

Practica 2

Juego Snake

Analizando Cache

- Juan Pablo Figueroa Martín 750901.
- Luis Pablo de los Reyes Rentería 750515.
- Carrera: Ingeniería en Ciberseguridad.
 - Maestro: Álvaro Gutiérrez Arce.
- Curso: Organización y Arquitectura de Computadoras.
- Fecha: 01 de Diciembre del 2024.

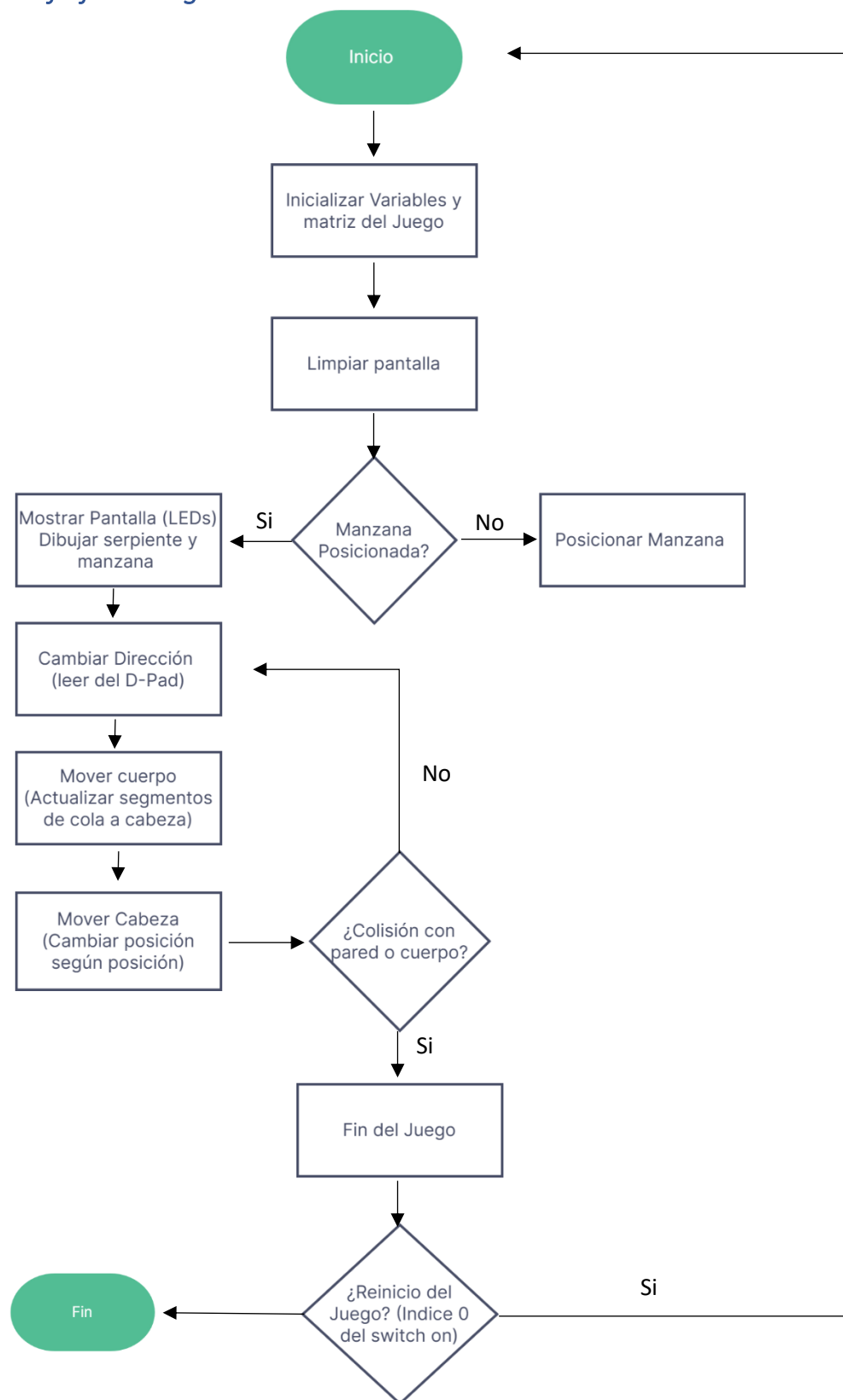


ITESO, Universidad
Jesuita de Guadalajara

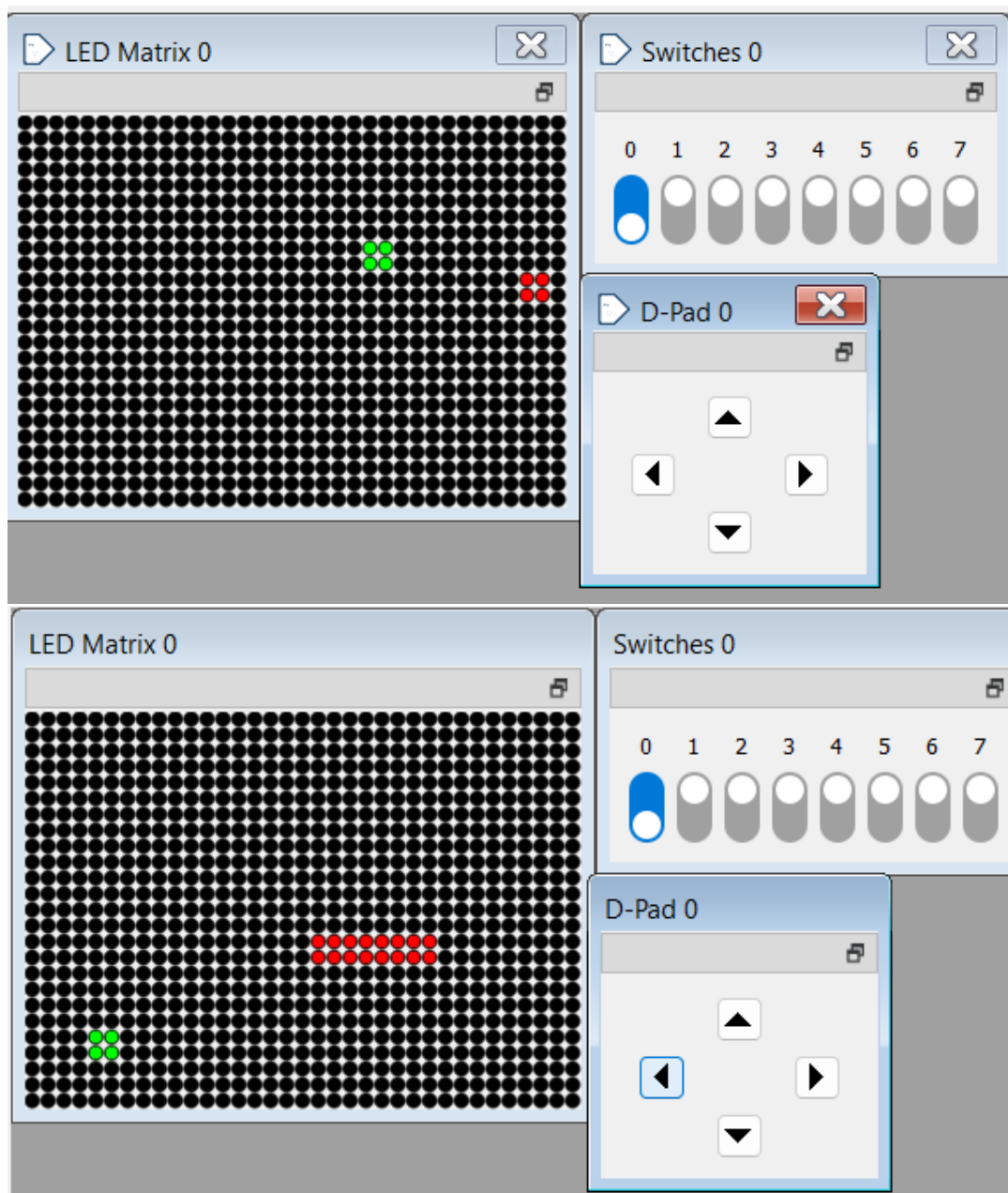
Índice

Índice.....	2
<i>Diagrama de flujo del Algoritmo.</i>	3
<i>Captura de Pantalla del Juego.</i>	4
<i>Pruebas en Memoria Cache.</i>	5
<i>Conclusiones de cada integrante del equipo.</i>	12
<i>Link de GitHub:</i>	12

Diagrama de flujo del Algoritmo.



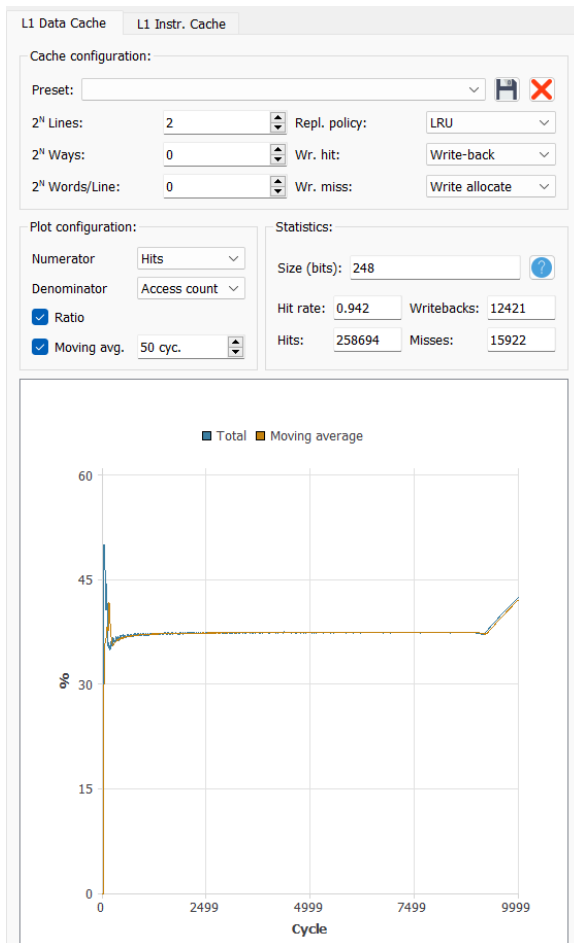
Captura de Pantalla del Juego.



Pruebas en Memoria Cache.

En base a las siguientes configuraciones de cache

- Mapeo directo 4 líneas

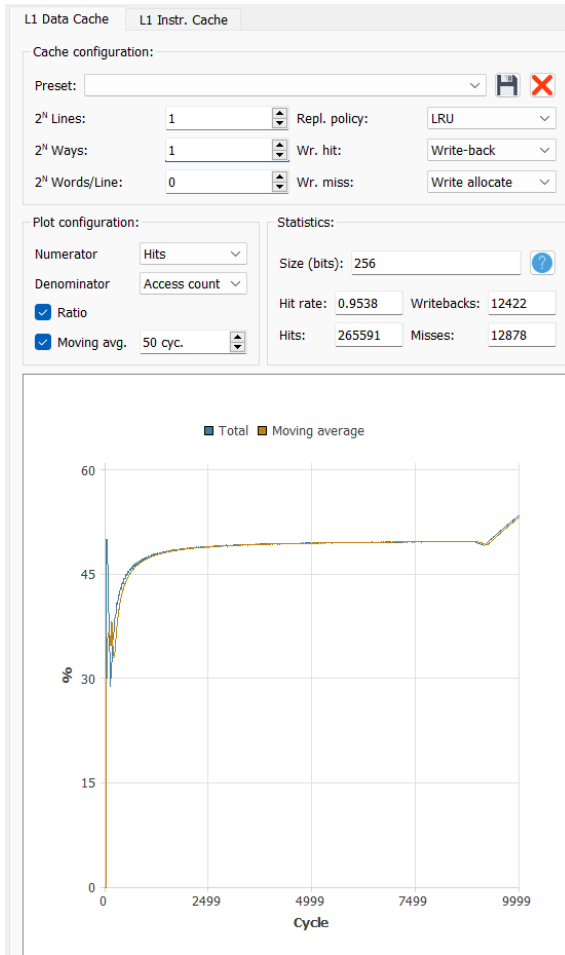


31 4 3 2 1 0

Access address

Index	V	D	Tag	Word 0
0	1	0	0x07ffffd7	0x0000000a
1	1	0	0x07fffffc	0x00000001
2	1	0	0x07fffffc	0x00000002
3	1	1	0x07ffffd5	0x0000123d

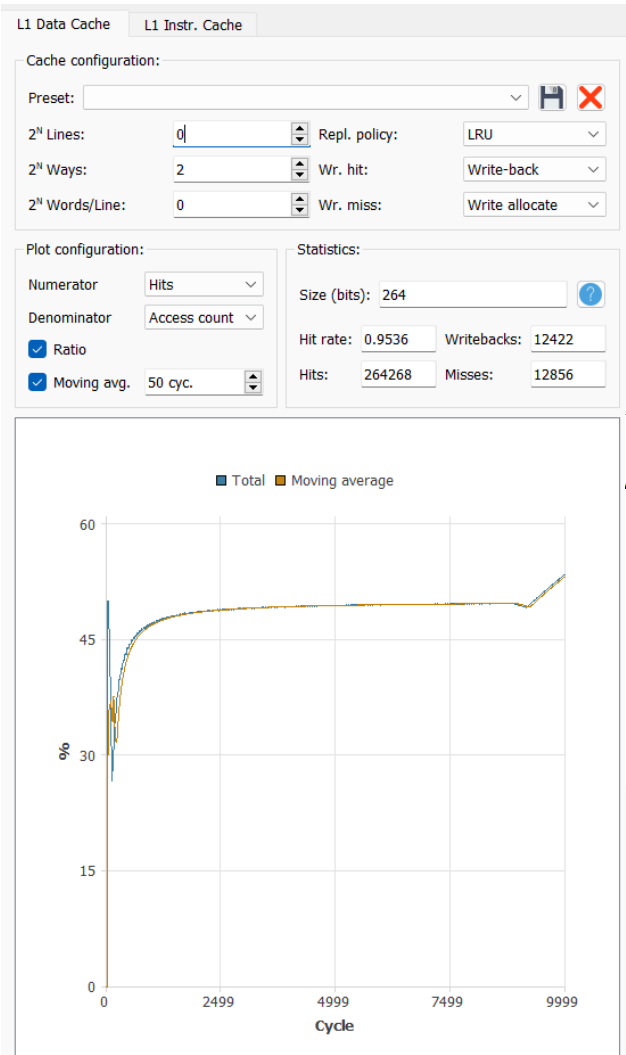
- Asociativa con 2 conjuntos (lines) y 2 vías (ways)



Access address 31 32 1 0

Index	V	D	LRU	Tag	Word 0
0	1	0	1	0x0ffffffae	0x0000000a
	1	0	0	0x0ffffff9	0x00000002
1	1	1	0	0x0ffffffab	0x000019c3
	1	1	1	0x0ffffffad	0x0000000c

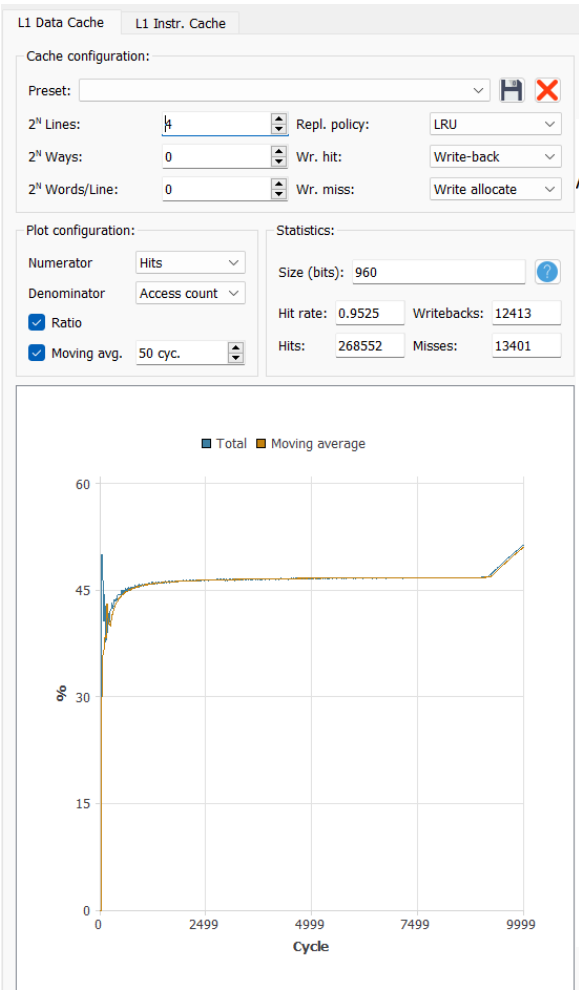
- Totalmente asociativa con 4 vías (ways)



Access address 31 21 0

Index	V	D	LRU	Tag	Word 0
0	1	1	2	0x1ffffff5b	0x0000000c
	1	0	1	0x1ffffff2	0x00000002
	1	1	0	0x1ffffff57	0x0000168e
	1	0	3	0x1ffffff1	0x00000001

- Mapeo directo 16 líneas

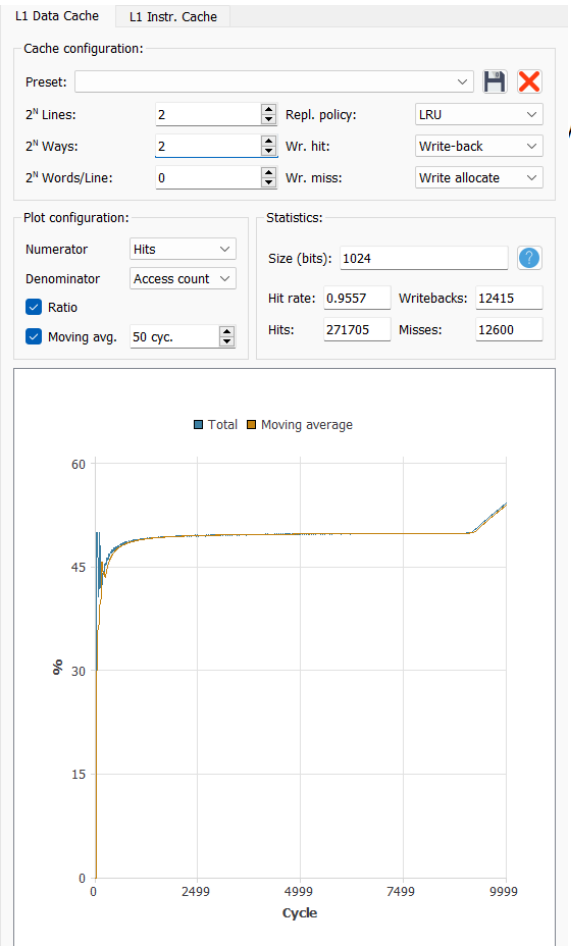


Access address 31 6 5 2 1 0

--	--	--	--

Index	V	D	Tag	Word 0
0	1	1	0x03c00036	0x00000000
1	1	0	0x01ffffff	0x00000001
2	1	1	0x01ffffff	0x00000002
3	1	1	0x01ffffff	0x00000006
4	1	1	0x03c0000d	0x0000ff00
5	1	1	0x03c0000d	0x0000ff00
6	1	1	0x01ffffff5	0x00000001
7	1	1	0x01ffffff5	0x00002091
8	1	1	0x03c0000f	0x0000ff00
9	1	0	0x0000047a	0xf000dbc
10	1	0	0x0000047a	0xf000db8
11	1	1	0x01ffffff5	0x0000000c
12	1	0	0x01ffffff5	0x0000000a
13	1	0	0x01ffffff5	0x00000003
14	1	0	0x03c00036	0x00000000
15	1	0	0x03c00036	0x00000000

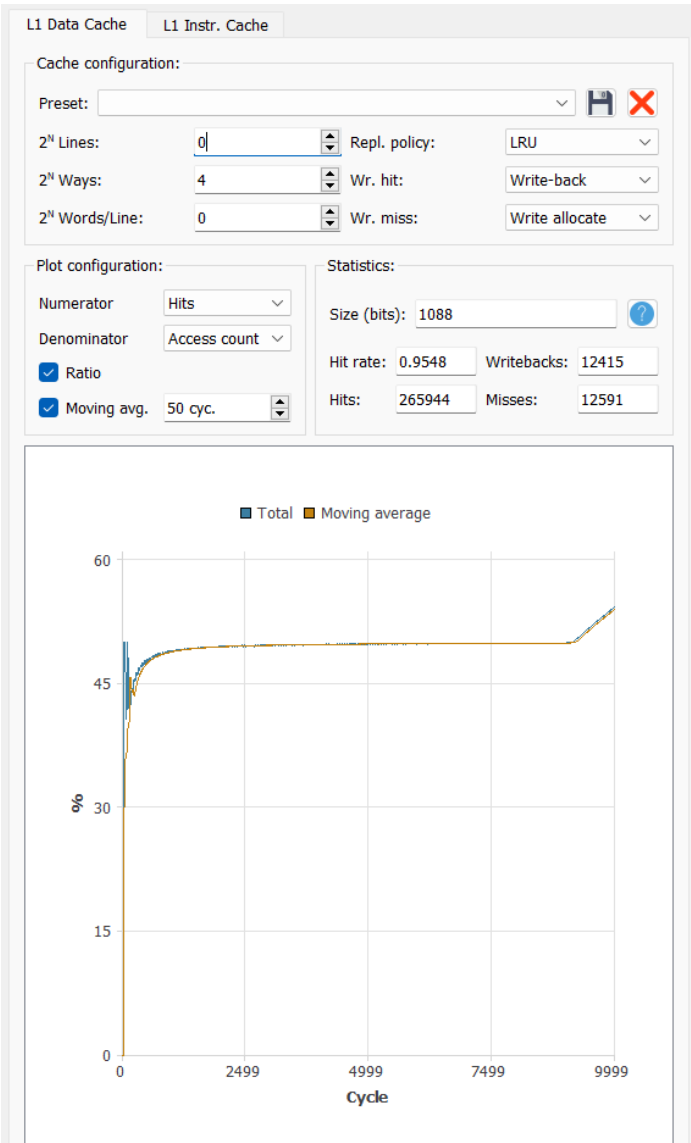
- Asociativa con 4 conjuntos (lines) y 4 vías (ways)



Access address 31 4 3 2 1 0

Index	V	D	LRU	Tag	Word 0
0	1	1	3	0x0f000063	0x00ff0000
	1	0	0	0x07ffffd7	0x0000000a
	1	0	1	0x0f0000db	0x00000000
	1	0	2	0x000011eb	0xf000db0
1	1	0	1	0x07ffffd7	0x00000003
	1	0	3	0x0f0000db	0x00000000
	1	0	2	0x000011ea	0xf000dbc
	1	0	0	0x07fffffc	0x00000001
2	1	1	0	0x07fffffc	0x00000002
	1	0	2	0x000011ea	0xf000db8
	1	0	1	0x0f0000db	0x00000000
	1	0	3	0x000011eb	0xf0000000
3	1	1	1	0x07ffffd6	0x0000000c
	1	0	3	0x000011ea	0xf000db4
	1	1	0	0x07ffffd5	0x0000252a
	1	0	2	0x0f0000db	0x00000000

- Totalmente asociativa con 16 vías



Access address 31 21 0

Index	V	D	LRU	Tag	Word 0
0	1	1	2	0x1fffff5b	0x0000000c
	1	1	1	0x1fffff72	0x00000002
	1	0	10	0x3c00036d	0x00000000
	1	0	4	0x1fffff5c	0x0000000a
	1	0	3	0x1fffff71	0x00000001
	1	0	13	0x000047ac	0xf000db0
	1	1	0	0x1fffff57	0x000019e5
	1	0	5	0x1fffff5d	0x00000003
	1	0	12	0x3c00036c	0x00000000
	1	0	8	0x3c00036e	0x00000000
	1	0	6	0x3c00036f	0x00000000
	1	0	15	0x000047ae	0xf0000000
	1	0	9	0x000047aa	0xf000db8
	1	0	7	0x000047a9	0xf000dbc
	1	1	14	0x3c00018c	0x00ff0000
	1	0	11	0x000047ab	0xf000db4

A continuación mostraremos una tabla con los datos organizados de mejor manera.

Configuración	Hit rate (%)	Miss rate (%)	Hits	Misses	Writebacks	Tamaño (Bits)
Mapeo directo, 4 líneas	94.2%	5.8%	258694	15922	12421	248
Asociativa 2 sets, 2 vías	95.38%	4.62%	265591	12878	12422	256
Totalmente asociativa, 4 vías	95.36%	4.64%	264268	12856	12422	264
Mapeo directo, 16 líneas	95.25%	4.75%	268552	13401	12413	960
Asociativa 4 sets, 4 vías	95.57%	4.43%	271705	12600	12415	1024
Totalmente asociativa, 16 vías	95.48%	4.52%	265944	12591	12415	1088

¿Cuál de las siguientes configuraciones tiene el mejor hit rate?

La configuración con el mejor hit rate es **Asociativa 4 sets, 4 vías**, con un hit rate del **95.57%**.

Conclusiones de cada integrante del equipo.

Luis Pablo De los Reyes Rentería:

Con la elaboración de este proyecto uno de los aspectos más importantes fue entender cómo las configuraciones de memoria caché impactan en el rendimiento. Durante las pruebas de las distintas configuraciones de caché (direct-mapped, asociativa y totalmente asociativa), pudimos ver cómo la velocidad de acceso a los datos y la eficiencia del procesamiento afectaban directamente la experiencia de juego. Además, durante la implementación del juego, se presentaron varios problemas, como la dificultad para optimizar ciertas funciones, la gestión de la memoria en el juego, y la adaptación de la visualización de los gráficos a los límites de la matriz de LEDs. También fue un reto gestionar la interacción entre el código de control del juego y las entradas de los botones de dirección, ya que cualquier error en el manejo de estos eventos podría hacer que el juego no respondiera correctamente.

Otro problema importante que tuvimos fue con la implementación de la función de colisión de la serpiente. Debido a la naturaleza del juego y las restricciones de hardware, asegurar que la serpiente se moviera correctamente, detectando las colisiones con las paredes o consigo misma, fue más complicado de lo esperado, pero logramos corregirlo.

Juan Pablo Figueroa Martín:

A lo largo del desarrollo de este proyecto, tuvimos que resolver varios problemas técnicos. Por ejemplo, la gestión del movimiento de la serpiente y la correcta detección de las colisiones resultaron ser más complejas de lo que inicialmente pensábamos. Las restricciones del simulador y la necesidad de que el juego funcionara en una matriz de LEDs pequeña nos obligaron a ser más cuidadosos con la programación y a realizar optimizaciones. Además, el manejo de la velocidad del juego y la respuesta de los controles fueron aspectos clave que requirieron ajustes para que funcionara de mejor manera.

Un aspecto crucial fue el análisis de las distintas configuraciones de memoria caché: direct-mapped, asociativa y totalmente asociativa. Al implementar estas variantes y realizar pruebas con el simulador RIPES, fuimos capaces de observar cómo cada una impactaba en la velocidad del juego, lo que nos ayudó a entender la importancia de optimizar el uso de la memoria en sistemas reales. Este tipo de experimentos no solo aumentan la eficiencia de un programa, sino que también son esenciales para comprender las limitaciones de los sistemas embebidos.

Link de GitHub:

https://github.com/750901-750515/Practica_2_Snake-