- U-Boot ...
- U-Boot ...
- U-Boot I...
- U-Boot I...
- U-Boot ...
- U-Boot ...
- U-Boot ...
- U-Boot ...
- U-Boot ...
- U-boot ...
- U-Boot ...
- U-Boot ...
- U-Boot ...
- U-Boot ...
- U-boot ...
- U-boot ...

U-Boot USB Driver

创建者Confluence Wiki Admin
由 Terry O'Neal 最近更新于 Jan 14, 2020

Table of Contents

- Introduction
- U-boot Configuration
  - For Zynq
- Device Tree
  - For ZynqMP
  - For Zynq
- Test Procedure
- DFU testing in u-boot
  - Host setup
  - Build u-boot for DFU
  - Testing at u-boot
- Features
  - For Zynq
  - For ZynqMP

Introduction

The USB controller is capable of fulfilling a wide range of applications for USB 2.0 implementations. Two identical controllers are in the Zynq-7000 device. Each controller is configured and controlled independently. The USB controller I/O uses the ULPI protocol to connect external ULPI PHY via the MIO pins. The ULPI interface provides an 8-bit parallel SDR data path from the controller's internal UTMI-like bus to the PHY. The ULPI interface minimizes device pin count and is controlled by a 60 MHz clock output from the PHY.

Xilinx Wiki

- U-Boot ...
- U-Boot ...
- U-Boot I...
- U-Boot I...
- U-Boot ...
- U-Boot ...
- U-Boot ...
- U-Boot ...
- U-boot ...
- U-Boot ...
- U-Boot ...
- U-Boot ...
- U-Boot ...
- U-boot ...
- U-boot ...

```
CONFIG_USB=y
CONFIG_USB_ULPI_VIEWPORT=y
CONFIG_USB_ULPI=y
CONFIG_USB_STORAGE=y
CONFIG_USB_GADGET=y
CONFIG_USB_GADGET_MANUFACTURER="Xilinx"
CONFIG_USB_GADGET_VENDOR_NUM=0x03fd
CONFIG_USB_GADGET_PRODUCT_NUM=0x0300
CONFIG_CMD_USB=y
CONFIG_USB_DWC3=y
CONFIG_USB_XHCI_DWC3=y
CONFIG_USB_XHCI_ZYNQMP=y
CONFIG_USB_DWC3_GADGET=y
CONFIG_USB_HOST=y
CONFIG_USB_DWC3_GENERIC=y
CONFIG_ZYNQMP_USB=y
CONFIG_USB_STORAGE=y
CONFIG_USB_XHCI_HCD=y
```

For Zynq

```
CONFIG_USB_GADGET_DOWNLOAD=y
CONFIG_USB_ULPI_VIEWPORT=y
CONFIG_USB_ULPI=y
CONFIG_USB=y
CONFIG_USB_EHCI_HCD=y
CONFIG_CMD_USB=y
CONFIG_USB_GADGET_VENDOR_NUM=0x03fd
CONFIG_USB_STORAGE=y
```

- U-Boot ...
- U-Boot ...
- U-Boot I...
- U-Boot I...
- U-Boot ...
- U-Boot ...
- U-Boot ...
- U-Boot ...
- U-boot ...
- U-Boot ...
- U-Boot ...
- U-Boot ...
- U-Boot ...
- U-boot ...
- U-boot ...

Device Tree

For ZynqMP

```
/* ULPI SMSC USB3320 */
&&usb0 {
	status = "okay";
	pinctrl-names = "default";
	pinctrl-0 = <&&pinctrl_usb0_default>;
};

&&dwc3_0 {
	status = "okay";
	dr_mode = "host";
	snps,usb3_lpm_capable;
	phy-names = "usb3-phy";
	phys = <&&lane2 PHY_TYPE_USB3 0 2 26000000>;
	maximum-speed = "super-speed";
};

	pinctrl_usb0_default: usb0-default {
		mux {
			groups = "usb0_0_grp";
			function = "usb0";
		};

		conf {
			groups = "usb0_0_grp";
			slew-rate = <SLEW_RATE_SLOW>;
```

Xilinx Wiki

- U-Boot ...
- U-Boot ...
- U-Boot I...
- U-Boot I...
- U-Boot ...
- U-Boot ...
- U-Boot ...
- U-Boot ...
- U-boot ...
- U-Boot ...
- U-Boot ...
- U-Boot ...
- U-Boot ...
- U-boot ...
- U-boot ...

```
                conf-rx {
                        pins = "MIO52", "MIO53", "MIO55";
                        bias-high-impedance;
                };

                conf-tx {
                        pins = "MIO54", "MIO56", "MIO57", "MIO58", "MIO59",
                                "MIO60", "MIO61", "MIO62", "MIO63";
                        bias-disable;
                };
        };
```

For Zynq

```
&&usb0 {
        status = "okay";
        dr_mode = "host";
        usb-phy = <&&usb_phy0>;
        pinctrl-names = "default";
        pinctrl-0 = <&&pinctrl_usb0_default>;
};


        usb_phy0: phy0@e0002000 {
                compatible = "ulpi-phy";
                #phy-cells = <0>;
                reg = <0xe0002000 0x1000>;
                view-port = <0x0170>;
                drv-vbus;
        };
```

```
                groups = "usb0_0_grp";
                function = "usb0";
        };

         conf {
                groups = "usb0_0_grp";
                slew-rate = <0>;
                io-standard = <1>;
        };

        conf-rx {
                pins = "MIO29", "MIO31", "MIO36";
                bias-high-impedance;
        };

        conf-tx {
                pins = "MIO28", "MIO30", "MIO32", "MIO33", "MIO34",
                       "MIO35", "MIO37", "MIO38", "MIO39";
               bias-disable;
        };
};
```

Test Procedure

```
ZynqMP> usb start
starting USB...
USB0:   Register 2000440 NbrPorts 2
Starting the controller
USB XHCI 1.00
scanning bus 0 for devices... usb_setup_device dev->devnum 1
```

```
usb_prepare_device 2 2
set address 2
usb_setup_device dev->devnum 3
usb_prepare_device 3 3
set address 3
usb_setup_device dev->devnum 3
usb_prepare_device 3 3
set address 3
3 USB Device(s) found
       scanning usb for storage devices... 0 Storage Device(s) found
       scanning usb for ethernet devices... 0 Ethernet Device(s) found
ZynqMP>

ZynqMP> usb info
1: Hub, USB Revision 3.0
- U-Boot XHCI Host Controller
- Class: Hub
- PacketSize: 9 Configurations: 1
- Vendor: 0x0000 Product 0x0000 Version 1.0
   Configuration: 1
   - Interfaces: 1 Self Powered 0mA
     Interface: 0
     - Alternate Setting 0, Endpoints: 1
     - Class Hub
     - Endpoint 1 In Interrupt MaxPacket 8 Interval 255ms

2: Hub, USB Revision 2.10
- Microchip Tech USB2744
- Class: Hub
- PacketSize: 64 Configurations: 1
- Vendor: 0x0424 Product 0x2743 Version 2.2
   Configuration: 1
```

Xilinx Wiki

- U-Boot ...
- U-Boot ...
- U-Boot I...
- U-Boot I...
- U-Boot ...
- U-Boot ...
- U-Boot ...
- U-Boot ...
- U-boot ...
- U-Boot ...
- U-Boot ...
- U-Boot ...
- U-Boot ...
- U-boot ...
- U-boot ...

```
      - Class Hub
      - Endpoint 1 In Interrupt MaxPacket 1 Interval 12ms
      - Endpoint 1 In Interrupt MaxPacket 1 Interval 12ms

3: Mass Storage, USB Revision 2.0
- SanDisk Cruzer Blade 4C531001580304114232
- Class: (from Interface) Mass Storage
- PacketSize: 64 Configurations: 1
- Vendor: 0x0781 Product 0x5567 Version 1.39
   Configuration: 1
   - Interfaces: 1 Bus Powered 200mA
     Interface: 0
     - Alternate Setting 0, Endpoints: 2
     - Class Mass Storage, Transp. SCSI, Bulk only
     - Endpoint 1 In Bulk MaxPacket 512
     - Endpoint 2 Out Bulk MaxPacket 512

4: Vendor specific, USB Revision 2.1
- Microchip Tech Hub Controller
- Class: (from Interface) Vendor specific
- PacketSize: 64 Configurations: 1
- Vendor: 0x0424 Product 0x2740 Version 2.0
   Configuration: 1
   - Interfaces: 1 Self Powered 0mA
     Interface: 0
     - Alternate Setting 0, Endpoints: 0
     - Class Vendor specific
```

DFU testing in u-boot

Host setup

Xilinx Wiki

- U-Boot ...
- U-Boot ...
- U-Boot I...
- U-Boot I...
- U-Boot ...
- U-Boot ...
- U-Boot ...
- U-Boot ...
- U-Boot ...
- U-boot ...
- U-Boot ...
- U-Boot ...
- U-Boot ...
- U-Boot ...
- U-boot ...
- U-boot ...

Build u-boot for DFU

In order to test DFU at u-boot ensure that usb node has below two properties as shown.

dr_mode = "peripheral";

maximum-speed = "high-speed";

The dr_mode with "peripheral" informs u-boot to act as usb device and maximum-speed is used to inform about max-speed that driver supports.

NOTE: For ZynqMP maximum-speed can be super-speed also but as of now **u-boot doesn't support USB 3.0 so, always ensure that maximum-speed was set to high-speed.**

In case if you are building through petalinux, please ensure that these properties were set through system-user.dtsi at path<PetaLinux-project>/project-spec/meta-user/recipes-bsp/device-tree/files/system-user.dtsi. But, if we modify here and build linux then the Linux also will be restricted to high-speed. So, first build u-boot, copy the u-boot image and then change this maximum-speed property to super-speed and the build linux..

Example:

1. Add the following to system-user.dtsi, so that it looks like:

/include/ "system-conf.dtsi"

/ {

};

&dwc3_0 {

dr_mode = "peripheral";

maximum-speed = "high-speed";

};

Testing at u-boot

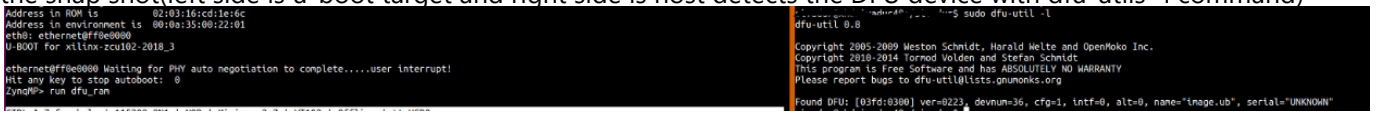
Xilinx Wiki

- U-Boot ...
- U-Boot ...
- U-Boot I...
- U-Boot I...
- U-Boot ...
- U-Boot ...
- U-Boot ...
- U-Boot ...
- U-boot ...
- U-Boot ...
- U-Boot ...
- U-Boot ...
- U-Boot ...
- U-boot ...
- U-boot ...

example:

```
"dfu_ram_info=" \
"setenv dfu_alt_info " \
"image.ub ram $fdt_addr $fdt_size\0" \
"dfu_ram=run dfu_ram_info && dfu 0 ram 0\0"
```

- Running dfu_ram lets the u-boot enter into DFU mode and it waits for image to be downloaded from the host. As per above env, it waits for kernel images image.ub to get downloaded on to ram at address "kernel_addr". Below is the snap shot(left side is u-boot target and right side is host detects the DFU device with dfu-utils -l command)



- Now at host, download the image.ub to target using dfu-util -D option. (example command : sudo dfu-util -d 03fd:0300 -D image.ub).
- This completes the downloading the kernel image using DFU,, now press ctrl+c at u-boot to comeout of DFU mode and the boot/use the downloaded image normally.

Features

For Zynq

The USB controller has the following key features:

- USB 2.0 High Speed Host controller (480 Mb/s)
- Intel® EHCI software programming model
- USB 2.0 HS and FS Device controller
- Up to 12 Endpoint: Control Endpoint plus 11 configurable Endpoints
- USB 1.1 legacy FS/LS
- Embedded Transaction Translator to support FS/LS in Host mode

For ZynqMP

- Two USB 2.0/3.0 controllers

- U-Boot ...
- U-Boot ...
- U-Boot I...
- U-Boot I...
- U-Boot ...
- U-Boot ...
- U-Boot ...
- U-Boot ...
- U-boot ...
- U-Boot ...
- U-Boot ...
- U-Boot ...
- U-Boot ...
- U-boot ...
- U-boot ...

- Provides simultaneous operation of the USB 2.0 and USB 3.0 interfaces (only in host mode)
- 64-bit AXI master port with built-in DMA
- AXI port for register programming
- Power management features: hibernation mode
- Support for 48-bit address space

Note: U-boot does not support USB3.0