

概述

在实际应用中，有时需要确定某个时钟的具体数值，需要将时钟引到芯片管脚上，以便使用示波器进行测量。本文将描述 PLL 的频率产生过程及将 PLL 时钟引到管脚上进行测量的具体方法。

SPIN TROL

目录

1	PLL 时钟的产生过程.....	6
2	时钟测量.....	8

SPIN TROL

图片列表

图 1-1: PLL 结构图.....6

SPIN TROL

版本历史

版本	日期	作者	状态	变更
A/0	2023-06-23	CanChai	Outdated	首次发布。
C/0	2024-03-26	Jiali Zhou	Released	修改排版格式。

SPIN TROL

术语或缩写

术语或缩写	描述
/	/

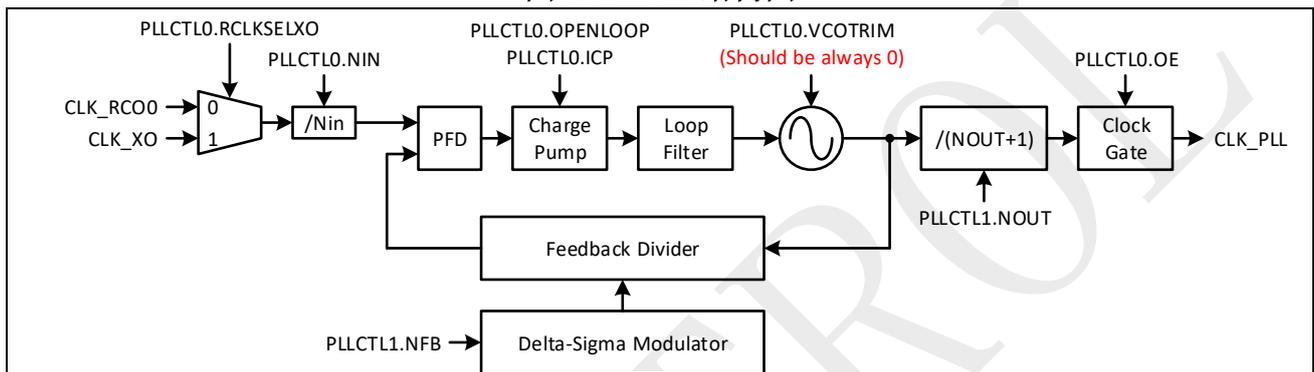
SPIN TROL

1 PLL 时钟的产生过程

PLL 模块的整体结构图如图 1-1 所示，从 RCO 或晶体振荡器作为输入的参考时钟，当环路被锁定时，VCO 频率将在 400MHz 到 600MHz 之间，最后通过可编程分频器进行分频，以提供通常为 25MHz 到 200MHz 的时钟，输出频率由以下公式给出：

$$f_{PLL} = \frac{f_{REF}}{NIN} \times \frac{NFB}{65536} \times \frac{1}{NOUT + 1}$$

图 1-1: PLL 结构图



想要得到某个 PLL 频率，需要按照如下步骤进行计算：

– 输入参考和分频

PLL 的输入参考时钟（ f_{in} ）为 4MHz 到 56MHz，最终送至鉴频鉴相器 PFD 的输入（ f_{pfd} ）为从 4MHz 到 8MHz。因此，输入分频比（ f_{in}/f_{pfd} ）应当如表 1-1 所示配置。

表 1-1: 输入分频比设置

输入频率 (MHz)	输入分频比
4~8	1
8~16	2
16~24	3
24~32	4
32~40	5
40~48	6
48~56	7

– 输出分频

VCO 的调频范围为 400MHz 到 600MHz，结合输出分频，最终时钟 CLK_PLL 的频率如表 1-2 所示。

表 1-2: 输出时钟频率和分频比

输出频率	分频比 (NOUT+1)
400MHz ~ 600MHz	1
200MHz ~ 300MHz	2
150MHz ~ 200MHz	3
100MHz ~ 150MHz	4

75MHz	~	100MHz	6
50MHz	~	75MHz	8
37.5MHz	~	50MHz	12
25MHz	~	37.5MHz	16

- 环路分频

基于由表 1-1 和表 1-2 得到的 NIN 和 $NOUT$ ，PLLCTL1 寄存器的 NFB 位段的值可以根据下面公式计算：

$$NFB = \frac{f_{PLL} \times (NOUT + 1)}{f_{REF} / NIN} \times 65536$$

通常使用 RCO0 作为时钟源，此时 f_{REF} 为 32000000， NIN 为 4，此时 f_{REF} / NIN 为 8000000

$$\begin{aligned} NFB &= \frac{f_{PLL} \times (NOUT + 1)}{8000000} \times 65536 \\ &= f_{PLL} \times (NOUT + 1) \times \frac{128}{15625} \end{aligned}$$

在使用整型计算时，为避免溢出，也会写成下面形式：

$$= \frac{f_{PLL} \times (NOUT + 1)}{25} \times \frac{128}{625}$$

2 时钟测量

可以将芯片内部时钟信号（RCO 或 PLL）分频后送出到 GPIO 进行观察，示例代码将 PLL 通过 GPIO 引出。

注意： 因 GPIO 翻转能力限制，分频后频率控制在 2M 以下。

Example Code

```
int main(void)
{
    /* Set Flash timing */
    FLASH_WALLOW();

    FLASH_SetTiming(200000000);
    /* Disable flash write access after flash operation had done */
    FLASH_WDIS();

    CLOCK_InitWithRCO(CLOCK_HCLK_200MHZ);

    /* Delay init */
    Delay_Init();

    /* UART init */
    GPIO_SetPinChannel(GPIO_34, GPIO34_UART_TXD);
    GPIO_SetPinChannel(GPIO_35, GPIO35_UART_RXD);

    UART_Init(UART, 38400);

    printf("Just a Sample...\n");

    GPIO_SetPinChannel(GPIO_11, GPIO11_CLK_DCLK);
    CLOCK->CLKDETCTL.bit.DCLKSEL = CLKDETCTL_BIT_DCLKSEL_PLL;
    CLOCK->CLKDETCTL.bit.DCLKDIV = CLKDETCTL_BIT_DCLKDIV_64;
    CLOCK->CLKDETCTL.bit.EN = CLKDETCTL_BIT_EN_ENABLE;

    while (1)
    {
    }
}
```