

文档编号: AN_056

上海东软载波微电子有限公司

应用笔记

HR7P 系列 MCU 向量中断模式

修订历史

版本	修订日期	修改概要
V1.0	2015-05-11	初版
V1.1	2015-07-28	统一修改公司名称、logo 及网址等

地 址：中国上海市龙漕路 299 号天华信息科技园 2A 楼 5 层

邮 编：200235

E-mail: support@essemi.com

电 话：+86-21-60910333

传 真：+86-21-60914991

网 址：<http://www.essemi.com/>

版权所有©

上海东软载波微电子有限公司

本资料内容为上海东软载波微电子有限公司在现有数据资料基础上慎重且力求准确无误编制而成，本资料中所记载的实例以正确的使用方法和标准操作为前提，使用方在应用该等实例时请充分考虑外部诸条件，上海东软载波微电子有限公司不担保或确认该等实例在使用方的适用性、适当性或完整性，上海东软载波微电子有限公司亦不对使用方因使用本资料所有内容而可能或已经带来的风险或后果承担任何法律责任。基于使本资料的内容更加完善等原因，上海东软载波微电子有限公司保留未经预告的修改权。使用方如需获得最新的产品信息，请随时用上述联系方式与上海东软载波微电子有限公司联系。

目 录

内容目录

第 1 章	概述	4
1.1	两种中断模式	4
1.2	默认中断模式	4
1.3	向量中断模式	4
1.3.1	中断向量分组	4
1.3.2	中断向量表配置	5
1.3.3	向量中断优先级	6
1.3.4	中断嵌套	6
1.3.5	向量中断使能	7
1.4	中断模式的编译	7
第 2 章	向量中断的使用说明	8
2.1	向量中断程序配置	8
2.2	向量中断函数语法	8
2.3	向量中断例程	9
2.3.1	自然优先级	9
2.3.2	中断嵌套	11
第 3 章	C 编译器设置	13
第 4 章	配置字设置	14

第1章 概述

1.1 两种中断模式

HR7P 系列 MCU 具有两种不同的中断处理机制：默认中断模式、向量中断模式。

1.2 默认中断模式

所有的 HR7P 系列 MCU 都支持默认中断模式。默认中断模式是所有中断向量的入口地址相同，通常是 0x0004 或者 0x0008。

在中断服务程序中，用户需通过软件判断各中断源的标志位及使能位的方式来确定中断源，并执行相应的中断处理程序。

在该模式下，各中断响应的优先级按中断服务程序中的中断源查询先后顺序排列，越先查询的，优先级越高。

该模式不支持中断嵌套。

1.3 向量中断模式

2T 架构（79 条指令集）的 HR7P 系列 MCU 支持向量中断模式，例如 HR7P9x、HR7P196 和 HR7P169 等芯片。向量中断模式是各个中断源可配置自己的中断向量入口地址。

在该模式下，根据触发的中断源，MCU 将自动执行相应中断向量入口处的服务程序，只有同组同优先级的中断，才需要用户通过软件方式判断中断源。

1.3.1 中断向量分组

2T 架构（79 条指令集）的 HR7P 系列 MCU 硬件中断源分成 8 组（IG0~IG7），软件中断除外。

不同的芯片硬件中断源不一样，分组方式也不一样，具体需要参考各芯片数据手册，这里以 HR7P169 为例介绍中断分组的配置：

中断组号	高低优先级选择位	中断名
IG0	IGP0	KINT
		T8NINT
IG1	IGP1	T8P1INT
		T8P2INT
		T8P3INT
IG2	IGP2	PINT0

		PINT1
IG3	IGP3	TXINT
		RXINT
IG4	IGP4	ADINT
IG5	IGP5	ACP1INT
		ACP2INT
		ACP3INT
		ACP4INT
		ACP5INT
IG6	IGP6	I2CINT
IG7	IGP7	—

1.3.2 中断向量表配置

MCU 系统配置为向量中断模式时，支持中断向量表。各中断组号对应一个中断向量入口地址。

中断向量的分配由 INTV<1:0>设置，分别对应固定的排列组合，具体见下表：

向量地址	INTV=00	INTV=01	INTV=10	INTV=11
0004H	软中断	软中断	软中断	软中断
0008H	IG0	IG0	IG4	IG7
000CH	IG1	IG1	IG5	IG6
0010H	IG2	IG6	IG2	IG5
0014H	IG3	IG7	IG3	IG4
0018H	IG4	IG4	IG0	IG3
001CH	IG5	IG5	IG1	IG2
0020H	IG6	IG2	IG6	IG1
0024H	IG7	IG3	IG7	IG0

1.3.3 向量中断优先级

向量中断模式下软中断固定为最高优先级，入口地址为 0x0004，

8 组硬件中断源的优先级可设置，硬件中断源的优先级设置分两步，即 IGPx 和 INTV<1:0>的设置。

通过选择 IGPx 可将各组硬件中断源设成高低两个优先级，高低优先级的仲裁分别通过 GIE 和 GIEL 中断使能位来使能。在执行低优先级中断时可嵌套响应高优先级中断。

当 IGPx 设置相同时，硬件中断源组根据 INTV<1:0>的设置，中断优先级按入口地址由低到高的自然优先级顺序排列，中断向量地址越低的优先级越高，此时不支持同级的中断嵌套。

举例：

若设置 INTV=0x01，INTP=0x59，试确定 T8NINT、T8P1INT、T8P2INT、PINT1、TXINT、ADINT、ACP3INT、I2CINT 等中断的优先级顺序。

向量地址	INTV=01	IGPx=0	IGPx=1	中断名	优先级排序
0008H	IG0		√	T8NINT	1
000CH	IG1	√		T8P1INT	5
				T8P2INT	5
0010H	IG6		√	I2CINT	2
0014H	IG7	√		—	—
0018H	IG4		√	ADINT	3
001CH	IG5	√		ACP3INT	6
0020H	IG2	√		PINT1	7
0024H	IG3		√	TXINT	4

即中断优先级顺序为：

T8NINT > I2CINT > ADINT > TXINT > T8P1INT = T8P2INT > ACP3INT > PINT1

其中 T8P1INT 和 T8P2INT 优先级相等，需要通过软件判断中断标志位。

1.3.4 中断嵌套

在执行低优先级分组内的中断服务程序时，如果触发了高优先级分组内的中断源，则系统将嵌套响应高优先级中断源，高优先级中断响应完再继续执行低优先级中断。

同优先级分组内的中断不支持相互嵌套响应。

1.3.5 向量中断使能

首先，通过配置字选择中断模式，因程序执行过程中无法动态改变。例如：HR7P169 配置字中的 INTVEN1 位。

其次，由软件设置寄存器的中断向量位。例如：HR7P169 的 INTG 寄存器中的 INTVEN0 位。

1.4 中断模式的编译

iDesigner 提供了两种中断程序的编译模式：默认中断模式、向量中断模式。

待编译项目所采用的中断编译模式是由目标 MCU 类型与编译参数确定。

对于不支持向量中断的 MCU，则不允许采用向量中断模式。

对于支持向量中断的 MCU，允许用户通过项目编译参数确定采用哪种中断编译模式。

同一项目中所有源文件及库文件的中断编译模式必须统一。

第2章 向量中断的使用说明

2.1 向量中断程序配置

各芯片的中断向量配置寄存器定义略有不同，具体参考各芯片数据手册。

这里以 HR7P169 为例介绍中断向量的配置：

1. 配置各中断源并使能中断
2. 配置中断全局寄存器 INTG 中 INTV<1:0>位，根据中断向量配置表选择各硬件中断源的中断向量
3. 配置中断优先级寄存器 INTP 设定各组中断的优先级, IGP0~IGP7 分别对应 IG0~IG7
4. 置 INTG 寄存器中的 INTVEN0 位使能向量中断模式
5. 使能高优先级中断位 GIE 和低优先级中断位 GIEL

2.2 向量中断函数语法

在向量中断模式下，低优先级分组的中断函数定义形式：

```
void <中断函数名> (void) interrupt_low <中断入口地址>  
{ [语句列表] }
```

在向量中断模式下，高优先级分组的中断函数定义形式：

```
void <中断函数名> (void) interrupt_high <中断入口地址>  
{ [语句列表] }
```

在使用向量中断函数时，需注意以下几点：

1. 中断函数形参列表和返回类型都必须为 void；
2. 中断函数是由 MCU 在产生中断时自动调用的，不允许被其他用户程序调用；
3. interrupt_low、interrupt_high 关键字必须出现在中断函数名与函数体之间；
4. 在向量中断模式下，中断函数定义时必须说明中断入口地址。一个项目中允许存在多个中断函数，但它们的中断入口地址不允许重复；
5. 中断服务函数的优先级分组及中断入口地址是静态确定的，不允许在程序执行过程中动态修改。编译器不会依据 INTV 寄存器的设置对中断服务函数的优先级分组及中断入口地址进行一致性校验；
6. 在中断函数定义之前必须加入 “#include <hic.h> ” 语句声明；
7. 中断现场保护、恢复代码（主要包括保存寄存器 A、PSW 及某些临时变量）是由编译器自动生成的，但编译器并不会保护用户定义的变量，用户需要注意自行保护。

2.3 向量中断例程

2.3.1 自然优先级

功能说明:

程序中有两个中断源 T8P1 和 T8N, 且都为同一优先级中断(同为高或同为低), INTV<1:0>=00, 则 T8P1 中断向量号为 0x000C, T8N 中断向量号为 0x0008, 根据 1.3.2 章节, T8N 的自然优先级高于 T8P1, 当 T8P1 和 T8N 的中断同时发生时, 程序会优先执行 T8N 的中断, 直到处理完退出, 才转而执行 T8P1 的中断。

程序使用内部 2MHz 晶振, 则定时器时钟源为 1μs, T8N 的定时时间为 255*2*1μs=510μs。当两个中断同时发生时, 优先执行自然优先级高的 T8N 中断。

T8P1 的中断服务程序中有 20ms 的延时程序, 这样在执行 T8P1 的中断服务程序时, 看不到 T8N 的嵌套执行。

C 例程:

```
#include <hic.h>
void Delay20ms(void);
//*****
//                                中断服务程序
//*****
void T8P1int() interrupt_low 0x000C
{
    if(T8P1IE && T8P1IF)
    {
        //进入 T8P 定时器中断
        T8P1IF = 0;           //清 T8P 中断标志
        T8P1 = T8P1+1;
        PA3 = ~PA3;           //输出端口取反
        Delay20ms();          //延时, 当同级中断发生时, 不会嵌套中断
    }
    else
    {
        T8P1IF = 0;
    }
}
void T8Nint() interrupt_low 0x0008
{
    if(T8NIE && T8NIF)
    {
        //进入 T8N 定时器中断
        T8NIF = 0;           //清 T8N 中断标志
        T8N = T8N + 200;      //重载 T8N 初值, 加上进入中断时间
        PA1 = ~PA1;          //输出端口取反
    }
    else
}
```

```

{
    T8NIF = 0;
}
}
//*****
//                                主程序
//*****
void main()
{
    ANSH = 0xFF;           //设置所有的模拟端口为数字端口
    ANSL = 0xFF;           //设置所有的模拟端口为数字端口
    PA = 0;                //设置 PA 端口输出低电平
    PAT = 0;               //设置 PA 端口为输出端口

    T8NC = 0x02;           //设置 T8N 为定时器模式，分频比为 1:8
    T8N = 6;               //设定定时器初值，2ms
    T8NPRE = 1;            //使能预分频器
    T8NIE = 1;             //使能 T8N 定时器中断
    T8P1IE = 1;            //T8P1 中断使能位

    IGP0 = 0;              //T8N 低优先级
    IGP1 = 0;              //T8P1 低优先级
    INTG = 0b11000100;    //bit1-0:INTV<1:0>中断向量表选择位
                           //bit2:INTVEN0, 0-默认中断模式；1-向量中断模式
                           //bit3:SOFTIF 0-无软件中断；1-有软件中断
                           //bit6:GIEL 0-禁止低优先级中断；1-使能低优先级中断
                           //bit7:GIE 0-禁止高优先级中断；1-使能高优先级中断

    T8NEN = 1;             //使能 T8N 定时器
    T8P1EN = 1;            //使能 T8P1 定时器
    while(1)
    {
        ;
    }
}
//*****
//                                延时程序
//*****
void Delay20ms(void)
{
    unsigned int Dcount;
    Dcount = 0x0682;       //若设主频 2MHz，指令周期 1us
    while(Dcount--);       //循环 1 次 12us（含 12 条指令），计数值计算为 20000/12=1666(0x682)
}

```

2.3.2 中断嵌套

功能说明:

程序中有两个中断源 T8P1 和 T8N, T8P1 为低优先级中断, 中断向量号为 0x000C, T8N 为高优先级中断, 中断向量号为 0x0008。

程序使用内部 2MHz 晶振, 则定时器时钟源为 1μs, T8N 的定时时间为 $255 \times 2 \times 1\mu\text{s} = 510\mu\text{s}$ 。T8P1 的中断服务程序中有 20ms 的延时程序, 这样在执行低优先级中断 T8P1 的中断服务程序时, 可以看到高优先级中断 T8N 的嵌套执行。

C 例程:

```
#include <hic.h>
void Delay20ms(void);
//*****
//                                中断服务程序
//*****
void T8P1int() interrupt_low 0x000C
{
    if(T8P1IE && T8P1IF)
    {
        //进入 T8P 定时器中断
        T8P1IF = 0;           //清 T8P 中断标志
        T8P1 = T8P1+1;
        PA3 = ~PA3;           //输出端口取反
        Delay20ms();          //延时, 当高优先级中断发生时, 会嵌套执行高优先级中断
    }
    else
    {
        T8P1IF = 0;
    }
}
void T8Nint() interrupt_high 0x0008
{
    if(T8NIE && T8NIF)
    {
        //进入 T8N 定时器中断
        T8NIF = 0;           //清 T8N 中断标志
        T8N = T8N + 200;      //重载 T8N 初值, 加上进入中断时间
        PA1 = ~PA1;          //输出端口取反
    }
    else
    {
        T8NIF = 0;
    }
}
```

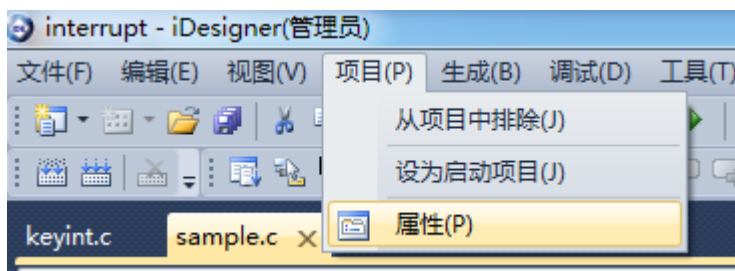
```
//*****  
//  
//主程序  
//*****  
void main()  
{  
    ANSH = 0xFF;           //设置所有的模拟端口为数字端口  
    ANSL = 0xFF;           //设置所有的模拟端口为数字端口  
    PA = 0;                //设置 PA 端口输出低电平  
    PAT = 0;               //设置 PA 端口为输出端口  
  
    T8NC = 0x02;           //设置 T8N 为定时器模式，分频比为 1:8  
    T8N = 6;               //设定定时器初值，2ms  
    T8NPRE = 1;           //使能预分频器  
    T8NIE = 1;            //使能 T8N 定时器中断  
    T8P1IE = 1;           //T8P1 中断使能位  
  
    IGP0 = 1;              //T8N 高优先级  
    IGP1 = 0;              //T8P1 低优先级  
    INTG = 0b11000100;    //bit1-0:INTV<1:0>中断向量表选择位  
                           //bit2:INTVEN0, 0-默认中断模式；1-向量中断模式  
                           //bit3:SOFTIF 0-无软件中断；1-有软件中断  
                           //bit6:GIEL 0-禁止低优先级中断；1-使能低优先级中断  
                           //bit7:GIE 0-禁止高优先级中断；1-使能高优先级中断  
  
    T8NEN = 1;             //使能 T8N 定时器  
    T8P1EN = 1;            //使能 T8P1 定时器  
    while(1)  
    {  
        ;  
    }  
}  
//*****  
//  
//延时程序  
//*****  
void Delay20ms(void)  
{  
    unsigned int Dcount;  
    Dcount = 0x0682;       //若设主频 2MHz，指令周期 1us  
    while(Dcount--);      //循环 1 次 12us（含 12 条指令），计数值计算为 20000/12=1666(0x682)  
}
```

第3章 C编译器设置

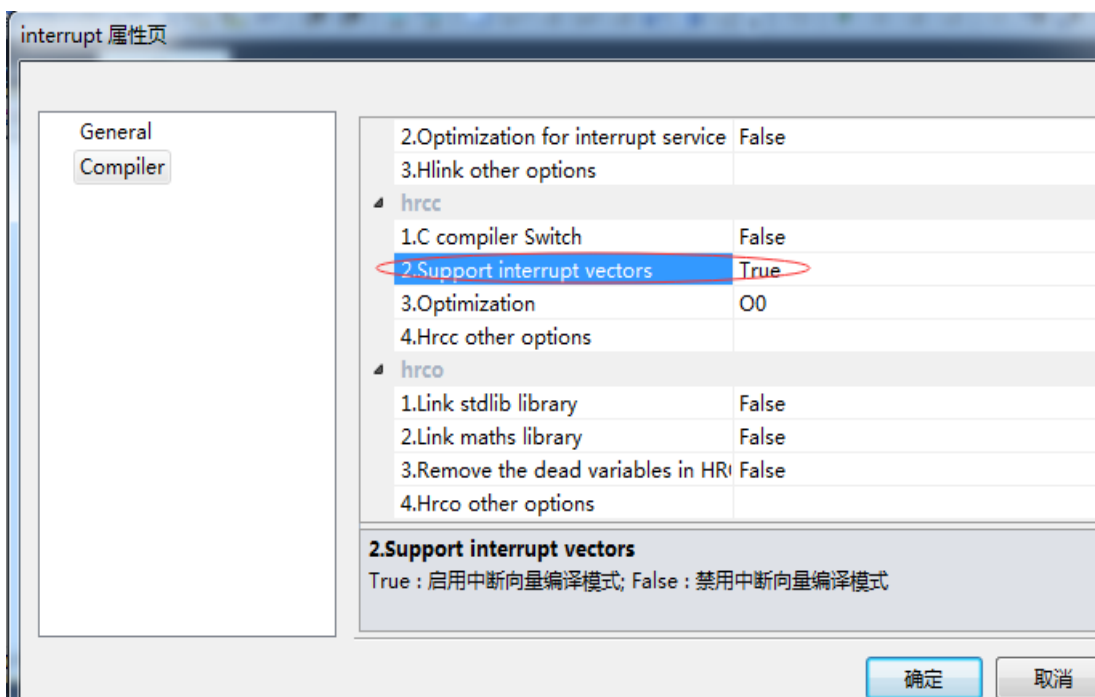
用户必须使用集成开发环境软件 iDesigner，以及工具链 HRCC V1.2.0.13 以上版本，才能支持向量中断的编译。

C 编译器编译时，通过“support interrupt vectors”选项来支持向量中断模式，具体操作如下：

1. 在 iDesigner 的菜单中打开项目→属性。



2. 在属性页的 Compiler 下，“Support interrupt vectors”选择“True”表示支持向量中断模式。



3. 点击“确定”。

第4章 配置字设置

在配置字中，需使能向量中断模式。

以 HR7P169 为例，配置字里 INTVEN1 选择“Vector Mode”，即向量中断模式。

