UUB device tree configuration

This document describes which files are safe to modify to get the functionality of the device (SPI, USB, etc) on UUB under Petalinux.

PetaLinux device tree configuration is associated with following config files that are located at `<plnx-projroot>/subsystems/linux/configs/device-tree/`.

Two device tree files used to build device tree are: `zynq-7000.dtsi` and `pcw.dtsi`

**Modifications for the file "zynq-7000.dtsi" for SPI0 and SPI1**

```dts
spio: spi@0xe0060000 {
    compatible = "xlnx,zynq-spi-r1p6";
    reg = <0xe006000 0x1000>;
    status = "okay";
    interrupt-parent = <&intc>;
    interrupts = <0 26 4>;
    clocks = <&clkc 25>, <&clkc 34>;
    clock-names = "ref_clk", "pelk";
    #address-cells = <1>;
    #size-cells = <0>;
    num-cs = <5>;
    is-decoded-cs = <1>;
}

spi: spi@0 {
    compatible = "spidev";
    reg = <0>;
    spi-max-frequency = <50000000>;
};

spi: spi@1 {
    compatible = "spidev";
    reg = <1>;
    spi-max-frequency = <50000000>;
};

spi: spi@2 {
    compatible = "spidev";
    reg = <2>;
    spi-max-frequency = <50000000>;
};

spi: spi@3 {
    compatible = "spidev";
    reg = <3>;
    spi-max-frequency = <50000000>;
};

spi: spi@4 {
    compatible = "spidev";
    reg = <4>;
    spi-max-frequency = <50000000>;
};
```
spi1: spi@e0007000 {
    compatible = "xlnx.zynq-spi-r1p6";
    reg = <0xe0007000 0x1000>;
    status = "disabled";
    interrupt-parent = <&intc>;
    interrupts = <0 49 4>;
    clocks = <&clk 26>, <&clk 35>;
    clock-names = "ref_clk", "pclk";
    #address-cells = <1>;
    #size-cells = <0>;
}
spidev@0 {
    compatible = "spidev";
    reg = <0>;
    spi-max-frequency = <50000000>;
};

Modifications of device tree for the file "zynq-7000.dtsi" for USB0

usb0: usb@e0002000 {
    compatible = "xlnx.zynq-usb-2.20a", "chipidea,usb2";
    status = "okay";
    clocks = <0x1 0x1c>;
    interrupt-parent = <0x3>;
    interrupts = <0x0 0x15 0x4>;
    reg = <0xe0002000 0x1000>;
    phy_type = "ulpi";
    dr_mode = "host";
    usb-phy = <0xb>;
};
phy0 {
    compatible = "usb-nop-xceiv";
    #phy-cells = <0>;
    reg = <0xe0002000 0x1000>;
    view-port = <0x0170>;
    drv-vbus;
    linux.phandle = <0xb>;
    phandle = <0xb>;
}

Modifications for the file "pcw.dtsi" for SPI0 and SPI1

&spi0 {
    is-decoded-cs = <1>;
    num-cs = <5>;
    status = "okay";
};

&spi1 {
    is-decoded-cs = <0>;
    num-cs = <1>;
    status = "okay";
}
Modification for petalinux-config

Deselect 'Device tree auto config' and run petalinux-build
SPI (Serial Peripheral Interface) busses

SPI busses can be described with a node for the SPI master device and a set of child nodes for each SPI slave on the bus. For this discussion, it is assumed that the system's SPI controller is in SPI master mode. This binding does not describe SPI controllers in slave mode.

The SPI master node requires the following properties:
- #address-cells - number of cells required to define a chip select address on the SPI bus.
- #size-cells - should be zero.
- compatible - name of SPI bus controller following generic names recommended practice.
- cs-gpios - (optional) gpios chip select.

No other properties are required in the SPI bus node. It is assumed that a driver for an SPI bus device will understand that it is an SPI bus. However, the binding does not attempt to define the specific method for assigning chip select numbers. Since SPI chip select configuration is flexible and non-standardized, it is left out of this binding with the assumption that board specific platform code will be used to manage chip selects. Individual drivers can define additional properties to support describing the chip select layout.

Optional property:
- num-cs : total number of chipselects

If cs-gpios is used the number of chip select will automatically increased with max(cs-gpios > hw cs)

So if for example the controller has 2 CS lines, and the cs-gpios property looks like this:

cs-gpios = <&gpio1 0 0> <0> <&gpio1 1 0> <&gpio1 2 0>;

is-decoded-cs : Flag to indicate whether decoder is used or not.

Then it should be configured so that num_chipselect = 4 with the following mapping:

cs0 : &gpio1 0 0
cs1 : native
cs2 : &gpio1 1 0
cs3 : &gpio1 2 0

SPI slave nodes must be children of the SPI master node and can contain the following properties.
- reg - (required) chip select address of device.
- compatible - (required) name of SPI device following generic names recommended practice
- spi-max-frequency - (required) Maximum SPI clocking speed of device in Hz
- spi-cpol - (optional) Empty property indicating device requires inverse clock polarity (CPOL) mode
- spi-cpha - (optional) Empty property indicating device requires shifted clock phase (CPHA) mode
- spi-cs-high - (optional) Empty property indicating device requires chip select active high
- spi-3wire - (optional) Empty property indicating device requires 3-wire mode.
- spi-lsb-first - (optional) Empty property indicating device requires LSB first mode.
- spi-tx-bus-width - (optional) The bus width(number of data wires) that used for MOSI Defaults to 1 if not present.
- spi-rx-bus-width - (optional) The bus width(number of data wires) that used for MISO. Defaults to 1 if not present.

Some SPI controllers and devices support Dual and Quad SPI transfer mode. It allows data in the SPI system to be transferred in 2 wires(DUAL) or 4 wires(QUAD). Now the value that spi-tx-bus-width and spi-rx-bus-width can receive is only 1(SINGLE), 2(DUAL) and 4(QUAD). Dual/Quad mode is not allowed when 3-wire mode is used.

If a gpio chipselect is used for the SPI slave the gpio number will be passed via the SPI master node cs-gpios property.

Cadence SPI controller Device Tree Bindings

Required properties:
- compatible : Should be "cdns.spi-r1p6" or "xlnx,zenq-spi-r1p6".
- reg : Physical base address and size of SPI registers map.
- interrupts : Property with a value describing the interrupt number.
- interrupt-parent : Must be core interrupt controller
- clock-names : List of input clock names - "ref_clk", "pclk"
  (See clock bindings for details).
- clocks : Clock phandles (see clock bindings for details).

Optional properties:
- num-cs : Number of chip selects used.
  If a decoder is used, this will be the number of chip selects after the decoder.
- is-decoded-cs : Flag to indicate whether decoder is used or not.