

ESTRUCTURA DE COMPUTADORES

Recuperación Segundo examen parcial

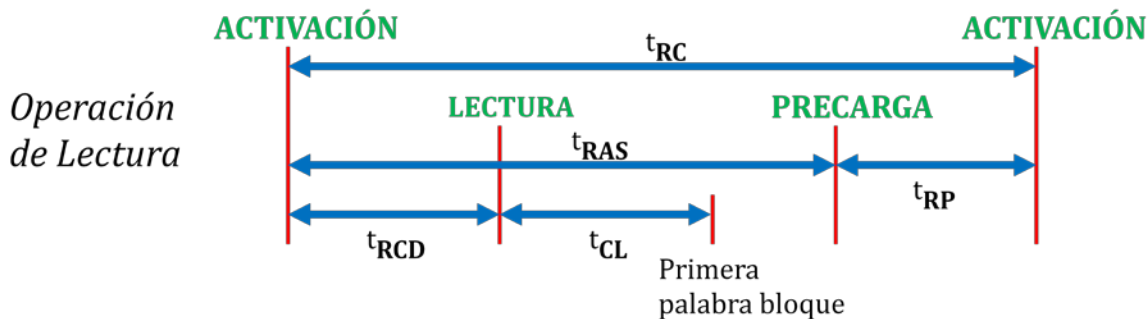
16 de junio de 2015

Apellidos y nombre	DNI	Grupo

- 1 (1 punto) Sólo P2** Una memoria SDRAM de 1 GByte y tamaño de palabra 32 bits se halla organizada en 8 bancos, cada uno de los cuales está formado por 8192 (8K) filas. Se pide:

Nº total de palabras de la SDRAM	256 M
Nº de palabras contenidas en una fila	4 K
Nº de líneas de selección de banco (BAi)	3
Nº de líneas Ai del bus de direcciones a la entrada de la SDRAM	13
Tamaño máximo de bloque que se puede definir (en nº de palabras)	4 K

- 2 (1 punto)** Un módulo DIMM de memoria SDRAM DDR de 128M x 64 bits trabaja a una frecuencia de reloj (f_{clk}) de 100 MHz. Su cronograma de lectura con la temporización de las órdenes es el que se muestra en la Figura 1, donde el valor de los parámetros temporales expresados en ciclos de reloj es el siguiente: $T_{RC}=10$, $T_{RAS}=6$, $T_{RP}=2$, $T_{RCD}=3$ y $CL=2$. Se pide:



- a) (0,5 puntos) Calcula el ancho de banda pico de dicho módulo, expresado en MBps.

$$BW_p = 2 \times 100 \text{ MHz} \times 8 \text{ B} = 1600 \text{ MBps}$$

- b) (0,5 puntos) Tiempo de lectura de un bloque de 8 palabras (tiempo total empleado desde el envío de la orden de "Activación" hasta la recepción del último dato del bloque), expresado en nanosegundos

$$\text{Tiempo lectura} = T_{clk} \times (T_{RCD} + CL + 8 \text{ palabras} / 2 \text{ ciclos}) = 10 \text{ ns} \times (3 + 2 + 4) \text{ ciclos} = 90 \text{ ns}$$

3 (2 puntos) El sistema de memoria principal de un procesador MIPS R2000 está constituido por dos módulos de memoria:

- Módulo M1: 512 MB a partir de la dirección 0x40000000
- Módulo M2: 128 MB a partir de la dirección 0xD0000000

Se pide:

a) (0,3 puntos) ¿Cuál es la mayor dirección contenida en el módulo M2?

0xD7FFFFFF

b) (0,3 puntos) ¿Cuál es el rango de direccionamiento del espacio libre comprendido entre los módulos M1 y M2?

0x60000000 – 0xCFFFFFFF

c) (0,3 puntos) ¿Cuál es el tamaño del espacio libre que queda entre los módulos M1 y M2?

0xD0000000 – 0x60000000 = 0x70000000 = 1792 MB

Alternativamente podría calcularse como ...

3x512 MB + 256 MB = 1792 MB

d) (0,3 puntos) ¿Cuál es la función de selección (activa por nivel bajo) del módulo M1?

/SEL = A31 + /A30 + A29

e) (0,5 puntos) Supóngase que se añade un nuevo módulo M3 cuya función de selección es /SEL = /A31 + A30 + /A29 + A28

- Calcula el tamaño del módulo M3

256 MB

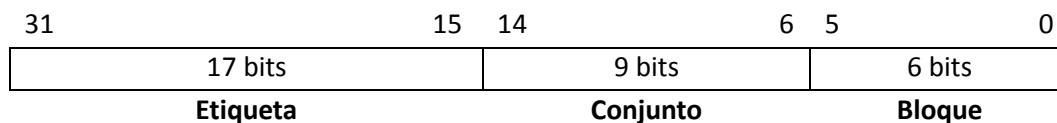
- ¿Cuál será el rango de direccionamiento del módulo M3?

0xA0000000 – 0xAFFFFFFF

f) (0,3 puntos) Indica el nombre de las líneas del bus de direcciones que se emplearían para el direccionamiento interno del módulo M2, tanto las líneas A_i como las BE_i correspondientes.

A26, A25, A24, ..., A3, A2, /BE3, /BE2, /BE1, /BE0

- 4 (2.5 puntos)** Se dispone de una memoria cache (MC) de 128 KB con correspondencia asociativa por conjuntos y memoria principal de 4 GB. El formato de la dirección de memoria principal es el que se muestra a continuación



Se pide:

- a) (0,5 puntos)

Nº de total de bloques la MC	$2^{17} / 2^6 = 2048$ bloques
Nº de conjuntos de la MC	512 conjuntos
Tamaño de bloque	64 B
Nº de vías	4-way

- b) (0,5 puntos) Suponiendo que la MC emplea algoritmo de reemplazo LRU, política actualización de escritura posterior (*write-back*) y política de ubicación en escritura (*write-allocate*), calcular el tamaño en bits de la memoria de control asociada a la cache. JUSTIFICAR el cálculo indicando el número total de entradas de la memoria de control, así como el número de bits de cada entrada, detallando los bits que componen cada uno de los campos en que conforman las entradas.

$$2048 \times [1(V) + 1(M) + 17(\text{Etiqueta}) + 2(\text{LRU})] = 2048 \times 21 \text{ bits} = 43008 \text{ bits}$$

- c) (0,5 puntos) **Sólo P2** ¿Cómo se podría reducir el tamaño de la memoria de control sin variar el tamaño de la MC ni la política de actualización *write-back*?

- 1) Reduciendo asociatividad (2-way o directa) para reducir el tamaño de las entradas
- 2) Aumentando tamaño de bloque para reducir el número de entradas
- 3) Empleo de un algoritmo de reemplazo que no precise bits de control (p.e., alg. aleatorio)

- d) (1 punto) **Sólo P2** A la hora de configurar el nivel L1 de cache se puede optar por una configuración unificada de 128 KB tanto para instrucciones como para datos o por una configuración dual de 64 KB (L1I) + 64 KB (L1D) para instrucciones y datos, respectivamente. Las tasas de acierto y tiempos de acceso se muestran a continuación en la tabla. Sabiendo que el tiempo de acceso a la memoria principal es de 200 ns y que un 70% de los accesos es a instrucciones, indicar cuál de las dos configuraciones proporciona un menor tiempo medio de acceso a memoria. JUSTIFICAR la respuesta

Cache Unificada		Cache dual			
HL1	TL1	HL1I	TL1I	HL1D	TL1D
0,97	2 ns	0,98	1 ns	0,95	1 ns

Cálculo tiempo medio de acceso cache unificada:

$$T_m = 0,97 \times 2 \text{ ns} + (1 - 0,97) \times 200 \text{ ns} = 7,94 \text{ ns}$$

Cálculo tiempo medio de acceso cache dual:

$$H_m = 0,98 \times 0,7 + 0,95 \times 0,3 = 0,971$$

$$T_m = 0,971 \times 1 \text{ ns} + (1 - 0,971) \times 200 \text{ ns} = 6,77 \text{ ns} \text{ (MEJOR CONFIGURACIÓN!!!!)}$$

5 (3.5 puntos) Considérese el siguiente programa que inicializa un vector V de números enteros positivos con el valor máximo de sus 68 componentes:

Ensamblador	
V:	<pre> .data 0x10000000 .word 8,15,... # 68 componentes .text 0x004000A0 lui \$at,0x0040 #carga puntero a V ori \$t0,\$at,0x00A0 li \$t1,68 #carga contador li \$t3,0 #inicializa reg. de máximo buc1: lw \$t2,0(\$t0) #lee elemento vector sltu \$at,\$t3,\$t2 #si \$t2>Máximo entonces \$at=1 beq \$at,\$zero,nomax add \$t3,\$zero,\$t2 #Si \$t2>Máximo entonces Actualiza nuevo máximo nomax: addi \$t1,\$t1,-1 #decrementa contador addi \$t0,\$t0,4 #actualiza puntero bnez \$t1,buc1 #permanece en bucle mientras contador>0 lui \$at,0x0040 #carga puntero a V ori \$t0,\$at,0x00A0 li \$t1,68 #carga contador buc2: sw \$t3,0(\$t0) #actualiza elemento vector con valor Máximo addi \$t1,\$t1,-1 #decrementa contador addi \$t0,\$t0,4 #actualiza puntero bnez \$t1,buc2 #permanece en bucle mientras contador>0 </pre>

El procesador dispone de memorias cache separadas para instrucciones y para datos (256 B + 256 B) con bloques de 16 bytes y correspondencia asociativa por conjuntos 2-way. En el momento en que comienza la ejecución del programa, todas las líneas de la cache son inválidas.

a) (1,5 puntos) **Sólo P2** En relación a la cache de instrucciones, indicar:

Instrucciones/bloque de código	4
Nº de bloques distintos de código que se referencian	5
Direcciones de los bloques de código referenciados (en hexadecimal)	0x004000A, 0x004000B, 0x004000C, 0x004000D, 0x004000E
Conjuntos de la cache a los que se mapearán los bloques referenciados	2, 3, 4, 5, 6

b) (2 puntos) En relación a la cache de datos, indicar:

Componentes de V/bloque de datos	4
Nº de bloques distintos de datos que se referencian	17
Número de accesos	$68(R) + 68(W) = 136$
Número de fallos (en su caso, indicar cuántos son de inicio, capacidad y conflicto)	Inicio: 17; Capacidad: 3; TOTAL= 20 fallos
Número de reemplazos	4
Tasa de aciertos	$136 - 20 / 136 = 0,853$