



**INSTITUTO POLITÉCNICO NACIONAL**  
**ESCUELA SUPERIOR DE CÓMPUTO**  
**ESCOM**



---

---

**Carrera:**

Ingeniería en sistemas computacionales

**Unidad de Aprendizaje:**

Sistemas en Chip

**Practica 7:**

LEDS MINIOS

**Integrantes:**

Cazares Cruz Jeremy Sajid

Bucio Barrera Oscar Daniel

Perez Ortiz Saul Acosta

Cortes Gerardo

**Fecha de entrega**

11 de noviembre de 2023





# LEDS NIOS

## Introducción

### Características

1. Arquitectura RISC: Como la mayoría de los procesadores modernos, NIOS II utiliza un conjunto de instrucciones reducido para simplificar el diseño y mejorar el rendimiento.
2. Customizable: Una de las principales ventajas de NIOS II es que puede ser altamente personalizado. Los diseñadores pueden agregar o eliminar características como la MMU (Memory Management Unit), cachés, unidades de punto flotante y más, según las necesidades del proyecto.
3. Herramientas de Desarrollo: El software de desarrollo de Altera/Intel, como Quartus y el NIOS II Software Build Tools (SBT), proporciona un entorno de desarrollo integrado (IDE) completo para diseño, síntesis, y depuración.
4. Interconexión Avalon: NIOS II utiliza el estándar de interconexión Avalon, que es un sistema modular y escalable que facilita la conexión entre el procesador y otros componentes periféricos.
5. Instrucciones personalizadas: Una característica distintiva es la capacidad de agregar instrucciones personalizadas al conjunto de instrucciones base, lo que

permite a los diseñadores optimizar operaciones específicas de una aplicación.

## Variantes

A lo largo del tiempo, diferentes variantes del NIOS II han sido lanzadas para adaptarse a diferentes necesidades, como:

NIOS II/e: La "economía" versión, para aplicaciones sensibles al coste.

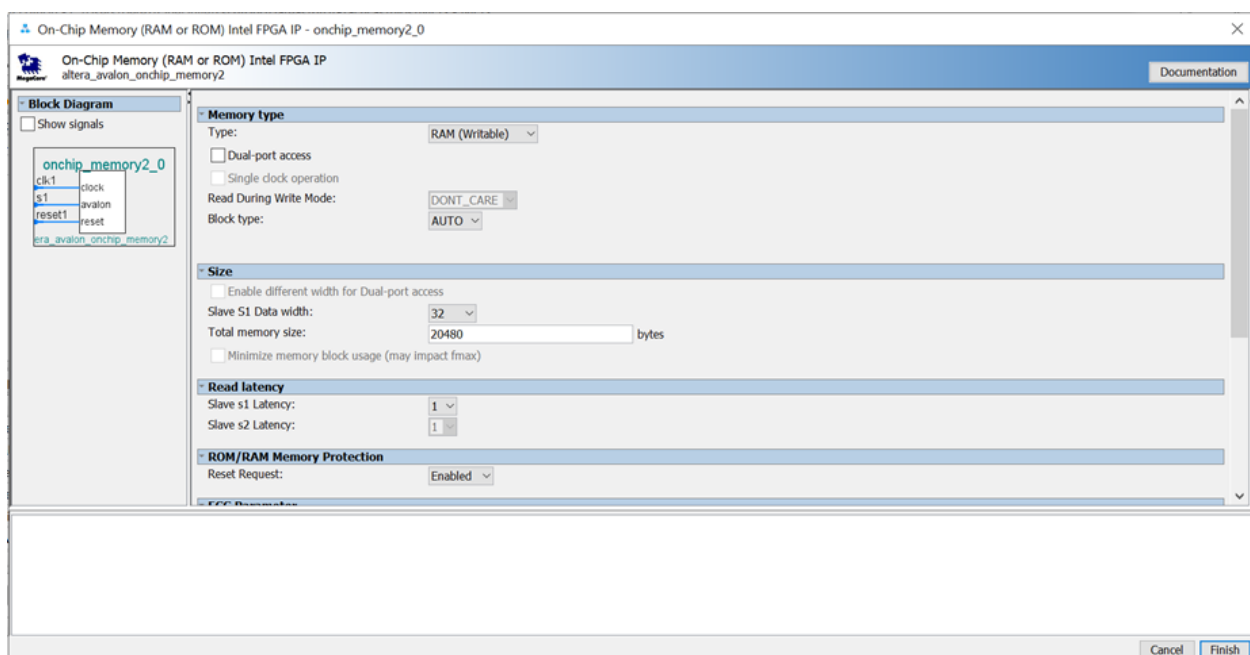
NIOS II/s: Versión "estándar" con un equilibrio entre precio y rendimiento.

NIOS II/f: "Fast", la versión de alto rendimiento.

## Aplicaciones

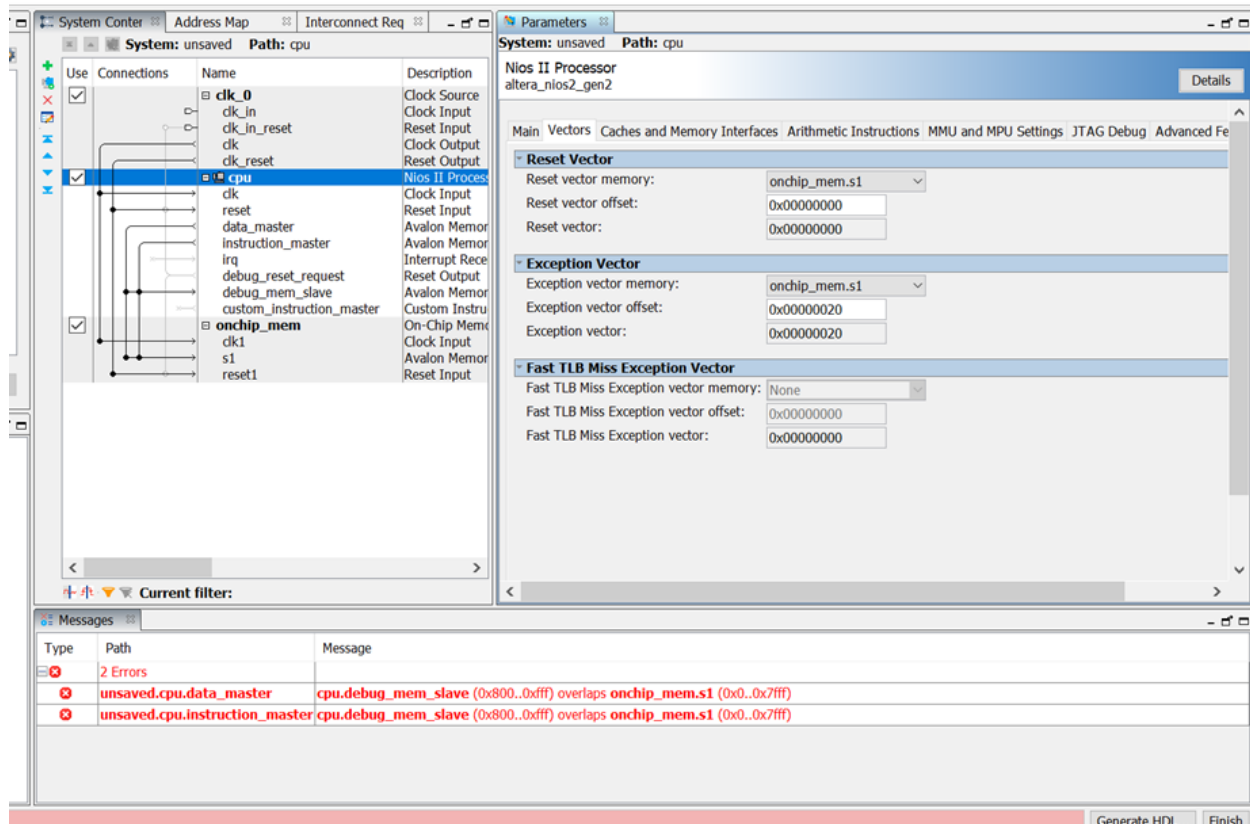
Dado que el NIOS II es un procesador soft-core, ha encontrado aplicaciones en una variedad de áreas, desde controladores en sistemas embebidos, sistemas de comunicación, hasta aplicaciones de procesamiento de señales y más.

# Desarrollo



1. **Tipo de Memoria:** La memoria está configurada como RAM (writable), lo cual indica que es una memoria de acceso aleatorio que permite la escritura y lectura de datos.
2. **Acceso Dual:** Está habilitada la opción de "Dual-port access", lo que significa que la memoria puede ser accedida por dos interfaces diferentes simultáneamente.

3. **Modo de Lectura/Escritura:** Está configurado en "DONT\_CARE" para el modo de lectura y escritura, lo que sugiere que para esta configuración particular, el modo no afecta la implementación de la memoria.
4. **Tamaño de la Memoria:** Se ha definido un tamaño total de memoria de 20480 bytes, lo cual es una especificación importante, ya que determina cuánta información se puede almacenar en esta memoria.
5. **Ancho de Datos para Acceso Dual:** El ancho de datos para el Slave S1 es de 32 bits, lo que indica el tamaño de las palabras de datos que se pueden leer o escribir en un acceso.
6. **Latencia de Lectura:** La latencia de lectura para ambos esclavos (Slave 1 y Slave 2) se establece en 1, lo cual es relevante para el tiempo de respuesta de la memoria después de una solicitud de lectura.
7. **Protección de Memoria:** La protección de memoria ROM/RAM está habilitada, esto puede ser para prevenir escrituras no autorizadas o corrupción de los datos.
8. **Solicitud de Reset:** Hay una opción de "Reset Request" que probablemente permita reiniciar la memoria a un estado inicial o a un estado seguro en caso de ser necesario.



1. **Estructura del Sistema:** La columna de la izquierda muestra una representación gráfica de los componentes del sistema y sus conexiones. Esto incluye un procesador Nios II y varios módulos de interfaz como `clk`, `reset`, `data_master`, `instruction_master`, y otros.
2. **Parámetros del Sistema:** La parte derecha de la ventana detalla los parámetros del sistema, divididos en varias pestañas como "Main", "Vectors", "Caches and Memory Interfaces", entre otros. Aquí puedes detallar cómo se configuran vectores de excepción y de reset, los cuales son cruciales para el manejo de interrupciones y la inicialización correcta del procesador.
3. **Vectores de Excepción:** Se puede ver que la memoria 'onchip\_mem' se utiliza para el vector de excepción y su offset está configurado en `0x00000020`. Esto indica dónde en la memoria se encuentran las rutinas de manejo de excepciones.
4. **Errores del Sistema:** En la parte inferior de la ventana, hay mensajes de error que indican que hay una superposición de direcciones entre `cpu_debug_mem_slave` y `onchip_mem_s1`. Estos errores necesitan ser resueltos para que el sistema funcione.

correctamente, y sería relevante discutir cómo se abordan y corrigen estos problemas en tu reporte.

5. **Configuración de la Interconexión:** La ventana central superior muestra la pestaña "Interconnect Req", lo que sugiere que se está configurando cómo se interconectan los diferentes módulos del sistema. Esto es importante para asegurar que los datos puedan fluir correctamente entre el procesador y la memoria, así como otros periféricos.

## Configuración del Sistema

### Estructura del Sistema

El sistema se ha estructurado alrededor del procesador Nios II, integrando diversos módulos que incluyen controladores de reloj ( `clk` ), reinicio ( `reset` ), y maestros de instrucciones y datos ( `instruction_master` y `data_master` ). La interconexión de estos elementos se representa gráficamente, mostrando la sinergia entre los distintos componentes del sistema.

### Parámetros y Vectores de Configuración

Se establecieron parámetros específicos dentro de la pestaña "Vectors":

- **Reset Vector:** Configurado para apuntar a la memoria en chip `onchip_mem.s1` con un offset de `0x00000000` , definiendo la ubicación inicial del código de arranque.
- **Exception Vector:** También se dirige a `onchip_mem.s1` , pero con un offset de `0x00000020` , lo cual especifica el inicio del manejo de excepciones.

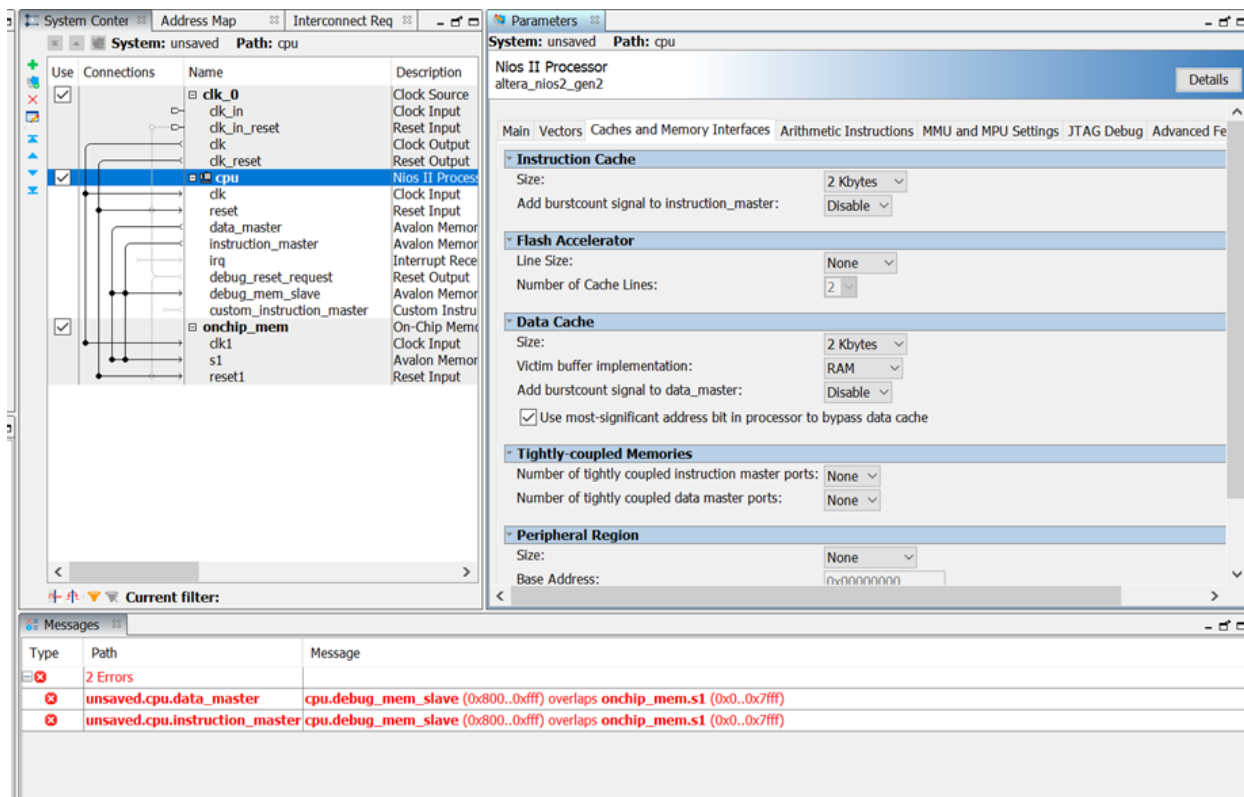
Estos vectores son críticos para el control del flujo del programa durante la operación normal y en situaciones de interrupción.

### Errores y Soluciones

Durante la configuración, se identificaron errores de superposición de direcciones entre `cpu_debug_mem_slave` y `onchip_mem.s1` . La resolución de estos conflictos fue crucial para garantizar un funcionamiento adecuado del sistema. Se ajustaron las direcciones para resolver las superposiciones y se validó la configuración mediante una simulación de integridad de señal.

## Discusión

La correcta configuración de vectores de excepción y reset es fundamental para el manejo eficiente de rutinas críticas, como la inicialización del sistema y la respuesta a interrupciones, que podrían ser utilizadas para modificar patrones de iluminación en la matriz de LEDs o para responder a entradas de usuario.



- **Instruction Cache:** Se observa que está configurada para tener un tamaño de 2 Kbytes, y la señal de burstcount hacia el `instruction_master` está deshabilitada.
- **Flash Accelerator:** No está en uso, y el número de líneas de caché es 2.
- **Data Cache:** Similarmente, se muestra un tamaño de 2 Kbytes para la caché de datos y se está utilizando RAM como la implementación del buffer. La señal de burstcount para el `data_master` también está deshabilitada, y se selecciona la opción para utilizar la dirección más significativa del procesador para pasar por alto la caché de datos.

- **Tightly-coupled Memories:** No se han configurado memorias estrechamente acopladas tanto para la instrucción como para los puertos maestros de datos.
- **Peripheral Region:** No hay ninguna configuración aplicada aquí, y la dirección base es cero.

En la parte inferior de la ventana, hay mensajes de error que indican que hay superposiciones en las direcciones entre el `cpu_debug_mem_slave` y el `onchip_mem.s1`, tal como se ha mencionado previamente. Esto podría significar que hay conflictos en la asignación de direcciones de memoria que necesitarán ser resueltos para el correcto funcionamiento del sistema.

Use	Connections	Name	Description	Export	Clock	Base	End	I...	Tags
<input checked="" type="checkbox"/>		<b>clk_0</b>	Clock Source						
		clk_in	Clock Input	clk	exported				
		clk_in_reset	Reset Input	reset					
		clk	Clock Output		clk_0				
		clk_reset	Reset Output						
<input checked="" type="checkbox"/>		<b>cpu</b>	Nios II Processor						
		clk	Clock Input		clk_0				
		reset	Reset Input		[dk]				
		data_master	Avalon Memory Mapped Master		[dk]				
		instruction_master	Avalon Memory Mapped Master		[dk]				
		irq	Interrupt Receiver		[dk]			IRQ 0	IRQ 31
		debug_reset_request	Reset Output		[dk]				
		debug_mem_slave	Avalon Memory Mapped Slave		[dk]	0x0800	0x0fff		
		custom_instruction_master	Custom Instruction Master						
<input checked="" type="checkbox"/>		<b>onchip_mem</b>	On-Chip Memory (RAM or ROM) Int...						
		clk1	Clock Input		clk_0				
		s1	Avalon Memory Mapped Slave		[dk1]	0x0000	0x4fff		
		reset1	Reset Input		[dk1]				
<input checked="" type="checkbox"/>		<b>jtag_uart</b>	JTAG UART Intel FPGA IP						
		clk	Clock Input		clk_0				
		reset	Reset Input		[dk]				
		avalon_jtag_slave	Avalon Mem...		[dk]	0x0000	0x0007		
		irq	Interrupt Se...		[dk]				
<input checked="" type="checkbox"/>		<b>sys_clk_timer</b>	Interval Tim...						
		clk	Clock Input		clk_0				
		reset	Reset Input		[dk]				
		s1	Avalon Memory Mapped Slave		[dk]	0x0000	0x001f		
		irq	Interrupt Sender		[dk]				
<input checked="" type="checkbox"/>		<b>sysid</b>	System ID Peripheral Intel FPGA IP						
		clk	Clock Input		clk_0				
		reset	Reset Input		[dk]				
		avalon_jtag_slave	Avalon Memory Mapped Slave		[dk]	0x0000	0x0007		
		control_slave							
<input checked="" type="checkbox"/>		<b>leds</b>	PIC (Parallel I/O) Intel FPGA IP						
		clk	Clock Input		clk_0				
		reset	Reset Input		[dk]				

## 1. Conexiones de Reloj y Reset:

- `clk_0` es la fuente de reloj para varios componentes.
- `reset` es la entrada de reinicio usada por múltiples componentes.
- Estas señales están etiquetadas como "exported", lo que significa que pueden ser accesibles desde fuera del módulo actual.

## 2. Interfaces Maestras y Esclavas:



- `data_master` e `instruction_master` son interfaces de memoria mapeada Avalon que probablemente se conectan al procesador Nios II para acceso a datos e instrucciones.
- `debug_mem_slave` y `custom_instruction_master` están relacionados con el depurador y posiblemente con la ejecución de instrucciones personalizadas.

### 3. Memoria On-Chip:

- `onchip_mem` es la memoria en chip configurada como RAM, con su base y límite de direcciones especificados, permitiendo el almacenamiento de datos y/o instrucciones del programa.

### 4. Interfaz UART:

- `jtag_uart` se muestra como una interfaz UART utilizada para la comunicación posiblemente para depuración o interacción con una PC.

### 5. Temporizador y Sistema de Interrupción:

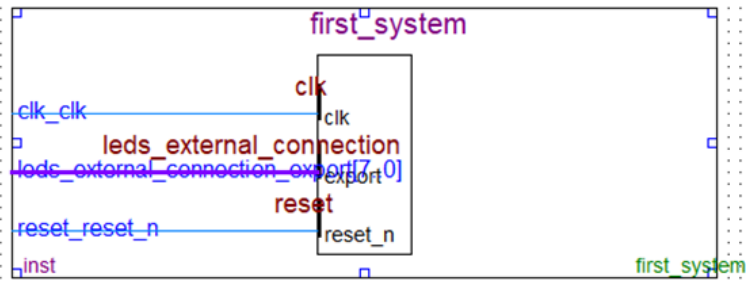
- `sys_clk_timer` es un temporizador que probablemente se use para funciones de temporización o para generar interrupciones periódicas.
- `irq` es la línea de interrupción que puede ser usada por varios periféricos para señalar al procesador eventos que requieren atención.

### 6. GPIO para LEDs:

- `led` es probablemente una interfaz de E/S general (GPIO) configurada para controlar LEDs. La base de direcciones indica donde en el espacio de direcciones los datos de esta interfaz pueden ser leídos o escritos para controlar los LEDs.

Este mapeo de señales y asignaciones de direcciones es fundamental en la configuración del sistema en un chip FPGA, ya que define cómo las diferentes partes del diseño se comunican entre sí y con el mundo exterior.

## Resultados



1. **Nombre del Módulo:** "first\_system" es probablemente el nombre del módulo o componente principal que está siendo configurado o diseñado.
2. **Señales de Reloj y Reset:**
  - `clk_clk` representa la entrada de señal de reloj para el sistema. Los sistemas digitales requieren una señal de reloj para la temporización de las operaciones.
  - `reset_reset_n` indica una señal de reset negada (activo bajo), que se usa para reiniciar el sistema a un estado inicial conocido.
3. **Conexiones Externas:**
  - `leds_external_connection_export[7:0]` sugiere una interfaz de 8 bits que se conecta a algo externo relacionado con LEDs, posiblemente una serie de LEDs para mostrar información o estado.
4. **Etiquetas de Conexión:**
  - `clk` (junto al `clk_clk`) indica que esta señal de reloj probablemente esté conectada o sea accesible fuera de este módulo.
  - `reset` (junto a `reset_reset_n`) sugiere lo mismo para la señal de reset.
  - La etiqueta `export` junto a la conexión de los LEDs implica que estos pines también están accesibles externamente, posiblemente para conectarse a una

matriz física de LEDs en una placa de desarrollo.

## 5. Conexión no Etiquetada:

- Hay una conexión en la parte inferior etiquetada como `_inst` que no tiene ningún cable conectado en la imagen. Esto podría ser una instancia de otro módulo o una interfaz que aún no se ha conectado.

Flow Summary	
<<Filter>>	
Flow Status	Successful - Fri Nov 3 19:53:01 2023
Quartus Prime Version	22.1std.1 Build 917 02/14/2023 SC Lite Edition
Revision Name	NIOS
Top-level Entity Name	NIOS
Family	Cyclone IV E
Device	EP4CE22F17C6
Timing Models	Final
Total logic elements	3,465 / 22,320 ( 16 % )
Total registers	2069
Total pins	10 / 154 ( 6 % )
Total virtual pins	0
Total memory bits	210,048 / 608,256 ( 35 % )
Embedded Multiplier 9-bit elements	6 / 132 ( 5 % )
Total PLLs	0 / 4 ( 0 % )