DOCUMENTO DE ARQUITECTURA Y DISEÑO DETALLADO DE SOFTWARE

Evaluador de microcontroladores para misiones espaciales

Versión A

Escrito por Gonzalo Nahuel Vaca

FIUBA

13 de agosto de 2021

Índice

Registros de cambios

Revisión	Detalles de los cambios realizados	Fecha
A	Creación del documento	11/08/2021

1. Introducción

1.1. Propósito

Este documento representa una especificación de requerimientos de software para un *Evaluador de microcontroladores para misiones espaciales*. El documento está dirigido a las personas que trabajen en la esfera de: análisis, diseño, implementación o pruebas.

1.2. Ámbito del sistema

El nombre del sistema será SISE y permitirá evaluar si el microcontrolador deseado puede tener un uso espacial. Además, facilitará la valoración de las técnicas de mitigación de errores. El proyecto incluirá dos módulos que funcionarán en ámbitos distintos. Ellos serán:

- Inyector por consola de comando (CCI)
- Proceso de dispositivo bajo prueba (DUT)

El ámbito de CCI será el ordenador del usuario; mientras que el proceso de DUT funcionará en el microcontrolador.

1.3. Definiciones, acrónimos y abreviaturas

1. Definiciones:
   - Single event effect: efecto de una partícula enérgicamente cargada sobre un microcontrolador.
   - Single event funtional interrupt: interrupción causada por el impacto de una sola partícula que conduce a una no funcionalidad temporal.
   - Single event upset: pulso transitorio en la lógica o circuitos de apoyo. Son *soft-errors* no destructivos.
   - Soft-error: tipo de error en donde una señal o dato es incorrecto.
2. Acrónimos:
   - API: interfaz de programación de aplicaciones.
   - DUT: dispositivo bajo prueba (microcontrolador).
   - FOM: figura de mérito.
   - IEEE: Instituto de Ingenieros Eléctricos y Electrónicos.
   - OCD: on-chip debugger.
   - SEE: single event effect.
   - SEFI: single event functional interrupt.
   - SEU: single event upset.
   - TBD: a ser determinado.
   - UART: universal asynchronous receiver-trasmitter.

3. Abreviaturas:
    - Std: estándar.

1.4. Referencias

INVAP - Propuesta de tesis: sistema de inyección de soft-errors.

1.5. Visión general del documento

Este documento se realizó según lo especificado en el estándar IEEE Std. 830-1998.

2. Descripción general

2.1. Perspectiva del producto

El proyecto aquí especificado es independiente de otros sistemas y no tiene relación con otros productos. Como se especificó en subsección 1.1, se realizarán los siguientes módulos: CCI y Proceso de DUT.

El módulo CCI tendrá la función de generar SEFIs que introduzcan SEUs. Los SEFI-SEU serán inyectados de forma electrónica; esto simulará los efectos de una partícula cargada que impacta en el DUT. Como se explicó en la subsección 1.2, el módulo residirá en el ordenador del usuario. La interacción se realizará a través de una *consola de línea de comandos*. Finalmente, se podrá configurar el ensayo a realizar.

En la figura 1 se puede observar el diagrama en bloques del módulo CCI. La consola de usuario será la interfaz que el ingeniero utilice para usar el sistema. El controlador de ensayos procesará los datos ingresados por el usuario, coordinará los SEFI-SEU y observará los reportes del DUT. El servidor OCD se encargará de realizar lecturas de registros y las inyecciones de SEFI-SEU. El OCD API será la interfaz entre el Controlador de ensayos y el Servidor OCD; utilizará un protocolo TBD. La Interfaz serie capturará los informes del DUT. Finalmente, la Persistencia de datos almacenará toda la información generada durante la ejecución del ensayo.

En la figura 2 se puede observar el diagrama en bloques del módulo *Proceso de DUT*. El firmware deberá recorrer todos los periféricos del DUT. En cada periférico se generará una operación de autovalidación. Finalizada la verificación de todos los periféricos, el firmware enviará un reporte a través de la UART. Finalmente, el bloque de debugger será el punto de ingreso para las SEFI-SEU.

2.2. Funciones del producto

El software aquí especificado brindará las siguientes funcionalidades:

1. Referentes al CCI:

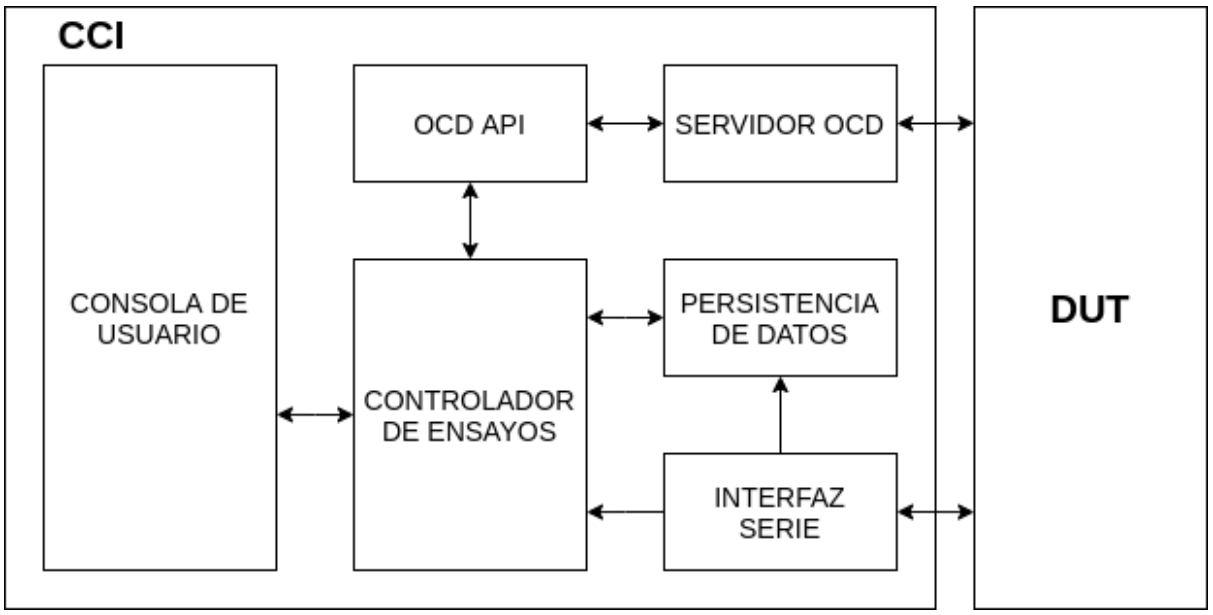

Figura 1. Diagrama en bloques del módulo CCI.

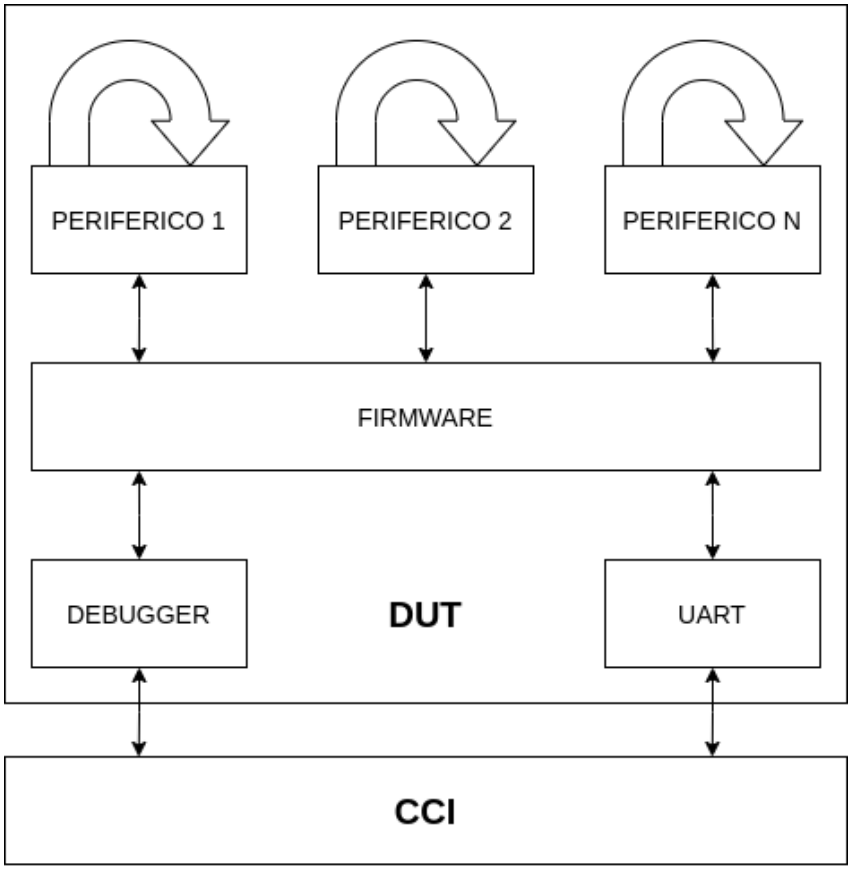

Figura 2. Diagrama en bloques del módulo Proceso de DUT.

1.1. Generará de una interfaz de usuario.
1.2. Permitirá configurar el ensayo a realizar.
1.3. Activará el Proceso de DUT.
1.4. Observará la salida del DUT.

1.5. Inyectará SEFI-SEU en el DUT.
1.6. Persistirá las operaciones, entradas y salidas.
1.7. Generará informes del ensayo realizado.

2. Referentes al Proceso de DUT:

2.1. Verificará el estado de los periféricos del DUT.
2.2. Detectará si el DUT perdió su secuencia.
2.3. Generará reportes de estado de periféricos y secuencia.
2.4. Permitirá que CCI configure el alcance de la secuencia.
2.5. Permitirá que CCI maneje el flujo de su secuencia.

2.3. Características de los usuarios

Los usuarios finales de este producto son ingenieros de desarrollo de INVAP.

2.4. Restricciones

Las restricciones del desarrollo del sistema son las siguientes:

- Utilización de repositorio con control de versiones *Gitlab*.
- Documentación del código con *Doxygen*.
- Utilización exclusiva del lenguaje de programación *Python 3*.

2.5. Suposiciones y dependencias

La suposición principal es que se tendrá acceso irrestricto al DUT seleccionado antes del día 01/01/2022.

3. Arquitectura

3.1. Patrones

Para este sistema se emplearan los siguientes patrones arquitectónicos:

1. Inyector por consola de comando:

1.1. Observar y reaccionar.
1.2. Segmentación de proceso.

2. Proceso en DUT:

2.1. Observar y reaccionar.
2.2. *Hardware Abstraction Layer* (HAL).

3.1.1. Observar y reaccionar

1. Este patrón se utiliza cuando un conjunto de sensores se monitorean y despliegan de manera rutinaria.
2. Uso en Inyector por consola de comando:
   2.1. Monitoreo de reportes del DUT.
   2.2. Captura de eventos en la interfaz de usuario.
   2.3. Persistencia en memoria y generación de informes.
3. Uso en Proceso en DUT:
   3.1. Monitoreo del estado de los periféricos.
   3.2. Captura de inyecciones de SEFI-SEU.
   3.3. Reportes del estado de la secuencia.

3.1.2. Segmentación de proceso

1. Este patrón se usa al transformarse datos de una representación a otra antes de que puedan procesarse.
2. El Inyector por consola de comando deberá transformar las salidas del DUT obtenidas entre secuencias para alimentar el proceso de generación de informes.

3.1.3. Hardware Abstraction Layer

1. Este patrón se utiliza para aprovechar la *base de conocimiento* existente sobre una arquitectura de microcontrolador y cuando se necesita crear código que satisfaga una interfaz de programación definida.
2. El Proceso en DUT correrá sobre un *Sistema Operativo de Tiempo Real* (RTOS) definido por INVAP.

4. Diseño detallado

En esta sección se incluye el diseño detallado de los componentes de software.

4.1. Inyector por consola de comando

1. Consola de usuario:
2. Controlador de ensayos:
3. Persistencia de datos:
4. Interfaz serie:

5. Servidor OCD:
6. OCD API:

4.2. Proceso en DUT

1. Autovalidación de periféricos:
2. UART:
3. Secuencia general: