```
pegmatite_upc_reg: pegmatite-regulator@3 {
        compatible = "pegmatite-reg";
        reg = <0 0xf9080000 0 0x8>;
        init-on = <1>;
        clocks = <&xio_clkgate>, <&xcpu_clkgate>, <&ipsbus_upc_clkgate>;
        regulator-name = "pegmatite_upc";
        dev-name = "upc0";
        supply-name = "islandpower";
};
```

pegmatite-regulator@3 device node中的clocks是怎么作用到对应的

xio_clkgate

xcpu_clkgate

ipsbus_upc_clkgate

在pegmatite-regulator driver中会通过如下API来使能clock

struct clk *of_clk_get(struct device_node *np, int index);

这里的np即指向pegmatite_upc_reg device node.

in drivers/clk/clkdev.c

比如

clk_0 = of_clk_get(np, 0);

clk_0 return xio_clkgate对应的clock

clk_1 = of_clk_get(np, 1);

clk_1 return xcpu_clkgate对应的clock

clk_2 = of_clk_get(np, 2);

clk_2 return ipsbus_upc_clkgate对应的clock

pegmatite_reg_enable(rdev) ---> islandPowerUp(rdev)

在islandPowerUp()中对clock的操作如下

```
/*
* Power up procedure
*     -- Assert resets, poll
*     -- wait 100 us
*     -- turn on switches (2, 20us apart?), poll
*     -- wait 100 us
*     -- turn on clocks 10 us to propagate resets, then off
*     -- wait 100 us
*     -- un-isolate outputs, poll
*     -- wait 100 us
*     -- turn on island clocks
*     -- de-assert resets, poll
*/

if (!info->pmu_regs) {
```

```
        pr_emerg("Island base address not initialized\n");
        return -1;
}

pr_debug("%s:  o Assert island resets by disabling clocks\n", __func__);
for (i=0; i < info->num_clks; i++)
{
        if (__clk_is_enabled(info->island_clks[i]))
                clk_disable_unprepare(info->island_clks[i]);
}

udelay(100);
pr_debug("%s:  o Turn on island power switches\n", __func__);
regMaskWrite(info->pmu_regs + PICR_REG_OFFSET,
        PMU_BULKPWRUP_MASK,
        PMU_BULKPWRUP,
        POWER_RAMP);
udelay(20);
regMaskWrite(info->pmu_regs + PICR_REG_OFFSET,
        PMU_BULKPWRUP_MASK,
        PMU_BULKPWRUP,
        SWITCH_FULL_ON);

/* Poll until the status bit changed */
i = 0;
```

```
while((regRead(info->pmu_regs + PISR_REG_OFFSET) & PMU_BULKPWRUP_MASK)
        != PMU_BULKPWRUP_MASK) {
        i++;
        if(i == TIMEOUT) {
                pr_err("%s:  ! BULKPWRUP Timeout reached\n", __func__);
                break;
        }
}

udelay(100);

pr_debug("%s:  o Turn island clocks on/off to propagte resets\n", __func__);

for (i=0; i < info->num_clks; i++) {
        clk_prepare_enable(info->island_clks[i]);
}

udelay(10);

for (i=0; i < info->num_clks; i++) {
        clk_disable(info->island_clks[i]);
}

udelay(100);
```

```
pr_debug("%s:  o Un-isolate island outputs\n", __func__);
regMaskWrite(info->pmu_regs + PICR_REG_OFFSET,
        PMU_ISOLATION_MASK,
        PMU_ISOLATION,
        ISOLATION_DISABLE);

/* Poll until the status bit updated */
i = 0;
while((regRead(info->pmu_regs + PISR_REG_OFFSET) & PMU_ISOLATION_MASK)
        != PMU_ISOLATION_MASK) {
        i++;
        if(i == TIMEOUT) {
                pr_err("%s:  ! ISOLATION_BUF Timeout reached\n", __func__);
                break;
        }
}

udelay(100);

pr_debug("%s:  o Turn on island clocks\n", __func__);
for (i=0; i < info->num_clks; i++) {
        clk_enable(info->island_clks[i]);
}
```

clk_enable(struct clk *clk) ---> clk->ops->enable(clk->hw);

---> pegmatite/clkgate.c

```
const struct clk_ops pegmatite_clkgate_ops = {
        .is_enabled = pegmatite_clkgate_is_enabled,
        .enable = pegmatite_clkgate_enable,
        .disable = pegmatite_clkgate_disable,
};

static int pegmatite_clkgate_enable(struct clk_hw *hw)
{
        struct pegmatite_clkgate *gate = to_pegmatite_clkgate(hw);
        int val = readl(gate->config);

        /*
         * Set the enable bit
         */
        val |= CLK_EN_MASK;
        writel(val, gate->config);

        /*
         * If this is a clock with a reset, set that too
         */
        if(gate->reset) {
                val |= CLK_RESET_MASK;
                writel(val, gate->config);
```

```
        }


        return 0;

}
```