Universidad de Costa Rica

Escuela de Ciencias de la Computación e Informática

CI-1323: Arquitectura de Computadoras

Proyecto: Simulador de un procesador MIPS para enteros con dos núcleos y doble hilo en el nucleo 0

Implementación del procesador MIPS

Profesora Ileana Alpízar

Estudiantes:

Calderón Calderón Elías, B51322

León Sarkis Josué, B53846

Montes de Oca Brenes Daniel, B54621

Primer ciclo, 2018

a. Detalles de implementación

El núcleo 0 es de **doble hilillo**. Esta solución se implementó con **dos hilos**. Se tiene un hilo principal y un hilo que se dedica a resolver fallos de caché. Para lograr la sincronización de los hilos se usó un tipo de barrera que en Java se conoce como *Phaser*. El phaser provee la posibilidad de realizar sincronización similar a una barrera usando un número variable de hilos durante la ejecución, algo muy conveniente para este caso, ya que el hilo para resolver fallos solo se ejecuta cuando se necesita. Además, los **cambios de contexto los hace cada hilo**, en lugar de ejecutarse en el hilo principal de la simulación (en nuestro caso, el simulation controller).

Detalles especiales de implementación

- Existe un hilo que se crea y ejecuta cuando hay un fallo de caché en el núcleo 0
- Como el hilo mencionado en el punto anterior no siempre se está ejecutando, puede existir una cantidad variable de hilos que tienen que llegar a la barrera para poder avanzar el ciclo. Para esto se utilizó un Phaser.
- El phaser provee una operación, register, para incrementar la cantidad de hilos que tienen que llegar a la barrera. Register se ejecuta al inicio de la ejecución de cada hilo (incluido el de fallo de caché), de la misma manera el hilo principal se “registra” mediante un parámetro del constructor. Eso se hace así para que el hilillo principal pueda esperar a que los otros hilos terminen su ejecución.
- De la misma manera provee arrive and deregister, para reducir la cantidad de hilos que tienen que llegar a la barrera. Este se ejecuta cuando finaliza la ejecución de cada hilo, también incluido el de fallo de caché. El hilo para fallos de caché es especialmente interesante porque es el único que puede registrarse y “deregistrarse” varias veces en una sola ejecución (aunque lo hace con diferentes instancias de él mismo).
- Existen una clase con las constantes necesarias para la simulación.
- El acierto del store se maneja como un fallo de caché en el núcleo 0 debido a que existe la posibilidad de que se tenga que acceder a memoria, eso ocurre en el caso de que la posición con la misma etiqueta estaba modificada en la otra caché.

b. Manual de instalación

Sobre el ejecutable entregado

Al ejecutable entregado se le realizó un cambio de extensión para no tener problemas con el envío. El ejecutable enviado tiene el nombre *processor-simulation-executable.zip*. A este se le tiene que cambiar el nombre a *processor-simulation-executable.jar*. En este documento varias veces se hace referencia al entregable renombrado.

Cómo correr el ejecutable

En la entrega se adjunta un jar ejecutable (con cambio de extensión a zip) el cuál se llama *processor-simulation-executable.jar*, el cuál contiene los hilillos de prueba. Para ejecutarlo se necesita utilizar una versión de Java 8 o mayor. Esto se puede solicitar desde consola usando el comando: `java -version`. A continuación se presenta una salida normal de ese comando.

```
java version "1.8.0_171"
Java(TM) SE Runtime Environment (build 1.8.0_171-b11)
Java HotSpot(TM) 64-Bit Server VM (build 25.171-b11, mixed mode)
```

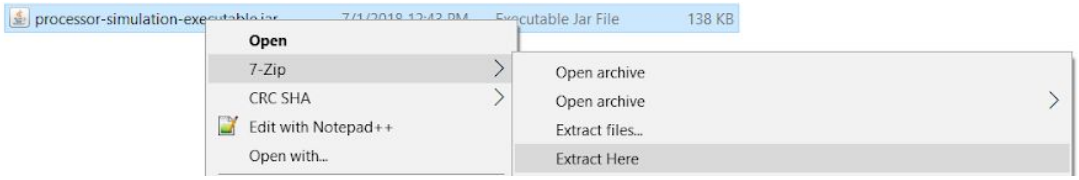
Ejemplo de salida de `java -version`

Al indicar 1.8.0_171 indica que se tiene la versión 8 de Java. Esto indica que el programa se puede ejecutar. Para hacerlo se puede utilizar el comando: `java -jar processor-simulation-executable.jar`.

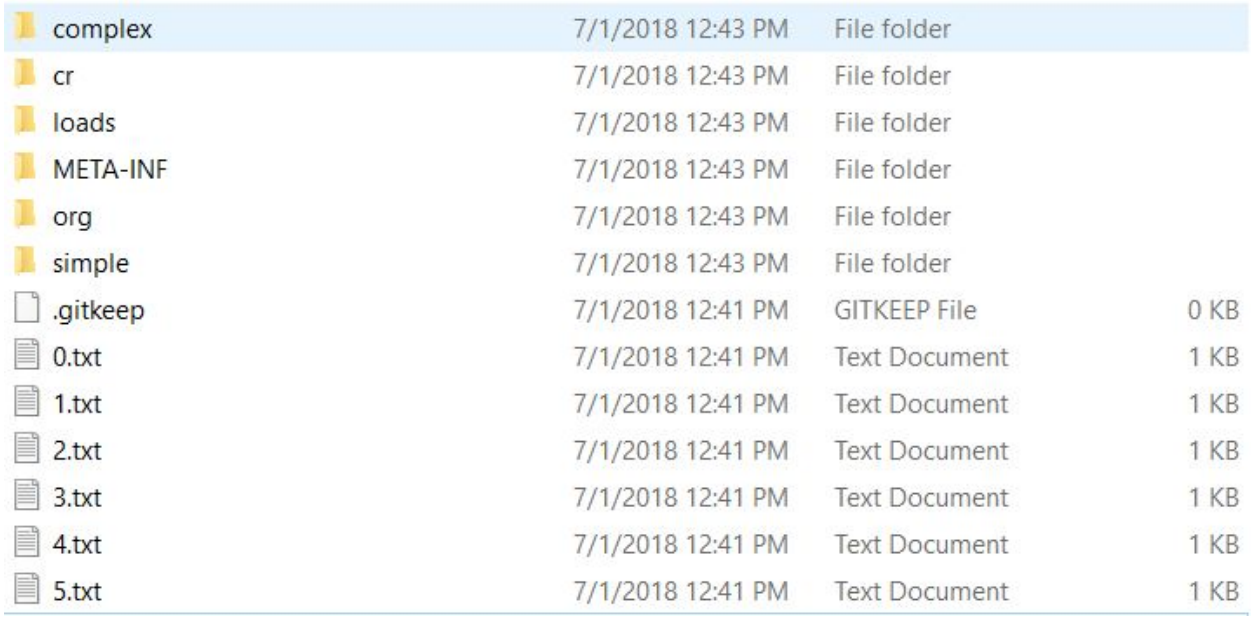
Cómo revisar los contenidos del jar

El ejecutable entregado `processor-simulation-executable.jar` no solo contiene lo que usualmente se necesita para ejecutar el programa. También contiene el código fuente y los hilillos de prueba. Este ejecutable, al igual que otros jars es navegable de manera similar a un archivo zip.

A pesar de que la entrega incluye un zip con el código fuente. Se provee la opción de llegar al código fuente u otra información al interior del ejecutable. Esto se puede lograr de la siguiente manera usando 7-zip (disponible en la mayoría de equipos de los laboratorios de la escuela).



Se selecciona la opción “Extract Here”



complex	7/1/2018 12:43 PM	File folder	
cr	7/1/2018 12:43 PM	File folder	
loads	7/1/2018 12:43 PM	File folder	
META-INF	7/1/2018 12:43 PM	File folder	
org	7/1/2018 12:43 PM	File folder	
simple	7/1/2018 12:43 PM	File folder	
.gitkeep	7/1/2018 12:41 PM	GITKEEP File	0 KB
0.txt	7/1/2018 12:41 PM	Text Document	1 KB
1.txt	7/1/2018 12:41 PM	Text Document	1 KB
2.txt	7/1/2018 12:41 PM	Text Document	1 KB
3.txt	7/1/2018 12:41 PM	Text Document	1 KB
4.txt	7/1/2018 12:41 PM	Text Document	1 KB
5.txt	7/1/2018 12:41 PM	Text Document	1 KB

Eso deja todos los archivos a disposición

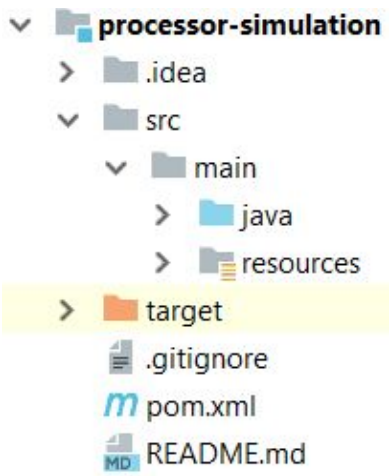
Al seguir las instrucciones el jar se descomprime en la carpeta en donde uno estaba ubicado cuando eligió usar “Extract Here”. Los hilillos de prueba son los archivos numerados que se encuentran en la raíz. El código fuente empieza a partir de la carpeta cr (utilizando la estructura tradicional de paquetes de java).

Cómo generar el ejecutable

Si por alguna razón, por ejemplo para cambiar los hilillos de prueba. se desea generar el ejecutable se requiere tener al menos la versión 3 de Maven. Para verificar la versión actual de Maven se puede usar el comando `mvn -version`.

```
Apache Maven 3.5.4 (1edded0938998edf8bf061f1ceb3cfdeccf443fe; 2018-06-17T12:33:14-06:00)
Maven home: C:\Users\danmd\Programs\apache-maven-3.5.4\bin\..
Java version: 1.8.0_171, vendor: Oracle Corporation, runtime: C:\Program Files\Java\jre1.8.0_171
Default locale: en_US, platform encoding: Cp1252
OS name: "windows 10", version: "10.0", arch: "amd64", family: "windows"
```

Los hillillos de prueba se encuentran en la carpeta resources, para facilitar el trabajo de encontrarlos se proporciona la siguiente captura de la estructura del proyecto.



Estructura del proyecto

Para generar el jar ejecutable se puede utilizar el comando `mvn package`. Con ese comando se generan 2 jars en la carpeta target. El que se llama `processor-simulation-executable.jar` es el que contiene las dependencias y por lo tanto es el que se puede ejecutar más fácilmente.

```
[INFO] Scanning for projects...
[WARNING]
[WARNING] Some problems were encountered while building the effective model for cr.ac.ucr.ecci.ci1323:processor-simulation:jar:0.1
[WARNING] 'build.plugins.plugin.version' for org.apache.maven.plugins:maven-shade-plugin is missing. @ line 30, column 21
[WARNING]
[WARNING] It is highly recommended to fix these problems because they threaten the stability of your build.
[WARNING]
[WARNING] For this reason, future Maven versions might no longer support building such malformed projects.
[WARNING]
[INFO]
[INFO] -------------< cr.ac.ucr.ecci.ci1323:processor-simulation >-------------
[INFO] Building processor-simulation 0.1
[INFO] --------------------------------[ jar ]---------------------------------
[WARNING] The artifact org.apache.commons:commons-io:jar:1.3.2 has been relocated to commons-io:commons-io:jar:1.3.2
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ processor-simulation ---
[WARNING] Using platform encoding (Cp1252 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] Copying 11 resources
[INFO]
[INFO] --- maven-compiler-plugin:3.7.0:compile (default-compile) @ processor-simulation ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ processor-simulation ---
[WARNING] Using platform encoding (Cp1252 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] skip non existing resourceDirectory C:\Users\danmd\IdeaProjects\processor-simulation\src\test\resources
[INFO]
[INFO] --- maven-compiler-plugin:3.7.0:testCompile (default-testCompile) @ processor-simulation ---
[INFO] No sources to compile
[INFO]
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ processor-simulation ---
[INFO] No tests to run.
[INFO]
[INFO] --- maven-jar-plugin:2.4:jar (default-jar) @ processor-simulation ---
[INFO]
[INFO] --- maven-shade-plugin:3.1.1:shade (default) @ processor-simulation ---
[INFO] Including commons-io:commons-io:jar:1.3.2 in the shaded jar.
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 1.468 s
[INFO] Finished at: 2018-06-26T20:01:19-06:00
[INFO] ------------------------------------------------------------------------
```

Ejemplo de salida de ejecutar mvn package

c. Problemas no resueltos

Se considera que el proyecto fue exitoso, ya que no quedaron problemas sin resolver.

d. Resultados de correr el grupo de hilillos de prueba

```
Memoria compartida de datos:
{ 88    1   1   1
1   22  1   44
1   1   1   1
1   1   1   1
1   1   1   1
1   1   1   1
1   1   1   1
5   5   5   5
5   5   5   5
5   5   5   5
5   5   5   5
5   5   5   2
2   2   2   2
2   2   2   2
2   2   2   2
2   2   3   3
4   4   4   4
4   4   4   3
3   3   3   3
3   3   3   3
3   3   3   3
3   3   3   3
1   1   1   33
99  1   1   205
 }
```

Memoria compartida de datos

```
Caches para el Nucleo #0
Cache de instrucciones:
Posicion #0: Etiqueta 56, Bloque de Instrucciones: { 43   0   22  68  63  0   0   0   8   0   21  12  34  22  22  22   }
Posicion #1: Etiqueta 57, Bloque de Instrucciones: { 8 0   2   2   14  21  2   23  8   21  21  -2  32  22  23  22   }
Posicion #2: Etiqueta 58, Bloque de Instrucciones: { 5 21  0   -4  2   31  0   0    }
Posicion #3: Etiqueta 35, Bloque de Instrucciones: { 8 0   28  28  8   0   24  240 8   0   5   4   43  24  3   0    }
Posicion #4: Etiqueta 36, Bloque de Instrucciones: { 43   24  3   4   8   5   5   -1  5   5   0   -2  43  24  3   8    }
Posicion #5: Etiqueta 37, Bloque de Instrucciones: { 43   24  3   12  34  28  4   28  8   24  24  16  5   28  0   -10  }
Posicion #6: Etiqueta 38, Bloque de Instrucciones: { 8 0   31  55  43  0   31  92  35  0   11  368 35  0   12  0    }
Posicion #7: Etiqueta 55, Bloque de Instrucciones: { 5 4   0   -3  3   0   0   16  12  22  2   22  43  0   5   64   }
Cache de datos:
Posicion #0: Etiqueta 0, Estado: SHARED, Bloque de Datos: { 881   1   1    }
Posicion #1: Etiqueta 1, Estado: SHARED, Bloque de Datos: { 1 22  1   44   }
Posicion #2: Etiqueta 18, Estado: SHARED, Bloque de Datos: { 33   3   3    }
Posicion #3: Etiqueta 11, Estado: MODIFIED, Bloque de Datos: { 5  5   5   5    }
Posicion #4: Etiqueta 4, Estado: MODIFIED, Bloque de Datos: { 45  42  1   1    }
Posicion #5: Etiqueta 5, Estado: MODIFIED, Bloque de Datos: { 1   1   1   55   }
Posicion #6: Etiqueta 22, Estado: SHARED, Bloque de Datos: { 11   1   33   }
Posicion #7: Etiqueta 23, Estado: SHARED, Bloque de Datos: { 99   1   1   205  }


Caches para el Nucleo #1
Cache de instrucciones:
Posicion #0: Etiqueta 52, Bloque de Instrucciones: { 43   0   1   380 8   0   31  33  43  0   31  364 35  0   11  368  }
Posicion #1: Etiqueta 53, Bloque de Instrucciones: { 35   0   12  0   35  0   13  92  35  0   14  28  63  0   0   0    }
Posicion #2: Etiqueta 50, Bloque de Instrucciones: { 8 2   2   16  32  1   6   1   32  1   7   1   32  1   8   1    }
Posicion #3: Etiqueta 51, Bloque de Instrucciones: { 32   1   9   1   8   5   5   -1  5   5   0   -2  5   3   0   -14  }
Cache de datos:
Posicion #0: Etiqueta 0, Estado: SHARED, Bloque de Datos: { 881   1   1    }
Posicion #1: Etiqueta 1, Estado: SHARED, Bloque de Datos: { 1 22  1   44   }
Posicion #2: Etiqueta 22, Estado: SHARED, Bloque de Datos: { 11   1   33   }
Posicion #3: Etiqueta 23, Estado: SHARED, Bloque de Datos: { 99   1   1   205  }
```

Cachés de núcleo 0 y núcleo 1

```
Contextos que finalizaron:
Contexto #1: PC = 552, Registros = { 0, 0, 2, 0, 4, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 272, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 88, } , Ciclos consumidos = 1699, Nucleo en que termino: 1

Contexto #3: PC = 760, Registros = { 0, 0, 0, 6, 4, 3, 1, 0, 0, 0, 4, 1, 88, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 288, 0, 0, 0, 0, 22, 44, } , Ciclos consumidos = 1192, Nucleo en que termino: 1

Contexto #0: PC = 468, Registros = { 0, 5, 192, 0, 4, 0, 0, 0, 0, 0, 0, 0, 88, 1, 44, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 99, } , Ciclos consumidos = 4588, Nucleo en que termino: 0

Contexto #4: PC = 864, Registros = { 0, 205, 352, 0, 4, 0, 3, 1, 1, 1, 0, 99, 88, 1, 44, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 33, } , Ciclos consumidos = 2072, Nucleo en que termino: 1

Contexto #2: PC = 636, Registros = { 0, 0, 0, 3, 4, 0, 0, 0, 0, 0, 0, 99, 88, 0, 44, 33, 0, 0, 0, 0, 0, 0, 0, 0, 0, 352, 0, 0, 0, 0, 0, 0, 55, } , Ciclos consumidos = 5227, Nucleo en que termino: 0

Contexto #5: PC = 904, Registros = { 0, 0, 2, 0, 0, 45, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 42, 1, 0, 0, 0, 0, 0, 0, 0, 888, } , Ciclos consumidos = 736, Nucleo en que termino: 0
```

Contenido de los registros

e. Nota para los miembros del grupo

Integrante	Elías Calderón	Josué León Sarkis	Daniel Montes de Oca
Nota	100	100	100