

Supón un sistema BigEndian. con ISA MIPS de 32 bits (palabra de 32 bits), y direccionamiento de memoria a nivel de byte. El sistema tiene una cache de datos de 64 KiB, asociativa por conjuntos de 2 vías y con tamaño de línea de 8 palabras. Considera además que en la posición de memoria M está almacenado el valor $M \% 256$, para cualquier dirección M de memoria (es decir, en cada dirección de memoria se encuentra almacenado un byte igual a los dos últimos dígitos hexadecimales de la dirección; por ejemplo, en la dirección 0x1234ABCD se encuentra el byte 0xCD).

Si inicialmente la cache está vacía, responde a las siguientes preguntas, justificando tu respuesta.

- Supón que el registro `$t1` guarda el valor 0x10100058, y se ejecuta la instrucción `lw $t2, 0($t1)`. Indica qué datos se traen de memoria a la cache y en qué conjunto de esta se guardan. ¿Qué valor se guarda en el campo etiqueta? ¿Qué valor se guarda en `$t2`?
- Si inmediatamente después de esa instrucción, se ejecutan las instrucciones `lw $t3, 4($t1)` y `lw $t4, 8($t1)`, ¿qué pasa en la cache?, ¿qué valores se guardan en los registros `$t3` y `$t4`?

Respuesta:

- Para determinar el número de conjuntos tenemos:

$$T_{cache} = N_{ctos} \times N_{Vias} \times T_{linea} \Rightarrow 64 \cdot 2^{10} = N_{ctos} \cdot 2 \cdot (8 \cdot 4) \Rightarrow N_{ctos} = 2^{10} = 1024$$

El formato de la dirección queda (palabra de 4 bytes, línea de 8 palabras):

```
eeeeeeeeeeeeeeee ccccccccc ppp bb
```

la etiqueta es de 17 bits.

Dirección

```
0x10100058 = 00010000000100000 0000000010 110 00
```

Se cargan en la cache los datos que van desde la dirección:

```
00010000000100000 0000000010 000 00 = 0x10100040
```

hasta la

```
00010000000100000 0000000010 111 11 = 0x1010005F
```

Es decir, se carga un bloque de 32 bytes (8 palabras * 4 bytes), con los bytes 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 4A, 4B, 4C, 4D, 4E, 4F, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 5A, 5B, 5C, 5D, 5E, 5F. Este bloque se guarda en cualquiera de las vías del conjunto 2.

La etiqueta es:

```
0 0010 0000 0010 0000 = 0x02020
```

¿Qué se guarda en `$t2`? Ya que la instrucción es `lw (load word)`, se carga en `$t2` una **palabra** completa (4 bytes). Y como el sistema es Big Endian, los 4 bytes son los que están en las direcciones 0x10100058 a 0x1010005B en ese orden. Es decir, en `$t2` se guarda la palabra **0x58595A5B**.

- `lw $t3, 4($t1)`, Se pide la dirección $0x10100058 + 4 = 0x1010005C$

```
0x1010005C = 00010000000100000 0000000010 111 00
```

Se va al conjunto 2 y se compara la etiqueta. Como ya está se produce un acierto. En `$t2` se guarda la palabra 0x5C5D5E5F.

- `lw $t4, 8($t1)`, Se pide la dirección $0x10100058 + 8 = 0x10100060$

```
0x10100060 = 00010000000100000 0000000011 000 00
```

va al conjunto 3, que está vacío. Fallo obligatorio. En `$t4` se guarda la palabra 0x60616263