CLOCK 使用指南

概述

本手册适用范围：

适用范围	
SPC1125 系列	SPC1125，SPC1128
SPC1168 系列	SPC1155， SPC1156， SPC1158， SPC1168，SPD1148， SPD1178， SPD1188， SPD1163，SPM1173
SPC2168 系列	SPC2168，SPC2165，SPC2166，SPC1198
SPC1169 系列	SPC1169，SPD1179，SPD1176
SPC2188 系列	SPC1185，SPC2188

目录

图片列表

表格列表

版本历史

版本	日期	作者	状态	变更
C/0	2024-08-08	Hang Su	Released	1. 首次发布。

术语或缩写

术语或缩写	描述
MCU	Microcontroller Unit，微控制器单元

1 概述

通过内部 RC 振荡器、环形振荡器或 PLL 驱动时钟。它还可以通过外部振荡器或连接到芯片上振荡器电路的晶振来提供时钟。

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。

2 功能描述

2.1 PLL

PLL 模块的整体结构图如图 2-1 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 CLK_MAX 的时钟。

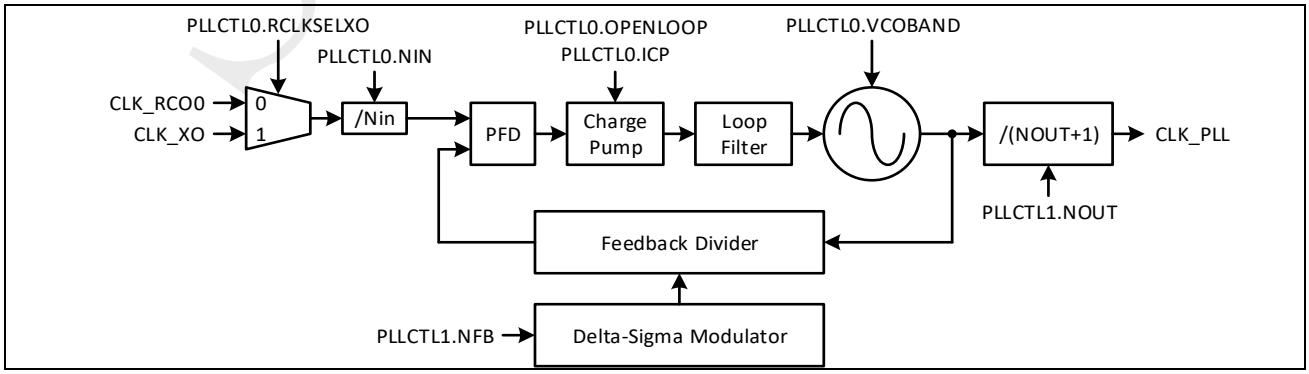
表 2-1：最大系统时钟

MCU	CLK_MAX
SPC1169，SPD1179，SPD1176	100 MHz
SPC2188，SPC1185	240 MHz
SPC1125，SPC1128	125 MHz
SPC2168，SPC2166，SPC1198	200 MHz
SPC1155，SPC1156，SPC1168，SPD1148，SPD1178，SPD1188，SPD1163，SPM1173	200 MHz
SPC1158，SPC1168L	100 MHz

输出频率由以下公式给出：

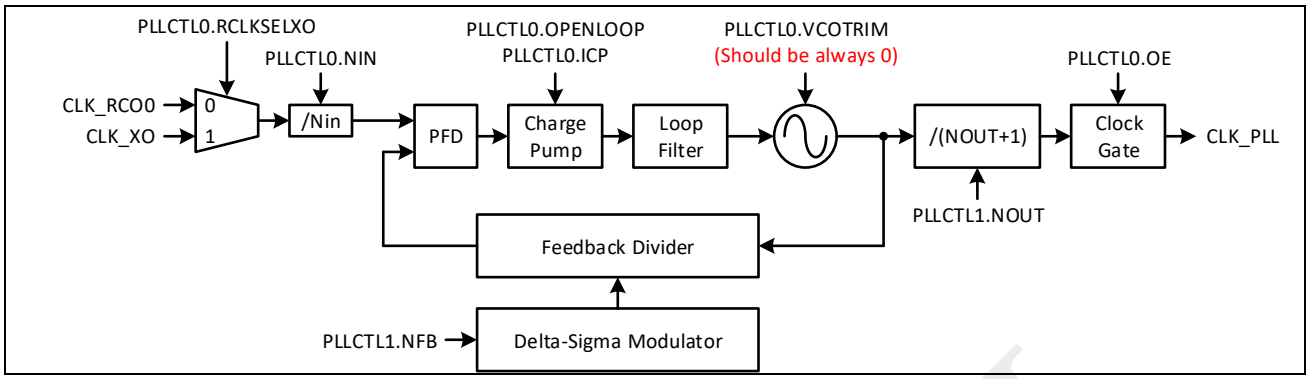
$$f_{PLL} = \frac{f_{REF}}{NIN} \times \frac{NFB}{65536} \times \frac{1}{NOUT + 1}$$

图 2-1：PLL 结构图



[1] SPC2188 系列，SPC1169 系列，SPC1128 系列的 PLL 均为此结构。

图 2-2：PLL 结构图



[1] SPC1168 系列，SPC2168 系列的 PLL 均为此结构。

想要得到某个 PLL 频率，需要按照如下步骤进行计算：

- 选择 NIN 值：PLL 的设计中规定其参考时钟的频率范围为 4MHz~56MHz，而输入到 PFD 子单元的时钟频率应该为 4MHz~8MHz，因此根据不同的参考时钟频率，我们需要设置不同的 NIN 数值，以便得到合理的f_{PFD}数值，手册提供了不同参考频率时需要设置的 NIN 数值，如表 2-2 所示，以便用户快速查阅。
- 设置 NOUT 值：从f_{PLL}计算公式中可以看出，影响最终f_{PLL}的变量过多，不利于计算，所以需要换一个思路来思考怎样计算f_{PLL}的最终值。上文提到环路被锁定时，VCO 的频率将在 400MHz~600MHz，由于 NFB 在反馈环路里，比较难确定，所以先确定 NOUT 的数值，如表 2-3 所示，列出了对应不同输出频率时需要设置的 NOUT 数值。
- 设置 NFB 值：在已经确定 NIN 及 NOUT 数值的情况下，可以推导出 NFB 的计算公式为：

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

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

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

在使用整型计算时，为避免溢出，也会写成下面形式：

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

表 2-2：参考时钟分频设置

参考时钟（MHz）	参考时钟分频（NIN = f_{REF}/f_{PFD}）
4~8	1
8~16	2
16~24	3
24~32	4
32~40	5
40~48	6
48~56	7

表 2-3：输出时钟分频设置

期望输出频率			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

3 功能实例

3.1 时钟引出到外部

3.1.1 功能需求

将时钟引出到外部。

3.1.2 功能实现

要想通过示波器测量时钟，那就需要将时钟信号接到芯片 IO 上，这里需要注意一个问题，由于 IO 的能力限制，时钟频率不能过高，需要对其进行分频处理，将其分频到 2M 内，然后再使用示波器进行测量。

Example Code

```
#include "spd1179.h"
#include <stdio.h>

int main(void)
{
    CLOCK_InitWithRCO(CLOCK_CPU_100MHZ);

    Delay_Init();

    /*
     * Init the UART
     */
    PIN_SetChannel(PIN_GPIO10, PIN_GPIO10_UART0_TXD);
    PIN_SetChannel(PIN_GPIO11, PIN_GPIO11_UART0_RXD);
    UART_Init(UART0, 38400);

    printf("Enter the test\n");

    PIN_SetChannel(PIN_GPIO2, PIN_GPIO2_CLKMON);
    /* 7:128 div */
    WRITE_FIELD(CLOCK->CLKDETCTL,CLKDETCTL_DCLKDIV_Msk,CLKDETCTL_DCLKDIV_Pos, 7);
    /* 0:RCO0, 1: RCO1, 2: XO, 3: PLL*/
    WRITE_FIELD(CLOCK->CLKDETCTL,CLKDETCTL_DCLKSEL_Msk,CLKDETCTL_DCLKSEL_Pos, 3);
    CLOCK_EnableDetection();

    while (1)
    {
    }
}
```

[1] 示例代码适用于 SPD1179。

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)
    {
    }
}
```

[1] 示例代码适用于 SPC1168。

Example Code

```
#include "spc2168.h"
#include <stdio.h>



int main(void)
{
  /* Set Flash timing */
  FLASH_WALLOW();
  FLASH_SetTiming(200000000);
  FLASH_WDIS();

  /* Clock init */
  CLOCK_InitWithRCO(200000000UL);

  /* Delay init */
  Delay_Init();


  /* UART init */
  GPIO_SetPinChannel(GPIO_44, GPIO44_UART_TXD);
  GPIO_SetPinChannel(GPIO_45, GPIO45_UART_RXD);
```

Example Code

```
  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)
  {
  }
}
```

[1] 示例代码适用于 SPC2168，其它系列产品的示例代码会根据实际需求进行补充。