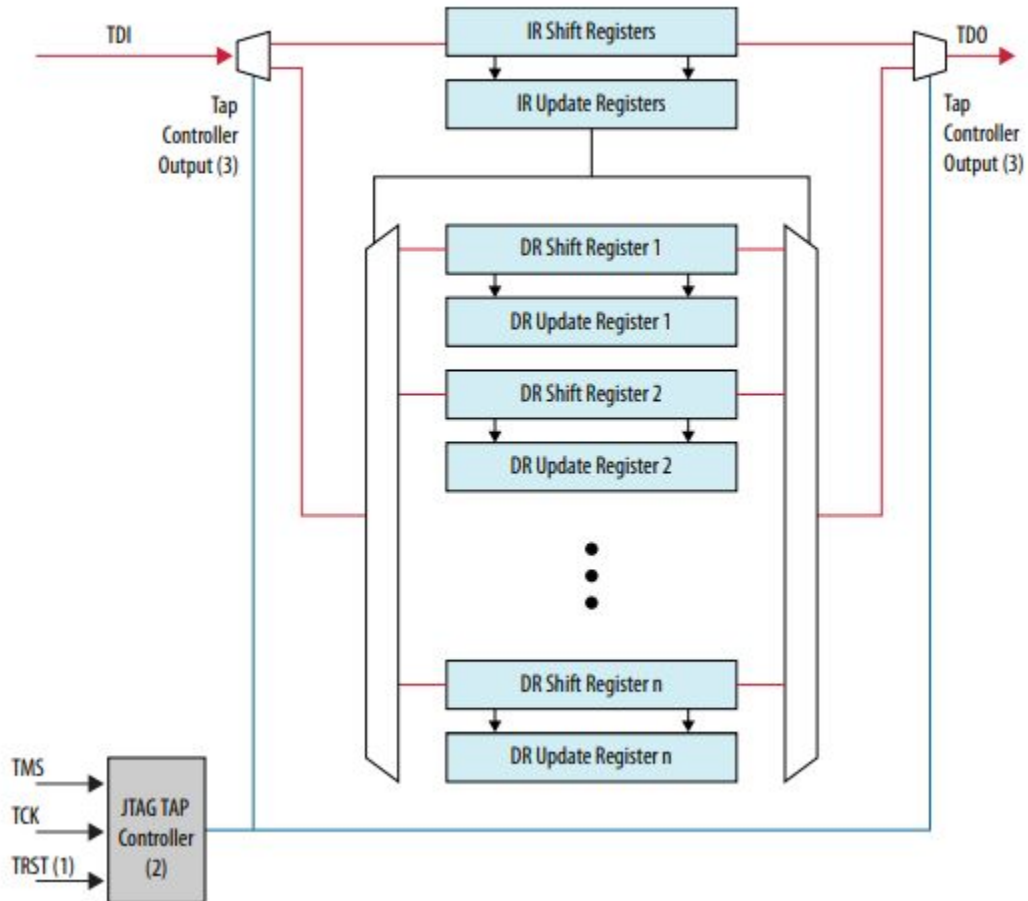

Guía Virtual JTAG

— Para DE10-Standard —

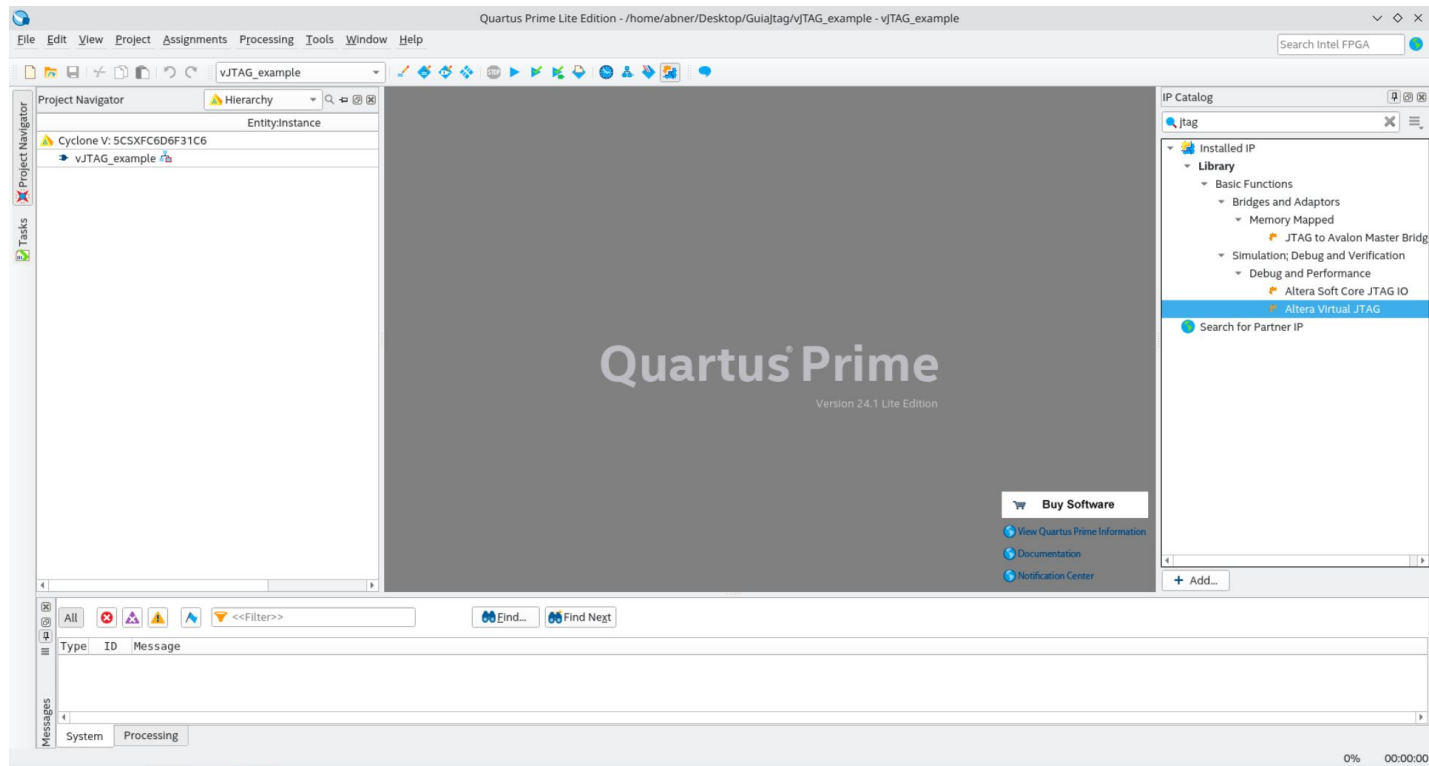
Abner Arroyo Quesada
Ingeniería en computadores

Functional Model of the JTAG Circuitry



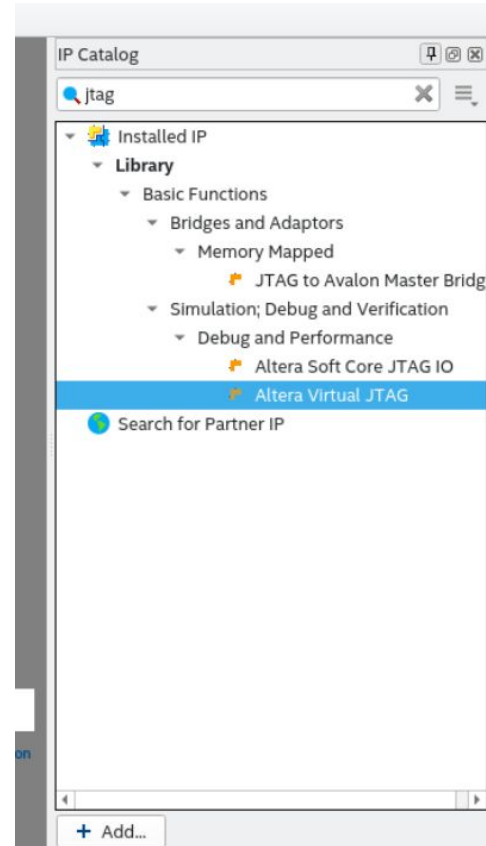
PASO 1

CREE UN
PROYECTO
NUEVO. EN
BLANCO



PASO 2

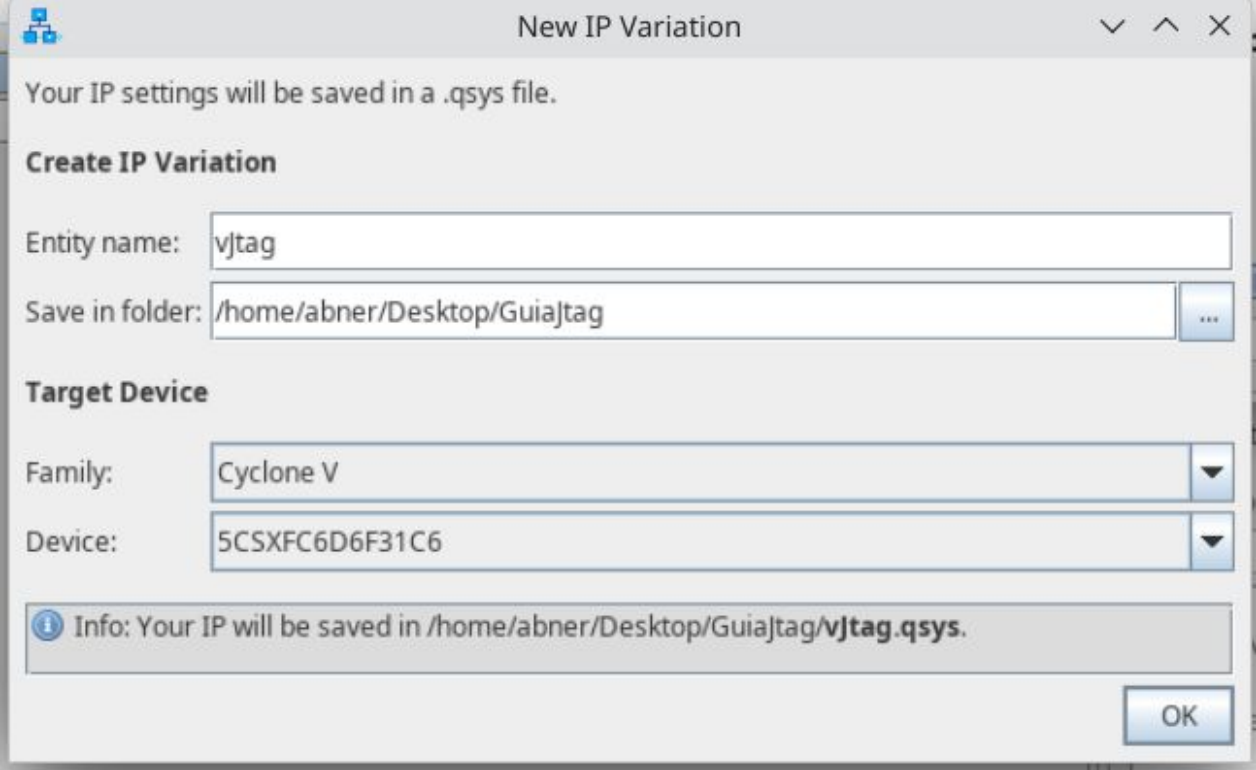
AGREGUE EL IP DE
**ALTERA VIRTUAL
JTAG**



PASO 3

DELE UN NOMBRE
A LA **ENTIDAD**.

EN ESTE CASO ES
vjtag



The screenshot shows a 'New IP Variation' dialog box. At the top, it states 'Your IP settings will be saved in a .qsys file.' Below this is a section titled 'Create IP Variation'. It contains two input fields: 'Entity name:' with the text 'vjtag' and 'Save in folder:' with the path '/home/abner/Desktop/GuiaJtag'. Below these is a section titled 'Target Device' with two dropdown menus: 'Family:' set to 'Cyclone V' and 'Device:' set to '5CSXFC6D6F31C6'. At the bottom, there is an information bar with an 'i' icon and the text 'Info: Your IP will be saved in /home/abner/Desktop/GuiaJtag/vjtag.qsys.' and an 'OK' button.

New IP Variation

Your IP settings will be saved in a .qsys file.

Create IP Variation

Entity name: vjtag

Save in folder: /home/abner/Desktop/GuiaJtag

Target Device

Family: Cyclone V

Device: 5CSXFC6D6F31C6

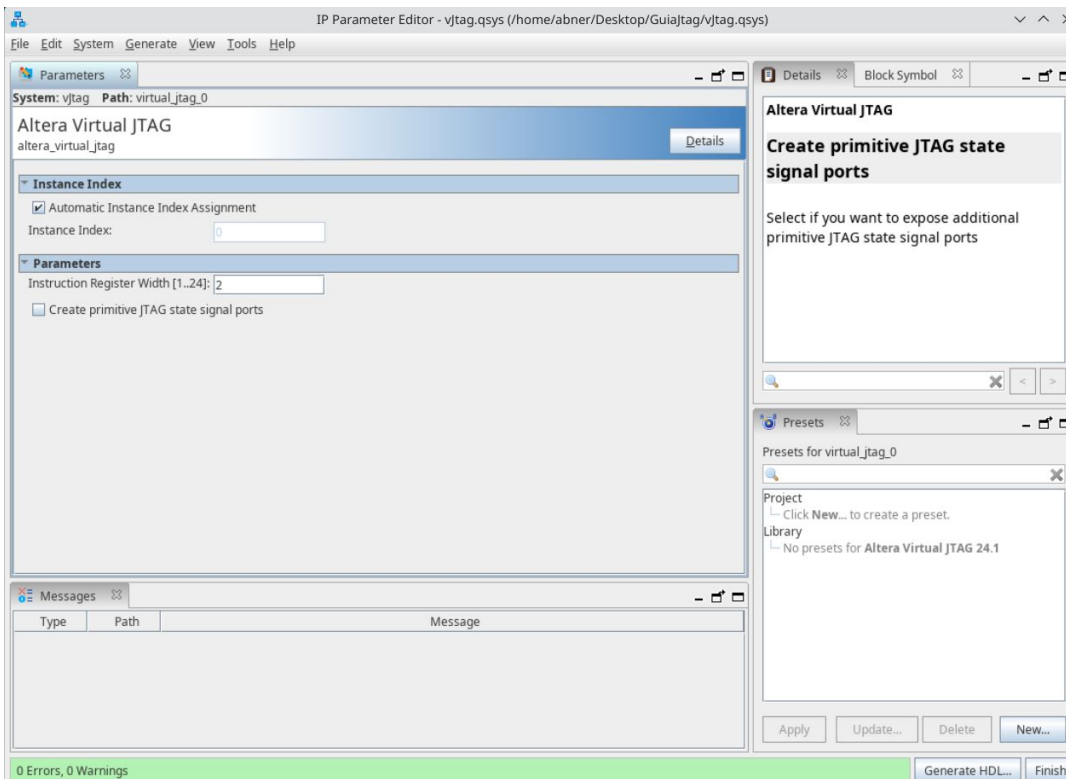
Info: Your IP will be saved in /home/abner/Desktop/GuiaJtag/vjtag.qsys.

OK

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**

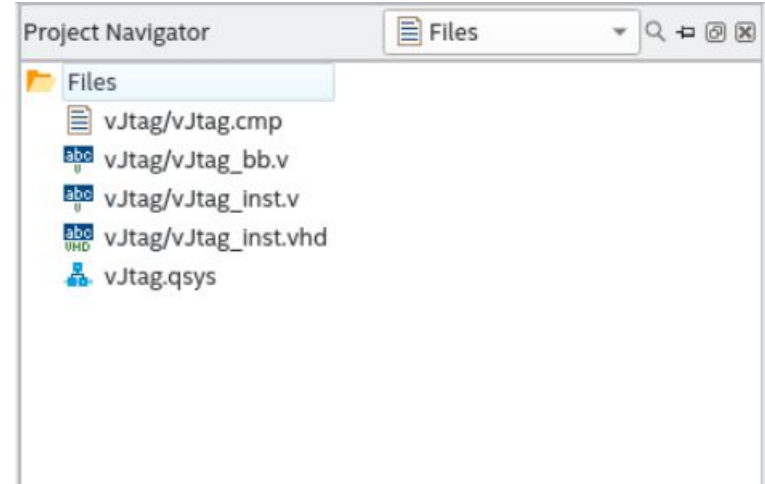
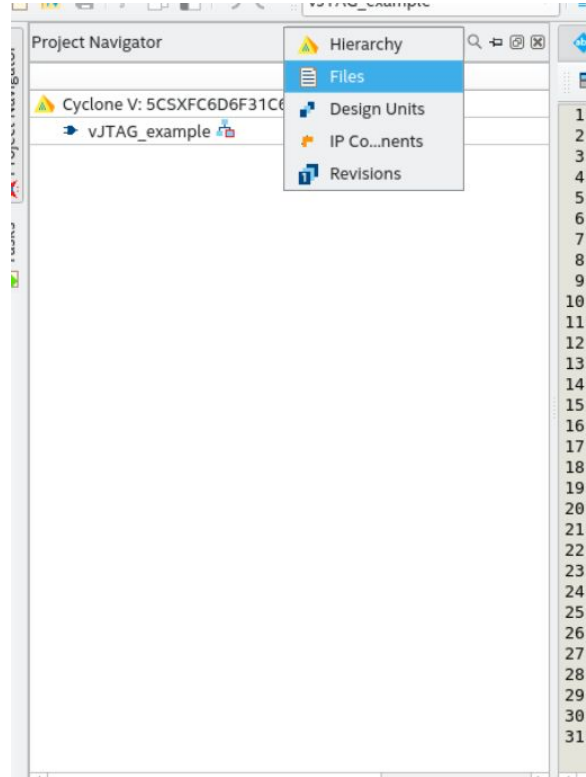
GENERE EL HDL



PASO 5

AGREGAR EL
ARCHIVO
TERMINADO EN
.qsys AL PROYECTO.

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

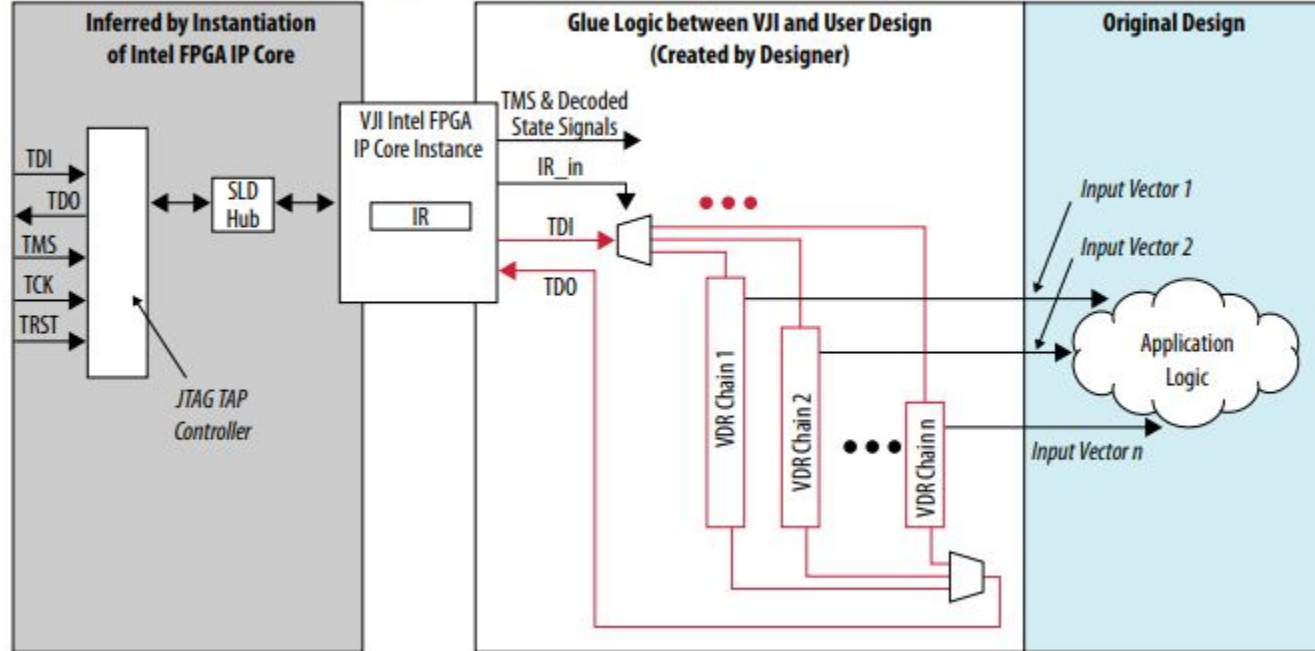


PASO 6

SE SIGUE ESTA
ARQUITECTURA.

UNA **INSTANCIA
DEL VIRTUAL
JTAG** CON UNA
**LÓGICA
INTERMEDIA**
QUE
COMUNIQUE EL
MÓDULO CON EL
PROYECTO

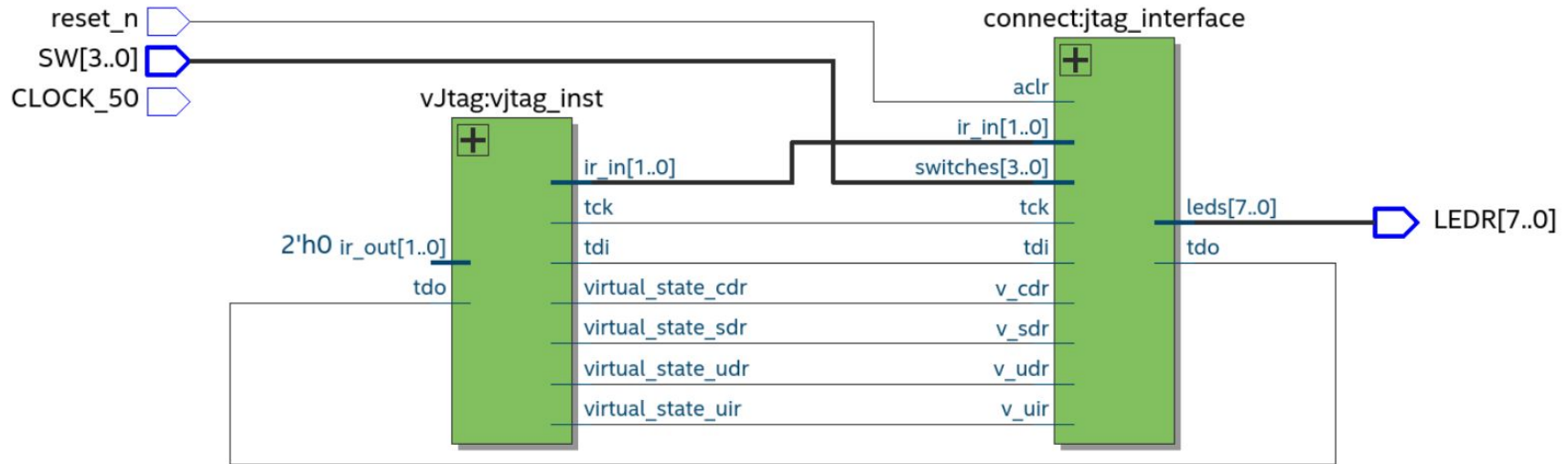
Block Diagram of a Design with a Single Virtual JTAG Instantiation



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 result
    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://cdrdv2-public.intel.com/781637/ug-683432-781637.pdf>

PASO 8

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

The screenshot displays the Intel Quartus II software interface, specifically the 'Tasks' window and the 'Program Device' section.

Tasks Window: This window shows the compilation process. The 'Compilation' dropdown is selected. The tasks are listed in a table with a 'Task' column and a 'Time' column. All tasks are marked with a green checkmark, indicating successful completion.

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: Below the tasks window, the 'Hardware Setup...' button is visible. The selected device is 'DE-SoC [1-3]'. The checkbox 'Enable real-time ISP to allow background programming when available' is unchecked.

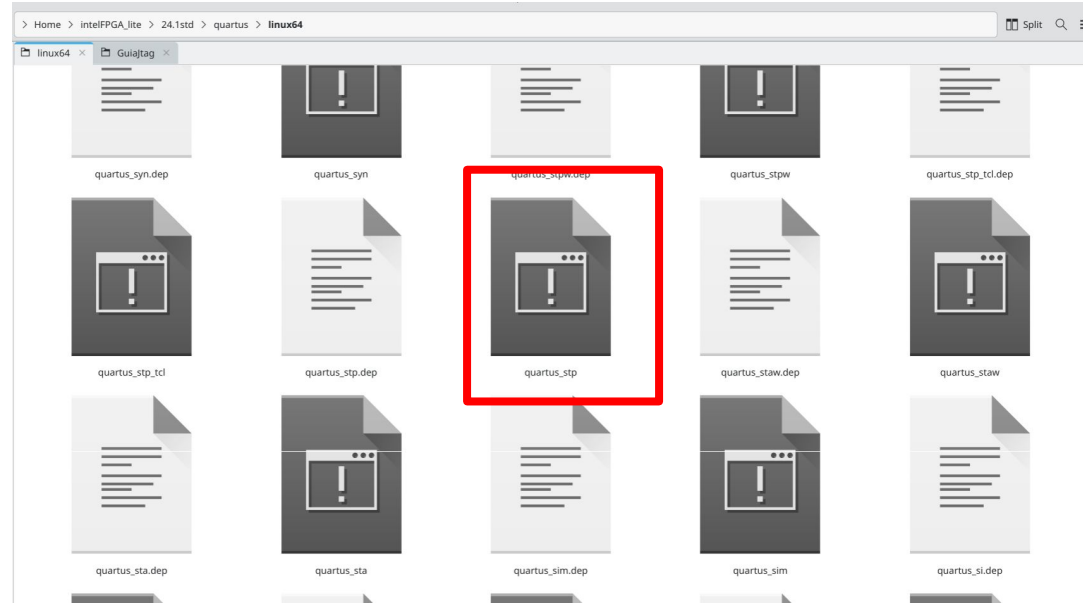
Program Device Section: This section contains a table with columns for File, Device, Checksum, Usercode, Program/Configure, Verify, Blank-Check, Examine, Security Bit, Erase, and Test. The first row shows the device 'SOCVHPS' with a checksum of '00000000' and a usercode of '<none>'. The second row shows the output file 'output_files/vJTAG_...' with a checksum of '00B204BC' and a usercode of '00B204BC'. The 'Program/Configure' checkbox is checked.

Left Panel: The left panel contains buttons for 'Start', 'Stop', 'Auto Detect', 'Delete', 'Add File...', 'Change File...', 'Save File', 'Add Device...', 'Up', and 'Down'.

Diagram: At the bottom, a diagram shows two Intel FPGAs. The first FPGA is labeled 'TDL' and has an arrow pointing to the second FPGA, which is labeled 'intel'.

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



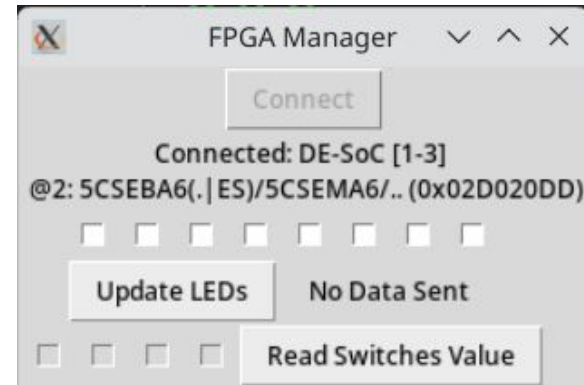
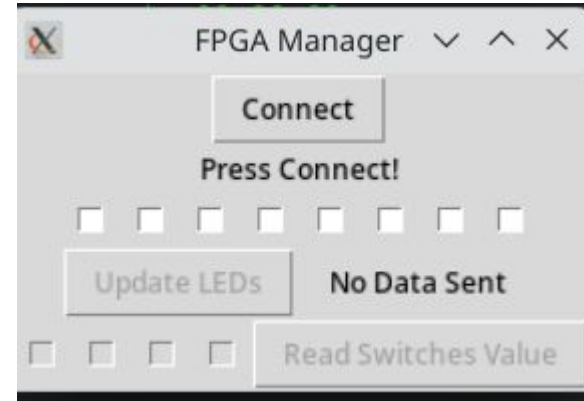
PASO 10

- 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**

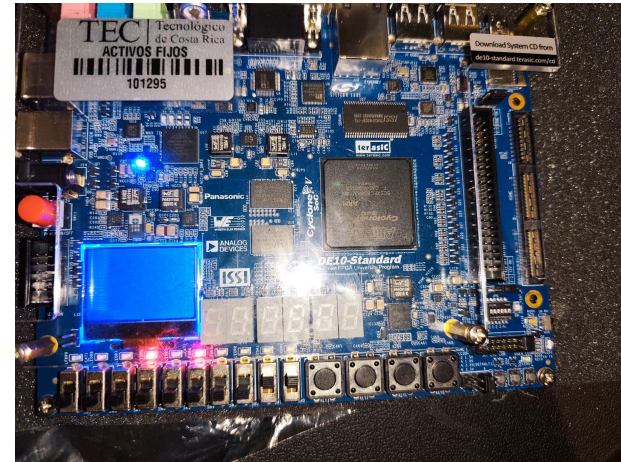
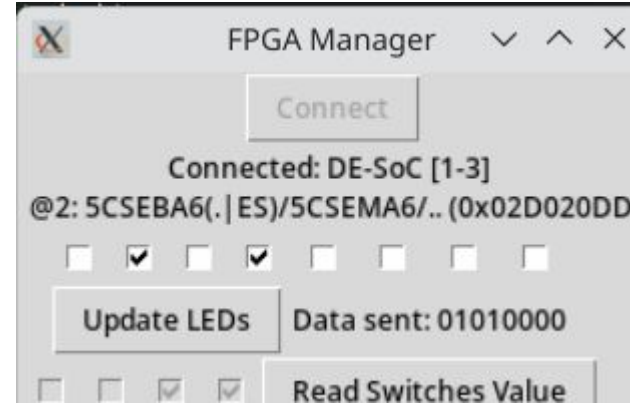
PASO 11

- 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/>