

**Instituto Tecnológico de Costa Rica**

**Área Académica de Ingeniería en Computadores**

(Computer Engineering Academic Area)

**Programa de Licenciatura en Ingeniería en Computadores**

(Licentiate Degree Program in Computer Engineering)

**Curso: CE5302- Proyecto de Diseño de Ingeniería en Computadores**

(Course: CE5302 Computer Engineering Design Project)



**Documento de Requerimientos: Herramienta de perfilado para una arquitectura basada en RISC V**

(Requirements Document: Profiling Tool for and architecture based on RISC-V)

**Realizado por:**

(Made by:)

**Arturo Salas Delgado, 201229804**

**Profesor:**

(Professor)

**Gustavo Cubas Euceda**

**Fecha: Cartago, 23 de Febrero, 2018**

(Date: Cartago, February 23, 2018)

## **Indice**

<b>Introducción</b>	<b>2</b>
Propósito	2
Alcance	2
Resumen del Producto	2
Perspectiva del Producto	2
Funciones del Producto	2
Características del Usuario	3
Limitaciones	3
Definiciones	3
<b>Referencias</b>	<b>4</b>
<b>Requerimientos Específicos</b>	<b>4</b>
Interfaces Externas	4
Funciones	4
Requerimientos de Usabilidad	5
Requerimientos de Rendimiento	6
Requerimientos de la Base de Datos Lógica	6
Limitaciones de Diseño	6
Atributos del Sistema de Software	6
Información de Soporte	6
<b>Verificación</b>	<b>6</b>
Interfaces Externas	6
Funciones	6
Requisitos de Usabilidad	7
Requisitos de Rendimiento	8
Requisito de Bases de Datos Lógica	8
Limitaciones de Diseño	8
Atributos del Sistema de Software	8
Información de Soporte	8
<b>Apéndices</b>	<b>8</b>
Supuestos y Dependencias	8
Siglas y Abreviaturas	9
Matriz de Trazabilidad	9

## **1. Introducción**

### **1.1. Propósito**

El propósito de este documento es mostrar todos los requerimientos que se necesitan para poder desarrollar el proyecto llamado “Herramienta de perfilado para una arquitectura basada en RISC V”.

La escuela de Ingeniería Electrónica, junto con el área de Ingeniería en Computadores e Ingeniería en Mecatrónica trabajan sobre la creación de distintos bloques de hardware, de manera personalizada, para la realización de tareas complejas. Es acá donde se pretende crear esta herramienta de manera libre, bajo términos de OpenSource, para que se pueda utilizar en la recopilación de estadísticas de un procesador pipeline basado en el set de instrucciones RISC V versión RV32I.

### **1.2. Alcance**

El proyecto es limitado para la escuela de Ingeniería Electrónica y las áreas de Ingeniería en Computadores e Ingeniería en Mecatrónica, específicamente para ser utilizado en laboratorios o para futuras generaciones que requieran adiciones o modificaciones en la herramienta.

### **1.3. Resumen del Producto**

El producto consta de tres etapas: se encarga de la generación de código binario a partir de un programa escrito en C estándar, la optimización de ese código binario y la generación de estadísticas sobre un procesador basado en la arquitectura RISC V ejecutando ese programa.

#### **1.3.1. Perspectiva del Producto**

El producto debe tener la capacidad de soportar distintos escenarios, donde se tendrá que ejecutar diferentes programas escritos en C estándar con funciones aritméticas simples, usando o no instrucciones de punto flotante, crear su similar en código binario y verificar su correcto funcionamiento en el procesador basado en RISC V, que estará corriendo sobre una placa de desarrollo de hardware, en este caso, Zedboard o Nexys 3.

#### **1.3.2. Funciones del Producto**

- **Generación de código binario basado en RISC V.** El usuario tendrá un código escrito en C estándar, con instrucciones aritméticas básicas como sumas, restas, multiplicaciones y divisiones. De ahí, la herramienta se

encarga de la generación de código binario basado en el set de instrucciones de RISC V versión RV32I, a partir del código en C escrito por el usuario.

- **Optimización de código binario.** La utilización de instrucciones en punto flotante para la arquitectura RISC V puede traer consigo inconvenientes de compilación, ya que no se realiza de manera optimizada. Es por esto que mediante la herramienta de perfilado se pretende aplicar una estrategia de soporte para este tipo de instrucciones.
- **Generación de estadísticas.** Una vez que se tiene el código binario generado, se pretende que este sea ejecutado por un procesador predefinido basado en la arquitectura RISC V. Es aquí donde la herramienta se encarga de la colección de datos, como tiempos de ejecución, para ser mostrados al usuario y poder obtener conclusiones cuantitativas sobre el proceso.

### 1.3.3. Características del Usuario

La herramienta de perfilado va orientada a personas que tengan conocimiento intermedio o avanzado del desarrollo de software en el lenguaje C, así como personas que entiendan y conozcan el funcionamiento de un procesador y puedan utilizar placas de desarrollo de hardware como Zedboards y Nexys 3.

### 1.3.4. Limitaciones

Una de las limitaciones es que la herramienta puede ser probada en un procesador que va completamente aparte de este proyecto, donde se debe de conocer su estructura y funcionamiento para poder realizar pruebas.

Además, el sistema debe de basarse en software OpenSource y hardware OpenCore, esto porque la herramienta podrá ser utilizada por la comunidad estudiantil o investigadores del Tecnológico de Costa Rica.

## 1.4. Definiciones

- **Interfaz Externa.** Es la interacción del proyecto con personas, sistemas de software o hardware u otros sistemas relevantes para su correcto funcionamiento.
- **Rendimiento.** Es una indicación de disponibilidad, velocidad, tiempos de recuperación o tiempos en que se dura en ejecutar determinadas funciones del sistema.
- **Optimización.** Es la acción requerida para buscar la mejor manera de realizar una tarea.

- **Disponibilidad.** Es la presencia funcional que hace posible dar respuestas, resolver problemas o proporcionar ayuda limitada referente a un sistema.
- **Requerimiento.** Es una condición o capacidad necesitada por un usuario para resolver problemas o alcanzar objetivos.
- **Código Abierto.** Es el software que es distribuido y desarrollado de manera libre, orientado a los beneficios de compartir el código.

## 2. Referencias

- Documento de requerimientos basado en las normas establecidas por el estándar System and Software Engineering ISO/IEC/IEEE 29148, 2011

## 3. Requerimientos Específicos

En esta sección se muestran los requerimientos asociados a la herramienta de perfilado que garantizan su correcto funcionamiento

### 3.1. Interfaces Externas

El sistema cuenta con una interfaz definida por un procesador desarrollado aparte al proyecto.

Tal como se explicó anteriormente, se deben de generar estadísticas de tiempos de ejecución a partir de un procesador predefinido, es aquí donde se intercala el proyecto con el funcionamiento del procesador, a través de un archivo de configuración que define la arquitectura y las características del procesador, es decir, las estadísticas a generar se basan en el archivo de configuración.

### 3.2. Funciones

A continuación, se muestra un listado de los requerimientos solicitados que garantizan el correcto funcionamiento de la herramienta de perfilado.

Código	Requerimiento
RQ-01	El sistema debe tener la capacidad de introducir un programa escrito en C estándar para su correcta ejecución en un procesador basado en la arquitectura RISC V.

RQ-02	El sistema debe generar código binario basado en la arquitectura RISC V versión RV32I a partir de un programa escrito en C estándar para ser ejecutado en un procesador basado en la arquitectura RISC V.
RQ-03	El sistema debe proporcionar un archivo de salida con el código binario generado para ser transferido a la memoria de instrucciones del procesador basado en RISC V.
RQ-04	El sistema debe optimizar el uso de instrucciones en punto flotante mediante una estrategia nueva a implementar para evitar una extensión de código ensamblado innecesaria y garantizar un mejor rendimiento de procesamiento.
RQ-05	El sistema debe generar estadísticas de tiempos de ejecución sobre el código binario ejecutado en el procesador basado en RISC V para obtener conclusiones cuantitativas de su rendimiento.
RQ-06	El sistema debe proporcionar un archivo de salida con las estadísticas recopiladas de la ejecución del programa en el procesador basado en RISC V para ser visualizado por el usuario.
RQ-07	El sistema debe de contar con un manual de usuario para garantizar el correcto uso de la herramienta.
RQ-08	El sistema debe de ser almacenado en un repositorio público en GitHub para garantizar el acceso a personas que deseen la herramienta para uso académico o de investigación.

**Tabla 1. Listado de requerimientos**

### **3.3. Requerimientos de Usabilidad**

Se debe brindar al usuario una interfaz por terminal intuitiva para que se facilite el uso de la herramienta, desde la integración del archivo de programación en C hasta las pruebas en una tarjeta de desarrollo, ya sea ZedBoard o Nexys 3. Sin embargo, cabe resaltar que dicha herramienta está dirigida para usuarios con conocimiento medio o avanzado en el área de programación y desarrollo de hardware.

### **3.4. Requerimientos de Rendimiento**

El rendimiento del sistema está enlazado directamente con el computador donde se ejecute, por lo cuál no es una parte de preocupación sobre la herramienta. Sin embargo, el sistema debe ser estable con una disponibilidad del 99.99%.

### **3.5. Requerimientos de la Base de Datos Lógica**

No aplica para el desarrollo del sistema.

### **3.6. Limitaciones de Diseño**

El sistema se limita a módulos de software OpenSource. Además, se limita a módulos de hardware OpenCore.

### **3.7. Atributos del Sistema de Software**

El software basa su funcionamiento en la recopilación de estadísticas a través de la simulación de ejecución de código binario basado en RISC V versión RV32I en un procesador basado sobre esa misma arquitectura.

### **3.8. Información de Soporte**

El sistema cuenta con un manual de usuario y un documento de diseño donde se explica detalladamente el uso del sistema. Por lo tanto, el sistema no posee información de soporte y de mantenimiento.

## **4. Verificación**

En esta parte se especifican los mecanismos de verificación de los requerimientos listados en la sección anterior.

### **4.1. Interfaces Externas**

Para verificar la funcionalidad de la interfaz entre el proyecto y el procesador para la generación de estadísticas, se pretende aplicar una serie de pruebas, donde se utilizan diferentes generados del código binario para ejecutarlo en la FPGA que contiene la descripción de hardware del procesador basado en RISC V. De aquí se pueden tomar tiempos de ejecución dependiendo del código ingresado en el procesador, verificando su correcta funcionalidad.

### **4.2. Funciones**

En la tabla siguiente se muestra la lista de requerimientos y su forma en que serán verificados, mediante un conjunto de pruebas.

<b>Código</b>	<b>Verificación</b>
RQ-01	Prueba donde se verifica si el usuario ha ingresado un programa válido para el sistema.
RQ-02	Se verifica la generación de código mediante la herramienta de desensamblado “riscv32-unknown-elf-objdump” para obtener su similar en código ensamblador, viendo de manera explícita el uso de instrucciones RISC V. Además, se verifica el código binario con la hoja de datos de la arquitectura RISC V.
RQ-03	Se verifica que el sistema ha generado un archivo con el código binario correspondiente al programa en C ingresado por el usuario, luego será probado en la FPGA con el procesador correspondiente.
RQ-04	Se verifica su funcionamiento realizando una comparación entre el código binario generado sin optimizar y optimizado. Una vez generados los binarios, se ejecutan en el procesador incorporado en la FPGA y se verifica el rendimiento, obteniendo conclusiones cuantitativas de la optimización.
RQ-05	Se compara los resultados teóricos con los resultados prácticos, de ahí se verifica el correcto funcionamiento.
RQ-06	Se realiza una prueba para verificar si el archivo de salida ha sido generado o no.
RQ-07	Verificación por parte del supervisor, quién dará el visto bueno.
RQ-08	Verificación por parte del supervisor, quién dará el visto bueno.

**Tabla 2. Verificación de requerimientos**

#### **4.3. Requisitos de Usabilidad**

Se realizarán pruebas de usabilidad con distintas personas pertenecientes al Tecnológico de Costa Rica, específicamente de las carreras de Ingeniería Electrónica, Ingeniería en Mecatrónica e Ingeniería en Computadores.



#### **4.4. Requisitos de Rendimiento**

Para su verificación, el sistema será probado en múltiples computadores que estén trabajando sobre un sistema operativo Linux. Además, se ejecuta el sistema en reiteradas ocasiones para verificar su disponibilidad de 99.99%. También se debe de verificar que se generen los archivos esperados, sin dependencia de las características del computador.

#### **4.5. Requisito de Bases de Datos Lógica**

No aplica para el desarrollo del sistema.

#### **4.6. Limitaciones de Diseño**

Para su verificación, se debe de generar una licencia OpenSource para la herramienta de perfilado.

#### **4.7. Atributos del Sistema de Software**

Para su verificación se realizan numerosas pruebas para la recopilación de estadísticas, donde se comparan los resultados teóricos con los resultados prácticos. De esta manera se pone a prueba los atributos de software y el funcionamiento de la herramienta de perfilado.

#### **4.8. Información de Soporte**

Ya que el sistema no cuenta con soporte y mantenimiento, no se realiza verificación de esta parte.

### **5. Apéndices**

#### **5.1. Supuestos y Dependencias**

Se hace el supuesto de que el compilador de RISC V versión RV32I aportado por la comunidad OpenSource funciona de forma correcta, sin necesidad de cambios o dependencias extras de las conocidas.

Además, el desarrollo del sistema depende de una archivo de configuración predefinido con las características del procesador basado en RISC V existente.

Los requerimientos son susceptibles a cambios durante el desarrollo del sistema semana a semana, donde se tiene el consentimiento total de Carlos Adrián Salazar García para sus validaciones, quién está a cargo del proyecto y es supervisor del mismo.

## 5.2. Siglas y Abreviaturas

A continuación, se muestran las siglas y abreviaturas utilizadas en este documento:

- **RISC V.** Consiste en un set de instrucciones y procesadores abiertos a la comunidad para su libre uso. Es un proyecto OpenSource.
- **FPGA.** Field Programmable Gate Array, por sus siglas en inglés, es un dispositivo donde se puede cargar descripciones de hardware mediante diferentes protocolos de transmisión de datos.
- **RV32I.** Versión 2.2 del set de instrucciones RISC V.

## 5.3. Matriz de Trazabilidad

No aplica para el desarrollo del sistema.