

Instituto Tecnológico de Costa Rica

Área Académica de Ingeniería en Computadores

(Computer Engineering Academic Area)

**Programa de Licenciatura en Ingeniería en
Computadores**

(Licentiate Degree Program in Computer Engineering)

Curso: CE-4302 Arquitectura de Computadores II

(Course: CE-4302 Computer Architecture II)



Especificación Proyecto I

(Project I specification)

Profesor:

(Professor)

Ing. Luis Barboza Artavia

Fecha de entrega: 3 de abril 2020

(Due Date: April 3rd, 2020)

Proyecto I. Protocolos de coherencia en sistemas multiprocesador

1. Objetivo

Mediante el desarrollo de este proyecto, el estudiante aplicará los conceptos de coherencia de caché y sistemas multiprocesador en el diseño de un modelo en software con un protocolo basado en directorios y monitoreo para un sistema multiprocesador simultáneo.

2. Atributos relacionados

A continuación se describen los atributos del graduado que se pretenden abordar con el desarrollo del proyecto.

2.1. Diseño (DI)

Capacidad para diseñar soluciones de problemas complejos de ingeniería, con final abierto y diseñar sistemas, componentes o procesos que cumplan con necesidades específicas, considerando la salud pública, seguridad, estándares pertinentes, así como los aspectos culturales, sociales, económicos y ambientales.

El atributo de diseño será evaluado tanto formativamente (reuniones de seguimiento con el profesor) como sumativamente, en especial en la sección de documentación de diseño de los entregables.

2.2. Habilidades de comunicación (HC)

Capacidad para comunicar conceptos complejos de Ingeniería, dentro de la profesión y con la sociedad en general. Estas habilidades incluyen: la habilidad de comprender y escribir efectivamente informes, documentación de diseños, realizar presentaciones efectivas, dar y recibir instrucciones claras. Es conveniente incentivar la capacidad de comunicarse en un segundo idioma.

El atributo de habilidades de comunicación será evaluado sumativamente en las secciones de artículo (*paper*) y video de presentación de los resultados de los entregables.

3. Descripción general

Los sistemas multiprocesador corresponden a la base funcional de los computadores modernos. Se pueden encontrar en cualquier sistema como los en chip, embebidos y *clusters* que permiten una computación de alto desempeño y procesamiento masivo en paralelo. Uno de los mayores retos que genera tener múltiples procesadores es la comunicación entre ellos. Los diseñadores de los computadores actuales han buscado mecanismos para optimizar la capacidad para compartir tareas y recursos entre los diferentes procesadores. La memoria se comparte con los demás procesadores y corresponde a uno de los más importantes, puesto que se compone de los datos necesarios para ejecutar las instrucciones. Las interfaces de memoria compartida representan un desafío porque múltiples procesadores pueden solicitar un mismo dato en un instante de

tiempo. Por lo tanto, el problema de incoherencia de caché todavía se investiga en el área de arquitectura de computadores.

Para este proyecto se deberán aplicar los conceptos de arquitectura de computadores, vista como una combinación de elementos de software y hardware, en el diseño de un modelo de software (plataforma de simulación) de un sistema multiprocesador con memoria compartida a través de memoria caché local L1 para cada núcleo y una memoria caché L2 compartida entre los núcleos del procesador, así como el protocolo de coherencia basado en directorios y monitoreo.

En el proyecto se desarrollará un acercamiento práctico al diseño de interfaces de memoria compartida y sincronización, así como diseño, integración y programación de sistemas computacionales en general.

4. Especificación

Para este proyecto se deberá diseñar una aplicación de software que realice un modelo de un sistema multiprocesador con dos procesadores (chip) que a su vez tienen dos elementos de procesamiento (núcleo). Estos componentes se encuentran conectados a una memoria compartida, a través de una una caché L2 para cada chip y una caché L1 para cada núcleo como se muestra en la figura 1. Ambas memorias caché son mapeadas directamente. Cada núcleo deberá generar solicitudes de procesamiento o acceso a memoria (lectura o escritura) a diferentes regiones de memoria. Sobre las memorias caché se deberá implementar el protocolo basado en directorios para solucionar el problema de las incoherencias en regiones compartidas de memoria.

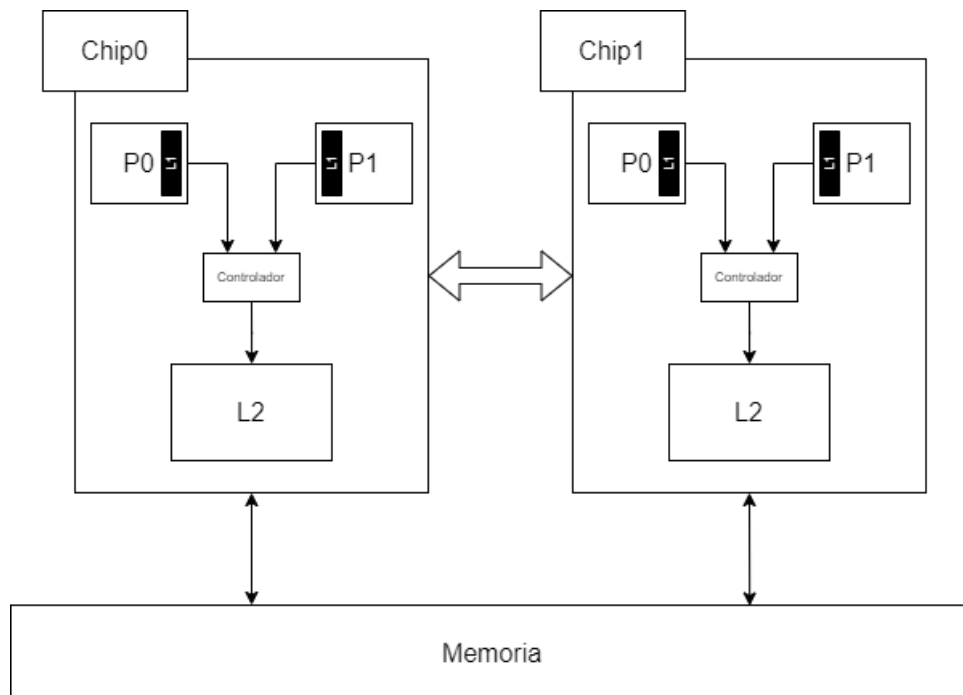


Figura 1: Arquitectura general del sistema.

A continuación se describe a mayor detalle la especificación del proyecto.

Modelo del sistema multiprocesador

Para el modelo del sistema multiprocesador, deberá crearse una instancia del procesador que tendrá cada núcleo de manera independiente y paralela, por ejemplo con hilos. Cada núcleo deberá generar, individualmente y de manera simultánea con los demás núcleos, instrucciones de procesamiento (no requiere memoria), escritura o lectura a uno de 16 bloques totales de la memoria compartida. Cada una de las instrucciones debe generarse utilizando una distribución de probabilidad formal (binaria, de Poisson, hipergeométrica, etc). Una vez obtenida la instrucción, la dirección del bloque de memoria también será asignada de manera aleatoria con una distribución previamente documentada y justificada.

Una instrucción generada debe tener los siguientes componentes:

- Número de núcleo.
- Número de chip.
- Operación: read, write y calc.
- En el caso de *write* se debe indicar la dirección de memoria y el dato a escribir, respectivamente.

Una instrucción tiene la siguiente estructura:

```
P0,0: READ 0100  
P1,1: CALC  
P1,0: WRITE 1010;4A3B
```

El modelo del sistema deberá proveer una base de temporización para la generación de los eventos de procesamiento, lecturas y escrituras a definir por cada estudiante. Dicha base debe permitir la correcta visualización de todas las acciones generadas por el sistemas multiprocesador y sistema de coherencia de caché.

En todo momento deberá mostrarse la acción de cada núcleo, así como el bloque de memoria que está utilizando. Cada núcleo será identificado por el número de chip y el número de núcleo. Además, todos los eventos deben almacenarse en un archivo `log` que registre cada ejecución de instrucción y la acción que se realizó por parte del protocolo encargado de manejar la coherencia.

Modelo de memoria principal

La memoria principal será un modelo de una memoria unificada y compartida, a través de un único bus. La funciones del modelo memoria principal serán: actualizar el contenido compartido de los bloques en escrituras (según política de escritura establecida) y permitir la visualización de los 16 bloques en todo momento. La información que contiene es:

- Dirección de memoria.

- Estado del bloque: DM (directorio modificado), DS (directorio compartido) y DI (directorio inválido).
- Chips que son dueños del bloque.
- Dato de 32 bits.

El sistema deberá simular un retardo propio de lectura o escritura a memoria, cuando corresponda. El tiempo debe ser proporcional a la frecuencia elegida anteriormente, de manera que exista el problema de que el procesador es más rápido que la memoria.

Al iniciar la aplicación, el contenido de los bloques de memoria deben ser 0. La representación gráfica de la memoria se muestra en la figura 2.

M			
	Estado	Dueño(s)	Dato
1011	DM	C0	FFFF
0110	DS	C0, C1	FFFF
0111	DI	C1	FFFF
...
1000	DS	C0	FFFF

Figura 2: Modelo de la memoria principal.

Modelo de memoria caché y sistema de coherencia

Para esta sección se deberá diseñar un modelo de caché para cada núcleo y chip, así como los protocolos de coherencia tanto basado en directorios como monitoreo. A continuación se explican los diversos componentes.

Memoria caché L1

Está situada en cada uno de los núcleos, se pueden almacenar 2 bloques y posee correspondencia directa. Debe implementar monitoreo con el fin de controlar la coherencia entre las memorias caché dentro de un mismo chip. El modelo de la memoria caché L1 se detalla en la figura 3. La información que contiene es:

- Número de bloque.
- Estado de coherencia: M (modificado), S (*shared*) y I (inválido).
- Dirección de memoria.
- Dato de 32 bits.

P0, 0			
	Estado de coherencia	Dirección de memoria	Dato
0	M	1011	FFFF
1	S	0110	FFFF

Figura 3: Modelo de la memoria caché L1

Memoria caché L2

Esta memoria es compartida por cada uno de los chips, por lo que cada núcleo en un chip se conecta a ella por medio de una interface cuya función es servir como controlador. El espacio disponible corresponde a 4 bloques y posee correspondencia directa. El modelo de la memoria caché L2 se detalla en la figura 4. La información que contiene es:

- Número de bloque.
- Estado del directorio: DM (directorio modificado), DS (directorio compartido) y DI (directorio inválido).
- Lista del núcleo dueño de la dirección, así como si se comparte de manera externa con otro chip.
- Dirección de memoria.
- Dato de 32 bits.

L2, 0				
	Estado	Dueño	Dirección de memoria	Dato
0	DM	P0,1	1011	FFFF
1	DS	P0,0;E	0110	FFFF
2	DI	P1,0	0111	FFFF
3	DS	P1,0	1000	FFFF

Figura 4: Modelo de la memoria caché L2

Se asumirá que si el dato es referenciado por primera vez, este no se encuentra en caché. La caché deberá generar alertas de *misses* tanto por escritura como por lectura. Estos eventos deben registrarse en el archivo `log`. Además, deberá permitir la visualización del contenido de los 8 bloques de cada chip, por cada nodo, en todo momento.

Protocolo basado en directorios

Se debe diseñar un modelo del sistema de coherencia basado en directorios con el fin de asegurar la coherencia entre todas las caché. Para ello, debe apoyarse en el uso de las caché L2 de ambos chips con el fin de implementar el directorio. Dentro del archivo `log` se debe incluir la acción que realizó el controlador de la coherencia para conocer qué otros directorios quedaron inválidos al realizar una escritura o lectura.

Al iniciar la aplicación el contenido de los bloques de caché debe ser 0.

Protocolo de monitoreo

Se debe diseñar un protocolo de monitoreo con el fin de asegurar la coherencia en los núcleos dentro del chip. Para ello, debe apoyarse en las memorias caché de cada núcleo. Dentro del archivo `log` se debe incluir la acción que realizó el controlador de la coherencia para conocer qué otros datos quedaron inválidos al realizar una escritura o lectura.

Al iniciar la aplicación el contenido de los bloques de caché debe ser 0.

Notas adicionales

- El desarrollo de este proyecto es individual.
- El lenguaje de programación o herramientas de desarrollo podrá ser elegido abiertamente por cada estudiante
- Todo diseño deberá tener al menos 2 propuestas detalladas adecuadamente y comparadas según criterios.

5. Entregables

Como entregables en este proyecto se evaluará lo siguiente:

- Presentación funcional completa (65 %): Para la presentación funcional, cada estudiante deberá grabar y entregar un video realizando las diferentes pruebas sobre la herramienta (grabación de la pantalla) y explicando el funcionamiento interno (protocolo utilizado, escenarios de ejecución, contenido de memoria, estado de bloques, etc.) y limitantes. El profesor evaluará las pruebas según rúbrica correspondiente. En caso de que haya duda de cualquiera de las partes, el estudiante o el profesor podrá solicitar una cita para la defensa del proyecto de forma presencial. Se deberá subir un archivo con el video o un link al video correspondiente junto con un enlace al repositorio del código fuente, en la sección de evaluaciones del TecDigital. Por motivos de acreditación del programa, se recomienda vehementemente que el enlace proporcionado esté disponible por cualquier persona que tenga acceso a él, hasta un año después de entregada esta evaluación.
- Artículo tipo *paper* (10 %): El paper a realizar deberá tener una extensión no mayor a 4 páginas, deberá ser realizado con \LaTeX , siguiendo el formato IEEE Transactions. En general el *paper* deberá contar con las siguientes secciones:

- Abstract e introducción
 - Sistema desarrollado
 - Resultados
 - Conclusiones
 - Referencias
- Presentación del *paper* y resultados (5 %): Cada persona deberá grabar y entregar un video de no más de 5 minutos, presentando la idea de su solución (puede utilizar diapositivas o algún otro medio de referencia). Debe considerar que el público meta de su presentación no tiene necesariamente el *background* técnico, por lo que deberá exponer de forma clara lo que se ha realizado, así como los resultados más importantes de su diseño. Puede utilizar el canal de Youtube “Two Minute Papers” <https://www.youtube.com/user/keeroyz>, como referencia. Por motivos de acreditación del programa, se recomienda vehementemente que el enlace proporcionado esté disponible por cualquier persona que tenga acceso a él, hasta un año después de entregada esta evaluación.
 - Documentación de diseño (20 %): La documentación del diseño deberá contener las siguientes secciones:
 - Listado de requerimientos del sistema: Cada estudiante deberá determinar los requerimientos de ingeniería del problema planteado, considerando partes involucradas, estado del arte, estándares, normas, entre otros.
 - Elaboración de opciones de solución al problema: Para el problema planteado deberán documentarse al menos dos opciones de solución. Cada solución deberá ser acompañada de algún tipo de diagrama.
 - Comparación de opciones de solución: Se deberán comparar explícitamente las opciones de solución, de acuerdo con los requerimientos y otros aspectos aplicables de salud, seguridad, ambientales, económicos, culturales, sociales y de estándares.
 - Selección de la propuesta final: Se deberá seleccionar una propuesta final de las opciones de solución, de acuerdo con los criterios de comparación.
 - Implementación del diseño: Se deberá documentar completamente el diseño final seleccionado. Para el caso de este proyecto esto incluye: descripción del protocolo diseñado, diagrama de bloques del modelo sistema multiprocesador, diagrama de bloques del computador (procesadores + memoria y aplicación), diagramas propios de diseño de software aplicables (de flujo, clases, composición, UML, patrones de diseño, etc) y descripción de algoritmo propuesto y distribuciones de probabilidad implementadas.