

## 1. Ejercicios

### 1.1. Documente qué significa cada una de las señales que forman la interfaz de memoria de picorv32.

Hay 7 señales en la interfaz de memoria del core de CPU picorv32[1]:

- **mem\_valid**: Esta señal de un bit funciona para señalar que el core esta iniciando una transferencia de memoria. Esta se mantiene en alto hasta que se active la señal **mem\_ready**.
- **mem\_instr**: Esta señal de un bit funciona en el caso de un instruction fetch para indicar que se esta haciendo esta operación.
- **mem\_ready**: Esta señal de un bit se utiliza para indicarle al core que la memoria esta lista para ser utilizada.
- **mem\_addr**: Esta señal de 32 bits contiene la dirección en memoria que esta siendo accedida.
- **mem\_wdata**: Esta señal de 32 bits contiene la información en 32 bits a escribir en memoria en caso de una operación de escritura.
- **mem\_wstrb**: Esta señal de 4 bits se utiliza para determinar si se esta operando una lectura, cuando esta en cero, o una escritura, cuando esta en otro valor distinto de cero. Dependiendo de que bits estén en alto en esta señal se escribirá en solo ciertos bits de la memoria. Los valores posibles son 0000, 1111, 1100, 0011, 1000, 0100, 0010, y 0001.
- **mem\_rdata**: Esta señal es el lugar donde se escribe la información actualmente en memoria en una posición particular.

### 1.2. ¿Qué es el miss/hit rate de un cache?

El hit rate o miss rate se refieren a la tasa de aciertos o fallos en la búsqueda de un dato en un cache. SI tengo un cache con un hit rate del 50 % eso implicaría que la mitad de las veces que se busco en el cache un dato el mismo se encontró. Similarmente se aplica para el miss rate. [2]

### 1.3. ¿Qué es un bloque? ¿Cómo se identifica un bloque dentro de un cache?

Un block de cache es la unidad básica para almacenamiento en un cache. Esto puede consistir de múltiples bytes o words dependiendo de como se diseñe el cache. Para identificar un bloque se usa el índice del cache en conjunto con el conocimiento de tamaño de bloque para identificar donde en memoria esta el inicio y final de cada bloque. En conjunto con esto, se utiliza el tag del bloque, un identificador único para un juego de datos, para confirmar si lo que se encontró corresponde a el bloque deseado. [3]

#### 1.4. ¿Cuál es la diferencia entre write-back y write-through?

La diferencia entre ambos métodos es que en el write through cada vez que hay un acierto, se actualiza, en paralelo, tanto la memoria cache como la memoria principal, esto provoca que haya un mayor numero de escrituras a memoria y que la memoria principal contenga siempre la misma información que la memoria cache. Mientras que en write back solo se actualiza el dato en la memoria cache, entonces la memoria cache y la memoria principal contienen datos diferentes y tiene menos cantidad de escrituras que el metodo write through.[4]

#### 1.5. Documente, cuál es el tamaño del byte offset, el tamaño del índice del bloque y el tamaño del tag para los siguientes tamaños de cache y bloque:

Tamaño de bloque	Tamaño del cache
2 palabras	1 kB
2 palabras	2 kB
4 palabras	2 kB
2 palabras	4 kB
4 palabras	4 kB

Figura 1: Tabla de valores a calcular

Dado que el tamaño del word en el picorv32 es de 4Bytes, entonces la tabla de valores es la siguiente.

Cuadro 1: Tamaño del offset, index y tag

Tamaño de bloque(Bytes)	Tamaño de cache(Bytes)	Cantidad de lineas de cache	Bits de Offset	Bits de Index	Bits de Tag
8	1024	128	6	7	51
8	2048	256	6	8	50
16	2048	128	7	7	114
8	4096	512	6	9	49
16	4096	256	7	8	113

Donde la cantidad de lineas de memoria cache se calculó como:

$$\#Lineas = \frac{Bytes\_cache}{Bytes/Bloque} \quad (1)$$

La cantidad de bits de offset, index y tag se calcularon con las siguientes ecuaciones, deducidas de [5]

$$offset = \log_2(Bytes/Bloque) \quad (2)$$

$$index = \log_2(\#Lineas) \quad (3)$$

$$tag = Bits/Bloque - offset - index \quad (4)$$

## Referencias

- [1] Yosys. *PicoRV32 - A Size-Optimized RISC-V CPU*. Yosys. 2022. URL: <https://github.com/YosysHQ/picorv32>.
- [2] Adam Novotný. *What is a Cache Hit Ratio and How do you Calculate it?* StormIT. 2022. URL: <https://www.stormit.cloud/post/cache-hit-ratio-what-is-it>.
- [3] Sat Garcia. *The Basics of Caches*. UC San Diego. 2007. URL: <https://cseweb.ucsd.edu/classes/su07/cse141/cache-handout.pdf>.
- [4] GeekForGeeks. *Differentiate between Write Through and Write Back Methods*. 2020. URL: <https://www.geeksforgeeks.org/differentiate-between-write-through-and-write-back-methods/>.
- [5] *How does cache work*. 2009. URL: <https://courses.cs.washington.edu/courses/cse378/09wi/lectures/lec15.pdf>.