símbolos que va a tener un sistema. Por ejemplo en base 10 tenemos del 0 al 9.

¿Qué es una puerta lógica?

Compuertas son dispositivos electrónicos que pueden realizar distintas funciones con estos dos valores lógicos como vimos en el Ingreso las compuertas básicas son: AND, OR, NOT.

Describe las diferencias entre Circuito combinatorio y uno secuencial:

Circuitos Combinacionales o Combinatorios: Responden a los valores lógicos en las entradas, la salida está determinada exclusivamente por los valores de las entradas en ese instante.

Si cambia la entrada, cambia la salida.

Los valores pasados de las entradas no influyen en los valores de las salidas.

Circuitos Secuenciales: Las salidas dependen tanto de las entradas como del estado interno del circuito. ¿Qué es el estado interno del circuito? Tienen la característica de “almacenar” valores lógicos internamente. Estos valores se almacenan aunque las entradas no estén.

Según la manera en que las salidas respondan a las señales lógicas presentes en la entrada, los biestables se clasifican en:

SR (Set-Reset)

J-K

D(Delay)

T (Toggle)

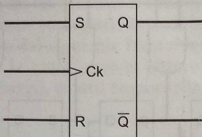
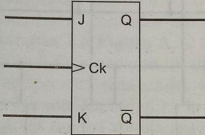
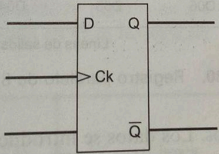
Nombre	Símbolo gráfico	Tabla característica															
S-R		<table><tr><th>S</th><th>R</th><th>Q_{n+1}</th></tr><tr><td>0</td><td>0</td><td>Q_n</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>—</td></tr></table>	S	R	Q_{n+1}	0	0	Q_n	0	1	0	1	0	1	1	1	—
S	R	Q_{n+1}															
0	0	Q_n															
0	1	0															
1	0	1															
1	1	—															
J-K		<table><tr><th>J</th><th>K</th><th>Q_{n+1}</th></tr><tr><td>0</td><td>0</td><td>Q_n</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>\bar{Q}_n</td></tr></table>	J	K	Q_{n+1}	0	0	Q_n	0	1	0	1	0	1	1	1	\bar{Q}_n
J	K	Q_{n+1}															
0	0	Q_n															
0	1	0															
1	0	1															
1	1	\bar{Q}_n															
D		<table><tr><th>D</th><th>Q_{n+1}</th></tr><tr><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td></tr></table>	D	Q_{n+1}	0	0	1	1									
D	Q_{n+1}																
0	0																
1	1																

Figura A.29. Biestables básicos.

Figura A.29. Biestables básicos.

Respecto del instante en que pueden cambiar dichas salidas, pueden ser:

Asincrónicos: cuando en la entrada se establece una combinación, las salidas cambiarán

Sincrónicos: la presencia de una entrada especial, determina “cuando” cambian las salidas acorde a las entradas.

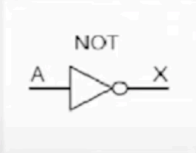
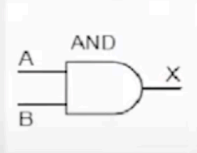
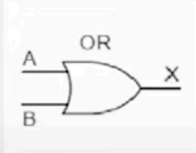
Memoria:

Se puede construir con un flip-flop una memoria de 1 bit.

➤ Se llama biestable porque el circuito posee sólo 2 estados posibles de funcionamiento, se queda en cada uno de ellos, salvo que las entradas provoquen un cambio.

¿Cuáles son las tres formas que se pueden utilizar para representar una función lógica?

- **Gráfica:** Describe la organización de las interconexiones entre puertas.
- **Tabla de verdad:** Para cada 2^n combinaciones posibles de las N señales de entradas, se enumera el valor binario de cada una de las m señales de salida.
- **Ecuación Booleana:** Cada señal de salida se representa como una función booleana de las señales de entrada.

	NOT	AND	OR																																				
Gráfica																																							
Tabla	<table border="1"> <tr> <th>A</th> <th>X</th> </tr> <tr> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> </tr> </table>	A	X	0	1	1	0	<table border="1"> <tr> <th>A</th> <th>B</th> <th>X</th> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </table>	A	B	X	0	0	0	0	1	0	1	0	0	1	1	1	<table border="1"> <tr> <th>A</th> <th>B</th> <th>X</th> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </table>	A	B	X	0	0	0	0	1	1	1	0	1	1	1	1
A	X																																						
0	1																																						
1	0																																						
A	B	X																																					
0	0	0																																					
0	1	0																																					
1	0	0																																					
1	1	1																																					
A	B	X																																					
0	0	0																																					
0	1	1																																					
1	0	1																																					
1	1	1																																					
Ecuación	$\neg A$	$A * B$	$A + B$																																				

¿Qué elementos debe poseer una instrucción?

Elementos de una instrucción de máquina:

- ❖ 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 uno ó más 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.
- ❖ Los operandos fuente y resultado pueden estar en tres lugares :
 - Memoria
 - Registro de la CPU
 - Dispositivo de E/S

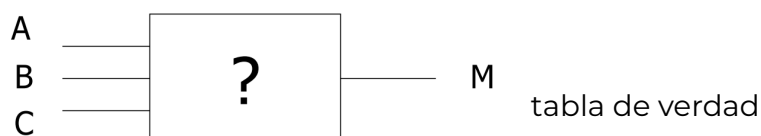
Describe el método de diseño de circuitos lógicos combinacionales denominado "Suma de productos".

- ❖ Hay tantos términos como 1s en la tabla
- ❖ Cada término vale 1 para una única combinación de A, B y C
- ❖ Las variables que valen 0 en la tabla aparecen aquí negadas

Por ej. Si tuvieramos el siguiente enunciado:

Ejemplo: construir la tabla de verdad e implementar el circuito de una función booleana M, de tres entradas A, B y C, tal que M=1 cuando la cantidad de '1' en A, B y C es 2 y M=0 en otro caso.

Obtenemos la siguiente



A	B	C	M
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

Por lo tanto, M tendrá tantos términos como "1" tiene la tabla de verdad, es decir cuatro términos, en cada término pondremos negada la entrada cuando esta sea "0".

$$M = (\neg ABC) + (A\neg BC) + (AB\neg C) + (ABC)$$

Después explicar el Teorema de Morgan, y con puertas Nand hacer Or, And y Not.

De Morgan propone:

La negación del producto es igual a la suma de los negados. La negación de la suma es igual al producto de los negados.

De Morgan	$\overline{A \cdot B} = \overline{A} + \overline{B}$	$\overline{A + B} = \overline{A} \cdot \overline{B}$
-----------	--	--

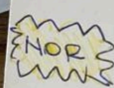
Primera ley de De Morgan: $\neg(A \wedge B) = \neg A \vee \neg B$

A	B	$A \wedge B$	$\neg(A \wedge B)$	$\neg A$	$\neg B$	$\neg A \vee \neg B$
0	0	0	1	1	1	1
0	1	0	1	1	0	1
1	0	0	1	0	1	1
1	1	1	0	0	0	0

Segunda ley de De Morgan: $\neg(A \vee B) = \neg A \wedge \neg B$

A	B	$A \vee B$	$\neg(A \vee B)$	$\neg A$	$\neg B$	$\neg A \wedge \neg B$
0	0	0	1	1	1	1
0	1	1	0	1	0	0
1	0	1	0	0	1	0
1	1	1	0	0	0	0

Convertir una compuerta NOT a un circuito solo compuesto por compuertas NAND o NOR.



CAMBIO CON
COMPUERTAS NAND

Hand-drawn logic diagram showing the implementation of the Boolean expression $A \cdot B$ using NAND gates. Input A is connected to a NAND gate with input B (labeled "NAND"). The output of this gate is $A \cdot B$. This output is then connected to a second NAND gate with input B (labeled "NAND"). The final output is $A \cdot B$.



A circuit diagram illustrating De Morgan's Law. It consists of a NAND gate with inputs A and B. The output of the NAND gate is connected to an inverter (represented by a small circle). The output of the inverter is then connected to an AND gate. The output of the AND gate is labeled $\overline{\overline{A \cdot B}}$.

CAMBIO CON COMPUERTAS NOR

Los circuitos combinatoriales se construyen más a menudo con compuertas NAND o NOR, esto es así porque cualquier función lógica puede expresarse usando solo estas puertas, además son las más sencillas de construir.



♦ NOT, AND Y OR CON COMPUERTAS NOR:

Describe las características principales de los elementos componentes que constituyen el modelo de Von Neumann

Modelo de Von Neumann: consta de 5 componentes principales:

1. Unidad de entrada: provee las instrucciones y los datos
2. Unidad de memoria: donde se almacenan datos e instrucciones
3. Unidad aritmético-lógica: procesa los datos
4. Unidad de control: dirige la operación
5. Unidad de salida: se envían los resultados

Describe características de un disco rígido. Menciona medio de almacenamiento, tiempo de acceso, formato y capacidad, etc.

¿Por qué la organización 2 ½ D de memoria semiconductora requiere "refresco"?

Al ser una memoria semiconductora, está hecha con celdas que almacenan datos como cargas eléctricas en capacitores.

El motivo por el que requiere refresco, es que los capacitores tienden a descargarse, por lo tanto se requieren de refresco periódicos para mantener memorizados los datos. Incluso si se encuentra alimentada.

Diferencia entre Dram y Sram

DRAM almacena más información que SRAM en la misma superficie.

- **DRAM** requiere refresco (basada en transistores): memoria dinámica, cargas almacenadas en transistores (capacitores), esta memoria es más lenta, usada como memoria principal.
- **SRAM** no requiere refresco (basada en flip-flops): memoria estática, más rápida, usada como memoria caché.

(2D : estática, 2 ½ D: dinámica)

¿Qué es el Ciclo de instrucción?

Es el procesamiento requerido para una sola instrucción.

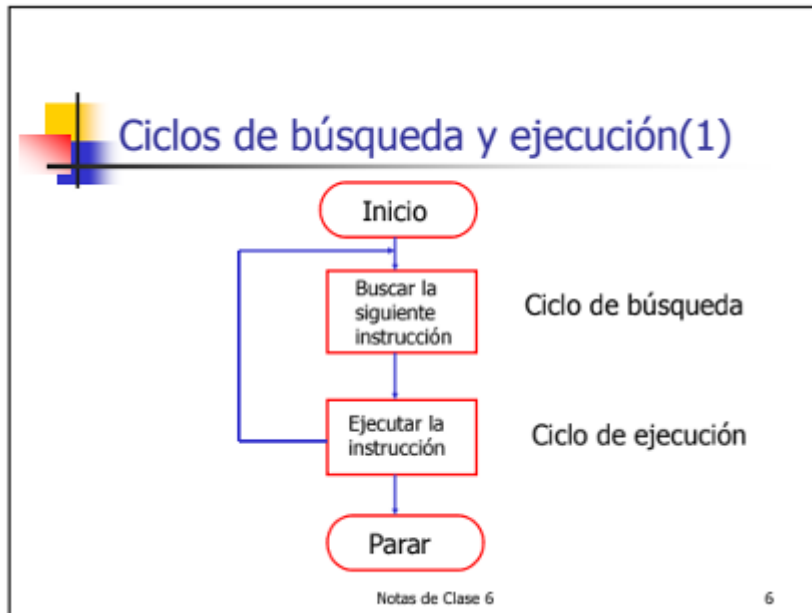
Se divide en dos etapas:

- **Búsqueda**: leer desde memoria, común a todas las instrucciones.
- **Ejecución**: ejecutar la instrucción, dependiendo de la instrucción puede implicar varias operaciones.

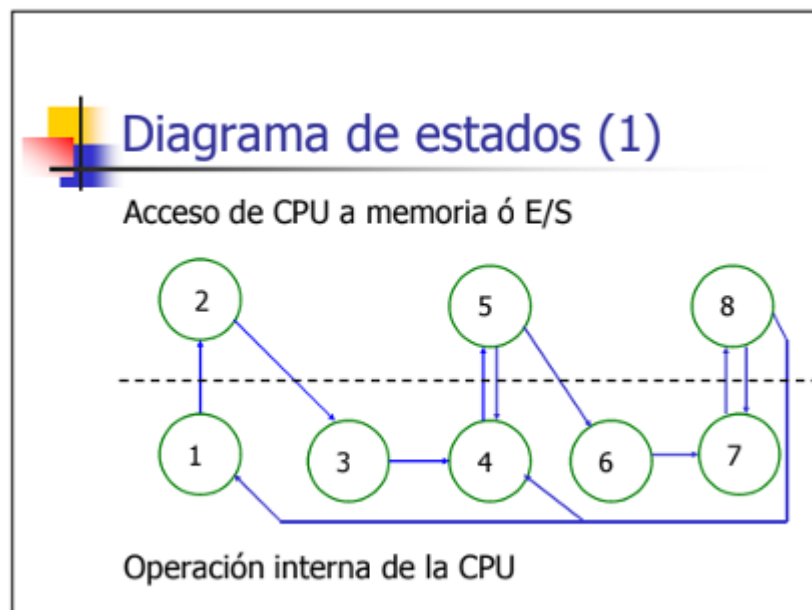
La CPU busca una instrucción en memoria, el PC (registro de la CPU) tiene la dirección de la próxima instrucción a buscar. Después de buscar cada instrucción se incrementa el PC.

La instrucción buscada se carga en un registro de la CPU llamado registro de instrucción (IR), se interpreta cada instrucción (en binario) y se llevan a cabo las acciones requeridas.

Dos pasos:



La ejecución del programa se interrumpe sólo si la máquina se apaga, hay un error o una instrucción que interrumpa a la computadora.



1. **Cálculo dirección instrucción:** determina la dirección de la siguiente instrucción a ejecutarse
2. **Búsqueda de instrucción:** lee la instrucción de su posición de memoria a la cpu. falta aclarar lo del mar y mbr
3. **Decodificación de la instrucción:** analiza la instrucción para determinar el tipo de operación a realizar y los operandos que se usarán.
4. **Cálculo dirección operando:** si la operación implica la referencia a un operando en la memoria ó e/s, entonces se determina la dirección.
5. **Búsqueda del operando:** busca el operando en la memoria ó e/s.
6. **Operación sobre los datos:** ejecuta la instrucción.
7. **Cálculo dirección resultado.**
8. **Almacenamiento resultado.**

¿Cómo está compuesta una "jerarquía de memoria"?

En la cima de la jerarquía están los **registros**, en la base, las **memorias secundarias** (discos magnéticos) y de almacenamiento "offline" (CD, DVD, Cintas)



A medida que ascendemos tenemos mayor rendimiento y más costo por bit. Entre la memoria principal y la secundaria hay otro tipo de memoria para salvar la brecha. Cuando ascendemos, también aumenta la frecuencia de accesos a ese tipo de memoria.

Es una organización de forma piramidal por niveles

1. **Registros:** Es un tipo de memoria de acceso rápido ubicado cerca del microprocesador
2. **Memoria Caché:** Es un tipo de memoria de acceso rápido pero de poca capacidad de almacenamiento.

Es la memoria de acceso rápido de una computadora, guarda los elementos recientemente usados (temporalmente)

La memoria caché es un buffer especial, funciona de manera igual que la memoria principal, pero con acceso directo y menor tamaño

3. **Memoria RAM:** Tiene mayor espacio pero es de menor velocidad
4. **Disco duro:** Tiene mayor espacio pero es de menor velocidad

¿Qué son los RAID?

Es una técnica que permite usar múltiples discos en forma conjunta vistos por el sistema operativo como un solo disco con el fin de construir un sistema de archivos más grande, rápido y confiable. Tiene como ventajas un incremento de la performance, mayor capacidad que una capacidad individual y un aumento de la confiabilidad.

Para los filesystems es un arreglo lineal de bloques que pueden ser leídos y escritos. Internamente debe calcular qué discos debe acceder para completar la solicitud. La cantidad de accesos físicos depende del nivel de RAID que se está utilizando. Están diseñados para detectar y recuperarse de determinados fallos de discos.

Niveles de raid: El cuadrito uwu



Nivel	Cant discos	Redundancia	Fallas de disco	Capacidad	Funcionamiento/datitos
0	2+	there is none. Tiene striping ¹ a nivel de bloque	si falla uno queda inaccesible	cantDiscos (siempre la menor)	dividis los datos en bloques y los esparcis por todos los discos, mejora la performance
1	2+	mediante el mirroring de datos NO TIENE STRIPING DE DATOS	si te falla un disco tenes el/los otro/s	cantDiscos/2	la mítica copia de seguridad, pero con el disco entero
4	3+	se agrega un disco para almacenar paridad stripping a nivel de bloque	tenemos 1 solo disco que almacene la paridad, a la hora de la escritura es como cuello de botella	cantDiscos (siempre la menor)	Hay un solo disco de paridad entre los discos del array, mínimo tiene 3
5	3+	striping a nivel de bloque y paridad distribuida	no ofrece solución al fallo simultáneo de discos	cantDiscos -1 destinado a la paridad	distribuye la información de paridad entre los discos del array
6	4+	striping a nivel de bloque y doble paridad distribuida	alta tolerancia a fallos (hasta 2 discos)	Cant discos -2 destinados a la doble paridad	operaciones de escritura más lentas debido al cálculo de la doble paridad recomendado cuando se tienen varios discos
0-1	4+	dos esquemas de nivel 0 espejados	desperdicio en el mirroring	cantDiscos (siempre la menor)	aumenta la confiabilidad

Por ejemplo, si tengo 5 discos de 1TB cada uno y uso un raid 5 me queda de espacio real 4tb ya que 1tb (osea un disco) es usado para la paridad. Si fuese raid 0 serian 5tb, si fuese raid 1 serian 2.5tb y si fuese raid 6 serian 3tb.

Los distintos niveles de raid son combinables

Algunas de estas respuestas las tengo en el cuaderno, debo pasarlas

Describe el funcionamiento de las instrucciones de movimiento de datos

¹ *****El striping de discos (denominado a veces RAID 0) es un método de incrementar el índice de transmisión del sistema (throughput) mediante el uso de varias unidades de disco en paralelo. En los discos donde no se utiliza striping el sistema operativo escribe en un solo bloque de un solo disco, mientras que en la configuración en striping, cada bloque se divide y las distintas partes de los datos se escriben en discos diferentes.

Son instrucciones donde el objetivo es mover el dato y **no** operar con él.

MOV dest/fuente: copia fuente en destino

PUSH fuente: carga fuente en tope de la pila

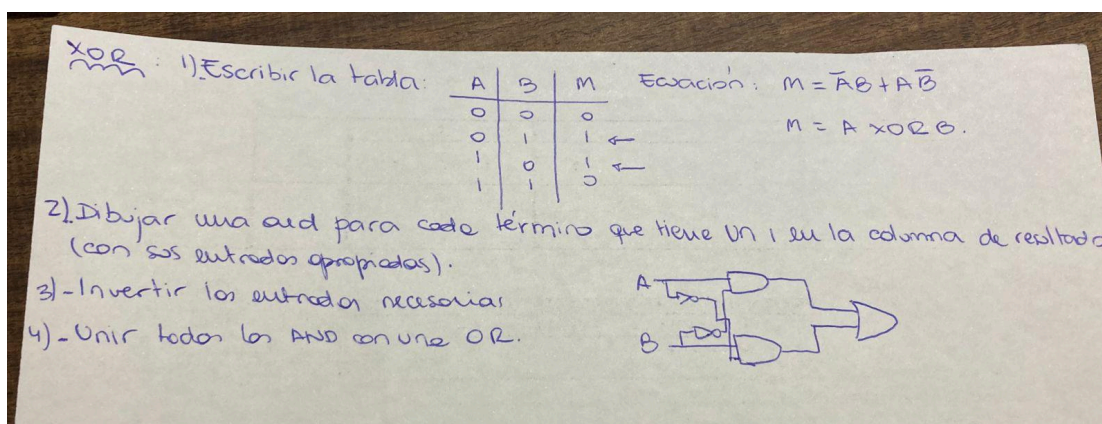
POP destino: desapila el tope de la pila y lo carga en dest

¿Qué modos de direccionamiento pueden ser utilizados en ellos? Todas

- Describa características de IEEE 754 simple precisión

- Mantisa: fraccionaria normalizada, con la coma después del primer bit que es siempre uno (1,) en M y S.
- Exponente: representado en exceso: $2n-1 - 1$
- Simple precisión: 1 bit de signo, 8 bits para el exponente, 23 bits para la mantisa

- Describa suma de productos e implemente la función XOR utilizando el mismo



¿Cuáles son los principios que permiten el funcionamiento de un sistema de memoria basado en jerarquía?

El uso de una jerarquía de memoria se sustenta de dos principios que exhiben **LOS PROGRAMAS**

1. **Principio de localidad espacial de referencia** cuando se accede a una palabra de memoria, es 'muy probable' que el próximo acceso sea en la vecindad de la palabra anterior.

2. **Principio de localidad temporal de referencia** cuando se accede a una posición de memoria, es 'muy probable' que un lapso de 'tiempo corto', dicha posición de memoria sea accedida nuevamente.

¿Qué se representa con el formato de instrucción?

qué hacer con qué, los datos y dónde almacenar los resultados

Digamos toda la información que necesita la UC para saber lo que tiene que hacer.

Describe y realice un esquema de org 2D y una capacidad de almacenamiento de 1024 palabras de 16 bits.

Organización 2D (2)

16 bits

B bits

Ej.

$$2^{10} = 1024$$

W líneas de dirección

Decodificador de renglón 1 de 2^W

2^W palabras

Chip de memoria

Lectura/Escritura

Bus de datos

Notas de clase 9

26

Por qué la organización 2 ½ D de memoria semiconductora utiliza 2 decodificadores?

Porque la dirección se divide en dos partes: una selección de renglón y una selección de columna

Describe 3 características fundamentales de un chip de memoria semiconductora 2 ½ D

2D:

- El arreglo está organizado en 2^W palabras de B bits cada una. Cada línea horizontal (una de 2^W) se conecta a cada posición de memoria, seleccionando un renglón.
- Las líneas verticales conectan cada bit a la salida.
- El decodificador que está en el chip, tiene 2^W salidas para W entradas (bits del bus de direcciones)

2 ½ D:

- El arreglo es 'cuadrado' y funciona igual que 2D.
- Los bits de una misma palabra están dispersos en distintos chips.
- La dirección se divide en dos partes: una selección de renglón y una selección de columna. Hay 2 decodificadores.

Comparación

- En 2D todos los bits están en el mismo chip. En 2 ½ D los bits de una misma palabra estarán en distintos chips.

- 2D es muy larga y estrecha, N° grande de palabras de pocos bits. Cada línea de selección de palabra tiene que tener un manejador y conectarse al decodificador.
- Ocupan mucha superficie.
- 2D dificulta el uso eficaz de los circuitos correctores de error. En $2\frac{1}{2}$ D al estar los bits dispersos en distintos chips hay menor probabilidad de error.
- En $2\frac{1}{2}$ D al usar decodificación separada de filas y columnas, reduce la complejidad de los decodificadores.

Describe características de un disco rígido. Menciona medio de almacenamiento, tiempo de acceso, formato y capacidad, etc.

Describe la diferencia en los pasos del ciclo de instrucción de una instrucción SUB y una CMP

Hacen lo mismo pero CMP no almacena el resultado.

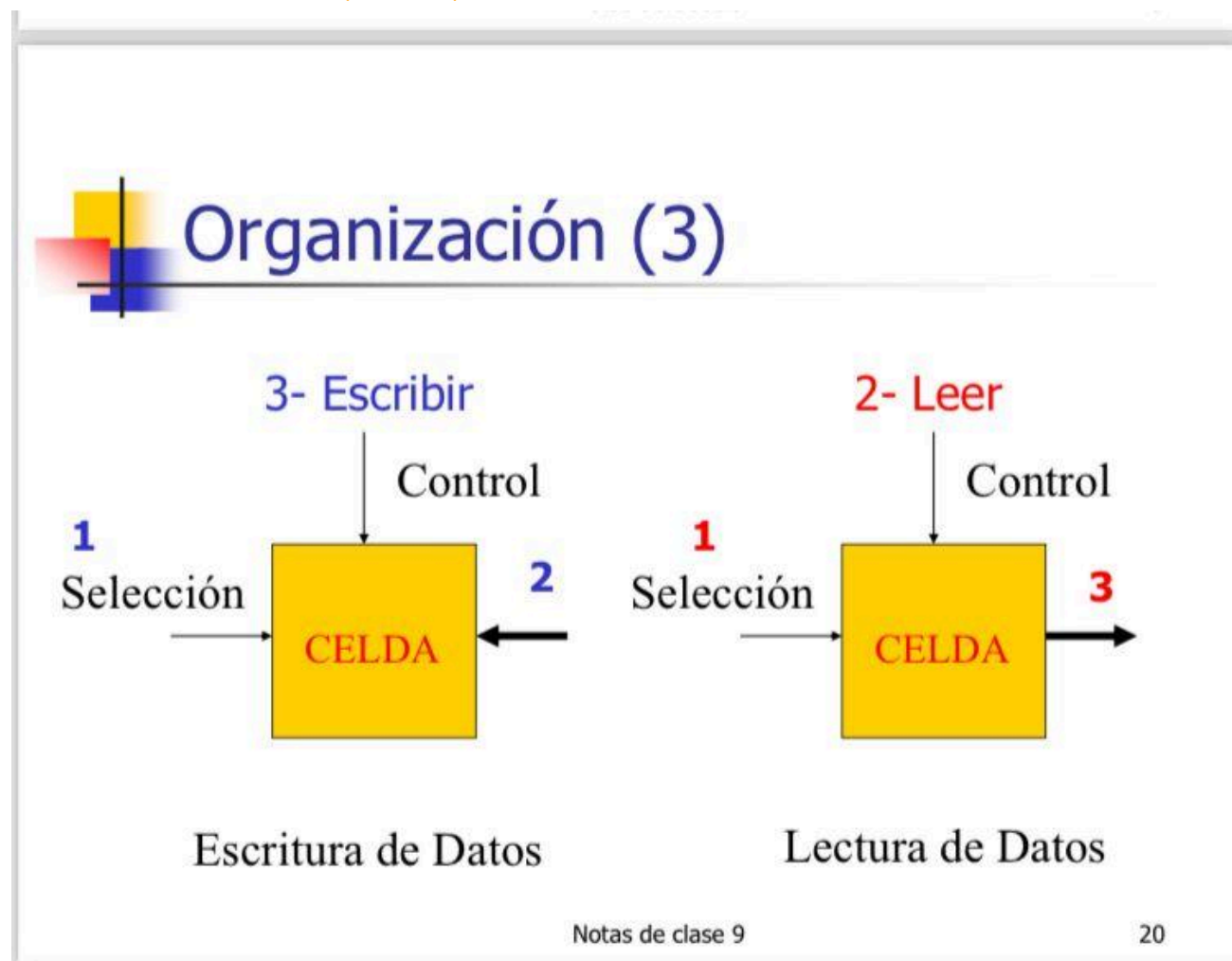
Explica el método de direccionamiento indirecto por registro con desplazamiento

- ❖ La instrucción tiene dos campos de dirección que se suman para dar la dirección efectiva.
- ❖ Se especifica el registro que tiene adentro una dirección [BX]
- ❖ Se accede una vez menos a memoria
- ❖ Se usan menos bits

Dar ejemplo de instrucciones y de sus variantes

- ❖ MOV AX, [BX + 2]
- ❖ MOV [BX + 2Ah], AX

¿Qué función debe cumplir un "punto de memoria"?



Cada celda tiene 3 terminales funcionales principales, seleccionar una celda de memoria
 Controlar si es de escritura o lectura
 Escribir/Leer datos

¿Qué es sumador completo de 1 bit?

Sumador completo de 1 bit

A ↓ B ↓ C ↓

Sumador completo

↓ ↓

C_0 S

A	B	C	C_0	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

$C_0 = \bar{A}BC + A\bar{B}C + A\bar{B}\bar{C} + ABC$

$S = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$

Describe los pasos del ciclo de instrucción de un salto condicional:

1. Buscar la instrucción
2. Decodificarla
3. El valor que figura en el campo se debe cargar en el contador de programa.

Describe los pasos del ciclo de instrucción de una operación aritmética:

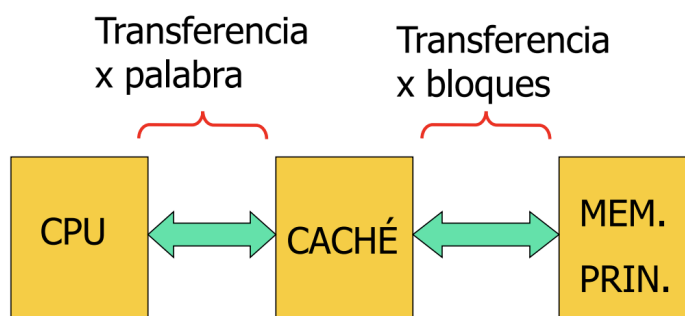
1. Buscar la instrucción en memoria.
 2. Incrementar el PC.
 3. Decodificar la instrucción.
 4. Si es necesario, buscar una constante en una dirección de memoria.
 5. Si es necesario, incrementar el PC para que apunte más allá de la constante.
 6. Si es necesario calcular la dirección del operando.
 7. Buscar uno de los operandos, desde memoria ó registro.
 8. Buscar el otro operando desde registro.
 9. Realizar la operación.
 10. Almacenar el resultado.
-

¿Qué es y para qué sirve la memoria Caché?

La memoria caché actúa como un búfer entre la CPU y la memoria principal. La idea general es que cuando se hace referencia a una palabra, ella y alguna de las vecinas se traen de la memoria grande y lenta a la caché, para que en el siguiente acceso la palabra buscada se encuentre en el caché.



• Ej.



Ejercicios de práctica de Finales

- Representar el número 7 en el método del inciso a (Describe características de IEEE 754 simple precisión) - (fecha:10/04/2023)
0 10000001 11000...(recordar: 1 bit de signo, 8 de exponente, 23 de mantisa)
- Usar suma de productos para representar una función de cuatro entradas (A,B,C,D) cuando dos de las entradas sea 0 (tabla, ecuación, gráfico)

Suma de productos para representar una f(x) de 4 entradas (A,B,C,D) cuando dos de las entradas sean 0 (tabla, ecuación, gráfico).

• Tabla $2^4 = 16$

A	B	C	D	m
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
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

• Ecuación

$$M = \bar{A}\bar{B}C\bar{D} + \bar{A}B\bar{C}\bar{D} + \bar{A}B\bar{C}D + \bar{A}B\bar{C}D + \bar{A}B\bar{C}D + \bar{A}B\bar{C}D$$

• Gráfico

- Dado un sistema de punto flotante con 6 bits de mantisa fraccionaria en BCS y exponente de 4 bits en exceso 8 (en ese orden de izq a derecha) - (fecha:08-22)

- ¿Cuál es el rango de representación del sistema?
- $[-124; 124]$
- Represente en punto flotante el número decimal 322
- No puede representarse ya que no pertenece al rango para este sistema

Una imagen en una pantalla de 100 cm por 50 cm posee una resolución de 100 puntos por centímetro:

- ¿Cuántos bytes de memoria se requieren para almacenar una imagen en true color?

$$100 \times 100 \times 50 \times 100 = 50 \times 10^6 \text{ (Total)}$$

(3 bytes por true color)

$$50.000.000 \times 3 = 150.000.000 \text{ (Total real)}$$

$$150.000.000 / 8 = 18.750.000 \text{ Bytes}$$

b) ¿Cuántas imágenes podría almacenar en esa memoria si la imagen a almacenar fuera "monocromática"?

24 si voy a multiplicar 24×24 , es lo mismo que multiplicar 24×1 , entonces cuántas imágenes necesito? 24

-Dado un sistema en punto flotante con 5 bits de mantisa en BCS con bit implícito y exponente de 5 bits en exceso 16 (en ese orden de izq a derecha) (fecha: 11-22)

a) ¿Cuál es el valor decimal mínimo positivo representable por el sistema?

b) 0 000

c) Represente en el sistema de punto flotante dado el número decimal 2,625
10,101 o sea 00101 10010

- Dado un sistema en punto flotante con 7 bits de mantisa en BCS y bit implícito y 8 bits de exponente en exceso 128 (en ese orden de izq a derecha) - (fecha: 21-02-22)

- Representar tu número de alumno (sin el nro después de la barra)

- 17208: 100001100111000

entonces me quedo con 1000011 que en este sistema sería 0000011

- el exponente es 10001111 ($15+128$)

- Calcular el error absoluto: $16768 = 440$

-Dado el número A118 en BCH donde 8 bits representan la mantisa en BCS con bit implícito y 8 bits para exponente en exceso 128 (fecha: 02-22 - Primera fecha)

a) Calcular el valor en decimal

Exponente:

Mantisa: $-(2 \text{ a la } -1 + 2 \text{ a la } -3 + 2 \text{ a la } -8) \times 2 \text{ a la } -104$ es decir $-(2 \text{ a la } -105 + 2 \text{ a la } -107 + 2 \text{ a la } -112)$

b) Calcular el error absoluto máximo

-Dado el número 41B6H en punto flotante con 8 bits de mantisa en BCS y bit implícito y 8 bits de exponente en exceso 128 (en ese orden de izq a derecha)(final: SF (sin fecha))

a) Calcular el valor en decimal 0100 0001 1011 0110, por lo tanto de exponente son $182-128 = 54$, entonces la mantisa para este sistema es : 1100 0001 por lo tanto quedaría $(2^{-1} + 2^{-2} + 2^{-8}) \times 2^{54} = 2^{53} + 2^{52} + 2^{46}$, lo calcula tu vieja, yo lo dejo expresado

b) Calcular el error absoluto máximo

-Un video musical monocromático de YouTube dura 200 segundos (corregido)

a) Calcule cuántos bytes de memoria se necesitan para almacenar el video completo si cada imagen es de **800 x 500 px**

b) ¿Cuál debería ser la velocidad de transferencia (en bytes/segundos) si la imagen debe ser cambiada **25 veces en 1 segundo**?

memoria de una imagen = $800 \times 500 \times 1 \text{ bit}$

video completo (25 imágenes por segundo en 200 segundos) =

$(800 \times 500 \times 25 \times 200) \text{ bits}$

en bytes = $(800 \times 500 \times 25 \times 200) / 8$

velocidad de transferencia: si una imagen es $(800 \times 500 \times 1 \text{ bit})$ y cambia 25 veces por segundo, entonces:

$800 \times 500 \times 1 \text{ bit} \times 25 \times (1 \text{ bit} / \text{s})$ la unidad es [bits/segundo]

-Dado un sistema de punto flotante con 6 bits de mantisa fraccionaria en BCS y exponente de 4 bits en exceso 8 (en ese orden de izq a derecha) - (fecha:08-23)

c) ¿Cuál es el rango de representación del sistema?

- [-124;124]

d) Represente en punto flotante el número decimal 32,2 decime cómo representar ese ,2 masterrrrr

-Un video monocromático de Tik Tok dura 15 segundos y cada imagen es de 500 x 800 px - (fecha:23-02-1)

e) ¿Cuántos bytes de memoria se necesitan para almacenar una imagen del video? supongamos escala de grises, entonces serían 8 bits (1 byte)

Respuesta: $500 \times 800 \times (1 \text{ byte de la escala de grises}) = 40 \times 10^4$

f) ¿Cuál debería ser la velocidad de transferencia a periférico (en bytes/segundos) si para 'percibir movimiento' cada imagen debe ser cambiada 25 veces en 1 segundo?

Respuesta: la velocidad de transferencia $1/((40 \times 10^4) \times 25)$ byte/seg

-Representar el valor hexadecimal 62000000H (fecha:SF2)

0110 0010 0000 0000 0000 0000 0000 0000

FINAL 04 DE DICIEMBRE 2023

1)a)Mantisa de 6 bits BCS con bit implícito

Exponente 5 bits Exceso a 16 puta :)

Representar (leg-20000)

b)Calcular EA(X) y Error Relativo

Siendo X': nro representado y X: nro a representar

Error Absoluto EA(X): $|X' - X|$

Error Relativo ER(X): $EA(X)/X$

2) a-Describa cómo puede construir un sumador binario completo de (1) Bit

b-definir cuáles funciones lógicas utilizará en él y su tabla de verdad

c-Realice un esquema de interconexión de compuertas de dichos sumador

3) a-Enumerar elementos a tener en cuenta para diseñar un conjunto de instrucciones

✓ ¿Instrucciones cortas o largas?

✓ Nro de bits/seg

✓ ancho de banda de la memoria

✓ Velocidad procesador/Velocidad memoria

✓ Instrucciones más cortas

✓ el procesador "parece" más rápido

✓ Suficientes bits para expresar todas las operaciones deseadas.

✓ La experiencia demuestra dejar bits libres para el futuro.

✓ Cantidad de bits de datos.

ESTAS CON TIC NO, LAS DEL COMENTARIO

según Runco son 5:

1) cuántas instrucciones necesito (nos da el tamaño)

2) tipos de datos a usar (por ejemplo ASCII)

3) cantidad de registros y de qué tipos son (ocultos al programador, visibles)

- 4) modos de direccionamiento
- 5) formato de instrucción (direcciones)

b-Qué es el ciclo de instrucción

El procesamiento requerido para una sola instrucción se llama ciclo de instrucción.

4) Completada ADD AX,mem

a-Con qué tipo de instrucciones continuará un programa para determinar el resultado obtenido es correcto o no

con una instrucción de transferencia de control (jump) que salte en caso de producirse V

b-Escriba la secuencia de instrucciones considerando los casos de representación de números en signos y en Ca2

en BSS si hubo Carry JC O JNC

y en Ca2 si hubo overflow O JO O JNO

5) a-Describir estructura interna y los elementos que componen una memoria semiconductora con organización 2D

(está respondido más arriba)

b-Cómo se calcula el tiempo de acceso a un disco magnético

Tiempo de Acceso: $T.seek + T.latencia$, siendo Tiempo de seek (búsqueda)

Mover al cilindro (o pista) correcto, Tiempo de latencia (por rotación)

Esperar que el sector "pase" por debajo de la cabeza

Nunca vi que lo pidan pero por las dudas:

Tipos de instrucciones

Podemos categorizar las instrucciones de máquina como de:

- ❖ Procesamiento de datos:
 - operaciones aritméticas y lógicas.
- ❖ Almacenamiento de datos
 - transferencias dentro del sistema.
- ❖ Instrucciones de E/S:
 - transferencia de datos entre la computadora y los mecanismos externos.
- ❖ Control

Más cosas que no vi que pidan pero suenan importantes: 🙄

1) Jerarquía de memoria: duración de la información.

- a) Memorias volátiles: RAM
- b) Memorias no volátiles: discos y cintas.
- c) Memorias permanentes: ROM, EPROM.

2) Modos de acceso a memoria:

- a) Acceso por palabra: memoria principal.
- b) Acceso por bloque: discos, caché.

3) Memorias semiconductoras:

- a) Tiempo de acceso: tiempo máximo que transcurre desde que se inicia la operación de lectura/escritura hasta obtener/almacenar el dato.
- b) Tiempo de ciclo: tiempo mínimo que tiene que haber entre dos operaciones sucesivas sobre la memoria.

Tiempo ciclo > tiempo acceso

4) Métodos de acceso:

- a) Acceso aleatorio: memoria principal. (RAM: random access memory)
- b) Acceso secuencial: unidades de cinta.
- c) Acceso directo: discos magnéticos.
- d) Acceso asociativo: memoria caché.

5) Memorias magnéticas:

- a) Tiempo de acceso: tiempo de posicionar el cabezal + tiempo de latencia
= **T seek + T latencia**
- b) Velocidad de transferencia: [bytes/segundo]

Tipos de memoria	Tiempo de acceso	Tamaño típico
Registros	1 ns	1 KB
Caché	5-20 ns	1 MB
Mem. Principal	60-80 ns	1 GB
Discos	10 ms	160 GB