**Jupiter**

**Índice:**

1

**1 Simulador** 2

1.1 Introducción 2

1.2 Interfaz 2

1.3 Memorias Caché 2

1.4 Pipeline 3

1.5 Entrada Salida 3

1.6 Trabajo con el simulador 3

2

**2 Instalación y Ejecución** 4

3

**3 Aspectos Destacables y Limitaciones** 4

4

**4** **Referencias** 4

**1 Simulador:**

**1.1 Introducción:**

Este documento se dedicará a la exploración del simulador del estándar RISC-V [1] Jupiter [2], simulador de entorno gráfico a modo de aplicación.

Este es un simulador relativamente simple dentro de su entorno, muy especializado en la ejecución de programas de ensamblador puro.

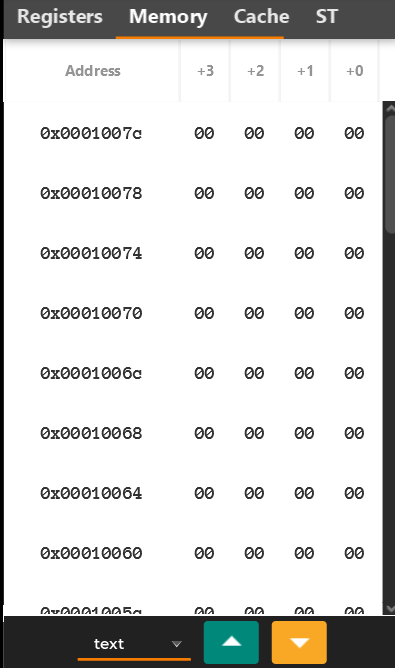
**1.2 Interfaz:**

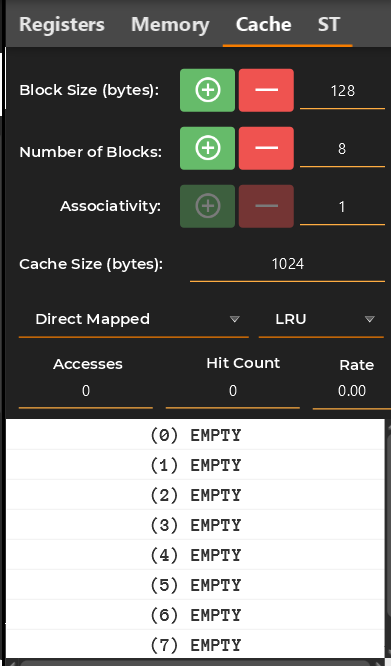
Jupiter es un simulador muy sencillo pensado para trabajar solo con un fichero ensamblador RISC-V, donde las funciones y los apartados de configuración son relativamente sencillos en su implementación. Sin contar que apenas tiene implementadas las operaciones básicas para números enteros con multiplicación y división.

Vemos entonces una interfaz sobria con lo mínimo e imprescindible para trabajar con el, esto puede constituir tanto una ventaja como una desventaja: aunque no aturulla al programador con un exceso de información, ciertas características si se echan de menos cuando se está ejecutando y depurando código.

**1.3 Memorias Caché:**

Este simulador cuenta con un pequeño apartado para la visualización del contenido en memoria, y con una implementación sencilla de las memoras caché por defecto, con algunas opciones de configuración. Sin embargo, esta puede dejar bastante que desear siendo que al igual que simuladores como RARS, no muestra el contenido sino solo si se ha producido un fallo en caché, y al aumentar el tamaño se vuelve mucho más difícil de manejar. Por otro lado, la visualización de la memoria principal no ofrece nada destacable por sobre otros simuladores solo se tienen un pequeño apartado para el trabajo con la misma.

Figura 1.1 (Visualización del contenido en memoria en Jupiter)

Figura 1.2 (Apartado de memoria caché de Jupiter)

**1.4 Pipeline:**

Este simulador no cuenta con una implementación de la ejecución segmentada, sino que solo ejecuta instrucción a instrucción.

**1.5 Entrada Salida:**

Este simulador, al igual que muchos de su entorno, no cuenta con una implementación de excepciones ni interrupciones, en este caso no teniendo ni siquiera implementados periféricos con los que trabajar la entrada salida, incluso si esta es programada.

**1.6 Trabajo con el simulador:**

Trabajar con este simulador se puede hacer algo incómodo en comparación a otros simuladores: cuando se carga el programa el código no es visible, por otra parte, solo se pueden visualizar o los registros o la memoria. Sin embargo, la sencillez que aporta el simulador en cuenta al procesamiento del programa hace que sea muy fácil acostumbrarse a el por lo que tiene ese punto a favor.

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.Figura 1.3

**2 Instalación y Ejecución:**

En el proceso de instalación y configuración inicial de este simulador hay que referirse al apartado installation del repositorio oficial [2] donde tendremos que instalar el archivo comprimido pertinente al sistema operativo que tengamos disponible.

**Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.**Figura 2.1

Una vez extraídos todos los archivos del fichero comprimido, para iniciar el simulador solo tendremos que llamar al ejecutable con su mismo nombre (jupiter) que se encuentra en la carpeta “/image/bin”. Para poder ejecutar el programa el único prerrequisito con el que contamos es tener instalada una versión del JDK de java 8 o superior, de manera contraria la ejecución fallará.

**3 Aspecto Destacable y Limitaciones**

Adicionalmente, un aspecto positivo con el que cuenta este simulador es la simplicidad del mismo en su implementación, es decir, que está escrito en un lenguaje amigable como es java y de una forma fácil de entender y con la que es fácil añadir funcionalidades.

Este simulador no ha presentado mayor problemática ni en su instalación, ni en su configuración ni en su ejecución. Quizá pudiendo atribuírselo a la sencillez del mismo simulador, siendo quizá esta la mayor de sus desventajas. Mientras que otros simuladores tienen implementaciones vastas y detalladas, este cuenta con lo mínimo e imprescindible para trabajar con el ensamblador, dando lugar a un entorno sumamente específico.

# Referencias

|  |  |
| --- | --- |
| [1] | RISC-V, «Ratified RISC-V Specifications,» [En línea]. Available: https://lf-riscv.atlassian.net/wiki/spaces/HOME/pages/16154769/RISC-V+Technical+Specifications. |
| [2] | andrescv, «Repositorio Oficial Jupiter,» [En línea]. Available: https://github.com/andrescv/jupiter. |
| [3] | «Documentación Oficial Jupiter,» [En línea]. Available: https://jupitersim.gitbook.io/jupiter/es. |