



MSP430系列单片机

张晞 王德银 张晨 编著

实用C语言 程序设计

- 介绍如何用C语言编写MSP430程序
- 讲述编译器以及C语言扩展部分的使用方法
- 涵盖可即时查阅的库函数使用说明
- 深入剖析程序设计的规范与结构
- 提供实用的程序功能模块



人民邮电出版社
POSTS & TELECOM PRESS

MSP430系列单片机

张晞 王德银 张晨 编著

实用C语言 程序设计

人民邮电出版社

图书在版编目 (CIP) 数据

MSP430 系列单片机实用 C 语言程序设计 / 张晞, 王德银, 张晨编著.

—北京: 人民邮电出版社, 2005.9

ISBN 7-115-13664-5

I. M... II. ①张...②王...③张... III. ①单片微型计算机, MSP430
②C 语言—程序设计 IV. ①TP368.1②TP312

中国版本图书馆 CIP 数据核字 (2005) 第 089732 号

内 容 提 要

本书从应用角度出发, 主要介绍 MSP430 的硬件基础部分和 IAR 公司的 MSP430 C 编译器 EW430, 并对 MSP430 中的各功能模块给出了应用实例。这些实例程序按照结构化编写, 经作者的封装后, 读者在开发中只需稍加修改即可直接调用。书中还介绍了 MSP430 的几种典型应用, 如软件模拟串行口、在线刷新 FLASH、实现中断嵌套等, 并为其编写了完整代码, 读者完全可以将其直接组合在自己的项目中。书中还讲解了单片机领域编写程序的规范、程序结构的安排以及如何提高编码效率等实际应用中的问题。随书光盘中包括了本书的所有程序代码。

MSP430 系列单片机实用 C 语言程序设计

- ◆ 编 著 张 晞 王德银 张 晨
责任编辑 刘映欣
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京顺义振华印刷厂印刷
新华书店总店北京发行所经销
- ◆ 开本: 787×1092 1/16
印张: 17.5 插页: 1
字数: 426 千字 2005 年 9 月第 1 版
印数: 1—5 000 册 2005 年 9 月北京第 1 次印刷

ISBN 7-115-13664-5/TP · 4784

定价: 35.00 元 (附光盘)

读者服务热线: (010) 67132692 印装质量热线: (010) 67129223

前 言

以前出版的有关 MSP430 的书籍主要是从 MSP430 内部结构的角度来讲述的,读者可以从中获得比较全面的相关知识,但这些书的描述往往过于底层,使得读者使用时必须花费相当多的时间认真推敲各个寄存器或者模块之间的关系。本书所描述的重点与以往的相关书籍不同,本书并不准备详细地将 MSP430 重新描述一遍,相关的知识可以从 MSP430 的数据手册或者类似书籍中获得。本书将通过实际应用来讲述如何用 C 语言编写 MSP430 的程序,所提供的例子涵盖 MSP430 大多数的功能模块,而且都是采用结构化编程方法,使得这些例子中的代码能够被读者重复利用,或者只需在例子的框架内进行修改,就可以满足需要。

MSP430 的功能非常强大,其内部的寄存器非常多,关系也比较复杂,对于初学者,往往会有无从下手的感觉。通过学习本书中的例子,初学者能够尽快渡过初学 MSP430 时的瓶颈期。对于已经有经验的读者,本书更像是一本编程的代码库,因为书中的例子都立足于应用,可以重复利用,能够大大提高工作效率。本书还详细讲述了 IAR 公司的 C 语言开发工具,可以作为 C 语言开发 MSP430 的使用手册。书中所有的例程都在 IAR 公司的 C 编译器 EW430 3.1A 版本下完成,并在北京东方美源公司提供的实验板上使用 JTAG 仿真器调试通过。由于没有详细介绍 MSP430 内部的寄存器,因此,读者阅读本书的时候,手边最好有相应的 MSP430 芯片手册,以便随时查阅。

本书由张晞、王德银、张晨三人编写而成。张晞,中国矿业大学机电学院副教授,机械制造及其自动化学科带头人,长期从事单片机的教学及研究工作。王德银,高级工程师,现任河南平顶山煤业集团五矿矿长,主要从事煤矿机电自动化和安全生产监测系统的研究。张晨,现任北京东方美源公司研发部主管,多年从事单片机的应用设计工作。本书在成书过程中得到北京东方美源科技有限公司的大力支持,在此表示感谢。

由于作者水平有限,成书时间仓促,书中必定会存在不妥之处,恳请广大读者批评指正,以便在以后的版本中及时修正,联系地址:mcu@mcu163.com(作者)或 liuyingxin@ptpress.com.cn(编辑)。欢迎访问作者的个人网站 <http://www.mcu163.com>,欢迎与作者进行技术交流。

编 者

2005 年 8 月

目 录

第 1 章 MSP430 硬件基础知识.....	1
1.1 概述.....	2
1.2 存储器结构.....	2
1.3 复位.....	4
1.4 系统时钟.....	6
1.5 低功耗模式.....	8
1.6 中断.....	10
1.7 外围模块.....	16
第 2 章 C 语言基础知识.....	17
2.1 标识符与关键字	18
2.1.1 标识符	18
2.1.2 关键字	18
2.2 数据类型.....	19
2.2.1 基本型	19
2.2.2 构造型	20
2.2.3 指针型	21
2.3 运算符.....	22
2.4 函数.....	26
2.5 数组.....	27
2.6 指针.....	27
2.7 位运算.....	29
2.8 存储类型.....	29
2.8.1 变量	29
2.8.2 函数	30
2.9 预处理功能.....	30
2.9.1 宏定义	30
2.9.2 条件编译	31
2.9.3 文件包含	31
2.10 程序的基本结构	32
2.10.1 顺序结构	32
2.10.2 选择结构	32
2.10.3 循环	34

2.10.4 跳转	35
第 3 章 IAR C 编译器的使用	37
3.1 概述	38
3.1.1 特性	38
3.1.2 软件结构	38
3.1.3 文件类型	39
3.2 开发调试环境	40
3.2.1 创建一个工程	40
3.2.2 编译链接项目	43
3.2.3 项目设置	44
3.2.4 调试	50
3.3 语言扩展	55
3.3.1 扩展关键字	56
3.3.2 内部函数	57
3.3.3 扩展定义	60
3.4 C 语言与汇编语言混合使用	63
3.4.1 调用内部函数	63
3.4.2 直接嵌入	63
3.4.3 调用汇编模块	64
3.5 编写高质量的代码	67
3.6 函数库	68
第 4 章 开发工具	97
4.1 JTAG 仿真器、编程器	98
4.2 BSL 编程器	99
第 5 章 程序设计的规范与结构	101
5.1 程序规范	102
5.2 程序结构	106
5.3 框架程序	109
第 6 章 MSP430 异步串行通信	123
6.1 串行通信简介	124
6.2 串行通信软件实现	125
第 7 章 定时器	137
7.1 16 位定时器	138
7.1.1 定时中断	139

7.1.2 PWM 输出	142
7.1.3 捕获脉冲信号周期	149
7.1.4 软件模拟异步串行通信	153
7.2 基本定时器	161
第 8 章 FLASH 的读写、擦除与 I/O 端口	167
8.1 FLASH 的读写和擦除	168
8.2 I/O 端口	174
8.2.1 非行列式键盘	175
8.2.2 行列式键盘	185
第 9 章 DMA 数据传输与 IIC 总线	193
9.1 DMA 数据传输	194
9.2 IIC 总线	200
第 10 章 FLL+锁频环与液晶屏驱动模块	215
10.1 FLL+锁频环	216
10.2 液晶屏驱动模块	218
第 11 章 AD、DA 转换	233
11.1 ADC12	234
11.2 DAC12	244
第 12 章 比较器 A	251
12.1 斜边 AD 转换	252
12.2 电阻值测量	260
第 13 章 特殊处理	267
13.1 中断嵌套	268
13.2 程序异常处理	270
附录 A MSP430 基本电路图	

第 1 章

第 1 章 MSP430 硬件基础知识

MSP430 是美国 Texas Instruments (TI) 公司于 1996 年开始推向市场的超低功耗微处理器。除了超低功耗这一特点外,在 MSP430 系列产品中还集成了很多功能模块,可以应用于各个方面,使得用一片 MSP430 芯片就能完成以前要用多片芯片组合才能完成的功能,大大地缩小了产品的体积,并且降低了成本。因此,MSP430 系列一经推出,就受到了业内人士的欢迎。迄今为止,MSP430 已经广泛应用于各个领域,尤其是便携式仪器仪表、监测、医疗器械以及汽车电子等领域。

本章将对 MSP430 的基础特性进行简明扼要地介绍,外围模块的具体操作则完全略过。对基础特性的了解是操作任何一种微处理器所必备的,仔细阅读本章内容可为学习全书打下良好的基础。

1.1 概 述

当前市场上存在很多种微处理器，每种微处理器都各有其特色。针对不同的应用选择合适的微处理器非常重要。MSP430 的主要特点为：

(1) 超低功耗 拥有 5 种低功耗模式，以适应不同的需要。CPU 从低功耗模式被唤醒，这个过程最多只需要 $6\mu\text{s}$ ，因此，在某些需要迅速作出反应的应用中，CPU 能够及时退出低功耗模式，进入工作模式。

(2) 灵活的时钟使用方式 除了片内集成一个晶体振荡器外，还可外接 1~2 个晶体振荡器。不同的内部功能模块可根据需要使用不同的晶体振荡器，在不需要时可以通过设置寄存器将其关闭，以降低功耗。

(3) 高速的运算能力 16 位 RISC 架构，125ns 指令周期。

(4) 丰富的功能模块 集成了大量的功能模块，这些功能模块包括：

- 多通道 10~14 位 AD 转换器
- 双路 12 位 DA 转换器
- 比较器
- 液晶驱动器
- 电源电压检测
- 串行口 USART (UART/SPI)
- 硬件乘法器
- 看门狗定时器和多个 16 位、8 位定时器（可进行捕获、比较、PWM 输出）
- DMA 控制器
- FLASH 存储器。它可以在运行过程中由程序控制写操作和段的擦除（In system programmable），不需要额外的高电压。

(5) 廉价专业的开发工具 MSP430 的芯片上包括 JTAG 接口，因此在仿真调试程序时，通过一个廉价的 JTAG 接口转换器就可以完成以往用昂贵的仿真器才能完成的功能，如设置断点、单步执行程序、读写寄存器等。

(6) 灵活快速的编程方式 可通过 JTAG 和 BSL 两种方式向 CPU 内装载程序。

(7) 高保密性 只需按照特定的方式将 MSP430 内部的熔丝熔断，JTAG 口便被物理性地阻断。BSL 方式所需要的密码长达 256 位，排列组合出来的密码量为 2 的 256 次幂，如此巨大的数量被破解的可能性微乎其微。

(8) 低电源电压范围 1.8~3.6V。

1.2 存储器结构

MSP430 内部存储器的类型有：程序存储器 FLASH、数据存储器 RAM、外围模块寄存

器、特殊功能寄存器。

典型微型计算机的存储器都是采用冯·诺依曼结构，也称普林斯顿结构，即存放程序指令的存储器——程序存储器和存放数据的存储器——数据存储器采取统一的地址编码结构。程序存储器与数据存储器分开的地址编码结构称为哈佛结构，如 MCS-51 系列微处理器。MSP430 采用冯·诺依曼结构。其安排如图 1-1 所示，全部寻址空间为 64K 字节。需要注意的是，虽然 MSP430 是 16 位的微处理器，但其寻址空间还是按照字节来计算的。

● 从 0 至 0xF 为特殊功能寄存器。共 16 个字节，包含中断标志寄存器 1 (IFG1)、中断标志寄存器 2 (IFG2)、中断允许寄存器 1 (IE1)、中断允许寄存器 2 (IE2)、模块允许寄存器 1 (ME1)、模块允许寄存器 2 (ME2)。

● 从 0x10 至 0x1FF 为外围模块寄存器。包含被定时器、AD 转换器、对外端口等模块用到的寄存器。

● 从 0x200 开始为数据存储器 RAM。不同型号中数据存储器的大小不同，但都是从 0x200 地址开始向高端地址扩展。如 MSP430F149 的数据存储器容量为 2KB，其地址范围即为 0x200~0x9FF。

● 从 0x0C00 到 0x0FFF 为 BOOT ROOM。其中存储的内容是生产芯片时掩模在芯片内的一段代码，此段代码用来完成 BSL (bootstrap) 功能。使芯片的保密熔丝熔断以后，仍然可以通过 BSL 方式修改芯片内的代码。

● 从 0x1000 到 0x107F 是 128 个字节的 FLASH 存储器，称为信息存储器 B。此段存储器与高端地址存储代码的 FLASH 存储器本质上没有任何不同，同样也可以存储代码并执行，只是这一段存储器的长度较小，只有 128 个字节。主要用来存储一些掉电后仍需保存的数据。由于它是 FLASH 存储器，因此可以按照字或者字节写入，但必须整段擦除。

● 从 0x1080 到 0x10FF 为信息存储器 A。功能与信息存储段 B 相同。

● 程序存储器从 0xFFFF 开始向低端地址扩展，不同型号中程序存储器的容量不同，但都是从 0xFFFF 开始向下扩展。如 MSP430F133 的程序存储器容量为 8KB，其地址范围即为 0xE000~0xFFFF；MSP430F149 的容量为 60KB，其地址范围为 0x1100~0xFFFF。需要注意的是，在程序存储器容量为 60KB 的芯片中，程序存储器与信息存储器 A、B 发生了重合，从地址 0xFFFF 向低端地址扩展 60KB，其地址范围为 0x1000~0xFFFF，而信息存储器 A 和 B 的地址范围为 0x1000~0x10FF。程序存储器是 flash 存储器，分为若干段进行管理，可以按照字或者字节写入，擦除时无法按照字或者字节擦除，每次至少擦除一段，每段长度为 512 字节。

● 0xFFE0~0xFFFF 是程序存储器的一部分，共 32 个字节。MSP430 规定用这一段存储器来存储各种中断的中断向量。

由于程序存储器、信息存储器、数据存储器都是统一寻址的，所以，程序在这 3 种存

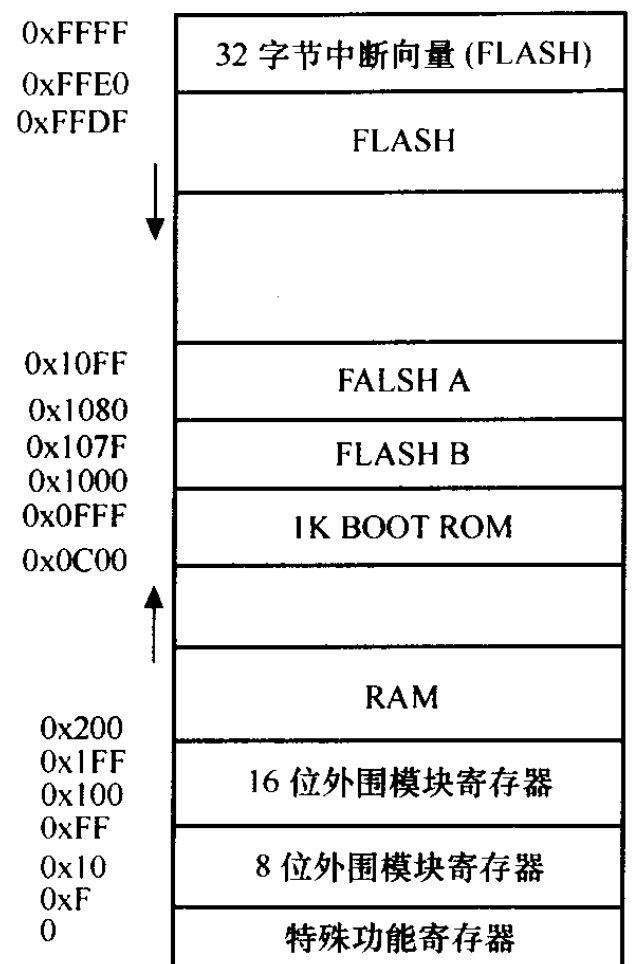


图 1-1 MSP430 存储器结构

存储器中均可执行。不同的是，程序放在数据存储器中，掉电后就会丢失，并且很容易在执行中被改写，故一般情况不会将程序放在数据存储器中执行。修改程序存储器中的内容必须经过解锁操作才能进行，否则会引起系统复位（PUC，参见 1.3 节）。有关解锁的操作会在后续章节中介绍。

MSP430 的存储结构使得使用 C 语言编程所得到的代码仍然有很高的执行效率，因此，除非对程序的大小和执行时间有很高的要求，否则都应该选用 C 语言编写程序。

1.3 复 位

复位是微处理器开始工作的起点，因此了解复位过程和复位结束时微处理器的状态对正确使用微处理器是至关重要的。

图 1-2 是 MSP430 复位的内部逻辑图。MSP430 的复位信号有两种：上电复位信号（POR）和上电清除信号（PUC）。某些型号的芯片用 Brownout Reset（BOR）模块取代了产生 POR 的模块。从图中可以看到，触发复位的信号除了通过 POR 监测电路检测系统上电而产生 POR 信号之外，还有多种信号能够触发 POR 和 PUC 信号，其中 5 种来自看门狗，1 种来自复位管脚，1 种来自写 FLASH 键值出现错误所产生的信号。

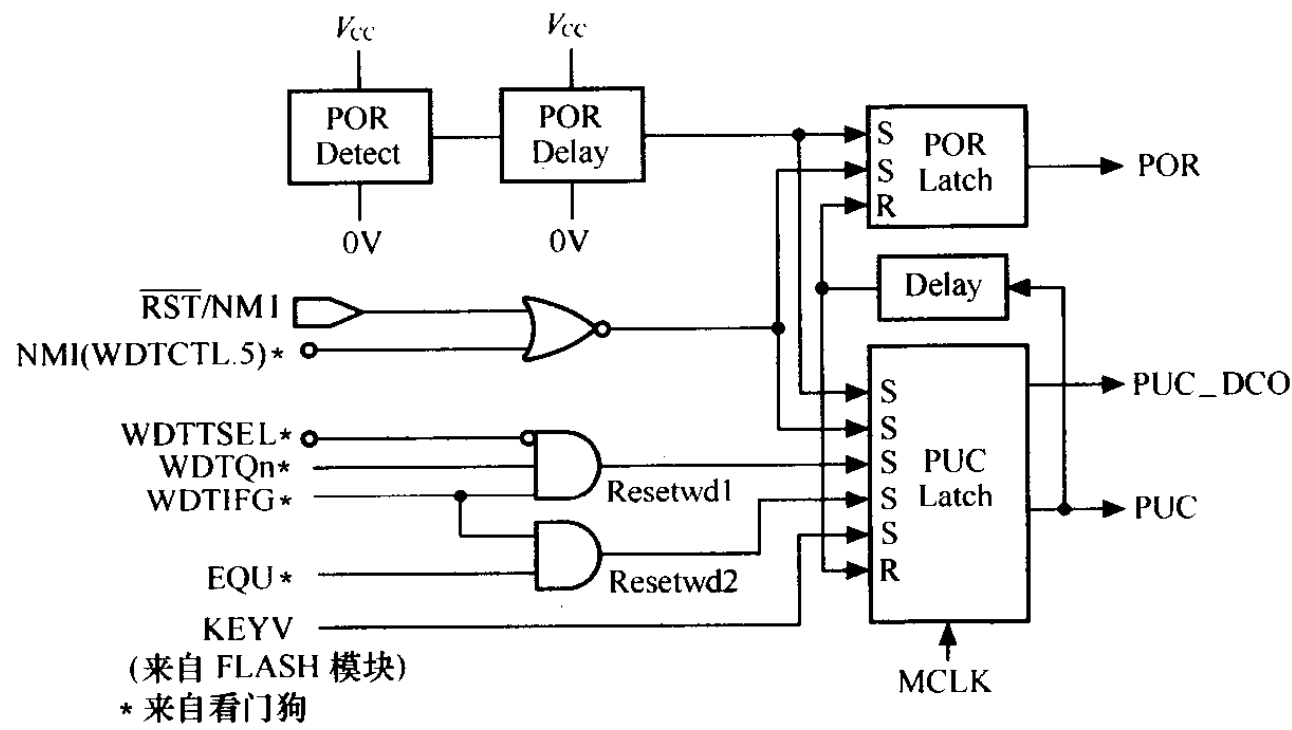


图 1-2 MSP430 内部复位逻辑电路

POR 信号只在两种情况下产生：

- 微处理器上电。
- RST/NMI 管脚被设置为复位功能，在此管脚上产生低电平时系统复位。

POR 的过程如图 1-3 所示，在微处理器上电的过程中，当电源电压超过 V_{POR} 时，延迟一段时间（POR_DELAY）后才使微处理器复位。在运行的过程中，如果电源电压 V_{cc} 下降到 $V_{(MIN)}$ 以下再恢复至超过 V_{POR} ，则仍然会延迟一段时间（POR_DELAY）后才使微处理器复位。延迟时间的作用是使芯片内各模块都能够有足够的时间有效地复位。如果电源电压未下降到

$V_{(\text{MIN})}$ 以下, 则不会产生 POR 过程。

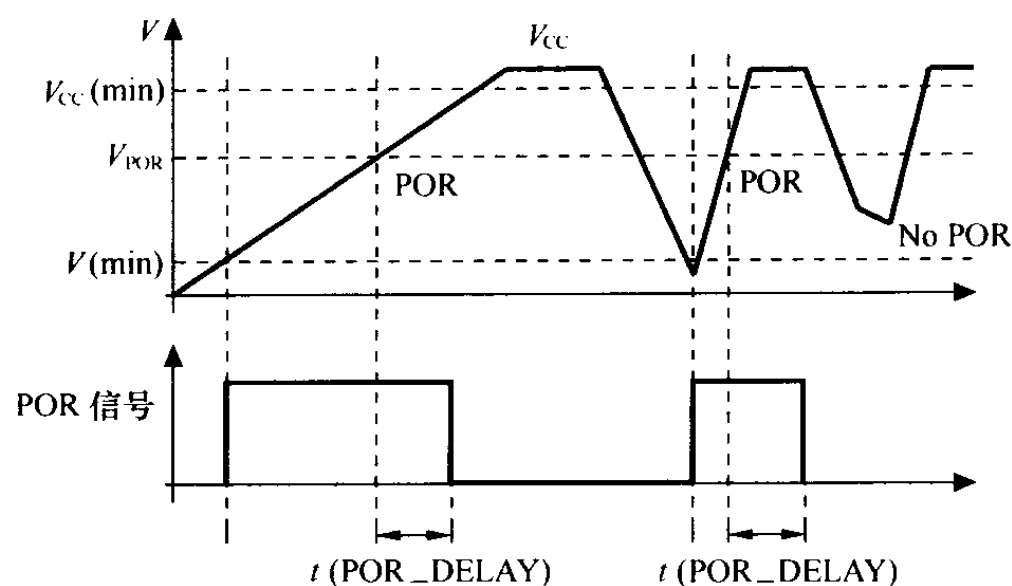


图 1-3 POR 过程

RST/NMI 管脚设置为复位功能时, 管脚上产生低电平, 同样会导致 POR 过程, 其过程与微处理器上电产生的 POR 一致。

BOR 与 POR 类似, 最终产生的信号也称为 POR 信号, 其过程如图 1-4 所示。BOR 检测电源电压的上升和下降, 上电和掉电都可以产生 POR 信号, 导致系统复位。当电源电压上升超过 $V_{\text{CC}(\text{start})}$ 时, POR 信号开始产生, 一直持续到电源电压超过 $V_{(\text{B_IT+})}$, 延迟 $t_{(\text{BOR})}$ 时间后, POR 信号结束。只有电源电压下降至 $V_{(\text{B_IT-})}$ 以下才能再次产生 POR。 $V_{(\text{B_IT-})}$ 比图 1-3 中 POR 过程中的 $V_{(\text{MIN})}$ 高很多, 所以, BOR 过程比 POR 过程严格, 也更能保证程序运行的安全性。

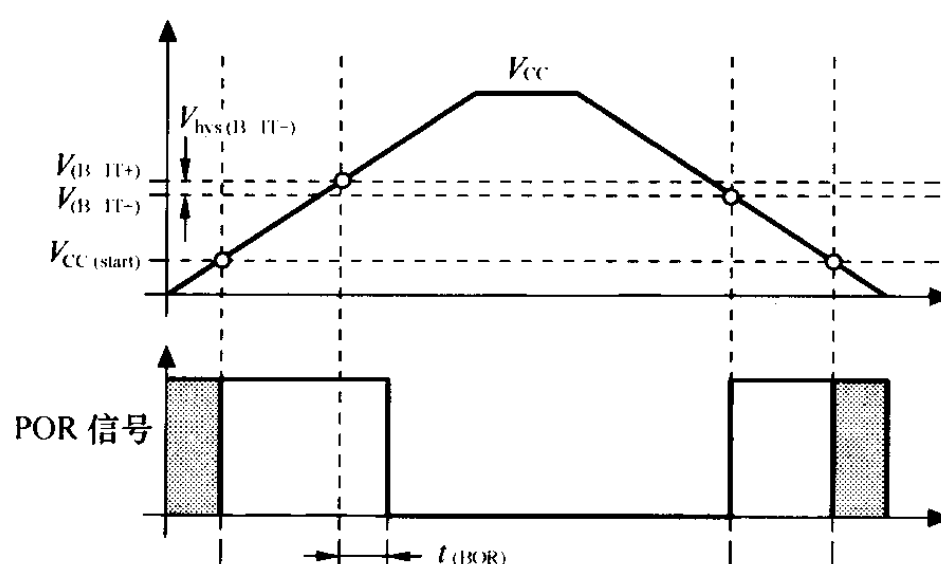


图 1-4 BOR 过程

PUC 信号产生的条件为:

- POR 信号产生。
- 看门狗有效时, 看门狗定时器溢出。
- 写看门狗定时器安全键值出现错误。
- 写 FLASH 存储器安全键值出现错误。

无论是 POR 信号还是 PUC 信号触发的复位, 都会使 MSP430 在地址 0xFFFFE 处读取复位中断向量, 程序从中断向量所指的地址处开始执行。触发 PUC 信号的条件中, 除了 POR 产生触发 PUC 信号外, 其他的都可以通过读取相应的中断向量来判断是何种原因引起的 PUC

信号，以便作出相应的处理。

POR 信号的出现会导致系统复位，并产生 PUC 信号。而 PUC 信号不会引起 POR 信号的产生。系统复位后（POR 之后）的状态为：

- RST/MIN 管脚功能被设置为复位功能。
- 所有 I/O 管脚被设置为输入。
- 外围模块被初始化，其寄存器值为相关手册上注明的默认值。
- 状态寄存器（SR）复位。
- 看门狗激活，进入工作模式。
- 程序计数器（PC）载入 0xFFFF（0xFFFF 为复位中断向量）处的地址，微处理器从此地址开始执行程序。

典型的复位电路有以下 3 种：

（1）由于 MSP430 具有上电复位功能，因此，上电后只要保持 RST/NMI（设置为复位功能）为高电平即可。通常的做法为，在 RST/NMI 管脚接 100kΩ 的上拉电阻，如图 1-5（a）所示。

（2）除了在 RST/NMI 管脚接 100kΩ 的上拉电阻外，还可以再接 0.1μF 的电容器，电容的另一端接地，可以使复位更加可靠。如图 1-5（b）所示。

（3）由于 MSP430 具有极低的功耗，如果系统断电后立即上电，则系统中电容所存储的电荷来不及释放，此时系统电压不会下降到最低复位电压以下，因而 MSP430 不会产生上电复位，同时 RST/NMI 管脚上也没有足够低的电平使 MSP430 复位。这样，系统断电后立即上电，MSP430 并没有被复位。为了解决这个问题，可增加一个二极管，这样断电后储存在复位电容中的电荷就可以通过二极管释放，从而加速电容的放电。二极管的型号可取 1N4008。如图 1-5（c）所示。

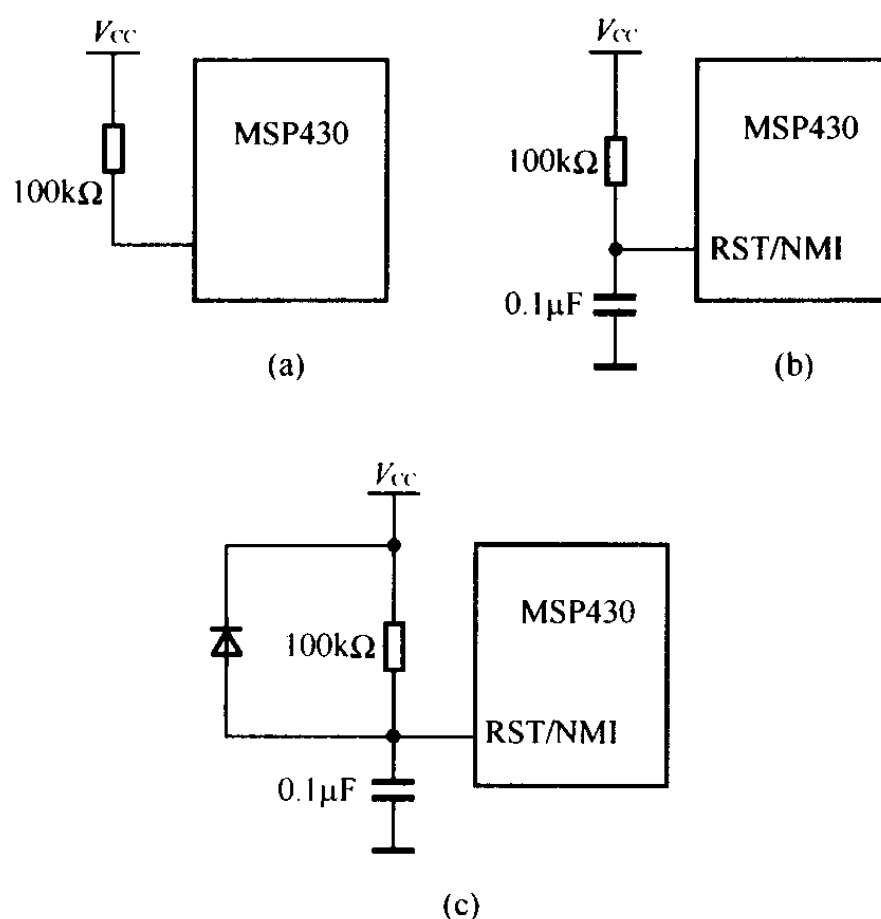


图 1-5 典型复位电路

1.4 系统时钟

耗电量与微处理器的工作频率有直接的关系。MSP430 微处理器根据型号的不同最多可以使用 3 个振荡器。使用者根据需选择合适的振荡频率，并可以在不需要时随时关闭其中一些振荡器，以节省功耗。这 3 个振荡器分别为：

（1）DCO 数控 RC 振荡器，位于芯片内部。不用时可以关闭。图 1-6 为 DCO 振荡器的逻辑控制图。

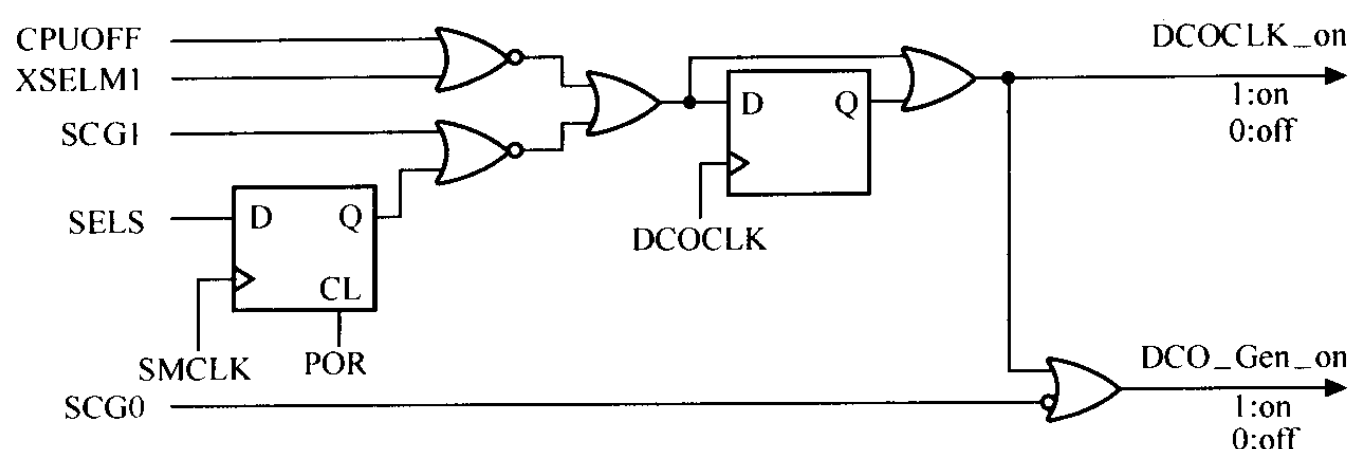


图 1-6 DCO 振荡器逻辑控制

(2) LFXT1 可以接低频振荡器, 典型的如 32.768kHz 的钟表振荡器, 此时振荡器不需要接负载电容。也可以接 450kHz~8MHz 的标准晶体振荡器, 此时振荡器需要接负载电容。图 1-7 为 LFXT1 振荡器逻辑控制图。

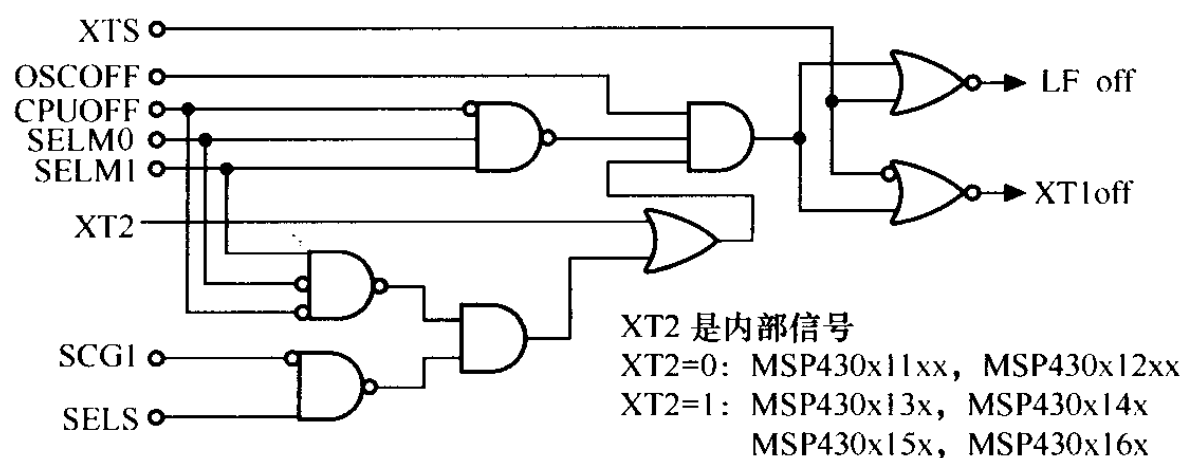


图 1-7 LFX1 振荡器逻辑控制

(3) XT2 接 450kHz~8MHz 的标准晶体振荡器, 此时振荡器要接负载电容, 不用时可以关闭。图 1-8 为 XT2 振荡器逻辑控制图。

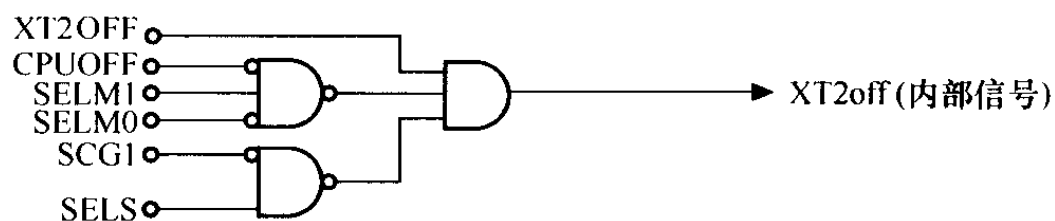


图 1-8 XT2 振荡器逻辑控制

通常低频振荡器用来降低能量消耗，例如使用电池的系统。高频振荡器用来对事件作出快速反应或者供 CPU 进行大量运算。

DCO 的振荡频率会受周围环境温度和 MSP430 工作电压的影响而产生变化, 并且同一型号的芯片所产生的频率也不相同。DCO 的调节功能可以改善它的性能。DCO 的调节分为以下 3 步:

- (1) 选择 BCSCTL1.RSELx 确定时钟的标称频率。
- (2) 选择 DCOCTL.DCOx 在标称频率基础上分段粗调。
- (3) 选择 DCOCTL.MODx 的值进行细调。

MSP430 定义了 3 种时钟信号，分别为：

(1) MCLK 系统主时钟。除了 CPU 运算使用此时钟信号外，外围模块也可以使用。MCLK 可以选择任何一个振荡器产生的时钟信号并进行 1、2、4、8 分频作为其信号源。

(2) SMCLK 系统子时钟。外围模块可以使用，并且在使用之前可以通过各模块的寄存器实现分频。SMCLK 可以选择任何一个振荡器产生的时钟信号并进行 1、2、4、8 分频作为其信号源。

(3) ACLK 辅助时钟。外围模块可以使用，并且在使用之前可以通过各模块的寄存器实现分频。ACLK 只能由 LFXT1 进行 1、2、4、8 分频作为其信号源。

PUC 结束时，MCLK 和 SMCLK 的信号源为 DCO，DCO 的振荡频率约为 800kHz（详见相关手册）。ACLK 的信号源为 LFXT1。

MSP430 内部含有晶体振荡器失效监测电路，监测 LFXT1（工作在高频模式）和 XT2 输出的时钟信号。当时钟信号丢失大约 50 μ s 时，监测电路捕捉到振荡器失效，如果 MCLK 信号来自 LFXT1 或者 XT2，那么 MSP430 自动把 MCLK 的信号源切换为 DCO，这样可以保证程序继续运行。MSP430 不对工作在低频模式的 LFXT1 进行监测。

1.5 低功耗模式

低功耗模式共有 5 种，为 LPM0~LPM4（LOW POWER MODE）。CPU 运行状态称为 AM（ACTIVE MODE）模式。图 1-9 显示出各种工作模式的耗电量，可以看出，AM 耗电最多，LPM4 耗电最省，仅为 0.1 μ A，几乎可以忽略不计。工作电压对于功耗也有影响，电压越低功耗也越低。

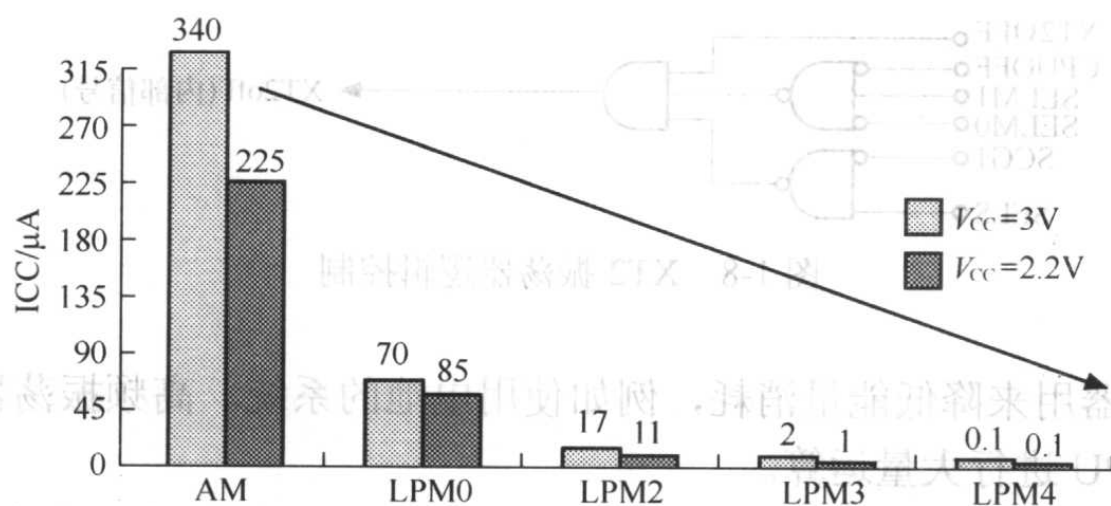


图 1-9 各模式耗电量比较

图 1-10 为 MSP430 工作模式的转换图。上电和外部复位信号产生 POR 信号，POR 信号会引起 PUC 信号。看门狗激活时，定时器溢出和安全键值错误也会引起 PUC 信号。PUC 信号结束后，MSP430 进入 AM 状态。在 AM 状态程序可以选择进入任何一种低功耗模式，然后在适当的时机，由外围模块的中断使 CPU 退出低功耗模式。

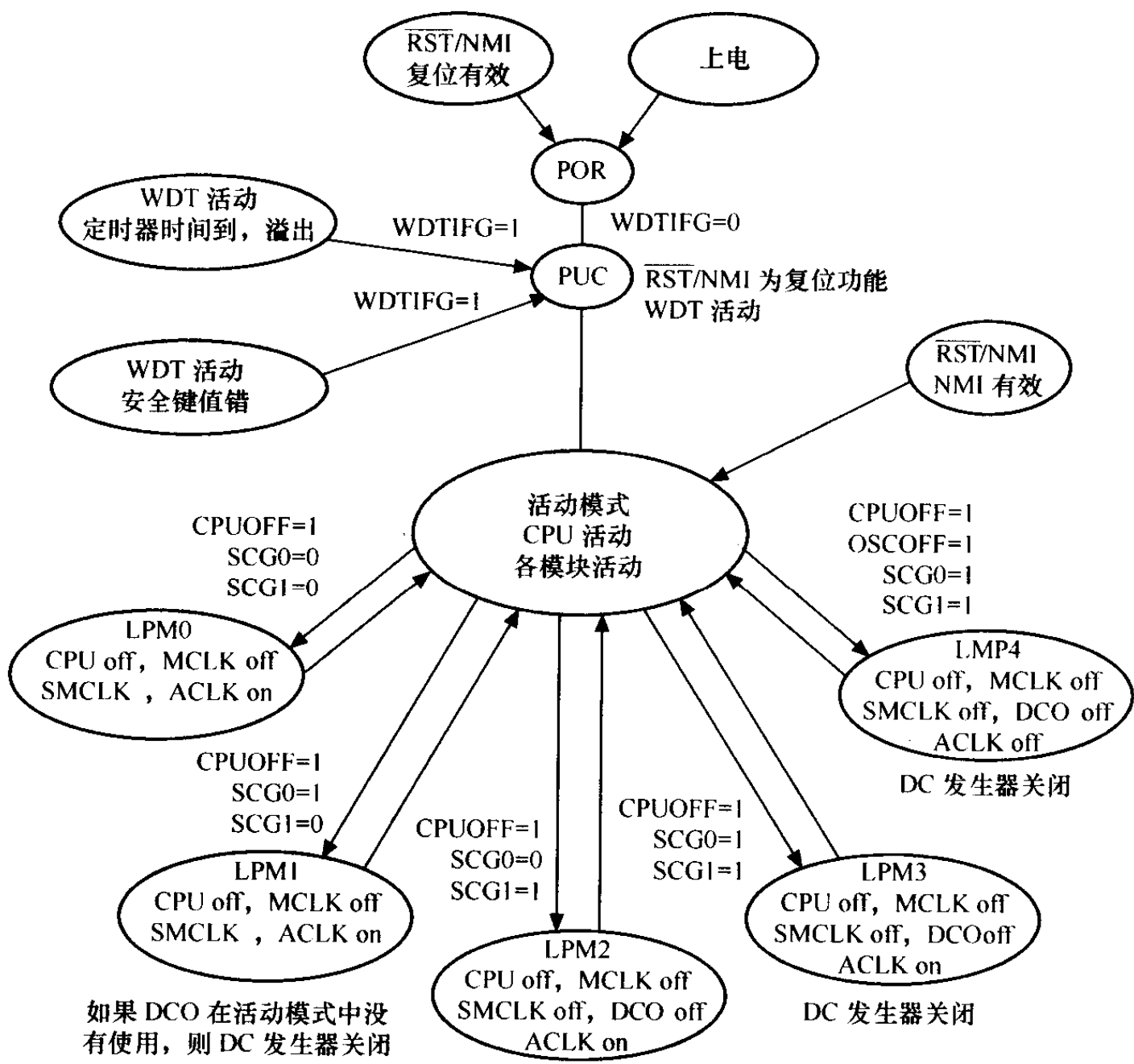


图 1-10 工作模式转换

表 1-1 显示了与选择工作模式有关的控制位。

表 1-1 工作模式选择

SCG1	SCG0	OSCOFF	CPUOFF	模式	CPU 和时钟状态
0	0	0	0	AM	CPU 活动 MCLK 活动 SMCLK 活动 ACLK 活动
0	0	0	1	LPM0	CPU 禁止 MCLK 禁止 SMCLK 活动 ACLK 活动
0	1	0	1	LPM1	CPU 禁止 MCLK 禁止 SMCLK 活动 ACLK 活动 在 AM 模式时若 DCO 没有用作 MCLK 或者 SMCLK, 则 DCO 的内部直流发生器关闭, DCO 禁止

续表

SCG1	SCG0	OSCOFF	CPUOFF	模式	CPU 和时钟状态
1	0	0	1	LPM2	CPU 禁止 MCLK 禁止 SMCLK 禁止 ACLK 活动 DCO 的内部直流发生器保留, DCO 禁止
1	1	0	1	LPM3	CPU 禁止 MCLK 禁止 SMCLK 禁止 ACLK 活动 DCO 的内部直流发生器关闭, DCO 禁止
1	1	1	1	LPM4	CPU 禁止 MCLK 禁止 SMCLK 禁止 ACLK 禁止 DCO 的内部直流发生器关闭, DCO 禁止

1.6 中 断

中断是 MSP430 微处理器的一大特色，有效地利用中断可以简化程序和提高执行效率。MS430 的中断比较多，几乎每个外围模块都能够产生中断，为 MSP430 针对事件（外围模块产生的中断）进行的编程打下了基础。MSP430 可以在没有事件发生时进入低功耗状态，事件发生时，通过中断唤醒 CPU，事件处理完毕后，CPU 再次进入低功耗状态。由于 CPU 的运算速度和退出低功耗状态的速度很快，所以，在很多应用中，CPU 大部分时间都能够处于低功耗状态，这是 MSP430 能够如此节省电能的重要原因之一。

MSP430 的中断分为：系统复位、不可屏蔽中断、可屏蔽中断。

系统复位前面已经作过介绍，其中断向量为 0xFFFFE。

不可屏蔽中断向量为 0xFFFC，产生不可屏蔽中断的原因如下：

(1) RST/NMI 管脚功能选择为 NMI 时，RST/NMI 管脚上产生一个上升沿或者下降沿（具体是上升沿还是下降沿由寄存器 WDTCTL 中的 NMIES 位决定）。NMI 中断可以用 WDTCTL 中的 NMIE 位屏蔽。需要注意的是，当 RST/NMI 管脚功能选择为 NMI 时，不要让 RST/NMI 管脚上的信号一直保持在低电平。原因是如果发生了 PUC，则 RST/NMI 管脚的功能被初始化为复位功能，而此时它上面的信号一直保持低电平，使 CPU 一直处于复位状态，不能正常工作。

(2) 振荡器失效中断允许时，振荡器失效。

(3) FLASH 存储器的非法访问中断允许时，对 FLASH 存储器进行了非法访问。

不可屏蔽中断可由各自的中断允许位禁止或打开。当一个不可屏蔽中断请求被接受时，