

# 改变CPU运行频率

regs.h:提供寄存器地址

common.h: 定义读取设置位的函数及类型宏定义

## pll.c

- void wait\_to\_lock(u32 \*pll\_reg) 等待指定的PLL进入锁定状态
- void set\_pll(pll\_e pll, u32 div) 设置PLL的倍频参数并等待其进入锁定状态
- u32 get\_pll(pll\_e pll) 获取PLL的输出频率
- static void set\_pfd(u32 \*reg, pfd\_e pfd, int gate, u32 frac) PFD参数设置
- void set\_pll\_pfd(pll\_e pll, pfd\_e pfd, int gate, u32 frac) 设置SYS\_PLL或USB1\_PLL的PFD状态和分频参数
- u32 get\_pll\_pfd(pll\_e pll, pfd\_e pfd) 获取SYS\_PLL或USB1\_PLL的PFD输出频率

## switcher.c

- void sel\_pll1\_sw\_clk(int sel\_pll1) 设置PLL1\_SW\_CLK的时钟路径
- u32 get\_pll1\_sw\_clk(void) 获取PLL1\_SW\_CLK的时钟频率
- u32 get\_pll3\_sw\_clk(void) 获取PLL3\_SW\_CLK的时钟频率
- u32 get\_pll4\_main\_clk(void) 获取PLL4\_MAIN\_CLK的时钟频率
- u32 get\_pll5\_main\_clk(void) 获取PLL5\_MAIN\_CLK的时钟频率

## clkroot.c

- void setup\_arm\_podf(u32 podf) 设置ARM\_CLK\_ROOT的分频参数
- u32 get\_arm\_clk\_root(void) 获取ARM\_CLK\_ROOT的频率
- static u32 get\_periph\_clk(void) 获取PERIPH\_CLK的频率
- u32 get\_ahb\_clk\_root(void) 获取AHB\_CLK\_ROOT的频率
- u32 get\_ipg\_clk\_root(void) 获取IPG\_CLK\_ROOT的频率
- u32 get\_axi\_clk\_root(void) 获取AXI\_CLK\_ROOT的频率
- u32 get\_fabric\_mmdc\_clk\_root(void) 获取FABRIC\_MMDC\_CLK\_ROOT的频率

## led.c

- void led\_init(void) led初始化GPIO
- void led\_on(void) 开灯
- void led\_off(void) 关灯
- void led\_toggle(void) 取反

## main.c

大致流程举例

- sel\_pll1\_sw\_clk(0); /\* 将ARM\_ROOT时钟切换至OSC \*/
- setup\_arm\_podf(8); /\* ARM\_ROOT的分频参数设置为8 \*/
- set\_pll(ARM\_PLL, 54); /\* 设置ARM\_PLL: 24\*54/2 = 648MHZ, ARM\_ROOT: 81MHZ \*/
- sel\_pll1\_sw\_clk(1); /\* 将ARM\_ROOT切换回ARM\_PLL, 此时CPU工作频率为81MHZ \*/