ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE

Evaluador de microcontroladores para misiones espaciales

Versión E

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	27/06/2021
B	Se agrega encabezado en la plantilla del documento. Modificación de la tabla de registro de cambios. Nuevo formato de enumeración de requisitos. Se amplia la sección 2.1.	03/07/2021
C	Se agregan casos de usos en el anexo	06/07/2021
D	Se conforma con las políticas de confidencialidad de INVAP. Se modifica el título del proyecto. Se revisan todos los requerimientos para que sean más específicos. Se modifican los casos de uso según los nuevos requerimientos.	11/07/2021
E	Correcciones generales para la publicación de la cátedra.	13/08/2021

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:

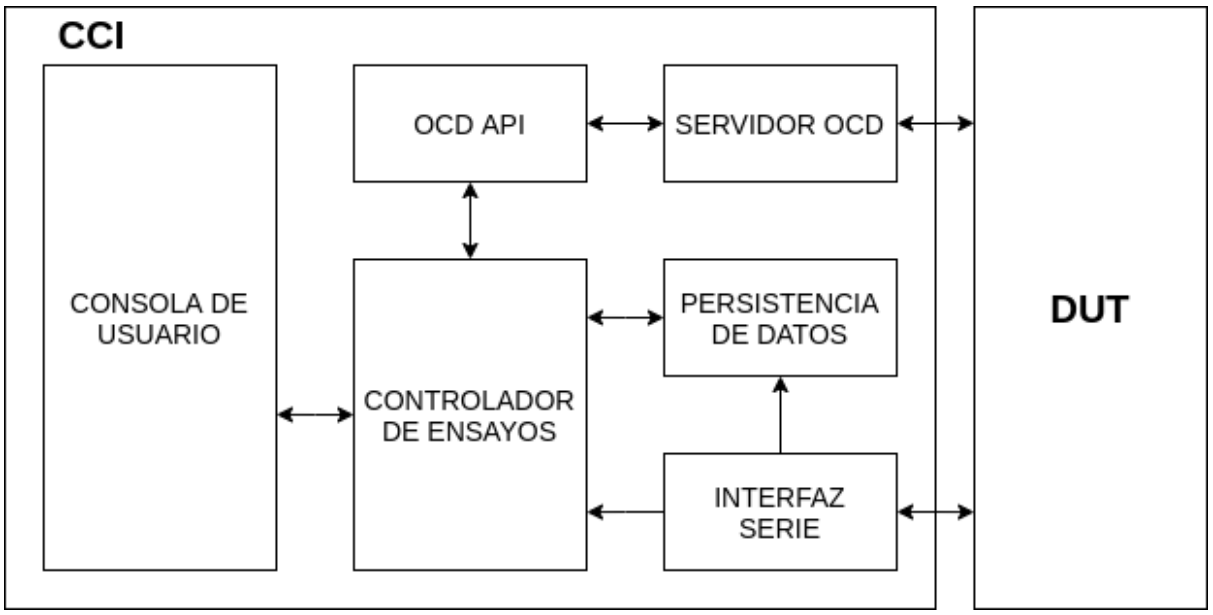

Figura 1. Diagrama en bloques del módulo CCI.

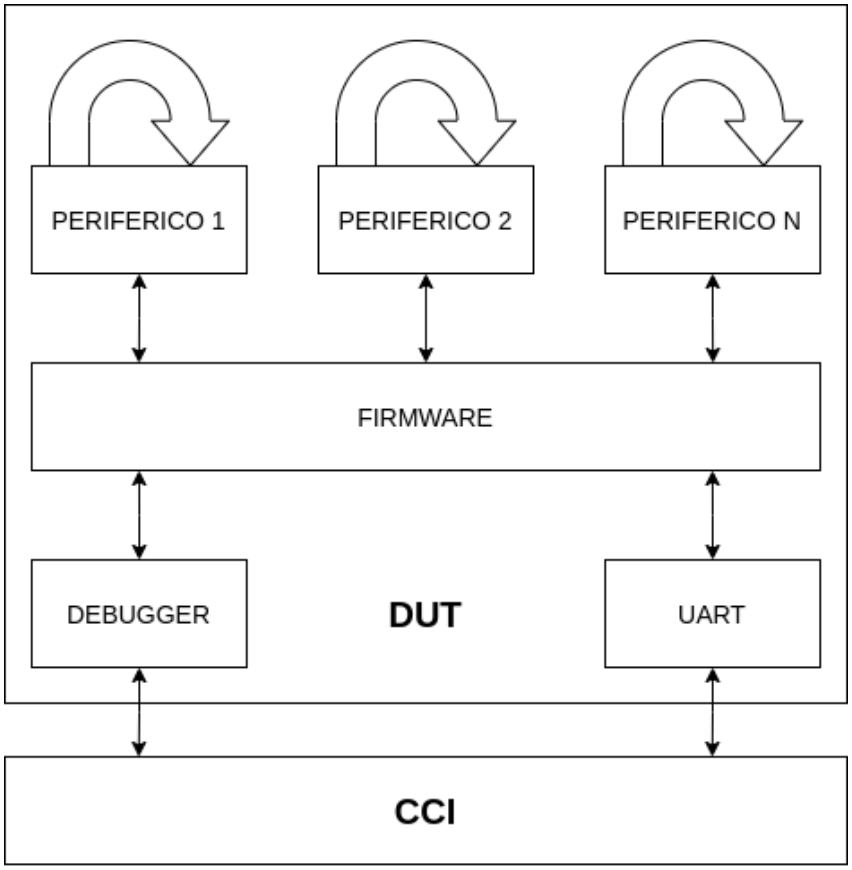

Figura 2. Diagrama en bloques del módulo Proceso de DUT.

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.

2.6. Requisitos futuros

N/A

3. Requisitos específicos

3.1. Interfaces externas

1. CCI:
   1.1. Módulo para el usuario:
      - **[SISE-RS-001]:** deberá representar todos los caracteres de ISO Std. 10646.

- **[SISE-RS-002]:** cumplirá con las secuencias de escape especificadas en ISO Std. 6429.
- **[SISE-RS-003]:** usará el castellano como idioma conforme a la Real Academia Española.
- **[SISE-RS-004]:** se aceptarán barbarismos que conformen la interfaz con los sistemas UNIX.
- **[SISE-RS-005]:** no deberá producir destellos ni cambios bruscos en su intensidad lumínica.
- **[SISE-RS-006]:** no deberá producir sonidos
- Títulos:
  - **[SISE-RS-007]:** los títulos deberán tener una longitud máxima de 30 caracteres.
  - **[SISE-RS-008]:** los títulos deberán estar correctamente capitalizados.
  - **[SISE-RS-009]:** los títulos deberán ser únicos.
- Comandos:
  - **[SISE-RS-010]:** el sistema se iniciará con el comando `sise.py`.
  - **[SISE-RS-011]:** el sistema imprimirá en pantalla un manual de ayuda con el comando `sise --help`.
  - **[SISE-RS-012]:** se podrá exportar la configuración del último ensayo realizado con el comando `sise --export=Ruta`.
  - **[SISE-RS-013]:** se podrá importar la configuración de un ensayo a realizar con el comando `sise --import=Ruta/Archivo`.
- Menú:
  - **[SISE-RS-014]:** el sistema de menú tendrá una arquitectura de árbol.
  - **[SISE-RS-015]:** la navegación entre los nodos del menú será consistente en todo el árbol.
  - **[SISE-RS-016]:** se indicará en todo momento el nodo actual y todos los nodos que lleven a la raíz del árbol.

1.2. Con DUT:

- **[SISE-RS-017]:** la comunicación con UART será en 9600 baudios, 8 bits de datos, 1 bit de parada y 0 bits de paridad.
- **[SISE-RS-018]:** la comunicación con el debugger conformará con la configuración recomendada por el fabricante.

2. Proceso de DUT:

- **[SISE-RS-019]:** la comunicación con el debugger estará disponible durante todo el flujo de la secuencia.
- **[SISE-RS-020]:** durante el flujo de la secuencia, la UART solo podrá transmitir información.
- **[SISE-RS-021]:** en el periodo entre secuencias, la UART podrá recibir y transmitir información.

3.2. Funciones

1. CCI:

- **[SISE-RS-022]:** detendrá la secuencia de duración T del DUT en un momento t definido como $t \epsilon \mathbb{R}^+ \wedge \mathrm{t} < \mathrm{T}$.
- **[SISE-RS-023]:** con la secuencia del DUT detenida, inyectará un SEFI-SEU que invertirá el valor de un bit de un registro interno.
- **[SISE-RS-024]:** La descripción del ensayo definirá el momento t de inyección de SEFI-SEU durante la secuencia de duración T y será un múltiplo de Δt definido como $\Delta t = T/N \, \forall \, N \epsilon \mathbb{N}$.
- **[SISE-RS-025]:** La descripción del ensayo definirá la cantidad M de registros involucrados en la prueba.
- **[SISE-RS-026]:** La cantidad de secuencias L a ejecutar quedará definida como $L = N \times M$.
- **[SISE-RS-027]:** Se ejecutará una secuencia de control sin inyección de SEFI-SEU antes de correr las L secuencias.
- **[SISE-RS-028]:** Por cada ejecución de una secuencia se obtendrá un valor de salida S del DUT.
- **[SISE-RS-030]:** Cada valor de salida S será persistido para su análisis.
- **[SISE-RS-031]:** Cada valor de salida S quedará asociado a su correspondiente secuencia con su inyección de SEFI-SEU y momento t.
- **[SISE-RS-032]:** Se generará un archivo de resultados llamado `resultados-AAAAMMDDHHmm.res`, siendo AAAA el año del ensayo, MM el mes, DD el día, HH la hora y mm los minutos.
- **[SISE-RS-033]:** El archivo de resultados acumulará los SEFI y SEU de cada registro del DUT.
- **[SISE-RS-034]:** El archivo de resultados acumulará los SEU de cada periférico del DUT.
- **[SISE-RS-035]:** El archivo de resultados indicará el FOM del registro definido como: $FOM_{REG} = (1 - \frac{SEU}{SEFI})$
- **[SISE-RS-036]:** El archivo de resultados indicará el FOM del DUT definido como: $FOM_{DUT} = \frac{1}{M} \times \sum_{i=1}^{i=M} FOM_i$ siendo i el número que representa un registro del DUT.
- **[SISE-RS-037]:** Se generará un archivo de histogramas llamado `histogramas-AAAAMMDDHHmm.his` siendo AAAA el año del ensayo, MM el mes, DD el día, HH la hora y mm los minutos.
- **[SISE-RS-038]:** El archivo de histogramas tendrá una tabla que indique la frecuencia de fallos como función de los SEFIs por registro del DUT.
- **[SISE-RS-039]:** El archivo de histogramas tendrá una tabla que indique la frecuencia de fallos como función de los SEFIs por periférico del DUT.

2. Proceso de DUT:

- **[SISE-RS-040]:** deberá correr una secuencia de autoevaluación cuya ejecución durará un tiempo T.
- **[SISE-RS-041]:** deberá producir una salida S que podrá ser un estado o una secuencia de estados.
- **[SISE-RS-042]:** este proceso podrá tener una entrada E.
- **[SISE-RS-043]:** deberá evaluar el estado de los periféricos del DUT.

- **[SISE-RS-044]:** tendrá una función de evaluación para cada uno de los periféricos del DUT.
- **[SISE-RS-045]:** se podrá definir por la entrada E si se desea excluir uno o más periféricos en la secuencia.
- **[SISE-RS-046]:** manejará una interrupción del flujo normal de la secuencia y generará una salida S indicando la excepción, por ejemplo: interrupción por *watchdog*.
- **[SISE-RS-047]:** la salida S utilizará la UART del DUT para ser transmitida.
- **[SISE-RS-048]:** la entrada E utilizará la UART del DUT para ser recibida.

3.3. Requisitos de rendimiento

- **[SISE-RS-049]:** la inyección de SEFI-SEU podrá tener un desvío en su momento t de 10 ms.
- **[SISE-RS-050]:** el desvío tolerado de t deberá representar como máximo el 1 % de la duración T de la secuencia del DUT.
- **[SISE-RS-051]:** aceptará un Δt que como mínimo represente el 5 % de la duración T de la secuencia del DUT.

3.4. Restricciones de diseño

- **[SISE-RS-052]:** se utilizará como dispositivo principal el microcontrolador seleccionado por INVAP.
- **[SISE-RS-053]:** se utilizará un sistema operativo de tiempo real para diseñar el Proceso de DUT.

3.5. Atributos del sistema

1. Mantenibilidad:
   - **[SISE-RS-053]:** el Proceso de DUT se desarrollará con un modelo de capas.

3.6. Otros requisitos

N/A.

4. Apéndices

4.1. Restricciones acerca del lenguaje de programación

El lenguaje de programación será *Python 3* y el código deberá ser documentado según las recomendaciones del manual de usuario de *Doxygen*.

4.2. Casos de uso

Cuadro 1. Caso de uso número 1

Título	Descripción
1. Nombre	Simulación de una misión espacial
1.1 Breve descripción	Se simulan los años de SEFI-SEU en un lapso de 24 horas
1.2 Actor principal	Ingeniero de INVAP
1.3 Disparadores	Comando de ejecución
2. Flujo de eventos	
2.1 Flujo básico	1. El software interpreta la descripción del ensayo 2. El software determina la tasa de fallos del ensayo 3. El software determina la probabilidad de falla de cada registro 4. El software determina la probabilidad de falla en memoria 5. El software realiza inyecciones según los parámetros calculados 6. El software persiste todas las inyecciones realizadas 7. El software entrega un informe final 8. El software retorna un código de tarea finalizada
2.2 Fuljo alternativo	1. El software detecta una anormalidad en el ensayo 2. El software aborta el ensayo 3. El software genera un informe de fallo 4. El software retorna un código de error
3. Pre-condiciones	1. Servidor OCD corriendo 2. Servidor OCD conectado al microcontrolador 3. Servidor OCD con puerto disponible
4. Pos-condiciones	Servidor OCD liberado

Cuadro 2. Caso de uso número 2

Título	Descripción
1. Nombre	Validación de hardware
1.1 Breve descripción	Se obtiene la figura de mérito de un microcontrolador
1.2 Actor principal	Ingeniero de INVAP
1.3 Disparadores	Comando de ejecución
2. Flujo de eventos	
2.1 Flujo básico	1. El usuario define la configuración del ensayo 2. El ensayo se ejecuta en su totalidad 3. El sistema entrega un archivo con la figura de mérito
2.2 Fuljo alternativo	1. Durante el ensayo sucede una excepción irrecuperable 2. El sistema genera un aviso de la situación por pantalla 3. El sistema genera un archivo con toda la información recolectada
3. Pre-condiciones	
4. Pos-condiciones	