

Tutorial Paso a Paso: Conexión Remota con FPGA Xilinx

Zacchigna, F. G.

9 de abril de 2020

Resumen

En este tutorial se explican las distintas maneras en las que se puede acceder al servidor remoto de FPGAs, junto con las ventajas y desventajas de cada una. También se presenta una tabla que contiene los datos necesarios para la conexión exitosa con el servidor, como por ejemplos nombres de usuario y contraseñas, así también los números de series de los dispositivos.

Índice

1. Introducción	2
2. Instancia Local de Vivado y Target FPGA Remoto	3
2.1. Requisitos	3
2.2. Vivado	3
2.2.1. Pasos Tradicionales	3
2.2.2. Conectarse a la FPGA	3
2.3. SDK	5
2.3.1. Pasos tradicionales	5
2.3.2. Configurar la corrida y lanzar	5
2.4. Ver Salida de la Terminal Serie	6
3. Instancia Remota de Vivado via x2go	7
3.1. Requisitos	7
3.2. Vivado y SDK	7
3.2.1. Conexión por x2go	7
3.3. Ver Salida de la Terminal Serie	7
4. Instancia Remota de Vivado via SSH y X11-Forwarding	8
4.1. Requisitos	8
4.2. Vivado y SDK	8
4.2.1. Conexión por SSH con X11-Forwarding Habilitado	8
4.3. Ver Salida de la Terminal Serie	8
5. Puerto Serie entre Servidor Remoto y FPGA	9
5.1. Identificar el dispositivo FTDI	9
5.2. Conectar una terminal al dispositivo serie	9
6. Troubleshooting	11
6.1. Sin conexión al server	11
6.2. No aparece listada la placa	11
6.3. No me deja conectarme por SSH	11
7. ToDo (Pendientes)	12
Bibliography	12

1. Introducción

Completar la introducción explicando mejor la idea, ventajas y desventajas.

Los kits de FPGA se encuentran conectados a un servidor. El mismo se puede acceder de manera remota para utilizar los kits. En este tutorial se indican tres procedimientos distintos de como acceder de manera remota al servidor, programar la FPGA y ver la salida serie de la misma. Cada uno con distintos requisitos, ventajas y desventajas. Los tres posibles procedimientos son:

- Instancia Local de Vivado y Target FPGA Remoto - Sección 2 (Recomendado): Es la manera recomendada, principalmente porque requiere poco ancho de banda. Válido para usuarios de Windows y Linux.
- Instancia Remota de Vivado via x2go - Sección 3: Válido para usuarios de Windows y Linux.
- Instancia Remota de Vivado via SSH y X11-Forwarding - Sección 4: Válido para usuarios de Linux.

Luego de elegir y completar alguno de los procedimientos para conectarse a la FPGA del servidor remoto, en la mayoría de los casos, se debe continuar con el siguiente paso, que es conectarse con el puerto serie de la FPGA en el servidor remoto.

- Puerto Serie entre Servidor Remoto y FPGA - Sección 5: Válido para usuarios de Linux y cualquier de las tres alternativas anteriores.

Los kits disponibles en el servidor se pueden ver en la tabla 2, junto con otros datos necesarios para realizar la conexión remota, lo cuales se explicarán a lo largo del tutorial. Con motivo de agrupar todos los datos utilizados en este tutorial, incluimos la tabla 2 en esta sección, la cual contiene los datos del servidor remoto.

Kit	ZedBoard - Kit 0	ArtyZ7 - Kit 0	ArtyZ7 - Kit 1
Target Vend.	Digilent	Digilent	
Target SN	210248470637	003017A4C81CA	
FPGA Model	xc7z020-1	xc7z010-1	xc7z010-1
Usuario	zedboard-user00	artyz7-user00	artyz7-user01
Password	Preg. al docente	Preg. al docente	Preg. al docente
FTDI SN	021E814E101F	003017A4C81C	

Cuadro 1: Datos de los Kits conectados al Servidor Remoto.

Completar Tabla con datos para Artyz7-Kit1

Nombre del Host	Puerto
186.19.137.18	3121

Cuadro 2: Datos del Servidor Remoto.

2. Instancia Local de Vivado y Target FPGA Remoto

2.1. Requisitos

- Vivado instalado en la máquina local (Versión $\geq 201x.x$).
- Algún cliente SSH: En linux ya viene integrado, para Windows puede utilizarse *Putty* [1].

Completar
versión nece-
saria

2.2. Vivado

2.2.1. Pasos Tradicionales

Los pasos tradicionales, al trabajar de manera local, para programar la FPGA son:

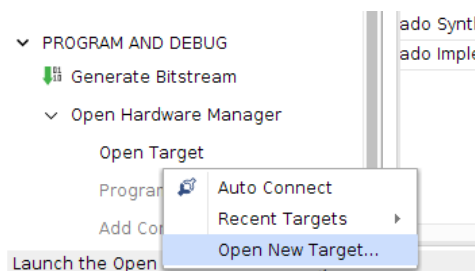
1. Abrir el Vivado.
2. Abrir el Proyecto.
3. Sintetizar, implementar y generar el bitstream (archivo `.bit`).
4. Una vez completado el paso anterior, conectarse a la FPGA del kit (*Open Target*).
5. Programar la FPGA con el bitstream.

Cada vez que se realiza una modificación en el diseño del Hardware se deben repetir los pasos 3 y 5.

El único paso que es distinto de lo tradicional al trabajar de manera remota es el paso 4, que debe realizarse una única vez en cada proyecto, y el mismo se describe a continuación.

2.2.2. Conectarse a la FPGA

Una vez se tenga el proyecto sintetizado, implementado, se procede a programar la FPGA. Para ello se debemos conectarnos con el servidor y luego con el kit de FPGA. Primero abrimos el menú correspondiente haciendo:



Aparecerá una nueva ventana y hacemos clic en *Next*. Luego debemos seleccionar del menú desplegable la opción de *Remote Server* en vez de *Local Server* que viene por defecto. A continuación se completan los datos del *Host name* y *Port*.

Open New Hardware Target	Open New Hardware Target
Hardware Server Settings Select local or remote hardware server, then configure the host name and port settings. Use Local server if the target is attached to the local	Hardware Server Settings Select local or remote hardware server, then configure the host name and port settings. Use Local server if the target is attached to the local
Connect to: Local server (target is on local machine) Local server (target is on local machine) Remote server (target is on remote machine)	Connect to: Remote server (target is on remote machine)
Click Next to launch and/or connect to the hw_server (port 3121) application on the local machine.	Remote Server Host name: 186.19.137.18 Port: 3121 [default is 3121] Click Next to launch and/or connect to the hw_server (port 3121) application on the remote machine '186.19.137.18'.
[?] < Back Next > Finish Cancel	[?] < Back Next > Finish Cancel

Los datos correctos para completar estos campos se obtienen de la tabla 2.

A continuación se mostrará el listado de targets (dispositivos JTAG) conectados al servidor y debajo aparece el listado dispositivos de cada target. El listado de dispositivos de cada target cambia al seleccionar distintos targets.

Open New Hardware Target

Select Hardware Target

Select a hardware target from the list of available targets, then set the appropriate JTAG clock (TCK) frequency. If you do not see the expected devices, decrease the frequency or select a different target.

Hardware Targets

Type	Name	JTAG Clock Frequency
xilinx_tcf	Digilent/210248470637	15000000
xilinx_tcf	Digilent/003017A4C81CA	15000000

Hardware Devices (for unknown devices, specify the Instruction Register (IR) length)

Name	ID Code	IR Le...
arm_dap_0	4BA00477	4
xc7z020_1	03727093	6

Hardware server: 186.19.137.18:3121

En este paso es importante seleccionar el target adecuado. En la tabla 1 se muestra el nombre del targets correspondiente a cada kit. Se debe seleccionar el target y hacer clic en *Next* y finalmente en *Finish*. Al cerrarse la venta se puede ver, en la solapa *Hardware*, el nombre del Host, el nombre del target seleccionado y el nombre del dispositivo (FPGA). Además, en la solapa *dashboard_1* de este ejemplo se muestra la temperatura de la FPGA conectada al servidor remoto provista por el XADC [2].

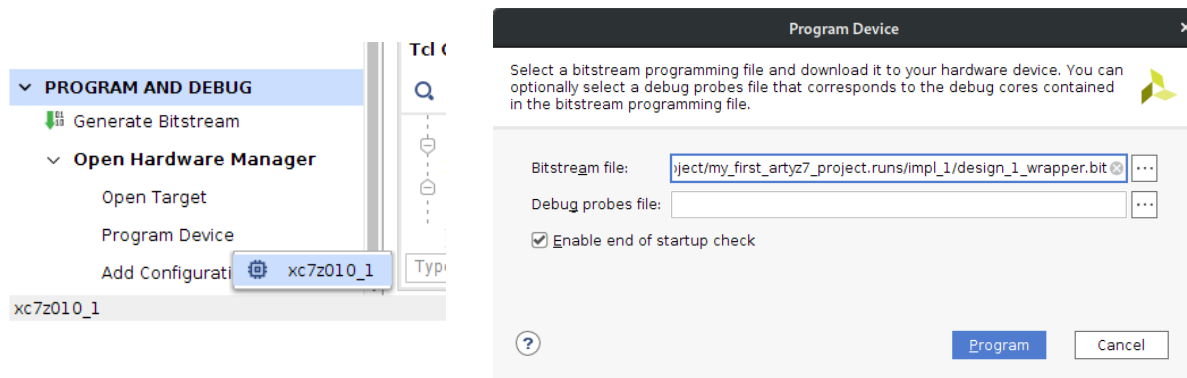
The screenshot shows the Vivado 2018.3 IDE interface. The top bar indicates the project path: `my_first_artyz7_project - [/home/fzacchigna/fpga/vivado_workspace/my_first_artyz7_project/my_first_artyz7_project.xpr] - Vivado 2018.3`. The main window is divided into several panes:

- HARDWARE MANAGER**: Shows a list of hardware components. The 'xc7z010_1' device is selected, with a status of 'Not program'. The 'XADC (System Monitor)' is also listed.
- Hardware Device Properties**: Displays details for the selected device 'xc7z010_1', including its name and properties.
- dashboard_1**: A window titled 'XADC (xc7z010_1)' showing a real-time temperature graph. The y-axis is 'Temperature (°C)' ranging from 32.5 to 42.5. The x-axis is 'Time (HH:MM:SS)' from 03:40:30 to 03:41:15. The graph shows a fluctuating temperature around 42°C.
- Tcl Console**: Displays the command history and output. The commands shown are:


```
current_hw_device [get_hw_devices xc7z010_1]
refresh_hw_device -update_hw_probes false [lindex [get_hw_devices xc7z010_1] 0]
INFO: [Labtools 27-1435] Device xc7z010 (JTAG device index = 1) is not programmed (DONE status = 0).
refresh_hw_device: Time (s): cpu = 00:00:09 ; elapsed = 00:00:07 ; Memory (MB): peak = 7443.145 ; gain = 29.410 ; free phys
```

El próximo paso depende del tipo de proyecto que se tenga. Si se desea lanzar el *Software Development Kit* (SDK) para descargar un nuevo programa en la memoria del procesador integrado, se debe continuar en la sección 2.3. Si se quiere programar la FPGA directamente con el bitstream. puede que se quiera programar la FPGA con el bitstream o Se procede de la manera habitual, haciendo clic en *Program Device* y luego sobre el nombre del

dispositivo. En caso de que haya un solo dispositivo, no será necesario el último clic. Se abre una nueva ventana y unos segundos después de hacer clic en *Program* termina de descargarse el archivo de configuración .bit en la FPGA:



2.3. SDK

2.3.1. Pasos tradicionales

Los pasos tradicionales, al trabajar de manera local, para ejecutar un programa en la plataforma de Hardware diseñada son:

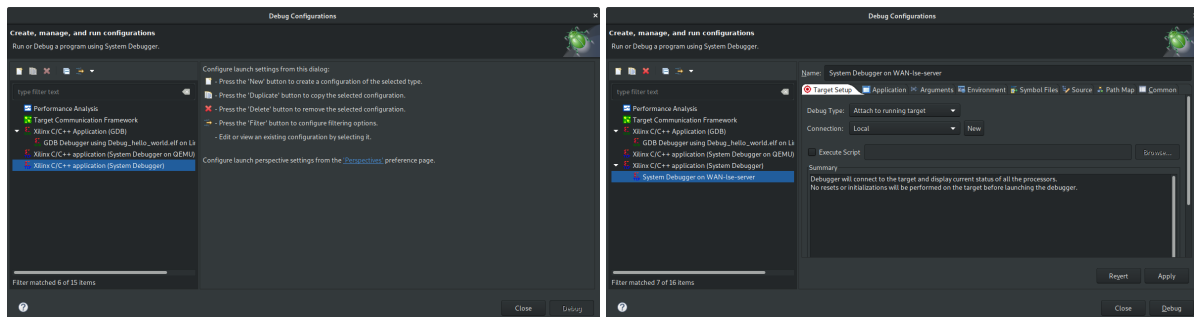
1. Exportar el Hardware desde Vivado incluyendo el bitstream.
2. Abrir el SDK.
3. Crear el *Board Support Package* (BSP).
4. Crear el Proyecto, por ejemplo, *Hello World*.
5. Configurar la corrida en modo *Run* o *Debug*.
6. Descargar el programa a la FPGA y ejecutarlo.

Para nuevos proyectos de Software sobre misma plataforma de Hardware solo se deben repetir los pasos 4 y 6. El único paso que es distinto de lo tradicional al trabajar de manera remota es el paso 5, el cual debe ejecutarse una única vez, por cada proyecto de Software nuevo.

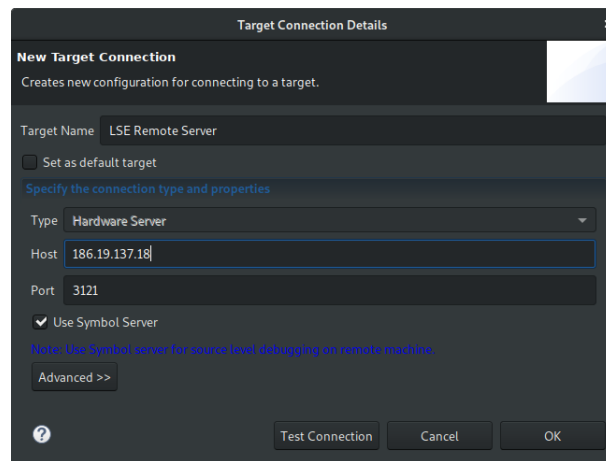
2.3.2. Configurar la corrida y lanzar

Una vez exportado el Hardware (incluyendo el bitstream) y con un proyecto de Software creado y listo para descargar a la FPGA se procede de la siguiente manera.

Seleccionar la aplicación para debugger e ir a la configuración de Debug o de Corrida (*Run Configurations...* o *Debug Configurations...*) Crear una nueva configuración, se recomienda la opción *System Debugger* y no *GDB*, ya que la última perderá soporte. Para crearla clic derecho en *Xilinx C/C++ application (System Debugger)* y *New*.



En la pestaña de *Target* hay un botón para generar una nueva conexión. Generar una nueva conexión y completar los datos: Nombre, *Host* y *Port*. El nombre puede ser cualquiera que sea representativo. Los datos del *Host* y *Port* salen de la tabla 2. Seleccionar la opción de *Use Symbol Server*.



Clic en *OK*. De vuelta en la ventana de *Debug/Run Configuration* asegurarse de seleccionar la nueva configuración remota recién creada. Clic en *Debug*.

En caso de que informe que la FPGA no esté configurada, cancelar el *Debug/Run* y configurar la FPGA haciendo clic en *Xilinx* y *Program FPGA*. En la ventana que aparece, asegurarse de elegir la conexión recientemente creada. La conexión que creamos ya va a quedar disponible para futuras configuraciones, inclusive si cambia el proyecto de Software (no el de Hardware de Vivado).

2.4. Ver Salida de la Terminal Serie

El primer paso es conectarse por SSH al servidor. Para ello diferenciamos dos casos, Linux y Windows:

- **Linux:** Abrir una terminal y poner el comando:

```
$ ssh artyz7-user00@186.19.137.18
```

y colocamos la contraseña. El host, nombre de usuario `artyz7-user00` y la correspondiente contraseña deben ser reemplazados por los valores que aparecen en la tabla 1 según el kit que se desee utilizar. La consola debe mostrar algo similar a:

```
[user@host ~]$ ssh artyz7-user00@186.19.137.18
artyz7-user00@186.19.137.18's password:
Last login: Wed Apr  8 01:22:49 2020 from 190.55.150.138
[artyz7-user00@lse-server-pc ~]$
```

- **Windows:** _____

Una vez conectado al servidor por SSH, ya sea en Linux o Windows, se debe continuar con la sección 5, en donde se muestra como identificar el dispositivo serie de la FPGA para luego conectarse al mismo.

Completar con ejemplo de Putty de Windows

3. Instancia Remota de Vivado via x2go

Completar Instancia Remota via x2go

3.1. Requisitos

3.2. Vivado y SDK

3.2.1. Conexión por x2go

3.3. Ver Salida de la Terminal Serie

Indicar saltar a Puerto Serie

4. Instancia Remota de Vivado via SSH y X11-Forwarding

Completar Instancia Remota via SSH

4.1. Requisitos

4.2. Vivado y SDK

4.2.1. Conexión por SSH con X11-Forwarding Habilitado

4.3. Ver Salida de la Terminal Serie

Indicar saltar a Puerto Serie

5. Puerto Serie entre Servidor Remoto y FPGA

En este paso asumimos que ya estamos conectados al Servidor Remoto ya sea por SSH o x2go. Hay dos pasos para poder visualizar la salida del puerto serie de la FPGA.

- Identificar el dispositivo FTDI de la FPGA.
- Conectar una terminal al dispositivo FTDI de la FPGA.

5.1. Identificar el dispositivo FTDI

Primero, mostramos brevemente el problema. Si listamos los dispositivos seriales (por USB) conectados al servidor remoto vamos a ver que aparecen varios y no es posible determinar cual es el correcto a simple vista. algo similar a lo siguiente:

```
[artyz7-user00@lse-server-pc ~]$ ls -la /dev/ttyUSB* /dev/ttyACM*
crw-rw----. 1 root dialout 188, 0 Apr  8 05:09 /dev/ttyUSB0
crw-rw----. 1 root dialout 188, 1 Apr  8 05:09 /dev/ttyUSB1
crw-rw----. 1 root dialout 188, 2 Apr  8 05:13 /dev/ttyUSB2
crw-rw----. 1 root dialout 166, 0 Apr  8 05:13 /dev/ttyACM0
```

Para identificarlo tenemos que tipear el comando

```
$ dmesg | grep -A 10 003017A4C81C
```

El número de serie, que aparece en negrita, debe ser reemplazado según el kit que corresponda (ver tabla 1). En caso de trabajar con una placa Arty se verá un mensaje similar al siguiente:

```
[artyz7-user00@lse-server-pc ~]$ dmesg | grep -A 10 003017A4C81C
...
[13793.840852] usb 1-13: SerialNumber: 003017A4C81C
[13793.845603] ftdi_sio 1-13:1.0: FTDI USB Serial Device converter detected
[13793.845651] usb 1-13: Detected FT2232H
[13793.845891] usb 1-13: FTDI USB Serial Device converter now attached to tttyUSB0
[13793.848590] ftdi_sio 1-13:1.1: FTDI USB Serial Device converter detected
[13793.848647] usb 1-13: Detected FT2232H
[13793.848937] usb 1-13: FTDI USB Serial Device converter now attached to tttyUSB1
...
```

y en caso de trabajar con un kit ZedBoard se verá un mensaje similar al siguiente:

```
[artyz7-user00@lse-server-pc ~]$ dmesg | grep -A 10 021E814E101F
...
[14022.160506] usb 1-10: SerialNumber: 021E814E101F
[14022.162953] cdc_acm 1-10:1.0: tttyACM0: USB ACM device
...
```

En ambos se puede ver el número de serie y **tttyUSB#**. o el número de serie y **tttyACM#**. En cualquiera de los dos casos, lo que debemos hacer es identificar el número que acompaña a **tttyUSB** o **tttyACM**. Para el caso de la Arty aparecen dos, nos interesa el segundo, en el ejemplo el **tttyUSB1**. Para el caso de la ZedBoard **tttyACM0**. Con este paso finaliza la identificación del dispositivo.

5.2. Conectar una terminal al dispositivo serie

Una vez identificado el dispositivo serie, procedemos a abrir el puerto serie conectándolo a una terminal. Para ello utilizamos, estando conectado al servidor remoto, una aplicación de comunicación de puerto serie. En este caso mostramos el ejemplo con **minicom**. Mediante el comando

```
$ minicom -D /dev/tttyUSB1 -b 115200
```

abrimos el dispositivo serie **/dev/tttyUSB1** con una velocidad de baudios de 115200 (esta velocidad se debe cambiar según la necesidad del usuario). Notar que utilizamos el nombre del dispositivo que acabamos de identificar en la sección anterior. En el programa de ejemplo que estamos corriendo en el procesador integrado a la FPGA podemos ver la siguiente salida:

```
Welcome to minicom 2.7.1

OPTIONS: I18n
Compiled on Jul 25 2019, 00:00:00.
Port /dev/ttyUSB1, 05:09:13

Press CTRL-A Z for help on special keys

Hello World 0!!
Hello World 1!!
Hello World 2!!
Hello World 3!!
Hello World 4!!
Hello World 5!!
Hello World 6!!
Hello World 7!!
```

Antes de terminar esta sección mostramos dos comandos de `minicom` de suma utilidad:

- `Ctrl-a` y luego `x`: Para salir de `minicom`.
- `Ctrl-a` y luego `z`: Para listar todos los comandos de `minicom`.

6. Troubleshooting

Completar el trouble shooting.

- 6.1. Sin conexión al server**
- 6.2. No aparece listada la placa**
- 6.3. No me deja conectarme por SSH**

7. ToDo (Pendientes)

Todo list

Completar la introducción explicando mejor la idea, ventajas y desventajas.	2
Completar Tabla con datos para Artyz7-Kit1	2
Completar versión necesaria	3
Completar con ejemplo de Putty de Windows	6
Completar Instancia Remota via x2go	7
Indicar saltar a Puerto Serie	7
Completar Instancia Remota via SSH	8
Indicar saltar a Puerto Serie	8
Completar el trouble shooting.	11

Referencias

- [1] Putty.org. <https://www.putty.org/>.
- [2] Xilinx System Monitor and XADC. <https://www.xilinx.com/products/technology/analog-mixed-signal.html>.