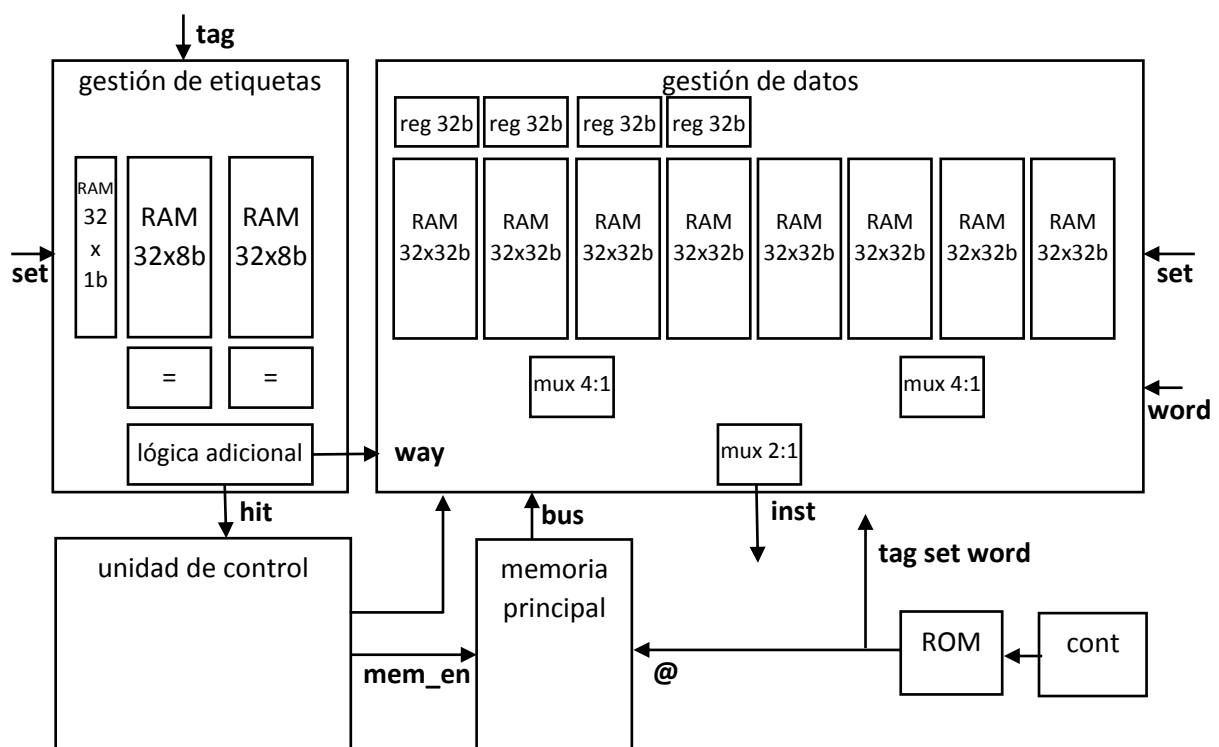


DISEÑO DE UNA MEMORIA CACHE DE INSTRUCCIONES

Práctica 2 – sesión 5



Arquitectura y Organización de Computadores 2
2º Grado Ingeniería Informática

Luis M. Ramos
luisma@unizar.es

Alejandro Valero
alvabre@unizar.es

José Luis Briz
briz@unizar.es

Javier Resano
jresano@unizar.es



Escuela de
Ingeniería y Arquitectura
Universidad Zaragoza



Departamento de
Informática e Ingeniería
de Sistemas
Universidad Zaragoza

1 RESUMEN

En esta práctica vamos a diseñar una memoria cache de instrucciones. Para probarla utilizaremos un circuito que irá generando direcciones. La memoria cache debe comprobar si cada dirección es un acierto o fallo y en caso de fallo solicitar a memoria principal el envío del bloque correspondiente.

El objetivo es entender cómo se descompone la dirección para obtener etiqueta, conjunto y palabra y la gestión de fallos.

Al entrar al laboratorio coloca el trabajo previo del apartado 2.1 encima de la mesa, de forma visible. Antes de empezar a trabajar con Logisim conéctate a Moodle desde uno de los equipos del laboratorio y realiza el control de asistencia a la sesión.

La práctica finaliza cuando el banco de registros funciona correctamente y ha sido entregado a través de Moodle el circuito del apartado 2.2.

2 ESPECIFICACIÓN DE LA MEMORIA CACHE

2.1 TRABAJO PREVIO: DISEÑO EN PAPEL

Características de la memoria cache de instrucciones a diseñar:

- tamaño de 1 KB
- asociativa con 2 vías
- acceso en paralelo a etiquetas y datos
- tamaño de bloque 16 bytes
- política de reemplazo FIFO
- tamaño de instrucción 32 bits

Se dispone además de una memoria principal de 64 KB, direccionable a nivel de byte. Se usará un contador y una memoria ROM para generar la traza de direcciones de instrucciones que generaría el procesador.

Descripción del funcionamiento (ver cronograma):

- Si la dirección solicitada está disponible en memoria cache, se activa la señal **hit** y se envía la instrucción al procesador a través de la salida **inst**. Todo esto ocurre en un solo ciclo.
- Si la instrucción no está disponible, la señal **hit** permanece desactivada y se activa la señal **mem_en**, para solicitar el bloque correspondiente a la memoria principal. Todo esto ocurre en el mismo ciclo (0). La memoria principal funciona de forma síncrona. El bloque solicitado se recibirá a través de **bus** (en cuatro palabras de 32 bits) en los ciclos 4, 6, 8 y 10. La señal **mem_en** debe permanecer activa hasta que acabe todo el proceso. Las cuatro palabras se irán cargando en los cuatro registros de 32 bits. En el ciclo 11, las cuatro palabras se escriben en los módulos de memoria correspondientes. En el ciclo 12 se vuelve a realizar el acceso a la instrucción, que ahora debe producir acierto.

ciclo	0	0	1	2	3	4	5	6	7	8	9	10	11	12
@	@ ₁	@ ₂	@ ₂	@ ₂	@ ₂	@ ₂	@ ₂	@ ₂	@ ₂	@ ₂	@ ₂	@ ₂	@ ₂	@ ₂
hit	1	0	0	0	0	0	0	0	0	0	0	0	0	1
mem_en	0	1	1	1	1	1	1	1	1	1	1	1	1	0
inst	inst ₁													inst ₂
bus						word ₀		word ₁		word ₂		word ₃		

Para el diseño de la memoria cache disponemos de ocho módulos de memoria RAM de 32x32b (almacenamiento de datos), dos módulos RAM 32x8b (almacenamiento de etiquetas y bit de validez), un módulo RAM 32x1b (indica la próxima vía a reemplazar en cada conjunto), dos multiplexores 4:1x32, un multiplexor 2:1x32, dos comparadores de 8 bits y un decodificador (para activar la escritura en el ciclo 11 en la vía correspondiente). Ver esquema de la portada.

- Identifica qué bits de la dirección se usarán para elegir la etiqueta, el conjunto y la palabra.
- Realiza las conexiones entre los distintos componentes del diseño.
- Dibuja el autómata de la unidad de control de la memoria cache, de forma que siga el funcionamiento descrito.

2.2 TRABAJO EN EL LABORATORIO: DISEÑO CON LOGISIM

- Crea una hoja nueva de Logisim y nómbrala NIA-cache.circ.
- Carga la librería con el componente de memoria principal (Proyecto / Cargar Librería / Librería Logisim / MP.circ) disponible en la carpeta Moodle *Logisim*. Para facilitar la depuración de la memoria cache hemos programado la memoria principal con instrucciones falsas. La primera instrucción es 0x00000000 y ocupa las direcciones de memoria (@0, @1, @2 y @3); la siguiente instrucción es 0x00000004 y ocupa las direcciones @4, @5, @6 y @7; la siguiente 0x00000008 en @8, @9, @10 y @11; ...
- Coloca todos los componentes: contador, ROM, memoria principal (componente MP), módulos de memoria RAM, registros, comparadores, multiplexores y decodificador. Interconéctalos según tu diseño.
- Implementa la unidad de control de la memoria cache. Puedes hacer una implementación usando un contador y un decodificador, similar a la usada para controlar la memoria principal (ver diseño del módulo MP), o usar un diseño tradicional: un registro y dos ROM.
- Programa la ROM con la secuencia de direcciones que quieras probar.
- Depura tu diseño hasta que funcione correctamente.
- Programa la ROM con la secuencia de direcciones que obtengas en la página webdiis.unizar.es/~luisma/aoc2/nia3.php a partir de tu NIA. Comprueba que funciona correctamente.
- Finalmente, entrega tu circuito a través del recurso Moodle “Entrega NIA-cache.circ”. Antes de entregar comprueba que usas un único componente Reloj.