

Fundamentos de Computadores 2018/19 (2º cuatrimestre)
Ejercicios – Hoja 3 (modulo 10)

1. Sea un computador con una MP de 1 Mbyte y dotado con una MC de 16 bloques de cache de 64 bytes cada uno. Determinar el formato de la dirección de memoria física desde el punto de vista del emplazamiento en memoria cache, especificando el número de bits que ocupa cada uno de los campos que la componen para emplazamiento directo.
2. Rellenar la siguiente tabla (siempre que sea posible), suponiendo que siempre se accede a un byte, donde se especifica el tamaño de la memoria principal, el tamaño de la memoria cache (emplazamiento directo), el tamaño del bloque y los bits que se reservan para la etiqueta:

Tamaño MP	Tamaño MC	Tamaño Bloque	Etiqueta
1MB	256B	32B	
1MB	1KB	512B	
1MB	512B		3
1MB		32B	5
2MB	16KB	1KB	
2MB		1KB	7
2MB	1KB	128B	
2MB		8KB	3
4MB	1MB		2
16MB	1MB	256KB	

3. Sea un computador con una memoria principal de 4 Kpalabras y una cache de emplazamiento directo con 4 bloques y 64 palabras por bloque. Se ejecuta 10 veces una serie de accesos en lectura. Estas lecturas referencian consecutivamente todas las palabras comprendidas entre las direcciones 0x096 y la 0x247 (expresadas en hexadecimal). Indicar el número de fallos que se producen, en qué bloques de la cache se ubica cada rango de direcciones y la etiqueta de cada bloque.
4. Sea un computador con memoria cache con las características siguientes:
 - Memoria principal: 64 KB
 - Memoria cache de 256 bytes con bloques de 64 bytes.

Se pide:

- a) Indicar el formato de la dirección para MP y para la MC
- b) Calcular el número de bits necesarios para implementar la MC (incluyendo datos y etiquetas).
- c) En un momento dado los contenidos del array de etiquetas de la cache (en hexadecimal) son los indicados en la tabla. Expresar en hexadecimal el rango de direcciones de memoria principal ubicadas en cada bloque de la memoria cache.
- d) Supongamos que un programa realiza la siguiente cadena de referencias en lectura (en hexadecimal): D380, 6010, D3F0, 6018, 1830. Calcular el número de fallos que se producen e indicar cómo se queda el array de etiquetas de la memoria cache después de acceder a la última dirección.

Etiqueta	Bloque
18	0
25	1
D3	2
51	3

5. Sea un computador con memoria cache con las características siguientes:
- Memoria principal de 32 KB.
 - Memoria cache de 512 bytes con bloques de 128 bytes.
- Indicar el formato de la dirección para MP y para la MC
 - En un momento dado los contenidos del array de etiquetas de la cache (en hexadecimal) son los indicados en la tabla. Expresar en hexadecimal el rango de direcciones de memoria principal ubicadas en cada bloque de la memoria.
 - Supongamos que un programa realiza la siguiente cadena de referencias (accesos a nivel de palabra): de 2080 a 209F, de 2880 a 289F y de 03F0 a 0410. Indicar el número de aciertos que se producen.

Etiqueta	Bloque
35	0
10	1
10	2
08	3

6. (Junio 2013) Sea un computador con una memoria principal de 1MB, con una memoria cache de emplazamiento directo de 1KB con bloques de 128 bytes. La memoria es direccionable por bytes y el tamaño de palabra es de 4 bytes. Se pide:
- (0,5 puntos) Indicar el formato de la dirección para MP y para MC.
 - (1 punto) En un momento dado sólo cuatro de las entradas de la cache contienen datos válidos, los contenidos del array de etiquetas de la cache (en hexadecimal) son los indicados en tabla inferior. Expresar en hexadecimal el rango de direcciones de memoria principal ubicadas en la memoria cache.

Bloque	Etiqueta
0	
1	
2	0x3A7
3	
4	0x100
5	0x100
6	
7	0x0D2

- Partiendo de la situación indicada en la tabla anterior, supongamos que un programa realiza la siguiente cadena de referencias en lectura (en hexadecimal): todas las palabras entre E9C38 y E9D7C ambos incluidos, todas las palabras entre las direcciones 4019C y 402AC, ambas incluidas, y todas las palabras entre las direcciones 34A84 y 34BC0, ambas incluidas. Indicar en cada caso los fallos que se producen e indicar cómo se queda el array de etiquetas de la memoria cache después de acceder a la última dirección.
7. (Septiembre 2013) Supongamos una jerarquía de memoria que consta de una Mp direccionable por palabras, donde cada palabra tiene una anchura de 32 bits, y una Mc de emplazamiento directo con 128 bloques, siendo cada bloque de 512 palabras. Además, sabemos que:
- Las direcciones de Mc tienen un campo de etiqueta de 8 bits.
 - El tiempo de acceso a la Mc es 1 ciclo de reloj.
 - La penalización por fallo es 130 ciclos de reloj.
 - El tiempo de acceso a Mp es 10 ciclos de reloj.

Sobre esta jerarquía se ejecuta 10 veces seguidas una cadena de referencias entre las direcciones 0xF59600 y 0xF698C7, ambas inclusive, estando la cache inicialmente vacía.

- (0,5 puntos) ¿Cuál es el tamaño de la Mp expresado en megabytes?

- b) (1 punto) Determina el número total de fallos producidos.
- c) (0,5 puntos) Teniendo en cuenta la siguiente definición, determina la Tasa de Fallos en este problema.

$$Tasa\ de\ Fallos = \frac{Número\ total\ de\ fallos}{Número\ total\ de\ referencias\ a\ memoria}$$

- d) (0,5 puntos) Determina el tiempo medio de acceso a memoria.
- e) (0,5 puntos) Calcula la mejora de velocidad que se obtiene al emplear esta jerarquía de memoria, en comparación con la alternativa de usar solamente la Mp.

8. (Junio 2014) El código de un determinado programa está constituido por instrucciones máquina con la siguiente frecuencia de ejecución: aritméticas 50%, carga 20%, almacenamiento 10% y salto condicional 20% (donde la mitad de los saltos se toman). En cierto procesador las instrucciones consumen los siguientes ciclos: aritméticas 4, carga 5, almacenamiento 4, salto no tomado 3, salto tomado 4.

- a) Calcula el CPI del procesador al ejecutar este programa.
- b) Sabiendo que el programa se ha ejecutado en 8.2ms, y que la frecuencia del procesador es de 1GHz, calcula el número de instrucciones que se han ejecutado.
- c) Suponiendo que por cada 100 instrucciones ejecutadas se produce 1 fallo de cache, que implica una penalización de 300 ciclos de reloj, calcula el nuevo tiempo de ejecución del programa. ¿Cuál sería entonces el nuevo CPI?

9. (Junio 2014) Sea un computador con un procesador ARMv4. EL procesador cuenta con una memoria cache de emplazamiento directo de 256 bytes con bloques de 64 bytes, unificada para datos e instrucciones (se almacenan en la misma cache los bloques de instrucciones accedidos en la etapa FETCH y los bloques de datos accedidos durante las etapas que hacen acceso a memoria en las instrucciones ldr y str).

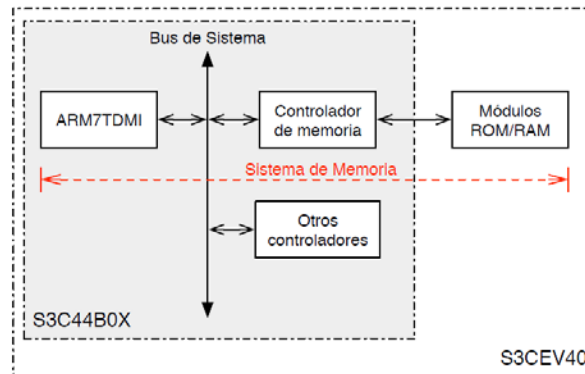
- a) En dicho sistema se quiere ejecutar el siguiente programa. Explique razonadamente lo que hace el código.

```
.equ N, 16
.data
A: .word N valores enteros separados por comas
B: .word N valores enteros separados por comas
.text
start:    ldr r0, =A
          ldr r1, =B
          mov r4, #N
L1:       ldr r2, [r0]
          ldr r3, [r1]
          add r2, r2, r3
          and r2, r2, #0xF
          str r2, [r0]
          add r0, r0, #4
          add r1, r1, #4
          sub r4, #1
          cmp r4, #0
          bne L1
end:      b .
```

- b) Con direcciones de 32 bits, indicar el formato de la dirección para MP y para la MC.
- c) El programa se enlaza ubicando la sección .data a partir de la dirección 0x0C000000, y la sección .text se coloca a continuación de la sección .data. Obtenga los rangos de direcciones que ocupan el array A, el array B y las instrucciones, indicando para cada rango el/los bloques de memoria correspondientes, y el marco de bloque (bloque de cache) y la etiqueta asociados.

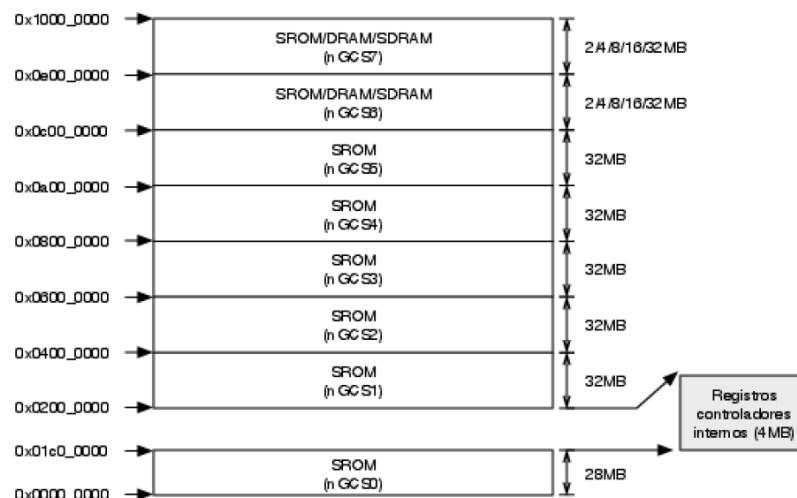
- d) Determine el número de aciertos y fallos de cache que se producirían al ejecutar el código anterior, hasta la primera vez que se ejecuta la instrucción que está en la etiqueta end.
- e) Suponga que el programador cambia el valor N por 32. Obtenga de nuevo los bloques de memoria en que se ubicarían los datos y las instrucciones. Explique cualitativa y razonadamente cómo afectaría este cambio a los fallos de cache (No es necesario calcular el nuevo número de fallos, sólo explicar lo que sucedería).

10. En la placa del laboratorio (Embest S3CEV40), tanto el procesador ARM7TDMI como el controlador de memoria, así como otros controladores y dispositivos de E/S, se encuentran integrados dentro un mismo chip, el System on Chip S3C44B0X de Samsung.



El bus de direcciones del ARM7TDI es de 32 bits, por lo que es capaz de direccionar potencialmente un espacio total de 4GB de memoria. El controlador memoria es el responsable de actuar de interfaz entre los módulos de memoria externos (ROM o RAM) y el bus del sistema. Es fácil adivinar que el mapa de memoria del sistema viene determinado en gran medida por las características y configuración de este elemento del sistema de memoria. Cuando el procesador pone una dirección en el bus, si ésta se encuentra dentro del rango del controlador, él se encarga de generar las señales necesarias para realizar el acceso al módulo de memoria que corresponda.

El controlador de memoria del S3C44B0X reduce el espacio de direcciones efectivo a 256MB y lo divide en 8 fragmentos independientes, denominados bancos, tal y como ilustra la siguiente figura. Cada banco puede ser asignado a un chip de memoria externo distinto (módulo), aunque sólo los dos últimos bancos admiten cualquier tipo de memoria (SRAM, SRAM o SDRAM).



Cuando se accede a una dirección dentro del rango de uno de estos bancos, el controlador de memoria activa automáticamente una señal de Enable asociada al banco.

Se quiere implementar una memoria que cubra parcialmente dos bancos en el siguiente rango de direcciones:

- 2MB de ROM Flash [0x00000000-0x001FFFFF] para el arranque del sistema.
- 8MB de SDRAM [0x0C000000-0x0C7FFFFF] para programas y datos.

Asumir que los chips de memorias ROM y SDRAM utilizados tienen un interfaz similar al del chip AS6C62256, pero con los siguientes tamaños:

- ROM 512K x 16bits.
- RAM 1M x 16bits.

Diseñar la estructura interna de estos dos bancos de memoria.

Simplificación: Tan solo se permiten accesos a memoria de tamaño palabra y además la memoria está alineada

11. Dado un procesador con bus de direcciones de 32 bits y un bus de datos de 32 bits al que se desea dotar de una memoria de 64 M-palabras situada a partir de la dirección 0x00000000. Para ello se dispone de chips SRAM de 16 MBytes con entrada de control CE* (Chip Enable, activa a nivel bajo) y puertas lógicas.

Se pide:

- El número de chips necesarios.
- La forma en que se conecta el bus de datos y el bus de direcciones del procesador y los chips de memoria.
- La conexión CE de cada uno de los chips de memoria al procesador si se emplea decodificación total.
- La conexión CE de cada uno de los chips de memoria al procesador si se emplea decodificación parcial.

12. (Septiembre 2014) Sea un computador con un procesador ARMv4. EL procesador cuenta con una memoria cache de emplazamiento directo de 512 bytes con bloques de 64 bytes, unificada para datos e instrucciones (se almacenan en la misma cache los bloques de instrucciones accedidos en la etapa FETCH y los bloques de datos accedidos en la ejecución de las instrucciones ldr y str).

- En dicho sistema se quiere ejecutar el siguiente programa. Explique razonadamente lo que hace el código (debe darse un pseudo-código tipo C/C++ equivalente).

```
.equ N, 32

.data
A: .word N valores enteros separados por comas
B: .word N valores enteros separados por comas

.bss
C: .space N*4

.text
start:    ldr r0, =A
          ldr r1, =B
          ldr r2, =C
          mov r4, #N
L1:       ldr r5, [r0]
          ldr r6, [r1]
          add r5, r5, r6
          mov r5, r5, lsl #2
          str r5, [r2]
          add r0, r0, #4
          add r1, r1, #4
          add r2, r2, #4
          sub r4, #1
          cmp r4, #0
          bne L1
end:      b .
```

- Con direcciones de 32 bits, indicar el formato de la dirección para MP y para la MC.

- c) El programa se enlaza ubicando la sección .data a partir de la dirección 0x0C000000, con la sección .bss colocada justo después de la sección .data y la sección .text a continuación de la sección .bss. Obtenga los rangos de direcciones que ocupan el array A, el array B, el array C y las instrucciones, indicando para cada rango el/los bloques de memoria correspondientes, y el marco de bloque (bloque de cache) y la etiqueta asociados.
- d) Determine el número de aciertos y fallos de cache que se producirían al ejecutar el código anterior, hasta la primera vez que se ejecuta la instrucción que está en la etiqueta end.
- e) Suponga que el programador cambia el valor N por 16. Obtenga de nuevo los bloques de memoria en que se ubicarían los datos y las instrucciones. Explique razonadamente cómo afectaría este cambio a los fallos de cache y a la tasa de fallos (cociente entre fallos y accesos totales)

13. (Junio 2015) Una jerarquía de memoria está compuesta por una memoria principal de 16 Mbytes y una cache con emplazamiento directo de 16 Kbytes. El tamaño de bloque es de 256 bytes. Sobre esta jerarquía de memoria se ejecuta un millón de accesos de lectura, obteniéndose una tasa de aciertos de 0,995. Se supone que el tiempo de acceso a la memoria principal es de 20 ns, el tiempo de acceso a la cache es de 1 ns y la penalización por fallo es de 400 ns. Responde a las siguientes cuestiones:

Nota: Todos los datos numéricos del problema están en decimal. Las respuestas numéricas pueden darse en el sistema decimal o hexadecimal, a elección del estudiante.

- a) Muestra el formato de la dirección para la Mp y la cache.
- b) Si el bloque nº 87 de la Mp tiene que llevarse a la cache ¿qué nº de bloque de cache se le asigna?
- c) ¿A cuántos bloques de Mp le corresponde el mismo bloque en la cache? Indica el nº del primer y el último bloque de Mp a los que les corresponde alojarse en el bloque 22 de la cache.
- d) ¿Cuál es el tiempo medio de acceso a memoria?
- e) ¿Cuánto se tarda en ejecutar la secuencia de accesos a memoria descrita?
- f) ¿Qué ganancia de velocidad se obtiene respecto del mismo sistema, pero sin la cache?

14. (Septiembre 2015) Sea un computador con una memoria principal de 1 GByte, una memoria cache con emplazamiento directo y con capacidad para albergar 16 bloques de 256 Bytes cada uno. Se pide:

- a) Indicar el formato de dirección para MP y MC si la memoria se direcciona tamaño byte.
- b) En un momento dado las etiquetas de algunos marcos de la cache son las indicadas en la tabla. Expresar en hexadecimal el rango de direcciones de memoria principal ubicadas en estos bloques de memoria cache.

Bloque MC (Hex.)	Etiqueta (Hex.)	Rango direcciones MP(Hex.)
6	33	-
A	77	-

- c) Partiendo del estado de cache del apartado anterior (el resto de marcos están vacíos), se ejecuta un programa con las siguientes referencias a memoria de tamaño byte. Calcular el número de fallos de cache que se producen:

Comentario	Dirección (hex)
Una vez	33008
	...
	347F7
Bucle 2 veces	347F8
	...
	35FF7

- d) La memoria principal tiene un tiempo de acceso de 10 ns, la memoria cache 1 ns y una penalización por un fallo de cache 500 ns. Calcula el tiempo de acceso a las referencias del programa anterior con y sin cache. Indica cuantas veces más rápido es el sistema con la cache.