

MSP430系统时钟 (UCS)

Unified Clock System

本节内容

- 时钟系统模块设计要求
- **MSP430X5XX / 6XX**系列时钟系统模块
- 五个时钟输入源振荡器模块
- **DCO**模块操作
- 外设模块请求时钟系统（低功耗运行模式下）
- 模块振荡器（**MODOSC**）
- 故障安全逻辑操作
- 时钟模块应用举例（**MSP430F5XX / 6XX**）
- 时钟模块库函数

时钟系统模块设计要求（1/1）

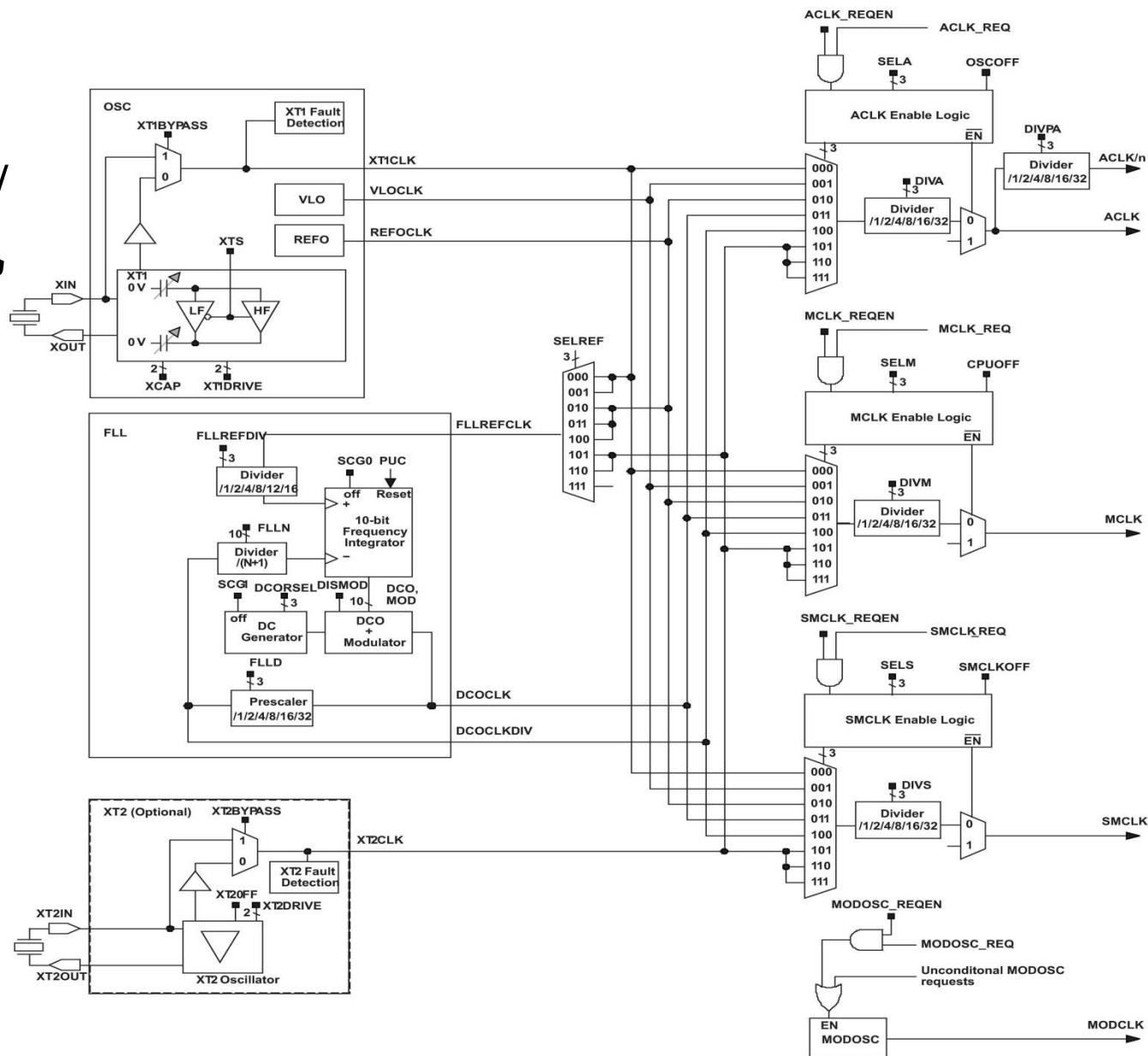
单片机各部件能有条不紊自动工作，实际上是在其系统时钟作用下，控制器指挥芯片内各个部件自动协调工作，使内部逻辑硬件产生各种操作所需的脉冲信号而实现的。

为适应系统和具体应用需求，单片机的系统时钟必须满足以下不同要求：

- **高频率** 用于对系统硬件需求和外部事件快速反应；
- **低频率** 用于降低电流消耗；
- **稳定的频率** 以满足定时应用，如实时时钟RTC；
- **低Q值振荡器** 用于保证开始及停止操作最小时间延迟。

MSP430X5XX / 6XX系列时钟系统模块 (1/3)

右图为
MSP430X5XX /
6XX系列单片机
时钟模块结构:



MSP430X5XX / 6XX系列时钟系统模块 (2/3)

从上图可以看出，MSP430F5XX / 6XX时钟模块有 5 个时钟输入源：

- **XT1CLK 低频或高频时钟源：** 可以使用标准晶振，振荡器或者外部时钟源输入4MHz ~ 32MHz。XT1CLK可以作为内部FLL模块的参考时钟。
- **XT2CLK 高频时钟源：** 可以使用标准晶振，振荡器或者外部时钟源输入4MHz ~ 32MHz。
- **VLOCLK 低功耗低频内部时钟源：** 典型值为10KHZ；
- **REFOCLK 低频修整内部参考时钟源：** 典型值为32768Hz，作为FLL基准时钟源；
- **DCOCLK 片内数字控制时钟源：** 通过FLL模块来稳定。

MSP430X5XX / 6XX系列时钟系统模块 (3/3)

基础时钟模块可提供3种时钟信号：

➤ **ACLK 辅助时钟：** ACLK可由软件选择来自XT1CLK、REFOCLK、VLOCLK、DCOCLK、DCOCLKDIV、XT2CLK（由具体器件决定）这几个时钟源之一。然后经1、2、4、8、16、32分频得到。ACLK可由软件选作各个外设模块的时钟信号，一般用于低速外设模块。

➤ **MCLK 系统主时钟：** MCLK可由软件选择来自上述5种时钟源，同样可经过分频得到。MCLK主要用于CPU和系统。

➤ **SMCLK 子系统时钟：** 可由软件选择来自上述5种时钟源，同样可经过分频得到。SMCLK可由软件选作各个外设模块的时钟信号，主要用于高速外设模块。

五个时钟输入源振荡器模块（1/6）

五个时钟输入源振荡器模块，包括：

- **XT1 振荡器**

- **XT2 振荡器**

- **低功耗低频内部振荡器（VLO）**

- **低频修整内部参考振荡器（REFO）**

- **片内数字控制振荡器（DCO）**

五个时钟输入源振荡器模块 (2/6)

一、XT1 振荡器

- XT1工作在低频 (LF) 模式时 (XTS=0) , 提供支持32768HZ时钟的超低功耗模式。晶振只需经过XIN和XOUT两个引脚连接, 不需要其他外部器件, 所有保证工作稳定的元件和移相电容都集成在芯片中。
- 在一些设备中当XT1选择高频 (HF) 模式时 (XTS=1) 也支持高频晶振或者振荡器。高频晶振或谐振器连接到XIN和XOUT引脚, 需要在两个端口配置电容。

五个时钟输入源振荡器模块 (3/6)

二、XT2 振荡器

- 一般称之为第二振荡器XT2，它产生时钟信号XT2CLK，它的工作特性与XT1振荡器工作在高频模式时类似。
- 系统频率和系统的工作电压密切相关，某些应用需要较高的工作电压，所以也需要系统提供相应较高的频率。系统频率和系统工作电压之间的关系下图所示：

五个时钟输入源振荡器模块 (4/6)

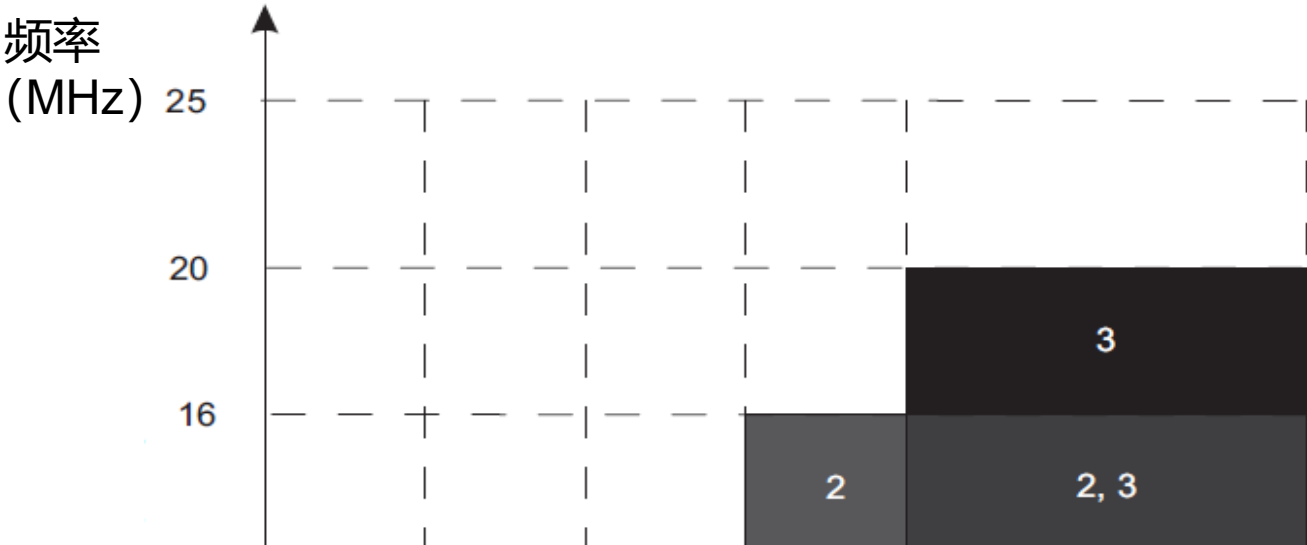


Table 2-16. PMMCTL0 Register Description

Bit	Field	Type	Reset	Description
15-8	PMMPW	RW	96h	PMM password. Always read as 096h. When using word operations, must be written with 0A5h or a PUC is generated. When using byte operation, writing 0A5h unlocks all PMM registers. When using byte operation, writing anything different than 0A5h locks all PMM registers.
7	Reserved	RW	0h	Reserved. Must always be written as 0.
6-5	Reserved	R	0h	Reserved. Always reads as 0.
4	PMMREGOFF	RW	0h	Regulator off (see the SYS chapter for details)
3	PMMSWPOR	RW	0h	Software power-on reset. Setting this bit to 1 triggers a POR. This bit is self clearing.
2	PMMSWBOR	RW	0h	Software brownout reset. Setting this bit to 1 triggers a BOR. This bit is self clearing.
1-0	<u>PMMCOREV</u>	RW	0h	Core voltage (see the device-specific data sheet for supported levels and corresponding voltages) 00b = V(CORE) level 0 01b = V(CORE) level 1 10b = V(CORE) level 2 11b = V(CORE) level 3

→ 压 (V)

在

五个时钟输入源振荡器模块 (5/6)

三、低功耗低频内部振荡器 (VLO)

- 低频低功耗内部振荡器 (VLO)能够提供典型10kHz的振荡频率（具体参数见数据手册），而不需要外接任何晶振。
- VLO可以对时钟精确要求不高的应用提供低成本和超低功耗的时钟源。

五个时钟输入源振荡器模块 (6/6)

四、低频修整内部参考振荡器 (REFO)

➤ REFO可以产生一个比较稳定的频率，其典型值为32768Hz，它可以用作FLLREFCLK。

➤ 低频修整内部参考振荡器 (REFO) 可以在没有外部晶振，对成本又比较敏感の場合得到很好的应用。

五、片内数字控制振荡器 (DCO)

➤ DCO振荡器是一个可数字控制的RC振荡器，它的频率随供电电压、环境温度变化而具有一定的不稳定性。

➤ DCO频率可以通过选择锁频环FLL的频率 $FLLRENCLK/n$ 来增强振荡频率的稳定性。

片内数字控制振荡器 (DCO) 操作

锁频环(FLL)

- DCOCLK可用作ACLK、MCLK、SMCLK，但它的频率随供电电压、环境温度变化而具有一定的不稳定性，FLL通过频率积分器和调制器的自动调节使DCOCLK的频率趋于稳定。
- FLL通过两个最相近的邻居频率之间进行切换，产生两个频率的加权频率，最终获得我们所需的频率。

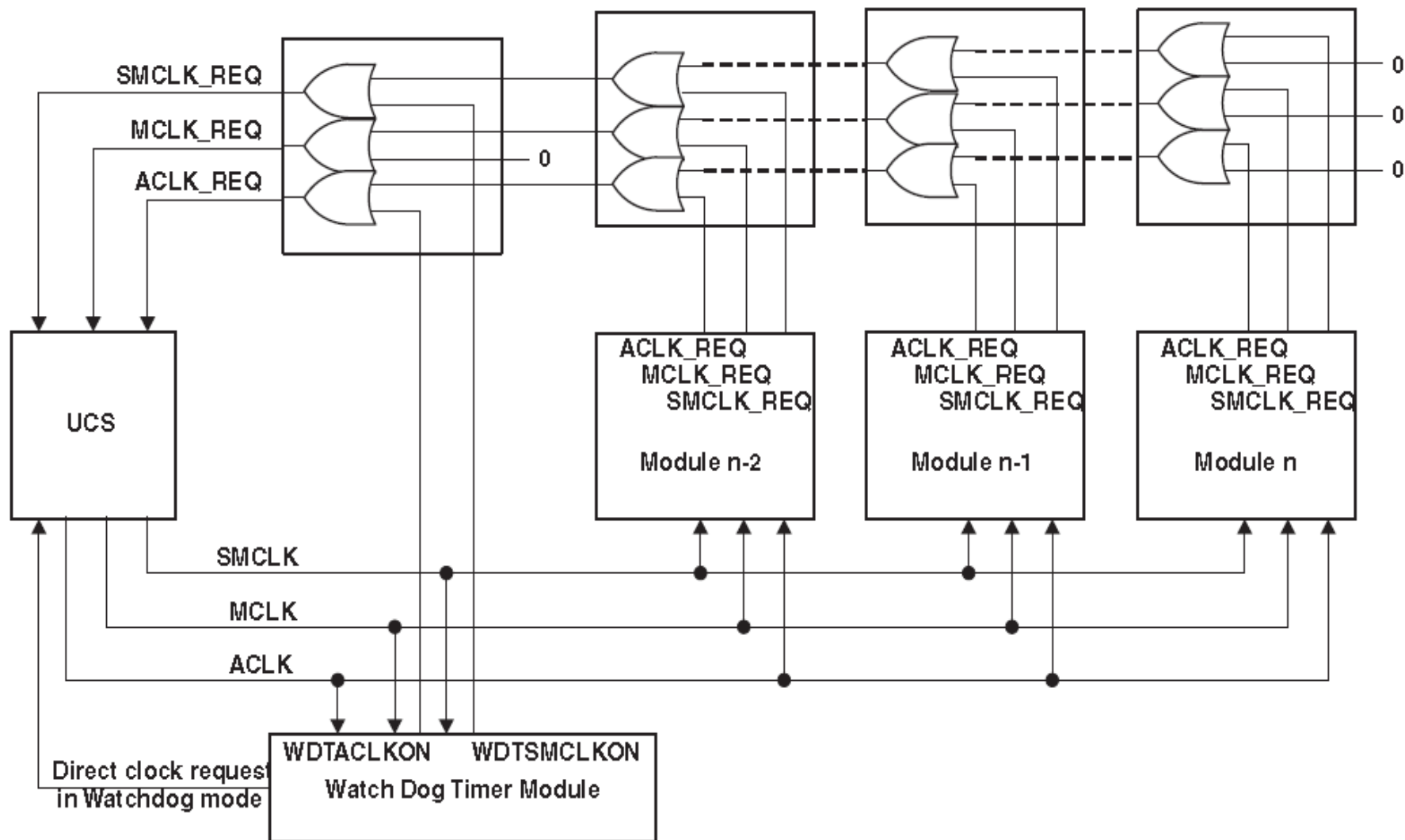
外设模块请求时钟系统 (1/2)

- ▶ 外设模块可以控制3个时钟请求信号中的一个来获得时钟 ACLK_REQ、MCLK_REQ、SMCLK_REQ。
- ▶ 不管在任何模式下，外设模块的正常操作都可以从标准时钟系统（UCS）请求时钟信号。

例如：如果定时器选择了ACLK作为时钟源，只要定时器允许，ACLK_REQ信号就一直有效并向UCS申请时钟，而UCS则不管当前是在什么LPM低功耗模式都会输出ACLK信号。

如下图所示，为外设模块请求时钟系统。

外设模块请求时钟系统 (2/2)



外设模块请求时钟系统

模块振荡器 (MODOSC)

➤ UCS模块还有一个内部的振荡器 (MODOSC) 。它主要给FLASH模块控制器或其他任意需要的模块提供时钟。

➤ MODOSC产生时钟信号MODCLK。

例：ADC12_A可以选择使用MODOSC作为转换时钟源，用户选择ADC12OSC作为转换时钟源时，ADC12OSC就来自MODOSC。

故障安全逻辑操作

- ▶ 时钟系统模块包含有晶振故障保护的功能。这个功能可以检测XT1、XT2、DCO的振荡器故障。
- ▶ 当晶体振荡器启用后，没有正常工作时，则相应的故障位XT1LFOFFG、XT1HFOFFG、XT2OFFG将被置位。
- ▶ 可检测的故障有：
XT1的LF模式下低频晶振故障（XT1LFOFFG）

1.14.2 SFRIFG1 Register

Interrupt Flag Register



Figure 1-9. SFRIFG1 Register

15	14	13	12	11	10	9	8
Reserved							
r0	r0	r0	r0	r0	r0	r0	r0
7	6	5	4	3	2	1	0
JMBOUTIFG	JMBINIFG	Reserved	NMIIFG	VMAIFG	Reserved	OFIFG ⁽¹⁾	WDTIFG ⁽²⁾
rw-(1)	rw-(0)	r0	rw-0	rw-0	r0	rw-(1)	rw-0

(1) See the UCS chapter for details.
(2) See the WDT_A chapter for details.

系统时钟模块寄存器

用户手册USER GUIDE P172

5.4	UCS Module Registers	172
5.4.1	UCSCTL0 Register	173
5.4.2	UCSCTL1 Register	174
5.4.3	UCSCTL2 Register	175
5.4.4	UCSCTL3 Register	176
5.4.5	UCSCTL4 Register	177
5.4.6	UCSCTL5 Register	178
5.4.7	UCSCTL6 Register	180
5.4.8	UCSCTL7 Register	182
5.4.9	UCSCTL8 Register	183
5.4.10	UCSCTL9 Register	184
1.14	SFR Registers	83
1.14.1	SFRIFG1 Register	84
1.14.2	SFRIFG1 Register	85
1.14.3	SFRIFG1 Register	87

系统时钟模块寄存器

5.4.5 UCSCTL4 Register

Unified Clock System Control 4 Register

Figure 5-10. UCSCTL4 Register

15	14	13	12	11	10	9	8
Reserved					SELA		
r0	r0	r0	r0	r0	rw-0	rw-0	rw-0
7	6	5	4	3	2	1	0
Reserved	SELS			Reserved	SELM		
r0	rw-1	rw-0	rw-0	r0	rw-1	rw-0	rw-0

5.4.7 UCSCTL6 Register

Unified Clock System Control 6 Register

Figure 5-12. UCSCTL6 Register

15	14	13	12	11	10	9	8
XT2DRIVE		Reserved	XT2BYPASS	Reserved			XT2OFF
rw-1	rw-1	r0	rw-0	r0	r0	r0	rw-1
7	6	5	4	3	2	1	0
XT1DRIVE		XTS	XT1BYPASS	XCAP		SMCLKOFF	XT1OFF
rw-1	rw-1	rw-0	rw-0	rw-1	rw-1	rw-0	rw-1

5.4.8 UCSCTL7 Register

Unified Clock System Control 7 Register

Figure 5-13. UCSCTL7 Register

15	14	13	12	11	10	9	8
Reserved		Reserved		Reserved		Reserved	
r0	r0	rw-0	rw-(0)	rw-(1)	rw-(1)	r-1	r-1
7	6	5	4	3	2	1	0
Reserved			Reserved	XT2OFFG ⁽¹⁾	XT1HFOFFG ⁽¹⁾	XT1LFOFFG	DCOFFG
r0	r0	r0	rw-(0)	rw-(0)	rw-(0)	rw-(1)	rw-(1)

时钟模块应用举例 (MSP430F5529) (1/2)

例，MSP430F5529演示例程：设ACLK = XT1 = 32768Hz，ACLK通过P1.0输出。

程序代码如下：

```
#include <msp430.h>
```

```
void main(void)
```

```
{
```

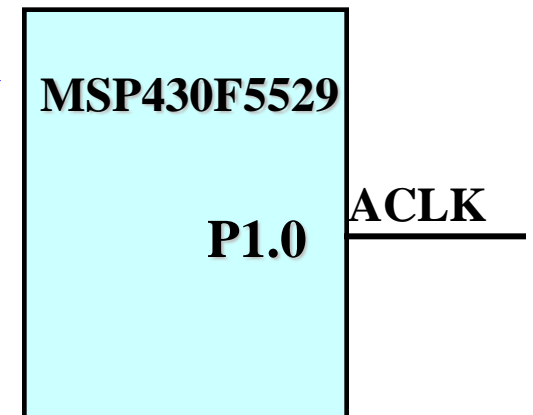
```
    WDTCTL = WDTPW + WDTHOLD; // 关闭看门狗
```

```
    P1DIR |= BIT0; // ACLK 通过 P1.0输出，输出方向
```

```
    P1SEL |= BIT0; // P1.0 为外设功能（ACLK）
```

```
    // 选择IO端口功能为 XT1，外设功能
```

```
    P5SEL |= BIT4+BIT5;
```



时钟模块应用举例 (MSP430F5529) (2/2)

```
UCSCTL6 &= ~(XT1OFF); // 使能 XT1, 设置寄存器UCSCTL6
UCSCTL6 |= XCAP_3;    // 配置内接电容值, 选择典型电容值XCAP_3
```

```
do
{
    // 清零XT1、DCO故障标志位
    UCSCTL7 &= ~( XT1LFOFFG + DCOFFG);
    SFRIFG1 &= ~OFIFG;           // 清零SFR中的故障标志位
}while (SFRIFG1&OFIFG);        // 检测振荡器故障标志位
```

```
UCSCTL4 |= SELA_0;    // 选择 ACLK的时钟源, 设置寄存器UCSCTL4
```

```
while(1)
{
    ;                               // 循环等待,空操作
}
```

```
}
```

时钟模块库函数 (1/3)

◆ 时钟系统(UCS)的API被分成三组函数:

- 时钟系统(UCS)常用配置和初始化的API函数
- 外部晶振特定的配置和初始化的API函数
- 对状态和配置进行设置和询问的API函数

◆ 时钟系统(UCS)常用配置和初始化的API函数

- UCS_clockSignalInit()
- UCS_initFLLSettle()
- UCS_enableClockRequest()
- UCS_disableClockRequest()
- UCS_SMCLKOff()
- UCS_SMCLKOn()

时钟模块库函数 (2/3)

◆ 外部晶振特定的配置和初始化的**API**函数

- UCS_setExternalClockSource()
- UCS_LFXT1Start()
- UCS_HFXT1Start()
- UCS_bypassXT1()
- UCS_LFXT1StartWithTimeout()
- UCS_HFXT1StartWithTimeout()
- UCS_bypassXT1WithTimeout()
- UCS_XT1Off()
- UCS_XT2Start()
- UCS_XT2Off()
- UCS_bypassXT2()
- UCS_XT2StartWithTimeout()
- UCS_bypassXT2WithTimeout()
- UCS_clearAllOscFlagsWithTimeout()

时钟模块库函数 (3/3)

◆对状态和配置进行设置和询问的**API**函数

➤UCS_faultFlagStatus()

➤UCS_clearFaultFlag()

➤UCS_getACLK()

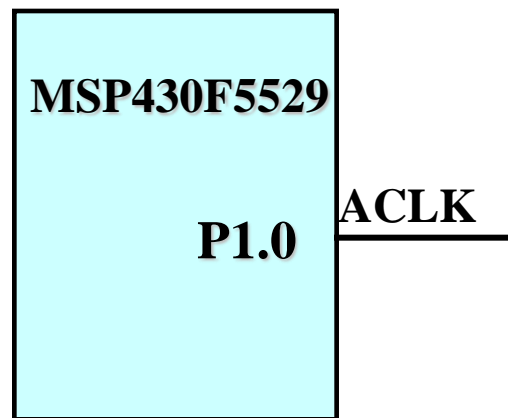
➤UCS_getSMCLK()

➤UCS_getMCLK()

系统时钟实验任务

- 课上实验1:

MSP430F5529演示例程：设 $ACLK = XT1 = 32768\text{Hz}$ ， $ACLK$ 通过P1.0输出。



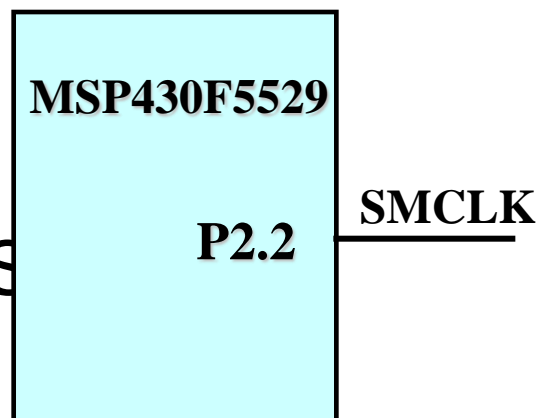
要求：编程，用示波器查看P1.0引脚输出时钟信号，是否为硬件接线中XT1晶振的频率32768Hz，结果给老师查看。

系统时钟实验作业

- 作业1:

参考课上系统时钟例程，自己查找硬件接线原理图，找到XT2晶振XT2IN和XT2OUT的GPIO复用引脚，同时找到SMCLK时钟输出的GPIO复用引脚。

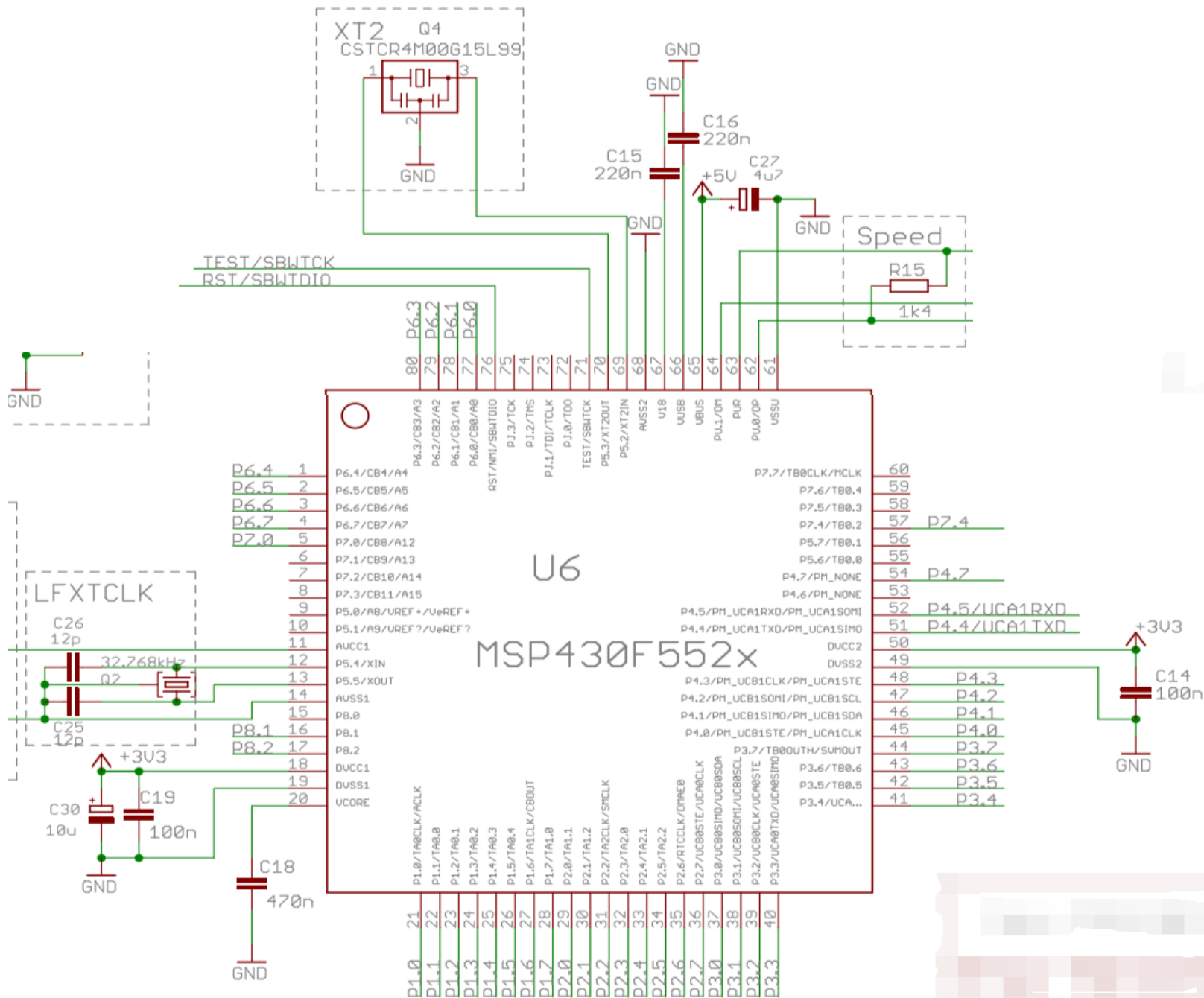
硬件接线：SMCLK = XT2 = 4MHz，S
过P2.2输出。

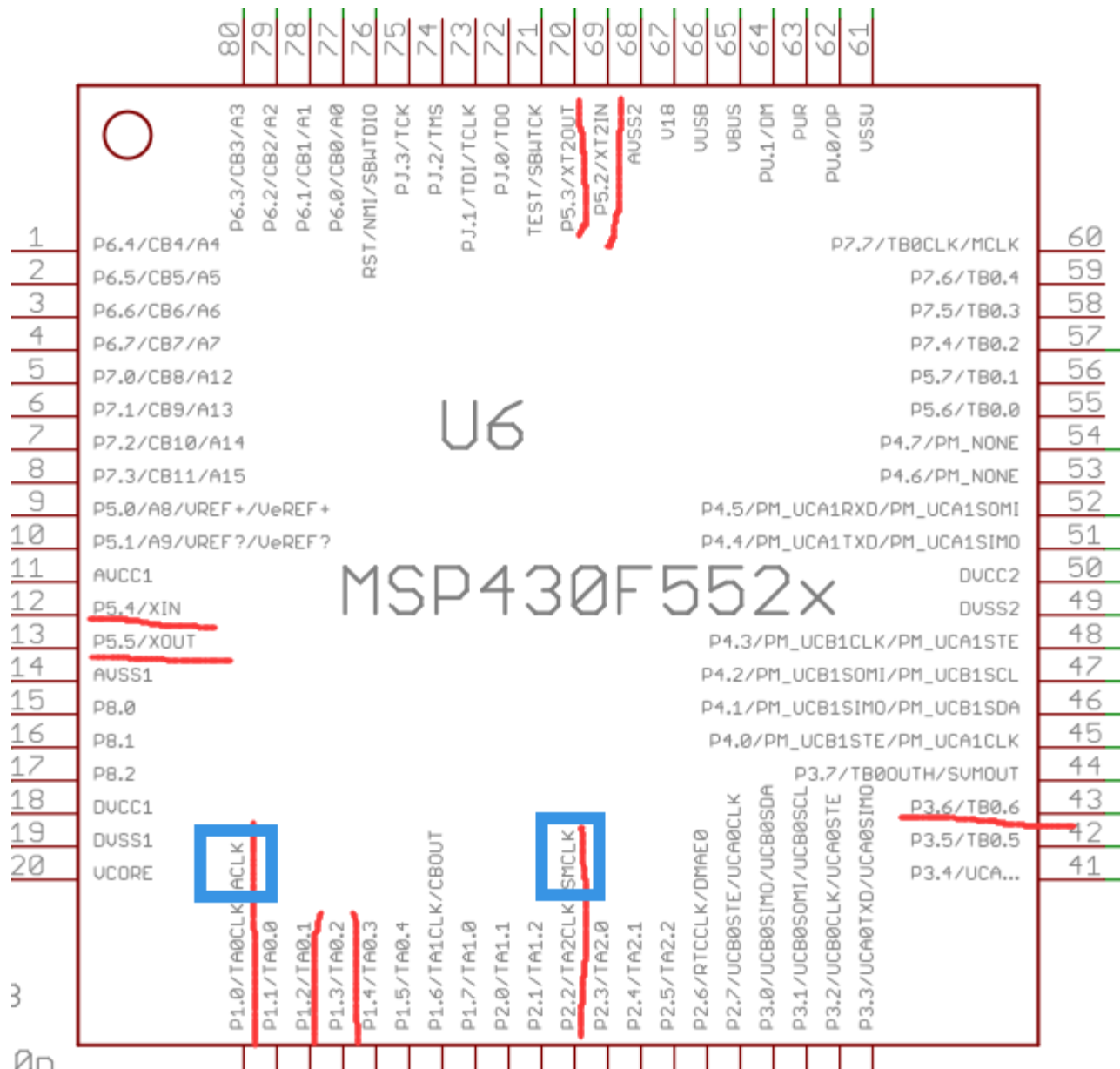


要求：编程，使SMCLK的源为外部晶振的4MHz，用示波器查看P2.2引脚输出时钟信号，是否为硬件接线中晶振的频率4MHz，如果不是，回答

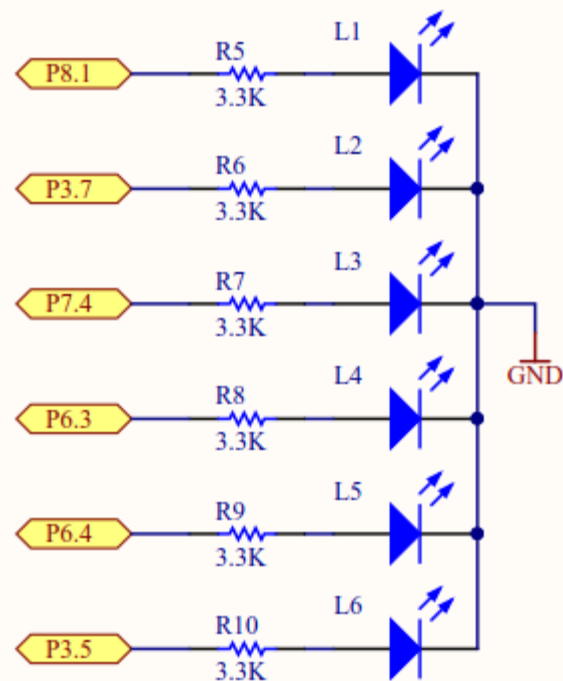
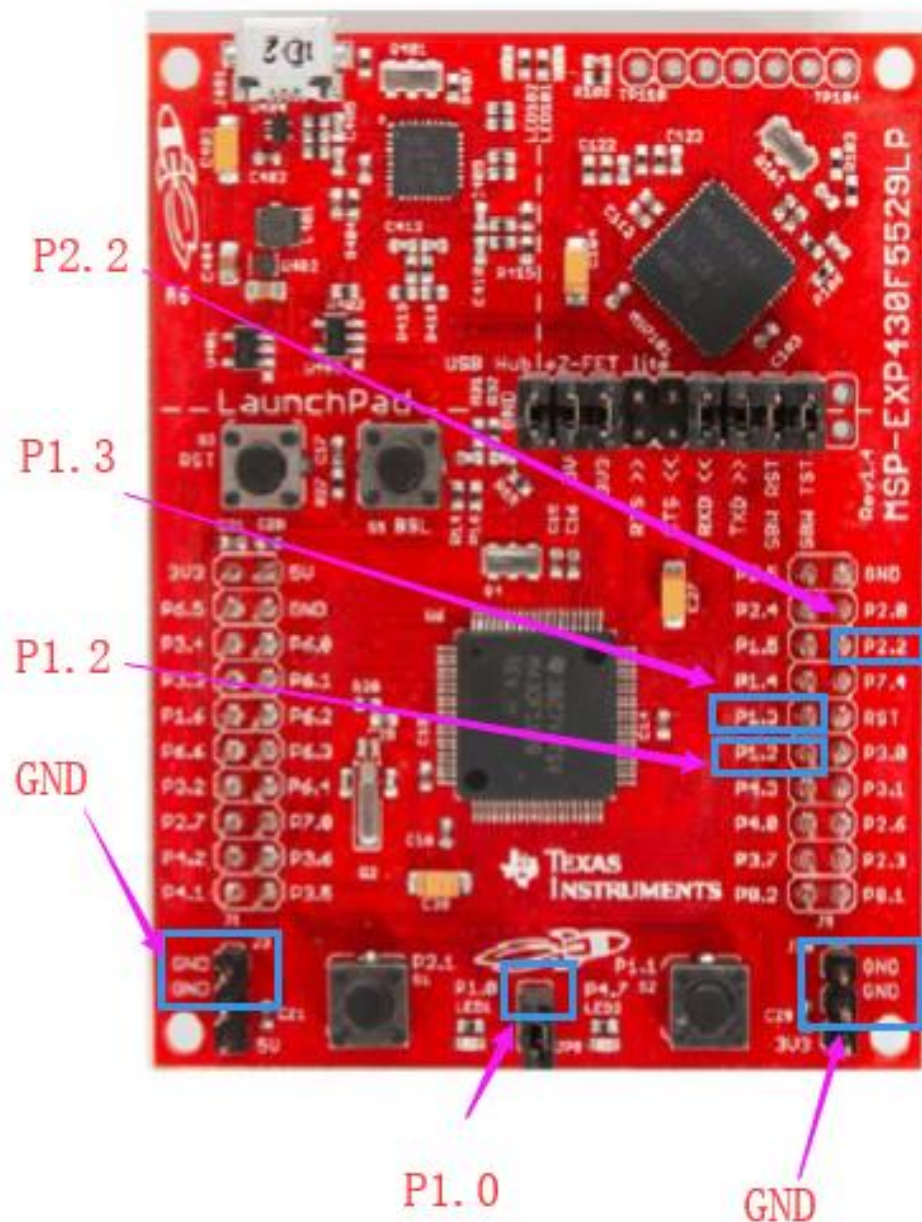
(1) 频率是多少？

(2) 为什么会出现此现象？

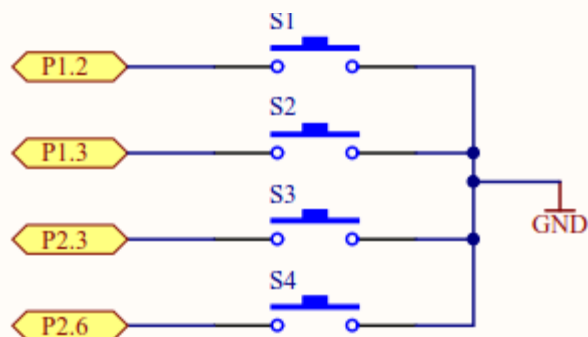




接线部分引脚



LED



USER Key