1 uboot 的板级配置 rk3568-evb.dts 添加 Pcie 相关节点

(1) 电源配置，主要是 pcie30_3v3 这个路，麻烦确认你们自己用的是哪根 gpio 来控制

```
pcie30_avdd0v9: pcie30-avdd0v9 {
        compatible = "regulator-fixed";
        regulator-name = "pcie30_avdd0v9";
        regulator-always-on;
        regulator-boot-on;
        regulator-min-microvolt = <900000>;
        regulator-max-microvolt = <900000>;
        vin-supply = <&vcc3v3_sys>;
};

pcie30_avdd1v8: pcie30-avdd1v8 {
        compatible = "regulator-fixed";
        regulator-name = "pcie30_avdd1v8";
        regulator-always-on;
        regulator-boot-on;
        regulator-min-microvolt = <1800000>;
        regulator-max-microvolt = <1800000>;
        vin-supply = <&vcc3v3_sys>;
};

pcie30_3v3: gpio-regulator {
        compatible = "regulator-gpio";
        regulator-name = "pcie30_3v3";
        regulator-min-microvolt = <100000>;
        regulator-max-microvolt = <3300000>;
        gpios = <&gpio0 RK_PD4 GPIO_ACTIVE_HIGH>;
        gpios-states = <0x1>;
        states = <100000 0x0
                  3300000 0x1>;
};
```

(2) 开启 phy 和控制器节点，尤其注意 reset-gpio 是配置#PERST 这个标准 IO 的，你们板子要自行确认是哪一根 IO。我这里的例子是 gpio2_d6。

```
&pcie30phy {
        status = "okay";
};

&pcie3x2 {
        reset-gpios = <&gpio2 RK_PD6 GPIO_ACTIVE_HIGH>;
        vpcie3v3-supply = <&pcie30_3v3>;
        status = "okay";
```

};

(3) Uboot 确认开启如下这些 config
CONFIG_DM_REGULATOR_GPIO=y
CONFIG_PCI=y
CONFIG_DM_PCI=y
CONFIG_DM_PCI_COMPAT=y
CONFIG_PCI_PNP=y
CONFIG_PCIE_DW_ROCKCHIP=y
CONFIG_PHY_ROCKCHIP_SNPS_PCIE3=y
CONFIG_PHY=y
CONFIG_CMD_PCI=y

(4)控制台启动 PCIe 扫描， 总线枚举识别到一个 Gen3 的 2 个 lane 的设备
=> pci enum
PCIe Linking... LTSSM is 0x1
PCIe Link up, LTSSM is 0x230011
PCIE-0: Link up (Gen3-x2, Bus0)

此程序其实调用的是 cmd/pci.c 文件的 pci_init()函数，实际函数定义在 drivers/pci/pci-uclass.c 中。如果希望自动运行，需要随便找个代码的地方，调用 pci_init()函数即可。

(4) 关于地址空间的问题

3568 按配置，给 FPGA 分配的 MEM 地址从 0x380900000 开始，也就意味着 FPGA 第一个可用 bar（一般是 bar0）的起始 MMIO 地址是 0x380900000， 3568 读写 0x3809000000 即可读写 FPGA 内部的内存。第二个 bar 的起始 MMIO 地址为 0x380900000 + 第一个 bar 所需的大小，依次类推。对于定下来的方案，这些都是已知信息，从 uboot 跳转到后续软件系统中，直接读写 0x3809000000 开始的地址，即可。 由于硬件资源类似内存，没有硬件互斥，后续多个 OS 自行协商各自访问的 MMIO 地址，防止越界。