

ORGANIZACIÓN Y ESTRUCTURA DE LA MEMORIA: CACHÉS Y MEMORIA VIRTUAL

3.1.1. Una máquina que utiliza un reloj de 50 MHz, dispone de un sistema de memoria de dos niveles. La tasa de aciertos en el primer nivel es del 95% y la penalización total por fallo es de 20 ciclos de reloj. El tiempo de acceso al primer nivel es de un ciclo de reloj. Si se duplica el tamaño de la unidad del primer nivel, se mejora la tasa de fallos a 0,03 fallos por referencia a memoria. Este aumento de tamaño hace que el tiempo de acceso a la unidad mejorada aumente en un 20%. Indicar si la mejora es positiva, utilizando como parámetro en ambos casos el tiempo de acceso medio.

3.1.2. En la tabla adjunta se indica el porcentaje de cada tipo de instrucciones y su duración media de un determinado sistema.

TIPO	ALU-INT	ALU-CF	Load/Store	OTRAS
PORCENTAJE	45%	25%	15%	15%
CICLOS DE EJECUCION	1	6	4	2

Además todas las instrucciones necesitan 4 ciclos para el acceso a memoria correspondiente a la captura de instrucción.

Se proponen dos mejoras independientes:

1.- Optimizar la ALU para coma flotante y reducir a la mitad el tiempo de ejecución.

2.- Incorporar una caché unificada con un tiempo de acceso de 1 ciclo, mejorando así tanto el tiempo de captura de instrucciones como la ejecución de Load/Store. La caché presenta una tasa de aciertos del 95% y un penalización total por cada fallo de 10 ciclos.

Se pide, utilizando necesariamente la ley de Amdahl:

- ¿Cuál de las dos mejoras es la más eficiente y en cuánto?
- ¿Cuál será la mejora total si se aplican ambas al tiempo?

3.1.3. Los parámetros que definen un sistema de memoria son: memoria principal de 32KB, memoria caché completamente asociativa de 4KB, tamaño de bloque de la caché, 8 palabras (16 bytes). El tiempo de acceso a la memoria principal es 10 veces mayor que el de la memoria caché. Se pide:

- ¿Cuántos comparadores de hardware son necesarios?
- ¿Cuál es el tamaño del campo para la etiqueta?
- Si se utilizara correspondencia directa, ¿cuál sería el campo de la etiqueta?
- Si el tiempo de acceso a caché es de 200 ns, ¿cuál será la razón de aciertos necesaria para lograr un tiempo de acceso medio de 500 ns?

3.1.4. La siguiente secuencia de números en decimal, representa una cadena de direcciones de 8 bits para el acceso a memoria: 1, 4, 8, 5, 23, 20, 17, 19, 23, 56, 9, 11, 4, 43, 5, 6, 9 y 17. Suponiendo una caché de correspondencia directa con 8 bloques de una palabra, que inicialmente está vacía, indicar para cada referencia de la lista si corresponde a un acierto o a un fallo. Indicar el valor en binario de las etiquetas almacenadas en caché al final de la secuencia. Suponer 1 palabra = 1 byte.

3.1.5. computador tiene una memoria principal de 16 Mbytes y emplea un bus de datos de 32 bits, dispone de una unidad caché de 8 kBytes. Determinar el número de bits de cada campo de la dirección según las siguientes organizaciones:

- Correspondencia directa con una palabra por bloque.
- Correspondencia directa con ocho palabras por bloque.
- Correspondencia asociativa por conjuntos de cuatro vías y dos palabras por bloque.
- Completamente asociativa y cuatro palabras por bloque.

3.1.6. Una memoria caché de correspondencia directa, tiene un total de 256 bloques, de 16 bytes cada uno. El procesador genera direcciones de 32 bits. En esta caché no son necesarios los bits de validez. Se pide:

- Indicar los bits de los campos en los que se divide la dirección.
- Indicar el número de fallos que se generan, cuando se solicitan desde la CPU un bucle que se ejecuta diez veces con la siguiente secuencia de direcciones (en hexadecimal): A3272105 -- A3502120 -- A3271130 -- A3272103 -- A3502104 -- A3261130 y A3272104.

NOTA: La caché está llena, pero inicialmente no contiene ninguna de las direcciones de la secuencia anterior.

ORGANIZACIÓN Y ESTRUCTURA DE LA MEMORIA: CACHÉS Y MEMORIA VIRTUAL

3.1.7. Una caché asociativa por conjuntos tiene un tamaño de bloque de cuatro palabras de 16 bits y un tamaño de conjunto de 2. La caché puede acomodar un total de 4096 palabras. El tamaño de memoria principal que es transferible a caché es de 64K x 32 bits.
Diseñar la estructura de la caché y mostrar como son interpretadas las direcciones del procesador.

3.1.8. Considérese que un programa accede a las siguientes direcciones de memoria (en decimal) 1, 4, 8, 5, 20, 17, 19, 56, 9, 11, 4, 43, 5, 6, 9,17
Cada una de estas referencias accede a datos de 1 byte de tamaño. Indicar en todos los casos el contenido final del directorio caché y el número total de fallos. En todos los casos se emplea una política de ubicación LRU. Suponga que se tiene una caché de 16 bytes de capacidad, inicialmente vacía.
a) Con bloques de 1 byte y de correspondencia directa.
b) Con bloques de 4 bytes y de correspondencia directa.
c) Con bloques de 1 byte y asociativa por conjuntos de dos vías.
d) Con bloques de 4 bytes y totalmente asociativa.
e) Con los datos obtenidos indicar que estructura caché es la mejor, cual la peor y cual es el porcentaje de mejora entre ambas organizaciones.

3.1.9. Considerar una unidad caché que contiene 512 bytes de información. La unidad es asociativa por bloques de 4 vías con 4 palabras por bloque. La unidad dispone de los bits de validez y de los bits de modificación (suciedad) necesarios. El bus de direcciones es de 32 bits mientras que el de datos es sólo de 16 bits. Se pide:
a) ¿Cómo se divide la dirección para el acceso a la caché?
b) ¿Cuál es el tamaño en bits de la unidad caché?
c) ¿Cuál es el orden (índice) de la línea de entrada en la que se debe guardar la información correspondiente a la dirección 000010AF₁₆?
d) La dirección de memoria anterior y la dirección FFFF7AXY₁₆ se encuentran simultáneamente guardadas en la unidad caché. Ambas tienen el mismo índice, señalan a la misma palabra dentro del bloque pero se encuentran en distinta vía. ¿Cuántas direcciones cumplen las condiciones anteriores? ¿Cuáles son los posibles valores de X e Y.

3.1.10. La siguiente figura representa los contenidos de una caché con 4 Vías, un tamaño de 64 bytes y política de funcionamiento de Escritura Directa sin asignación en escritura (EDSAE) y reemplazamiento LRU.

VIA0			VIA1			VIA2			VIA3		
Etiqueta	Word1	Word0	Etiqueta	Word1	Word0	Etiqueta	Word1	Word0	Etiqueta	Word1	Word0
1101	A1	A0	0001	A3	A2	1110	A5	A4	1111	AF	AE
0011	B1	B0	0001	B3	B2	0111	B5	B4			
1110	C1	C0	1010	C3	C2						
1110	D1	D0	1101	D3	D2	1111	D5	D4	0000	D7	D6
0001	E1	E0									
0100	F1	F0	1111	F3	F2						

La etiqueta está expresada en binario y los campos Word1 y Word0 están expresados en hexadecimal y se corresponden respectivamente con la palabra alta y baja de cada bloque.

- a) Indique todos los recursos necesarios para su implementación.
b) ¿Qué dirección de memoria corresponde a las siguientes palabras que se encuentran almacenadas en la caché?

Dato	Dirección
C3	
A0	
A3	

- c) Suponga que se realiza la secuencia de accesos de la tabla, en la cual la primera columna indica si el acceso es una lectura o una escritura. Suponga que al iniciar la secuencia, la vía LRU en todas las entradas corresponde a los datos guardados en la vía 3. Rellene las columnas con los valores correspondientes, indicando si hay cambio en la caché, el lugar donde se produce.

R/W	Dirección	Vía	Número de entrada	Acierto/Fallo
Lectura	11100110			
Escritura	11110011			
Lectura	11110010			
Escritura	11110011			
Lectura	00000010			

ORGANIZACIÓN Y ESTRUCTURA DE LA MEMORIA: CACHÉS Y MEMORIA VIRTUAL

d) Partiendo de situación mostrada en la figura, indique la menor secuencia de accesos a memoria que complete la última entrada de todas las vías.

e) Se modifica la estructura de la caché incrementando en 3 bits el campo de etiqueta y se mantiene en 64 bytes su capacidad total con el mismo número de bytes por bloque. ¿Cuál es la organización de la nueva caché?

3.1.11. Se tiene un sencillo sistema RISC donde el bus de datos y de direcciones son ambos de 16 bits. El sistema dispone de una unidad caché no unificada. Ambas cachés tienen una capacidad de 16 KB, tienen una estructura asociativa de 4 vías, con una política LRU para el reemplazamiento, guardan 8 palabras por bloque y existen bits de suciedad o modificación donde corresponda. Se sabe que el acceso a memoria DRAM es de 100 ns y el acceso a caché es de 10 ns. Inicialmente ambas cachés están vacías. Justificando brevemente las respuestas, se pide:

- a) Los campos en los que se divide la dirección para el acceso a las cachés.
- b) La tasa de aciertos en la caché de instrucciones.
- c) La tasa de aciertos en la caché de datos
- d) El tiempo total utilizado para el acceso a memoria en la ejecución del código mostrado.

Nota: El código suma 100 a un vector de 50 elementos de 2 bytes, ubicado a partir de la posición 4400₁₆ y lo reescribe modificado en la misma posición.

Dirección	Instrucción	
00EC	MOVE 0,R1	; 0 ⇒ R1
00EE	MOVE 50,R2	; 50 ⇒ R2
00F0	L1: SUB R2,1,R2	; R2 -1 ⇒ R2
00F2	Bneg Fin	; Salto condicional. Si N = 1, salta a Fin
00F4	LOAD #4400, R1, R4	; Mem[4400+(R1)] ⇒ R4
00F6	ADD R1, 2, R1	; R1+ 2 ⇒ R1
00F8	CALL Sub	; Llamada a la rutina Sub
00FA	STORE R6, #4400, R1	; R6 ⇒ Mem[4400+(R1)]
00FC	JUMP L1	; Salta incondicional a la etiqueta L1
00FE	Fin: STOP	; Fin del programa principal
.....	
0400	Sub: ADD R4,100,R6	; R4 + 100 ⇒ R6
0402	RETURN	; Vuelve de la rutina

3.1.12. Un cierto sistema dispone de un bus de direcciones de 32 bits, siendo el tamaño de la palabra de un byte. Se le desea dotar de una pequeña unidad caché para instrucciones, con las siguientes características: 16 bloques, siendo cada bloque de 4 bytes. No son necesarios los bits de validez y la estrategia para reemplazamiento es LRU. Se consideran dos diferentes opciones, una caché de correspondencia directa (caché A) y una completamente asociativa (caché B).

Mostrar en un dibujo los bloques en los que se divide la dirección de memoria para el acceso a memoria en cada una de las opciones de la caché.

Dar una expresión matemática que a partir de la dirección de memoria n ($n \geq 0$), calcule la entrada de acceso a la unidad caché A.

Asumir que inicialmente la caché está vacía, y que el tamaño de instrucción es de 1 byte. Suponer que se lleva a cabo la siguiente secuencia de instrucciones: Las direcciones de memoria de 0-63 se ejecutan una vez. Las direcciones de memoria de la 64-131, pertenecen a un bucle que se ejecuta 10 veces.

Encontrar el número de faltas en ambas cachés. Calcular el tiempo de acceso medio en ambos casos y señalar para este programa cuál es la caché más rentable y en qué porcentaje se valora la mejora. Se conoce que tiempo de acceso para ambas cachés es $t_{CACHE} = 20$ ns y el tiempo de penalización por fallo en ambos casos es $t_B = 200$ ns.

3.1.13. Considerar tres máquinas con diferentes configuraciones de memoria caché:

MAQ1: Correspondencia directa con bloques de 1 palabra. La tasa de fallos es del 4% para instrucciones y del 8% para datos.

MAQ2: Correspondencia directa con bloques de 4 palabras. La tasa de fallos es del 2% para instrucciones y del 5% para datos.

MAQ3: Asociativa por conjuntos de dos vías con bloques de 4 palabras. La tasa de fallos es del 2% para instrucciones y del 4% para datos.

Para estas máquinas, la mitad de las instrucciones contienen una referencia de datos. Suponer que la penalización, en ciclos, de fallos de la caché es 6 más el tamaño del bloque en palabras. El CPI suponiendo máquinas ideales (sin fallos) vale 2 en todos los casos.

a) Determinar que máquina emplea más ciclos en los fallos de la caché.

ORGANIZACIÓN Y ESTRUCTURA DE LA MEMORIA: CACHÉS Y MEMORIA VIRTUAL

b) Si las frecuencias de funcionamiento son 100 MHz para MAQ1 y MAQ2 y 80 MHz para MAQ3, determinar el rendimiento de la máquina más rápida respecto a la más lenta.

3.1.14. Se tiene un sencillo sistema RISC donde el bus de datos es de 16 bits y el de direcciones de 12 bits. Todas las instrucciones ocupan una palabra. El sistema dispone de una unidad caché no unificada. Las cachés tienen una capacidad de 32 Bytes.

La caché de instrucciones tiene una estructura asociativa de 2 vías, con una política LRU para el reemplazamiento. La caché de datos es de correspondencia directa (CD). Ambas cachés guardan dos palabras por bloque. No existen bits de control de ningún tipo. Para cada palabra, el byte menos significativo se almacena en la posición de memoria más baja. Dentro de un bloque la palabra de la izquierda corresponde a una posición de memoria superior respecto a la de la derecha

Justificando necesaria y brevemente las respuestas, se pide:

a) Los campos en los que se divide la dirección para el acceso a las cachés.

b) El estado final de las memorias cachés tras la ejecución del programa. Suponer las cachés vacías y que la LRU en caso de ser necesaria es la vía 1. Para indicar el contenido de la memoria caché utilizar la notación D(XYZ) para indicar la palabra contenida en la dirección XYZ. Por ejemplo D(1EA) es el contenido de la primera instrucción del programa. Las lecturas de memoria principal son por bloques completos.

c) La tasa de aciertos en la caché de instrucciones y de datos.

Nota: El código realiza la suma de los 10 elementos de un vector (con datos de 2 bytes), ubicado a partir de la posición 402₁₆ y queda el resultado en el Registro R3.

1EA	MOVE 0,R1	;0 ⇒ R1, apuntador al arreglo
1EC	MOVE 10,R2	;10 ⇒ R2, cuenta las iteraciones
1EE	MOVE 0,R3	;0 ⇒ R3, lleva la suma
1F0 L1:	SUB R2,1,R2	;R2-1 ⇒ R2
1F2	Bneg Fin	;Salto condicional. Si N = -1, salta a Fin
1F4	LOAD #402,R1,R4	;Mem[402+(R1)] ⇒ R4
1F6	ADD R1,2,R1	;R1+2 ⇒ R1
1F8	ADD R3,R3,R4	;R3+R4 ⇒ R3
1FA	JUMP L1	;Salta incondicional a la etiqueta L1
1FC	Fin: STOP	;Fin del programa

3.1.15. La caché de un cierto sistema es unificada, asociativa de dos vías (CA2V), 16 entradas por vía y cuatro palabras por bloque. En un momento determinado, justo tras finalizar la instrucción Clear R2, la caché presenta el estado que se muestra en el esquema adjunto, donde se marcan con XXXX el dato que cada bloque contiene. Además asociado a cada bloque, se indican el estado del contador (1bit) utilizado para el algoritmo LRU implementado siguiendo el sistema estudiado en clase. A continuación se ejecuta el bucle 5 veces y se termina con la ejecución de la instrucción Halt. Nota: El sistema sólo maneja instrucciones y datos de 32 bits.

```

PC
0F00A4      Clear  R2                ; 0 → R2
0F00A8  Bucle: Load R1, R2, #7000 ; Mem [(R2) + 007000] → R1
0F00AC      Add   R3,R1,R5        ; R1 + R5 → R3
0F00B0      Add   R2, R2, #4       ; R2 + 4 → R2
0F00B4      Bne   R2, #20, Bucle   ; Si R2 ≠ 2010, salta a bucle
0F00B8      Jmp   Fin              ; Salta a Fin
-----
-----
F00500  Fin:   Halt                ; Finaliza la ejecución
    
```

VIA 1			VIA 2		
DIRECTORIO	BLOQUES	LRU	DIRECTORIO	BLOQUES	LRU
0F00	XXXX	1	F000	XXXX	0
00F0	XXXX	0	0000	XXXX	1
000F	XXXX	0	000F	XXXX	1
0000	XXXX	1	00F0	XXXX	0
F000	XXXX	1	0F00	XXXX	0
0F00	XXXX	0	F000	XXXX	1
00F0	XXXX	0	0000	XXXX	1
000F	XXXX	1	000F	XXXX	0
0000	XXXX	1	00F0	XXXX	0
F000	XXXX	0	0F00	XXXX	1
0F00	XXXX	0	F000	XXXX	1
00F0	XXXX	1	0000	XXXX	0
000F	XXXX	1	000F	XXXX	0
0000	XXXX	0	00F0	XXXX	1
F000	XXXX	0	0F00	XXXX	1
0F00	XXXX	1	F000	XXXX	0

Se pide, con los datos de los que se dispone y justificando cada una de las respuestas:

- Los campos en los que se divide la dirección para el acceso a la caché.
- El tamaño de la caché, señalando los bytes de datos, los bits asociativos y los comparadores de los que dispone.
- La tasa de aciertos para el código ejecutado
- El estado de la caché tras la ejecución, incluyendo los contadores del algoritmo. Utilizando el esquema facilitado, señalar sólo aquellas entradas que hayan variado durante la ejecución, indicando el valor del índice de la entrada correspondiente y los valores para la etiqueta y contador LRU finales.

3.1.16. El siguiente programa escrito en C se ejecuta en un procesador que dispone de una caché de datos con 8 palabras por bloque (32 bytes) y un tamaño de 256 bytes.

```

int i, j, c, inc, array[512];
...
for ( i=0 ; i<1000; i++)
    for (j=0; j<512 ; j=j+inc)
        c = array[ j ] + 17;
...
    
```

Sabiendo que un entero ocupa una palabra, que la caché está inicialmente vacía y considerando que toda la actividad de la caché se genera en los accesos a memoria de la variable array, se pide **justificando necesariamente la respuesta** cual es la tasa de fallos esperada en el acceso a la caché si:

- La caché es de correspondencia directa (CD) y la variable inc = 256.
- La caché es de correspondencia directa y la variable inc = 255.
- Que ocurre en ambos casos anteriores si la caché es asociativa de 2 vías (CA2V).

Nota: Cualquier respuesta sin su correspondiente explicación no será calificada aunque sea correcta.

ORGANIZACIÓN Y ESTRUCTURA DE LA MEMORIA: CACHÉS Y MEMORIA VIRTUAL

3.1.17. El microprocesador Microblaze es un procesador RISC de 32 bits (palabras, direcciones de memoria y ruta de datos de 32 bits) segmentado con 5 etapas destinado a sistemas empujados basados en lógica programable. Posee una estructura MMU y cache de datos configurable en tamaño. Las características de las caché son:

- Correspondencia directa (Asociativo de una vía).
- Escritura directa (Write Through), con buffer de escritura (en caso de ser necesario).
- Tamaño de la caché y de la etiqueta configurable.
- El tamaño del bloque en caché se puede configurar como 4 ó 8 palabras.

El sistema permite “cachear” un cierto rango de posiciones consecutivas a través de la configuración de dos registros (CACHE_BASEADDR y CACHE_HIGHADDR). El resto de las direcciones quedan sin cachear. Existe un único bit de control que es el de validez que se utiliza tanto en la inicialización como para invalidar un bloque de datos por el software. En caso de generarse una dirección en el espacio “no cacheable” el sistema caché descarta la petición y se trae el dato de la memoria principal sin que la caché actúe.

Suponga que se configura un sistema con una caché de instrucciones que tiene un total de 16KB organizado en bloques de 4 palabras y una caché de datos de 64KB en bloques de 8 palabras y que los registros de direcciones “cacheables” son:

C_ICACHE_BASEADDR= 0x00300000 y C_ICACHE_HIGHADDR=0x0030FFFF (Caché Instrucciones).

C_DCACHE_BASEADDR= 0x00400000 y C_DCACHE_HIGHADDR=0x0043FFFF (Caché Datos).

Se pide:

a) Tamaño de memoria de datos e instrucciones que se han de cachear

b) Considerando sólo el rango de direcciones cacheable, señalar la forma en que se divide la dirección para el acceso a caché de instrucciones.

c) Considerando sólo el rango de direcciones cacheable, señalar la forma en que se divide la dirección para el acceso a caché de Datos.

d) Tamaño total de las memorias caché, cantidad y tamaño de comparadores necesarios.

e) Si no se utiliza la caché de datos el sistema necesita de 3 ciclos de reloj para acceder a la memoria de datos tanto en lectura como escritura. La memoria de instrucciones es independiente (se puede acceder en paralelo) y posee un buffer de pre-captura que hace que el tiempo medio de acceso sea de 1,2 ciclos. Mientras el sistema accede a memoria el pipeline se detiene (*stall*). Suponga además que la cantidad de instrucciones **load** representan el 6 % y las de **store** el 4%. En la figura se observan los ciclos de *stall* que agrega el acceso a la memoria de datos, pero no se muestran las detenciones de la memoria de instrucciones.

	cycle 1	cycle 2	cycle 3	cycle 4	cycle 5	cycle 6	cycle 7	cycle 8	cycle 9
instruction 1	IF	OF	EX	MEM	WB				
instruction 2		IF	OF	EX	MEM	MEM	MEM	WB	
instruction 3			IF	OF	EX	Stall	Stall	MEM	WB

¿Qué CPI posee el sistema sin contar los riesgos de saltos?

f) Suponga que se agrega una caché de instrucciones que tiene un tiempo de acceso en acierto de 1 ciclo y cuando falla, con una probabilidad del 3%, al margen de traer el bloque necesita un ciclo extra para actualizar las etiquetas. El tiempo de lectura-escritura es de 2 ciclos por cada palabra. También se agrega una caché de datos, la que tiene una probabilidad de acierto del 96% que tarda 1 ciclo en caso de acierto y en caso de fallo, tras traer el bloque, usa otro ciclo extra para actualizar las etiquetas. La memoria de datos necesita 3 ciclos para leer ó escribir cada palabra.

¿Qué CPI tiene el sistema ahora?

ORGANIZACIÓN Y ESTRUCTURA DE LA MEMORIA: CACHÉS Y MEMORIA VIRTUAL

3.1.18. Se pretende analizar y comparar el rendimiento de dos procesadores. El PROC_1, es un procesador de emisión simple segmentado en 5 etapas (como el visto en la teoría y práctica) que tiene las siguientes etapas: IF, ID, EX, MEM, WR. Este procesador funciona a una velocidad de reloj de 200 MHz. El camino crítico en este sistema es el acceso a las memorias cachés (IF y MEM).

Para acelerar el cómputo se rediseña la segmentación para dividir los accesos a memoria en dos ciclos. De este modo el nuevo procesador (PROC_2) tiene 7 etapas (IF1, IF2, ID, EX, M1, M2, WR) y funciona a 250 MHz. Con esto el acceso a cachés implica enviar la dirección en el ciclo IF1 y recibir el resultado en el ciclo IF2 y esto se puede producir en cada ciclo de reloj (es decir en caso de acierto en caché no agrega demora). Lo mismo sucede para el acceso a la caché de datos en los ciclos M1, M2.

Considere que los procesadores ejecutan un código con la siguiente distribución en número de instrucciones:

Load	Store	Salto Cond	Salto Incod	Enteros
12%	8%	10%	2%	68%

En ambos procesadores la caché de instrucciones tiene una tasa de fallos de 2% y la de datos del 3%. El movimiento de los bloques de memoria a la caché implica 10 ciclos de reloj para la actualización de etiquetas y otras tareas en la caché de datos y de sólo 6 ciclos en la de instrucciones. Además cada transferencia implica 200 ns de tiempo de transferencia del bloque completo.

Se supone que todos los riesgos de datos se pueden adelantar y/o salvar por medio del compilador y que los riesgos de control en el PROC_1 penalizan 1 ciclo para los saltos incondicionales, 2 ciclos para los saltos efectivos y 0 ciclos para los saltos no efectivos.

Para el procesador segmentado en 7 etapas, en la etapa EX se conoce el resultado de la condición y el cálculo de la dirección se lleva a cabo en ID. El porcentaje de saltos condicionales (vale para ambos procesadores) con salto efectivo es del 33% y la estrategia de predicción de saltos de No efectiva.

- ¿Qué sistema es más rápido y por cuánto? No tenga en cuenta el tiempo de llenado y vaciado del pipeline. Para la justificación de los ciclos de demora en los riesgos de control del PROC_2 utilice necesariamente las tablas adjuntas y responda en el sitio correspondiente.
- Al segundo sistema se le rediseña el acceso a memoria principal agregando un segundo nivel de caché (L2). Con esto, las tasas de fallos locales en la caché de instrucciones son $fi1 = 2\%$ y $fi2=5\%$, y para caché de datos $fd1=3\%$ y $fd2=10\%$ siendo la penalización por fallo en caché de 10 y 60 ciclos respectivamente ($TB1 = 10$ ciclos y $TB2=60$). ¿En qué porcentaje mejora al sistema PROC_2? Responda en el sitio correspondiente.

a. ¿Qué sistema es más rápido y por cuánto? No tenga en cuenta el tiempo de llenado y vaciado del pipeline.

b. Al segundo sistema se le rediseña el acceso a memoria principal agregando un segundo nivel de caché (L2). Con esto, las tasas de fallos locales en la caché de instrucciones son $fi1 = 2\%$ y $fi2=5\%$, y para caché de datos $fd1=3\%$ y $fd2=10\%$ siendo la penalización por fallo en caché de 10 y 60 ciclos respectivamente ($TB1 = 10$ ciclos y $TB2=60$). ¿En qué porcentaje mejora al sistema PROC_2? Responda en el sitio correspondiente.

ORGANIZACIÓN Y ESTRUCTURA DE LA MEMORIA: CACHÉS Y MEMORIA VIRTUAL

3.2.1. Se dispone de un sistema de memoria virtual paginado en dos niveles, en el que para el acceso a memoria se necesitan 4 ciclos, 2 para la traducción de la dirección virtual-real y dos adicionales para el acceso a los datos. En la tabla adjunta se muestra la estadística de uso de un conjunto de instrucciones y los ciclos promedio para la ejecución de cada tipo. De los CPI indicados, 4 ciclos corresponden a la fase de acceso a memoria señaladas anteriormente.

Tipo Instrucción	LOAD/STORE	Salto	Coma Flotante	Enteros
Promedio de uso	15%	10%	30%	45%
CPI	8	6	8	5

Se plantean dos mejoras independientes, la primera consiste en añadir una unidad específica para las instrucciones de coma flotante que implica reducir el tiempo de ejecución a la mitad. La segunda mejora pretende optimizar el acceso a memoria. Para ello se incorpora un TLB y una caché real. Las características de estas dos nuevas unidades son: tiempo de acceso al TLB 1 ciclo y frecuencia de fallos en el mismo del 5%. El acceso a la caché (1 palabra por bloque) es también de 1 ciclo y la frecuencia de fallos del 8%. Cada fallo se penaliza como si no hubiera habido mejora. Se quiere saber utilizando necesariamente la ley de Amdahl:

- a) La mejora obtenida en ambos casos. ¿En cuánto es mejor una que la otra?
- b) ¿Cuál será el rendimiento de la mejora conjunta del sistema?

3.2.2. Suponer que un sistema de memoria virtual tiene páginas de 4kbytes, dispone de un TLB completamente asociativo de 4 entradas como el de la figura. Se pide: **a)** Indicar el tamaño en bytes de la memoria virtual y real y **b)** Convertir la dirección virtual 1234567816 en la dirección real correspondiente.

P.V.	M.P.R.
0FF78	F27
01234	C43
12345	2F0
A5678	0B5

3.2.3. Un determinado sistema permite el acceso virtual a 4 Gbytes de memoria y dispone de un mapa de memoria física de 1 Mbyte. El sistema virtual es paginado puro con tamaño de página de 4 kbytes. Para la traducción de una dirección virtual (DV) a una real (DR), dispone de una unidad TLB, común para código y datos, asociativa de 2 vías, que guarda un total de 32 direcciones virtuales. Se pide:

- a) Los campos en los que se divide la DV para el acceso al TLB. Indicar la estructura y tamaño del TLB. Suponer que no existe ningún bit de control.
- b) Suponiendo el TLB lleno, indicar razonando la respuesta, el número máximo de referencias a memoria que puede generar la CPU sin que se produzca un fallo de página en el TLB.
- c) Suponiendo el TLB vacío, señalar el estado del mismo tras la ejecución del siguiente programa, suponiendo que el bucle se ejecuta cinco veces. En cada caso elegir cualquier dirección real válida.

Dirección	Instrucción	Comentario
00FFF000	CLR A1	; Limpia el contenido del registro A1
00FFF002	L1: LOAD D1, \$30H(A1)	; Lee memoria con direccionamiento relativo a registro
00FFF004	ADD D2, D1	; D1+ D2= D2
00FFF006	INC A1	; A1+ 1 = A1
00FFF008	INC A1	; A1+ 1 = A1
00FFF00A	JMP L1	; Salta a la etiqueta L1

ORGANIZACIÓN Y ESTRUCTURA DE LA MEMORIA: CACHÉS Y MEMORIA VIRTUAL

3.2.4. Sea un sistema de memoria virtual paginado de 256Mbytes, con tamaño de página de 1kbyte. El sistema dispone de un TLB asociativo de 2 vías y un sistema multipaginado subdividido en niveles con tablas de páginas a su vez también paginadas que se encuentra en memoria principal. Calcular los campos en los que se divide la dirección virtual para el acceso a las tablas de páginas si se conoce que el tamaño del descriptor de página para cualquier nivel es de 16 bytes.

3.2.5. Un determinado sistema de memoria dispone de un TLB completamente asociativo con 16 entradas con tamaño de página de 256 bytes. El sistema dispone de una unidad caché de 1 Kbytes asociativa de 4 vías con dos palabras por bloque. El sistema soporta 262.144 (256k) direcciones virtuales. El bus de direcciones y el de datos son ambos de 16 bits. Se pide:

- a) El tamaño del TLB y de la caché cuando la caché es real. Explicar como se obtiene el resultado.
 b) El tamaño del TLB y de la caché cuando la caché es virtual. Explicar como se obtiene el resultado.

NOTA: Considerar en ambos casos que no existen bits de control

3.2.6. En un Pentium se ha desactivado la segmentación y seleccionado un tamaño de página de 4 Kbytes. Su sistema de paginación consta de dos niveles. El primer nivel denominado directorio de páginas tiene 1024 entradas y la dirección base del directorio es la indicada por el registro CR3.

Suponiendo que el contenido en hexadecimal del registro **CR3** es **00033000** y el de algunas posiciones de memoria:

DIRECCION	DATO	DIRECCION	DATO	DIRECCION	DATO
00033000	03307000	0021729C	31320100	003BC618	00218063
00033700	003BC000	0021829C	535309AA	003FF93C	00113000
0011329C	67845260	00337186	003FF6A0	03307186	00217000

- a) ¿De cuantas entradas dispone el segundo nivel (tabla de páginas)? b) Fijada una entrada del primer nivel (directorio de páginas) ¿Cuánta memoria se puede referenciar a partir de ella? Suponiendo que se quiere convertir la dirección virtual **7018629C**, se pide determinar: c) La dirección de memoria correspondiente para la entrada en el directorio de páginas. d) La dirección de memoria donde comienza la tabla de páginas. e) La dirección de memoria correspondiente para la entrada en la tabla de páginas. f) La dirección de memoria del dato referenciado y el valor contenido.

3.2.7. Un sistema ordenador de 32 bits, puede direccionar hasta 64Mbytes de memoria real, dispone además de las siguientes características: Espacio de direcciones virtuales de 1Gbyte con un tamaño de página de 4 Kbytes. TLB unificado completamente asociativo con 40 entradas y un byte de control por entrada. Unidad caché real y unificada de 20 Kbytes de capacidad, es asociativa por conjuntos con 5 vías, cada una con un tamaño de bloque de 32 bytes. Algoritmo de reemplazamiento LRU. Se pide:

- a) Indicar en un esquema los campos de una entrada del TLB, indicando su tamaño. Calcular el tamaño total del TLB.
 b) Indicar en un esquema los campos de una entrada de la unidad caché, indicando su tamaño. Calcular el tamaño total de la caché. No existen bits de control excepto los mínimos necesarios para el algoritmo LRU
 c) Suponga una dirección virtual cualquiera que se encuentra en el TLB, explique como se calcula la correspondiente supuesta dirección real y con la ayuda de un esquema, indique como se accede a cada elemento de la caché, detallando que bits de la dirección se utilizan para extraer la palabra correcta

3.2.8. Se tiene un sistema de memoria virtual de 64 Gbytes y con memoria física 16 veces menor. El sistema dispone de un TLB completamente asociativo con 32 entradas y una caché real asociativa de 4 vías y 256 kbytes de datos. Indicar, justificando brevemente la respuesta, a) el tamaño de página que permita el acceso en paralelo al TLB y a la unidad caché. b) Utilizando el valor anterior, calcular el tamaño en bits del TLB.

ORGANIZACIÓN Y ESTRUCTURA DE LA MEMORIA: CACHÉS Y MEMORIA VIRTUAL

3.2.9. Suponga que conoce los siguientes parámetros de un sistema: Dirección virtual (DV) de 32 bits. Dirección real (DR) de 24 bit. Paginación con tamaño de página de 256 Bytes. TLB completamente asociativo de 4 entradas. 8 KByte de caché real asociativa con 2 vías y guardando 16 bytes por bloque.

Indicar: **a)** Los campos en los que se dividirán la DV para trasladarla a DR y los campos de la DR para acceder a la caché o a memoria indicando su tamaño en bits. **b)** Tamaño del TLB y de la caché. **c)** Supóngase la caché vacía y que el procesador carga en un registro interno el contenido de la DV = $1A23F03E_{16}$. Esta DV se traslada en la DR = $24F03E_{16}$ mediante un acierto en el TLB y se sabe que el contenido de cada dirección de memoria coincide con el valor de los 4 bits menos significativos de la dirección (ejemplo: $F03423_{16}$ contiene 03). Se pide: **c1)** Rellenar el TLB, la caché y la memoria con los valores que corresponden al supuesto anterior, en la situación final, es decir, después de que el procesador haya cargado el registro. **c2)** Sobre el esquema por medio de flechas numeradas, indicar cómo y en qué orden se realizan los accesos a los distintos elementos del sistema.

ORGANIZACIÓN Y ESTRUCTURA DE LA MEMORIA: CACHÉS Y MEMORIA VIRTUAL

3.2.10. Se tiene un sistema de memoria virtual con las siguientes características: Sistema paginado con un TLB completamente asociativo de 4 entradas y una caché real para datos de 8 Kbytes asociativa con 2 vías y guardando 8 bytes por bloque.

En un determinado momento, la CPU solicita leer un byte cuya dirección es $D35F072B_{16}$. Se produce un acierto en el TLB pero un fallo en la caché. El sistema accede a la dirección de memoria $45702B_{16}$ y se actualiza el bloque en la caché desde la cual la CPU puede leer el byte solicitado.

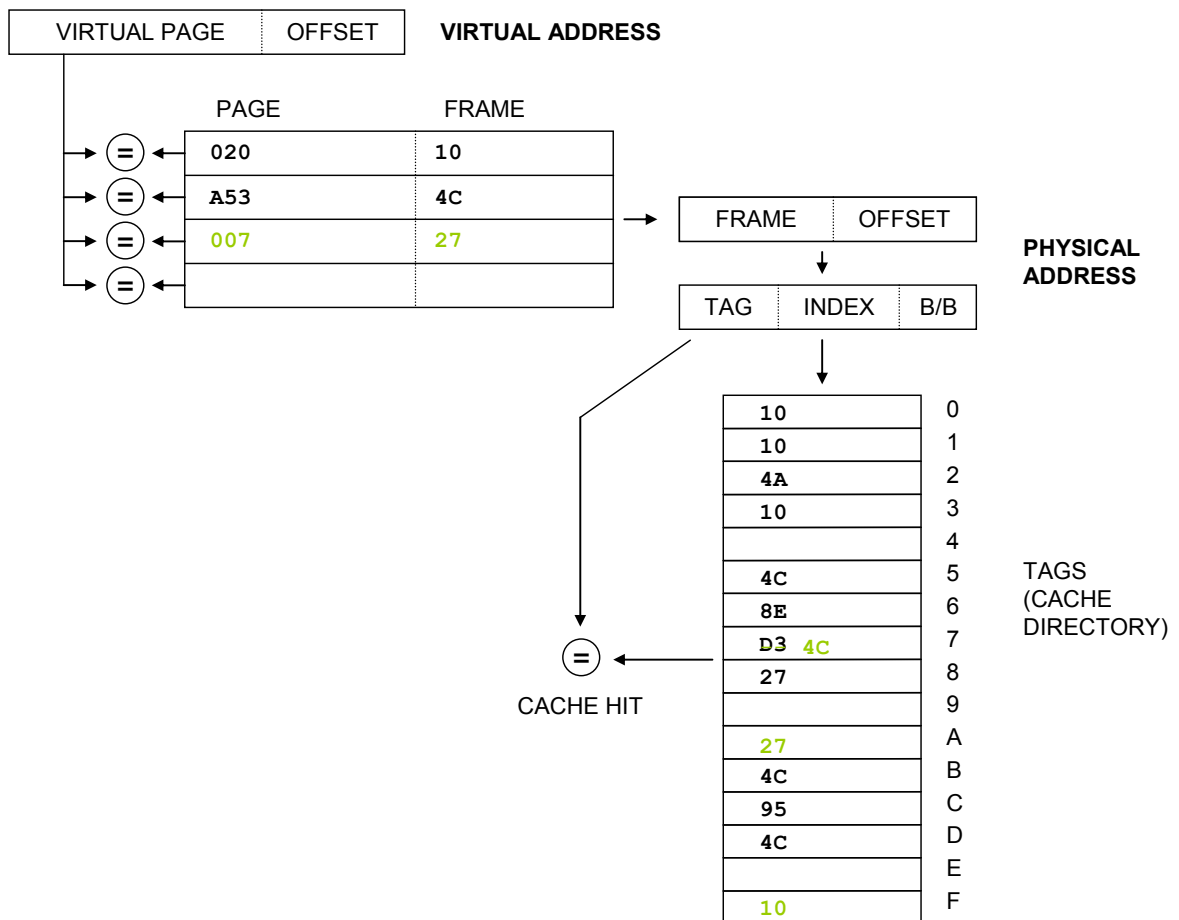
Con esta información se pide contestar, justificando brevemente la respuesta, las siguientes preguntas:

- a) Tamaño en bits de las direcciones virtual y real:
- b) Tamaño máximo de página:
- c) Sin tener en cuenta ningún bit de control, el contenido de los campos de la entrada del TLB en donde se ha producido el acierto.
- d) Sin tener en cuenta ningún bit de control, el contenido de todos los campos en la entrada de la caché donde se ha guardado el nuevo bloque. Se sabe que el contenido de las direcciones de la memoria real contiguas a la solicitada es:

Dirección ($4570..$) ₁₆	..30	..2F	..2E	..2D	..2C	..2B	..2A	..29	..28	..27	..26	..25	..24
Byte guardado:	19	AA	23	45	76	89	F5	3D	66	3D	FA	2C	76

ORGANIZACIÓN Y ESTRUCTURA DE LA MEMORIA: CACHÉS Y MEMORIA VIRTUAL

3.2.11. La siguiente figura muestra un TLB junto a una caché (sólo aparece el directorio caché). Como se puede ver, el TLB tiene 4 entradas y la caché 16. El sistema de memoria usa paginación y tiene las siguientes



características: direccionamiento virtual de 1 MB, direccionamiento real o físico de 64 KB, páginas de 256 bytes. En este momento el TLB y la caché están parcialmente llenos, y se sabe que el rango de direcciones virtuales desde 0x00000 a 0x01FFF se corresponde con el rango de direcciones reales 0x2000 a 0x3FFF.

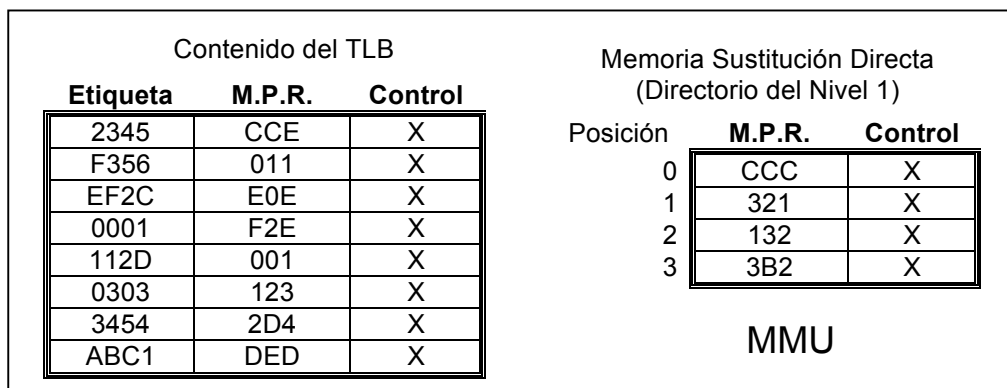
- ¿Qué organización tiene el TLB? ¿Y la caché? ¿Es real o virtual? Justifica tus respuestas.
- ¿Cuál es el tamaño de los campos PÁGINA VIRTUAL, OFFSET, MARCO, TAG, INDEX y B/B?
- Suponiendo que la CPU envía la siguiente secuencia de direcciones (en hexadecimal): 02034 – 020F1 – A5354 – 007AB – 007A3 – 02015 – 0203A – A53B9 – A5377 – 02001. ¿Cuáles provocarán un fallo de página? ¿Y cuáles un fallo en la caché? Escriba sobre la figura los contenidos finales del TLB y del directorio caché, tachando los actuales si tienen que ser reemplazados.

ORGANIZACIÓN Y ESTRUCTURA DE LA MEMORIA: CACHÉS Y MEMORIA VIRTUAL

3.2.12. Se dispone de un ordenador con un procesador cuyas direcciones virtuales son de 24 bits y que dispone de una memoria real de 1 Mbyte. Se opta por un sistema de memoria paginado, con un tamaño de página de 256 bytes y longitud del descriptor de página de 2 bytes, en el que se incluye el marco de página, un bit de presencia y otros bits de control. Como la tabla de páginas es muy grande, ésta a su vez está paginada en tres niveles y sólo se guarda una parte en memoria real (los dos últimos niveles). La unidad de manejo de memoria (MMU) del sistema dispone de una unidad TLB completamente asociativa de 8 entradas. Además del TLB, dispone de una memoria de sustitución directa, que contiene todos los descriptores del primer nivel de las páginas en las que se ha dividido la tabla de páginas. El tamaño de la ruta de datos y por tanto el tamaño de las lecturas es de 16 bits. Cada subtabla de la tabla de páginas multinivel ocupa 1 página. Se pide:

- a) El tamaño del TLB y de la memoria de sustitución directa presente en la MMU, así como la longitud y campos de los descriptores de página en ella guardados.
- b) Indicar los pasos que debe seguir la MMU cuando el procesador solicite las siguientes lecturas: EF2C58₁₆ y CC25B4₁₆ y qué resultado se devolverá al procesador.
- c) ¿Cuántas excepciones por falta de página se pueden producir como máximo en un acceso a memoria en este sistema? Justificar brevemente.

El tamaño de la ruta de datos y por tanto el tamaño de las lecturas es de 16 bits. En la figura se representa el contenido de la MMU y de algunas posiciones de memoria relevantes para el problema. Los datos están organizados "big endian" (byte de mayor peso en dirección más baja). La información del marco está en los bits de mayor peso de las palabras. Suponer que la página que se busca está presente.



Contenido de la Memoria Principal

Pos(hex)	Val(H)	Pos(hex)	Val(H)	Pos(hex)	Val(H)	Pos(hex)	Val(H)
00000	XX	1B001	A2	55C4A	CC	CCDB6	56
...	...	1B002	AA	55C4B	DD	CCDB7	78
00100	12	1B003	22
00101	13	8010F	45	DED09	40
00102	14	3B22E	31	80110	67	DED0A	20
00103	24	3B22F	32	80111	88	DED0B	06
...	...	3B230	55	80112	98
01130	22	3B231	CF	E0E58	22
01131	23	3B232	32	CCD57	01	E0E59	DE
01132	24	CCD58	33	E0E5A	D3
01133	25	55400	22	CCD59	32
...	...	55401	23	CCD60	56	F1A0B	BB
132D4	E0	55402	24	CCD61	78	F1A0C	DE
132D5	E3
...	...	55C48	CC	CCDB4	20	FFFFE	XX
1B000	A0	55C49	CC	CCDB5	05	FFFFF	XX

3.2.13. Se dispone de un ordenador con un procesador cuyas direcciones virtuales son de 48 bits y que dispone de una memoria real de 4 Gbyte. Se opta por un sistema de memoria paginado, con un tamaño de página de 64 Kbytes y longitud del descriptor de página de 4 bytes en el que se incluye, el marco de página, un bit de presencia y varios bits de control más. Como la tabla de páginas es muy grande, ésta a su vez está paginada en tres niveles y sólo se guarda una parte en memoria real (los dos últimos niveles). La unidad de manejo de memoria (MMU) del sistema dispone de una unidad TLB 4-asociativo de 128 posiciones. Además del TLB, dispone de una memoria de sustitución directa, que contiene todos los descriptores del primer nivel de las páginas en las que se ha dividido la tabla de páginas. Se pide, justificando la respuesta, señalar:

- a) Campos en que se divide la dirección virtual para el acceso al TLB y para el acceso a las tablas, la dirección real, los campos del TLB y los descriptores de página
- b) El tamaño del TLB y de la memoria de sustitución directa presente en la MMU

ORGANIZACIÓN Y ESTRUCTURA DE LA MEMORIA: CACHÉS Y MEMORIA VIRTUAL

- c) El tamaño total de las tablas de páginas (solo nivel 2 y nivel 3). Si todas las páginas estuviesen en memoria que porcentaje representan respecto del total de memoria.
- d) Cuantos accesos a memoria principal se deben realizar para obtener un dato, si la dirección no está en el TLB

3.2.14. Se dispone de un sistema de memoria virtual de 16 MBytes, con una memoria real de hasta 1 MByte. El sistema es paginado con tamaño de página de 4 kBytes y cuenta con los siguientes elementos: caché virtual unificada, asociativa de 4 vías, con capacidad para guardar un total de 64 bloques de 32 bytes cada uno. En la MMU se dispone de un TLB completamente asociativo con 8 entradas. Se muestran algunas de las entradas en el estado actual de la caché, el TLB y parte de la memoria real. Se pide, tras ejecutar la secuencia de referencias a memoria siguiente: 4F20AE -- F50D80 – AA45F0 – F50D9C.

- a) El tamaño total de la caché y del TLB, señalando en cada caso el número de bits de memoria que se necesitan y el número y tamaño de los comparadores. Para ello se ignorarán los posibles bits de control.
- b) Teniendo en cuenta que los bloques LRU son los contenidos en la VIA 1 de la caché, se pide el estado final de la caché (señalar sólo aquellos casos en los que se cambia el contenido).
- c) Indicar, justificando la respuesta, la estrategia elegida por la caché para las escrituras.

Nota: Para facilitar la lectura, en la etiqueta se utiliza una notación "mixta" con dígitos en hexadecimal (negrita) y en binario. Para los bloques sólo se indica a la izquierda el byte con la dirección de memoria más alta y a la derecha el de dirección de memoria más baja de los 32 bytes que contiene un bloque.

	VIA 1		VIA 2	VIA 3	VIA 4
0	AA4 000	FF 33	----	----	----
--	-----	-----	----	----	----
--	-----	-----	----	----	----
5	4F2 000	4C AA	----	----	----
--	-----	-----	----	----	----
--	-----	-----	----	----	----
C	55F 000	22 55	----	----	----
--	-----	-----	----	----	----
--	-----	-----	----	----	----
F	100 101	D3 C4	----	----	----

TLB	
AA4	CC
4F2	F0
324	D5
55F	35
DF0	C7
F50	50
AAA	10
100	2C

MEMORIA									
DIRECC.	DATO	DIRECC.	DATO	DIRECC.	DATO	DIRECC.	DATO	DIRECC.	DATO
10020	C4	35180	FF	50D80	44	CC5E0	66	F00A0	AA
10021	2F	35181	44	50D81	5C	CC5E1	42	F00A1	FC
----	----	----	----	----	----	----	----	----	----
----	----	----	----	----	----	----	----	----	----
1003E	00	3519E	33	50D9E	32	CC5FE	D3	F00BE	00
1003F	3F	3519F	22	50D9F	F4	CC5FF	F5	F00BF	4C

ORGANIZACIÓN Y ESTRUCTURA DE LA MEMORIA: CACHÉS Y MEMORIA VIRTUAL

3.2.15. Se tienen dos sistemas RISC diseñados con el mismo sistema de memoria, cuyas principales características son: Memoria virtual de 64Gbytes y real de 4 Gbytes. Tamaño de página 2 kbytes. Bus de datos de 32 bits. TLB con un total de 512 entradas repartidas en una estructura asociativa de 2 vías. Caché real unificada de 256 kbytes asociativa de 2 vías, con 8 palabras/bloque.

La diferencia principal entre ambas máquinas se basa en que la denominada MAQ1 dispone de un sistema de ventanas de registros tipo SPARC, mientras que la denominada MAQ2 no. La frecuencia de reloj en ambos sistemas es la misma, así como el porcentaje medio de instrucciones que junto con el coste en ciclos para cada tipo se muestran en la tabla adjunta:

Load/Store	Salto	Llamadas a rutinas	Oper. ALU
33%	15%	2%	50%
1,2 ciclos	1,4 ciclos	1,4 ciclos	1,0 ciclos

Por tener distinto sistema de registros, la diferencia entre MAQ1 y MAQ2 viene dada las diferentes operaciones adicionales que se deben realizar para guardar y restaurar datos en las llamadas a rutinas y su posterior retorno.

* En la MAQ1, gracias al sistema de ventanas de registros, sólo se producen operaciones adicionales cuando se produce un *overflow* por la limitación en el número de ventanas del sistema. El *overflow* ocurre un 5% de las llamadas a rutina e implica una penalización de 50 instrucciones del tipo Load/Store, 10 Saltos y 20 operaciones con la ALU.

* En cada llamada a rutina, la MAQ2 necesita ejecutar un 7% adicional en operaciones del tipo Load/Store.

a) Indicar los campos en los que se divide la dirección virtual para el acceso al TLB y la dirección real para el acceso a la caché.

b) Suponiendo un sistema de memoria perfecto, se pide el CPI para cada máquina indicando cual de las dos es mas eficiente y en cuanto. Se ha analizado un determinado programa que utiliza un 40% de instrucciones con memoria. Se ha medido una tasa de fallos del 0,1% en el TLB y se quiere conocer el efecto de variar el tamaño del bloque de la caché. Se dispone de la siguiente información:

Tamaño Bloque (bytes)	16	32	64
Tasa de fallos	1,5%	1,0%	0,8%

Se sabe que la penalización por fallo en el TLB es de 25 ciclos y por fallo en la caché es de 6 ciclos más un ciclo adicional por cada palabra en el bloque. Se pide:

c) ¿Cuál es el tamaño óptimo de bloque que minimiza las pérdidas por accesos a memoria?

d) Para la MAQ1 la pérdida de rendimiento en el sistema al tener en cuenta el sistema real de memoria

3.2.16. Sea un sistema de memoria virtual, que opera con instrucciones y datos de 32 bits. El sistema dispone de una unidad TLB, completamente asociativa de 8 entradas y de una unidad caché virtual unificada, asociativa de dos vías, con 16 entradas por vía, 16 bytes por bloque y un algoritmo LRU para las sustituciones. Se facilitan algunos contenidos relevantes en el estado inicial del TLB, de la caché y de memoria. A continuación, se ejecuta el código adjunto:

NOTA: sólo en algunas ocasiones, el contenido facilitado de la caché es relevante.

PC	Instrucción	Comentario
0A040	MOV R4, 0	; R4 <= 0
0A044	MOV R1, 2	; R1 <= 2
0A048	MOV R2, A305Ch	; R2 <= A305C ₁₆
0A04C	L1: LOAD R3, R2, R0	; R3 <= MEM[R2+0]
0A050	ADD R4, R4, R3	; R4 <= R4 + R3
0A054	ADD R2, R2, 4	; R2 <= R2 + 4
0A058	ADD R1, R1, -1	; R1 <= R1 - 1
0A05C	BNZ L1	; Salta a L1 si Z = 0
0A060	JMP FIN	; Salta a FIN

0A0E0	FIN: STORE R4, R2, A0h	; R4 => MEM[R2 + A0 ₁₆]
0A0E4	HALT	

TLB	
FC5	02
A30	B0
0BC	F5
0A0	3C

UNIDAD CACHE					
VIA 1			VIA 2		
0 =>	032	B0030200 A02A305C A0300002 A0400000	BC0	A0A2D4D5 88AA3460 45803D59 95203580	
1 =>					
4 =>	0A0	0020300B C503A02A 2000001A 00000004A	3F2	A0A3D420 40FFDC60 458032FF 45203580	
5 =>	4B2		A30	10102020 30405060 708090A0 B0C0D0E0	
6 =>	A03		ABC		
E =>	0A0	10102020 30405060 708090A0 5060D0E0	3F0	A0A3D420 40FFDC60 458032FF 45203580	
F =>	3FF	A0A3D420 40FFDC60 458032FF 45203580	4C0		

ORGANIZACIÓN Y ESTRUCTURA DE LA MEMORIA: CACHÉS Y MEMORIA VIRTUAL

MEMORIA															
DIR	VAL	DIR	VAL	DIR	VAL	-	DIR	VAL	-	DIR	VAL	DIR	VAL	DIR	VAL
3C50	50	3C58	51	3C60	B0		A030	F4		B058	10	B060	40	B068	20
3C51	04	3C59	01	3C61	0F		A031	DE		B059	20	B061	40	B069	40
3C52	04	3C5A	01	3C62	FF		A032	C0		B05A	30	B062	A0	B06A	3F
3C53	03	3C5B	FF	3C63	80		A033	0A		B05B	40	B063	A0	B06B	20
3C54	51	3C5C	F0	3C64	3F		A034	45		B05C	50	B064	C0	B06C	4B
3C55	02	3C5D	0F	3C65	40		A035	74		B05D	60	B065	F5	B06D	BC
3C56	02	3C5E	FF	3C66	55		A036	22		B05E	70	B066	D6	B06E	D0
3C57	04	3C5F	F6	3C67	D0		A037	80		B05F	80	B067	FF	B06F	00

Con la información facilitada, se pide:

a) Justificando brevemente la respuesta, el tamaño de página así como el de la memoria virtual y real.

--

b) Justificando brevemente la respuesta, el porcentaje de aciertos al ejecutar el código completo

c) Sabiendo que en la caché el byte más a la derecha del bloque se corresponde con la dirección más baja de memoria y que se utiliza notación “*big endian*”, señalar en la tabla adjunta el valor en hexadecimal de los registros R2, R3 y R4, tras la ejecución del código.

RESULTADO	
Valor de R2:	
Valor de R3:	
Valor de R4:	

ORGANIZACIÓN Y ESTRUCTURA DE LA MEMORIA: CACHÉS Y MEMORIA VIRTUAL

3.2.17. El ordenador educativo ARQUI01 dispone de un procesador cuyas direcciones virtuales son de 20 bits y dispone de una memoria real de 64 kBytes. Posee un sistema de memoria paginado, con un tamaño de página de 256 bytes y longitud del descriptor de página de 4 Bytes en el que se incluye el marco de página, bit de presencia y varios bits de control. Como la tabla de páginas es muy grande, ésta a su vez está paginada en dos niveles (el primero siempre está presente en la memoria principal). La unidad de manejo de memoria (MMU) del sistema dispone de un TLB completamente asociativo de 8 entradas. El tamaño de la ruta de datos y, por tanto, el tamaño de las lecturas es de 16 bits. El sistema posee además caché virtual unificada asociativa de 4 vías, 16 bloques en total y 2 palabras por bloque.

En las figuras se representa el contenido del TLB, el contenido de la caché y de algunas posiciones de memoria relevantes para el problema. Los datos están organizados "little endian" (byte de menor peso en dirección más baja). La información del marco está en los bits de menor peso de las 2 palabras. Suponer que la página que se busca está presente. Se pide:

- a) Cómo se divide la dirección para el acceso a caché, TLB y memoria (rellenar las plantillas). Indicar el tamaño del TLB y de la caché. Expresar tanto el tamaño de datos como el tamaño total y cantidad de comparadores en el caso de la memoria caché.
- b) Indicar los pasos que se han de seguir cuando el procesador genere las direcciones E2C58₁₆, CC2BC₁₆ y 1C2B2₁₆ y qué resultado se devolverá al procesador.
- c) Indique (en la tabla adjunta de la próxima página) en qué posiciones de la memoria caché habrá actualizaciones para los accesos a memoria del apartado b). Suponga que la vía LRU para todos los casos es la vía 4. No es necesario indicar en las tablas la actualización de la caché, sólo en qué posiciones.

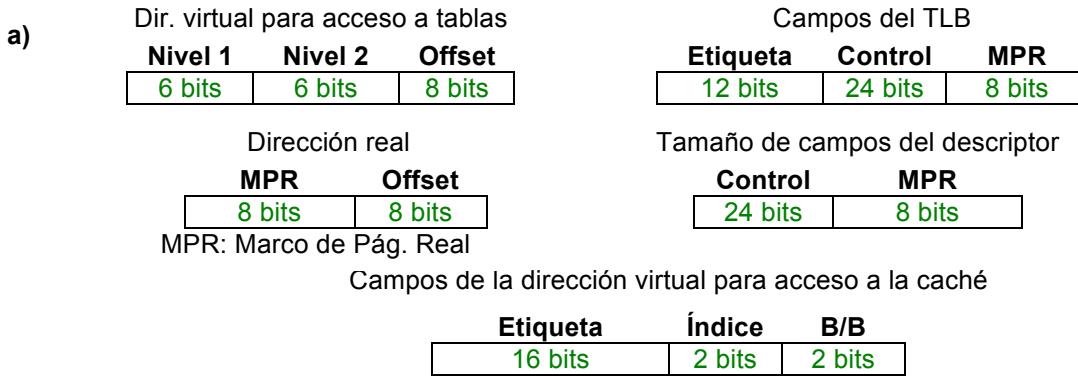
Contenido del TLB			Caché Virtual Unificada							
Etiqu.	Ctrl	MPR	Vía 1		Vía 2		Vía 3		Vía 4	
Etiqu.	Datos	Etiqu.	Datos	Etiqu.	Datos	Etiqu.	Datos	Etiqu.	Datos	
245	X...X	CE								
F56	X...X	22	0	CCCC ABCD EF12	CCCC ABDC EF21	CDCC ABCE EF13	CCFC ABEC EF31			
E2C	X...X	01	1	CCCC ABCD EF14	CCCC ABDC EF23	CDCC ABCE EF15	CCFC ABEC EF33			
001	X...X	2E	2	A132 2007 2006	E2C5 2008 2009	31B2 EABC 1ECD	1132 ABCD EF12			
11D	X...X	01	3	3B21 5E83 197F	1234 FA19 38B4	1C2B 29D3 8930	13B2 3830 5740			
CC2	X...X	2B								
344	X...X	D4								
AB1	X...X	ED								

Base 1er nivel de páginas: 5500 hex

Contenido de la Memoria Principal

Pos(hex)	Val(H)	Pos(hex)	Val(H)	Pos(hex)	Val(H)	Pos(hex)	Val(H)
0000	XX	2BBD	A2	5C4A	CC	CDB6	56
...	...	2BBE	AA	5C4B	DD	CDB7	78
0100	12	2BBF	22
0101	13	810F	45	DE09	40
0102	14	3B2E	31	8110	67	DE0A	20
0103	24	3B2F	32	8111	88
...	...	3B30	55	8112	98	E0B0	21
0158	09	3B31	CF	E0B1	EE
0159	20	3B32	32	C857	01	E0B2	D0
015A	08			C858	33	E0B3	DE
015B	20	551C	CC
...	...	551D	22	CC08	E0	F1AB	BB
13D4	E0	551E	24	CC09	E1	F1AC	DE
13D5	E3	551F	DD
...	CDB4	20	FFFE	XX
2BBC	A0	5C49	CC	CDB5	05	FFFF	XX

ORGANIZACIÓN Y ESTRUCTURA DE LA MEMORIA: CACHÉS Y MEMORIA VIRTUAL



Breve justificación y tamaños TLB y caché:

DV: 20 bits; DR: 16 bits; Páginas: 256 Bytes = 2^8 Bytes => 8 bits de offset

En las tablas de páginas alojadas en páginas se pueden guardar $256/4 = 64 = 2^6$ descriptores

Si la tabla tiene 2 niveles: 20 (total) = 6 (nivel1) + 6 (nivel2) + 8 (offset)

Acceso a la caché 20 bits; 2 Byte en Bloque (2 palabras de 2 bytes cada una); 2 bits de índice ya que hay 4 bloques por vía (16 bloques/4 vías), Etiqueta: $20-2-2 = 16$ bits

Tamaño total TLB: $8 \times (12 \text{ bits etiquetas} + 32 \text{ bits descriptor}) = 352 \text{ bits} = 44 \text{ Bytes}$

Tamaño caché, parte de datos: $4 \text{ vías} \times 4 \text{ bloques/vía} \times 2 \text{ palabras/bloque} \times 2 \text{ Bytes/pal} = 64 \text{ Bytes}$

Tamaño total caché: $4 \text{ vías} \times 4 \text{ bloques/vía} \times (2 \text{ Bytes de etiqueta} + 4 \text{ Bytes datos}) = 96 \text{ Bytes}$

Comparadores caché: 4 comparadores de 16 bits

b) Indicar los pasos que se han de seguir cuando el procesador genere las direcciones $E2C58_{16}$, $CC2BC_{16}$ y $1C2B2_{16}$ y qué resultado se devolverá al procesador.

b.1) Dir Virtual: $E2C58_{16}$; Búsqueda en caché: Etiq: $E2C5$ con índice 10_2 , acierto en vía 2; Dato Obtenido: 2009_{16} ya que el B/B es 00 (parte baja).

b.2) Dir Virtual: $CC2BC_{16}$; Índice para acceso a caché = 11_2 ; Fallo en caché;

Acceso al TLB: Etiqueta: CC2; acierto en TLB, marco 2B; luego la DR= $2BBC$

Dato Obtenido: **A2A0**₁₆ Direcciones (2BBD y 2BBC). Se carga en caché entrada 11_2 las posiciones de memoria 2BBF-2BBE-2BBD-2BBC (22AA A2A0) con etiqueta CC2B

b.3) Dir Virtual: $1C2B2$; índice para acceso a caché: = 00_2 ; Fallo en caché;

Acceso al TLB: Etiqueta: 1C2; Fallo en TLB

Búsqueda en la tabla: #pág en binario 0001 1100 0010 => índice L1 = $000111_2 = 7_{16}$; índice L2 = $000010_2 = 2_{16}$;

Luego si la base de página es 5500, la entrada en la tabla de 1er nivel es: $5500_{16} + 7_{16} * 4 = 551C_{16}$. Contenido DD242CC. Es decir, el marco es CC.

El acceso al segundo nivel es $CC00_{16} + 2_{16} * 4 = CC08_{16}$. Siendo el marco buscado E0. Luego DR= $E0B2$.

Dato Obtenido: **DED0**₁₆ Direcciones (E0B3 y E0B2). Se actualiza caché con las direcciones (E0B3, E0B2 E0B1 y E0B0). DED0 EE21, etiqueta 1C2B y TLB con etiqueta 1C2 y marco E0.

c) Indique (en la tabla adjunta a continuación) en qué posiciones de la memoria caché habrá actualizaciones para los accesos a memoria del apartado b). Suponga que la vía LRU para todos los casos es la vía 4. No es necesario indicar en las tablas la actualización de la caché, sólo en qué posiciones.

	Vía 1		Vía 2		Vía 3		Vía 4	
	Etiq	Datos	Etiq	Datos	Etiq	Datos	Etiq	Datos
0							1C2B	DED0 EE21
1								
2								
3							CC2B	22AA A2A0

- b.1 No modifica caché.
- b.2. Se carga en caché entrada 11_2 las posiciones de memoria 2BBF-2BBE-2BBD-2BBC (22AA A2A0), como la LRU es la vía 4 se coloca en ella. La etiqueta es CC2B.
- b.3. Se actualiza caché con las direcciones (E0B3, E0B2, E0B1 y E0B0). DED0 EE21, etiqueta 1C2B.

ORGANIZACIÓN Y ESTRUCTURA DE LA MEMORIA: CACHÉS Y MEMORIA VIRTUAL

3.2.18. Se dispone de un procesador con memoria virtual capaz de direccionar hasta 16 MBytes, mientras que la memoria física puede ser de hasta 1 MByte. El sistema es paginado puro con tamaño de página de 256 Bytes, siendo el tamaño de palabra de 2 Bytes. Se dispone de un TLB completamente asociativo de 8 entradas y de caché de dos niveles (L1 y L2), tratándose en ambos casos de caché real unificada. La caché de nivel 1 (L1) es asociativa de 4 vías, con 64 bloques por vía y tamaño de bloque 4 Bytes. Por su parte, la caché de nivel 2 (L2) es de correspondencia directa con 256 bloques de 16 Bytes cada uno.

Se muestra parcialmente el contenido del TLB, las dos cachés y la memoria real. Los contenidos no mostrados de las cachés se considerarán como fallos en caso de acceso.

Nota: para facilitar la lectura se utiliza una notación "mixta" con dígitos en hexadecimal (negrita) y en binario. Se representa siempre a la izquierda el byte más significativo y el sistema es *little endian* (byte menos significativo en la dirección más baja). No se muestran en ningún caso los bits de control.

Caché L1					
Índice	VÍA 1		VÍA 2	VÍA 3	VÍA 4
	Etiqueta	Bloque			
000000	C34	00010203	----	----	----
--	----	----	----	----	----
000101	A3F	8371AB04	----	----	----
--	----	----	----	----	----
010101	1C4	E739010C	----	----	----
--	----	----	----	----	----
100001	476	020D6593	----	----	----
--	----	----	----	----	----
111111	100	D3838FC4	----	----	----

TLB	
Nº pág.	Marco
AA40	C30
73BA	2D5
0302	D58
4F23	1C4
1DF0	0B3
F250	A50
C340	3E6
1040	C36

Caché L2		
Índice	Etiqueta	Bloque
00	C3	00010203 3764FC39 8371AC05 E739010D
--	----	-----
45	1C	3764DC39 8371AB04 E739010C EAF0384
--	----	-----
58	2D	E739010C 02040608 4285A8F0 940D9438
--	----	-----
60	C3	12983476 94E8C385 020D6593 00010203
--	----	-----
FF	10	3764DD39 D3838EC4 3764DC3A E739020D

MEMORIA									
DIRECC.	DATO	DIRECC.	DATO	DIRECC.	DATO	DIRECC.	DATO	DIRECC.	DATO
----	--	----	--	----	--	----	--	----	--
100FC	C4	2D586	85	3E600	09	C3000	0D	C3600	03
100FD	8F	2D587	42	3E601	01	C3001	01	C3600	02
----	--	----	--	----	--	----	--	----	--
1C454	0C	34060	33	50D9E	32	C3008	39	A3F140	04
1C455	01	34061	22	50D9F	F4	C3009	FC	A3F141	AB
----	--	----	--	----	--	----	--	----	--

a) Rellenar el número de bits utilizados para cada campo en los siguientes casos.

<table border="1" style="width: 100%;"> <tr> <td style="width: 50%;"></td> <td style="width: 50%; text-align: center;">Nº pág.</td> <td style="width: 50%;"></td> <td style="width: 50%; text-align: center;">Offset</td> </tr> <tr> <td style="text-align: right;">Direc. Virtual</td> <td style="text-align: center;">16</td> <td style="text-align: center;">8</td> <td style="text-align: center;">8</td> </tr> </table>		Nº pág.		Offset	Direc. Virtual	16	8	8	<table border="1" style="width: 100%;"> <tr> <td style="width: 50%;"></td> <td style="width: 50%; text-align: center;">Marco</td> <td style="width: 50%;"></td> <td style="width: 50%; text-align: center;">Offset</td> </tr> <tr> <td style="text-align: right;">Direc. Real</td> <td style="text-align: center;">12</td> <td style="text-align: center;">8</td> <td style="text-align: center;">8</td> </tr> </table>		Marco		Offset	Direc. Real	12	8	8
	Nº pág.		Offset														
Direc. Virtual	16	8	8														
	Marco		Offset														
Direc. Real	12	8	8														
<table border="1" style="width: 100%;"> <tr> <td style="width: 33%;"></td> <td style="width: 33%; text-align: center;">Etiqueta</td> <td style="width: 33%; text-align: center;">Índice</td> <td style="width: 33%; text-align: center;">B/B</td> </tr> <tr> <td style="text-align: right;">Caché L1</td> <td style="text-align: center;">12</td> <td style="text-align: center;">6</td> <td style="text-align: center;">2</td> </tr> </table>		Etiqueta	Índice	B/B	Caché L1	12	6	2	<table border="1" style="width: 100%;"> <tr> <td style="width: 33%;"></td> <td style="width: 33%; text-align: center;">Etiqueta</td> <td style="width: 33%; text-align: center;">Índice</td> <td style="width: 33%; text-align: center;">B/B</td> </tr> <tr> <td style="text-align: right;">Caché L2</td> <td style="text-align: center;">8</td> <td style="text-align: center;">8</td> <td style="text-align: center;">4</td> </tr> </table>		Etiqueta	Índice	B/B	Caché L2	8	8	4
	Etiqueta	Índice	B/B														
Caché L1	12	6	2														
	Etiqueta	Índice	B/B														
Caché L2	8	8	4														

b) Se utiliza una tabla de páginas en un solo nivel, con descriptores de 2 Bytes. ¿Cuánto ocuparía la tabla en caso de estar completa?

Hay un total de 2^{16} páginas virtuales distintas. Para cada página posible se necesitan 2 Bytes (descriptor), así que ocuparía un total de 2^{17} Bytes, es decir, 128 kBytes (una octava parte de la máxima memoria real disponible en este micro).

ORGANIZACIÓN Y ESTRUCTURA DE LA MEMORIA: CACHÉS Y MEMORIA VIRTUAL

c) ¿Cuál sería el dato devuelto para cada una de las siguientes direcciones virtuales? **4F2354** – **73BA86** – **C34000**. Justifique brevemente la respuesta, indicando si hay éxito o fallo en TLB, L1 y L2.

- Dirección virtual **4F2354**. Acierto en TLB (**4F23** se traduce por **1C4**). Dirección real: **1C454**. Acceso a L1: etiqueta **1C4**, índice 010101 y B/B 00. Acierto en L1, palabra devuelta **010C**.
- Dirección virtual **73BA86**. Acierto en TLB (**73BA** se traduce por **2D5**). Dirección real: **2D586**. Acceso a L1: etiqueta **2D5**, índice 100001 y B/B 10. Fallo en L1. Acceso a L2: etiqueta **2D**, índice **58**, B/B **6**. Acierto en L2, palabra devuelta **4285**.
- Dirección virtual **C34000**. Acierto en TLB (**C340** se traduce por **3E6**). Dirección real: **3E600**. Acceso a L1: etiqueta **3E6**, índice 000000 y B/B 00. Fallo en L1. Acceso a L2: etiqueta **3E**, índice **60**, B/B **0**. Fallo en L2. Acceso a memoria con dirección **3E600**, palabra devuelta **0109**.

d) Suponiendo que el acceso a MMU se da siempre antes de empezar el acceso a cachés, se pide calcular el tiempo medio de acceso a memoria (en ciclos) con los siguientes datos. Tiempo de acceso a TLB $t_{TLB}=1$, tiempo de acceso a tablas $t_{tablas}=15$, tiempo de acceso a caché L1 $t_{L1}=1$, tiempo de acceso a caché L2 $t_{L2}=3$, tiempo de acceso a memoria $t_{mem}=10$. Los buses entre L1 y L2, y entre L2 y memoria son ambos de 4 Bytes de ancho, y el tiempo de acceso incluye acceder a todo el ancho de bus. La tasa de acierto local en L1 es del 90%, en L2 del 80% y en TLB del 98%, y se supone que nunca se falla en tablas de páginas.

Si se falla en L1, traer un bloque a L1 desde L2 requiere un único acceso (4 Bytes). Por tanto, $t_{B1}=t_{L2}=3$. Pero si se falla en L2, traer un bloque a L2 desde memoria requiere 4 accesos (16 Bytes), por lo que esta penalización es de $t_{B2}=4 \cdot t_{mem}=40$ ciclos.

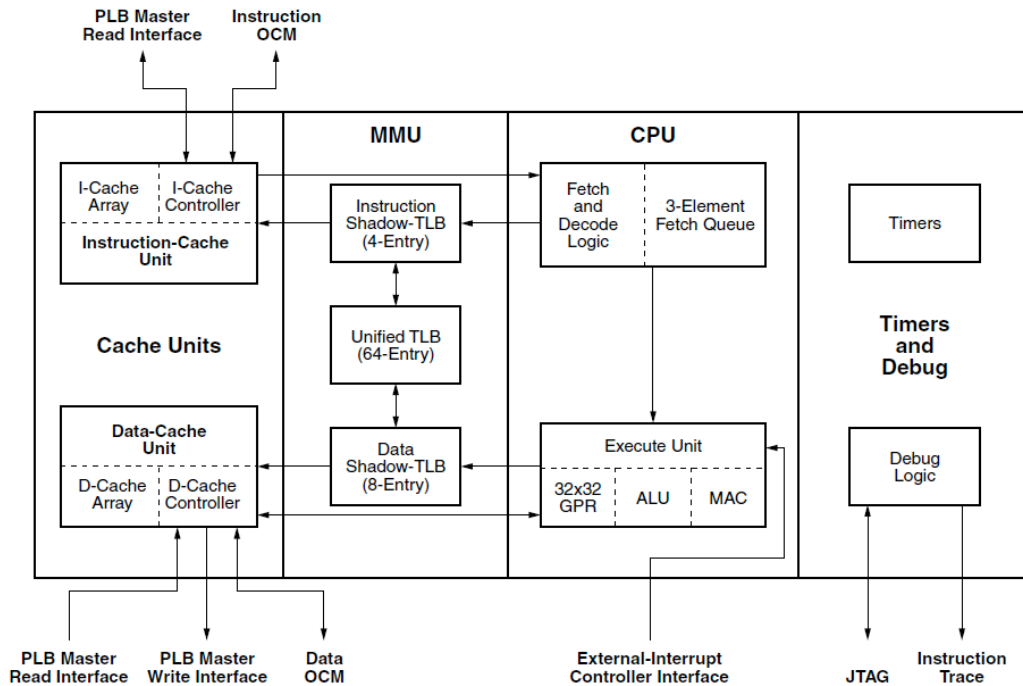
$$t_{acc} = t_{MMU} + t_{dato} = (t_{TLB} + f_{TLB} \cdot t_{tablas}) + (t_{L1} + f_{L1} \cdot (t_{L2} + f_{L2} \cdot 4 \cdot t_{mem})) = (1 + 0,02 \cdot 15) + (1 + 0,1 \cdot (3 + 0,2 \cdot 4 \cdot 10)) = 1,3 + 2,1 = 3,4 \text{ ciclos.}$$

e) Se mejora la integración de caché y TLB, de tal forma que se comience el acceso a caché buscando la etiqueta sin tener la traducción todavía realizada, siempre que esto sea posible. ¿Qué caché(s) se puede(n) integrar de esta forma y por qué?

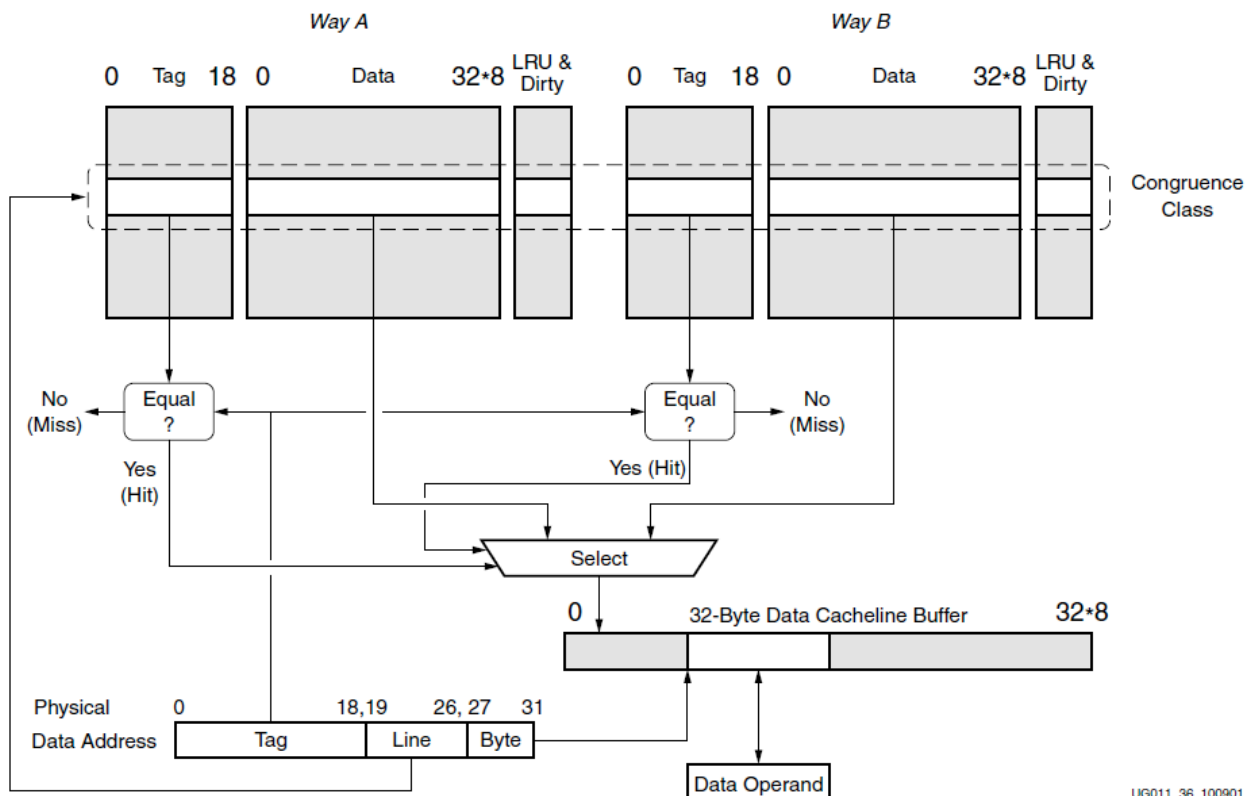
La caché L1 sí se puede integrar de esta forma, ya que índice y B/B son parte del offset, que no hace falta traducir. Por tanto, mientras se busca el marco con el número de página de la dirección virtual, el offset de la dirección virtual (que coincide con el de la dirección real) ya se puede ir utilizando en L1 para buscar la etiqueta. Luego sólo quedará comparar (en caso de acierto en ambas, TLB y caché L1) si el marco coincide con la etiqueta de la caché.

La caché L2 no se puede integrar de esta forma, ya que parte de su índice pertenece al offset y parte al marco, que sólo se conocerá tras realizar la traducción de dirección virtual a real.

3.2.19. PowerPC 405 es un microprocesador muy usado en sistemas empotrados. A continuación se muestra su arquitectura:



PLB y OCM son los buses con los que el microprocesador se conecta con las memorias y los periféricos, y JTAG e *Instruction Trace* son dos interfaces que se usan para depuración de código. Ambas cachés son similares, teniendo esta arquitectura la de datos:



UG011_38_100901

Tenga en cuenta que es un procesador big endian, por lo que en la figura el bit 0 es el MSB y el bit 31 es el LSB. Finalmente, la MMU se basa en un TLB gestionado por software. El sistema operativo añade hasta 64 entradas al TLB usando unas instrucciones especiales del procesador (p.ej. *tlbwe*, *TLB Write Entry*). Si hay fallo

ORGANIZACIÓN Y ESTRUCTURA DE LA MEMORIA: CACHÉS Y MEMORIA VIRTUAL

en el TLB se produce una excepción, que es gestionada por el sistema operativo añadiendo una entrada al TLB para resolver el fallo de página que acaba de ocurrir.

El TLB se compone de tres componentes, Instruction Shadow-TLB, Data Shadow-TLB y Unified TLB, todos completamente asociativos. Si un acceso tanto de instrucciones como de datos tiene acierto en su correspondiente Shadow-TLB, no hay penalización. Si no, se hace la búsqueda en el TLB unificado, con una penalización de 3 ciclos en el caso de acceso de datos o 4 ciclos si es una lectura de instrucción. Si se produce un fallo en el TLB unificado salta una excepción, cuya rutina de atención tarda en media 3500 ciclos en ejecutarse.

a) Responda RAZONADAMENTE a las siguientes preguntas, usando exclusivamente el espacio reservado debajo de cada una

¿La arquitectura del procesador es Harvard o Von Neumann?

En la figura superior de la página anterior se observa con claridad que hay dos caminos diferentes para instrucciones y datos, con buses separados. Es por tanto una arquitectura Harvard.

¿La caché es real o virtual?

La caché es real (de direcciones reales) porque trabaja con direcciones reales, como se puede ver en la figura superior se encuentra después de la MMU, y además en la figura inferior se indica claramente "Physical Data Address"

¿Cuál es el tamaño máximo de la memoria física de datos?

Como se puede ver en la figura inferior, las direcciones físicas de datos son de 32 bits, por lo que puede haber hasta 4 GB de memoria física de datos

¿La caché de datos es de escritura directa o de post-escritura?

La caché es de post-escritura, porque en la figura inferior puede verse que tiene bits de "suciedad" o "dirty" para indicar los bloques que han sido modificados y que todavía no se han escrito en la memoria principal

¿Cuál es el tamaño de la caché y su organización? Indique también el número de bytes por bloque (línea)

Como puede verse en la figura inferior, la caché es asociativa de 2 vías. 8 bits de la dirección se usan como índice para seleccionar la línea, que tiene en total 32 bytes. Por lo tanto, cada vía tiene $2^8 \cdot 32$ Bytes, esto es 8 KB y entonces la caché es de 16 KB.

ORGANIZACIÓN Y ESTRUCTURA DE LA MEMORIA: CACHÉS Y MEMORIA VIRTUAL

b) Las siguientes tablas muestran los contenidos del TLB y de las caches de instrucciones y datos. El tamaño de las páginas es de 4 KBytes, y en los bloques de datos de las cachés se sigue una representación big endian (el byte más significativo se corresponde con la dirección más baja). Tanto para el TLB como para las cachés, considerar como vacíos los contenidos no mostrados (indicados con puntos suspensivos). Para facilitar la lectura de los TAGs de las cachés se utiliza una notación "mixta" con dígitos en hexadecimal (negrita) y en binario (tipo de letra más pequeño).

Instruction Shadow-TLB	
Pag. Virt.	Marco Real
0	3FD6B78 0ED44
1	003556A 5344A
2	12AC771 B126C
3	12AC772 B126D

Unified TLB	
Pag. Virt.	Marco Real
0	0A7413E 33AF6
1	0A7413F 33AF7
2	0A74140 33AF8
3	0A74141 33AFB
4	66FB7A3 C6652
5	8667224 78AAB
	...
61	988FDDE 98412
62	0012AB0 B2398
63	0012AB1 B2399

Data Shadow-TLB	
Pag. Virt.	Marco Real
0	5477ABE 457FC
1	0A7413D 33AF5
2	6455ABC C172B
3	7622091 A190A
4	AB43910 7CC6D
5	9983F3A 43BBB
6	789100B ABED3
7	AB43911 610DC

Cache Instrucciones				
VIA 1			VIA 2	
IND	TAG	DATA	TAG	DATA
0	0019987	702C11BE20215A8E04C78D3C158052123EECED491D1D079D2F37E6C73DB6CFD8
...
D5	0019D7D	7942A98A137565B536290564A812998E04797393F4B1C4DCDC75F3043CDD90EA
D6	011C555	6D317495543B6887492215081FC5A082B7F3E74EB76891A8780C66755A7B0951
...
FF	1018936	18207A9B804EF21295549F7674389E90C50CA1F93CDB536792BDDEEA22E631C5

Cache Datos				
VIA 1			VIA 2	
IND	TAG	DATA	TAG	DATA
0	111BC34	9E3DDA2AAF7BC598588AAE515D40A11E2D5DEB0F04009A3911CFC0133B5BB596
...
47	0105DF0	3CF73A8F107299446B0F3E83DBA49D1309BDC996A07B093802A9FA9518278FAC
48	0015BBA	014A7C42B49D7ABD955F34CEAF723C0E57346B3FA89D3F3AB42E842CCCB82165
...
FF	1117B31	C620871B79DEA8644CCE3C3653257D90A653909AE0694F1CCB5B5351C1495F55

Independientemente de que la arquitectura del procesador sea Harvard o Von Neumann, se supone que hay una memoria principal compartida para instrucciones y datos, con ancho de palabra de 32 bits, de la que se conocen sólo los contenidos de las siguientes direcciones, pero que ocupa todo el espacio de direccionamiento real disponible en el microprocesador:

Dirección	Dato	Dirección	Dato	Dirección	Dato
0B542D94	A9BC8160	55720EC0	E87653E4	B126DFE8	7C3D19CA
11586B58	4F611703	610DC31C	46F6C776	C64636B4	18990CDA
47B45DF0	E8E1D065	8DC2E0DC	3569CC85	D8C8E220	E6FE1233

ORGANIZACIÓN Y ESTRUCTURA DE LA MEMORIA: CACHÉS Y MEMORIA VIRTUAL

Para las siguientes operaciones de lectura de instrucciones y datos, obtenga el resultado y el número de ciclos de penalización, teniendo en cuenta que las cachés de datos e instrucciones no penalizan en acierto, pero penalizan 40 ciclos en fallo. Si con los datos proporcionados no se puede saber alguno de los campos (resultado o ciclos), rellene la casilla con una X. SUPONER QUE SON ACCESOS INDEPENDIENTES: Los posibles fallos de TLB y/o caché en un acceso no deben ser tenidos en cuenta para el siguiente (esto es, considerar los contenidos de TLB, cachés y memoria de la página anterior como constantes).

Tipo	Dirección virtual	Resultado	Ciclos
INSTRUCCIÓN	0A74141ABC	7942A98A	4
DATO	8667224AD8	X	43
INSTRUCCIÓN	12AC772FE4	92BDDEEA	0
INSTRUCCIÓN	7622091248	X	X
DATO	AB4391131C	46F6C776	40

c) Un programa típico de 100000 instrucciones se ejecuta en este microprocesador con un CPI=1,4 (sin contar penalizaciones por fallos en TLB y/o cachés) y la frecuencia de operación es de 250 MHz. Calcular el tiempo de ejecución suponiendo que la tasa de fallos local del Shadow-TLB de instrucciones es de 0,032, la del Shadow-TLB de datos es 0,027, la del TLB unificado es 0,015, la de la caché de datos es 0,067 y la de la caché de instrucciones es 0,071. Suponer además que el 15% de las instrucciones son load o stores.

$$CPI_{total} = CPI_{ideal} + CDM$$

$$CDM = F_{Shadow-TLB-Instr}(t_{Shadow-TLB-Instr} + F_{Unified-TLB} \cdot t_{Unified-TLB}) + F_{Cache-Instr} \cdot t_{Cache-Instr} +$$

$$0,15 \cdot [F_{Shadow-TLB-Data}(t_{Shadow-TLB-Data} + F_{Unified-TLB} \cdot t_{Unified-TLB}) + F_{Cache-Data} \cdot t_{Cache-Data}] =$$

$$= 1,808 + 2,84 + 0,15 \cdot (1,4985 + 2,68) = 5,274$$

$$T_{CPU} = CPI_{total} \cdot NI \cdot T_{Cik} = (1,4 + 5,274) \cdot 100000 \cdot 4ns = 2,67 \text{ ms}$$

d) Se mejora el sistema de memoria principal, de tal manera que los fallos en caché de datos y/o instrucciones sólo penalizarán 20 ciclos. Calcular, usando necesariamente la ley de Amdahl, la mejora en el tiempo de ejecución del caso descrito en el apartado anterior.

$$F_m = [F_{Cache-Instr} \cdot t_{Cache-Instr} + 0,15 \cdot F_{Cache-Data} \cdot t_{Cache-Data}] / CDM = 0,486$$

$$A_m = 2$$

$$A_G = 1 / (1 - F_m + F_m / A_m) = 1,32$$

El tiempo de ejecución mejorará un 32%, esto es, será ahora $2,67ms / 1,32 = 2,02 \text{ ms}$

ORGANIZACIÓN Y ESTRUCTURA DE LA MEMORIA: CACHÉS Y MEMORIA VIRTUAL

b) Cuántos ciclos de reloj necesitará la ejecución del código que utiliza restas condicionales (SUBGT y SUBLT) y el que no las utiliza para el caso $i=35$; $j=24$ que hace ejecutar el ciclo principal 6 veces con $i>j$ y 3 veces con $j>i$. Para justificar sus respuestas utilice las tablas adjuntas

b.1: Cantidad de ciclos del código 1:
 9 iteraciones de 6 ciclos, más dos ciclos última iteración (ó -2 si no se considera vaciar el pipeline)
 $9 \times 6 + 2 = 56$ ciclos ó $9 \times 6 - 2 = 52$ (es aceptado como correcto, incluso sin sumar ó restar los 2 ciclos)
 En realidad este algoritmo necesita una iteración extra para salir donde $i=j$ tras la última actualización.
 El resultado correcto es $10 \times 6 + 2 = 62$ ciclos ó $10 \times 6 - 2 = 58$ si no se considera el vaciado del pipeline

b.2: Cantidad de ciclos del código 2:
 6 iteraciones de 8 ciclos, 3 iteraciones de 7 ciclos más dos ciclos última iteración para el vaciado ó -2 si no se considera el vaciado
 $6 \times 8 + 3 \times 7 + 2 = 71$ ciclos ó $6 \times 8 + 3 \times 7 - 2 = 67$ ciclos sin el vaciado.

Justificación de ciclos utilizados para la ejecución del código con instrucciones condicionales

CMP	I	D	E	M	W										
SUBGT		I	D	E	M	W									
SUBLT			I	D	E	M	W								
BNE				I	D	E	M	W							
N+1					I	D	-								
N+2						I	-								
CMP							I	D	E						
SUBGT								I	D						
...	1	2	3	4	5	6									

Siempre compara y pone los "flags". Las instr. siempre se ejecutan y efectúa el computo según el valor de los flags.

dos ciclos perdidos si salto ef Comienza nuevo ciclo

Se usan 6 ciclos por iteración

Justificación de ciclos utilizados para la ejecución del código sin instrucciones condicionales ($i<j$)

BLT (ef)	I	D	E	M	W										
SUB		I	D	-											
BEQ			I	-											
SUB				I	D	E	M	W							
BNE (ef)					I	D	E	M	W						
N+1						I	D	-							
N+2							I	-							
BGT								I	D	E	M	W			
SUB									I	D	E/-	M/-	W/-		
...	1	2	3	4	5	6	7								

1er salto efectivo

Este salto no se ejecuta

Si salto efectivo se pierden dos ciclos adicionales

Comienza nuevo ciclo

Se usan 7 ciclos por iteración E/- indica que depende de si BGT es efectivo ó no si se pasa por las etapas E, M y W de la siguiente instrucción.

Justificación de ciclos utilizados para la ejecución del código sin instrucciones condicionales ($i>j$)

BLT (ne)	I	D	E	M	W										
SUB		I	D	E	M	W									
BEQ (ef)			I	D	E	M	W								
SUB				I	D	-									
BNE					I	-									
BNE (ef)						I	D	E	M						
N+1							I	D	-						
N+2								I	-						
BGT									I	D	E	M	W		
SUB										I	D	E/-	M/-		
...	1	2	3	4	5	6	7	8							

1er salto no efectivo

Este salto se ejecuta

este salto se carga 2 veces si salto efectivo se pierden dos ciclos adicionales

Comienza nuevo ciclo

Se usan 8 ciclos por iteración

c) Se agregan a este procesador memorias caché. La caché de instrucciones es de 32 kBytes organizada en 64 bytes por bloque y asociativa de 8 vías. Recordar que es un procesador de 32 bits donde las palabras, direcciones e instrucciones son de ese tamaño.

C1. ¿Cómo se divide la dirección para el acceso a memoria caché?
 Total Bloques: Tamaño Total / Tamaño bloque = 32 KB / 64 B = 512 bloques.
 Como es de 8 vías hay un total de 512 bloques / 8 vías = 64 entradas (6 bits para direccionarla)

ORGANIZACIÓN Y ESTRUCTURA DE LA MEMORIA: CACHÉS Y MEMORIA VIRTUAL

Cada bloque posee 64 bytes (16 palabras de 4 bytes) luego el campo "Byte en Bloque" tiene 6 bits

El resto de bits $32 - 6 - 6 = 20$ bits de etiqueta.

Etiqueta	Índice	Byte en Bloque
20	6	6

C2. Suponiendo que el código ensamblador que utiliza restas condicionales (SUBGT y SUBLT) está escrito a partir de la posición de memoria 0000_8034 (hexadecimal), que el programa no está presente en la caché al iniciar el análisis y que durante su ejecución no se ejecuta otro programa concurrentemente, ¿cuántos fallos de caché se producen? Justifique su respuesta.

El programa ocupa 4 instrucciones que comienzan en: 0000_8034, 0000_8038, 0000_803C y 0000_8040, como la última es un salto también se capturan las dos siguientes (0000_8044 y 0000_8048), antes de actualizar el contador de programas.

La división en los campos de acceso nos da:

Dir	tag	índice	BB	1ra vez	Resto Iterac
0000_8034	00008	000	114	F	A
0000_8038	00008	000	118	A	A
0000_803C	00008	000	11C	A	A
0000_8040	00008	001	000	F	A
0000_8044	00008	001	004	A	A
0000_8048	00008	001	008	A	A

Es decir ocuparán dos entradas diferentes. En una de las 8 vías de la entrada 0 (con 6 bits) estarán las 3 primeras instrucciones, en tanto que en una de las 8 vías de la entrada 1 (con 6 bits) estarán la última instrucciones del programa y las otros dos que se capturan innecesariamente.

Esto provocará un total de **dos fallos** en la primera ejecución y ningún otro a menos que otros procesos ó programas desaloje las 8 entradas de cada una de las vías.