**RISC-V Venus Simulator**

**Índice:**

1

**1 Simulador** 2

1.1 Introducción 2

1.2 Interfaz 2

1.3 Memorias Caché 2

1.4 Pipeline 2

1.5 Entrada Salida 2

1.6 Trabajo con el simulador 2

2

**2 Instalación y Ejecución** 2

3

**3 Aspectos Destacables y Limitaciones** 2

4

**4** **Referencias** 2

**1 Simulador:**

* 1. **Introducción:**

Este documento se dedicará a la exploración del simulador del estándar RISC-V: Venus Simulator. Este simulador es un tanto especial puesto que se monta sobre uno de los entornos de programación más usados del sector, Visual Studio Code, aplicación de Microsoft disponible tanto para sistemas operativos Windows como para sistemas operativos Linux, como una extensión de libre desarrollo.

**1.2 Interfaz:**

La interfaz del simulador es la misma que la del entorno, siendo que la única diferencia con respecto al editor de texto, es cuando se abre la interfaz para el trabajo de depuración, donde se abren algunos apartados adicionales: para visualización de los registros, memoria, campos, etc.

**1.3 Memorias Caché:**

Este simulador cuenta con un pequeño apartado para la visualización del contenido de la memoria, sin embargo, no cuenta con un apartado de memorias caché.

**1.4 Pipeline:**

Este simulador cuenta con una única implementación que ejecuta en un único nivel, es decir, no cuenta con opciones para ejecución segmentada.

**1.5 Entrada Salida:**

Este simulador cuenta con algunos periféricos implementados con los que se puede trabajar con la entrada salida programada. Sin embargo, no cuenta con una implementación de interrupciones ni excepciones.

**1.6 Trabajo con el simulador:**

Para trabajar con el simulador, …

**2 Instalación y Ejecución:**

Este simulador se instala como extensión de la aplicación de Microsoft: Visual Studio Code. Es por esto que el proceso de instalación es prácticamente inmediato siempre que se cuente con la aplicación ya instalada. Aplicación que está disponible independientemente del sistema operativo.

Para ejecutar el simulador tenemos que trabajar con un fichero ensamblador de extensión “.s” donde tenemos que referirnos al botón de ejecución. Una vez hecho esto podremos ver el inicio de la interfaz y tendremos disponibles los registros y variables del mismo.

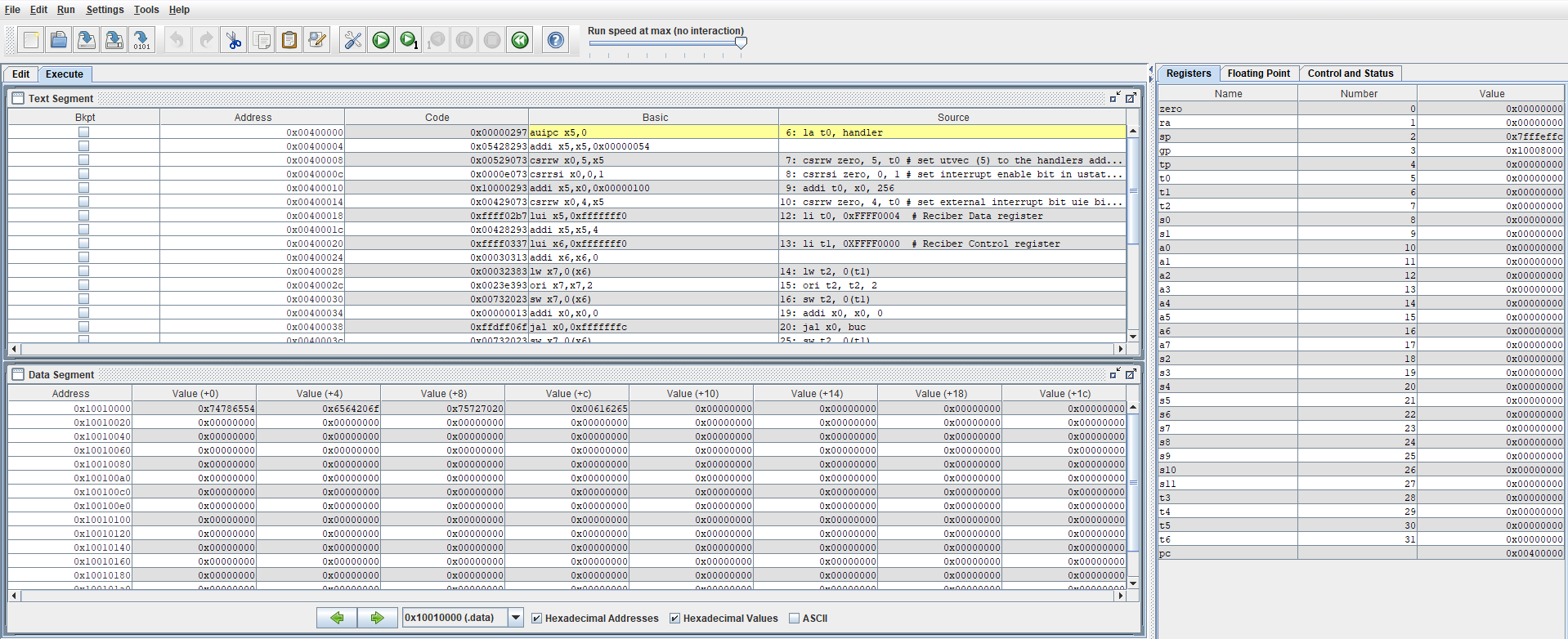
**3 Aspectos Destacables y Limitaciones**

Como aspecto a destacar, tenemos que este simulador se desarrolla sobre uno de los entornos más usados dentro del mundo de la programación, siendo que la familiaridad con el entorno puede ayudar al programador en sí mismo.

Por otro lado, este sistema tiene algunas limitaciones, entre ellas el formato y visualización de los registros, que se ven como variables, teniendo muy poco espacio de interfaz dedicado. También cabe mencionar que no cuenta con ninguna implementación interesante como puede ser la entrada salida por interrupciones, un apartado para memorias caché, ejecución segmentada, etc. Esto, junto a la escasa configuración posible del simulador, lo que lo vuelve altamente específico para programar con un único fichero ensamblador.

# 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] | «vscode-riscv-venus,» [En línea]. Available: https://github.com/hm-riscv/vscode-riscv-venus. |
| [3] | «RISC-V Venus Simulator,» [En línea]. Available: https://marketplace.visualstudio.com/items?itemName=hm.riscv-venus. |

Ejemplo ImagenFigura 1.6