

Cómo acceder a las cadenas de JTAG en Vivado

Creador: David Rubio G.

Entrada: <https://soceame.wordpress.com/2024/07/22/como-acceder-a-las-cadenas-de-jtag-en-vivado/>

Blog: <https://soceame.wordpress.com/>

GitHub: <https://github.com/DRubioG>

Fecha última modificación: 23/02/2025

NOTA: no domino mucho sobre este tema porque no lo he manejado nunca, aún así voy a tratar de explicar cómo acceder a ella y cómo utilizarla

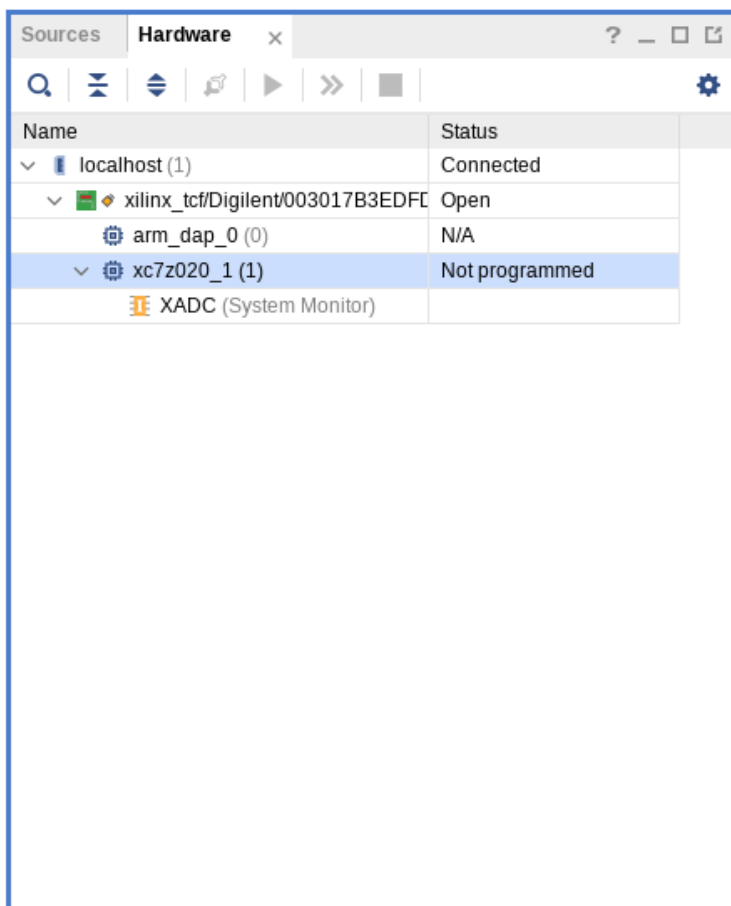
Las cadena de JTAG son un sistema que tienen los dispositivos que se comunican por JTAG que permite programar varios dispositivos en línea, para ello tienen que estar acoplados a las líneas de JTAG. Este sistema también permite realizar **Boundary Scan**, aplicado a dispositivos de complejo acceso, como los que están encapsulados en BGA, de poder comprobar las conexiones eléctricas verificar que está bien soldado a la placa. Para ello lo que se hace es mover una señal discreta por un pin y comprobar por dónde se lee esa discreta.

Nota: hacer **Boundary Scan** es bastante caro, no hay mucha documentación al respecto y es complejo de aplicar porque necesitas varios dispositivos que sean compatibles con JTAG.

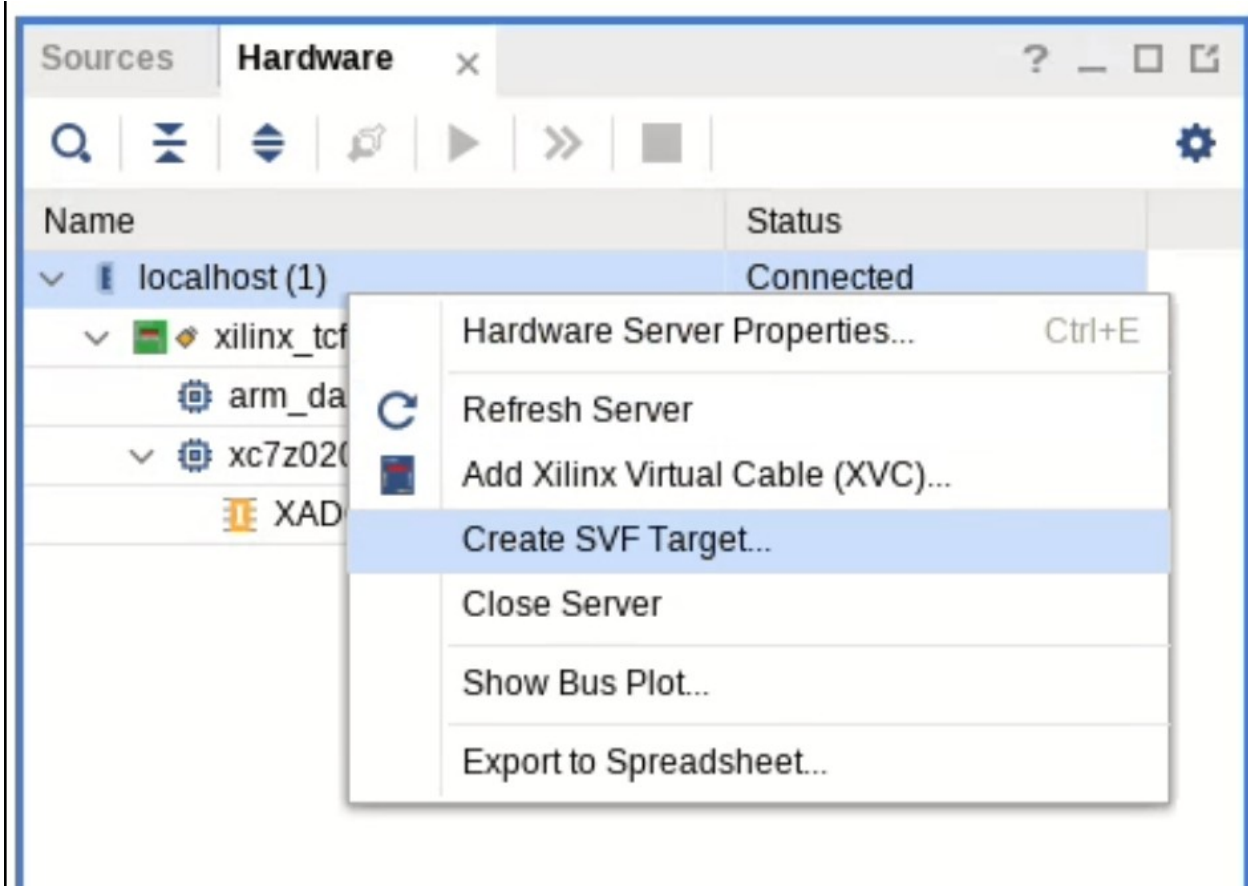
Para ello Vivado permite crear unos ficheros llamados VSF que son los que facilitan el manejo de estas cadenas de JTAG.

Acceder al sistema

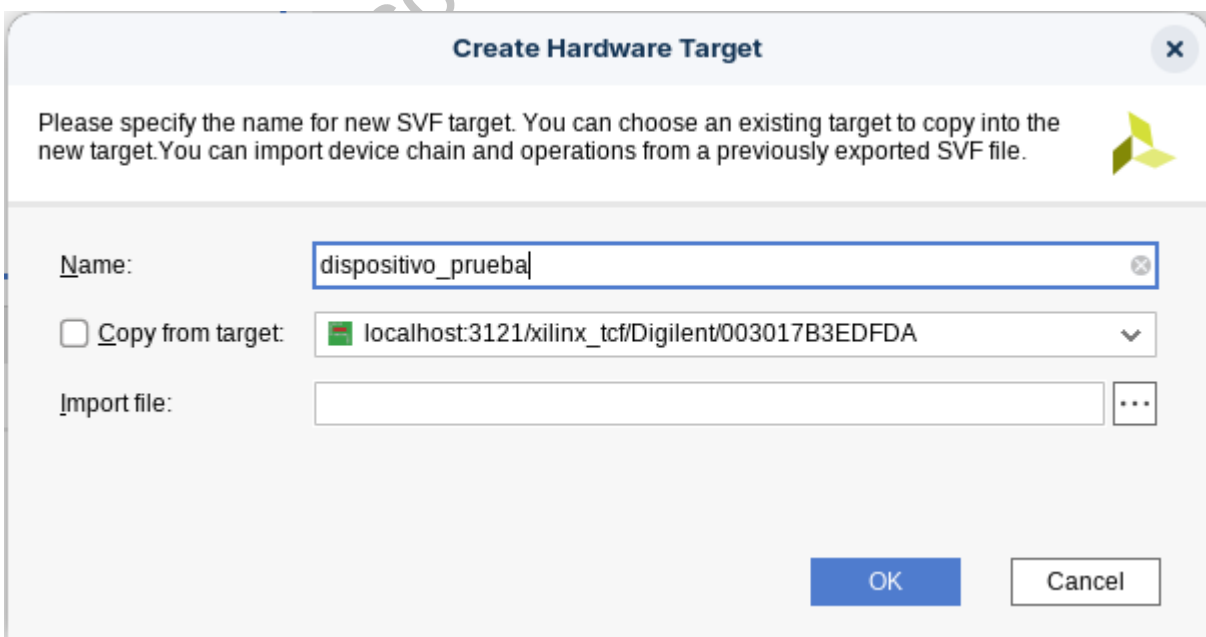
Para acceder a ellos, primero acoplamos una placa a Vivado, como si de un dispositivo normal fuese.



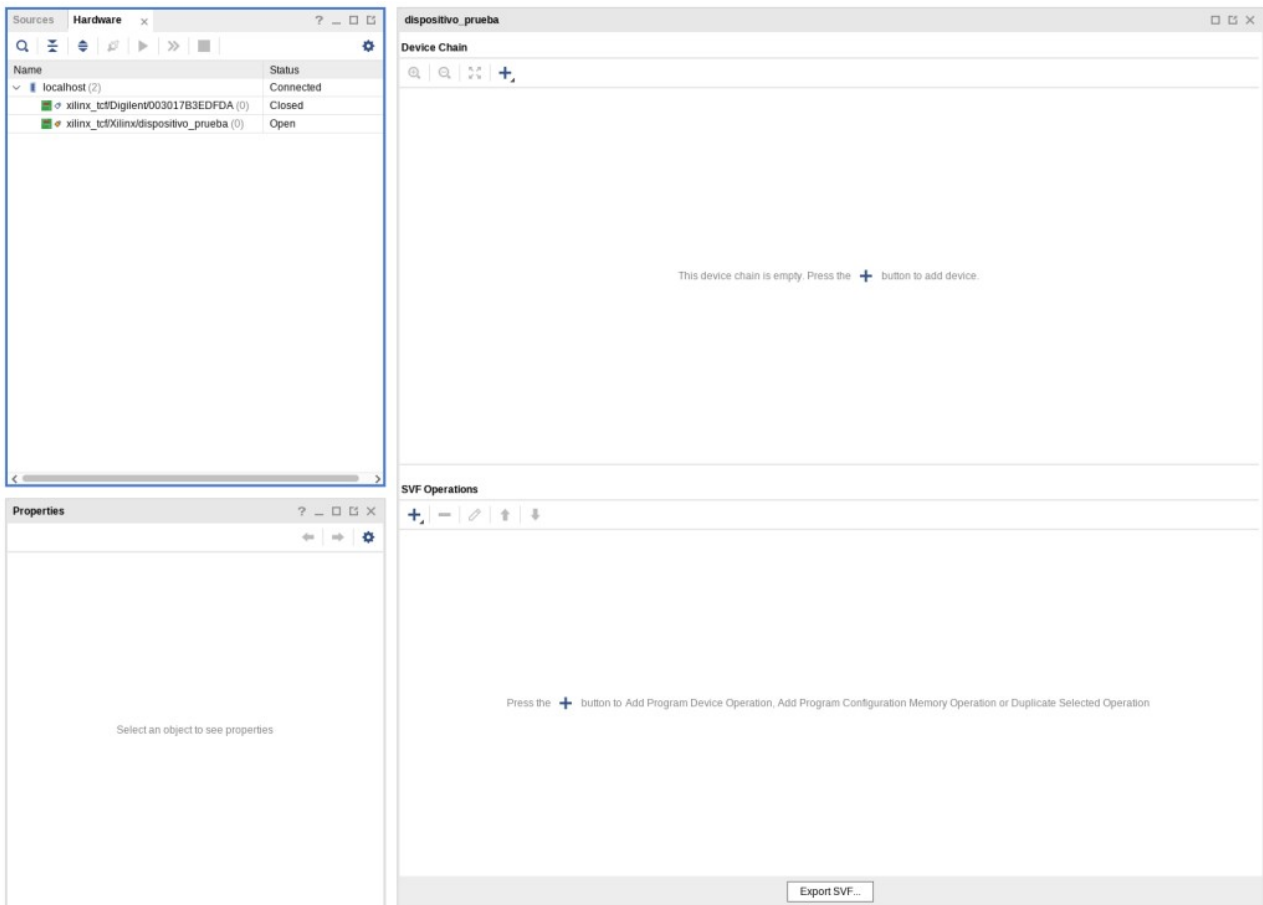
En esta pestaña aparece la configuración normal de programación de FPGAs. Ahora le damos clic derecho en *localhost* y nos aparece la opción de **Create SVF Target**.



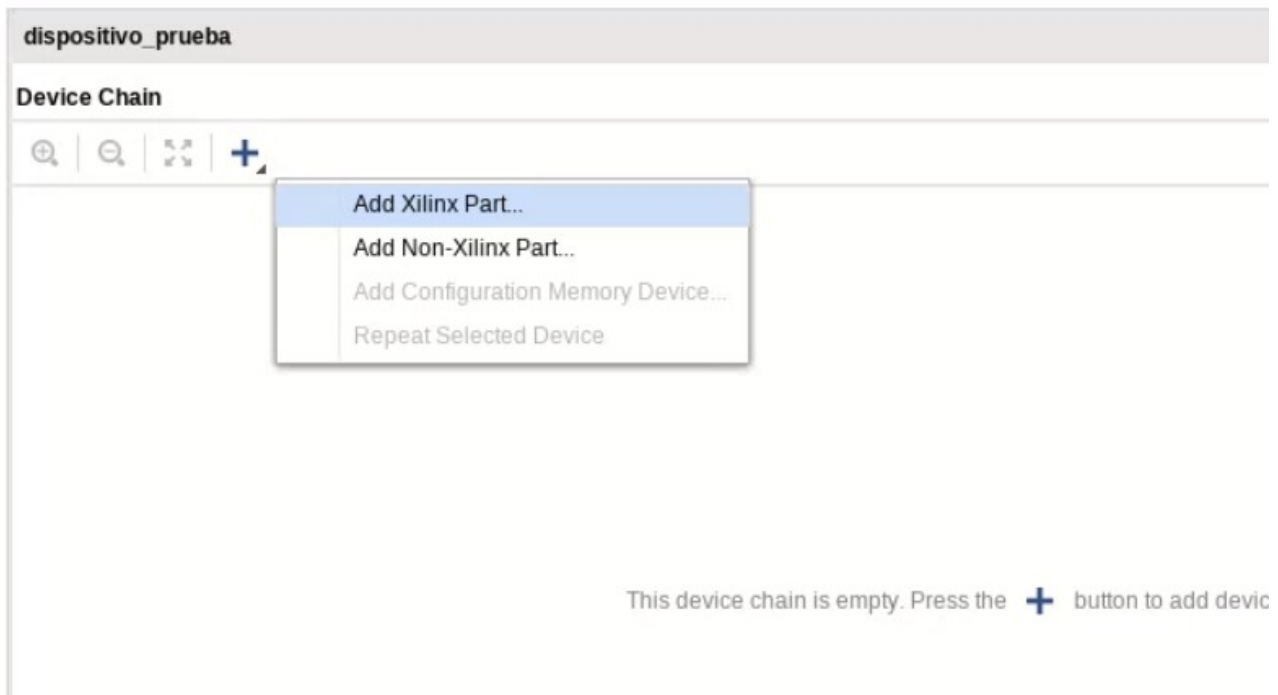
Se nos abre una pestaña en la que le podemos dar un nombre al SVF, también podemos elegir el dispositivo de programación, o importar un SVF previo que ya tengamos.



Se nos abre una pestaña en la que aparece la cadena de JTAG y los SVF que tenemos creados.



Para añadir un nuevo dispositivo tenemos varias opciones, añadir un componente de Xilinx(FPGA o SoC) o una parte externa de Vivado. Además, permite repetir lo que tengamos creado en el mismo esquema o añadir un fichero MCS.

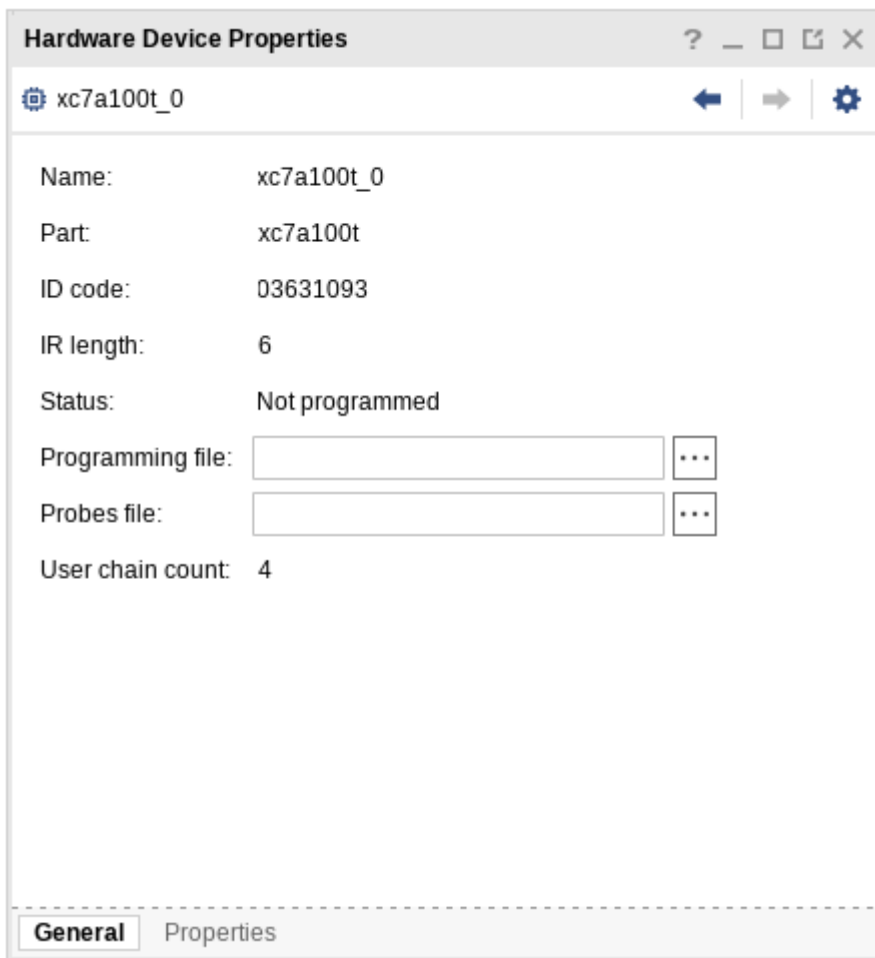


Ejemplos de componentes

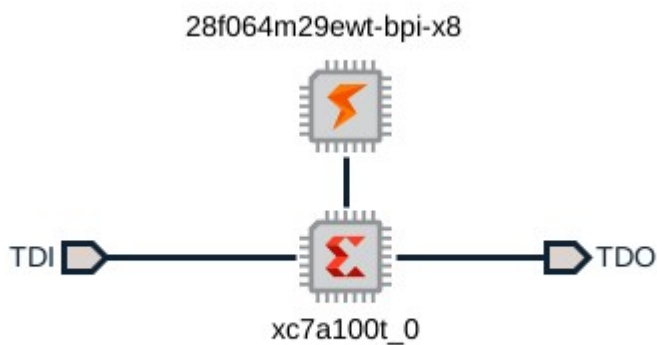
- Si añadimos una FPGA de Xilinx



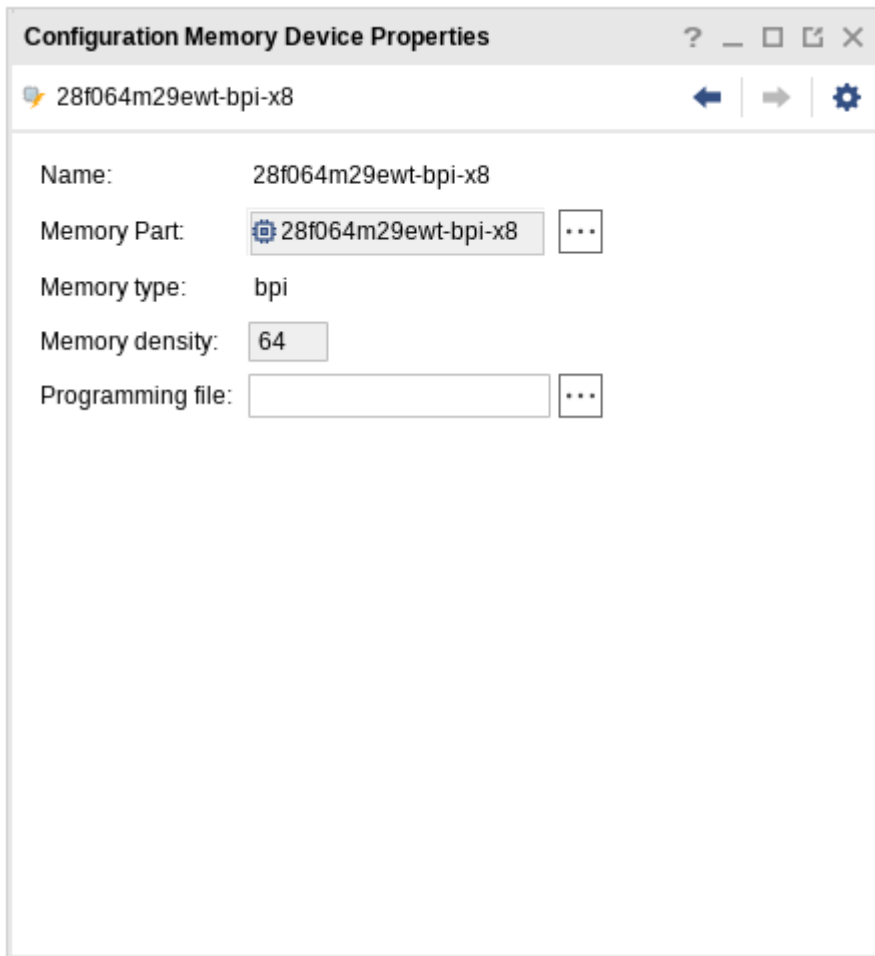
Para añadirle un bitstream se hace en **SVF Operations**, (también parece que se podría por el **Hardware Device Properties** (también permite meter un ILA))



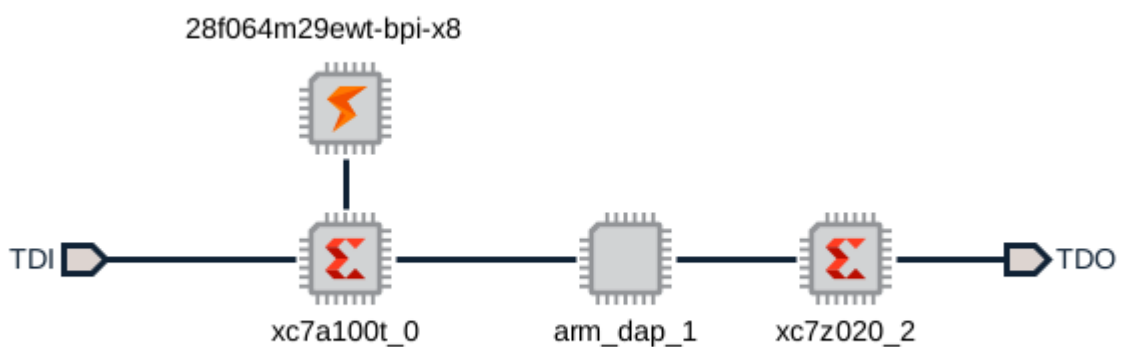
- Si a la FPGA le añadimos una configuración de memoria con un MCS



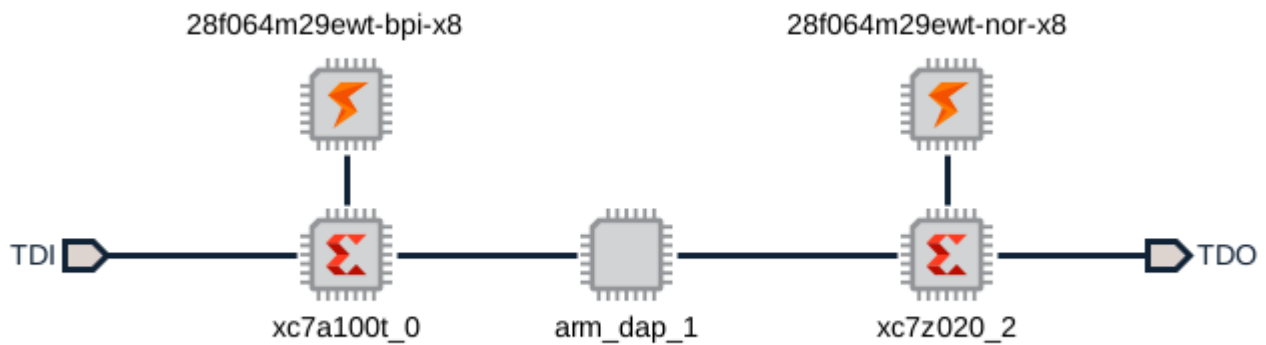
Para programar el MCS de la configuración de memoria, en **Properties**



- Si añadimos un SoC nos añade las dos partes programables del SoC



La parte del ARM no se puede grabar, la de la FPGA sí, y además se le puede añadir una memoria.



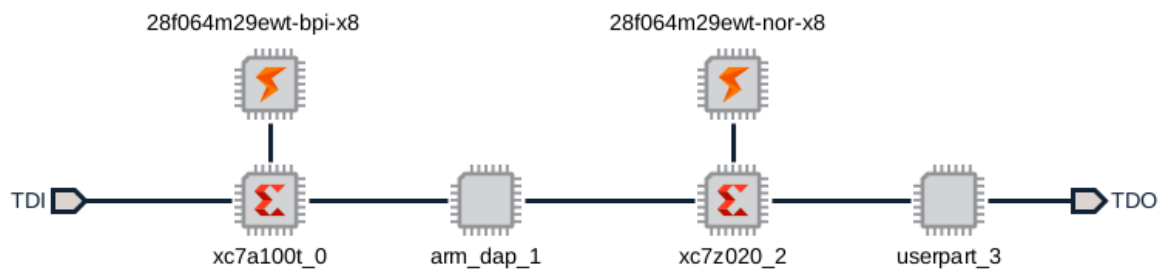
- Para añadir una parte externa a Xilinx (**Add Non-Xilinx Device**), te solicita que rellenes los campos. (a modo de ejemplo se puede rellenar con: ID code -> 000000A5 ; IR length -> 16 ; Mask -> ffff)

Dialog box titled "Add Non-Xilinx Device" showing fields for adding a device to the JTAG chain:

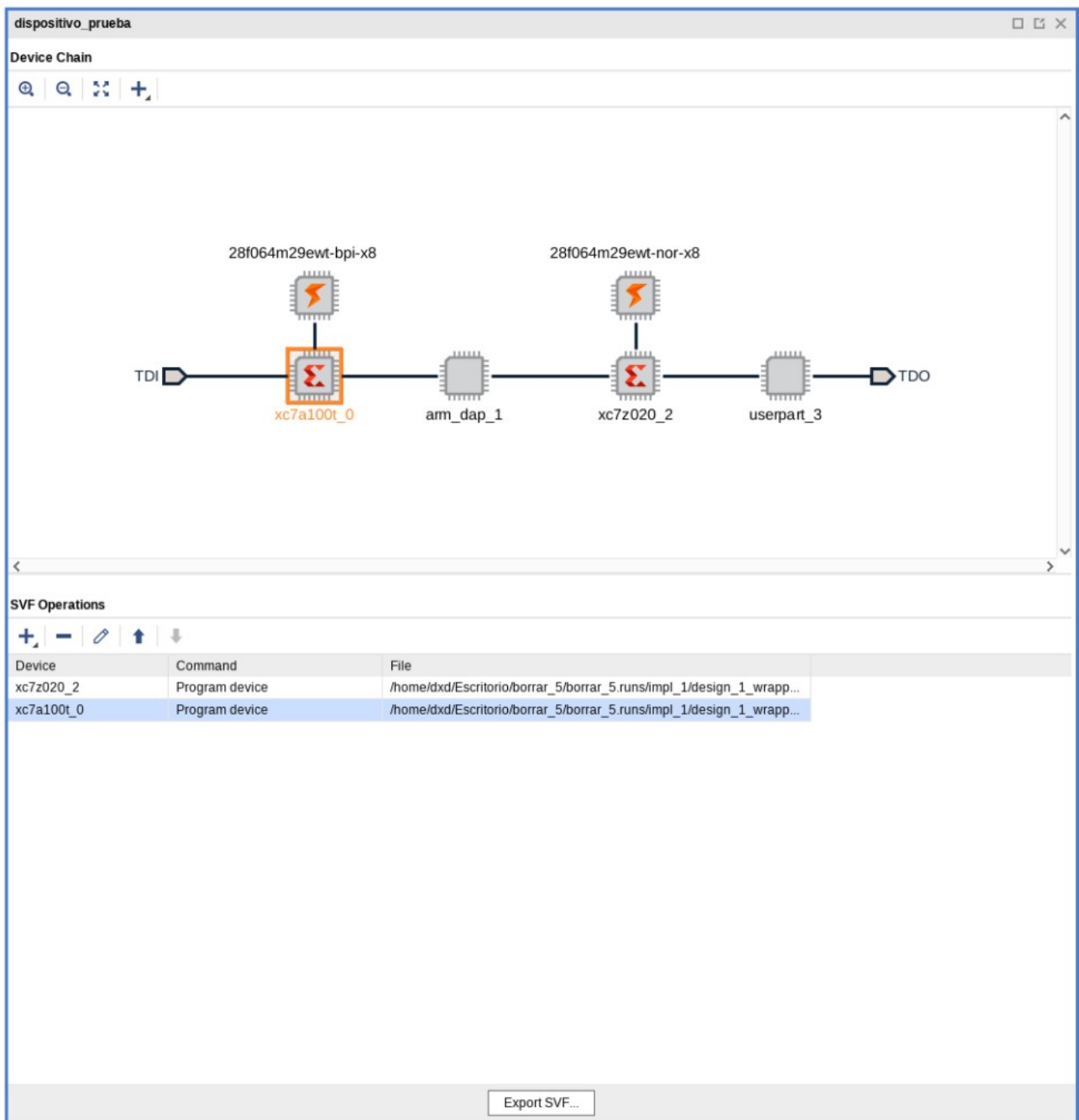
- Part Name: userpart
- ID code (Hex):
- IR Length:
- Mask (Hex):

Buttons: OK, Cancel

Y después te la añade a la cadena.



Una vez ya esté toda la cadena preparada, en la opción **SVF Operations** aparecen todos los ficheros .bit y .mcs que se han utilizado.



Después le das a **Export SVF** y te genera el fichero SVF que se puede programar.

Programación

Para programarlo (según fabricante, aunque no he conseguido programar un fichero de ejemplo) se tiene que usar el comando **execute_hw_svf**

```
execute_hw_svf <fichero SVF>.svf
```

Si se consigue ejecutar este comando se logra programar la cadena de JTAG.

Documentación

Para más información en el documento de Vivado UG908

- <https://docs.amd.com/viewer/book-attachment/mV2tIfp565YdTxAxpXOEKQ/lY0cn8t8VMfFN2splL0zCA>

<https://soceame.wordpress.com/>