



Área Académica de Ingeniería en Computadores

Curso

Arquitectura de Computadores 2

Proyecto 1 Propuesta 2 Manual de Usuario

Profesor: Ronald García Fernández

Elaborado por:

Leonardo Guillen Fernandez

Josue Cubero Montero

Juan Peña Rostrán

Brandon Gomez Gomez

II Semestre 2023, Cartago, Costa Rica

Introducción:

Esta herramienta de simulación y modelado de coherencia de caché, implementada en C, nos ayuda a comprender cómo los procesadores multinúcleo gestionan la memoria compartida. Asimismo, nos permite explorar protocolos de coherencia de caché como MESI y MOESI simulando CPU y un bus de comunicación. De esta manera, podemos analizar los estados de la caché, la coherencia de la memoria y los efectos del protocolo en el rendimiento del sistema, proporcionando información valiosa sobre los entornos informáticos modernos.

Herramientas y Dependencias necesarias:

- GCC compiler
- Make Utility
- POSIX Threads (pthread) library: <pthread.h>
- Message Queue library (for inter-thread communication): <mqueue.h>
- Standard C libraries
- GTK for C/C++ Interfaz Gráfica: instalar con -> sudo apt-get install libgtk-3-dev

Instalación:

1. Clona el repositorio en tu máquina local:

- git clone https://github.com/BrandonMGG/Coherence-Modelling-and-Evaluation
- cd Coherence-Modelling-and-Evaluation

2. Compile y ejecute la interfaz de usuario: make mainUI, make run_simulator

Información de la Interfaz:

1. Posterior a ejecutar el comando make run_simulator, se desplegará la siguiente interfaz, en esta es posible, visualizar tres elementos de procesamiento denominados PE1, PE2 y PE3.

PE1 Current: 3			
	state	address	data
A0	0	-1	0
A1	0	-1	0
A2	0	-1	0
A3	0	-1	0

PE2 Current: 0			
	state	address	data
B0	0	-1	0
B1	0	-1	0
B2	0	-1	0
B3	0	-1	0

PE3 Current: 0			
	state	address	data
C0	0	-1	0
C1	0	-1	0
C2	0	-1	0
C3	0	-1	0

State Nomenclature	
Invalid = 0	Exclusive = 1
Shared = 2	Owned = 4
Modified = 3	

Main Memory	
Address	Data
0	2
1	17
2	52
3	9
4	121
5	42
6	110
7	12
8	62
9	69
A	36
B	94
C	51
D	24
E	39
F	81

Message	Address	Core ID
0		0

Select PE

Operation

Address

Protocol

Start

Stop

Step

Run

Generate Report

Cada una de estas unidades de procesamiento cuenta con datos de gran relevancia para comprender el flujo de información como lo son los siguientes:

- a. Current: Muestra la instrucción en ejecución, ya sea escritura, lectura o incremento como (WRITE, READ, INCR), así como el valor con el que se opera.
- b. A0, A1, A2, A3: Corresponden a las líneas de cache las cuales cuentan con múltiples datos.
- c. State: Muestra el estado de la línea de cache determinado por el protocolo en ejecución.
- d. Address: Muestra la dirección de memoria del dato correspondiente.
- e. Data: Muestra el dato almacenado en la dirección de memoria.

PE1			
Current:			3
	state	address	data
A0	0	-1	0
A1	0	-1	0
A2	0	-1	0
A3	0	-1	0

2. Por otra parte, se tiene la sección State Nomenclature donde se describen los valores que pueden tener los "state" de cada una de los PEs de acuerdo a los protocolos MESI y MOESI donde se puede tener:
 - a. Invalid
 - b. Exclusive
 - c. Shared
 - d. Modified
 - e. Owned

State Nomenclature	
Invalid = 0	Exclusive = 1
Shared = 2	Modified = 3
Owned = 4	

3. Asimismo, se tiene la sección de memoria principal donde se muestra continuamente los datos, de la memoria compartida entre los PEs y como los mismos se actualizan.

Main Memory	
Address	Data
0	2
1	17
2	52
3	9
4	121
5	42
6	110
7	12
8	62
9	69
A	36
B	94
C	51
D	24
E	39
F	81

4. Por último, se tiene la sección más importante, que es donde se pueden manipular los datos de cada una de las PEs. Para iniciar se tiene:
 - a. Sección superior, correspondiente a “Message”, “Address” y “CoreID”, la cual muestra si se dio un read-miss=0, write-miss=1, con su respectiva dirección de memoria y el núcleo donde está corriendo.
 - b. Por otra parte, se tiene la sección inferior donde se pueden introducir los datos:
 - i. Para hacerlo es necesario, no haber empezado la ejecución automática con “start” y utilizar los desplegables, así como el espacio de texto. Para introducir los datos, como se observa en la imagen de ejemplo

Enter Data		ADD
41		
Select PE	PE3	
Operation	WRITE	
Address	0x4	
Protocol	MOESI	Generate Report

- ii. Una vez ingresados los datos, se puede dar click al botón “ADD”, lo cual añadirá la instrucción al PE seleccionado.

Información de Ejecución:

Al conocer ya la estructura de la interfaz de la sección previa, se puede describir los modos de ejecución del programa.

1. Para empezar, se tiene el modo automático, donde al dar click en el botón de “start” el programa se ejecutará automáticamente con instrucciones y datos de memoria principal generados de manera aleatoria, simulando el comportamiento de un flujo de datos entre los PEs ejecutándose.
2. Por otra parte, se tiene el “step” el cual ejecutara de uno a uno cada una de las transiciones.
3. “Stop” funciona para detener la ejecución de manera momentánea para observar el estado de los datos del sistema.
4. Run ejecuta las instrucciones generadas con el “ADD”

5. Y “Generate Report” el cual genera el reporte con las métricas obtenidas del sistema para visualizarla.

mainUI

PE1
Current: 3

	state	address	data
A0	0	-1	0
A1	0	-1	0
A2	0	-1	0
A3	0	-1	0

State Nomenclature

Invalid = 0

Exclusive = 1

Shared = 2

Modified = 3

Owned = 4

Main Memory

Address	Data
0	2
1	17
2	52
3	9
4	121
5	42
6	110
7	12
8	62
9	69
A	36
B	94
C	51
D	24
E	39
F	81

PE2
Current: 0

	state	address	data
B0	0	-1	0
B1	0	-1	0
B2	0	-1	0
B3	0	-1	0

PE3
Current: 0

	state	address	data
C0	0	-1	0
C1	0	-1	0
C2	0	-1	0
C3	0	-1	0

Message 0

Address | Core ID 0

Enter Data

ADD

Select PE

Operation

Address

Protocol

Generate Report

Start

Stop

Step

Run