in pegmatite-clocks.dtsi

```
    preaudio_clk: preaudioclk {
        compatible = "marvell,pegmatite-clkgen";
        #clock-cells = <0>;
        reg = <0 0xf9080100 0 0x8>;
        prediv-shift = <25>;
        clocks = <&system_pll_gate>, <&ref_clk25mhz>;
        clock-source = <0>;
        max-divide = <30>;
        clock-frequency = <83333333>;
    };

    audio_clk: audioclk {
        compatible = "marvell,pegmatite-clkgen";
        #clock-cells = <0>;
        reg = <0 0xf9080108 0 0x8>;
        clocks = <&preaudio_clk>;
        max-divide = <60>;
        clock-frequency = <1536000>;
    };

    audio_clkgate: audioclkgate {
        compatible = "marvell,pegmatite-clkgate";
        #clock-cells = <0>;
        reg = <0 0xf9080108 0 0x8>;
        clocks = <&audio_clk>;
    };
```

in audio-ddac.c

```
    clk = clk_get(&pdev->dev, NULL);    ①
    if(IS_ERR(clk))
    {
        err = PTR_ERR(clk);
        dev_dbg(&pdev->dev, "fail to get clock!\n");
        goto err_handling_3;
    }

    if(clk_prepare_enable(clk))
    {
        dev_dbg(&pdev->dev, "fail to prepare clock!\n");
        clk_put(clk);
        goto err_handling_2;
    }
```

①
这里获取的clk就是pegmatite-clocks.dtsi中的audio_clkgate.

preaudio_clk有两个father，但真正工作时只能是选择其一。

> clock-source = <0>;

clock-source就用于选择哪个father。

```
    ref_clk25mhz: clk25mhz {
        compatible = "fixed-clock";
        #clock-cells = <0>;
        clock-frequency = <25000000>;
    };

    ......

    system_pll: systempll {
        compatible = "marvell,pegmatite-pll";
        #clock-cells = <0>;
        reg = <0 0xd0621800 0 0x200>;
        clocks = <&ref_clk25mhz>;
        clock-frequency = <2500000000>;
    };

    system_pll_gate: systempllgate {
        compatible = "marvell,pegmatite-clkgate";
        #clock-cells = <0>;
        reg = <0 0xd0627018 0 0x8>;
        clocks = <&system_pll>;
        always-used;
    };
```

ref_clk25mhz是源头，可能是晶振吧，固定提供25M HZ的clock。system_pll锁相环则把频率提高了100倍，达到了2.5G HZ,可能也
是提供给core的clock吧。

system_pll_gate的功能就是提供可以对system_pll开关(gate)的功能，但这里使用了always-used   property

> always-used;

其实就等于gate总是打开的。

in drivers/clk/pegmatite/clkgate.c

```c
static void __init of_pegmatite_clkgate_setup(struct device_node *node)
{
    ......

    always_used  = of_property_read_bool(node, "always-used");

    clk = clk_register(NULL, &gate->hw);
    if(WARN_ON(IS_ERR(clk)))
        goto map_out;

    if (always_used) {
        clk_prepare_enable(clk);
    }

    of_clk_add_provider(node, of_clk_src_simple_get, clk);

    ......
}
```

这样到preaudio_clk的时候，如果

> clocks = <&system_pll_gate>, <&ref_clk25mhz>;
>
> clock-source = <0>;

则选择的是2.5G HZ的system_pll。

如果

> clocks = <&system_pll_gate>, <&ref_clk25mhz>;
>
> clock-source = < 1 >;

则选择的是25M HZ的ref_clk25mhz。

---

preaudio_clk和audio_clk都用来分频，即降低频率。

preaudio_clk

6270 Programmer Guide有如下描述

> The maximum frequency for this clock is **625 MHz**. It is important to ensure that
>
> the configuration settings for this clock result in an output frequency less
>
> than this value.

in drivers/clk/pegmatite/clkgen.c/of_pegmatite_clkgen_setup() function

```
    /*
     * clock-frequency can be set to enable a default clock rate for the clock
     */
    if (of_property_read_u32(node, "clock-frequency", &default_rate)) {   ①
        default_rate = 0;
    }

    clk_base = of_iomap(node, 0);
    if(WARN_ON(!clk_base))
        goto free_out2;

    init->name = kasprintf(GFP_KERNEL, "%s", node->name);
    init->ops = &pegmatite_clkgen_ops;
    init->flags = 0;
    parent_clk = of_clk_get(node, gen->clock_source);         ②
    parent_name = __clk_get_name(parent_clk);                 ③
    init->parent_names = &parent_name;                        ④
    init->num_parents = 1;

    gen->hw.init = init;
    gen->config = clk_base;
    val = readl(gen->config);                                 ⑤
    val &= ~(SRCSEL_MASK << SRCSEL_SHIFT);
    val |= (gen->clock_source & SRCSEL_MASK) << SRCSEL_SHIFT;
    writel(val, gen->config);

    clk = clk_register(NULL, &gen->hw);
    if(WARN_ON(IS_ERR(clk)))
        goto map_out;

    of_clk_add_provider(node, of_clk_src_simple_get, clk);

    /*
     * If a default rate was specified in the device tree, set it here
     * If this clock can be gated, setting the default rate does not ungate it
     */
    if(default_rate > 0)                                      ⑥
        clk_set_rate(clk, default_rate);
```

①
对应dts中的

> clock-frequency = <83333333>;

②

```c
    /*
     * Some clocks have multiple possible clock sources
     */
    if (of_property_read_u32(node, "clock-source", &gen->clock_source)) {
        gen->clock_source = 0;
    }
```

那么这里就是

> parent_clk = of_clk_get(node, 0);

③

> clocks = <&system_pll_gate>, <&ref_clk25mhz>;

这里取得的clock name就是system_pll_gate的name。

④
虽然有两个father，但当前正工作的只有一个，其中一个。

⑤
下面几行code就是去设置SRCSEL bit

> Clock Source Select
>
> Selects the source for the clock generator
>
> 0: System   PLL Clock
>
> 1: 25MHz reference clock

⑥

> clk_set_rate(clk,83333333);