

# WUOLAH



Info\_sw

[www.wuolah.com/student/Info\\_sw](http://www.wuolah.com/student/Info_sw)



790

## Resumen\_ARQ\_CMP.pdf

*Resumen\_ARQ\_CMP*



**2º Arquitectura de Computadores**



**Grado en Ingeniería Informática**



**Escuela Técnica Superior de Ingeniería  
UHU - Universidad de Huelva**

**LA ÚNICA BEBIDA ENERGÉTICA CON  
UN GRAN SABOR A COCA-COLA**

**EXPANDE TU ENERGÍA POSITIVA**



Año contenido en Cafeína. Ver envase. ©2019 The Coca-Cola Company. Todos los derechos reservados. COCA-COLA es una marca registrada de The Coca-Cola Company.

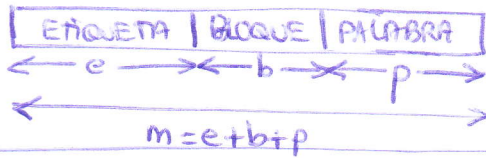
## PREGUNTAS TEST TEMA 2.

En correspondencia directa sólo necesito 1 comparador. Es la más económica.

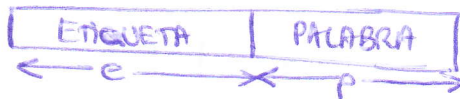
En correspondencia totalmente asociativa necesitaremos tantos comparadores como bloques tenga la H.C

En correspondencia asociativa por conjuntos representados tantos comparadores como bloques tenga un conjunto

DIRECTA:

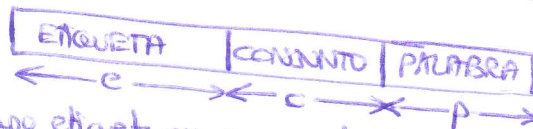


TOTALMENTE ASOCIATIVA:



Cuando todos los bloques de H.C. están llenos, habrá que utilizar un algoritmo de reemplazo.

### ASOCIATIVA POR CONJUNTOS:



Vias: n° de bloques  
que tiene un conjunto

El campo etiquetado en correspondencia asociativa por conjuntos es menor que el totalmente asociativo pero mayor que el directo.

TEST: en una memoria caché de un mismo tamaño:

- si aumento el tamaño de los bloques tendré menos fallos forzados o de 1ª referencia.
  - " " " " " " " " más fallos de conflicto.
- D.C. en correspondencia al tamaño de los bloques.

Una M.C. en correspondencia totalmente asociativa NO tiene folios de conflicto.

### TEMA 3.

## MUODOS DE DIRECCIONAMIENTO

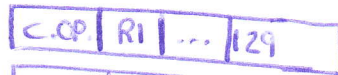
1. IMMEDIATE : ADD R1, #59



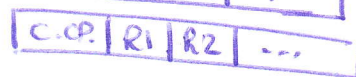
## 2. DIRECTO :

## 2.1. ABSOLUTE:

2.1.1. A MEMORIS: ADD R1, 129



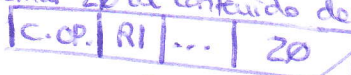
2.1.2. A REGISTRO: ADD R1, R2



2.1.3. A PÁGINA Ø.

## 2.2. RELATIVO:

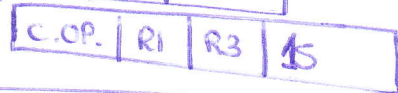
2.2.1. APC : ADD R1, #20



2.2.2. - A REG. BASE : ADD R1, 15(R3)



2.2.3. A REG. INDEX: (ADD R1, 15 (++R3))



### 3. INDIRECTO :

### 3.1. Absolute

3.1.1. A MEMORIA: ADD RI, (129)

3.1.2. A REGISTRO: ADD R1, (R2)

3.2. RELATIVE A REGISTER BASE: ADD RI, (15(R3))

Suma el contenido de R1 con el contenido de la dirección de memoria que está almacenada en la dirección 129.  
→ Suma el contenido de R1 con el contenido de la dirección de memoria que está almacenada en R2.

4. Implícito: CLC

5x1=5, 5x2=10, 5x3=15  
la tabla del cinco te la  
sabes gracias a la  
musiquita porque...

con  
**MÚSICA**  
entra mejor

Si quieres ganar entradas dobles  
para los mejores festivales

**CONTROL**  
Feel make Feel

Síguenos

**CONTROL**  
Feel make Feel

HACER USO DE LA PILA EN LOS SIGUIENTES CASOS:

CALL → guardamos PC.

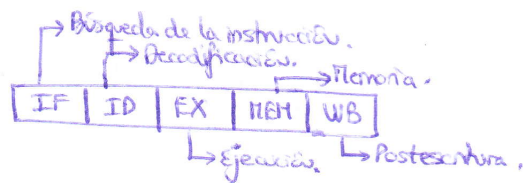
INT → guardaremos PC y SR (registro de estado).

RET → debemos extraer PC de la cabecera de la pila.

IRET → debemos extraer PC, SR en orden inverso a como fueron guardados en la pila.

TEMA 4: DLX.

FASES DEL PROCESADOR DLX:



**Bilbao**  
BBK Live

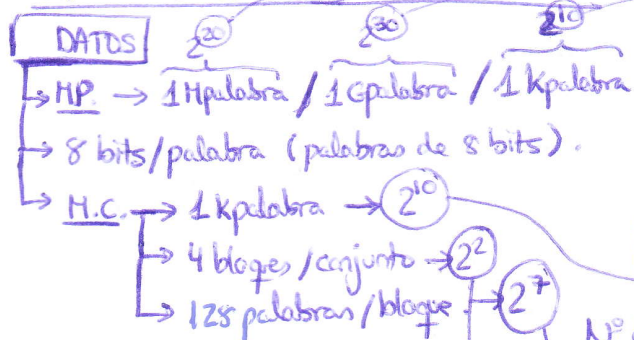
Festival de les Arts

**CABO DE PLATA**

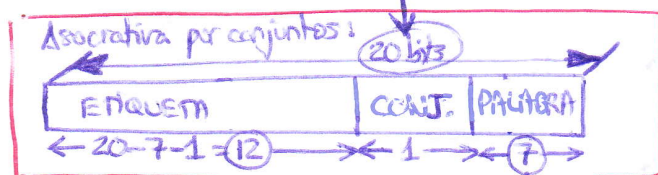
WUOLAH



## EJERCICIOS TEMA 2: MEMORIA CACHE.



→ N° de bits del formato de la dirección.



$$2^x \geq 128 \Rightarrow x = 7 \text{ bits}$$

$$\text{N° de bloques} = \frac{\text{Tamaño M.C.}}{\text{palabras/bloque}} = \frac{2^{10}}{2^7} = 2^3 \text{ bloques} = \boxed{8 \text{ bloques}}$$

$$\text{N° de conjuntos} = \frac{\text{N° de bloques}}{\text{bloques/conjunto}} = \frac{2^3}{2^2} = 2 \text{ conjuntos}$$

↓  
Necesito 1 bit para distinguir 2 conjuntos

### FRECUENCIA DE USO DE UN BLOQUE:

Respecto al global de M.C.:

$$\text{Fuso}_{\text{B}} = \frac{\text{N° de veces q se ha referenciado el bloque}}{\text{N° de referencias totales de todos los bloques}} \cdot 100 = \text{resultado en tanto por ciento}$$

Respecto a un conjunto determinado:

$$\text{Fuso}_{\text{C}} = \frac{\text{N° de veces q se ha referenciado el bloque}}{\text{N° de referencias totales de todos los bloques del conjunto}} \cdot 100 = (\%)$$

### TEORÍA:

Correspondencia

- Correspondencia directa: los bloques pueden asociarse en cualquier bloque de la M.C.
- Asociativa por conjuntos: cada bloque tiene asociado un conjunto de bloques de M.C.
- Totalmente asociativa: cada bloque de M.P. tiene asociado un determinado lugar en M.C.

### Tipos de fallos:

- Forzoso o de 1ª referencia: cuando se referencia por primera vez un bloque.
- De capacidad: cuando llamamos a un bloque q ya se referencia anteriormente y el conjunto de la M.C. está lleno.
- De conflicto: igual q fallo de capacidad pero el conjunto no está lleno. Estos fallos nunca existiran en correspondencia totalmente asociativa.

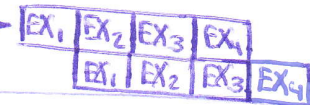
## EXERCICIOS TEMA 4 : DLX.

→ memoria con un único puerto de acceso.

- ▶ Con H. caché unificada, una fase HEH que utilice la memoria no puede compartir ciclo con una fase IF. En estos casos pondré detecciones por riesgo estructural. (Con H.C. partida no tenemos este problema)
- ▶ Las instrucciones que hacen uso de la memoria en la fase HEH son las de carga y almacenamiento.
- ▶ Adelantamiento (bypass) generalizado:
  - Adelantamiento desde la salida de la ALU a la entrada de la ALU. ( $EX \rightarrow EX$ )
  - Adelantamiento desde la salida de HEH/ALU a la entrada de la ALU.

Explicados abajo ↓

- ▶ Unidad aritmético-lógica segmentada: se podrán ejecutar en un mismo ciclo distintas fases de EX de operaciones del mismo tipo (operaciones desempeñadas por una misma unidad funcional)
- ▶ Unidad aritmético-lógica NO segmentada: no se pueden ejecutar fases EX de una misma unidad funcional



- ▶ Sin bypass implica que en caso de riesgo por dependencia de datos tipo lectura después de escritura tenga que detener y retrasar la fase de búsqueda (ID) hasta que termine la postescritura (WB) de la instrucción anterior a la que depende. En este caso, la fase WB e ID se podrán realizar en el mismo ciclo ya que consideramos que la postescritura (WB) se realiza en la primera mitad del ciclo y que la fase ID se realiza en la segunda mitad del ciclo.
- ▶ RESUMIENDO: siempre que haya dependencia hay que detener hasta postescritura (WB) para hacer fase ID.

(clasificación)

- ▶ Cuando tenemos que hacer un adelantamiento, si la instrucción de la que tenemos dependencia se trata de una instrucción de carga o almacenamiento haremos adelantamiento desde salida de memoria a entrada de ALU ( $HEH \rightarrow EX$ ). En caso contrario, efectuaremos adelantamiento desde salida de ALU a entrada de ALU ( $EX \rightarrow EX$ ).

Importante

Importante!!!

- ▶ DETENCIONES
  - Riesgo por dependencia de datos
  - Riesgo estructural