

Instanciación del microprocesador NIOS V/m en un dispositivo FPGA Cyclone10 modelo 10CL025U256

Kalun Lau

Doctorado en Ingeniería

Mecatrónica

Universidad Nacional de San
Agustín

Arequipa, Perú

klau@unsa.edu.pe

ORCID: 0000-0002-2289-0492

José Becerra

Doctorado en Ingeniería

Mecatrónica

Universidad Nacional de San
Agustín

Arequipa, Perú

jbecerraf@unsa.edu.pe

ORCID: 0000-0001-5098-611X

José Oliden

Doctorado en Ingeniería

Mecatrónica

Universidad Nacional de San
Agustín

Arequipa, Perú

joliden@unsa.edu.pe

ORCID: 0000-0003-2643-327X

Resumen—El avance de la tecnología en el ámbito de la electrónica permite hoy en día personalizar las especificaciones y funcionalidades de un sistema basado en microprocesadores. Esto se logra al usar un dispositivo lógico programable FPGA en conjunto con un núcleo de propiedad intelectual de un microprocesador y un ecosistema de periféricos que lo acompañan.

Index Terms—FPGA, Sistemas Embebidos, Procesadores Soft-core, IoT.

I. INTRODUCCIÓN

En los últimos años las principales empresas dedicadas al rubro de los dispositivos lógicos programables han sido absorbidas por los principales fabricantes de microprocesadores, como es el caso de la fusión entre Altera e Intel [1] y el caso de la fusión entre AMD y Xilinx [2]. Esto constituye un punto de partida en el cambio de paradigma que tienen los actuales fabricantes de microprocesadores donde tradicionalmente orientaban sus esfuerzos en desarrollar dispositivos con mayor índice de desempeño tanto en velocidad como en paralelismo de ejecución de tareas, pero esto ya ha llegado a ser tan complejo que las evoluciones de dichos microprocesadores no presentan mejoras significativas frente a su antecesor.

Esto ha abierto un sinnúmero de debates en foros de discusión en Internet [3] acerca de si hay alguna mejora sustancial y si es que vale la pena migrar hacia una generación más actual o permanecer expectante hasta que pasen al menos dos a tres generaciones más.

Era de esperarse que tanto Intel como AMD (rivales entre ellos) opten por buscar nuevas maneras de innovar los microprocesadores, apostando por los dispositivos lógicos programables para potenciar los microprocesadores, en un inicio como un apoyo a las tareas de ejecución de procesos complejos hasta en un futuro llegar a utilizar un FPGA como base fundamental de los microprocesadores aunque pierdan desempeño [4].

Fabricante	Soft-core CPU
Intel (Altera)	NIOS II, NIOS V
AMD (Xilinx)	MicroBlaze, PicoBlaze
Lattice	Mico8, Mico32

Cuadro I: Principales Fabricantes y sus Soft-core CPUs. Elaboración propia

Sin embargo, al día de hoy podemos "saborear" estas futuras implementaciones de CPU en un FPGA, casi todos los principales fabricantes de FPGA ofrecen un núcleo de propiedad intelectual de un microprocesador soft-core para ser implementados en sus dispositivos FPGA tal como se muestra en el Cuadro I.

Actualmente, para iniciarse en el mundo de los dispositivos lógicos programables FPGA, los mismos fabricantes de estos dispositivos y terceros brindan a los desarrolladores placas de entrenamiento o desarrollo ya que manipular individualmente un FPGA resulta difícil ya que normalmente el desarrollador de sistemas embebidos emplea *breadboards* y dispositivos *thru-hole* en la fase de prototipos. Estas placas de entrenamiento permiten pasar esa barrera ofreciendo a parte del dispositivo FPGA unos periféricos de entrada y salida (LEDs, pulsadores, pines de conexión externa, etc) para que puedan validar los diseños iniciales.

Según Intel, los dispositivos Cyclone II ya están catalogados como dispositivos en obsolescencia [5], [6] y en lo que respecta al soft-core NIOS II, también ha sido reportado como discontinuado [7], [12]. Por lo que se ha seleccionado el procesador soft-core NIOS V/m y un dispositivo FPGA Cyclone 10 LP del fabricante Intel montado en la tarjeta de desarrollo Intel Cyclone 10 LP Evaluation Kit EK10CL025U256 para evitar problemas de obsolescencia en el mediano plazo. La tarjeta de desarrollo mencionada se muestra en la Figura 1.

Debido a que esta tarjeta de desarrollo proviene del mismo fabricante del FPGA (Intel), existe un largo repositorio de

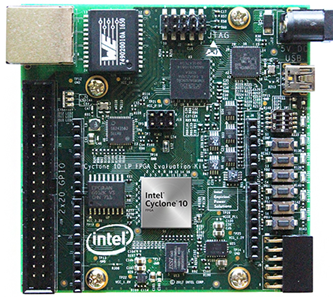


Figura 1: Tarjeta de desarrollo Ep2C5T. Obtenido de Naylamp Mechatronics [8]

documentos técnicos, tutoriales y soporte completo por parte de los softwares de desarrollo, esto representa una ventaja frente a soluciones de tarjetas de desarrollo por parte de terceros.

Otra de las ventajas que tiene la tarjeta de desarrollo Intel Cyclone 10 LP Evaluation Kit EK10CL025U256 de Intel es su bajo costo y disponibilidad. En el distribuidor Mouser dicha tarjeta tiene un precio de \$99.95 sin inclusión de gastos de envío [9]

Se está contemplando usar la versión más actual del software Intel Quartus Prime Lite Edition que al momento de la redacción de este manuscrito es la 23.1.1. Ésta es la que se encuentra disponible de manera gratuita y tiene soporte a dispositivos FPGA Cyclone 10 que se usarán en este artículo.

Considerando las observaciones realizadas, este artículo propone presentar una metodología de desarrollo e implementación de un sistema embebido basado en el microprocesador soft-core NIOS V/m montado en una tarjeta de desarrollo basado en el dispositivo FPGA Cyclone 10 y validar una aplicación básica donde se integre un programa desarrollado en C corriendo sobre dicho microprocesador.

II. ANTECEDENTES

[13] Instala el sistema operativo Linux en un kit de desarrollo Spartan3AN basado en el FPGA Spartan3AN de Xilinx. Con este fin, implementaron el microprocesador soft-core MicroBlaze en el FPGA, utilizaron las memorias RAM y ROM que forman parte del kit de desarrollo. Como sistema operativo utilizaron el kernel uCLINUX y como herramientas de desarrollo utilizaron ISE y Xilinx EDK, además se apoyaron en las herramientas de Petalinux y Petalogix Autoconfig para la sincronización entre hardware y software.

[14] implementa un controlador PID difuso en un FPGA XC4010XL utilizando datos de entrada y salida de 11 bits, convertidores A/D y D/A así como comunicación con la PC a través del puerto paralelo. Realizaron la simulación de un control de nivel logrando obtener resultados satisfactorios.

[15] utiliza FPGAs buscando reducir los tiempos de ejecución en el procesamiento de imágenes. Primero implementan algoritmos de reconocimiento de imágenes en un FPGA con un procesador, luego dos procesadores; y, finalmente, en un sistema con dos procesadores con un coprocesador. Con este fin, utilizan el microprocesador soft-core MicroBlaze y se evaluó el rendimiento de cada caso utilizando la herramienta Testbench de Xilinx. Obtuvieron resultados favorables en la reducción de tiempos de procesamiento.

[16] desarrolla un generador de efectos de audio para la tarjeta XUP Virtex-II PRO de Digilent Inc. en colaboración con Xilinx. Se divide en tres etapas principales: aprender la arquitectura y funcionamiento del FPGA Virtex-II PRO, dominar el lenguaje VHDL y las herramientas de Xilinx para configurar el FPGA, y finalmente, diseñar e implementar el sistema embebido con los efectos de audio como periféricos.

[17] propone un framework como herramienta para programar sistemas de cómputo heterogéneos y un nuevo benchmark para evaluar el rendimiento de los sistemas de cómputo heterogéneos basados en FPGAs.

[18] presenta el diseño de un sistema embebido basado en FPGA para adquirir señales de variables físicas como luz, sonido, temperatura, humedad, CO, CH₄, y C₃H₈ + C₄H₁₀. Estas señales se visualizan en diversas plataformas con interfaces de usuario y se gestionan en una base de datos para generar reportes técnicos sobre las condiciones ambientales. La FPGA actúa como la unidad central del sistema, permitiendo una rápida adquisición de datos con un Scan Rate en nanosegundos, crucial para monitorear cambios ambientales rápidos y tomar decisiones informadas.

[19] integra implementaciones de hardware de algoritmos criptográficos con la finalidad de asegurar redes TCP/IP. Se implementan los algoritmos AES y las funciones resumen SHA-1 y SHA-256. La implementación se realiza como coprocesadores del procesador MicroBlaze utilizando interfaces FSL para el intercambio de datos entre ellos. Finalmente, se realizan pruebas de velocidad utilizando la herramienta OpenVPN.

[20] desarrolla un sistema de ayuda a la conducción mediante visión artificial, implementando un detector de carriles basado en una cámara OV7670 y una FPGA Altera Cyclone IV. Se toman los datos de la cámara y se almacenan en una memoria SDRAM. Así mismo, se configura el protocolo de comunicación RS232 con capacidad de transmisión. Todo este trabajo se realizó utilizando el software Quartus II Lite.

[21] implementa diversas aplicaciones en una tarjeta de FPGA Cyclone EP2C5T empleando el procesador NIOS II y empleando el software Altera Quartus II versión 13.1 sp1, y documenta todo el proceso de desarrollo. Documenta además el proceso de colocar el código de ejecución del procesador

NIOS II en la memoria de configuración logrando que no solamente el diseño de la plataforma NIOS II se ejecute desde dicha memoria de configuración sino que además procede a obtener el código de la aplicación y la traslada a la memoria de la plataforma NIOS II pasando la limitación de volatilidad del FPGA.

III. MARCO TEÓRICO

A. Microprocesador

Es un circuito integrado que cuenta con una unidad central de procesamiento y una unidad de memoria. Para poder funcionar necesita ser conectado a periféricos de entrada y salida. [22].

B. Tipos de Microprocesadores

Existen dos tipos de microprocesadores, los procesadores tipo CISC (complex instruction set computer) el cual cuenta con un número elevado de instrucciones, algunas instrucciones son más complejas y los tiempos de ejecución son mayores. Así mismo, se tienen los procesadores tipo RISC (reduced instruction set computer) la cantidad de instrucciones es reducida y son de simple función, por lo que se ejecutan a mayor velocidad. [22].

C. Arquitectura Harvard

Es una arquitectura de microprocesador que cuenta con dos memorias, una de datos (RAM) y una de programa (FLASH) los buses de comunicación pueden tener distintos tamaños, además, el tiempo de ejecución es menor debido a que se puede superponer la lectura de las instrucciones con la lectura de los datos. [23].

D. Sistema embebido

Se define de esta forma a cualquier sistema computarizado que realice tareas específicas. Pueden ser implementados en microcontroladores o en FPGAs, teniendo estos últimos la ventaja de poder modificar la configuración del hardware. [24].

E. Field Programmable Gate Array (FPGA)

. Es un dispositivo lógico programable con una capacidad tal que soporta la implementación de sistemas lógicos de gran tamaño. Contiene una matriz de bloques lógicos y bloques de Entrada / Salida (E/S). Cada bloque lógico tiene la función de un pequeño sistema con un número limitado de entradas y salidas. [25].

F. Lenguaje de Descripción de Hardware (HDL)

Es el lenguaje utilizado para describir el hardware de un FPGA u otro sistema digital, mediante este es necesario describir el funcionamiento, comportamiento y/o estructura del sistema a implementar, además se debe describir la concurrencia entre los elementos del hardware a implementar. [26].

G. VHDL

Es un acrónimo que viene de VHSIC (very high speed integrated circuits) y de HDL. Este lenguaje permite describir el comportamiento de los sistemas digitales desde niveles muy básicos hasta niveles muy complejos. [27].

IV. SOLUCIÓN PROPUESTA

La solución propuesta contempla la descripción de una metodología para lograr implementar un sistema embebido basado en un microprocesador soft-core, esto conlleva a

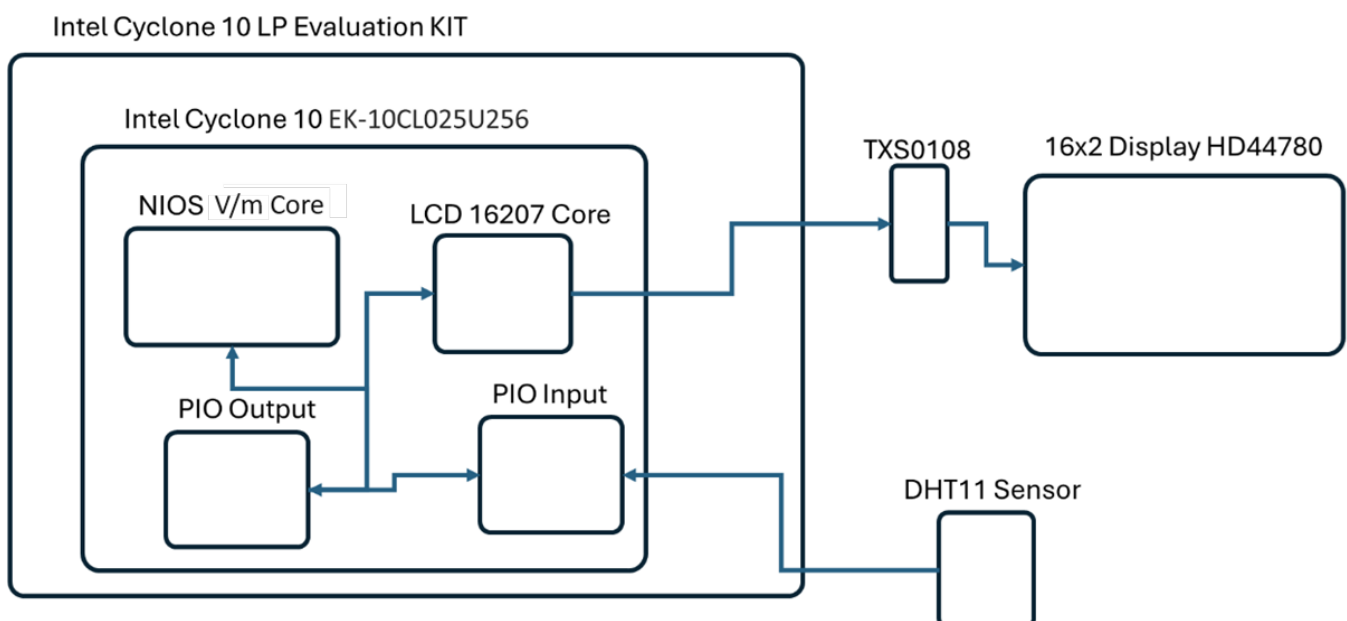


Figura 2: Diagrama de bloques pictórico
Elaborado por: Elaboración propia

desarrollar inicialmente el hardware en el entorno de desarrollo Quartus Prime empleando como base el procesador NIOS V/m y un dispositivo FPGA Cyclone 10, posteriormente se desarrollará en el software Ashling RiscFree una aplicación básica para que sea ejecutada por el procesador NIOS V/m tal como se muestra en la Figura 2.

Para el desarrollo de la plataforma NIOS II se empleará la herramienta QSys en donde se establecerán los siguientes puntos:

- 1) Modelo de procesador NIOS II
- 2) Memoria
- 3) Puertos de entrada y salida
- 4) Puerto de depuración

Se está contemplando el lenguaje VHDL como estándar descriptor de circuitos en el software Intel Quartus Prime ya que luego de desarrollar la plataforma NIOS V/m, éste deberá de instanciarse dentro del proyecto y para ello se requiere de usar un lenguaje HDL para hacer las conexiones con demás módulos y con el dispositivo FPGA.

Las pruebas de validación del funcionamiento del procesador NIOS V/m se harán con el desarrollo de una aplicación básica de interacción de entradas y salidas empleando un protoboard externo y dispositivos LEDs y pulsadores. Para esto se requerirá de hacer un programa en lenguaje C y éste sea compilado y almacenado en la memoria de programa de la plataforma NIOS V/m implementada.

V. METODOLOGÍA

A. Resumen y análisis de investigación

En esta etapa, se realiza una revisión exhaustiva de la literatura existente sobre lenguaje VHDL, arquitectura de computadoras, el procesador NIOS, lenguaje C para sistemas embebidos y aplicaciones basadas en microcontroladores. Se recopilan y analizan datos relevantes, investigaciones previas, tecnologías y prácticas recomendadas para informar el diseño y la implementación del proyecto.

B. Diseño de la plataforma NIOS V/m en Quartus Prime

Se elaborará la plataforma NIOS V/m en el entorno de desarrollo Platform Designer (anteriormente denominado QSys) dentro del software Intel Quartus Prime según requerimientos de la aplicación a desarrollar y las capacidades del dispositivo FPGA Cyclone 10CL025U256.

C. Instanciación del NIOS V en el proyecto en Quartus Prime

En esta sección se instancia la plataforma desarrollada en el anterior punto dentro del proyecto en el software Intel Quartus Prime, para ello se requerirá de confeccionar un archivo HDL de estilo estructural y que permitirá llamar como componente al NIOS V/m y demás periféricos para luego realizar la interconexión entre ellos y con los puertos del FPGA.

D. Desarrollo de la aplicación a correr en el NIOS V/m en Ashling RiscFree

En esta etapa, se elaborará la aplicación que estará ejecutando el NIOS V/m, para ello se elabora el diagrama de flujo y posteriormente se codifica en un lenguaje de programación. Se ha contemplado usar el lenguaje C y la plataforma Ashling RiscFree como entorno de desarrollo de software. Cabe destacar que la herramienta Ashling RiscFree proveída por el fabricante contiene las herramientas de compilación y depuración para el procesador NIOS V/m.

E. Etapa de pruebas y validación

Para las pruebas se contemplará conectar la tarjeta de desarrollo a la computadora mediante una interface USB-JTAG (Intel/Altera USB-Blaster) y en la sección de entradas y salidas se conectará una placa de prototipos breadboard donde se conectarán LEDs y pulsadores, ellos servirán para interactuar con la aplicación ejemplo desarrollado en el ítem anterior.

F. Planos y documentación

Finalmente, se elaboran los planos técnicos detallados y la documentación del proyecto, que incluyen lista de materiales, diagramas de circuito, diagramas de flujo, manuales de operación y cualquier otra información relevante para la implementación y operación del sistema. Esta documentación proporciona una guía clara y completa para futuras referencias y modificaciones.

VI. DESARROLLO

A. Diseño de la plataforma NIOS V/m

En esta primera etapa se implementa el circuito de prueba, haciendo las conexiones tanto entre el computador y la tarjeta de desarrollo EK-10CL025U256 como de ésta hacia la base de prototipos breadboard donde estarán los componentes externos de la aplicación básica de validación.

Como se muestra en la Figura 3, la tarjeta de desarrollo cuenta con periféricos de apoyo a la labor del FPGA como una fuente de reloj de 50MHz, conectividad Ethernet, memoria Flash, un CPLD que se encarga de añadir entradas de señales analógicas, y borneras para conectar periféricos externos a la tarjeta.

La tarjeta de desarrollo posee una entrada de fuente de alimentación de 5VDC mientras que las señales de E/S que tiene el FPGA Cyclone 10 es de 3.3 VDC, muy importante debido a que los periféricos comúnmente utilizados en aplicaciones con microcontroladores trabajan con niveles lógicos de 5VDC. Para ello se usan módulos de conversión de niveles lógicos, para los ensayos de este artículo se emplea un módulo conversor basado en el circuito integrado TXS0108B.

Para la aplicación básica se ha contemplado conectar un sensor de humedad relativa y temperatura DHT11 [28] y un display LCD alfanumérico con controlador Hitachi HD44780 [29], esta implementación esta mostrado en la Figura 4.

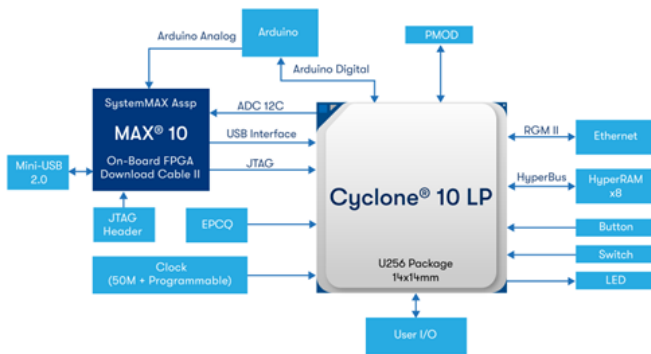


Figura 3: Diagrama de bloques de la tarjeta de desarrollo Intel Cyclone 10 LP Evaluation Kit EK10CL025U256. Por Intel [8]

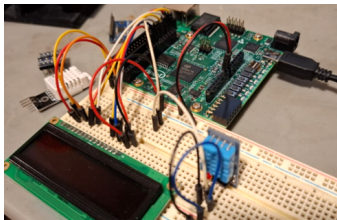


Figura 4: Hardware de prueba para la aplicación básica

B. Instanciación del NIOS V en el proyecto en Quartus Prime

El software Quartus Prime Lite mostrado en la Figura 5 es el entorno de desarrollo que nos facilita el fabricante Intel para hacer las aplicaciones con sus dispositivos lógicos programables, dentro de él tenemos diversas herramientas de trabajo que se detallan a continuación los principales:

- Project Wizard: Asistente para la creación de proyectos.
- IP Catalog: Catálogos de modelos funcionales de circuitos y sistemas digitales encasillados como núcleos de propiedad intelectual (IP Cores)
- Platform Designer: Anteriormente denominado QSys, es el asistente para el desarrollo de un sistema embebido basado en microprocesador a partir de componentes.
- Ashling RiscFree™ IDE: Entorno de desarrollo para construir aplicaciones de software que van a ser ejecutadas por el procesador NIOS V.
- Pin Planner: Herramienta que permite la asignación de pines del FPGA con los puertos declarados en el proyecto.
- Programmer: Herramienta que permite cargar el proyecto compilado en el dispositivo lógico programable a través de diversas interfaces, una de ellas es el USB Blaster que emplea JTAG en la comunicación.
- Questa*-Intel® [11]: El simulador incluido en la versión 23.1.1 basado en el núcleo EDA Questa* de Siemens que soporta simulación comportamental, testbench de código HDL y otros.

Se creará un proyecto teniendo especial cuidado en el formato del nombre del proyecto y la ruta de acceso: que sea corto, no tenga espacios y sin caracteres especiales. Para ello se recomienda utilizar el Asistente de creación de proyectos (Project Wizard) en donde a parte del nombre se

debe de seleccionar el dispositivo FPGA que será utilizado posteriormente.

En el entorno Platform Designer inicialmente se llama al procesador NIOS V, durante la configuración del NIOS V se debe de especificar el tipo /m, luego los demás componentes esenciales para el funcionamiento del NIOS V como son la memoria, el depurador JTAG y los puertos de E/S.

Luego se harán las conexiones necesarias entre los componentes llamados empleando la matriz de conexiones ubicada en la sección izquierda del Platform Designer.

Finalmente se deberá de mapear correctamente las direcciones de acceso para evitar conflicto entre los componentes. El resultado de todo el llamado de componentes y su respectiva conexión y configuración se muestra en la Figura 6.

Al culminar el desarrollo del NIOS V/m, éste debe de ser instanciado dentro del proyecto actual en el Quartus Prime usando un programa desarrollado en VHDL Estructural, aquí también se detallan las conexiones entre el NIOS V/m y las conexiones a los puertos declarados en la entidad como se muestra en la Figura 7.

El siguiente paso es hacer una compilación previa para que el Quartus Prime actualice las aplicaciones internas atendiendo la ventana de reporte de mensajes, sobre todo el Pin Planner (ver Figura 8) quien es el encargado de establecer las asignaciones de pines del FPGA en el proyecto. Se debe de tener en cuenta que dicha asignación proviene de los puertos de la entidad declarada en el VHDL Estructural.

Una vez asignados los pines del FPGA con los puertos del proyecto se realiza una nueva compilación para que se actualice y se obtengan los archivos bitstream que serán utilizados en el proceso de grabación del dispositivo FPGA.

Para la etapa de grabado del proyecto en el FPGA se utiliza la aplicación interna del Quartus Prime denominado Programmer como se muestra en la Figura 9, éste se encargará de detectar los dispositivos del fabricante que se encuentren conectados al computador en sus diferentes interfaces, entre ellas el JTAG que es el que está soportado por la tarjeta de desarrollo Intel Cyclone 10 LP Evaluation Kit EK10CL025U256 para luego brindar las diferentes opciones de configuración de los dispositivos detectados. Para el caso del FPGA Cyclone 10CL025U256, se llamará al archivo .sof que fué generado producto de la compilación del proyecto y se procederá a grabar al hacer clic en el botón Start.

En esta etapa en procesador NIOS V/m estará implementado en la tarjeta de desarrollo Intel Cyclone 10 LP Evaluation Kit EK10CL025U256 y se encontrará listo para recibir una aplicación desarrollada en Ashling RiscFree usando la interfaz

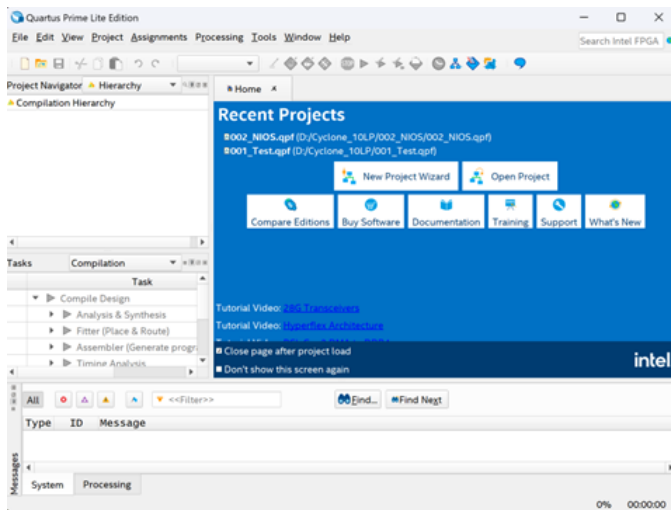


Figura 5: Vista principal del software Intel Quartus Prime Lite versión 23.1.1

```

1 library IEEE;
2 use IEEE.std_logic_1164.all;
3 use IEEE.std_logic_arith.all;
4 use IEEE.std_logic_unsigned.all;
5
6 entity test_dht11 is
7 port(
8     in_clk      : in  std_logic           := 'X';      -- clk
9     lcd_RS      : out std_logic           := 'X';      -- RS
10    lcd_RW       : out std_logic           := 'X';      -- RW
11    lcd_data     : inout std_logic_vector(7 downto 0) := (others => 'X'); -- data
12    lcd_E        : out std_logic           := 'X';      -- E
13    dht11_data   : inout std_logic
14 );
15 end test_dht11;
16
17 architecture estructura of test_dht11 is
18     component nios_dht11_lcd is
19     port (
20         clk_clk      : in  std_logic           := 'X';      -- clk
21         lcd_16207_0_external_RS : out std_logic := 'X';      -- RS
22         lcd_16207_0_external_RW : out std_logic := 'X';      -- RW
23         lcd_16207_0_external_data : inout std_logic_vector(7 downto 0) := (others => 'X'); -- data
24         lcd_16207_0_external_E   : out std_logic := 'X';      -- E
25         dht11_data_export        : inout std_logic := 'X';      -- export
26     );
27 end component nios_dht11_lcd;
28
29 begin
30     u0 : component nios_dht11_lcd
31     port map (
32         clk_clk      => in_clk,      -- clk.clk
33         lcd_16207_0_external_RS => lcd_RS, -- lcd_16207_0_external_RS
34         lcd_16207_0_external_RW => lcd_RW, -- lcd_16207_0_external_RW
35         lcd_16207_0_external_data => lcd_data, -- lcd_data
36         lcd_16207_0_external_E   => lcd_E, -- lcd_E
37         dht11_data_export        => dht11_data -- dht11_data.export
38     );
39 end estructura;

```

Figura 7: Código VHDL Estructural para la instanciación del procesador NIOS dentro del proyecto en Quartus Prime.

Connect...	Name	Description	Export	Clock	Base	End
clk_0	clk_0	Clock Source	clk	reset	exported	
clk_in	clk_in	Clock Input	clk	reset	exported	
clk_in_reset	clk_in_reset	Reset Input	clk	reset	exported	
clk	clk	Clock Output	clk	reset	exported	
clk_reset	clk_reset	Reset Output	clk	reset	exported	
intel_niosv_m_0	intel_niosv_m_0	Nios V/m Microcontroller Intel FPGA IP	clk	reset	exported	
reset	reset	Reset Input	clk	reset	exported	
platform_irq_rx	platform_irq_rx	Interrupt Receiver	clk	reset	exported	
timer_sw_agent	timer_sw_agent	Avalon Memory Mapped Slave	clk	reset	exported	
instruction_manager	instruction_manager	AXI4Lite Master	clk	reset	exported	
data_manager	data_manager	AXI4Lite Master	clk	reset	exported	
dm_agent	dm_agent	Avalon Memory Mapped Slave	clk	reset	exported	
onchip_memory2_0	onchip_memory2_0	On-Chip Memory (RAM or ROM) Intel FPGA IP	clk	reset	exported	
clk1	clk1	Clock Input	clk	reset	exported	
s1	s1	Avalon Memory Mapped Slave	clk	reset	exported	
reset1	reset1	Reset Input	clk	reset	exported	
jtag_uart_0	jtag_uart_0	JTAG UART Intel FPGA IP	clk	reset	exported	
clk	clk	Clock Input	clk	reset	exported	
avalon_jtag_slave	avalon_jtag_slave	Avalon Memory Mapped Slave	clk	reset	exported	
irq	irq	Interrupt Sender	clk	reset	exported	
p10_0	p10_0	PIO (Parallel I/O) Intel FPGA IP	clk	reset	exported	
clk	clk	Clock Input	clk	reset	exported	
reset	reset	Reset Input	clk	reset	exported	
s1	s1	Avalon Memory Mapped Slave	clk	reset	exported	
external_connection	external_connection	Conduit	clk	reset	exported	
p10_1	p10_1	PIO (Parallel I/O) Intel FPGA IP	clk	reset	exported	
clk	clk	Clock Input	clk	reset	exported	
reset	reset	Reset Input	clk	reset	exported	
s1	s1	Avalon Memory Mapped Slave	clk	reset	exported	
external_connection	external_connection	Conduit	clk	reset	exported	

Figura 6: Configuración de una plataforma basada en el procesador NIOS V/m dentro del entorno Platform Designer del Quartus Prime

de depuración JTAG implementada previamente en la etapa de configuración del NIOS V en el entorno Platform Designer.

C. Desarrollo de la aplicación a correr en el NIOS V/m

La aplicación básica a desarrollar será la de obtener el valor de la temperatura y la humedad relativa del ambiente proveniente del sensor DHT11 y de visualizar dicha información en el display alfanumérico 16x2 previamente formateado a unidades estándar con una tasa de refresco de un segundo, esto se encuentra detallado como diagrama de flujo en la figura 10.

Para la codificación de la aplicación básica se plantea el lenguaje C embebido y uso del entorno Ashling RiscFree (mostrado en la Figura 11) que es la actual aplicación reemplazando el entorno Eclipse usado anteriormente cuando se desarrollaban aplicaciones con el procesador NIOS II.

Para el desarrollo de la aplicación básica es necesario que el RiscFree tenga configurado el BSP del procesador NIOS V

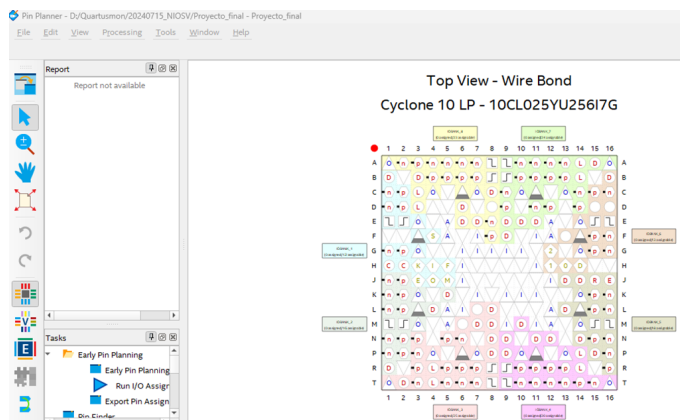


Figura 8: Vista de los pines del dispositivo FPGA Cyclone 10CL025U256 en el entorno Pin Planner del Quartus Prime.

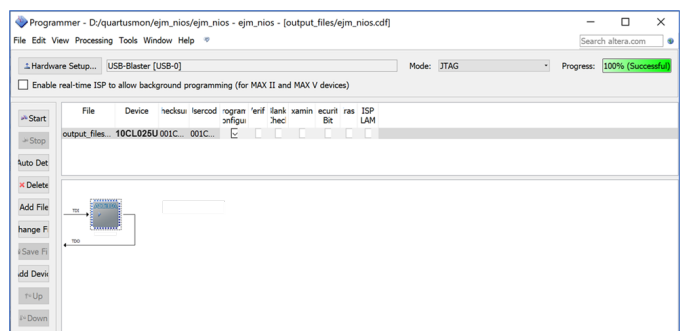


Figura 9: Entorno Programmer del Quartus Prime para grabar el dispositivo lógico programable.

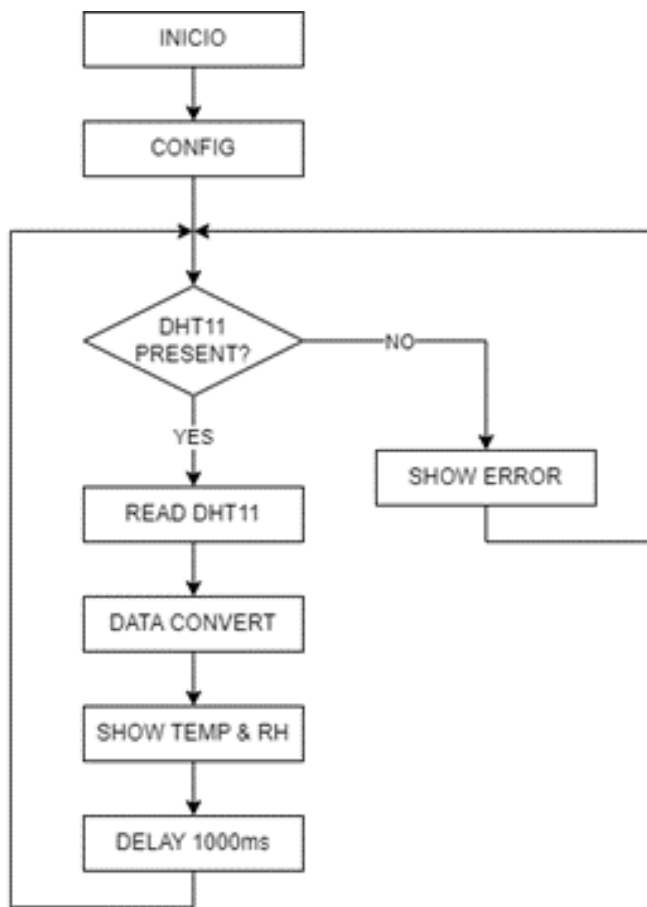


Figura 10: Diagrama de flujo de la aplicación básica a ser ejecutada por el procesador NIOS V instanciado.

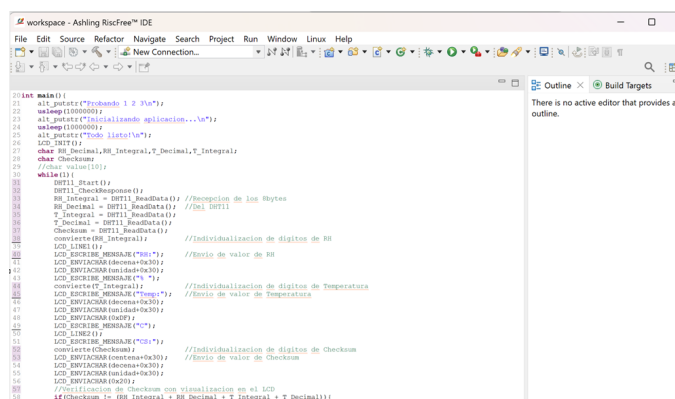


Figura 11: Entorno Ashling RiscFree para el desarrollo de aplicaciones con el NIOS V.

desarrollado anteriormente. Luego se entabla comunicación mediante el JTAG UART.

Una vez realizado el código de la aplicación se procede a compilar y grabar en la memoria del procesador NIOS V empleando la opción “Run as NIOS V hardware”.

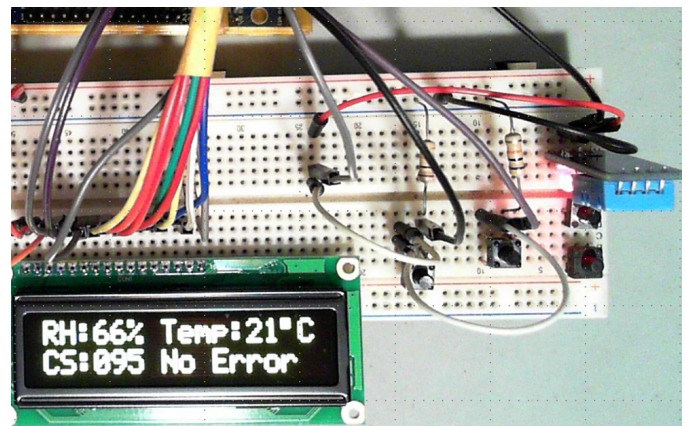


Figura 12: Pruebas funcionales de la aplicación básica con el NIOS V.

D. Etapa de pruebas y validación

En esta etapa se busca inicialmente verificar funcionalmente el funcionamiento de la aplicación básica, que es la de primero entablar la comunicación correcta con el sensor DHT11, el formateo de los valores a unidades en grados centígrados y porcentaje de humedad relativa para finalmente visualizarlos en el display LCD mediante uso de comandos específicos al controlador HD44780.

Las pruebas funcionales se muestran en la Figura 12.

VII. CONCLUSIONES

- Se confirmó la implementación de la plataforma de procesador basado en el NIOS V/m e implementada en un dispositivo FPGA Cyclone 10, también se desarrolló una aplicación básica de lectura de un sensor DHT11 (temperatura y humedad relativa del ambiente) y la correspondiente visualización en un display alfanumérico 16x2 con controlador HD44780, desarrollado en el entorno Ashley RiscFree y todo ello desarrollado en el Intel Quartus Prime versión 23.1.1.
- De acuerdo a los antecedentes revisados, es factible utilizar esta implementación para instalar sistemas operativos como Linux para realizar el control y supervisión de variables de proceso. Así mismo, utilizando los sistemas soft-core es factible optimizar los tiempos de respuesta de un sistema de control PID difuso.

REFERENCIAS

- [1] Intel Press. (2015). Intel Completes Acquisition of Altera [Online]. Available: <https://www.intel.com/news-events/press-releases/detail/302/intel-completes-acquisition-of-altera>
- [2] AMD Press. (2022). AMD Completes Acquisition of Xilinx [Online]. Available: <https://www.amd.com/en/newsroom/press-releases/2022-2-14-amd-completes-acquisition-of-xilinx.html>
- [3] Quora. (2023). Has the Intel CPU generation really helped in the performance of PCs today? [Online]. Available: <https://www.quora.com/Has-the-Intel-CPU-generation-really-helped-in-the-performance-of-PCs-today#:~:text=Intel's%20CPU%20generations%20have%20significantly,faster%20and%20more%20efficient%20processing>
- [4] Quora. (2019). Is It Possible To Make a CPU With An FPGA? [Online]. Available: <https://www.quora.com/Is-it-possible-to-make-a-CPU-with-an-FPGA>

- [5] Intel. (2024). Quartus® Software Latest Version for Device Support [Online]. Available: <https://www.intel.com/content/www/us/en/support/programmable/support-resources/design-software/devices-support.html#cyclone%C2%AEseries>
- [6] Intel. (2024). PDN2401 Intel® Discontinuance Notification for selected Field Programmable Gate Array (FPGA) and Complex Programmable Logic Device (CPLD) [Online]. Available: <https://www.intel.com/content/www/us/en/content-details/813536/pdn2401-intel-discontinuance-notification-for-selected-field-programmable-gate-array-and-complex-programmable-logic-device-cpld.html>
- [7] Intel. (2024). Intel is discontinuing IP ordering codes listed in PDN2312 for Nios® II IP [Online]. Available: <https://www.intel.com/content/www/us/en/content-details/781327/intel-is-discontinuing-ip-ordering-codes-listed-in-pdn2312-for-nios-ii-ip.html>
- [8] Intel. (2020). Intel® Cyclone® 10 LP FPGA Evaluation Kit [Online]. Available: <https://www.intel.com/content/www/us/en/products/details/fpga/development-kits/cyclone/10-lp-evaluation-kit.html>
- [9] Mouser.com. (2024). Altera EK-10CL025U256 Programmable Logic IC Development Tools Cyclone 10 FPGA 10CL025 Evaluation Kit [Online]. Available: <https://www.mouser.com/ProductDetail/Altera/EK-10CL025U256?qs=HXFqYaX1Q2zh%2FEGiixG4LQ%3D%3D>
- [10] Intel. (2024). Intel® Quartus® Prime Lite Edition Design Software Version 23.1.1 for Windows [Online]. Available: <https://www.intel.com/content/www/us/en/software-kit/825278/intel-quartus-prime-lite-edition-design-software-version-23-1-1-for-windows.html>
- [11] Intel. (2024). Questa*-Intel® FPGA Edition Software [Online]. Available: <https://www.intel.com/content/www/us/en/software/programmable/quartus-prime/questa-edition.html>
- [12] Intel. (2020). Intel® Cyclone® 10 LP FPGA Evaluation Kit [Online]. Available: <https://www.intel.com/content/www/us/en/developer/topic-technology/edge-5g/hardware/fpga-intel-cyclone10-lp.html>
- [13] P. E. Calleja, M. E. Iglesias and J. F. Carmona, "Linux embebido en FPGA para sistemas de monitoreo industrial, Revista Cubana de Ciencias Informáticas, vol. 7, no. 1, pp. 44-54, 2013.
- [14] L. J. López, F. M. Zulay and A. S. Pateti, "Metodología de implementación de un controlador PID difuso en una FPGA, Universidad, Ciencia y Tecnología, vol. 10, no. 39, pp. 130-133, 2006.
- [15] J. R. Osio, "Procesamiento digital de imágenes médicas sobre plataformas FPGAs," Tesis de Maestría, Departamento de Ingeniería, Universidad Nacional de La Plata, La Plata, Argentina, 2023. Available: <https://doi.org/10.35537/10915/83001>
- [16] J. León, "Diseño e implementación de un sistema embebido generador de efectos de audio usando FPGA," Tesis de Ingeniería, Departamento de Ingeniería, Universidad Peruana de Ciencias Aplicadas, Lima, Perú, 2008. Available: <http://hdl.handle.net/10757/579590>
- [17] R. Valdez and Y. Maldonado, "Computación Heterogénea y FPGAs como Aceleradores Eficientes," in N. Callaos, J. Horne, E. F. Ruiz-Ledesma, B. Sánchez, and A. Tremante (Eds.), Memorias de la Décima Segunda Conferencia Iberoamericana de Complejidad, Informática y Cibernética: CICIC 2022, pp. 25-29, International Institute of Informatics and Cybernetics, 2022. Available: <https://doi.org/10.54808/CICIC2022.01.25>
- [18] P. G. Encalada Ruíz, M. A. Córdova Suárez, O. E. Ruíz Robalino, J. G. Vega Pérez, T. de los Ángeles Liger Manzano, and L. Sánchez Almeida, "Sistema embebido basado en FPGA para el monitoreo de metadatos condiciones ambientales," CD, vol. 2, no. 4, pp. 177-189, Oct. 2018.
- [19] A. Cabrera Aldaya and A. J. Cabrera Sarmiento, "Diseño e integración de algoritmos criptográficos en sistemas empotrados sobre FPGA, Ingeniería Electrónica, Automática y Comunicaciones, vol. 34, no. 3, pp. 41-51, 2013.
- [20] B. Escanilla, "Desarrollo de un sistema de conducción autónoma mediante visión artificial basado en una FPGA," M.S. thesis, Universidad Oberta de Catalunya, Catalunya, España, 2020.
- [21] K. Lau, Repositorio de aplicaciones con el NIOS II en Github". 2021. Available: <https://github.com/tocache/Altera-Cyclone-II-FPGA/tree/master/NIOS%20II>
- [22] P. Quiroga, Arquitectura de computadoras, Buenos Aires: Alfaomega, 2010.
- [23] R. B. Reese, Microprocessors: from assembly language to C using the PIC18Fxx2, Hingham, Mass: Da Vinci Engineering Press, 2005.
- [24] C. GU, Building Embedded Systems, San Diego: Apress, 2016.
- [25] S. Brown and Z. Vranesic, Fundamentals of Digital Logic with VHDL design, New York: McGraw-Hill, 2009.
- [26] F. Vahid, Digital design, with RTL design, VHDL, and Verilog, Hoboken, NJ: Wiley, 2011.
- [27] J. P. Deschamps, G. J. Antoine Bioul and G. D. Sutter, Synthesis of Arithmetic Circuits: FPGA, ASIC, and Embedded Systems, Hoboken, New Jersey: Wiley, 2006.
- [28] Mouser.com. (n.d.). DHT11 Temperature and Relative Humidity Sensor Datasheet [Online]. Available: <https://www.mouser.com/datasheet/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf>
- [29] Sparkfun.com. (n.d.). HD44780U Dot Matrix Liquid Crystal Display Controller/Driver Datasheet [Online]. Available: <https://www.sparkfun.com/datasheets/LCD/HD44780.pdf>
- [30] Intel. (2023). Nios® V Embedded Processor Design Handbook [Online]. Available: <https://www.intel.com/content/www/us/en/docs/programmable/726952/23-1/about-the-embedded-processor.html>
- [31] Intel. (2023). AN 985: Nios® V Processor Tutorial [Online]. Available: <https://cdrdv2-public.intel.com/784469/an-784468-784469.pdf>