	Universidad de Costa Rica Escuela de ingeniería Eléctrica Experimento 6	EIE Escuela de Ingeniería Eléctrica
IE0424: Laboratorio de Circuitos Digitales II-2020		

Objetivo General

Aplicar los conceptos de arquitectura de computadoras para el diseño de jerarquías de memoria.

Objetivos específicos

- Analizar interfaces de memoria de un CPU e implementar un cache asociativo.
- Recolectar y analizar datos de rendimiento de un cache.

Preguntas a responder en el anteproyecto.

¿Qué es la asociatividad de un cache? ¿En qué afecta?

Mencione y explique 3 políticas de reemplazo diferentes. ¿Cuál es la importancia de las políticas de reemplazo?

Investigue acerca de cómo funciona la generación de números aleatorios usando registros de desplazamiento (LSFR o Linear feedback shift register).

Propuesta del problema.

Proceda ahora a obtener el código inicial del proyecto. Para ello, siga el siguiente enlace:

```
https://classroom.github.com/g/O6oHe3Cx
```

De la misma forma que en los Laboratorios anteriores, autorice la aplicación, busque su nombre (si su nombre no aparece, contacte al profesor) y seleccione un equipo (si su equipo no aparece listado, proceda a crear uno). Una vez finalizado un repositorio debió haber sido creado. Este repositorio va a ser el repositorio con el que va a trabajar y va a ser uno de los entregables de este laboratorio.

Proceda a clonar el repositorio que ha sido creado:

```
$ git clone https://github.com/ucr-ie-0424/...
```

La dirección completa de su repositorio debe aparecer seleccionando el botón verde que dice *Code.*

El código que se provee es el mismo código que se usó como punto de inicio de los Laboratorios anteriores.

El objetivo de este experimento es diseñar una cache asociativa con políticas de reemplazo LRU y pseudo-aleatorio y estrategia de escritura de writeback.

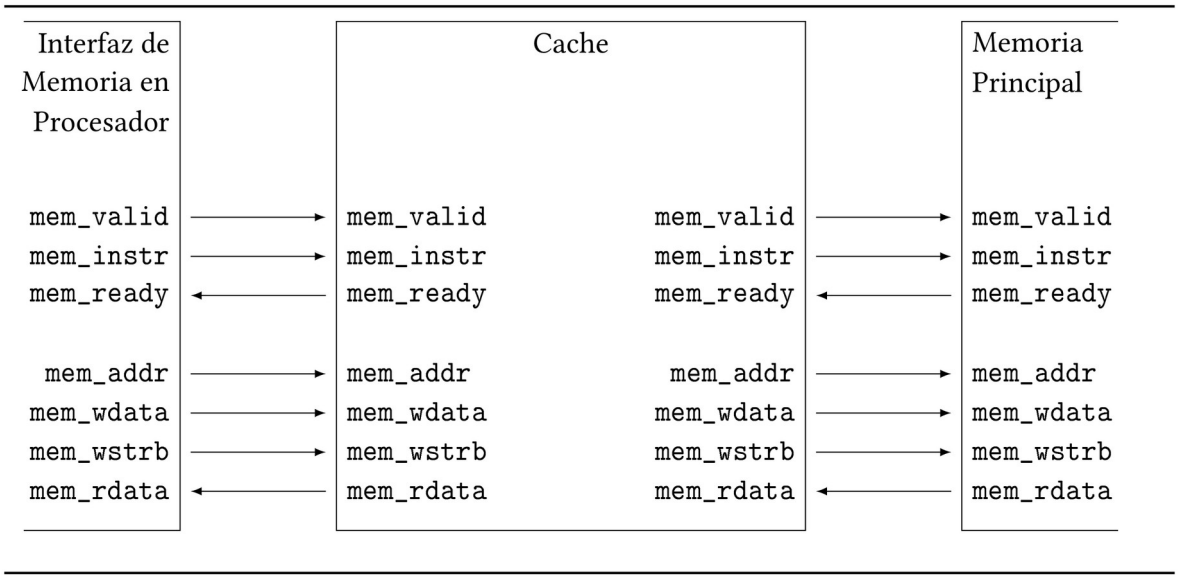
Ejercicio 1 (Obligatorio):

Implemente un cache asociativo de 2 ways con política de reemplazo aleatorio y estrategia de escritura write-back.

El diseño de su cache deberá soportar parámetros que definan el tamaño del bloque y el tamaño del cache. Inicialmente utilice un tamaño de bloque de 2 palabras y un tamaño de cache de 1 kB.

Asimismo, deberá tener contadores de hits y misses de forma tal que se pueda luego documentar los rates.

Recuerde del Laboratorio 5 el siguiente diagrama como referencia para la conexión del cache:



Asimismo recuerde definir en 0 el parámetro de *FAST_MEMORY* de *system.v*.

El cache a implementar, debe de estar instanciado en system.v y el módulo debe de tener el nombre *cache_2_way_random*. Asimismo debe de contener las entradas y salidas como se documentaron en el diagrama anterior.

Para la política de reemplazo del bloque, use una implementación de un módulo que use LSFR para generar números aleatorios.

Una vez implementado, utilice el firmware del Ejercicio 1 del Laboratorio 5 y documente el hit y miss rate para el recorrido completo de la lista enlazada para detectar los últimos 5 datos impares.

Agregue en el reporte la cantidad de slices que el diseño consume y documente los 5 últimos 5 datos impares que su firmware encontró.

Ejercicio 2 (Obligatorio):

Utilice los siguientes valores de tamaño de bloque y tamaño de cache (cambiando los parámetros del diseño del cache del Ejercicio 1):

Tamaño de bloque	Tamaño del cache
2 palabras	2 kB
4 palabras	2 kB
2 palabras	4 kB
4 palabras	4 kB

Asimismo, para cada uno de las entradas de la tabla anterior, utilice el firmware y recorra la lista para encontrar los últimos 5 datos impares. Documente el hit/miss rate y compare los valores obtenidos.

Anote la cantidad de slices que consume el diseño con cada uno de los diferentes parámetros con los que sintetizó su cache.

Ejercicio 3 (Obligatorio):

Implemente un cache asociativo de 4 ways con política de reemplazo LRU y estrategia de escritura write-back.

El diseño de su cache deberá soportar parámetros que definan el tamaño del bloque y el tamaño del cache. Inicialmente utilice un tamaño de bloque de 2 palabras y un tamaño de cache de 1 kB.

Asimismo, deberá tener contadores de hits y misses de forma tal que se pueda luego documentar los rates.

El cache a implementar, debe de estar instanciado en system.v y el módulo debe de tener el nombre *cache_4_way_lru*. Asimismo debe de contener las entradas y salidas como se documentaron en el Ejercicio 1.

Una vez implementado, utilice el firmware del Ejercicio 1 del Laboratorio 5 y documente el hit y miss rate para el recorrido completo de la lista enlazada para detectar los últimos 5 datos impares.

Agregue en el reporte la cantidad de slices que el diseño consume y documente los 5 últimos 5 datos impares que su firmware encontró.

Ejercicio 4 (Obligatorio):

Utilice los siguientes valores de tamaño de bloque y tamaño de cache (cambiando los parámetros del diseño del cache del Ejercicio 3):

Tamaño de bloque	Tamaño del cache
2 palabras	2 kB
4 palabras	2 kB
2 palabras	4 kB
4 palabras	4 kB

Asimismo, para cada uno de las entradas de la tabla anterior, utilice el firmware y recorra la lista para encontrar los últimos 5 datos impares. Documente el hit/miss rate y compare los valores obtenidos.

Anote la cantidad de slices que consume el diseño con cada uno de los diferentes parámetros con los que sintetizó su cache.

Ejercicio 5 (Opcional):

Reemplace la memoria instanciada en *system.v* por la memoria DDR2 que la tarjeta Nexys 4 DDR contiene. Para ello consulte el capítulo 4.1 del Manual de Referencia de la tarjeta.