

# Preguntas teoría.pdf



**Juanrajarote**



**Arquitectura de Computadores**



**2º Grado en Ingeniería Informática**



**Escuela Politécnica Superior de Córdoba  
Universidad de Córdoba**



[Accede al documento original](#)



Escuela de  
Organización  
Industrial

Contigo que evolucionas.  
Contigo que lideras. Contigo que transformas.

**Esto es EOI.  
Mismo propósito,  
nueva energía.**



Descubre más aquí



**EOI** Escuela de  
Organización  
Industrial

Importante

Puedo eliminar la publi de este documento con 1 coin

¿Cómo consigo coins?

Plan Turbo: barato

Planes pro: más coins

pierdo espacio



Necesito  
concentración

ali ali ooh  
esto con 1 coin me  
lo quito yo...

WUOLAH

Juan Ramón Palomino Jiménez

### Preguntas teoría tipo

**Microprogramación horizontal y vertical. Ventajas e inconvenientes?** Horizontal: formato no codificado, microinstrucciones muy largas, alto grado de paralelismo, alto consumo de memoria de control, más rápidos. Muy flexible y fácil de diseñar/analizar. Vertical: formato codificado, microinstrucciones cortas, bajo paralelismo, menor consumo de la memoria de control, menos rápidas. Más lenta y costosa en circuitería.

**Explicar cómo se soluciona el problema de sobreflujo de división en el algoritmo correspondiente para números en representación de punto flotante. Señala en el diagrama dónde se encuentra esta corrección.** En el algoritmo de división de números en representación coma flotante no existe el sobreflujo de división puesto que hay un método para evitar que ocurra. El método es comprobar si los "N" bits más significativos del dividendo representan un número mayor que los "N" bits del divisor. -Si no es mayor, se procede al algoritmo de división. -Si es mayor, la solución consiste en realizar un desplazamiento a la derecha sobre el dividendo, con su correspondiente incremento de exponente. Como son números normalizados, después de este desplazamiento los "N" bits más significativos del dividendo representan un número menor que los "N" bits del divisor, con lo que no hay sobreflujo de divisor.

**Supón q en CM se modifica el OPR pasando a 5 b. a) Máx. número d instrucciones en el repertorio de la máquina.** Sería según los bits que tenga el registro OPR, en este caso son 5 bits, por lo que el número máximo de instrucciones es 32 (2 elevado a 5). **b) Tamaño de la MP expresado en número de palabras por ancho de palabra.** (2 elevado a 7) \* 12 bits.

**Se tienen 2 registros, A y B hay que compararlos para saber si  $A > B$ ,  $B < A$ ,  $B = A$ . Describe cómo se puede determinar estos tres casos si:** a) **Se usa un restador completo:** -- $A > B$ : Último préstamo=0 -> por lo tanto el resultado es  $A - B$ . -- $A = B$ : Último préstamo=0 -> Por lo tanto el resultado es 0. -- $A < B$ : Último préstamo=1 -> Por lo tanto el resultado es:  $C2 (B - A)$ . b) **Se posee un sumador completo con complemento (es decir, permite sumar 1 de los 2 registros en  $C2$ ):** -- $A > B$ : Último préstamo=1 -> Por lo tanto el resultado es  $A - B$ . -- $A = B$ : Último préstamo=1 -> Por lo tanto el resultado es 0. -- $A < B$ : Último préstamo=0 -> Por lo tanto el resultado es  $C2 (B - A)$ .

**Significado LCB, I, B, R, E.** LCB: Lógica de Control de Bifurcación. Controla los movimientos dentro de la ejecución del programa. Incremento (I): Pasa a ejecutar la microinstrucción que se encuentra en la dirección CROM consecutiva. Bifurcación (B): Pasa a ejecutar la microinstrucción que se encuentra en una dirección CROM específica. Carga de rutina (R): Pasa a ejecutar la primera microinstrucción asociada al ciclo de ejecución de una instrucción determinada. Enable (E): Bloque para habilitar o inhabilitar la ejecución de microoperaciones.

WUOLAH

**Punto flotante. Te dan algoritmos de división y multiplicación, y pregunta por qué en uno se resta el sesgo y en el otro se le suma:** En el caso de la multiplicación, el exponente del resultado será la suma de los exponentes de los operandos. Para esto hay que tener una consideración, y es que cada uno de los exponentes de los operandos está sesgado. Para tener el exponente del producto correctamente sesgado, debe restársele el sesgo a la suma de los dos exponentes. En el caso de la división, el exponente del resultado se obtiene restándole al exponente del dividendo el exponente del divisor, hay que tener en cuenta que al restar 2 números sesgados, desaparece el sesgo. Luego para tener la representación de la división correctamente sesgado, hay que sumarle el sesgo a la resta de exponentes.

# Preguntas\_Teoria\_REVISADAS.pdf



**TEAM\_GETPPID\_\_**



**Arquitectura de Computadores**



**2º Grado en Ingeniería Informática**



**Escuela Politécnica Superior de Córdoba  
Universidad de Córdoba**



**Accede al documento original**



Escuela de  
Organización  
Industrial

Contigo que evolucionas.  
Contigo que lideras. Contigo que transformas.

**Esto es EOI.  
Mismo propósito,  
nueva energía.**



Descubre más aquí



**EOI** Escuela de  
Organización  
Industrial

Importante

Puedo eliminar la publi de este documento con 1 coin

¿Cómo consigo coins? → Plan Turbo: barato  
→ Planes pro: más coins

pierdo  
espacio



Necesito  
concentración

ali ali ooh  
esto con 1 coin me  
lo quito yo...

WUOLAH

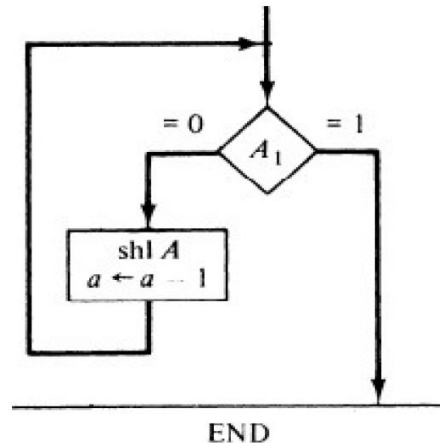
## ARQUITECTURA DE COMPUTADORES. RECOPIACIÓN PREGUNTAS TEÓRICAS

Jose Luis Gordillo relaño

WUOLAH

**1) Algoritmo Suma/Resta en punto flotante.**

**¿Qué problema soluciona la sección marcada? (1 Punto)**



Resuelve el problema de normalización del resultado. Una vez realizada la suma se debe normalizar desplazando los bits de la mantisa hacia la izquierda con lo cual habrá que cambiar el valor del exponente.

**2) Computadora de Von Neumann. (2 Puntos).**

Von Neumann presento un modelo consistente en 4 unidades:

- |                             |                      |
|-----------------------------|----------------------|
| 1.Unidad de entrada/Salida. |                      |
| 2.Unidad de Memoria         |                      |
| 3.Unidad Aritmético-Lógica  | } Constituyen la CPU |
| 4.Unidad de Control         |                      |

**Memoria Principal**

Se caracteriza por dos parámetros: Tamaño de memoria (nº direcciones) y ancho de palabra (nº bits componen cada palabra), permite operaciones de lectura y escritura. Compuesta por un conjunto de celdas agrupadas en bloques del mismo tamaño(palabra) El acceso a cada palabra tiene asociado un numero único llamado dirección.

**Unidad Aritmético-Lógica (ALU)**

Contiene micro operaciones capaces de tomar datos y realizar operaciones elementales. Incorpora registros que almacenan operandos y guardan resultados de la operación.

-Registro de propósito general, R. específicos, R. de control, R. de estado.

**Unidad de Control**

Encargada de manejar las unidades de la computadora, de forma que al trabajar coordinadamente puede realizar operaciones complejas.

Recoge las instrucciones de la memoria, las decodifica y prepara las señales para las diferentes unidades funcionales que generará en los diferentes ciclos, las señales necesarias para acceder y controlar son:

- Memoria principal (lectura/escritura)
- ALU (diversas), Banco de registros (lectura/escritura)
- E/S (diversas)

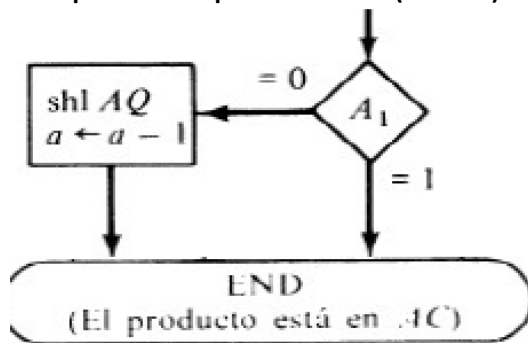
Para llevar la cuenta de la instrucción actual, incorporara un apuntador denominado Contador de Programa (PC).

**Unidad E/S**

Permite al computador obtener datos desde el exterior, mediante la unidad de E/S realiza transferencias de información con los periféricos, capaz de volcar información desde memoria secundaria (HDD, Pendrive) hasta memoria principal, también permite visualizar por pantalla, imprimir, etc..



3) Multiplicación en punto flotante. (1 Punto).



Resuelve el problema de normalización del resultado. Ya que se debe normalizar desplazando los bits de la mantisa hacia la izquierda con lo cual habrá que cambiar el valor del exponente.

4) Algoritmo suma/resta punto flotante.

- Los operandos deben tener el mismo exponente:
  - Se desplaza a la derecha el operando con el menor exponente y se incrementa su exponente, hasta que ambos exponentes coinciden.
- Una vez comprobado que los operandos tienen el mismo exponente: el exponente del resultado será el común y la mantisa del resultado se obtendrá mediante la suma/resta (respectivamente) en signo-magnitud de las mantisas de los operandos.
- Por último, se comprueba si el resultado de la suma/resta está normalizado, si no está normalizado, se debe realizar su normalización.

5) ¿Qué soluciona el algoritmo de Booth?

El algoritmo de Booth resuelve el problema dado en otros algoritmos de multiplicación de S-M, en los cuales el signo del multiplicador siempre debía ser positivo. De este modo podemos realizar el producto cuando tenemos un multiplicador con signo negativo.

6) Teniendo en cuenta la computadora mejorada a la que se ha añadido la unidad de calculo de S-M. ¿Cómo resolverías el problema del overflow de la división de números en representación S-M sin realizar cambios en la unidad de cálculo? Indica las modificaciones que realizarías en el algoritmo de la división.

En la toma de decisión E, es donde comprueba si hay overflow o no. Si lo hay finaliza la operación, sino realiza la división tal y como continua con las instrucciones.

7) Los algoritmos de multiplicación de números en complemento a 2 tienen en común un problema que no tienen los algoritmos de Booth. ¿Qué problema es?

Cuando se produce sobreflujo de suma con números en C2, se produce una inversión de signo en el resultado:

- Si se suman dos números positivos, se obtiene como resultado un número negativo.
- Si se suman dos negativos, se obtiene como resultado un positivo.

Por tanto, tras realizar la suma parcial, a la hora de realizar el desplazamiento aritmético a la derecha:

- Si no hubo sobreflujo tras la suma ( $V=0$ ), se desplazan todos los bits del signo.
- Si hubo sobreflujo tras la suma ( $V=1$ ), se desplazan todos los bits a la derecha y como el bit de signo es incorrecto, se complementa.

Importante

Puedo eliminar la publi de este documento con 1 coin

¿Cómo consigo coins? → Plan Turbo: barato  
→ Planes pro: más coins

pierdo espacio



Necesito concentración

ali ali ooh  
esto con 1 coin me  
lo quito yo...

WUOLAH

8) ¿Cómo se resuelve el problema del overflow de la división de números en representación punto flotante? ¿Qué problemas aparecen al resolver el problema del overflow?

- En punto flotante no existe el problema del sobreflujo de división. Inicialmente uno comprueba si los N bits más significativos del dividendo representan un número mayor que los N bits del divisor.
  - Si no es mayor, se procede al algoritmo de división.
  - Si es mayor la solución consiste en realizar un desplazamiento al a derecha sobre el dividendo con el incremento del exponente.
  - Como son números normalizados, los N bits más significativos del dividendo representan un número menor que los N bits del divisor, con lo que no hay sobreflujo de división.

9) Comparación de números binarios mediante operaciones aritméticas. Indicar un ejemplo de cada una de ellas.

Una forma de saber si dos números son iguales es restándolos.

- ✓ Si son iguales, el resultado será 0.
- ✓ Si el primero es mayor que el segundo, el resultado será positivo.
- ✓ Si el primero es menor que el segundo, el resultado será negativo.

Utilizando un restado completo:

- $A > B \rightarrow \text{último préstamo} = 0 \rightarrow \text{Resultado} = A - B$
- $A = B \rightarrow \text{último préstamo} = 0 \rightarrow \text{Resultado} = 0$
- $A < B \rightarrow \text{último préstamo} = 1 \rightarrow \text{Resultado} = 2^n - (B - A) = C_2(B - A)$

Por ejemplo:

$$\begin{array}{r} 2-1 \\ 0010 \\ -0001 \\ \hline 0001 \end{array} \quad \begin{array}{r} 2-2 \\ 0010 \\ -0010 \\ \hline 0000 \end{array} \quad \begin{array}{r} 1-3 \\ 0001 \\ -0011 \\ \hline 1110 \end{array} \quad \begin{array}{r} 0010 \\ 3 \quad 1101 \\ -1 \quad 1110 \\ \hline 2 \quad 1110 \end{array}$$

• La resta se puede realizar mediante la suma en complemento a 2 sustrayendo.

$$A - B = A + C_2(B) = A + C_1(B) + 1$$

• Utilizando un sumador completo:

- $A > B \Rightarrow \text{Último Acarreo} = 1 \Rightarrow \text{Resultado} = A - B$
- $A = B \Rightarrow \text{Último Acarreo} = 1 \Rightarrow \text{Resultado} = 0$
- $A < B \Rightarrow \text{Último Acarreo} = 0 \Rightarrow \text{Resultado} = 2^n - (B - A) = C_2(B - A)$

• Ej.

$$\begin{array}{r} 7-6 \\ 0111 \\ +1001 \\ +1 \\ \hline 10001 \end{array} \quad \begin{array}{r} 7-7 \\ 0111 \\ +1000 \\ +1 \\ \hline 10000 \end{array} \quad \begin{array}{r} 6-7 \\ 0110 \\ +1000 \\ +1 \\ \hline 01111 \end{array}$$

WUOLAH



## 10) ¿Para qué sirve la lógica de bifurcación?

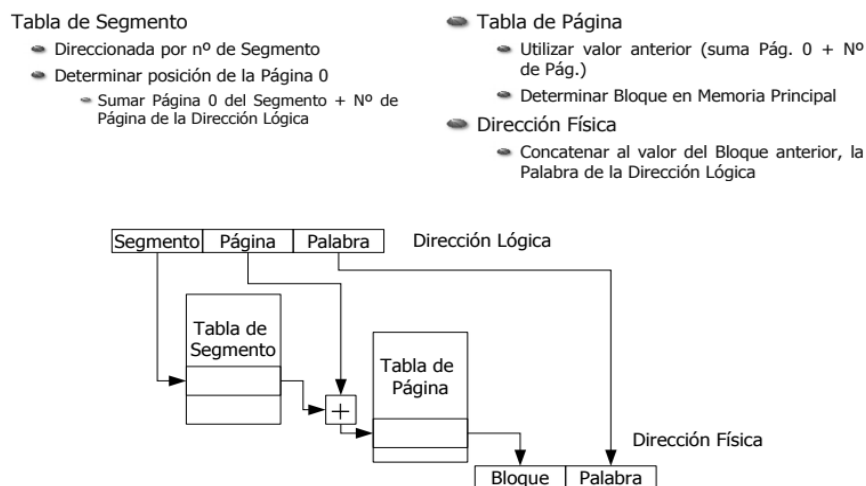
Sirve para saltar a una microinstrucción que se encuentra en una dirección CROM en particular. Dicha dirección debe serle indicada para poder realizar el salto.

Bifurcación Condicional: Dependiendo de los bits de estado, se realizará un salto o no.

## 11) ¿Qué es la segmentación?

Es una técnica de gestión de memoria, consistente en la división variable de un programa en N segmentos, asociando la longitud de cada segmento a un número variable de páginas que conformarán una dirección lógica, que puede ser mayor, igual o menor que la dirección física.

Encontramos 3 campos: **Campo de Segmento, Campo de Página y Campo de Palabra.** Su funcionamiento de traducción se basa en:



Para realizar la traducción será necesario utilizar dos memorias adicionales, que en memoria aleatoria implicará un tiempo de acceso muy lento, pero en asociativa determinaremos que par de (Segmento y Página) encontramos en la tabla, devolviendo el valor del Bloque (si está presente) y si no lanzará un algoritmo de reemplazo.




























## 12) ¿Qué es el descriptor de segmentos?

Contienen la dirección base y el límite del segmento en el espacio lógico, su estructura es la siguiente:

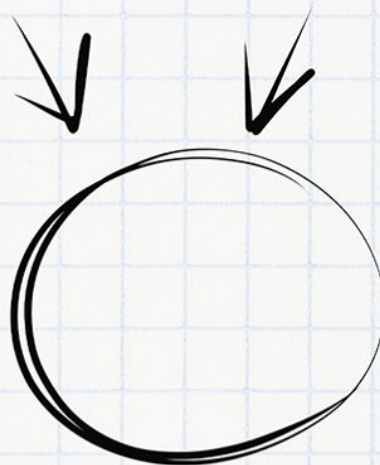
- Dirección: Dirección de la tabla de Páginas / Dirección del Bloque Base.
- Longitud: Tamaño del segmento.
- Protección: Permisos de acceso dados a dicho segmento.

# Imagínate aprobando el examen

## Necesitas tiempo y concentración

Planes	 PLAN TURBO	 PLAN PRO	 PLAN PRO+
 Descargas sin publi al mes	10 	40 	80 
 Elimina el video entre descargas			
 Descarga carpetas			
 Descarga archivos grandes			
 Visualiza apuntes online sin publi			
 Elimina toda la publi web			
 Precios <span>Anual <input type="checkbox"/></span>	0,99 € / mes	3,99 € / mes	7,99 € / mes

Ahora que puedes conseguirlo,  
¿Qué nota vas a sacar?



# WUOLAH

### 13) ¿Qué es la TLB?

**Buffer de traducción adelantada:** Es una memoria caché administrada por la Unidad de gestión de memoria, que contiene partes de la tabla de paginación, es decir, relaciones entre direcciones lógicas y físicas.

Se utiliza para obtener la traducción rápida de direcciones, ya que la CPU solicitará una página, primeramente, se buscará en la TLB, al no encontrarla la solicitará a la tabla de paginación, que, al obtener la página y bloque, la devolverá a CPU (insertándola previamente en la TLB).

De esta forma cuando la CPU vuelva a solicitar dicha página, bastará con revisar la TLB. De esta forma se acelera en muchos ciclos la traducción de direcciones lógicas a físicas.

- 14) Supongamos una computadora que incluye una memoria pila construida como parte de la memoria principal. Disponemos de un puntero pila SP de 8 bits para poder direccionar hasta 256 palabras distintas y que es inicializado a FFh. Si la posición de la pila en memoria principal comienza en la dirección 00h en orden creciente de direcciones. Definir las secuencias de micro operaciones para realizar una operación de escritura y otra de lectura en la memoria pila.

Pila Parte Inferior			
SP Contiene la dirección del último dato insertado.		SP Contiene la siguiente dirección de memoria libre	
PUSH	POP	PUSH	POP
SP-1 → SP	SP → MAR	SP → MAR	SP + 1 → SP
SP → MAR	M → GPR, SP + 1 → SP	GPR → M	SP → MAR
GPR → M		SP - 1 → SP	M → GPR
Pila en la parte Superior			
SP contiene la dirección del último dato insertado		SP Contiene la siguiente dirección de memoria libre.	
PUSH	POP	PUSH	POP
SP + 1 → SP	SP → MAR	SP → MAR	SP - 1 → SP
SP → MAR	M → GPR	GPR → M	SP → MAR
GPR → M	SP – 1 → SP	SP + 1 → SP	M → GPR

¿Cómo consigo coins?  Plan Turbo: barato  
 Planes pro: más coins

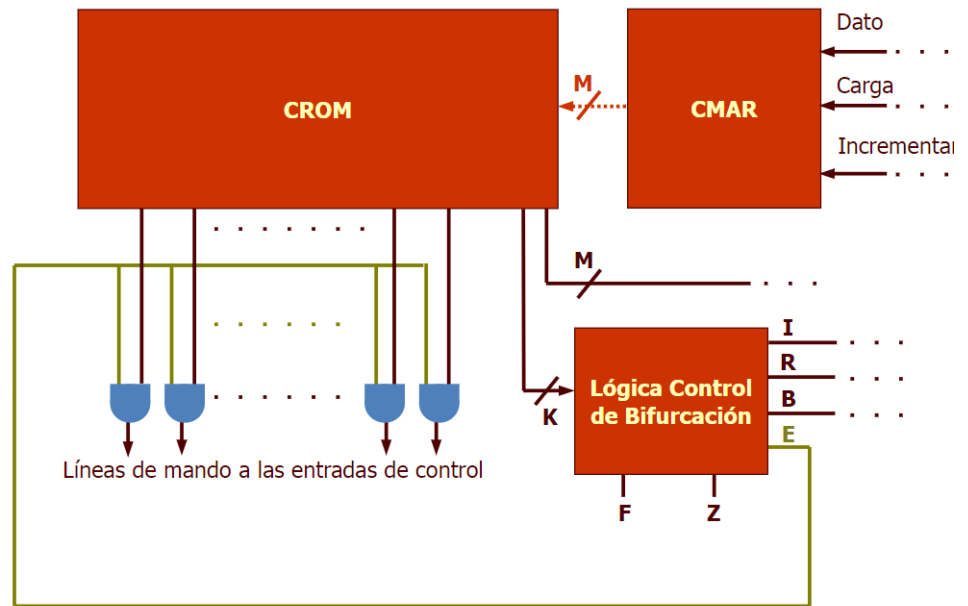
perdo  
espacio



Necesito  
concentración

ali ali ooh  
esto con l coin me  
lo qito yo...

# WUOLAH



**CMAR:** El secuenciamiento se consigue variando la dirección contenida en el registro de dirección de acceso a la CROM.

**Incremento:** Pasar al microinstrucción que se encuentra en la dirección CROM consecutiva => Incrementar el contenido de CMAR.

**Bifurcación:** Saltar a una microinstrucción que se encuentra en una dirección CROM en particular → Dicha dirección debe ser indicada (codificación de la dirección en la propia micro palabra) → Carga paralela de dicha dirección.

**Carga de rutina:** Saltar a la microinstrucción de una de las rutinas de las instrucciones  
→ La dirección de inicio de cada instrucción debe conocerse  
→ Carga paralela de dicha dirección

**CROM:** Contiene las microinstrucciones que serán ejecutadas.