Application Note

Interrupt generated by NMI

For H8/3694 series

Contact Info:

Brazen Tek, Inc.
20121 Ventura Blvd.
Suite 310
Woodland Hills, CA 91364
USA
Tel/Fax: (818) 710-9262
E-mail: info@brazentek.com

January 2009

Introduction

Non-maskable interrupt request input pin (NMI) is generated by an external component and enables a 16-bit counter to be triggered.

Target Device

H8/3694 series

Contents

1. Specifications

1. Changing the state of NMI's external component generates an interrupt and a 16-bit counter is triggered.
2. The falling edge of the NMI pin is the source of its generated interrupt.
3. The 16-bit counter starts counting up while interrupt is generated.
4. An external component such as an LED can demonstrate the overflow state of the 16-bit counter.

2. Description of used Functions

NMI interrupt triggers the 16-bit counter. It is important to note that:

- The external component to the NMI pin generates the interrupt. By using NMIEG bit of the interrupt edge select register1 (IEGR1), falling or rising edge for non-maskable interrupt is determined.
- Regardless of CCR I bit value, NMI interrupt has highest priority.

3. Description of Operations

The processes for NMI interrupt performed by hardware and software are described in figure 3.1.

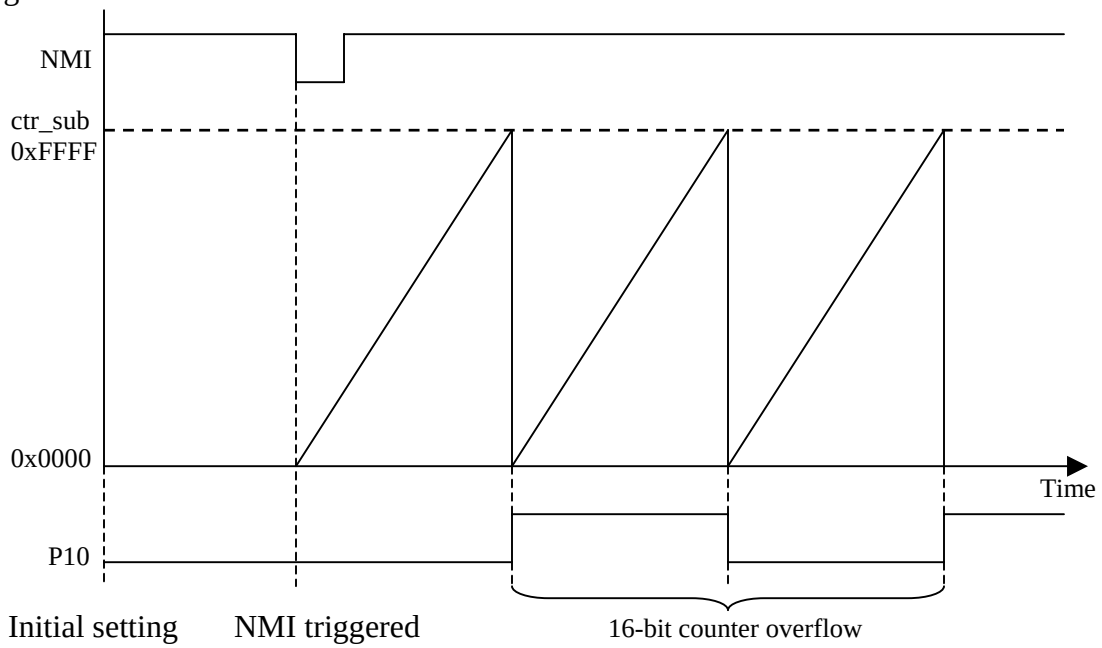


Figure 3.1 Hardware and Software operation

4. Description of Software

Module Name	Label	Function
Main routine	Main	Non-maskable interrupt edge selection, triggers 16-bit counter
State On	NMI	sets SWO NF to 1 in the NMI routine

Table 4.1 Description of firmware functions

The following describes the internal registers for this application.

Register	Name	Function	Address	Setting
PDR1	P10	Port data register 1 (port data register 10) P10 = 0: output level of pin P10 is low P10 = 1: output level of pin P10 is high	H'FDD4 Bit 0	0
PCR1	PCR10	Port control register 1 PCR10 = 1: I/O pin P10 functions as an Output pin	H'FFE4 Bit 0	0
IEGR1	NMIEG	Interrupt edge select register 1 NMIEG = 0: Set for falling edge of NMI	H'FFF2 Bit 7	0

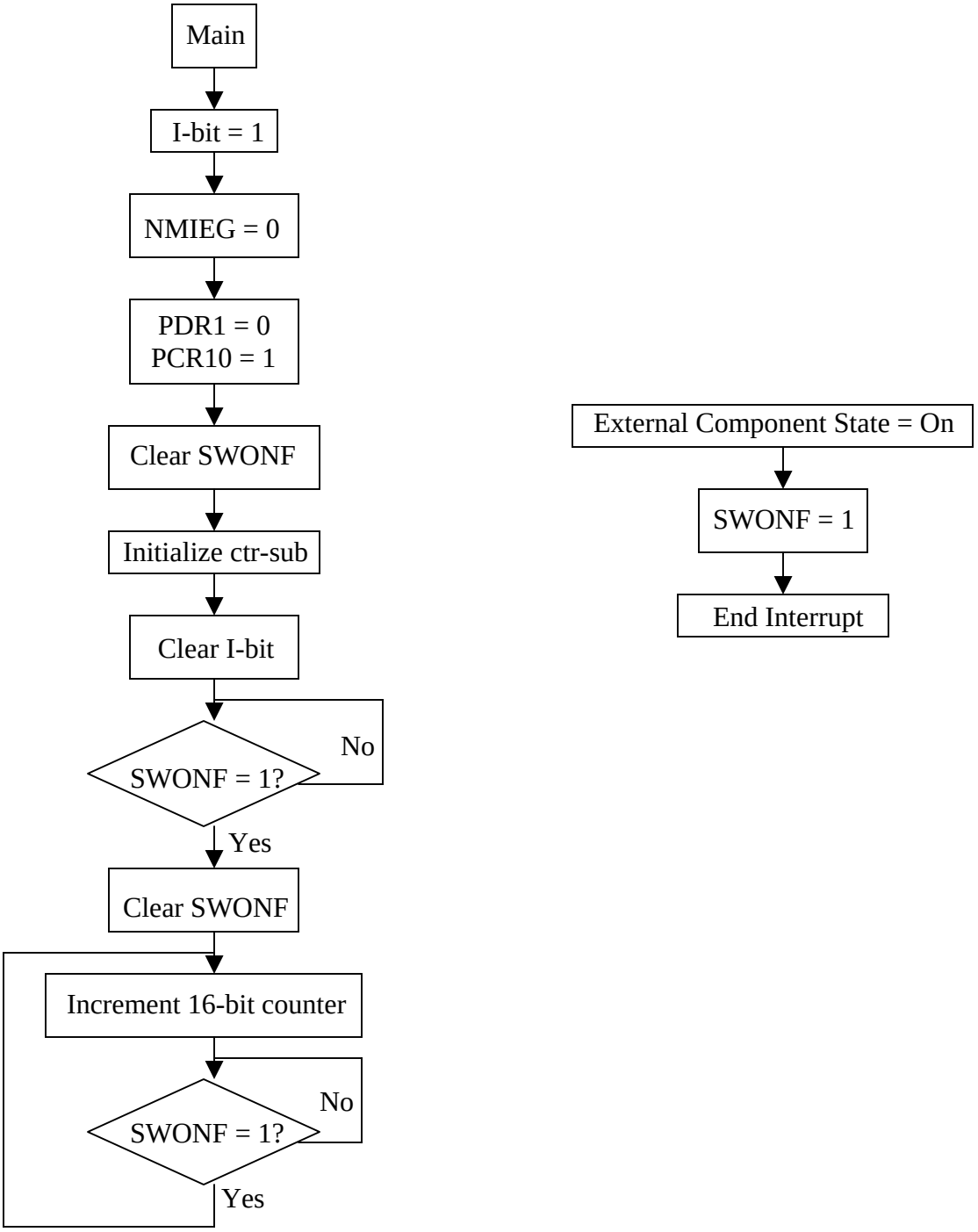
Table 4.2 Internal Registers

The following describes the RAM used for this application.

Label	Name	Function	Address	Used in
ctr_sub		indication of 16-bit counter overflow	H'FB80	Main routine
USRF	SWONF	external component's state: high or low	H'FB82 Bit 0	Main routine

Table 4.3 RAM used for functions

5. Flowchart


Main
I-bit = 1
NMIEG = 0
PDR1 = 0
PCR10 = 1
Clear SWONF
Initialize ctr-sub
Clear I-bit
SWONF = 1?
No
Yes
Clear SWONF
Increment 16-bit counter
SWONF = 1?
No
Yes
External Component State = On
SWONF = 1
End Interrupt

6. Program Listing

```
Initialize source
.EXPORT   _INIT
.IMPORT   _main
;
.SECTION  P,CODE
_INIT:
MOV.W     #H'FF80,R7
LDC.B     #B'10000000,CCR
JMP       @_main
;
.END
/***************************************************************************/
/* */
/* H8/300H Tiny Series -H8/3694- */
/* Application Note */
/* */
/* Interrupt generated by NMI*/
/* */
/* External Clock : 16MHz */
/* Internal Clock : 16MHz */
/* Sub Clock : 32.768kHz */
/* */
/*************************************************************************************************************/
#include    <C:\ch38\include\machine.h>
/*************************************************************************************************************/
/* Symbol Defnition */
/*************************************************************************************************************/
struct BIT {
unsigned char   b7:1;                                   /* bit7 */
unsigned char   b6:1;                                   /* bit6 */
unsigned char   b5:1;                                   /* bit5 */
unsigned char   b4:1;                                   /* bit4 */
unsigned char   b3:1;                                   /* bit3 */
unsigned char   b2:1;                                   /* bit2 */
unsigned char   b1:1;                                   /* bit1 */
unsigned char   b0:1;                                   /* bit0 */
};
#define           IEGR1  *(volatile unsigned char *)0xFFF2       /* Interrupt Edge select
Register1 */
#define IEGR1_BIT (*(struct BIT *)0xFFF2)               /* Interrupt Edge select Register1 */
#define  NMIEG IEGR1_BIT.b7                             /* NMI Edge Serect  */
#define PCR1 *(volatile unsigned char *)0xFFE4          /* Port Control Register 1 */
#define PCR1_BIT  (*(struct BIT *)0xFFE4)               /* Port Control Register 1 */
#define PCR10 PCR1_BIT.b0                               /* Port Control Register Bit 0  */
#define PDR1 *(volatile unsigned char *)0xFDD4          /* Port Data Register 1 */
```

```
#define PDR1_BIT (*(struct BIT *)0xFDD4)          /* Port Data Register 1 */


#define P10 PDR1_BIT.b0                           /* Port Data Register Bit 0 */

#pragma       interrupt    (NMI)

/*****************************************************************/
/* RAM define */
/*****************************************************************/
unsigned int ctr_sub;                             /* 16-Bit Up Counter */
unsigned char USRF;                               /* User Flag */
#define USRF_BIT    (*(struct BIT *)&USRF)
#define SWONF        USRF_BIT.b0                  /* Switch ON/OFF Judgment Flag */
extern void _INITSCT();
/*****************************************************************/
/* Function definition */
/*****************************************************************/
extern void INIT( void );                                 /* Stack Pointer Set */
extern void NMI( void );                                  /* NMI Interrupt Routine */
void main ( void );
/*****************************************************************/
/* Vector Address */
/*****************************************************************/
#pragma section     V1                                    /* VECTOR SECTION SET */
void (*const VEC_TBL1[])(void) = { /* 0x00 - 0x0f  */
INIT /* 00 Reset */
};
#pragma section     V2                                    /* VECTOR SECTOIN SET */
void (*const VEC_TBL2[])(void) = {                        /* 0x0e - 0x0f */
NMI                                                       /* 07 NMI */
};
#pragma section                                           /* Pause */
/*****************************************************************/
/* Main Program */
/*****************************************************************/
void main ( void )
{
_INITSCT();
NMIEG = 0;                                                /* NMI Falling Edge Interrupt */
P10 = 0;                                                  /* Port10 "0" Output */
PCR10 = 1;                                                /* Port10 Output */
SWONF = 0;                                                /* Clear External Comp.'s State */
ctr_sub = 0x0000;                                         /* Clear 16 Bit Counter */
while(SWONF != 1){                                        /* is SWONF = 1 ? */
;
}
SWONF = 0;                                                /* SWONF = 0 */

while(1){
do{
ctr_sub++;                                                /* Increment 16-bit Counter */
}while(ctr_sub != 0x0000);                                /* Is 16-bit Counter initialized? */
```

```
P10 = 0;                                        /* Output state Low */


}
else{
P10 = 1;                                        /* Output state high */
}
}
}
/***************************************************************/
/* NMI Interrupt */
/***************************************************************/
void NMI( void )
{
SWONF = 1;                                      /* SWONF = 1 */
}
```

Important Notes

1. This document is provided for reference purposes only. Brazen Tek neither makes warranties or representations with respect to the accuracy or completeness of the information contained in this document nor grants any license to any intellectual property rights.
2. Brazen Tek shall have no liability for damages or infringement of any intellectual property or other rights arising out of the use of any information in this document, including, but not limited to, product data, diagrams, charts, programs, algorithms, and application circuit examples.
3. This application note or the technology described in this document is not for the purpose of military applications such as the development of weapons of mass destruction or for the purpose of any other military use.
4. Brazen Tek assumes no liability whatsoever for any damages incurred as a result of errors or omissions in the information included in this document.
5. When using or otherwise relying on the information in this document, you should evaluate the information in light of the total system before deciding about the applicability of such information to the intended application. Brazen Tek makes no representations, warranties or guaranties regarding the suitability of its products for any particular application and specifically disclaims any liability arising out of the application and use of the information in this document.
12. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written approval from Brazen Tek.