Application Note

Address Break

For H8/36077 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

September 2008

Introduction

An address break interrupt is observed by turning an LED on and off.

Target Device

H8/36077 group

Contents

1. Specifications

i. An LED is to be connected to pin 23 (P10/TMOW) of the MCU.
ii. The setting of the LED is as follows: On if P10 is high and off if P10 is low.
iii. By having the MCU power (V_{cc}) pin and ground (V_{ss}) pin connected, see the state of LED at pin P10.
iv. By using address break interrupt, LED's state should differ from the previous state. This means if LED is off in step iii), with address break it should turn on and vice versa.

2. Description of Functions used

This application note is to describe the functionality of an address break interrupt. Figure 2 shows the handling of the address break interrupt.

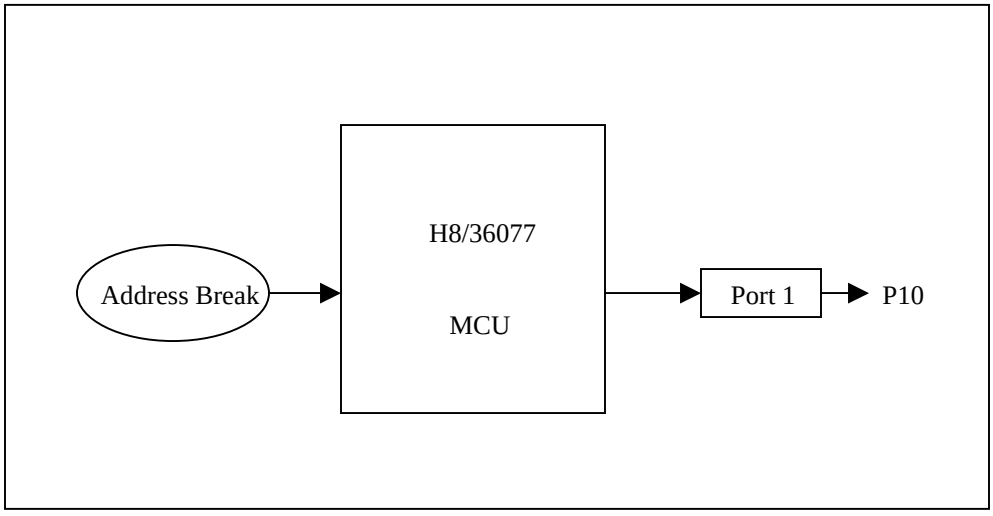


Figure 1 – Address break interrupt functionality

The address break block diagram is displayed in figure 3.The following are the ones used in the block diagram:

- Address break status register (ABRKSR)
  It consists of the address break interrupt flag and the address break interrupt enable bit. The address break interrupt request is enabled after the BAR is executed.
- Break address register (BAR: BARH and BARL)
  Being a 16-bit readable/writable register, this generates an address break interrupt by being set to H'025A. BARH is the higher eight bits and BARL is the lower eight bits.
- Break data registers (BDRH, BDRL)

16-bit read-write registers that set the data for generating an address break interrupt. BDRH is compared with the upper 8-bit data bus. BDRL is compared with the lower 8-bit data bus.

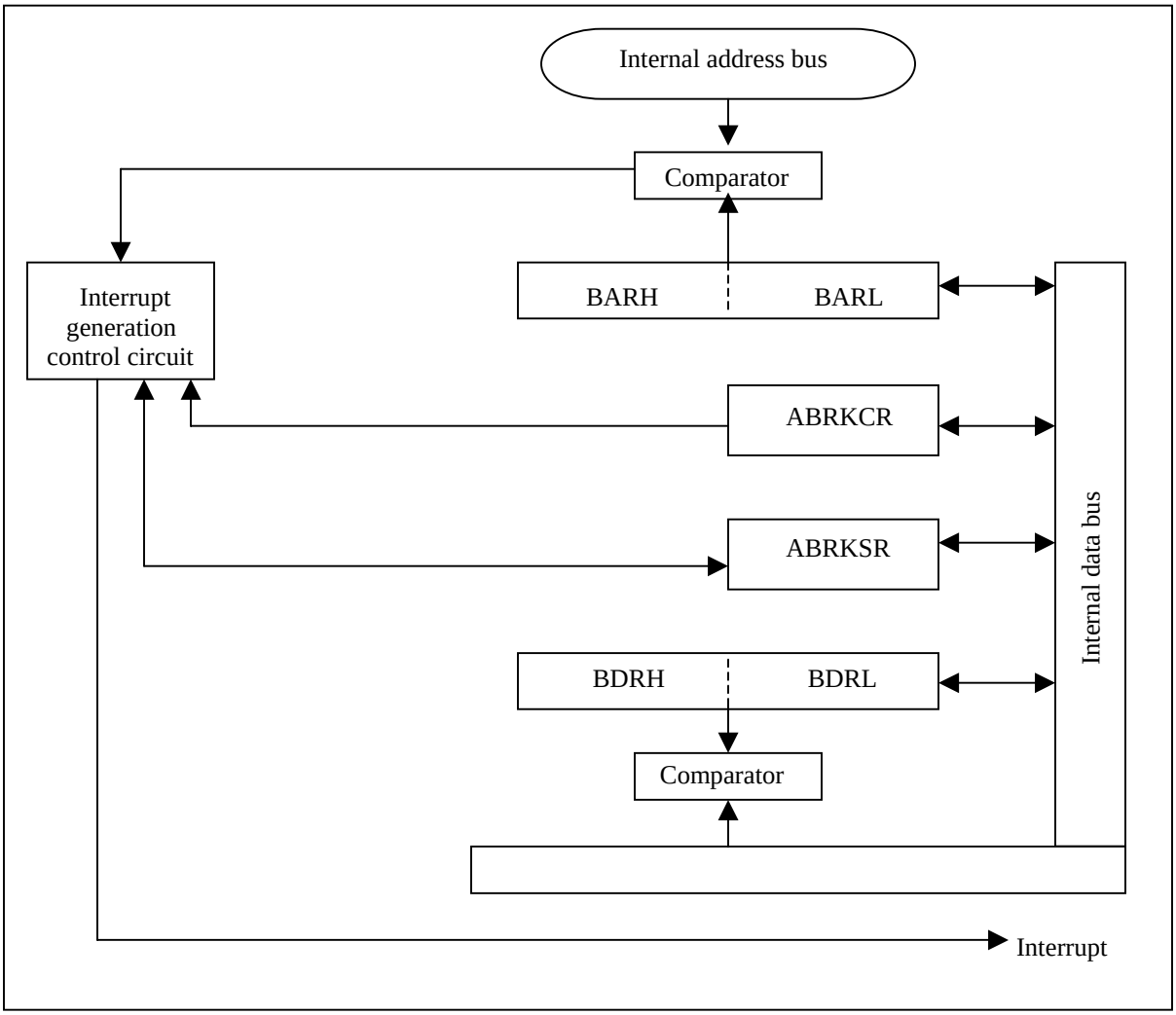


Figure 2 – Address break block diagram

3. Description of Operations

The hardware and firmware settings are described in figure 3. In this AN, the LED is first turned on until an interrupt request by address break is generated. It requests an address

break interrupt when the set break condition is satisfied. The interrupt request is not affected by the I bit of CCR.

i. Initial setting

```
PCR10 = high; (sets P10 as an output pin)
P10 = high; (turns the LED on)
ABRKCR = H'80;
BAR = H'011E;
ABIE = 1
```

ii. Program execution

```
address          source
   |               |
   |               |
H'0112           BAR = H'011E;
H'011A           ABIE = 1;
H'011E           P10 = 1;
H'0122
   |               |
   |               |
```

iii. Address break interrupt

```
P10 = 0; (Turn off the LED)
```

iv. End of address break

```
address          source
   |               |
   |               |
H'0112           BAR = H'011E;
H'011A           ABIE = 1;
H'011E           P10 = 1;
H'0122
   |               |
   |               |
```

Figure 3 - Address break interrupt operation

4. Description of Software

4.1 Description of Modules

Table 4.1 describes the firmware used in this App. Note.

Module Name	Label Name	Function
Main	main	Sets port 1 as output. Turns LED on P10 based on the setting
Address break interrupt	abint	Turns the LED off based on P10 setting

Table 4.1 Description of Modules

4.2 Description of Arguments

None are used for this AN.

4.3 Description of Internal Registers

Table 4.2 describes the internal registers used in this AN.

Register	Name	Function	Address	Bit	Setting
ABRKCR	RTINTE	RTE Interrupt Enable If RTINTE is cleared, Masks the interrupt If RTINTE is set to 1, Does not mask the interrupt	H'FFC8	7	1
	CSEL1	Condition select 1 CSEL=0, ASEL=0: Instruction execution cycle	H'FFC8	6	0
	CSEL0	Condition select 0 Sets the address break condition	H'FFC8	5	0
	ACMP2	Address compare 2 Does comparison between BAR and internal address bus. Performs 16-bit comparison	H'FFC8	4	0
	ACMP1	Address compare 1 Does comparison between BAR and internal address bus. Performs 16-bit comparison	H'FFC8	3	0

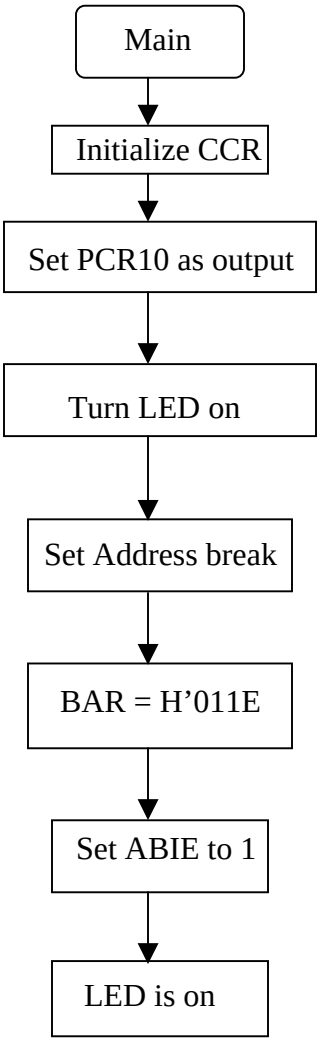
Register	Name	Function	Address	Bit	Setting
	ACMP0	Address compare 0 Does comparison between BAR and internal address bus. Performs 16-bit comparison	H'FFC8	2	0
	DCMP1	Data compare 1 Does not perform any comparison	H'FFC8	1	0
	DCMP0	Data compare 0 Does not perform any comparison	H'FFC8	0	0
ABRKSR	ABIF	Address break interrupt request flag If ABIF is set, condition is set in ABRKSR If ABIF is cleared, condition is cleared in ABRKSR	H'FFC9	7	0
	ABIE	Address break interrupt request enabled If ABIE is set, address break interrupt is enabled If ABIE is cleared, address break interrupt is masked	H'FFC9	6	1
BAR	BARH	Break address register (upper 8-bit)	H'FFCA		H'011E
	BARL	Break address register (lower 8-bit)	H'FFCB		H'011E
PDR1	P10	Port data register 1 (pin P10) If P10 is cleared, P10 is low at output If P10 is set to 1, P10 is high at output	H'FFD4	1	1
PCR1	PCR10	Port control register 1 If PCR10 is cleared, P10 is an input port If PCR10 is set to 1, P10 is an output port	H'FFE4	1	1

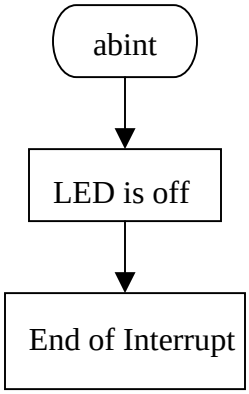
Table 4.2 - Internal Registers

4.4 Description of RAM

None used for this AN

5. Flowcharts





6. Program Listing

```
/************************************************************/
/* */
/* H8/36077 Group             */
/* Application Note           */
/*                            */
/* 'Address Break'            */
/*                            */
/* External Clock: 16MHz      */
/* Internal Clock: 16MHz      */
/* Sub Clock: 32.768kHz       */
/************************************************************/
#include <machine.h>
/************************************************************/
/* Symbol Definition */
/************************************************************/
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 ABRKCR *(volatile unsigned char *)H'FFC8 /* Address Break
Control Register */
#define ABRKSR_BIT (*(struct BIT *)H'FFC9) /* Address Break Status
Register */
#define ABIE ABRKSR_BIT.b6    /* Address Break Interrupt Enable */
#define BAR *(volatile unsigned short *)H'FFCA /* Break Address Register
H */
#define PCR1_BIT (*(struct BIT *)H'FFE4)  /* Port Control Register 1 */
#define PCR10 PCR1_BIT.b1 /* Port Control Register 10 */
#define PDR1_BIT (*(struct BIT *)H'FFD4)  /* Port Data Register 1 */
#define P10 PDR1_BIT.b1                   /* Port 10 */
#pragma interrupt (abint)
```

```
/************************************************************/
/* Function define */
/************************************************************/
void main ( void );
void abint( void );
/************************************************************/
/* Vector Address */
/************************************************************/
#pragma section V1 /* VECTOR SECTION SET */
void (*const VEC_TBL1[])(void) = {
main
};
```

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.
6. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written approval from Brazen Tek.