

GPIO controllers on MPC8xxx SoCs

This is for the non-QE/CPM/GUTs GPIO controllers as found on 8349, 8572, 8610 and compatible.

Every GPIO controller node must have #gpio-cells property defined, this information will be used to translate gpio-specifiers.

Required properties:

- compatible : "fsl,<CHIP>-gpio" followed by "fsl,mpc8349-gpio" for 83xx, "fsl,mpc8572-gpio" for 85xx and "fsl,mpc8610-gpio" for 86xx.
- #gpio-cells : Should be two. The first cell is the pin number and the second cell is used to specify optional parameters (currently unused).
- interrupts : Interrupt mapping for GPIO IRQ.
- interrupt-parent : Phandle for the interrupt controller that services interrupts for this device.
- gpio-controller : Marks the port as GPIO controller.

Example of gpio-controller nodes for a MPC8347 SoC:

```

gpio1: gpio-controller@c00 {
    #gpio-cells = <2>;
    compatible = "fsl,mpc8347-gpio", "fsl,mpc8349-gpio";
    reg = <0xc00 0x100>;
    interrupts = <74 0x8>;
    interrupt-parent = <&ipic>;
    gpio-controller;
};

gpio2: gpio-controller@d00 {
    #gpio-cells = <2>;
    compatible = "fsl,mpc8347-gpio", "fsl,mpc8349-gpio";
    reg = <0xd00 0x100>;
    interrupts = <75 0x8>;
    interrupt-parent = <&ipic>;
    gpio-controller;
};

```

See booting-without-of.txt for details of how to specify GPIO information for devices.

To use GPIO pins as interrupt sources for peripherals, specify the GPIO controller as the interrupt parent and define GPIO number + trigger mode using the interrupts property, which is defined like this:

interrupts = <number trigger>, where:

- number: GPIO pin (0..31)
- trigger: trigger mode:
  - 2 = trigger on falling edge
  - 3 = trigger on both edges

Example of device using this is:

```

funkyfpga@0 {
    compatible = "funky-fpga";

```

8xxx\_gpio.txt

```
...  
interrupts = <4 3>;  
interrupt-parent = <&gpio1>;  
};
```