**SD Card/JTAG Boot up for Zynq board:**

The bootable SD Card contains fallowing images

1. BOOT.BIN – contains fsbl + uboot + bit file(optional)
2. Image.ub – contain kernel image + dtb+ ramfs

By using peta linux we can generate both images. The below documents describes the steps to generate images fsbl.elf, u-boot.elf, uImage, devicetree.dtb, uramdisk.image.gz with out using peta linux

**Stage-1:**

**Generate BOOT.bin (FSBL+.bit file + U-BOOT)**

**FSBL Generation:**

1. Generate FSBL from SDK with menu option

File ->New ->Application Project->Zynq FSBL

Target File fsbl.elf (Boot loader)

**U-boot Generation:**

Pull the source code from

* git clone <https://github.com/Xilinx/u-boot-xlnx.git>
* make zynq\_zc706\_config
* make

Target file u-boot, please rename it as **u-boot.elf**

**.Bit file:**

Copy .Bit file generated in Vivado from fallowing loation

/project\_4/project\_4.runs/impl\_1/ design\_1\_wrapper.bit

**BOOT.bin Image Generation:**

Xilinx Tools -> Create Boot Image -> (Please check below picture)

The output.bif file contains as fallows:

the\_ROM\_image:

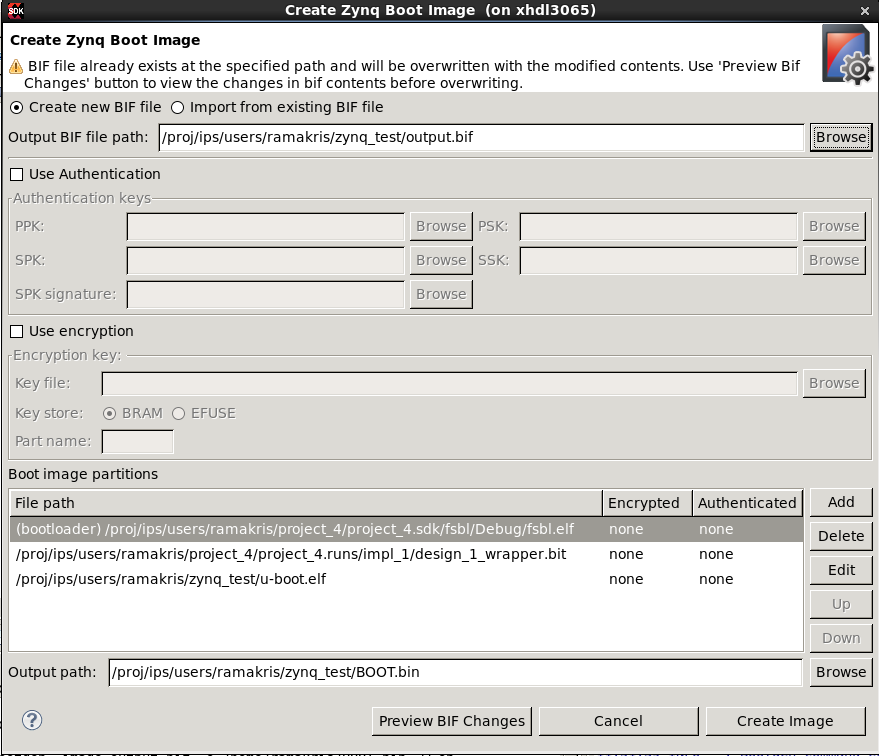
{

[bootloader]/proj/ips/users/ramakris/project\_4/project\_4.sdk/fsbl/Debug/fsbl.elf

/proj/ips/users/ramakris/project\_4/project\_4.runs/impl\_1/design\_1\_wrapper.bit

/proj/ips/users/ramakris/zynq\_test/u-boot.elf

}



**Stage-2:**

**Kernel Image:**

Download source code from

bash$ git clone <https://github.com/Xilinx/linux-xlnx.git>

make ARCH=arm xilinx\_zynq\_defconfig

make ARCH=arm menuconfig

To produce Kernel image

make ARCH=arm UIMAGE\_LOADADDR=0x8000 uImage

Taget Image: linux-xlnx/arch/arm/boot/uImage

Note: mkimage command if it is not availabe, please add the fallowing to your PATH variable

/proj/epdsw1/Sundeep/mkimage

|  |  |
| --- | --- |
| **DTB file:** |  |

**Generate HDF file from hardware project**

* Open the hardware project in Vivado.
* Generate Block Design

IP Integrator: Generate Block Design

# Export the hardware system to SDK:

Vivado Menu: File > Export > Export Hardware

Generate a Device Tree Source (.dts/.dtsi) files from SDK

1. Open SDK from Vivado or open SDK via command line (xsdk -hwspec <filename>.hdf -workspace <workspace>  
   Vivado Menu: File > Launch SDK
2. Clone Device tree code bash$ **git clone** [**https://github.com/Xilinx/device-tree-xlnx.git**](https://github.com/Xilinx/device-tree-xlnx.git)
3. Add the BSP repository in SDK

SDK Menu: Xilinx Tools > Repositories > New... (<bsp repo>) > OK

1. Create a Device Tree Board Support Package (BSP):

SDK Menu: File > New > Board Support Package > Board Support Package OS: device-tree > Finish

After above steps the **system.dts** will be created under project\_x/project\_x.sdk/device\_tree\_bsp\_0

Generate **devicetree.dtb** as fallows

linux-xlnx/scripts/dtc/dtc -I dts -O dtb -o devicetree.dtb system.dts

Target output **devicetree.dtb**

**Stage-3:**

**Root File system:**

To use already exisiting **uramdisk.image.gz**, download from

<http://www.wiki.xilinx.com/Zynq+Release+14.3>

**Modifying the root File system:**

The below procedure add one **startup application.**

There are two types of formats being used for root filesystems.

Before Vivado/SDK version 2013.1, Linux for Zynq AP SoC used an initial ramdisk/initrd (ramdisk.image.gz);

Since then, the Zynq uses initramfs. MicroBlaze and PowerPC are using initramfs (initramfs.cpio.gz).

We use **initramfs** in the below section

**To modify initramfs:**

1. Unwrap the uboot header for the existing **uramdisk.image.gz**

dd if=uramdisk.image.gz bs=64 skip=1 of= ramdisk.image.gz

1. mkdir my\_mount
2. gunzip -c ramdisk.image.gz | sudo sh -c 'cd my\_mount/ && cpio -i'
3. cd my\_mount
4. cd home/root
5. cp /home/ramakris/file.sh . (file.sh is simple hello world script, you can also place arm bcross compiled executable)
6. cd ../
7. cd etc/init.d
8. vi rc
9. add the fallowing contents at end of rc file

cd /home/root

sh file.sh

1. Repack file system into cpio archive

sh -c 'cd my\_mount/ && sudo find . | sudo cpio -H newc -o' | gzip -9 > new\_initramfs.cpio.gz

1. mkimage -A arm -T ramdisk -d new\_initramfs.cpio.gz uramdisk.image.gz

---------------------------------SD Card Boot Log with above images--------------------------------

Xilinx First Stage Boot Loader

Release 2015.4 Dec 15 2015-11:47:30

Devcfg driver initialized

Silicon Version 3.1

Boot mode is SD

SD Init Done

Flash Base Address: 0xE0100000

Reboot status register: 0x60400000

Multiboot Register: 0x0000C000

Image Start Address: 0x00000000

Partition Header Offset:0x00000C80

Partition Count: 3

Partition Number: 1

Header Dump

Image Word Len: 0x0032D130

Data Word Len: 0x0032D130

Partition Word Len:0x0032D130

Load Addr: 0x00000000

Exec Addr: 0x00000000

Partition Start: 0x000065D0

Partition Attr: 0x00000020

Partition Checksum Offset: 0x00000000

Section Count: 0x00000001

Checksum: 0xFF67242E

Bitstream

Partition Number: 2

Header Dump

Image Word Len: 0x0001B5B0

Data Word Len: 0x0001B5B0

Partition Word Len:0x0001B5B0

Load Addr: 0x04000000

Exec Addr: 0x04000000

Partition Start: 0x00333700

Partition Attr: 0x00000010

Partition Checksum Offset: 0x00000000

Section Count: 0x00000001

Checksum: 0xF7C7A57E

Application

Handoff Address: 0x04000000

SUCCESSFUL\_HANDOFF

FSBL Status = 0x1

U-Boot 2015.07-00013-g3c141dc (Dec 08 2015 - 09:26:39 -0500)

Model: Zynq ZC706 Development Board

I2C: ready

DRAM: ECC disabled 1 GiB

MMC: zynq\_sdhci: 0

SF: Detected S25FL128S\_64K with page size 512 Bytes, erase size 128 KiB, total 32 MiB

\*\*\* Warning - bad CRC, using default environment

In: serial

Out: serial

Err: serial

Model: Zynq ZC706 Development Board

Net: Gem.e000b000

Hit any key to stop autoboot: 0

Device: zynq\_sdhci

Manufacturer ID: 3

OEM: 5344

Name: SU04G

Tran Speed: 50000000

Rd Block Len: 512

SD version 3.0

High Capacity: Yes

Capacity: 3.7 GiB

Bus Width: 4-bit

Erase Group Size: 512 Bytes

reading uEnv.txt

\*\* Unable to read file uEnv.txt \*\*

Copying Linux from SD to RAM...

reading uImage

3532176 bytes read in 307 ms (11 MiB/s)

reading devicetree.dtb

8909 bytes read in 17 ms (511.7 KiB/s)

reading uramdisk.image.gz

6167142 bytes read in 530 ms (11.1 MiB/s)

## Booting kernel from Legacy Image at 02080000 ...

Image Name: Linux-4.0.0-xilinx-00020-g5b3c91

Image Type: ARM Linux Kernel Image (uncompressed)

Data Size: 3532112 Bytes = 3.4 MiB

Load Address: 00008000

Entry Point: 00008000

Verifying Checksum ... OK

## Loading init Ramdisk from Legacy Image at 04000000 ...

Image Name:

Image Type: ARM Linux RAMDisk Image (gzip compressed)

Data Size: 6167078 Bytes = 5.9 MiB

Load Address: 00000000

Entry Point: 00000000

Verifying Checksum ... OK

## Flattened Device Tree blob at 02000000

Booting using the fdt blob at 0x2000000

Loading Kernel Image ... OK

Loading Ramdisk to 1fa1e000, end 1ffffa26 ... OK

Loading Device Tree to 1fa18000, end 1fa1d2cc ... OK

Starting kernel ...

Booting Linux on physical CPU 0x0

Linux version 4.0.0-xilinx-00020-g5b3c91b (root@xhdipslabd05) (gcc version 4.9.1 (Sourcery CodeBench Lite 2014.11-30) ) #1 SMP PREEMPT Thu Dec 10 14:04:27 EST 2015

CPU: ARMv7 Processor [413fc090] revision 0 (ARMv7), cr=18c5387d

CPU: PIPT / VIPT nonaliasing data cache, VIPT aliasing instruction cache

Machine model: xlnx,zynq-7000

cma: Reserved 16 MiB at 0x3f000000

Memory policy: Data cache writealloc

PERCPU: Embedded 11 pages/cpu @eefd5000 s12672 r8192 d24192 u45056

Built 1 zonelists in Zone order, mobility grouping on. Total pages: 260624

Kernel command line: console=ttyPS0,115200

PID hash table entries: 4096 (order: 2, 16384 bytes)

Dentry cache hash table entries: 131072 (order: 7, 524288 bytes)

Inode-cache hash table entries: 65536 (order: 6, 262144 bytes)

Memory: 1009852K/1048576K available (4762K kernel code, 223K rwdata, 1708K rodata, 224K init, 208K bss, 22340K reserved, 16384K cma-reserved, 253952K highmem)

Virtual kernel memory layout:

vector : 0xffff0000 - 0xffff1000 ( 4 kB)

fixmap : 0xffc00000 - 0xfff00000 (3072 kB)

vmalloc : 0xf0000000 - 0xff000000 ( 240 MB)

lowmem : 0xc0000000 - 0xef800000 ( 760 MB)

pkmap : 0xbfe00000 - 0xc0000000 ( 2 MB)

modules : 0xbf000000 - 0xbfe00000 ( 14 MB)

.text : 0xc0008000 - 0xc0659e6c (6472 kB)

.init : 0xc065a000 - 0xc0692000 ( 224 kB)

.data : 0xc0692000 - 0xc06c9ce0 ( 224 kB)

.bss : 0xc06c9ce0 - 0xc06fdff4 ( 209 kB)

Preemptible hierarchical RCU implementation.

Additional per-CPU info printed with stalls.

RCU restricting CPUs from NR\_CPUS=4 to nr\_cpu\_ids=2.

RCU: Adjusting geometry for rcu\_fanout\_leaf=16, nr\_cpu\_ids=2

NR\_IRQS:16 nr\_irqs:16 16

L2C: platform modifies aux control register: 0x72360000 -> 0x72760000

L2C: DT/platform modifies aux control register: 0x72360000 -> 0x72760000

L2C-310 erratum 769419 enabled

L2C-310 enabling early BRESP for Cortex-A9

L2C-310 full line of zeros enabled for Cortex-A9

L2C-310 ID prefetch enabled, offset 1 lines

L2C-310 dynamic clock gating enabled, standby mode enabled

L2C-310 cache controller enabled, 8 ways, 512 kB

L2C-310: CACHE\_ID 0x410000c8, AUX\_CTRL 0x76760001

slcr mapped to f0004000

zynq\_clock\_init: clkc starts at f0004100

Zynq clock init

sched\_clock: 64 bits at 333MHz, resolution 3ns, wraps every 3298534883328ns

timer #0 at f0008000, irq=17

Console: colour dummy device 80x30

Calibrating delay loop... 1332.01 BogoMIPS (lpj=6660096)

pid\_max: default: 32768 minimum: 301

Mount-cache hash table entries: 2048 (order: 1, 8192 bytes)

Mountpoint-cache hash table entries: 2048 (order: 1, 8192 bytes)

CPU: Testing write buffer coherency: ok

CPU0: thread -1, cpu 0, socket 0, mpidr 80000000

Setting up static identity map for 0x482980 - 0x4829d8

CPU1: thread -1, cpu 1, socket 0, mpidr 80000001

Brought up 2 CPUs

SMP: Total of 2 processors activated (2664.03 BogoMIPS).

CPU: All CPU(s) started in SVC mode.

devtmpfs: initialized

VFP support v0.3: implementor 41 architecture 3 part 30 variant 9 rev 4

pinctrl core: initialized pinctrl subsystem

NET: Registered protocol family 16

DMA: preallocated 256 KiB pool for atomic coherent allocations

cpuidle: using governor ladder

cpuidle: using governor menu

hw-breakpoint: found 5 (+1 reserved) breakpoint and 1 watchpoint registers.

hw-breakpoint: maximum watchpoint size is 4 bytes.

zynq-ocm f800c000.ocmc: ZYNQ OCM pool: 256 KiB @ 0xf0080000

vgaarb: loaded

SCSI subsystem initialized

usbcore: registered new interface driver usbfs

usbcore: registered new interface driver hub

usbcore: registered new device driver usb

media: Linux media interface: v0.10

Linux video capture interface: v2.00

pps\_core: LinuxPPS API ver. 1 registered

pps\_core: Software ver. 5.3.6 - Copyright 2005-2007 Rodolfo Giometti <giometti@linux.it>

PTP clock support registered

EDAC MC: Ver: 3.0.0

Advanced Linux Sound Architecture Driver Initialized.

Switched to clocksource arm\_global\_timer

NET: Registered protocol family 2

TCP established hash table entries: 8192 (order: 3, 32768 bytes)

TCP bind hash table entries: 8192 (order: 4, 65536 bytes)

TCP: Hash tables configured (established 8192 bind 8192)

TCP: reno registered

UDP hash table entries: 512 (order: 2, 16384 bytes)

UDP-Lite hash table entries: 512 (order: 2, 16384 bytes)

NET: Registered protocol family 1

RPC: Registered named UNIX socket transport module.

RPC: Registered udp transport module.

RPC: Registered tcp transport module.

RPC: Registered tcp NFSv4.1 backchannel transport module.

Trying to unpack rootfs image as initramfs...

Freeing initrd memory: 6024K (dfa1e000 - e0000000)

hw perfevents: enabled with armv7\_cortex\_a9 PMU driver, 7 counters available

futex hash table entries: 512 (order: 3, 32768 bytes)

jffs2: version 2.2. (NAND) (SUMMARY) © 2001-2006 Red Hat, Inc.

bounce: pool size: 64 pages

io scheduler noop registered

io scheduler deadline registered

io scheduler cfq registered (default)

zynq-pinctrl 700.pinctrl: zynq pinctrl initialized

dma-pl330 f8003000.dmac: Loaded driver for PL330 DMAC-241330

dma-pl330 f8003000.dmac: DBUFF-128x8bytes Num\_Chans-8 Num\_Peri-4 Num\_Events-16

e0001000.serial: ttyPS0 at MMIO 0xe0001000 (irq = 144, base\_baud = 3125000) is a xuartps

console [ttyPS0] enabled

xdevcfg f8007000.devcfg: ioremap 0xf8007000 to f006c000

[drm] Initialized drm 1.1.0 20060810

brd: module loaded

loop: module loaded

CAN device driver interface

libphy: MACB\_mii\_bus: probed

macb e000b000.ethernet eth0: Cadence GEM rev 0x00020118 at 0xe000b000 irq 146 (00:0a:35:00:01:22)

macb e000b000.ethernet eth0: attached PHY driver [Marvell 88E1116R] (mii\_bus:phy\_addr=e000b000.etherne:07, irq=-1)

e1000e: Intel(R) PRO/1000 Network Driver - 2.3.2-k

e1000e: Copyright(c) 1999 - 2014 Intel Corporation.

ehci\_hcd: USB 2.0 'Enhanced' Host Controller (EHCI) Driver

ehci-pci: EHCI PCI platform driver

usbcore: registered new interface driver usb-storage

mousedev: PS/2 mouse device common for all mice

i2c /dev entries driver

cdns-i2c e0004000.i2c: 400 kHz mmio e0004000 irq 141

Xilinx Zynq CpuIdle Driver started

Driver 'mmcblk' needs updating - please use bus\_type methods

sdhci: Secure Digital Host Controller Interface driver

sdhci: Copyright(c) Pierre Ossman

sdhci-pltfm: SDHCI platform and OF driver helper

sdhci-arasan e0100000.sdhci: No vmmc regulator found

sdhci-arasan e0100000.sdhci: No vqmmc regulator found

mmc0: SDHCI controller on e0100000.sdhci [e0100000.sdhci] using ADMA

mmc0: mmc\_rescan\_try\_freq: trying to init card at 400000 Hz

ledtrig-cpu: registered to indicate activity on CPUs

usbcore: registered new interface driver usbhid

usbhid: USB HID core driver

TCP: cubic registered

NET: Registered protocol family 17

can: controller area network core (rev 20120528 abi 9)

NET: Registered protocol family 29

can: raw protocol (rev 20120528)

can: broadcast manager protocol (rev 20120528 t)

can: netlink gateway (rev 20130117) max\_hops=1

Registering SWP/SWPB emulation handler

drivers/rtc/hctosys.c: unable to open rtc device (rtc0)

ALSA device list:

No soundcards found.

Freeing unused kernel memory: 224K (c065a000 - c0692000)

INIT: version 2.88 booting

mmc0: new high speed SDHC card at address aaaa

mmcblk0: mmc0:aaaa SU04G 3.69 GiB

mmcblk0: p1 p2

mkdir: can't create directory '/run/media/mmcblk0p1': No such file or directory

mount: mounting /dev/mmcblk0p1 on /run/media/mmcblk0p1 failed: No such file or directory

mkdir: can't create directory '/run/media/mmcblk0p2': No such file or directory

mount: mounting /dev/mmcblk0p2 on /run/media/mmcblk0p2 failed: No such file or directory

Creating /dev/flash/\* device nodes

random: dd urandom read with 1 bits of entropy available

starting Busybox inet Daemon: inetd... done.

update-rc.d: /etc/init.d/run-postinsts exists during rc.d purge (continuing)

Removing any system startup links for run-postinsts ...

/etc/rcS.d/S99run-postinsts

Hello world

INIT: Entering runlevel: 5

Configuring network interfaces... udhcpc (v1.23.1) started

Sending discover...

Sending discover...

Sending discover...

No lease, forking to background

done.

Starting Dropbear SSH server: Generating key, this may take a while...

Public key portion is:

ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCtG9nke3ZBnluCxZlodDxTQmKptvuh/cif0OaRtdY0BTiLFKW8ofauij7p+xRdrgM8tr0/HbpcM7PUdW1HDFpT7oA5u+u1x9vZP8H9pN2v7k0fJjF8OfjCKccEgVGlHdAA8hg5h40Reat/AXW7uE+65lzD1yVR0AfrnyT/cwkbOtejDpveUdr4ig0tQHidVC0RJMQxuXBFBmZ63r3UTEQA9xgnzBvklIHW48ShzUq671Ppl+tcx9CKkTePYhOXSpvR4crMFg+EaviEGTzklGIm/q+d5p0b0EVYlGHH8gOS5QhvZ0krr9We/fq3UbPXcvSca6PEDRRKKgQKkgvePt25 root@zynq

Fingerprint: md5 85:c4:4b:f3:7b:c5:5b:43:65:1f:f6:1a:17:ff:3d:15

dropbear.

Starting tcf-agent: OK

Hello world

Built with PetaLinux v2015.2 (Yocto 1.8) zynq /dev/ttyPS0

zynq login: root

Password:

login[916]: root login on 'ttyPS0'

root@zynq:~# root

-sh: root: command not found

root@zynq:~#

------------------------JTAG Boot Procedure------------------------------------

Connect the JTAG debugger to the processor

xmd% connect arm hw

Download and run the FSBL

xmd% dow fsbl.elf

xmd% con

# wait a little while

xmd% stop

Download the Linux image

xmd% dow -data devicetree.dtb 0x2a000000

xmd% dow -data uramdisk.image.gz 0x2000000

xmd% dow -data uImage 0x3000000

Download and run U-Boot

xmd% dow u-boot

xmd% con

This executes U-Boot, which will try booting the system. Interrupt U-Boots automatic boot attempt and boot the system with

u-boot> bootm 0x3000000 0x2000000 0x2a000000