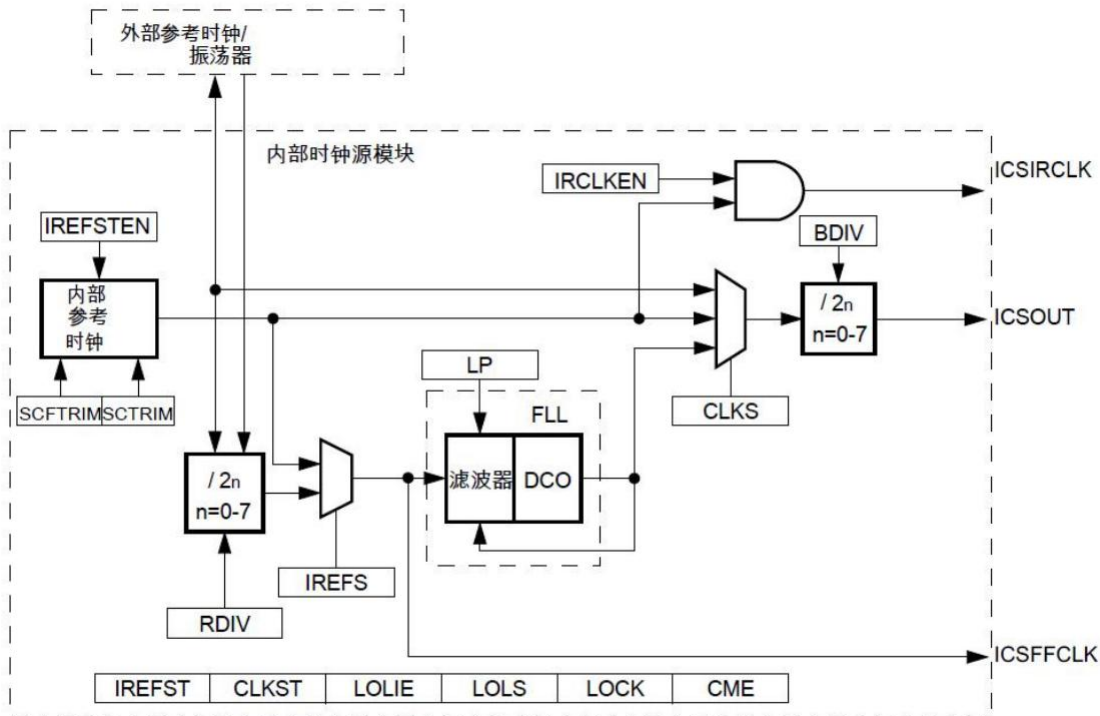


## 内部时钟源模块

### 1 工作模式

内部时钟源模块为 MCU 提供时钟源选择。



由以下三个寄存器

- (1) ICS\_C1[IREF]，选择内部时钟源，还是选择外部时钟源。
- (2) ICS\_C1[CLKS]，选择内部时钟还是外部时钟。
- (3) ICS\_C2[LP]，选择关闭内部 FLL 进入低功耗模式

组合得到以下几个时钟模式：

FEI：内部 IRC 作为参考时钟源，FLL 工作

FEE：外部 OSC 作为参考时钟源，FLL 工作

FBI：跳过 FLL，内部 IRC 直接作为时钟，FLL 照常工作

FBE：跳过 FLL，使用外部 OSC 作为时钟，FLL 照常工作

FBILP：FBI 情况下关闭 FLL

FBELP：FBE 情况下关闭 FLL

FBE\_OSC：跳过内部倍频，用外部输入频率作为时钟（有源）；

FEE\_OSC：外部输入时钟作为时钟源，FLL 工作,和 FEE 的区别是 OSC 不等 4096 个稳定周期，直接启动。

使用内部 IRC 作为参考时钟源，ICSOUT 频率输出由 ICS\_TRIM 和 BDIV 来决定，ICS\_TRIM 值是用来修调内部 IRC,  $F_{ICSOUT} = (F_{IRC} * 1280) / BDIV$ 。典型值:BDIV=0, 不分频的情况下 (0x25 48M) (0x4c 40M)

使用外部 OSC 或者输入频率作为参考时钟源，ICSOUT 频率输出由 RDIV 以及 BDIV 来决定，RDIV 是有要求的，FOSC/RDIV 必须在 30~46k 之间，例如外部为 8M，那么 RDIV 只能为 256

#### 1.1 FLL 内部启用 (FEI)

该模式为默认模式，在该模式下，ICS 提供一个来源 FLL 的时钟，FLL 的时钟由内部参考时钟控制。FLL 环路将频率锁定到内部参考频率的 1280 倍。该模式寄存器配置如下：

1. 00b 写入 ICS\_C1[CLKS]
2. 1b 写入 ICS\_C1[IREFS]

内部时钟源——片上 RC 振荡器，范围为 31.25 - 39.0625kHz，作为 FLL 输入基准。

表 1-1 基于内部参考的可行的 ICS 总线输出频率

基准		ICSOUTCLK
FEI	BDIV=0	40MHz~50MHz <sup>1</sup>
	BDIV=1	20MHz~25MHz
	BDIV=2	10MHz~12.5MHz
	BDIV=4	5MHz~6.25MHz
	BDIV=8	2.5MHz~3.125MHz
	BDIV=16	1.25MHz~1.5625MHz
	BDIV=32	625kHz~781.25kHz
	BDIV=64	312.5kHz~390.625kHz
	BDIV=128	156.25kHz~195.3125kHz

## 1.2 FLL 外部使用模式（FEE）

该模式下，ICS 提供一个源于 FLL 的时钟，该 FLL 时钟由外部参考时钟控制。FLL 循环将 FLL 频率锁定到外部基准频率（ICS\_C1[RDIV]和 OSC\_CR[RANGE]选择）的 1280 倍。该模式寄存器配置如下：

1. 00b 写入 ICS\_C1[CLKS]
2. 0b 写入 ICS\_C1[IREFS]
3. 对 ICS\_C1[RDIV]和 OSC\_CR[RANGE]进行操作可将外部时钟分频到 31.25~39.0625kHz 范围内。参考时钟的分频结果必须在 26k~45k 范围内。

该模式下 ICS 输出时钟频率计算如下：

例，当把 1b 写入 OSC\_CR[RANGE]，100b 写入 ICS\_C1[RDIV]，0b 写入 ICS\_C2[BDIV]，外部时钟晶振选择为 16MHz；

则外部基准频率=16000/512=31.25k

ICS 输出时钟频率=31.25\*1280=40M

## 1.3 FLL 内部旁路（FBI）

该模式下，FLL 使能并通过内部参考时钟来控制，但处于旁路状态。ICS 提供从内部参考时钟分频而来的时钟，参看表 1-1。FLL 循环将 FLL 频率锁定到内部参考频率的 1280 倍。该模式寄存器的配置如下：

1. 01b 写入 ICS\_C1[CLKS]
2. 1b 写入 ICS\_C1[IREFS]

## 1.4 FLL 内部旁路低功耗（FBILP）

该模式下，FLL 禁用并旁路，ICS 提供从内部参考时钟分频而来的时钟，参看表 1-1。该模式下寄存器的配置如下：

1. 01b 写入 ICS\_C1[CLKS]
2. 1b 写入 ICS\_C1[IREFS]
3. 1b 写入 ICS\_C2[LP]

## 1.5 FLL 外部旁路（FBE）

该模式下，FLL 使能并通过外部参考时钟来控制，但处于旁路状态。ICS 提供从外部参考时钟分频而来的时钟。FLL 循环将 FLL 频率锁定到外部基准频率（ICS\_C1[RDIV]和 OSC\_CR[RANGE]选择）的 1280 倍。该模式寄存器的配置如下：

1. 10b 写入 ICS\_C1[CLKS]
2. 0b 写入 ICS\_C1[IREFS]
3. 对 ICS\_C1[RDIV]和 OSC\_CR[RANGE]进行操作可将外部时钟分频到 31.25~39.0625kHz 范围内。参考时钟的分频结果必须是 26k~40k。

## 1.6 FLL 外部旁路低功耗（FBELP）

该模式下，FLL 禁用并旁路，ICS 提供从外部参考时钟分频而来的时钟。该模式下寄存器的配置如下：

1. 10b 写入 ICS\_C1[CLKS]
2. 0b 写入 ICS\_C1[IREFS]
3. 1b 写入 ICS\_C2[LP]

## 1.7 FLL 停止模式（STOP）

该模式下 FLL 禁用，内部或 ICS 外部基准时钟源(OSC\_OUT)是使能还是禁用可以选型。ICS 不提供任何 MCU 时钟源。只要 MCU 进入 STOP 状态，就会进入停止模式。在该模式下，所有 ICS 时钟信号都保持静态，但例外情况如下：

出现以下所有条件时，ICSIRCLK 在停止模式下无效：

1. 1b 写入 ICS\_C1[IRCLKEN]。
2. 1b 写入 ICS\_C1[IREFSTEN]。

注意：DCO 频率从预停止值变为其复位值，FLL 需要在频率稳定前重新获取锁定。时序敏感的操作在执行前，必须等待 FLL 获取时间 tAcquire

## 1.8 模式切换

ICS\_C1[IREFS]可以随时更改，但实际切换到新选择的时钟是由 ICS\_S[IREFST]指示。在 FLL 内部启用(FEI) 和 FLL 外部启用(FEE) 模式之间切换时，FLL 在切换完成后再次开始锁定。

ICS\_C1[CLKS]也可以随时更改，但实际切换到新选择的时钟是由 ICS\_S[CLKST]指示。如果新选择的时钟不可用，那么仍会选择之前的时钟。

注意：从 FEE、FBE 或 FBELP 切换到 FEI 模式时，建议等待 IREFST 切换完成，然后再更改 ICS\_C1[CLKS]。

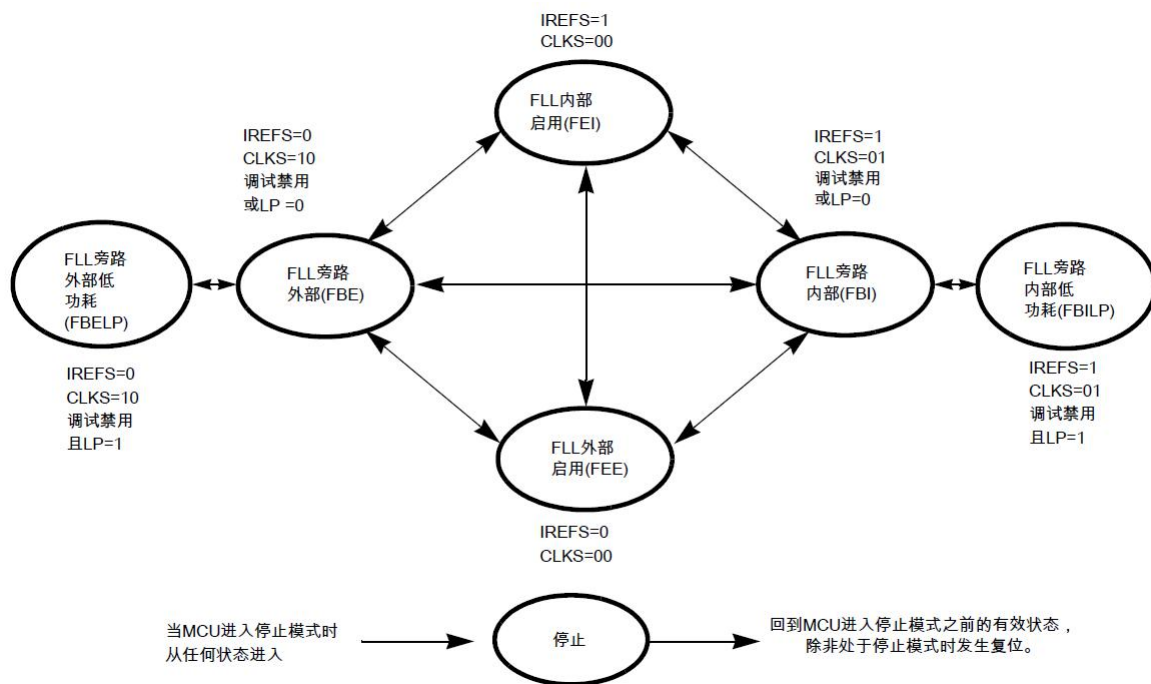
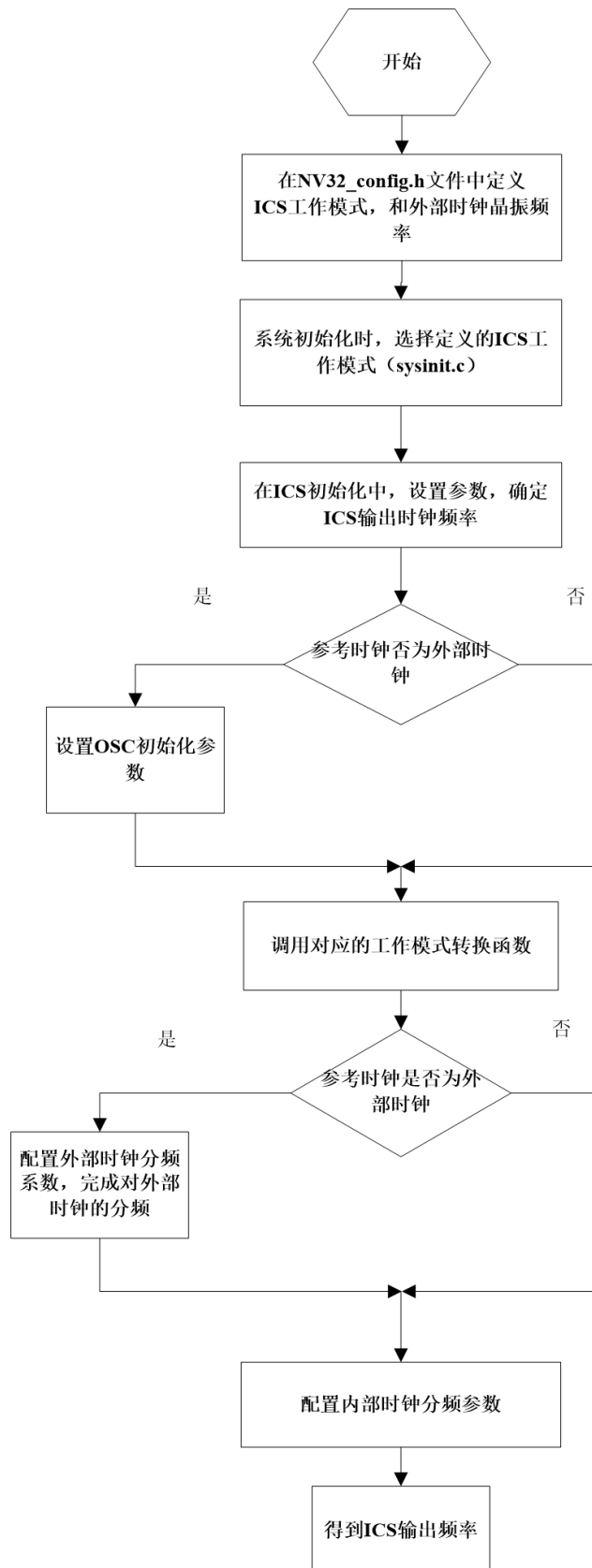


图 1-1 时钟切换模式

## 2.1 ICS 时钟模式的设置



ICS 系统初始化时默认的模式为 FEI 模式, 可根据对寄存器的配置实现 ICS 工作模式的转换。ICS 各模式之间的切换关系如图 1-1 所示。现以配置 ICS 的工作模式由 FEI 模式转换成 FEE 模式为例。其他模式间

的准换与此类似。

配置 FEE 模式的具体操作具体如下：

1. 在首先在 NV32\_Cconfig.h 文件中选择定义外部是时钟（#define USE\_FEE）和外部时钟的晶振频率（#define EXT\_CLK\_FREQ\_KHZ 10000）；
2. 系统初始化时（sysinit.c），系统会根据选择定义的时钟模式，初始化 ICS 模块为定义的工作模式。
3. 当 ICS 初始化为 FEE 模式时，首先是 OCS 模块的初始化设置。在 OCS 模块初始化设置中，当写 0b 到 ICS\_CR[RANGE] 时外部晶振频率为 32K 的低频范围，当写 1b 到 ICS\_CR[RANGE] 时，外部晶振频率范围在 4MHz~48MHz。
4. 经过系统的初始化，ClkFreqKHz= EXT\_CLK\_FREQ\_KHZ，分频函数 ICS\_SetClkDivider() 将对外部时钟晶振频率进行分频，对外部时钟的分频结果必须在 26K~46k 之间，因此分频系数的选择 ICS\_C1\_RDIV(x) 的选择要要根据自己定义的外部时钟晶振频率决定。

例如当选择的时钟晶振频率为 10Mhz 时，对其进行 256 倍时钟分频，分频结果； $10000/256 = 39.0625K$  外部时钟其他晶振频率的分频，可在分频函数中进行配置：

```
void ICS_SetClkDivider(uint32_t u32ClkFreqKHz)
```

```
{
```

```
    switch(u32ClkFreqKHz)
```

```
    {
```

```
        case 8000L:
```

```
        case 10000L:
```

```
            ICS->C1 = (ICS->C1 & ~(ICS_C1_RDIV_MASK)) | ICS_C1_RDIV(3); //8MHz 分频 8000/256 = 31.25K
```

```
                                //10MHz 分频 10000/256 =
```

```
39.0625K
```

```
            break;
```

```
        case 4000L:
```

```
            ICS->C1 = (ICS->C1 & ~(ICS_C1_RDIV_MASK)) | ICS_C1_RDIV(2); // 4MHz 分频 4000/128 = 31.25K
```

```
            break;
```

```
        case 16000L:
```

```
            ICS->C1 = (ICS->C1 & ~(ICS_C1_RDIV_MASK)) | ICS_C1_RDIV(4); //16MHz 分频 16000/512 = 31.25K
```

```
            break;
```

```
        case 20000L:
```

```
            ICS->C1 = (ICS->C1 & ~(ICS_C1_RDIV_MASK)) | ICS_C1_RDIV(4); //20MHz 分频 20000/512 =
```

```
39.0625K
```

```
            break;
```

```
        case 32L:
```

```

        ICS->C1  &= ~(ICS_C1_RDIV_MASK); //30k
        break;
    default:
        break;
    }
}

```

5. ICS 初始化最后是模式的转换；由 FEI 模式转化成 FEE 模式，在转换函数中通过设定内部时钟的分频系数确定 ICS 输出的时钟频率；

例：通过对 10MHz 的外部时钟分频得到分频结果为：10000/256=39.0625k；

内部时钟源分频系数为 2；则 ICS 输出时钟频率为： 39.0625/2=19.5313k

FEI 模式转换 FEE 模式函数如下；

```
void FEI_to_FEE(ICS_ConfigType *pConfig)
```

```
{
```

```
    OSC_Init(&pConfig->oscConfig); /* 使能 OSC 模块 */
```

```
    ICS_SetClkDivider(pConfig->u32ClkFreq); //对外部时钟进行分频
```

```
    ICS->C1 = ICS->C1 & ~ICS_C1_IREFS_MASK; //选择外部时钟源
```

```
#if defined(IAR)
```

```
asm(
```

```
"nop \n"
```

```
"nop \n"
```

```
);
```

```
#elif defined(__MWERKS__)
```

```
asm{
```

```
nop
```

```
nop
```

```
};
```

```
#endif
```

```
while(ICS->S & ICS_S_IREFST_MASK); //等待参考时钟变为外部时钟
```

```
while(!(ICS->S & ICS_S_LOCK_MASK)); //等待选择 FLL 时钟
```

```
/*对所选钟源做 2 分频，通过改变 ICS_C2_BDIV(X)，x 的数值实现分频参数设置*/
```

```
#if defined(CPU_NV32)
```

```
if(((ICS->C2 & ICS_C2_BDIV_MASK)>>5) != 1) //读取 ICS_CS 寄存器的值，判断当前是否为 2 分频
```

```
{
```

```

ICS->C2 = (ICS->C2 & ~(ICS_C2_BDIV_MASK)) | ICS_C2_BDIV(1); 对所选时钟进行 2 分频
}
#else
    ICS->C2 = (ICS->C2 & ~(ICS_C2_BDIV_MASK)) | ICS_C2_BDIV(0);对所选时钟进行 1 分频
#endif
ICS->S |= ICS_S_LOLS_MASK;//读取 loss of Lock 状态
}

```

## 2.2 常用外部时钟设置参考

外部晶振	目标频率	RDIV	FLL 输出 频率	BDIV	ICS 工作 模式
4M	40M	128	40M	0x00	FEE
4M	10M	128	40M	0x02	FEE
4M	5M	128	40M	0x03	FEE
4M	4M	128	40M	0x00	FBE
4M	4M	0	0	0x00	FBE_LP
8M	20M	256	40M	0x01	FEE
9.6M	48M	256	48M	0x00	FEE
9.6M	24M	256	48M	0x01	FEE
10M	50M	256	50M	0x00	FEE
12M	60M	256	60M	0x00	FEE*
48M	48M	0	0	0x00	FBE_LP

### \*不建议使用

注意：设置好 RDIV 和 BDIV 需要将 BUS\_CLK\_HZ 改成你算出来的值

例如：

```
#if defined(USE_FEI)
    #define BUS_CLK_HZ 48000000L //ics trim 值为 0x25 RDIV=0,BDIV=0
#elif (EXT_CLK_FREQ_KHZ == 20000)
    #define BUS_CLK_HZ 50000000L //RDIV=512 BDIV=0x00
#elif (EXT_CLK_FREQ_KHZ == 12000)
    #define BUS_CLK_HZ 15000000L //RDIV=512 BDIV=0x01
#elif (EXT_CLK_FREQ_KHZ == 8000)
    #define BUS_CLK_HZ 40000000L //RDIV=256 BDIV=0x00
#elif (EXT_CLK_FREQ_KHZ == 4000)
    #define BUS_CLK_HZ 40000000L //RDIV=128 BDIV=0x00
#elif (EXT_CLK_FREQ_KHZ == 32)
    #define BUS_CLK_HZ 16777216L
#else
    #define BUS_CLK_HZ 60000000L
#endif
```