Auger UUB flash memory definitions

Sequence of commands to create partitions into the Flash memory using UBI sub-system

September 2016 - P. Allison - R. Assiro

The first boot needs to be done over JTAG, because nothing is in flash yet. We want to use XMD and petalinux-boot to do this. Petalinux-boot does not seem to be able to force a reset, however, so to be safe, we need to use XMD to do that

UUB jumper in jtag mode

```
XMD% connect arm hw       - direct connection by XMD console and jtag to Zynq processor

CortexA9 Processor Configuration

-------------------------------------
Version.............................0x00000003
User ID.............................0x00000000
No of PC Breakpoints................6
No of Addr/Data Watchpoints.........4

Connected to "arm" target. id = 64
Starting GDB server for "arm" target (id = 64) at TCP port no 1234
XMD% fpga -f C:/petalinux/images/fsbl_uboot_allison/fpga.bit      - Stable with bitstream
XMD% dow C:/petalinux/images/zynq_fsbl_JTAG.elf
XMD% con                                         - run first step boot loader
XMD% stop                                        - stop fsbl (jtag version)
XMD% dow C:/petalinux/images/u-boot.elf          - load u-boot
XMD% con                                         - run u-boot
####################### NEXT COMMANDS UNDER U-BOOT  ##################################
Xilinx First Stage Boot Loader
Release 2015.2  Jan  7 2016-11:21:38
Devcfg driver initialized
Silicon Version 3.1
Boot mode is JTAG
RESET_SUCCESS is high.
Model: uub-linux
DRAM:  ECC disabled 512 MiB
SF: Detected N25Q1024 with page size 256 Bytes, erase size 64 KiB, total 256 MiB
UBI: attaching mtd1 to ubi0
UBI: scanning is finished
UBI: attached mtd1 (name "mtd=1", size 24 MiB) to ubi0
UBI: PEB size: 65536 bytes (64 KiB), LEB size: 65408 bytes
UBI: min./max. I/O unit sizes: 1/256, sub-page size 1
UBI: VID header offset: 64 (aligned 64), data offset: 128
UBI: good PEBs: 384, bad PEBs: 0, corrupted PEBs: 0
UBI: user volume: 3, internal volumes: 1, max. volumes count: 128
UBI: max/mean erase counter: 2/0, WL threshold: 4096, image sequence number: 85381688
UBI: available PEBs: 0, total reserved PEBs: 384, PEBs reserved for bad PEB handling: 0
In:    serial
Out:   serial
Err:   serial
Net:   Gem.e000b000
Error: Gem.e000b000 address not set.

Hit any key to stop autoboot:  0
U-Boot-PetaLinux>
```

```
U-Boot-PetaLinux> sf probe
U-Boot-PetaLinux> sf erase qspi-ubi-itb 0x1800000        - erase flash
U-Boot-PetaLinux> ubi part qspi-ubi-itb
U-Boot-PetaLinux> ubi create u-boot-env1 0xFF80 s        - create the volumes
U-Boot-PetaLinux> ubi create u-boot-env2 0xFF80 s
```

```
U-Boot-PetaLinux> ubi create itbs 0x1794300
```

U-BOOT enviroment variables settings:

```
U-Boot-PetaLinux> set default_bootcmd "ubifsmount ubi0:itbs && ubifsload 0x1000000 fpga.bit &&
fpga loadb 0 0x1000000 0x3dbafc && ubifsmount ubi0:itbs && ubifsload ${loadaddr} image.ub &&
bootm"

U-Boot-PetaLinux> set mtdparts "mtdparts=spi32764.0:2m(qspi-fsbl-uboot),24m(qspi-ubi-
itb),102m(qspi-ubi-rootfs)"

U-Boot-PetaLinux> saveenv                        - save environment to UBI volumes
U-Boot-PetaLinux> saveenv                        - and do it again
```

Steps to first boot of Petalinux and contents of MTD1 and MTD2

Now U-Boot has prepared the UBI environment. So we now need to boot Linux to get access to everything!
I launch petalinux by TFTP from my linux server machine

```
U-Boot-PetaLinux> run netboot             - download by TFTP image.ub and launch petallinux

####################### NEXT COMMANDS UNDER PETALINUX  ##################################
MTD1 --- BOOT.BIN file is generated by SDK with inside fsbl.elf and u-boot.elf
uilt with PetaLinux v2015.2.1 (Yocto 1.8) Auger-uub /dev/ttyPS1
Auger-uub login: root
Password: root
login[886]: root login on 'ttyPS1'
root@Auger-uub:/#
root@Auger-uub:/# cd /var/ftp
root@Auger-uub:/var/ftp tftp -g -l BOOT.BIN -r BOOT.BIN 172.16.17.198  - load BOOT.BIN
root@Auger-uub:/# flash_erase /dev/mtd1 0 0         - erase MTD1 partition of flash
root@Auger-uub:/# flashcp BOOT.BIN /dev/mtd1         - write BOOT.BIN into flash MTD1
```

MTD2 contents and commands sequence to Update the system

```
root@Auger-uub:/var/ftp tftp -g -l image.ub -r image.ub 172.16.17.198   - load image.ub
root@Auger-uub:/var/ftp tftp -g -l fpga.bit -r image.ub 172.16.17.198   - load bitstream
root@Auger-uub:/var/ftp ubiattach -p /dev/mtd2               - attach partition to UBI
root@Auger-uub:/var/ftp mount -t ubifs ubi0:itbs /boot       - mount volume itbs to UBI
root@Auger-uub:/var/ftp cp image.ub /boot
root@Auger-uub:/var/ftp cp fpga.bit /boot
root@Auger-uub:/var/ftp sync                           - synchronization with flash memory
```

image.ub (linux image) and fpga.bit (bitstream) are stored into MTD2 partition of flash memory.

Turn off the UUB and switch UUB jumper in flash memory mode

At this point, the system is ready. Reboot, and U-Boot should load the bitsteam into FPGA, and then automatically load image.ub from the filesystem and boot

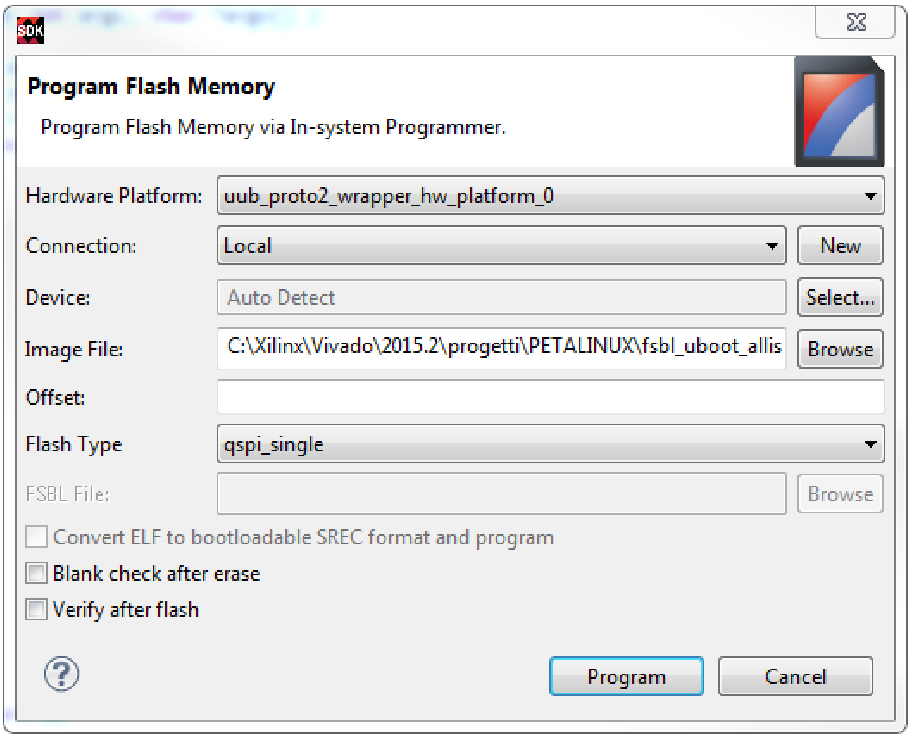
How to flash the memory by TFTP

- Turn on the UUB and wait to run u-boot
- press enter before the seconds counting and will get the prompt
- U-Boot-PetaLinux>
- U-Boot-PetaLinux> **sf probe**
- U-Boot-PetaLinux> **TFTP 0x1000000 UUB.BIN**

```
Load address: 0x1000000

Loading: #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################

         946.3 KiB/s

done

Bytes transferred = 33554432 (2000000 hex)

U-Boot-PetaLinux>sf update 0x1000000 0x0 0x2000000

device 0 offset 0x0, size 0x2000000

18481152 bytes written, 15073280 bytes skipped in 59.551s, speed 578524 B/s

U-Boot-PetaLinux>
```

How to flash the memory by SDK tool

Flash the memory by Jtag and DLC10 xilinx programmer



Program Flash Memory

Program Flash Memory via In-system Programmer.

Hardware Platform: uub_proto2_wrapper_hw_platform_0

Connection: Local

Device: Auto Detect

Image File: C:\Xilinx\Vivado\2015.2\progetti\PETALINUX\fsbl_uboot_allis

Offset:

Flash Type: qspi_single

FSBL File:

□ Convert ELF to bootloadable SREC format and program

□ Blank check after erase

□ Verify after flash

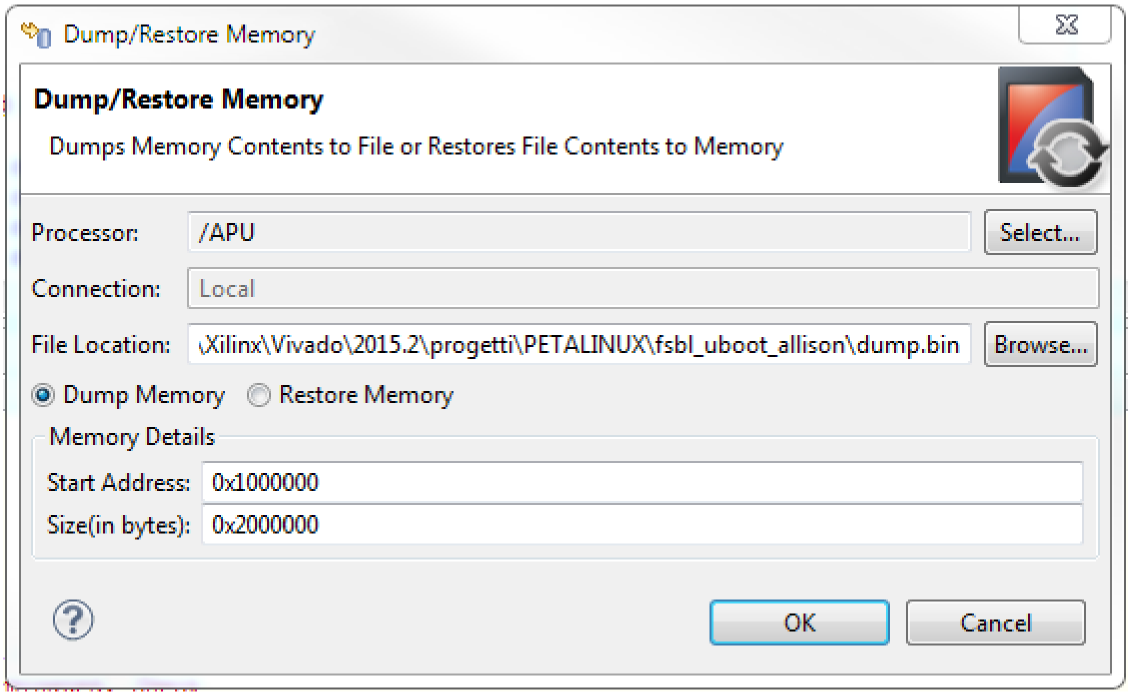
Sequence to read the image of entire flash memory

This is the procedure to get a copy of entire flash memory to clone it and distribute to other boards

```
U-Boot-PetaLinux> sf probe
U-Boot-PetaLinux> sf read 0x1000000 0x0 0x2000000        - transfer memory content to RAM
- 0x1000000 RAM address
- 0x0 first flash memory address
- 0x2000000 last flash memory address
```

Dump/Restore Memory is the SDK's tool to read the RAM contents to create the bin file.

Launch the tool during u-boot is running



Dump/Restore Memory

Dump/Restore Memory

Dumps Memory Contents to File or Restores File Contents to Memory

Processor: /APU

Connection: Local

File Location: \Xilinx\Vivado\2015.2\progetti\PETALINUX\fsbl_uboot_allison\dump.bin

● Dump Memory ○ Restore Memory

Memory Details

Start Address: 0x1000000

Size(in bytes): 0x2000000

Click OK and wait about 12 minutes to read and create the file (32 Mb) in the folder