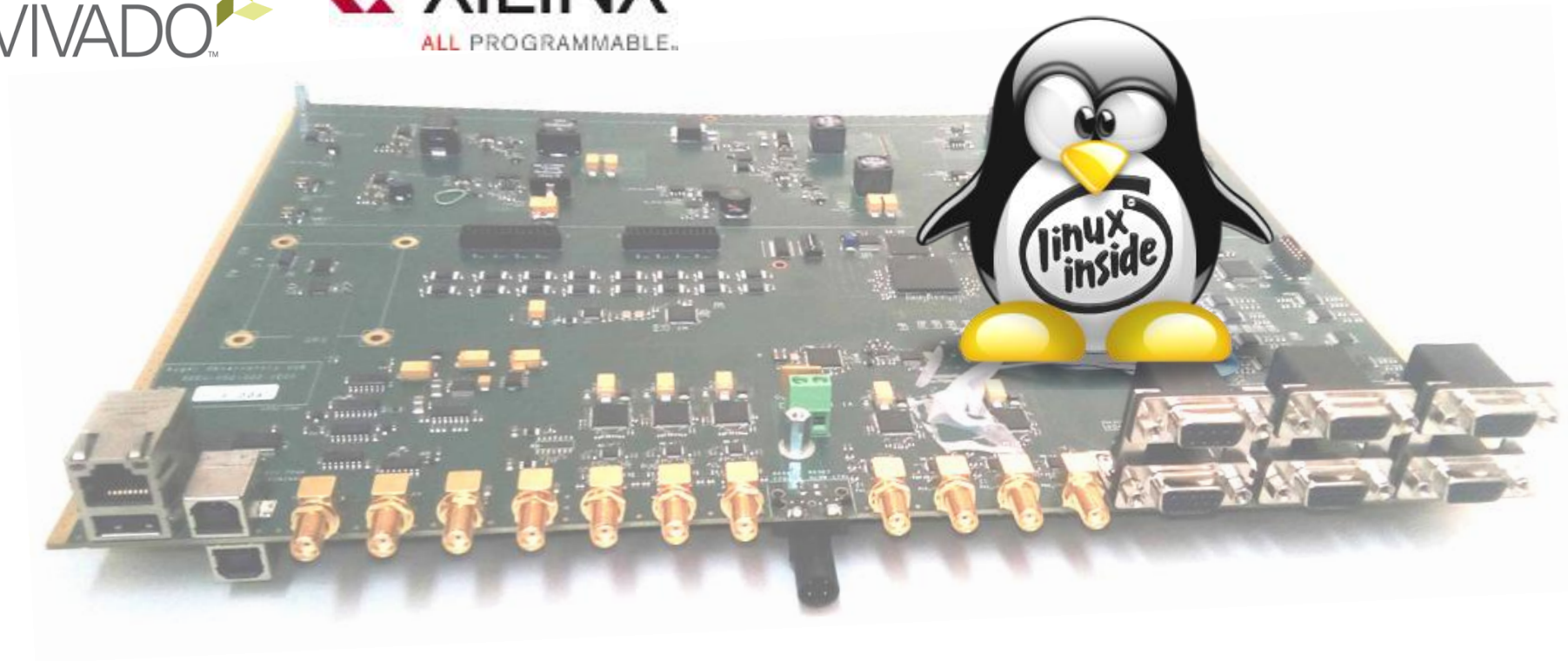
WP1- Documentation

Booting Petalinux from QSPI on UUB



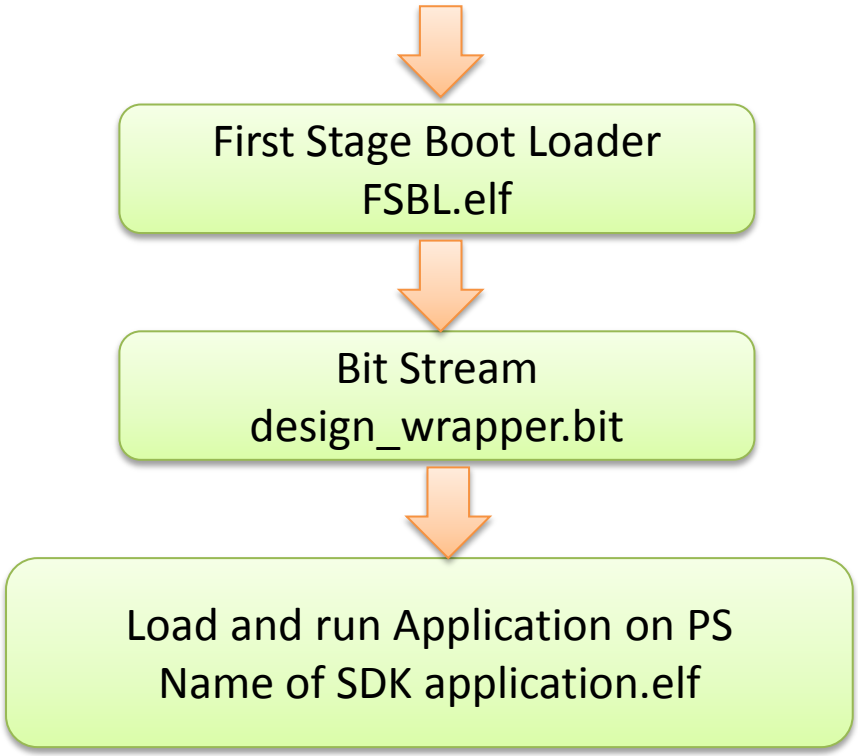
R. Assiro

Create Boot image for Zynq 7020 on UUB architecture

The Zynq boot process begins with running code inside the Boot ROM.
The boot ROM manages the early boot process by selecting the boot medium and quickly loading **the First Stage Boot Loader (FSBL).**
The FSBL does important early system initialization, configuring the DDR
The FSBL is created by Xilinx tools (SDK) using information from your hardware project or by **Petalinux build process.**

The FSBL is the stepping point between Xilinx's code and your code, or petalinux built. What happens next depends on what type of software your system needs. If your device runs Linux it's very likely that your next step is loading U-boot or if your device runs a stand alone applications your next step is your .elf file..

Running a stand alone application

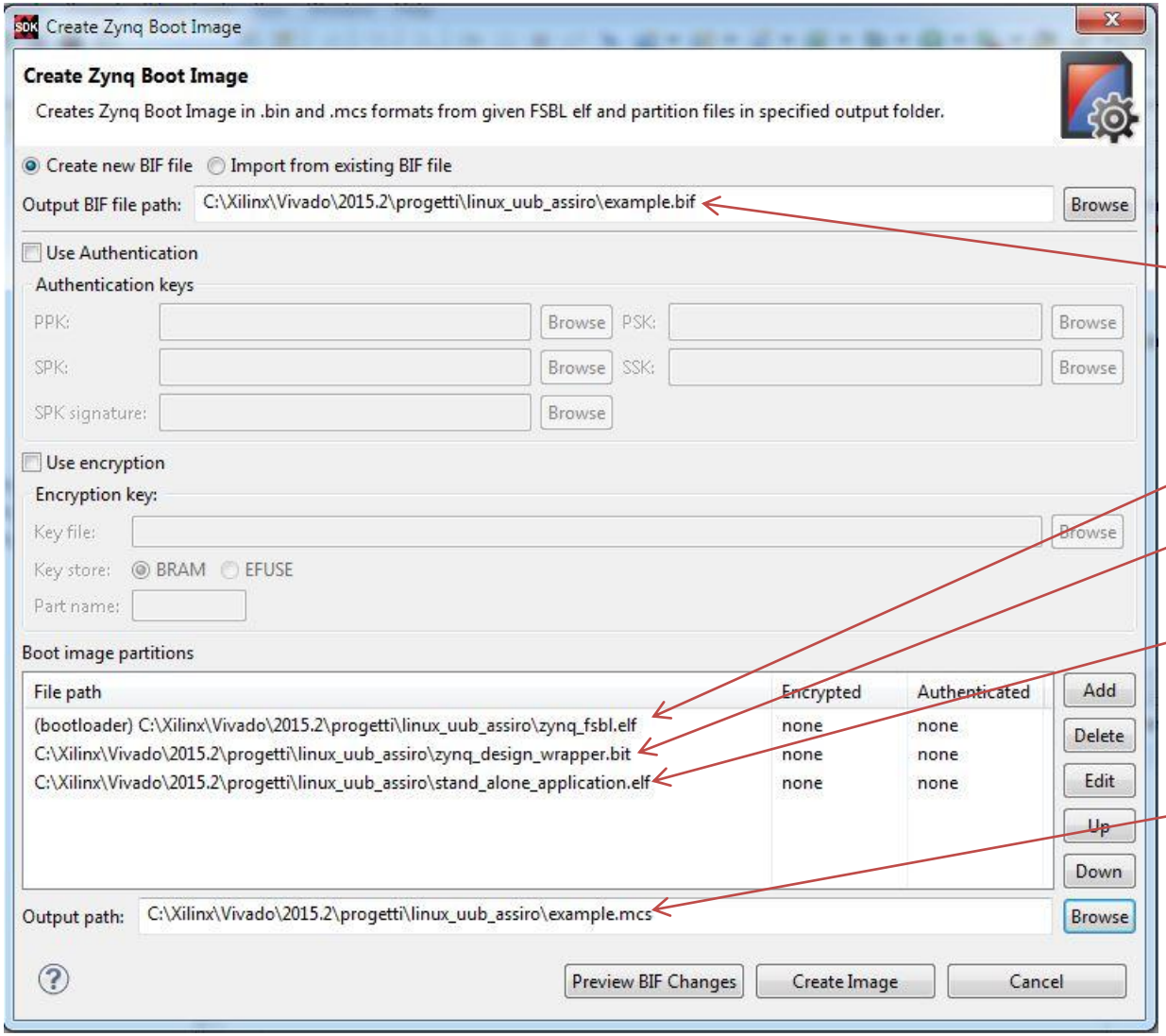


To run a stand alone application on the Zynq you need:
Software Xilinx SDK
Files: FSBL (generated by SDK), bitstream file (generated from Vivado for the PL side of the FPGA) and your application program written and compiled in SDK (executable file.elf)

How to create Boot image by SDK

This procedure is for a stand alone application program.

Start SDK and select from the Xilinx Tools menu “**Create Zynq boot image**". This window will be displayed.



.bif file is the building definitions of the image

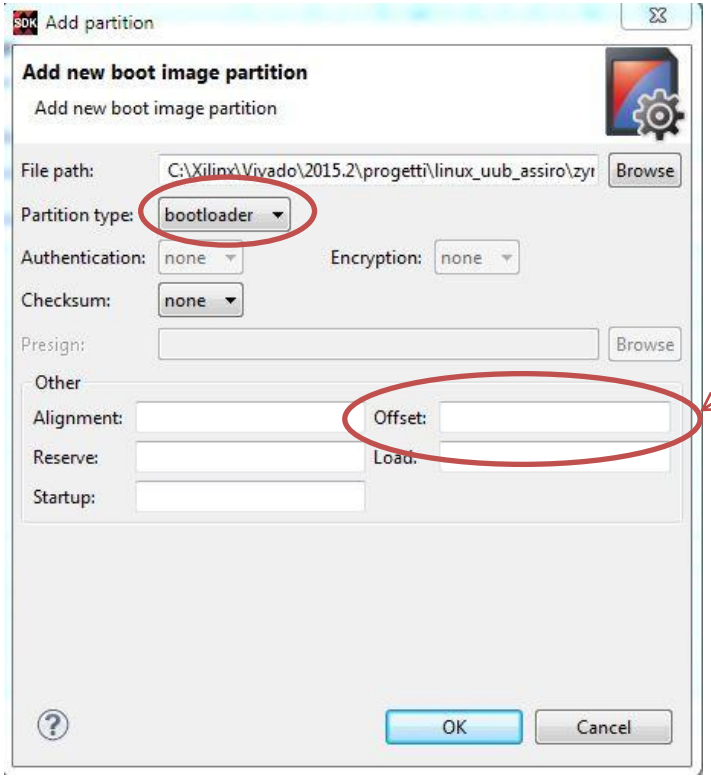
To run a stand alone application on the Zynq you need three files for the QSPI memory flash:

Files: FSBL (generated by SDK)

bitstream file (generated from Vivado for the PL side of the FPGA)

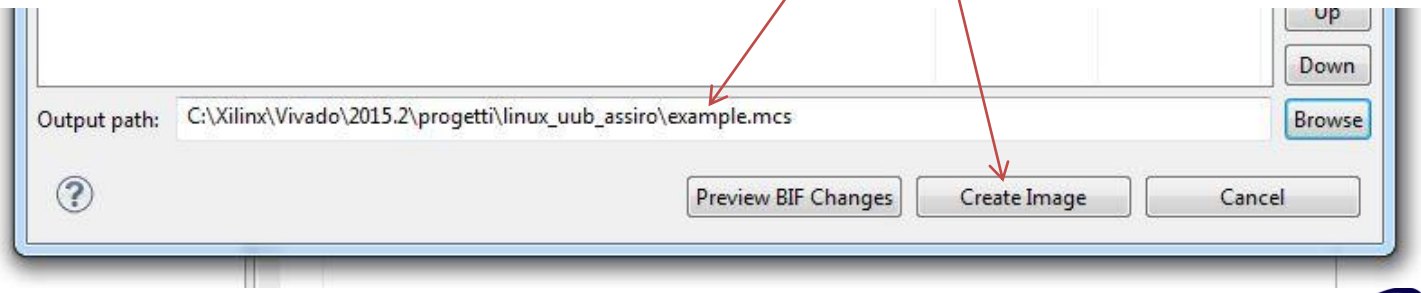
Application.elf program written and compiled in SDK (executable file.elf)

.bin or .msc are formats for the file to send, by progammer tools to the QSPI
Both formats are compatible for Jtag programmer



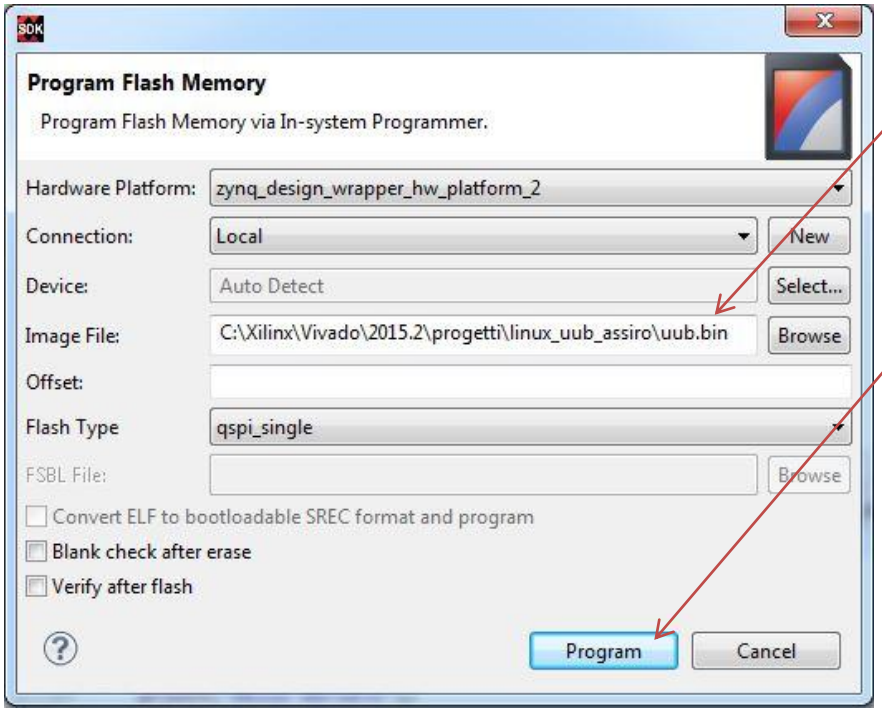
Click Add button to create a new partition into the memory, and select the file by Browse button.
First file is FSBL as bootloader type
All the other files are automatically selected as datafile partition type.
No OFFSET needed, all files will be allocated into the QSPI memory consecutively

Select name and format extension
Of the file and click on Create Image.
The image file will be create in the project folder



How to program Boot image in the QSPI Flash of UUB

In SDK select from the Xilinx Tools menu "**Program flash**". This window will be displayed**.**

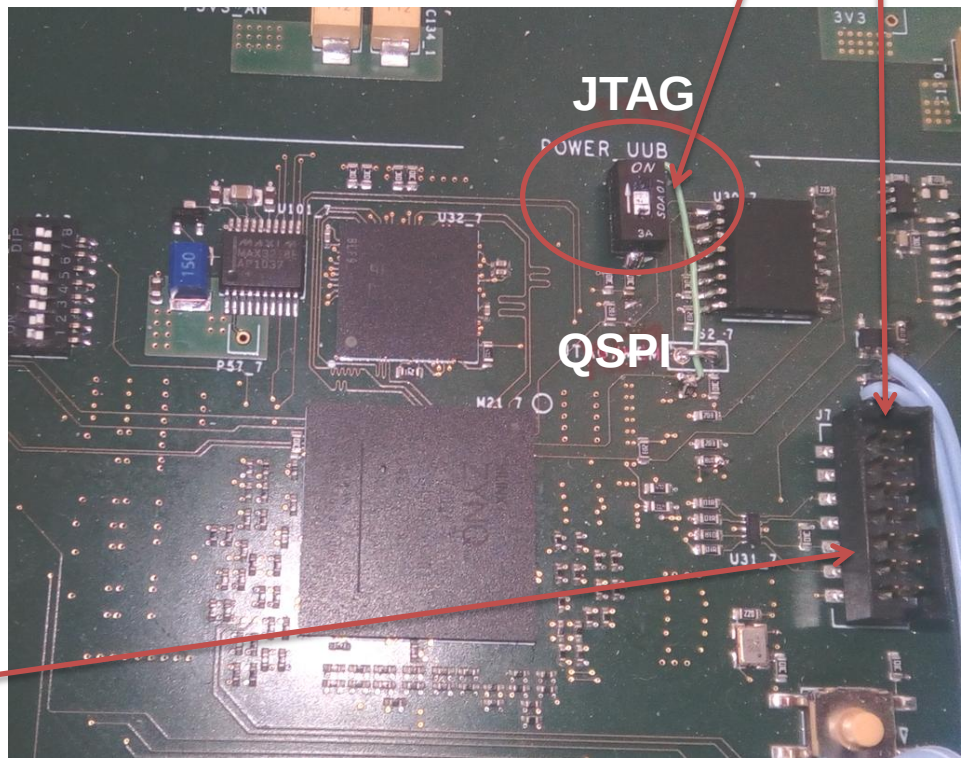


Select the image file generated (.mcs or .bin)

Connect the JTAG programmer to the UUB to the J7 connector

Switch up on JTAG side

Power on to the UUB (24Volt) and click on Program



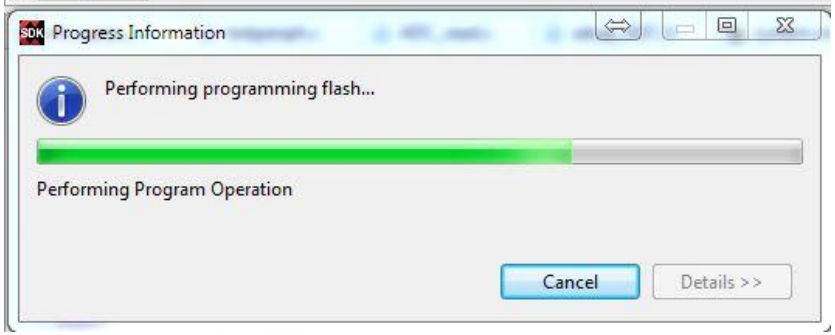
DLC10 xilinx platform cable USB JTAG programmer

The process to program the QSPI takes a very long time (about 20 minutes)

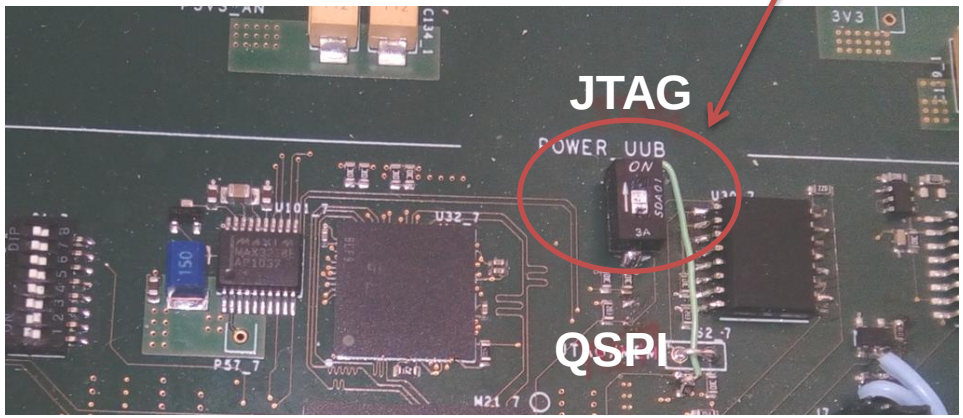
Console

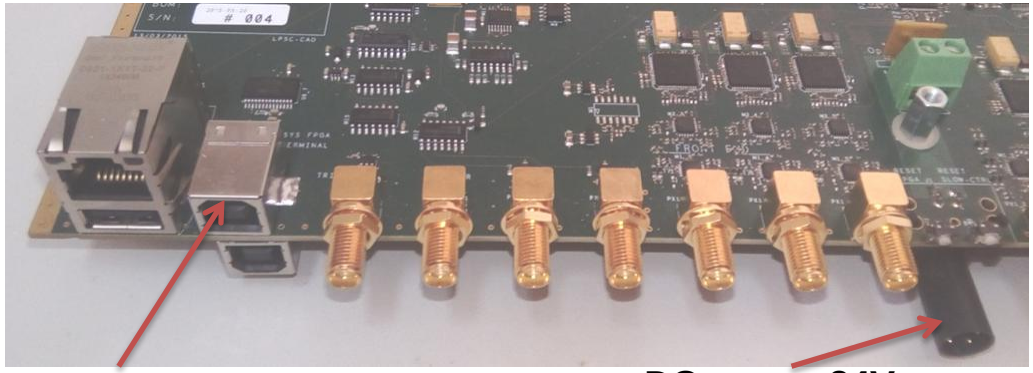
Program Flash

```
Device    ID Code          IR Length    Part Name
 1        4ba00477            4         arm_dap
 2        23727093            6         xc7z020

-----------------------------------------------------
Enabling extended memory access checks for Zynq.
Writes to reserved memory are not permitted and reads return 0.
To disable this feature, run "debugconfig -memory_access_check disable".

-----------------------------------------------------

CortexA9 Processor Configuration
-------------------------------------
Version.............................0x00000003
User ID.............................0x00000000
No of PC Breakpoints................6
No of Addr/Data Watchpoints.........4
Processor Reset .... DONE
Performing Erase Operation...
Erase Operation successful.
Performing Program Operation...
0%.....................................................................
```



When the programmig is done place the switch on **QSPI side** and reset the Zynq.
Connect a USB cable to console
Application is now runnung.in to the Zynq….

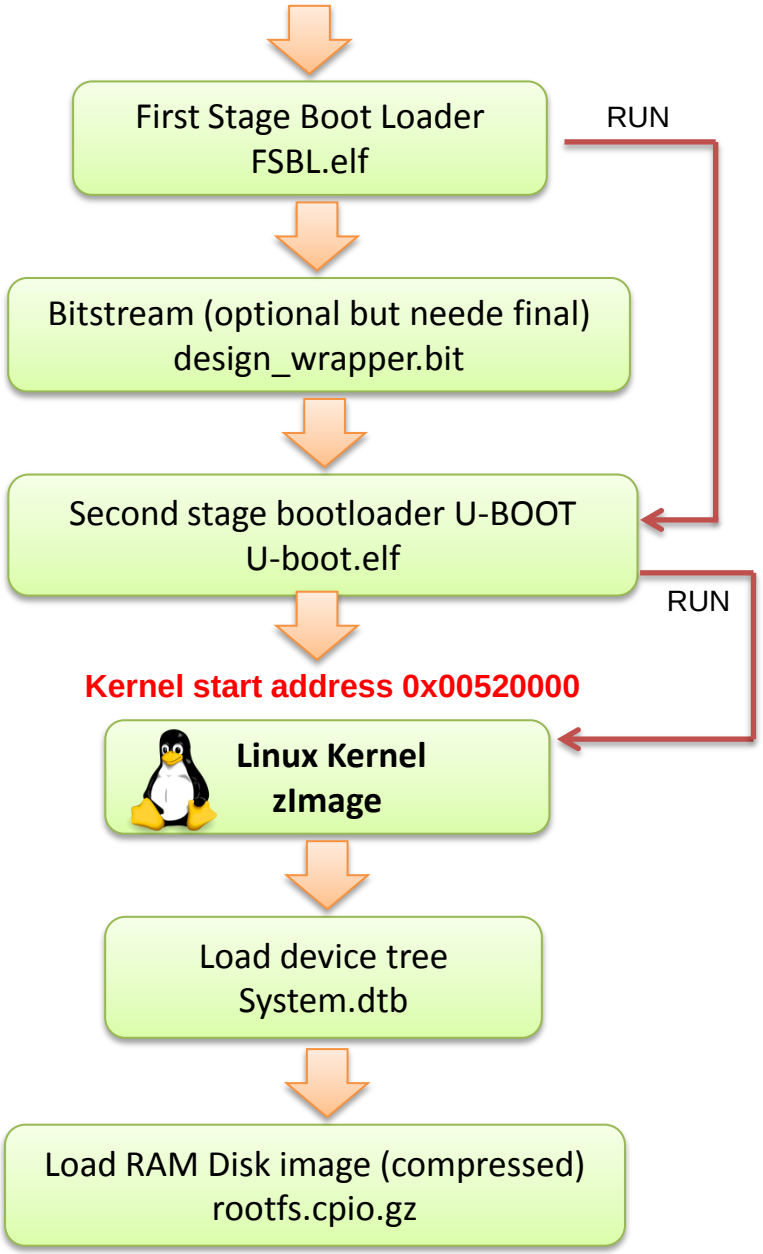




Zynq serial console (top side)

DC power 24V

Running Petalinux (QSPI memory map)



To build a linux image for QSPI memory on UUB's Zynq you need:
Software Xilinx SDK
Files:
- FSBL (generated by petalinux builder)
- Bitstream file (generated from Vivado for the PL side of the FPGA)
- Universal Booloader for Zynq (folder images of petalinux builder)
- Kernel image (zImage)
- Device tree file and image of root file system compressed.

All of this file are generated in petalinux compliling process. I will describe this process in my next document.
The QSPI memory allocation:

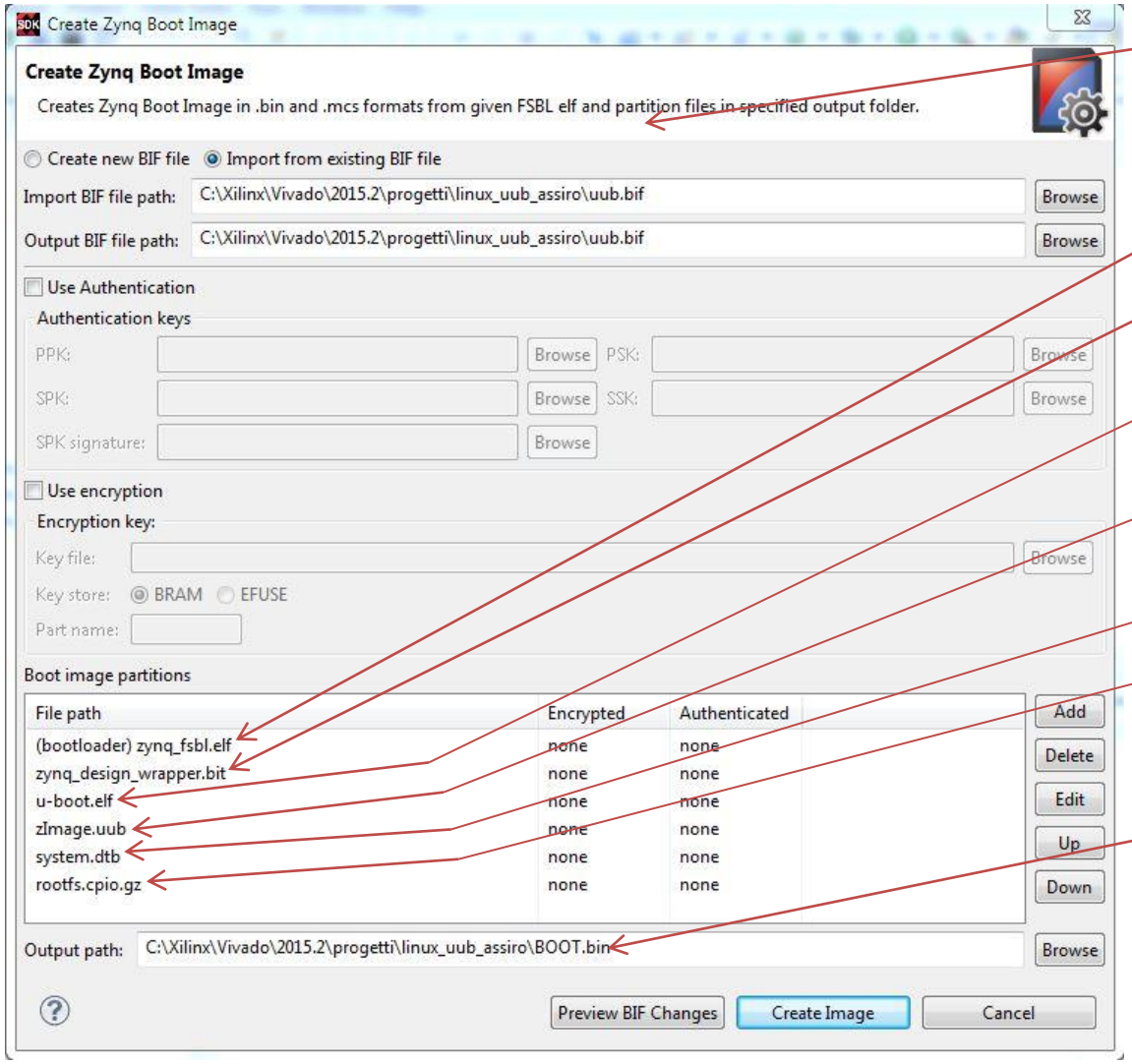
FBSL from the address 0x00001700 runs U-boot for the second stage of the boot

U-Boot runs automatically the kernel image from the address memory 0x520000 of the QSPI flash. (to change this parameter I think is necessary to recompile U-BOOT). By serial terminal windows is possible to stop the process to control U-boot directly.

How to create Petalinux Boot image by SDK

This procedure is for programming a petalinux build

Start SDK and select from the Xilinx Tools menu "**Create Zynq boot image**". This window will be displayed.



.bif file is the build definitions of the image

To run petalinux on the Zynq you need five files for the QSPI memory flash:

Files: FSBL (generated by SDK)

bitstream file (generated from Vivado for the PL side of the FPGA)

u-boot.elf universal boot from petalinux image folder

zImage is the petalinux kernel
Is important to allocate it at **address 0x520000**

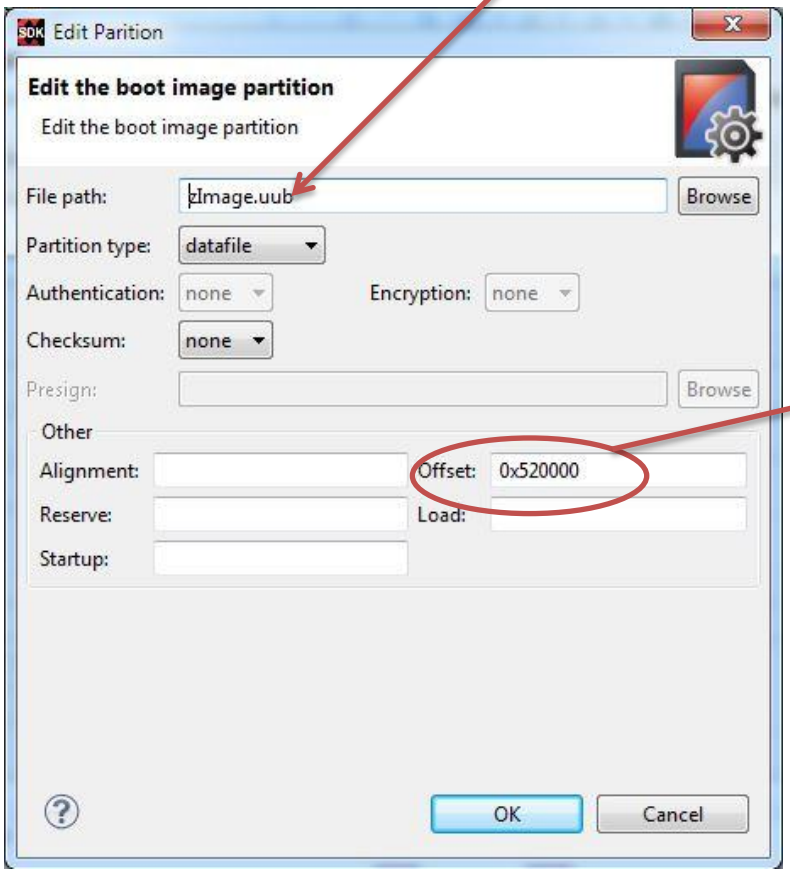
Sistem.dtb device tree

Rootfs.cpio.gz is the root file sistem compressed

.bin or .msc are formats for the file to send, by progammer tools to the QSPI
Both formats are compatible for Jtag programmer

QSPI Flash Memory partitions

Create boot image requires a file extension
This is just a renamed uImage file



QSPI memory map