Guía para el Trabajo de Laboratorio Número 2

Arquitectura de Computadoras 20/05/2026

Objetivo

El objetivo del laboratorio es enfatizar los conceptos enseñados en clase sobre el funcionamiento de las memorias caché. Para ello se utilizará el simulador **CachéSim 2.0**, desarrollado en nuestra Facultad, que permite analizar el comportamiento de estos dispositivos, así como también las distintas alternativas de diseño: grado de asociatividad, tamaño del bloque, algoritmo de reemplazo, etc.

El simulador funciona completamente como una aplicación WEB que puede ser accedida en la dirección *http://microprocesadores.unt.edu.ar/cachésim/*. En el mismo sitio se incluye un manual de funcionamiento del mismo. Debido a su carácter didáctico, el simulador trabaja con un solo nivel de caché.

Generación de una traza

Una traza no es más que un listado de accesos a memoria, es decir, una secuencia de direcciones de memoria que son referenciadas para su lectura o escritura. Es importante comprender el concepto de traza porque en el desarrollo de este laboratorio constituye en muchos casos la base a partir de la cual se analiza el funcionamiento de las distintas configuraciones de memoria caché.

En varios ejercicios del laboratorio será necesario escribir las trazas a partir de las cuales se realizarán las simulaciones utilizando el CachéSim. Por ello es importante hacer una breve reseña del formato que deberá tener este listado de accesos a memoria para que el mismo pueda ser interpretado correctamente por el simulador.

En primer lugar es importante destacar que durante el laboratorio se utilizarán tres tipos de acceso a memoria soportados por el simulador:

- Lectura de Instrucciones (Identificador 0)
- Lectura de Datos (Identificador 2)
- Escritura de Datos (Identificador 3)

La traza no es más que un archivo de texto plano con extensión .prg el cual contiene cada uno de los accesos a memoria ubicando uno por cada línea del archivo. Es importante destacar que el primer número de cada línea indicará el tipo de acceso a memoria (ver identificador en los tres tipos de acceso) y estará separada por un espacio de la dirección de memoria expresada en hexadecimal.

De este modo el contenido de un ejemplo de archivo de traza, por ejemplo Traza.prg sería el siguiente:

```
0 00000170
2 0000CB80
0 00000174
0 00000178
0 0000017C
0 00000180
3 00007108
0 00000184
3 0000710C
```

En el ejemplo puede observarse que se accede a un total de 8 instrucciones ubicadas a partir de la dirección `0x00000170`. Además puede observarse que existe un acceso de lectura a memoria de datos a la dirección `0x0000CB80` y dos escrituras de datos en las direcciones `0x00007108` y `0x0000710C`.

Generación de una traza a partir de lenguaje ensamblador

En primer lugar resulta importante tener en cuenta las directivas al compilador `.data` y `.text` ya que las mismas indican a partir de que dirección se ubicarán los datos y las instrucciones respectivamente. Es fundamental para realizar esta conversión identificar la dirección en la que se encuentra cada instrucción del código y además tener en cuenta que por cada instrucción de acceso a datos (ya sea load o store) se generarán dos entradas en la traza, una para acceder a la instrucción y otra para leer o escribir el dato según corresponda. Vale la pena resaltar que como las pseudoinstrucciones se pueden expandir a más de una instrucción, también en este caso se pueden producir más de un acceso a memoria de instrucciones y, por lo tanto, dos entradas en la traza.

Por otro lado, es importante tener en cuenta los lazos que pudieran existir dentro del código recordando que las instrucciones permanecen almacenadas en la misma ubicación de memoria y teniendo cuidado con el manejo de los índices que pueden alterar la ubicación en la cual se leen o escriben datos. En el siguiente ejemplo se observa como se traduce un fragmento de un programa escrito en assembler del RISC-V a la traza de accesos que el mismo genera.

```
        .data 0x10000000        # Direccion de Origen de Datos
n:      .word 4

        .text 0x00000000        # Direccion de Origen de Codigo
main:                           # Traza de Accesos Resultante
        li t0, 0x10000000       # 0 00000000
                                # 0 00000004
        lw t1, 0(t0)            # 0 00000008
                                # 2 10000000
        add t5, t1, t3          # 0 0000000C
        sw  t5, 4(t0)           # 0 00000010
                                # 3 10000004
```

Problemas Propuestos

1) *[Modificado]* En un caché de mapeo directo de 512 bytes y 16 palabras de 32 bits por bloque, y utilizando la traza de memoria Traza.prg provista con el simulador, se pide que:
   a) Indique cómo se descomponen las direcciones. Verifique la misma con el simulador.

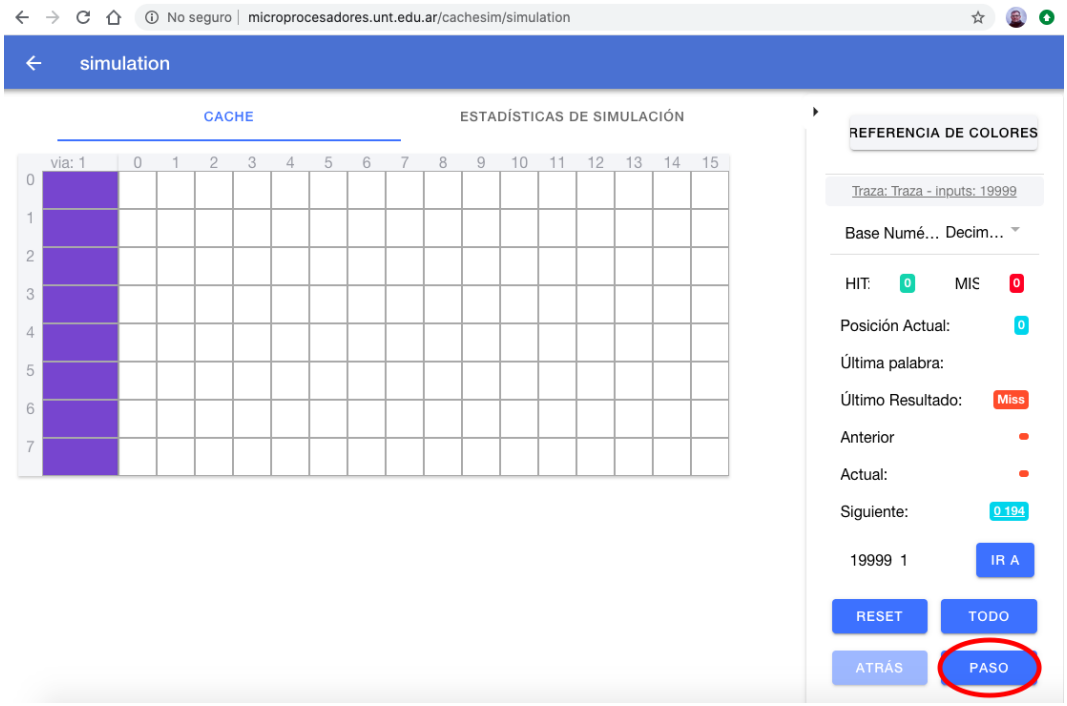
Soluciones Completas

1)
   a) Los 23 bits más significativos corresponden a la etiqueta (tag) del caché, los siguientes 3 bits son el índice, los 4 siguientes la selección de la palabra, y los restantes 2 bits seleccionan el byte.

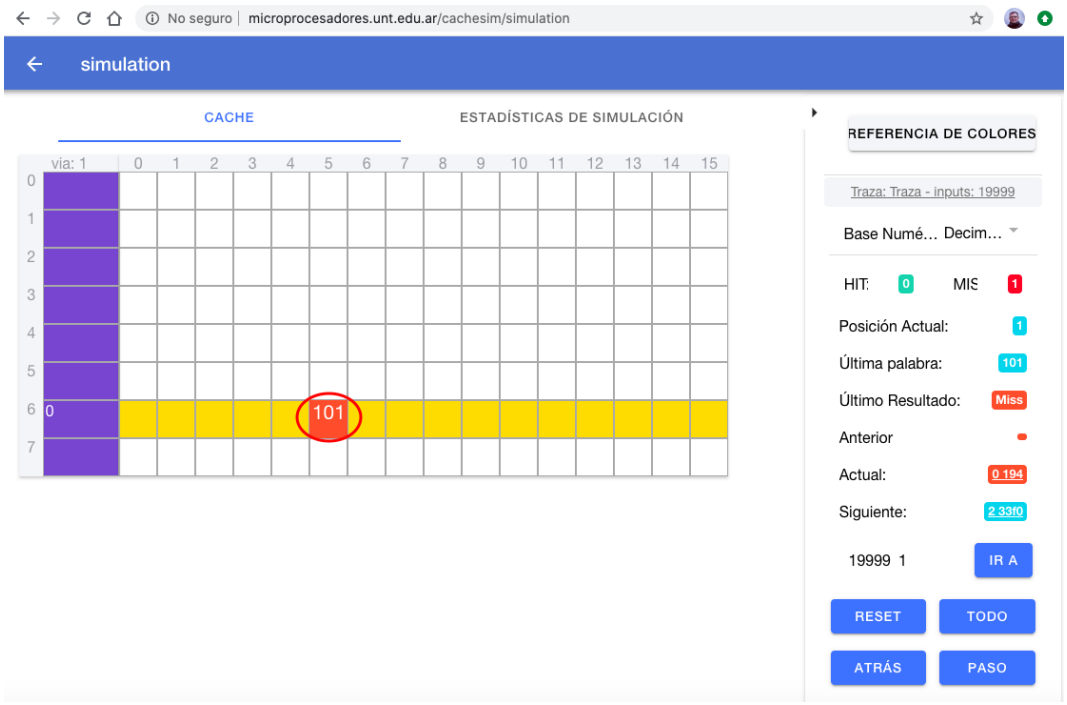
   Para verificar la descomposición de las direcciones en el simulador debemos ingresar a *http://microprocesadores.unt.edu.ar/cachésim/home*
   para luego configurar el simulador según los siguientes datos



Una vez configurado, presionamos el boton *simular* para pasar a la siguiente pantalla



Aquí debemos presionar el botón *paso* para ejecutar el primer paso de la simulación



Por último, presionamos el dato cargado en el caché para verificar la descomposición de la dirección del primer dato 0x194

