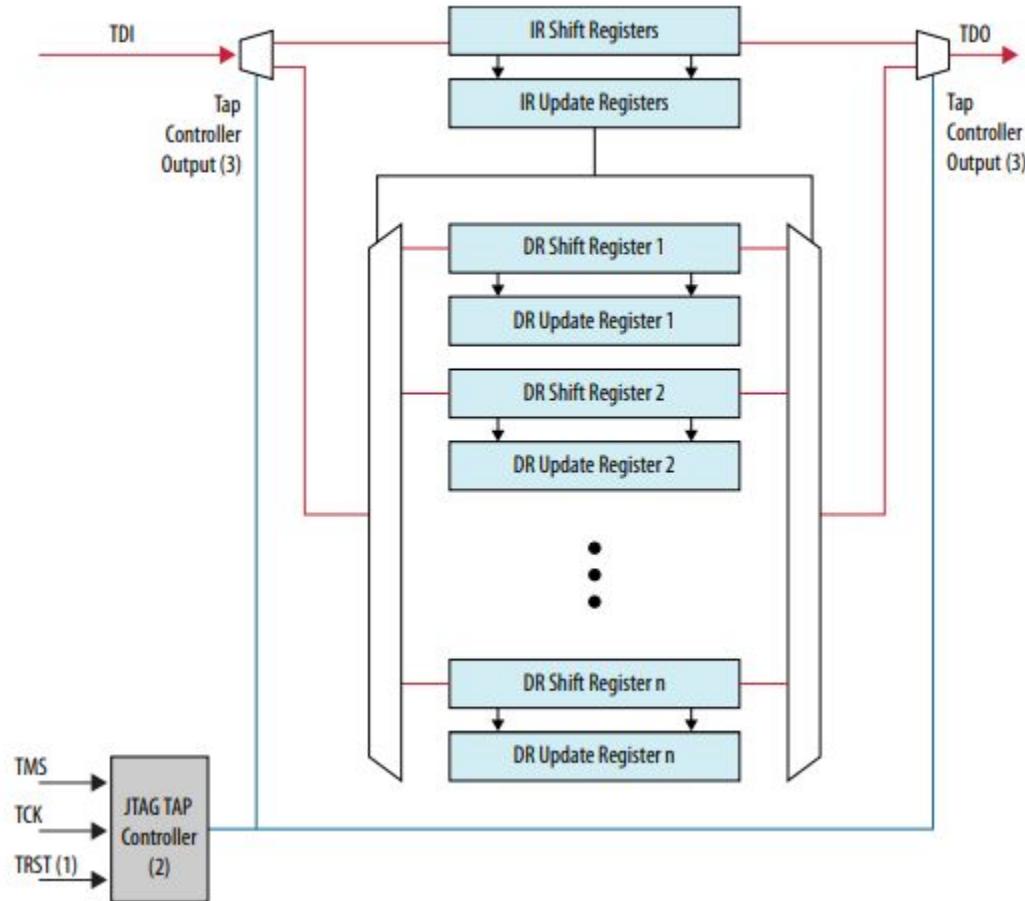

Guía Virtual JTAG

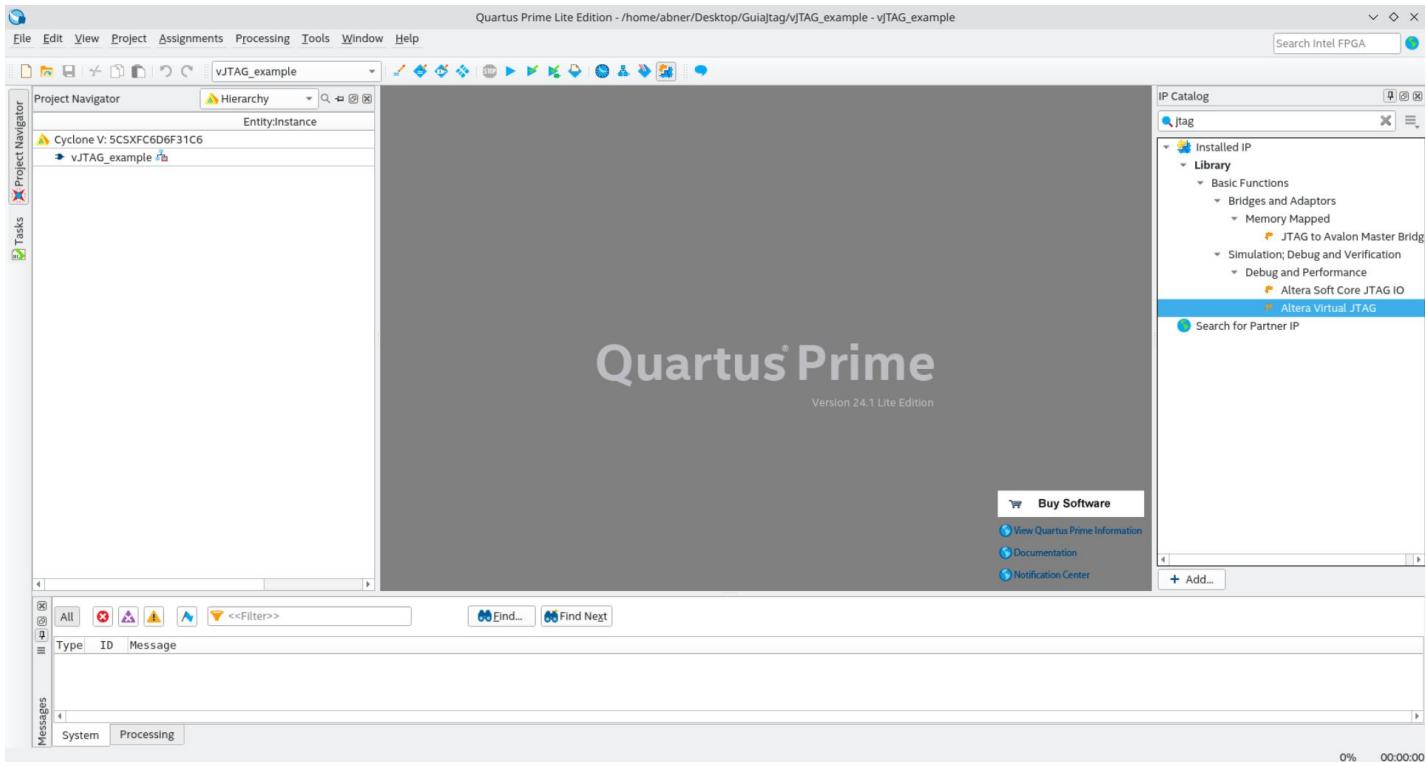
Para DE10-Standard

Functional Model of the JTAG Circuitry



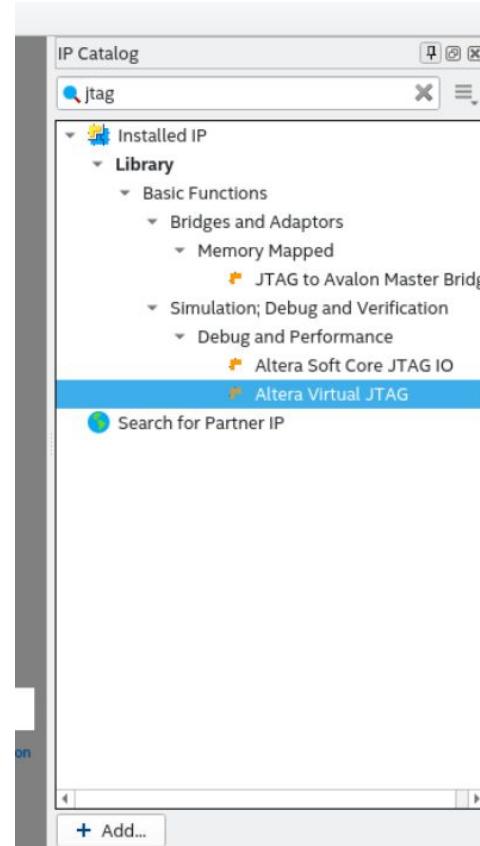
PASO 1

CREE UN
PROYECTO
NUEVO. EN
BLANCO



PASO 2

AGREGUE EL IP DE
ALTERA VIRTUAL
JTAG



PASO 3

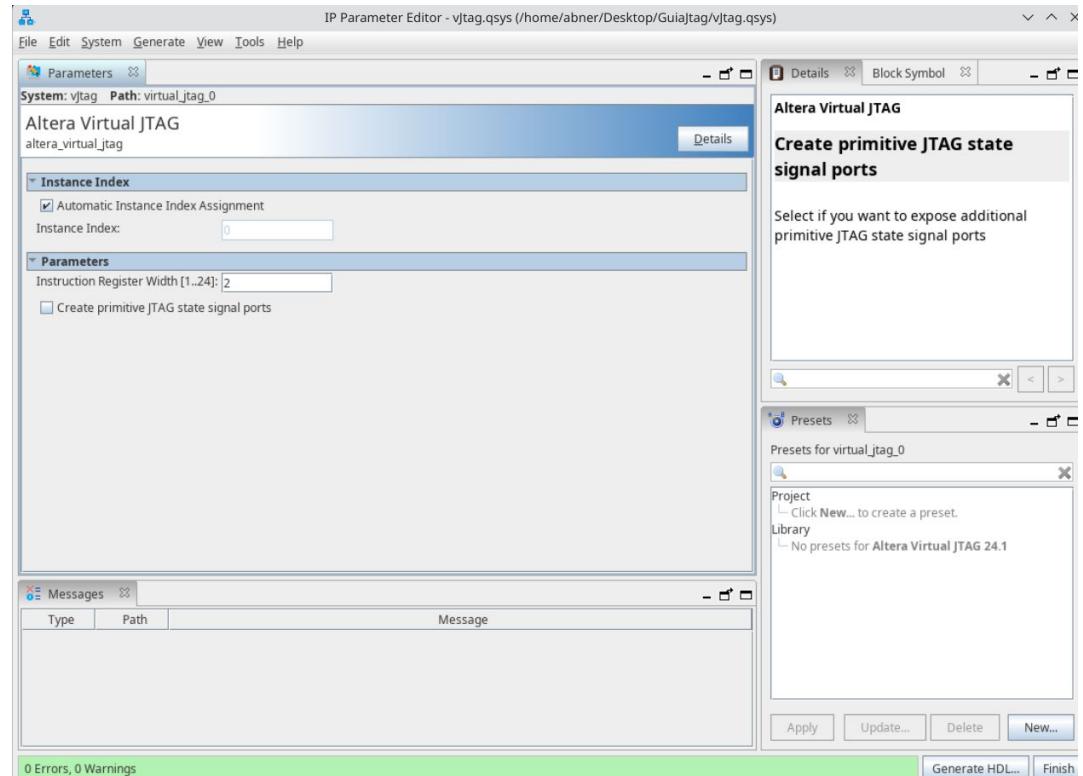
DELE UN NOMBRE
A LA ENTIDAD.
EN ESTE CASO ES
vjtag



PASO 4

SELECCIONE LA CANTIDAD DE BITS DEL **REGISTRO DE INSTRUCCIONES** (ESTO DEPENDERÁ DE LA CANTIDAD DE INSTRUCCIONES QUE USTED NECESITE). EN ESTE CASO SELECCIONAMOS **2**

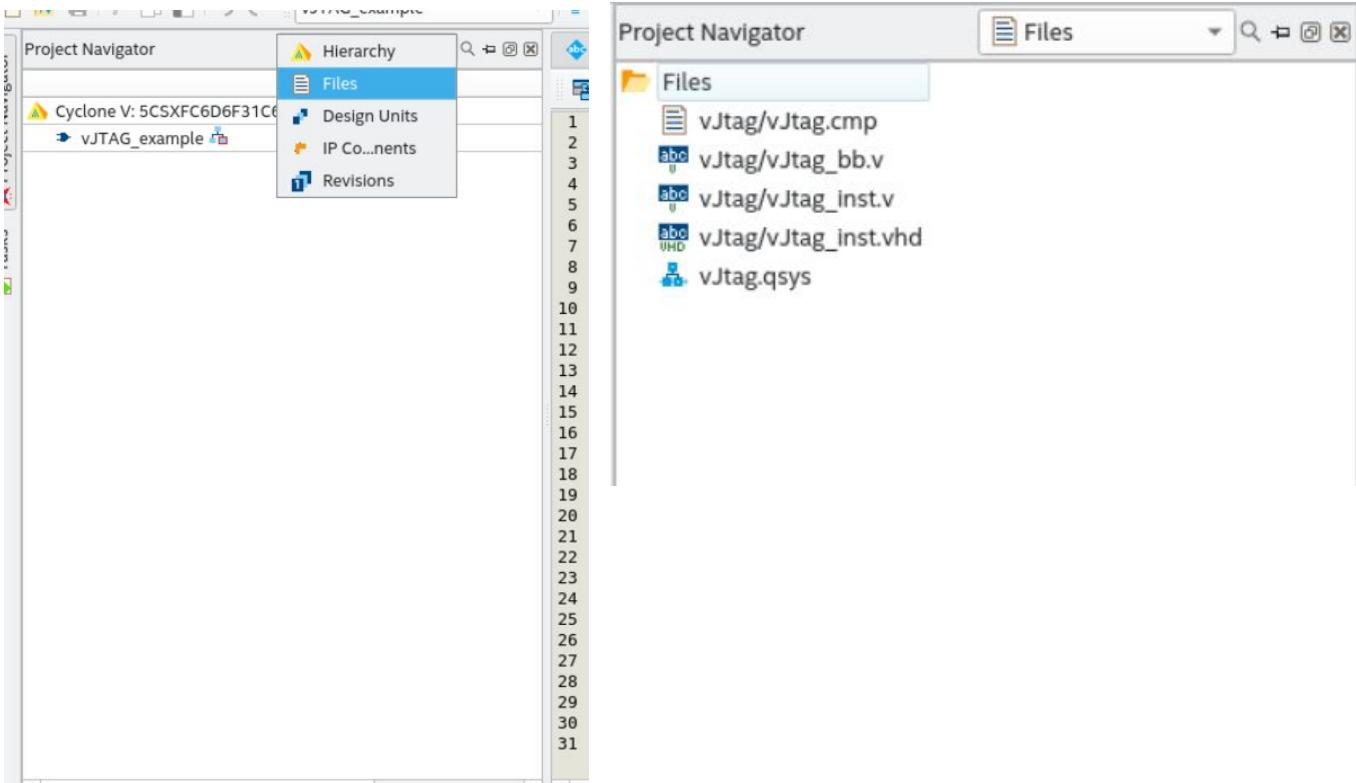
GENERE EL HDL



PASO 5

AGREGAR EL
ARCHIVO
TERMINADO EN
.qsys AL PROYECTO.

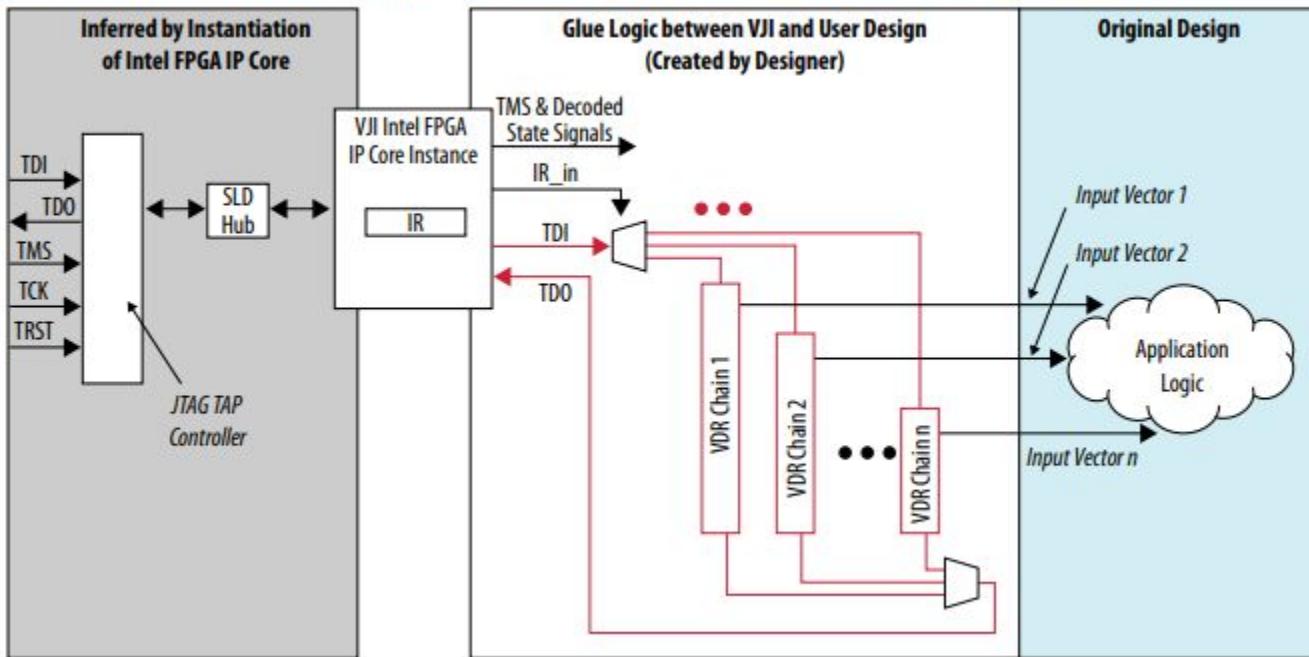
EL ARCHIVO
TERMINADO EN
_inst.v NOS INDICA
LA MANERA DE
INSTANCIARLO



PASO 6

SE SIGUE ESTA ARQUITECTURA.
UNA **INSTANCIA DEL VIRTUAL JTAG CON UNA LÓGICA INTERMEDIA** QUE COMUNIQUE EL MÓDULO CON EL PROYECTO

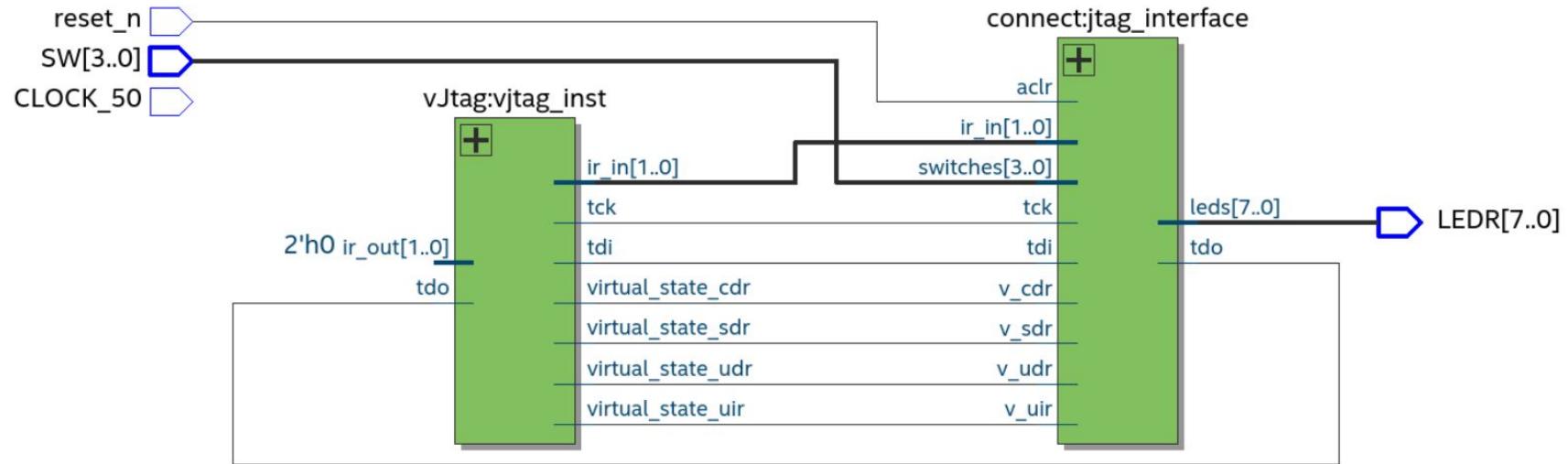
Block Diagram of a Design with a Single Virtual JTAG Instantiation



PROYECTO DE EJEMPLO

- **CONTROL** DE 8 LEDs (PARA EJEMPLIFICAR LA ESCRITURA DE REGISTROS)
- **LEE** 4 SWITCHES (PARA EJEMPLIFICAR LECTURA DE REGISTROS)
- INSTANCIA **vJTAG**
- INSTANCIA **connect** ("GLUE LOGIC")

RTL DEL PROYECTO DE EJEMPLO



PROYECTO DE EJEMPLO SE PUEDE ENCONTRAR EN

<https://github.com/Abner2111/GuiaItag>

PASO 7

- CREE **SCRIPT TCL**
- ESTE SE ENCARGA DE FACILITAR LA COMUNICACIÓN MEDIANTE JTAG
- SE ENCUENTRA UN EJEMPLO BIEN DOCUMENTADO ,EN EL REPO, CON EL NOMBRE **form.tcl**

```
# Función para enviar datos a los LEDs de la FPGA
# Esta función lee el estado de los checkboxes (d0-d7) y los envía como datos binarios
proc send_data {} {
    # Variables globales: d0-d7 representan el estado de los checkboxes, displayData muestra el resultado
    global d0 d1 d2 d3 d4 d5 d6 d7 displayData
    # String que contendrá el valor binario final
    set led ""
    # Representa bit 1
    set one 1
    # Representa bit 0
    set zero 0

    # Construir el string binario basado en el estado de cada checkbox
    # Si el checkbox está marcado (==1), agregar "1", sino agregar "0"
    if {$d0 == 1} {set led $led$one} else {set led $led$zero}
    if {$d1 == 1} {set led $led$one} else {set led $led$zero}
    if {$d2 == 1} {set led $led$one} else {set led $led$zero}
    if {$d3 == 1} {set led $led$one} else {set led $led$zero}
    if {$d4 == 1} {set led $led$one} else {set led $led$zero}
    if {$d5 == 1} {set led $led$one} else {set led $led$zero}
    if {$d6 == 1} {set led $led$one} else {set led $led$zero}
    if {$d7 == 1} {set led $led$one} else {set led $led$zero}

    # Actualizar el texto que se muestra en la interfaz
    set displayData "Data sent: $led"

    # Secuencia de comunicación JTAG con la FPGA
    # Abrir conexión con el dispositivo
    open_port
    # Bloquear dispositivo por 10 segundos máximo
    device_lock -timeout 10000
    # Cambiar a modo de escritura de LEDs (IR=2)
    # device_virtual_ir_shift es un comando específico de Quartus para cambiar el registro IR
    device_virtual_ir_shift -instance_index 0 -ir_value 2 -no_captured_ir_value
    # Enviar los 8 bits de datos a los LEDs
    #device_virtual_dr_shift es un comando específico de Quartus para cambiar el registro DR
    set l [device_virtual_dr_shift -dr_value $led -instance_index 0 -length 8]
    # Imprimir respuesta en consola
    puts $l
    # Regresar a modo inactivo (IR=0)
```

TCL SCRIPTING

PARA MÁS INFORMACIÓN DE SCRIPTING, CONSULTAR LA DOCUMENTACIÓN OFICIAL: Intel® Quartus® Prime Pro Edition User Guide Scripting

<https://cdrv2-public.intel.com/781637/ug-683432-781637.pdf>

PASO 8

ASIGNAR PINES, COMPILAR Y
PROGRAMAR FPGA UNA VEZ
ESTÉ EL PROYECTO LISTO

Tasks

Task		Time
✓	▶ Compile Design	00:01:25
✓	▶ Analysis & Synthesis	00:00:25
✓	▶ Fitter (Place & Route)	00:00:42
✓	▶ Assembler (Generate programming files)	00:00:08
✓	▶ Timing Analysis	00:00:06
✓	▶ EDA Netlist Writer	00:00:04
Edit Settings		
▶ Program Device (Open Programmer)		

Hardware Setup... DE-SoC [1-3]

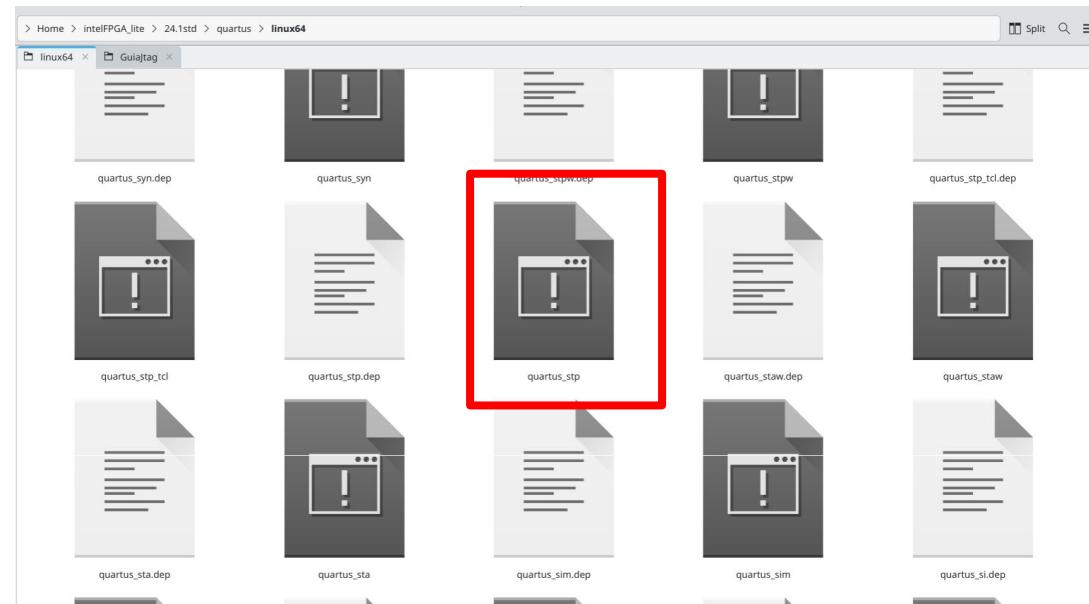
Enable real-time ISP to allow background programming when available

Start	Device	Checksum	Usercode	Program/Configure	Verify	Blank-Check	Examine	Security Bit	Erase	Read	Write	ISPs	CLAs
<none>	SOCVHPS	00000000	<none>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
output_files/vJTAG_...	5CSXFC6D6F31	00B204BC	00B204BC	<input checked="" type="checkbox"/>	<input type="checkbox"/>								

TDI →  → 

PASO 9

- BUSCAR EN LA CARPETA DE INSTALACIÓN DE QUARTUS EL PROGRAMA **quartus_stp.exe** (en windows) y **quartus_stp** (en linux).
- COPIE LA LOCALIZACIÓN PARA CORRERLO EN LA CONSOLA DEL SISTEMA

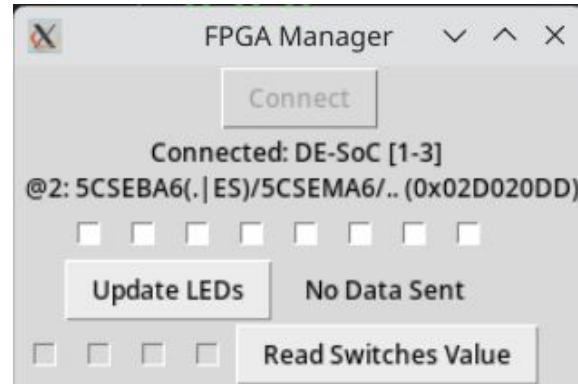
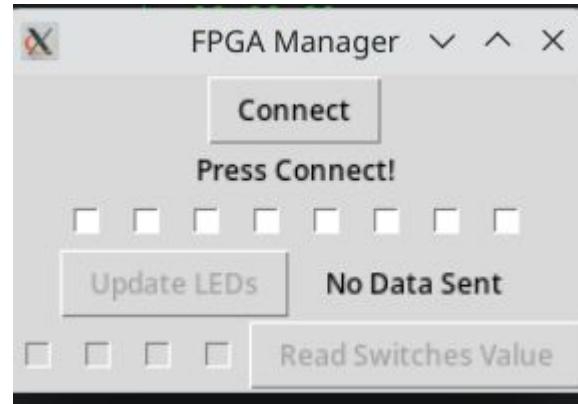


PASO 10

- CON LA FPGA “**FLASHEADA**” Y **CONECTADA** A LA COMPUTADORA CORRA EL PROGRAMA ENCONTRADO EN EL **PASO 9** DE LA SIGUIENTE MANERA:
`<path del Quartus_stp> -t <path del form.tcl>`
- ESTO ABRIRÁ UN **GUI** DEFINIDO EN EL **TCL** PARA COMUNICARSE POR MEDIO DE **JTAG**

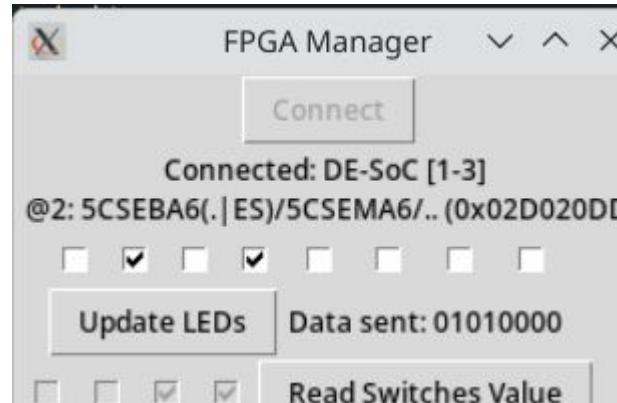
PASO 11

- PRESIONE CONNECT
- YA PUEDE CONTROLAR LOS LEDS Y SWITCHES



PASO 12

- VERIFICAR FUNCIONALIDAD
 - LED 1 Y 3 ENCENDIDOS Y SWITCHES 2 Y 3 ENCENDIDOS



REFERENCIAS

- IEEE Std 1149.1-2013: Estándar de Puerto de Acceso de Prueba y Arquitectura de Escaneo de Frontera
- Guía del Usuario del Núcleo IP Virtual JTAG de Altera
- Manual de Referencia del Lenguaje SystemVerilog IEEE 1800-2017
- Virtual JTAG Intel® FPGA IP Core User Guide
- Intel® Quartus® Prime Pro Edition User Guide Scripting
- <https://idle-logic.com/2012/04/15/talking-to-the-de0-nano-using-the-virtual-jtag-interface/>