



MSP430F2 系列 16 位超低功耗单片机模块原理

第 2 章 系统复位,中断,工作模式

版本: 1.3

日期: 2007.5.

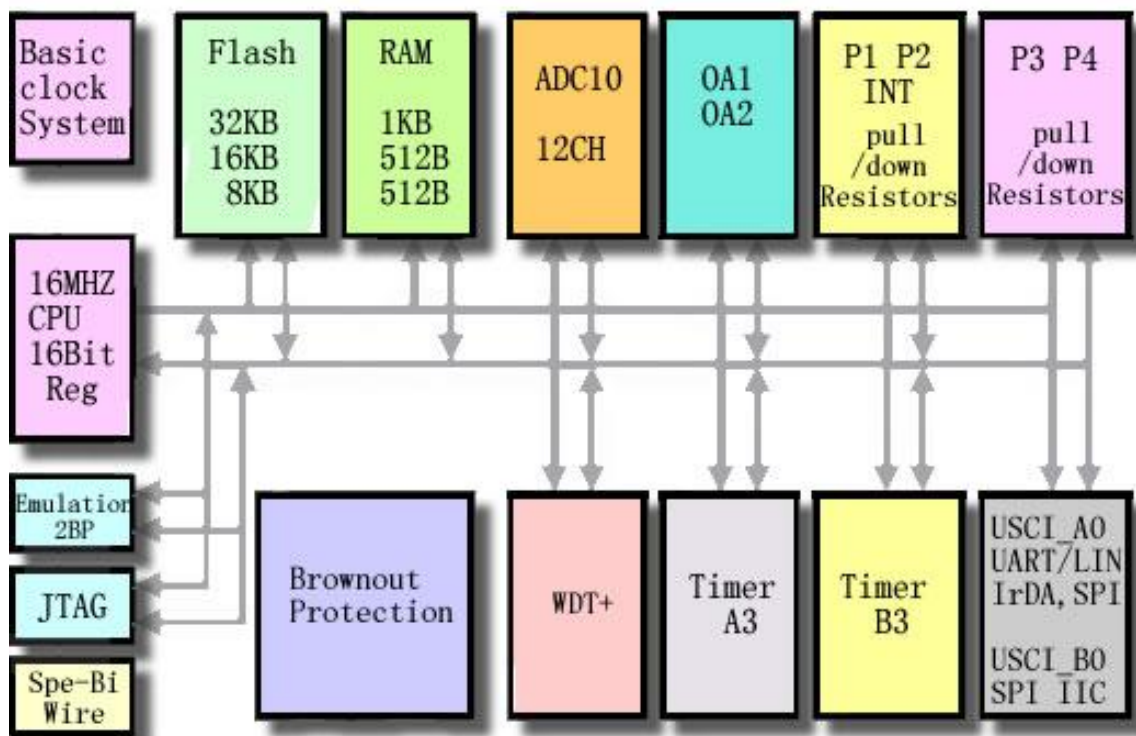
原文: TI MSP430x2xxfamily.pdf

翻译: 袁德纯

编辑: DC 微控技术论坛总版主

注: 以下文章是翻译 TI MSP430x2xxfamily.pdf 文件中的部分内容。由于我们翻译水平有限, 有整理过程中难免有所不足或错误; 所以以下内容只供参考, 一切以原文为准。

详情请密切留意微控技术论坛。



第2章 系统的复位、中断和工作模式

本章主要描述了MSP430x2xx系统的复位、中断和工作模式

本章主要内容：

- 2.1 系统的复位和初始化
- 2.2 工作模式
- 2.3 低功耗应用的原理
- 2.4 未使用的引脚的连接

2.1 系统的复位和初始化

系统的复位电路如图2-1所示，由POR信号和PUC信号驱动。各种不同的事件能触发产生这些复位信号，而根据不同的复位信号会产生不同的初始化状态。

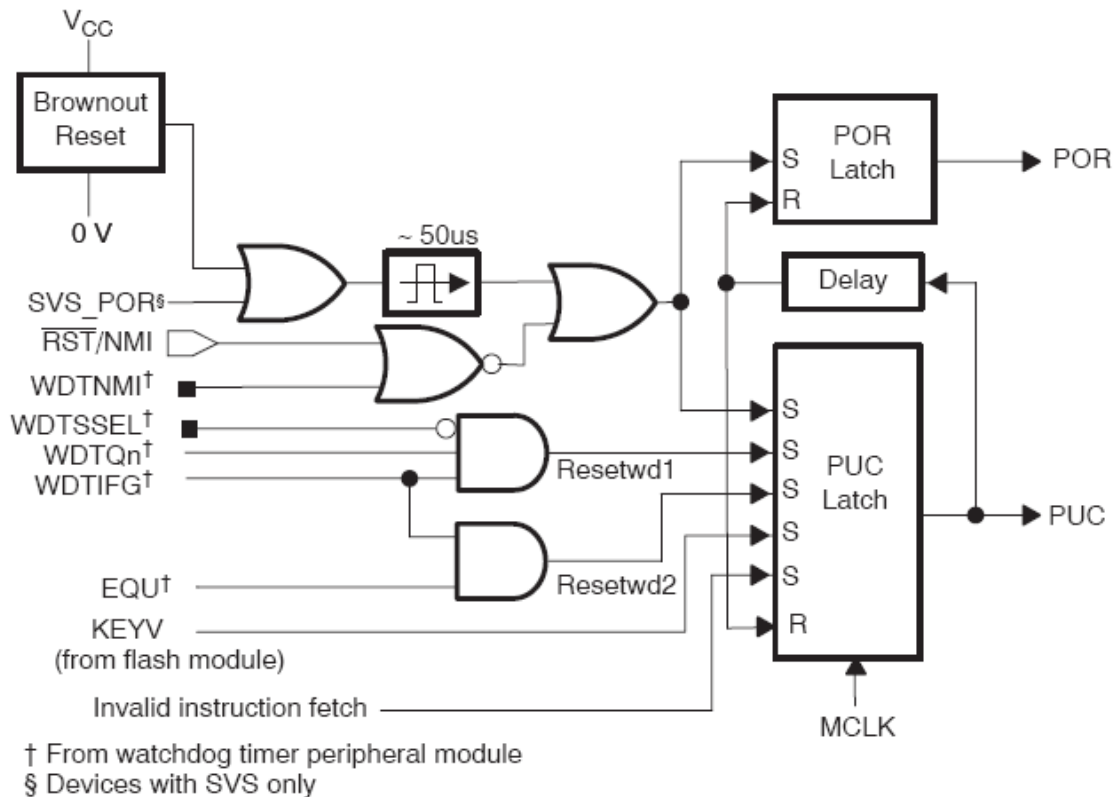


图2-1

POR信号是一个设备的复位信号，而且这个复位信号只能被以下的三种信号所产生：

1. 器件上电。
2. 复位模式配置的时候RST/NMI引脚上出现低电平
3. 当PORON =1 时，一个SVS 低状态。

一个POR的信号产生将有一个PUC信号同时产生，但是POR信号不是由PUC信号所产生的。以下事件将触发一个PUC信号：

1. 一个POR信号的产生。
2. 在WDT模式下，WDT定时器时间到。
3. WDT定时器写入错误的安全键值。
4. FLASH存储器写入错误的安全键值。

5. 一个CPU指令从地址范围为0h - 01FFh外部设备中取数据的时候。

2.1.1 BOR复位信号

BOR复位信号用于监测低的供电电压，比如一个供电电压被使用或者从VCC的终端中被清除。BOR电路复位设备是依靠在供电被使用或者被切除的时所触发一个POR信号。具体的操作路线如图2-2所示

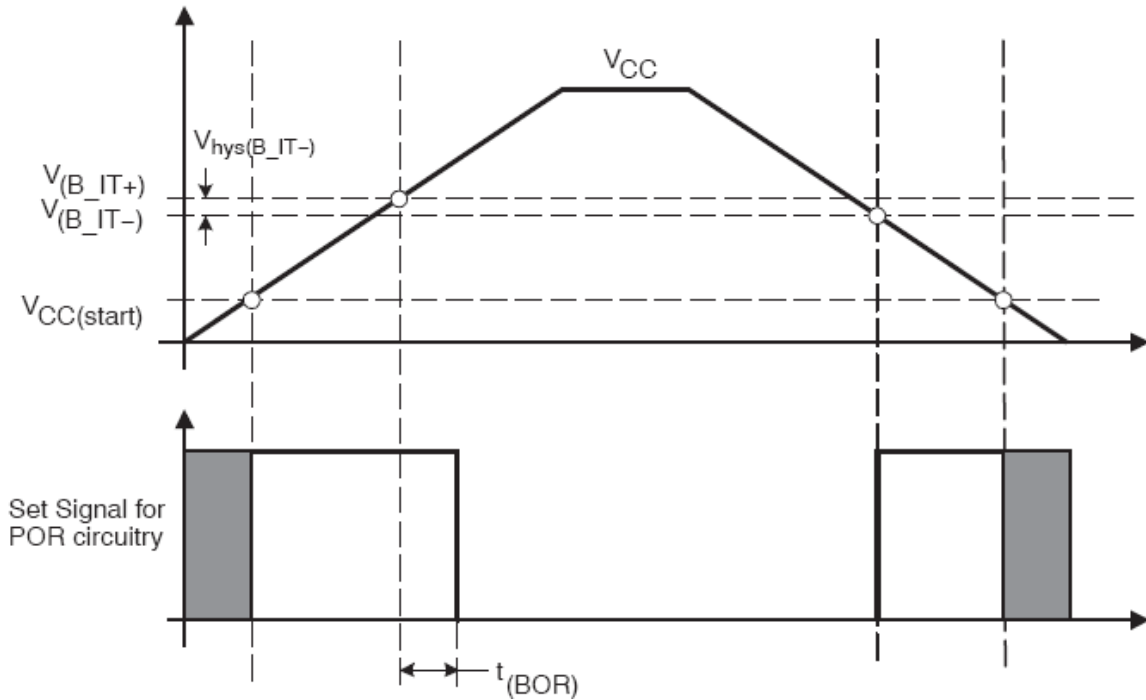


图2-2

在 $V(B_IT-)$ 明显高于POR电路的 V_{in} 时，BOR电路将在 V_{cc} 不低于 V_{in} 的位置产生一个复位信号。请参照相关的数据手册中相关的参数。

2.1.2 系统复位之后设备的初始化条件

一个POR信号之后，初始化MSP430的条件如下：

- 在复位模式时，RST/NMI引脚被配置
- I/O引脚被设置为输出，在数字I/O的章节有详细的叙述
- 外围设备模块将被初始化，参照本数据手册相关的章节
- 状态寄存器被复位
- WDT定时器将会在WDT的模式下被激活
- 程序指针将被载入到复位向量的区域（0FFFFh）。如果复位指针的向量指向0FFFFh时，设备将进入最低功耗模式。

软件初始化

系统复位之后，用户必需使用软件初始化设备以满足自己的使用需求。以下列举出了初始化必要的步骤：

- 初始化SP，RAM的最顶部
- 根据应用要求初始化WDT
- 根据应用的要求初始化外围设备模块

此外，WDT定时器，晶振的起振标志，flash存储器的标志位将会被监测去决定产生什么样的复

位信号源

2.2 中断系统

中断的优先级是固定的。中断优先级的是以各个模块的连接关系来定义的，如图2-3所示。最靠近CPU的模块是CPU/NMIRS模块，具有最高的优先级别。当系统同时有多个中断时，将依据中断优先级别去决定执行哪一个中断。

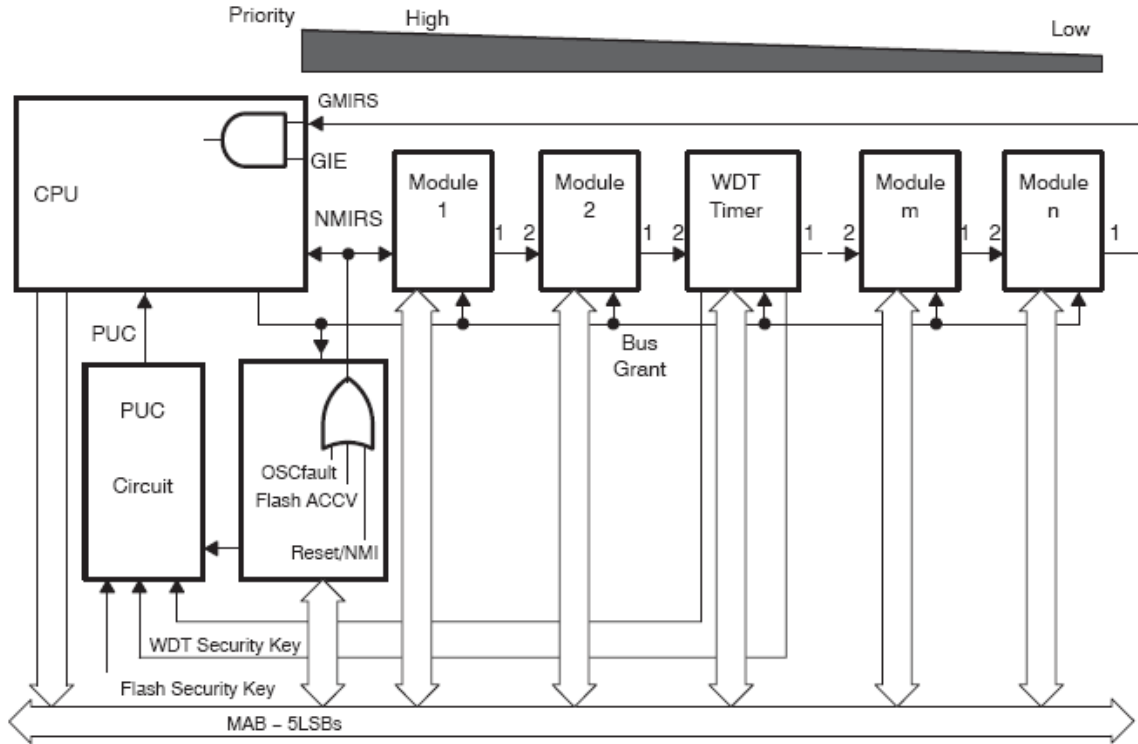


图2-3

以下三种类型的中断：

- 系统中断
- 不可屏蔽中断
- 可屏蔽中断

2.2.1 不可屏蔽中断

不可屏蔽中断不能被普通的中断使能位所屏蔽，它的使能与否是通过一个单独的中断使能位（NMIIE, ACCVIE, OFIE）来实现的。当不可屏蔽中断接收到中断信号时，所以的不可屏蔽中断使能位将会被自动复位。程序的执行的开始地址被存储于不可屏蔽中断的中断向量0FFFCH中。用户程序必须设置不可屏蔽的中断使能位，以便以后设置中断。不可屏蔽中断的方框图如图2-4所示：

不可屏蔽的中断源主要有以下三种：

1. 在NMI模式下配置系统时，RST/NMI引脚的一个上升边沿
2. 晶振发生故障
3. FLASH的存取冲突

RST/NMI 引脚

给系统上电的时候，RST/NMI引脚被配置为复位模式。RST/NMI引脚的功能在看门狗寄存

器WDTCTL中被选择。如果RST/NMI引脚被设置为复位模式，RST/NMI引脚处于低电平时CPU将一直保持复位状态。当一个高电平到来时，CPU将从存储在复位向量OFFFEH中的地址开始执行程序，RSTIFG将被置位。

如果

RST/NMI引脚被用户的软件所配置位不可屏蔽中断时，信号边沿由WDTNMIES位来选择。如果NMIIE位被置位时，RST/NMI引脚将产生一个NMI中断。RST/NMI的标志位NMIIFG将会被置1。

注释1：RST/NMI持续低电平

在NMI模式中被配置时，一个信号将发生一个NMI事件后，将不再保持RST/NMI引脚的低电平状态。如果一个PUC信号产生时，NMI信号是低电平时，设备将处于复位状态，因为一个PUC信号将使RST/NMI引脚变为复位模式。

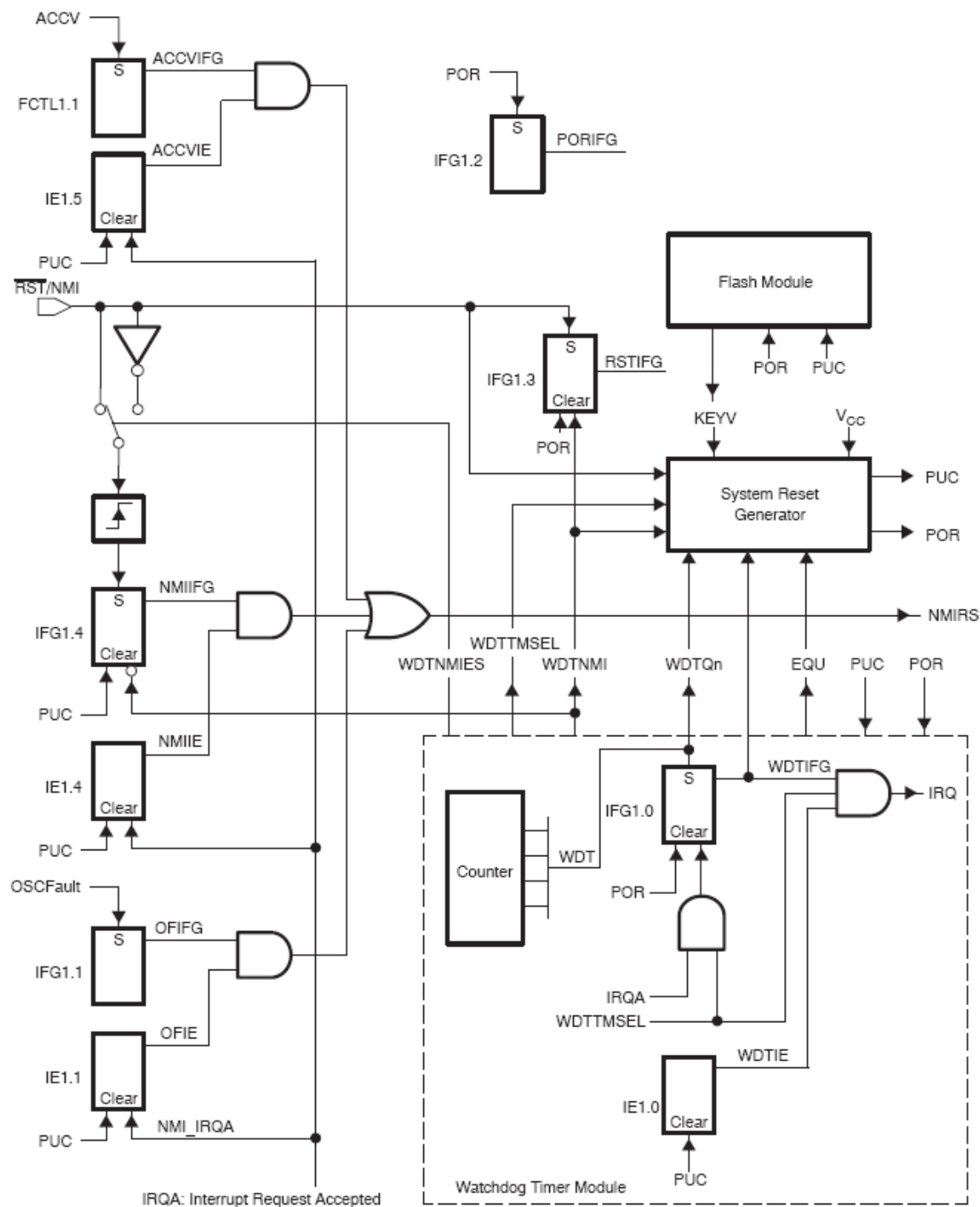


图2-4

注释2: 修改WDTNMIES

当NMI模式被选择时, WDTNMIES位将被改变。NMI是否产生将依据于实际的RST/NMI引脚的状态。在系统被设置为NMI模式前, 如果NMI的边沿选择位发生了改变, 将不会有NMI产生。

FLASH的存取冲突

当FLASH存取冲突发生时, ACCVIFG位将会被置1。置ACCVIE为1后, 在发生FLASH存取冲突发

生时，将能够产生一个NMI中断。ACCVIFG位将会被NMI中断服务程序所测试来判断一个NMI中断是否由FLASH存取冲突所产生的。

振荡器错误

振荡器的错误信号将对晶振的错误情形发出警告。置位OFIE将能够使振荡器发生错误时产生一个NMI中断。OFIFG位将会被中断服务程序所测试来判断这个NMI中断是否由振荡器错误所产生。

一个PUC信号将可能触发一个振荡器错误。因为PUC信号将能够使系统由LFXT1模式转变为LF模式，因此将关断HF模式。因此PUC信号同样可能关闭振荡器XT2。

一个NMI中断处理的实例

NMI中断是一个多源中断。一个NMI中断将自动复位NMIIE, OFIE 和ACCVIE等中断使能位。根据实际的应用需要，用户的NMI中断服务子程序将能够将中断使能位复位，以便能够从新使用NMI中断。如图2-5所示：

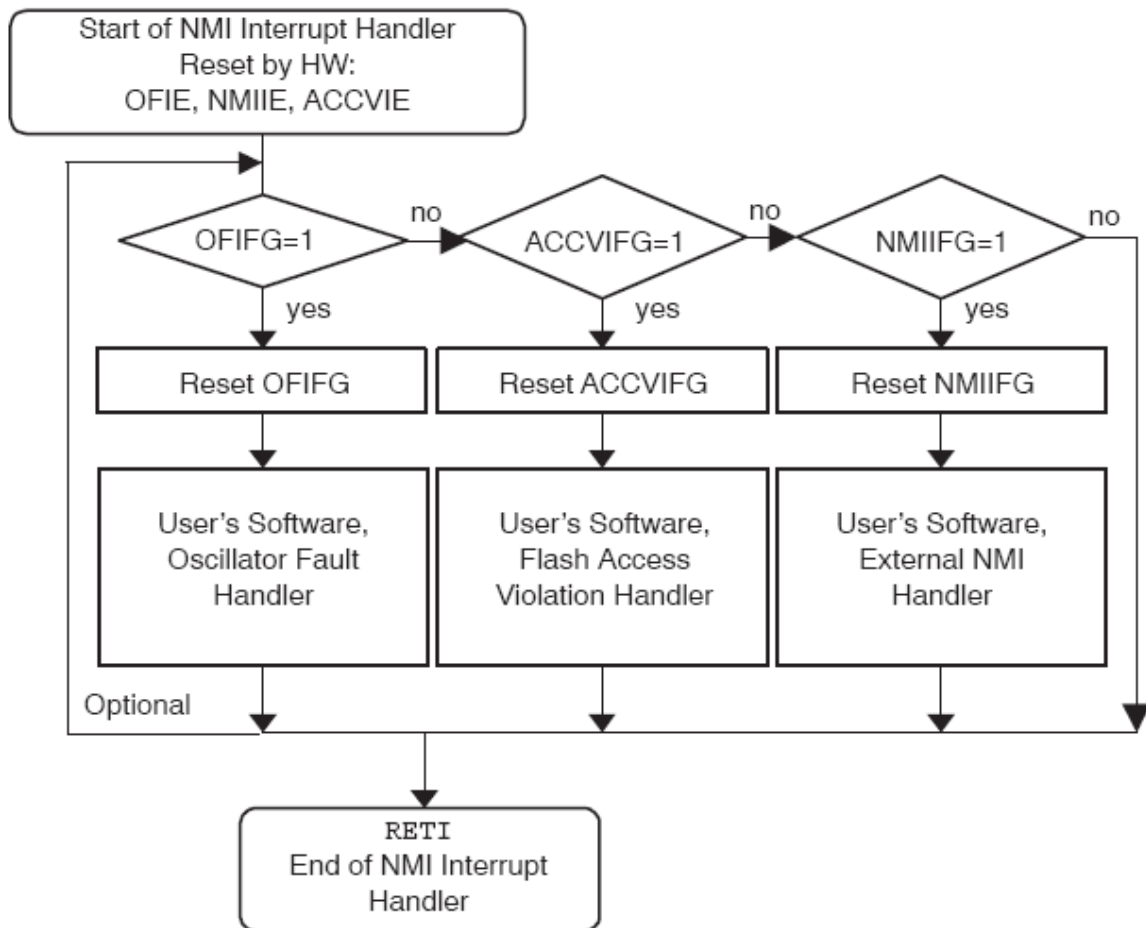


图 2-5

注释：关于NMI中断过程中ACCVIE, NMIIIE和OFIE的设置

为了防止中断嵌套，不应该在中断服务子程序中被设置ACCVIE, NMIIIE, OFIE。

2.2.2可屏蔽中断

可屏蔽中断将由有中断的能力的外围设备所产生，包括看门狗溢出中断。每个可屏蔽中断源可以被各自的中断使能位所关闭，并且所有可屏蔽中断也可以由通用使能GIE位所关闭。在本书中，各个外围设备的中断将在相关外围设备中详细讨论。

2.2.3中断处理

当一个来自于外部设备的中断被请求时，外部设备的中断使能位和通用中断使能位GIE将会被置位，中断服务子程序将会被调用。但是个别的中断使能位必须要被设置为（不）可屏蔽中断被请求。

中断的反应时间为6个机器周期，接受到中断请求的时候便开始一直持续到中断服务子程序执行第一条指令之前，如图2-6所示。中断逻辑执行在下文所述：

- 1) .任何当前指令的完成
- 2) .PC指向的下一条指令被压入到堆栈中
- 3) .SR被压入到堆栈中
- 4) .在执行最后一条指令发生了多个中断，拥有最高的优先级的中断被选择而且没有被响应
- 5) .对于单个中断源的中断标志位将会被自动复位。而多源中断的中断标志位将还是由软件去进行设置
- 6) .状态寄存器SR被清除。这将终止任何低功耗模式。因为GIE位被清除，所以任何的中断都不能发生
- 7) .中断向量中的内容将会被载入到PC中，程序将从中断开始的地址处继续执行。

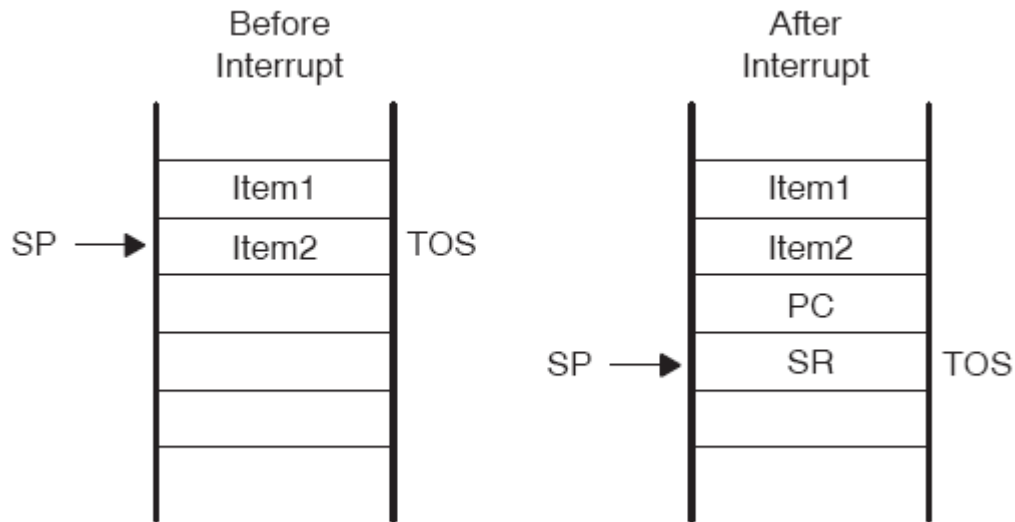


图2-6

中断返回

指令RETI (return from an interrupt service routine)将终止中断处理子程序。中断返回将占用5个周期去执行以下行为，如图2-7：

- 1) .压入到堆栈中的SR的数据出栈，所有先前（中断发生之前）的设置将有效，无论在中断服务子程序中是如何设置的
- 2) .PC出栈，从断点处执行程序。

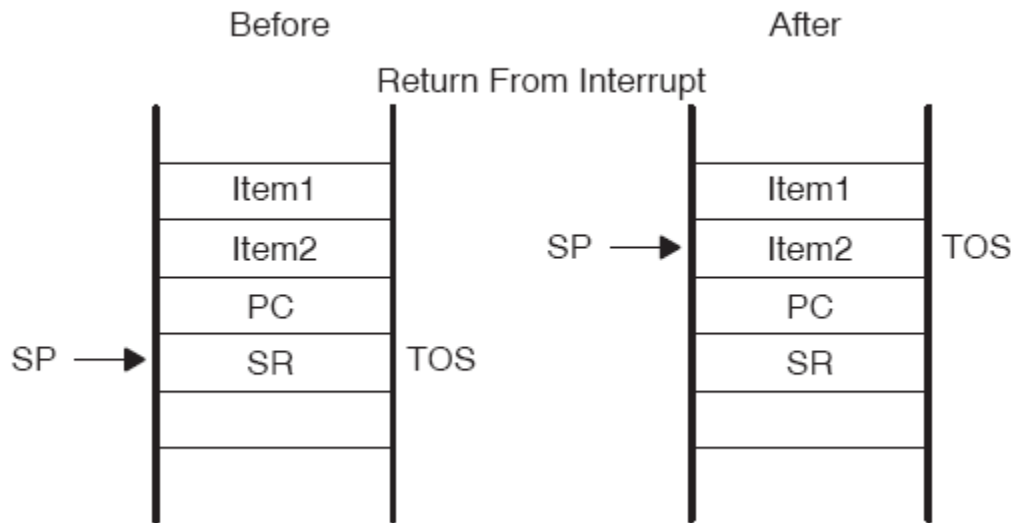


图2-7

中断嵌套

如果在中断服务子程序中设置了GIE位，此时将发生中断嵌套。当中断嵌套被激活时，任何正在执行中断服务子程序的中断将会被打断，无论它的中断优先级如何！

2.2.4 中断向量

中断向量和上电后程序运行的地址都被存放在范围为0FFFFh - 0FFC0h地址中，如表2-1所示。用户可以对每个向量所对应的16位中断服务子程序的地址编程。完整的中断向量列表还请参见相关章节中的叙述。

建议对一个模块中所分配的每个中断向量都提供一个中断服务子程序。一个哑元中断服务子程序能够由RETI指令和多个所指向它的中断向量所组成。如果有必要的话，未被分配的中断向量能够被正常的程序代码所利用。一些模块式能位，中断使能位，中断标志位将被放置在SFR寄存器中。SFR寄存器将位于低地址范围内并能够被字节指令初始化。SFR寄存器必须用字节指令访问。关于SFR如何配置还请参见设备手册中的相关章节。

中断源	中断标志	系统中断	字地址	优先权
上电复位，外部 复位，看门狗和 Flash 非法操作	PORIFG RSTIFG WDTIFG KEYV	复位	0FFFEh	31,最高
NMI，振荡器故障	NMIIFG	非屏蔽		
闪速存储器非法 存操作	OFIFG	非屏蔽e	0FFFCh	30
	ACCVIFG	非屏蔽e		
器件详情			0FFFAh	29
器件详情			0FFF8h	28
器件详情			0FFF6h	27
看门狗	WDTIFG	屏蔽	0FFF4h	26
器件详情			0FFF2h	25
器件详情			0FFF0h	24

器件详情	0FFEEh	23
器件详情	0FFECCh	22
器件详情	0FFEAh	21
器件详情	0FFE8h	20
器件详情	0FFE6h	19
器件详情	0FFE4h	18
器件详情	0FFE2h	17
器件详情	0FFE0h	16
器件详情	0FFDEh	15
器件详情	0FFDCh	14
器件详情	0FFDAh	13
器件详情	0FFD8h	12
器件详情	0FFD6h	11
器件详情	0FFD4h	10
器件详情	0FFD2h	9
器件详情	0FFD0h	8
器件详情	0FFCEh	7
器件详情	0FFCCh	6
器件详情	0FFCAh	5
器件详情	0FFC8h	4
器件详情	0FFC6h	3
器件详情	0FFC4h	2
器件详情	0FFC2h	1
器件详情	0FFC0h	0, 最低

表2-1

2.3运行模式

MSP430系列单片机被设计用于超低功耗的应用，有多种运行模式可被使用，如图2-8所示

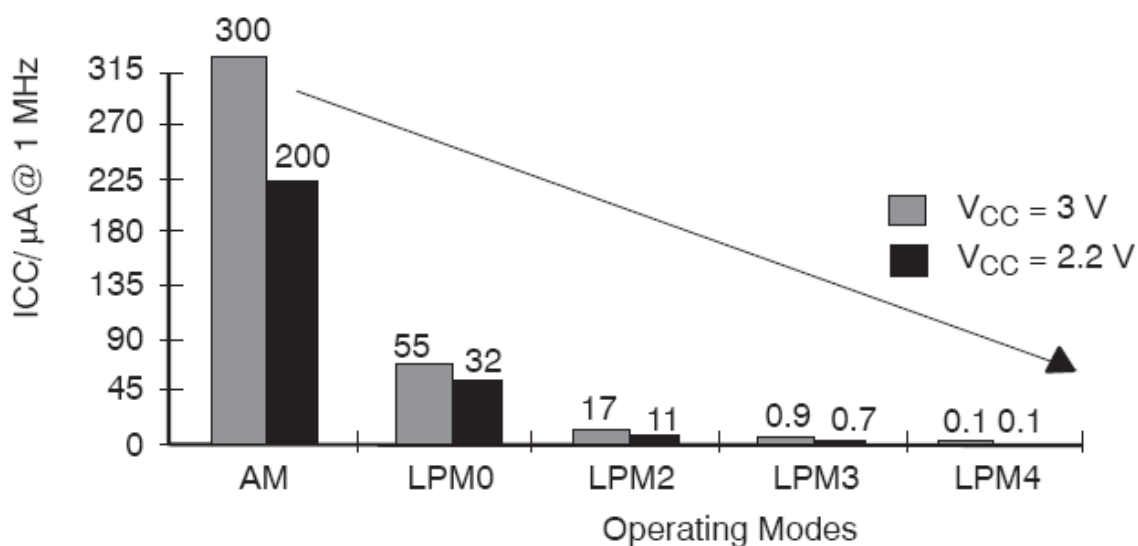


图2-8

运行模式将着重考虑以下的三方面需求：

- [1]超低功耗
- [2]速度和数据吞吐量
- [3]个别外围器件的电流损耗

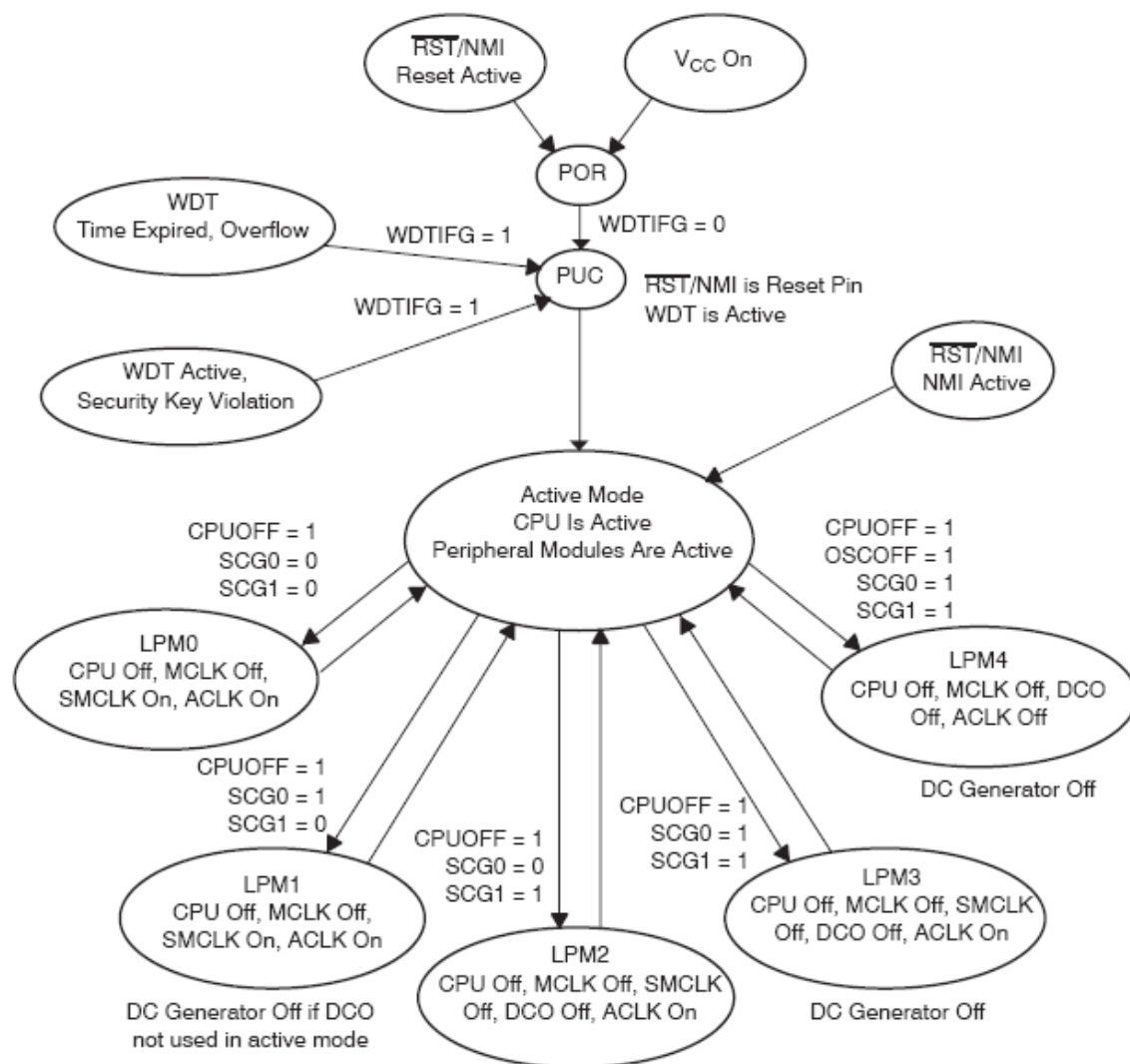


图2-9

MSP430的典型电流损耗如图2-9所示

SCG1	SCG0	OSCOFF	CPUOFF	Mode	CPU 和 时钟状态
0	0	0	0	活动	CPU是活动的，所有时钟是允许活动的。

0	0	0	1	LPM0	CPU, MCLK 被禁止。SMCLK, ACLK 是活动的。
0	1	0	1	LPM1	CPU, MCLK, DCO osc被禁止。如果DCO不使MCLK或SMCLK 在活动模式, DC产生器是被禁止。SMCLK, ACLK活动
1	0	0	1	LPM2	CPU, MCLK, SMCLK, DCO osc被禁止。DC产生器保护允许ACLK活动。
1	1	0	1	LPM3	CPU, MCLK, SMCLK, DCO osc被禁止。DC产生器 被禁止。ACLK活动。
1	1	1	1	LPM4	CPU和所有时钟是被禁止。

低功耗的0-4模式可以通过设置SR寄存器中的CPUOFF, OSCOFF, SCG0, SCG1位来配置。在执行中断服务子程序时, SR寄存器的内容将会被压入到堆栈中如果在中断服务子程序中没有改变先SR中的内容, 那么程序将返回到先前设置的运行模式。如果中断服务子程序操作存储于堆栈中的SR寄存器中的内容, 将使得程序返回至一个不同的运行模式。模式控制位和堆栈将允许任何指令进行存取访问。当对模式控制位进行设置后, 被选中的模式将会被立即执行。任何的时钟禁止信号将能够使得外围设备停止工作直到时钟信号有效为止。所以的I/O端口和RAM或寄存器均没有被改变过。尽量通过中断方式来唤醒时钟。

2.3.1 进入和退出低功耗模式

一个中断将能够将MSP430从任意的低功耗运行模式中唤醒。程序流程如下:

进入中断服务子程序:

PC和SR中的值将会被存储在堆栈中

CPUOFF, SCG1和OSCOFF位将会被自动复位

中断从中断服务子程序中返回的一些选项:

原始的SR出栈, 恢复先前的运行模式

存储在堆栈中SR位能够在执行中断服务子程序的过程中被修改, 当指令RETI执行时将会返回到一个不同的运行模式中。

进入LPM0的例子:

```
BIS #GIE+CPUOFF, SR.....
```

退出LPM0的例子:

```
BIC #CPUOFF, 0 (SP)
```

```
RETI
```

进入LPM3的例子:

```
BIS #GIE+CPUOFF+SCG1+SCG0, SR
.....
```

退出LPM3的例子:

```
BIC #CPUOFF+SCG1+SCG0, 0 (SP)
RETI
```

2.4 低功耗应用的原理

通常, 减少功耗的最重要的因素就是使所使用的430时钟系统能够在LPM3的模式下得到最优化的使用。在实时时钟功能和所有的中断都被激活的情况下, LPM3模式下的典型功耗小于2uA。

32KHz的晶振可用于ACLK的时钟信号源，CPU时钟来自于DCO，此时只需要6US的唤醒时间。

[1]用中断去唤醒处理器和控制程序

[2]外围设备在需要时被打开

[3]用低功耗集成外围设备的模式去取代软件驱动的功能。例如，在CPU不工作的情形下，Timer_A和Timer_B能够自动的产生PWM波形，并捕捉外部时钟。

[4]分支计算和快速查表应该代替查询标志位和长时间的软件计算。

[5]避免频繁的子程序和功能调用。

[6]对于较长的软件程序中，单周期的CPU寄存器应该被使用。

2.5 不用引脚的连接方法

没有使用引脚的正确接法如表2-2所示

引脚	电平	说明
AVCC	DVCC	
AVSS	DVSS	
VREF+	开路	
VeREF+	DVSS	
VREF- /VeREF-	DVSS	
XIN	DVCC	
XOUT	开路	
XT2IN	开路	
XT2OUT	开路	
Px.0 to Px.7	开路	不用的端口方向设置成输出 或设为输入和上拉/下拉有效
RST/NMI	DVCC or VCC	用47 k Ω 上拉电阻和10nF电容下拉



MSP430F22x4 评估板

专业提供 **MSP430** 单片机开发工具