

# Proyecto 1 - Coherencia de caché

Gerardo Zeledon Martinez  
gerardo.zeledon14@gmail.com  
Área Académica de Ingeniería en Computadores  
Instituto Tecnológico de Costa Rica

**Resumen**—En este documento se presentará el proceso de creación y resultados del desarrollo de un sistema que simula el manejo de coherencia de cache con dos procesadores, con dos núcleos cada uno, una memoria unificada y una L2 compartida entre los dos chips.

**Palabras clave**—MSI, Coherencia, Write through, cache.

## I. INTRODUCCIÓN

En la actualidad es de suma importancia la comunicación entre los diferentes núcleos y procesadores en un computador ya que en muchas ocasiones estos comparten memorias o simplemente deben estar sincronizados, es por ese motivo que han nacido diversos protocolos que facilitan reglas para estandarizar esta comunicación. De la misma manera estos protocolos son de suma importancia a la hora de mantener la concordancia entre las diferentes memorias caché de los procesadores, es por esa razón que en este documento se va a mostrar un sistema en el cual se realiza el chequeo de la coherencia de caché en un sistema de dos procesadores con dos núcleos cada uno y una memoria unificada.

## II. SISTEMA DESARROLLADO

Como ya se menciono anteriormente en el diagrama de la figura 1 se puede observar la estructura del sistema en general donde hay una aplicacion gráfica en la cual se inicia el sistema y además muestra la información de lo que esta sucediendo en los núcleos.

Cada uno de los núcleos del sistema contarán con una cache L1 de dos bloques mapeados con correspondencia directa. Los procesadores cuentan con un cache L2 compartida entre los dos chips la cual también es de correspondencia directa, pero esta tiene 4 bloques. Finalmente la memoria principal es de 16 bloques de tamaño los cuales almacenan un dato de 32 bits de tamaño.

La comunicación entre las cache L2 se da cuando no se encuentra un dado en la cache local entonces se consulta en la externa para ver si ella lo tiene.

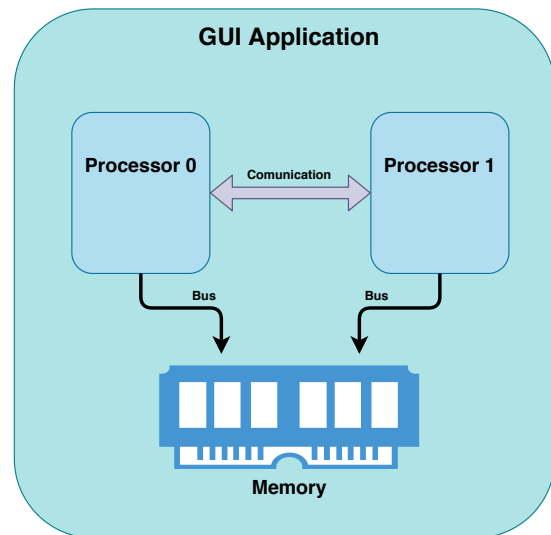


Figura 1: Diagrama de bloques del sistema.

Para la implementación del sistema se desarrollo el diagrama de clases de la figura 2 la cual posee principalmente las siguientes clases:

- **Core:** Es el nivel mas bajo del sistema y la base para el funcionamiento en general. En caso de que se dé una escritura o una lectura, esta clase levantara unas banderas, las cuales avisaran a un nivel superior que el núcleo necesita atención por parte del procesador.
- **CPU:** esta clase es la encargada de guardar la matriz L2, instanciara dos objetos de tipo Core. Al igual que la clase Core, esta clase posee banderas que servirán para avisar que se necesita atención.
- **Sistema:** esta es la clase principal del sistema la cual se encargará de manejar a los dos procesadores que tendrá instanciados, almacenará los datos de la memoria principal y proveerá métodos para atender las banderas que generan los procesadores.
- **Instruction Generator:** Aquí es donde se generan las instrucciones de manera completamente aleatoria usando una distribución binomial.
- **Main:** Ventana principal de la aplicación, aquí se define el tamaño de la ventana, icono y se referencia al archivo el cual posee todas las vistas de la interfaz de usuario

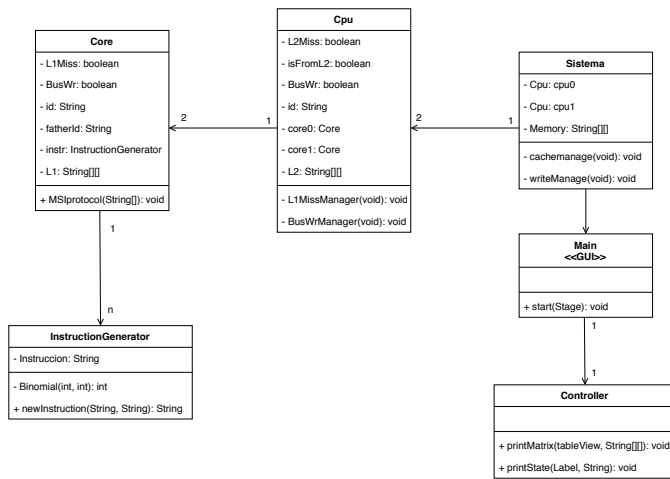


Figura 2: Diagrama de clases del sistema.

El tratamiento que se le va a dar a las instrucciones será tomando en cuenta el protocolo de coherencia de cache MSI y la política de escritura Write through. Tomando en cuenta estas primicias se desarrolló el diagrama de flujo de la imagen 3 en el cual se manejan las instrucciones de la siguiente manera:

- **Write:** Cuando se da el núcleo escribe en cache L1 y levanta una bandera para que el dato también sea escrito en la cache L2 y en la memoria principal. Además invalida los datos de la otras cache que ya lo estuvieran usando.
- **Read:** Aquí se pregunta si el dato esta en la L1, en caso de que el dato no este, se pregunta en la L2 local y si esta tampoco lo tiene se le pregunta a la L2 del otro procesador y si aquí no se encuentra, el dato se trae desde memoria.
- **Calc:** como esta operacion no tiene que ver nada con las cache ni con la memoria principal el procesador simplemente se queda realizado la operacion por un tiempo dado.

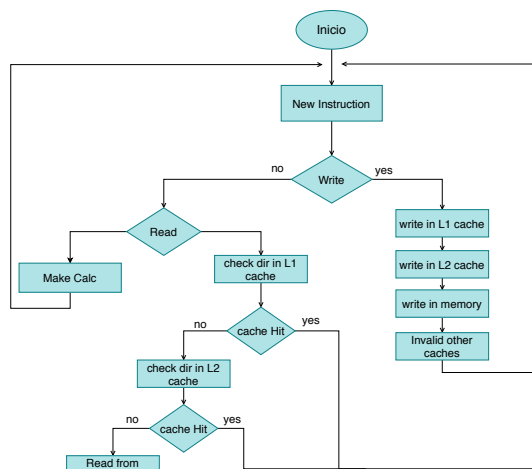


Figura 3: Diagrama de flujo del sistema.

### III. RESULTADOS

Una vez realizado toda la programación del sistema se obtuvo el el resultado de la imagen 4 en la cual se muestra de manera gráfica en forma de tablas cada una de las cache L1 de los cuatro núcleos del sistema, así como las tablas para las dos cache L2 y La memoria principal.

En la imagen se puede observa como se tienen los estados del protocolo MSI según la image 5 además en las cache L2 se muestra cual es el núcleo que esta haciendo uso del bloque o bien si se esta compartiendo de manera externa con el otro procesador.

**Cache L1 P0, 0**  
Nueva instrucción P0;0;CALC generada para P0,0

Bloque	Coherencia	Dir. Mem	Dato
0	S	0010	0000
1	S	0001	0000

**Cache L1 P1, 0**  
Inicio de calculo en P1,0

Bloque	Coherencia	Dir. Mem	Dato
0	S	0110	0000
1	M	1001	e95a

**Cache L1 P0, 1**  
Nueva instrucción P0;1;READ;0101 generada para P0,1

Bloque	Coherencia	Dir. Mem	Dato
0	S	0000	0000
1	S	0001	0000

**Cache L1 P1, 1**  
Nueva instrucción P1;1;CALC generada para P1,1

Bloque	Coherencia	Dir. Mem	Dato
0	S	0110	0000
1	M	1001	e95a

**Cache L2, P0**

Bloque	Estado	Dueño	Dir. Memoria	Dato
0	DS	P0,1	0000	0000
1	DS	P0,1	0001	0000
2	DS	P0,1	0111	0000
3	DS	P0,1	0111	0000

**Cache L2, P1**

Bloque	Estado	Dueño	Dir. Memoria	Dato
0	DM	P1,1	1001	e95a
1	DS	P1,1	0110	0000

**Memoria Principal**

Bloque	Dueño	Dato
0000	P0	0000
0001	P0	0000
0010	P1	0000
0011		0000
0100		0000
0101		0000
0110	P1, P0	0000
0111	P0	0000
1000	P0	0000
1001	P1	e95a

Figura 4: Interfaz Gráfica.

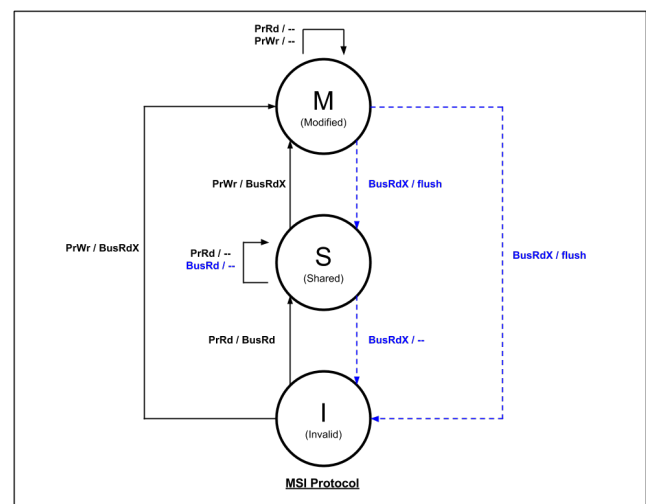


Figura 5: Diagrama de MSI.

Además, se crea un archivo con los log del sistema para que quede documentados todos los eventos que ocurren durante la ejecución del programa. Dicho log se observa en la imagen

```
jun 02, 2020 1:34:40 AM Logging.Log newInfo
INFORMACIÓN: Nueva instruccion P0;1;CALC generada para P0,1

jun 02, 2020 1:34:40 AM Logging.Log newInfo
INFORMACIÓN: Nueva instruccion P1;0;READ;0110 generada para P1,0

jun 02, 2020 1:34:40 AM Logging.Log newInfo
INFORMACIÓN: Nueva instruccion P1;1;WRITE;1001;a95a generada para P1,1

jun 02, 2020 1:34:40 AM Logging.Log newInfo
INFORMACIÓN: Nueva instruccion P0;0;READ;1000 generada para P0,0

jun 02, 2020 1:34:42 AM Logging.Log newInfo
INFORMACIÓN: Inicio de calculo en P0,1

jun 02, 2020 1:34:51 AM Logging.Log newInfo
INFORMACIÓN: Fin de calculo en P0,1

jun 02, 2020 1:34:52 AM Logging.Log newInfo
INFORMACIÓN: Nueva instruccion P0;1;READ;0111 generada para P0,1

jun 02, 2020 1:34:52 AM Logging.Log newInfo
INFORMACIÓN: Dato no esta en L1 de P0,0
```

Figura 6: Log del sistema.

#### IV. CONCLUSIONES

Los protocolos de coherencia de caché son sumamente necesarios para el desarrollo de los sistemas computacionales ya que si se desea un buen funcionamiento del sistema en general, el que la memoria no tenga fallos por la incoherencia es imprescindible.

Escribir en memoria inmediatamente que hay un cambio en la caché es beneficioso para mantener la cache coherente pero eso implica un mayor uso del bus y por ende disminuir el rendimiento del sistema en general. Por otro lado, escribir hasta que se de un fallo de cache es mas propenso a tener fallos de coherencia, pero ayuda al rendimiento del sistema en general. A la hora de diseñar un sistema se debe escoger cual usar con mucho cuidado.

Seleccionar una buena herramienta para el desarrollo de la interfaz gráfica de usuario puede significar un ahorro significativo de tiempo. Además se puede crear mejores vistas si se determina la herramienta correcta.

#### REFERENCIAS

- [1] Hennesy, J. And Patterson, David. Computer Architecture: A Quantitative Approach. 5th Edition. Elsevier – Morgan Kaufmann. 2012.
- [2] “Coherencia de Cachés” Universidad Europea de Madrid. [Online]. Available: [http://www.cartagena99.com/recursos/alumnos/apuntes/INF\\_EST\\_COM\\_U6\\_R2\\_T\\_PDF.pdf/](http://www.cartagena99.com/recursos/alumnos/apuntes/INF_EST_COM_U6_R2_T_PDF.pdf/). [Accessed: 24-May-2020].