Order this document
by AN1844/D

AN1844

Using a 68HC908MR32 in Place of a 68HC908MR24

By Bill Lucas
Freescale Microcontroller Division
Austin, Texas

Introduction

This application note documents the differences between the 68HC908MR24 (MR24) and the 68HC908MR32 (MR32). The information is intended to help users migrate from the MR24 to the MR32.

The new features of the MR32 and differences between the two are:

- MR32 has 32 Kbytes of FLASH memory; the MR24 has 24 Kbytes of FLASH.
- FLASH programming and block protection schemes are different on the MR32 and the MR24.
- A FLASH memory low-power mode has been added to the MR32.
- A break module has been added to the MR32 and is used to aid in user software debug.
- MR32 has an optional, user-enabled, low-power STOP instruction.
- Monitor mode enhancements on the MR32 aid in initial FLASH memory programming.

Background

The 68HC908MR32 is an improved version of the 68HC908MR24 and was launched to enhance Motorola's family of products designed for motor control applications.

The MR32 is a mechanical drop-in replacement for the MR24.

From a software point of view, the MR24 and MR32 have a few address differences. These address changes primarily revolve around the FLASH memory array.

Table 1 shows a thumbnail sketch of the functional and address differences between the two devices.

Table 1. MR24 and MR32 Functional/Address Differences

Function	68HC908MR24	68HC908MR32
CONFIG register	Bit 1 not used	Bit 1 is STOP enable
Break module	None	Included
FLASH mapping	$A000–$FDFF	$8000–$FDFF
FLASH erased state	Erased bit = 0	Erased bit = 1
FLASH page size (bytes)	8	128
FLASH minimum erase size (bytes)	64	128
FLASH erase size (bytes)	64/512/16k/24 k	128/32 k
FLASH program row size (bytes)	8	64
FLASH block protect size (bytes)	4 k, 8 k, 16 k, 24 k	128, 256, 384...32 k
FLASH standby low power mode	No	Yes
SIM break status register SBSR	Unimplemented	$FE00
SIM break flag control register SBFCR	Unimplemented	$FE03
SIM break address register high BRKH	Unimplemented	$FE0C
SIM break address register low BRKL	Unimplemented	$FE0D
SIM status and control register SBKSCR	Unimplemented	$FE0E
FLASH block protect register FLBPR	$FF80	$FF7E

From a Hardware Point of View

The physical footprint of the MR32 is the same as that of the MR24.

The MR24's FLASH memory module has been replaced on the MR32 with a memory module that is designed differently. Because of that change, FLASH control registers reside in different addresses. For example, on the MR24, an erased FLASH bit reads as a logic 0 and a programmed bit reads as a logic 1. On the MR32, an erased FLASH bit reads as a logic 1 and a programmed bit reads as a logic 0.

The MR32 will enter monitor mode when the reset vector ($FFFE–$FFFF) is erased, eliminating the requirement of applying V_{HI} to the IRQ input and specific logic levels on three port C pins. This feature simplifies initial in-circuit programming of the device.

A break module has been added to the MR32. Its primary purpose is to act as an aid for software debug. The break module can generate a break interrupt that stops normal program flow at a defined address to enter a background program.

From a Software Point of View

A number of control and data register addresses have changed from the MR24 to the MR32. For convenience, input/output (I/O) include files for the MR32 for both C and assembler languages are included in this application note.

On the MR24, the STOP instruction was disabled in the CPU (central processor unit) and bit 1 of the mask option register (MOR) was not used. The low-power STOP instruction has been added to the MR32. This instruction is enabled by setting bit 1 in the write-once MOR. The default for the STOP instruction is STOP disabled.

The addition of a break module to the MR32 is for use primarily by development tools. It is also accessible to users allowing a break upon an address match as a possible aid to debugging programs.

Features of the break module include:

- Accessible I/O registers during the break interrupt
- CPU-generated break interrupts
- Software-generated break interrupts
- COP disabling during break interrupts

When the internal address bus matches the value written in the break address registers, the break module issues a breakpoint signal to the CPU. The CPU then loads the instruction register with a software interrupt instruction (SWI) after completion of the current CPU instruction. The program counter vectors to $FFFC and $FFFD ($FEFC and $FEFD in monitor mode).

Either of these two events can cause a break interrupt to occur:

- A CPU-generated address (the address in the program counter) matches the contents of the break address registers.
- Software writes a logic 1 to the BRKA bit in the break status and control register.

When a CPU-generated address matches the contents of the break address registers, the break interrupt begins after the CPU completes its current instruction. A return-from-interrupt instruction (RTI) in the break routine ends the break interrupt and returns the MCU to normal operation.

Address Changes

The addresses listed here have changed from the MR24 to the MR32, while the remainder of the I/O and status/control addresses stayed the same. The bits within these registers also remain the same.

The new addresses for I/O and status/control addresses on the MR32 are listed here.

NOTE: *The following registers in the MR32 have different addresses and/or register bits from the MR24. New registers also have been added to the MR32. Refer to the 68HC908MR32 Advance Information, Freescale document order number MC68HC908MR32/D, for more detailed information.*

FLASH Module

$FF7E FLASH Block Protect Register (FLBPR)

The block protect register (FLBPR) is implemented as a byte within the FLASH memory, and therefore can be written only during a programming sequence of the FLASH memory. The value in this register determines the starting location of the protected range within the FLASH memory.

Address: $FF7E

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	BPR7	BPR6	BPR5	BPR4	BPR3	BPR2	BPR1	BPR0
Reset:	U	U	U	U	U	U	U	U

U = Unaffected by reset. Initial value from factory is 1.

Figure 1. FLASH Block Protect Register (FLBPR)

BPR[7:0] — FLASH Block Protect Bits

These eight bits represent bits [14:7] of a 16-bit memory address. Bit 15 is logic 1 and bits [6:0] are logic 0s.

The resultant 16-bit address is used for specifying the start address of the FLASH memory for block protection. The FLASH is protected from this start address to the end of FLASH memory, at $FFFF. With this mechanism, the protect start address can be XX00 and XX80 (128 bytes page boundaries) within the FLASH memory.

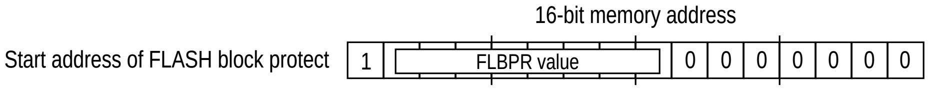

Figure 2. FLASH Block Protect Start Address

Examples of protect start address:

BPR[7:0]	Start of Address of Protect Range
$00	The entire FLASH memory is protected.
$01 (**0000 0001**)	$8080 (1**000 0000 1**000 0000)
$02 (**0000 0010**)	$8100 (1**000 0001 0**000 0000)
and so on . . .	
$FE (**1111 1110**)	$FF00 (1**111 1111 0**000 0000)
$FF	The entire FLASH memory is not protected.

Note: The end address of the protected range is always $FFFF.

By programming the block protect bits, a portion of the memory will be locked so that no further erase or program operations may be performed.

Break Module

$FE00 SIM Break Status Register (SBSR)

The SIM break status register (SBSR) contains a flag indicating when a break causes an exit from wait mode. The flag is useful in applications requiring a return to wait mode after exiting from a break interrupt.

Address: $FE00

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	R	R	R	R	R	R	BW	R
Write:							Note	
Reset:	0	0	0	0	0	0	0	0

Note: Writing a logic 0 clears BW.

R = Reserved

Figure 3. SIM Break Status Register (SBSR)

BW — Break Wait Bit

This read/write bit is set when a break interrupt causes an exit from wait mode. Clear BW by writing a logic 0 to it. Reset clears BW.

1 = Break interrupt during wait mode

0 = No break interrupt during wait mode

BW can be read within the break interrupt routine. The user can modify the return address on the stack by subtracting 1 from it. The example code that follows works if the H register was stacked in the break interrupt routine.

Execute this code at the end of the break interrupt routine.

```
HIBYTE  EQU   5
LOBYTE  EQU   6
                                 ; If not BW, do RTI
        BRCLR BW,BSR, RETURN     ; See if wait mode or stop
                                 ; mode was exited by break.
        TST   LOBYTE,SP          ; If RETURNLO is not 0,
        BNE   DOLO               ; then just decrement low byte.
        DEC   HIBYTE,SP          ; Else deal with high byte also.
DOLO    DEC   LOBYTE,SP          ; Point to WAIT/STOP opcode.
RETURN  PULH                     ; Restore H register.
        RTI
```

$FE03 SIM Break Flag Control Register (SBFCR)

The SIM break flag control register (SBFCR) contains a bit that enables software to clear status bits while the MCU is in a break state.

Address: $FE03

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	BCFE	R	R	R	R	R	R	R
Reset:	0							

R = Reserved

Figure 4. SIM Break Flag Control Register (SBFCR)

BCFE — Break Clear Flag Enable Bit

This read/write bit enables software to clear status bits by accessing status registers while the MCU is in a break state. To clear status bits during the break state, the BCFE bit must be set.

1 = Status bits clearable during break
0 = Status bits not clearable during break

$FE0C SIM Break Address Register High/Low (BRKH/BRKL)

The break address registers (BRKH and BRKL) contain the high and low bytes of the desired breakpoint address. Reset clears the break address registers.

Address: $FE0C

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	Bit 15	14	13	12	11	10	9	Bit 8
Reset:	0	0	0	0	0	0	0	0

Figure 5. SIM Break Address Register High (BRKH)

Address: $FE0D

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0
Reset:	0	0	0	0	0	0	0	0

Figure 6. SIM Break Address Register Low (BRKL)

$FE0E SIM Break Status and Control Register (SBKSCR)

The break status and control register (BRKSCR) contains break module enable and status bits.

Address: $FE0E

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	BRKE	BRKA	0	0	0	0	0	0
Write:			▨	▨	▨	▨	▨	▨
Reset:	0	0	0	0	0	0	0	0

▨ = Unimplemented

Figure 7. Break Status and Control Register (BRKSCR)

BRKE — Break Enable Bit

This read/write bit enables breaks on break address register matches. Clear BRKE by writing a logic 0 to bit 7. Reset clears the BRKE bit.

1 = Breaks enabled on 16-bit address match
0 = Breaks disabled on 16-bit address match

BRKA — Break Active Bit

This read/write status and control bit is set when a break address match occurs. Writing a logic 1 to BRKA generates a break interrupt. Clear BRKA by writing a logic 0 to it before exiting the break routine. Reset clears the BRKA bit.

1 = When read, break address match
0 = When read, no break address match

C and Assembler Include Files for the 68HC908MR32

I/O Definitions for the MR32

These definitions are for use with Cosmic Software’s MC68HC08 compiler. For more information, go to the company’s Web site at www.cosmic.com. Be aware that different compilers may require slight syntax changes.

C Header File for the MR32

```
I/O PORTS
@tiny volatile  char     PORTA @0x00;        /* port A                           */
@tiny volatile  char     PORTB @0x01;        /* port B                           */
@tiny volatile  char     PORTC @0x02;        /* port C                           */
@tiny volatile  char     PORTD @0x03;        /* port D                           */
@tiny volatile  char     PORTE @0x08;        /* port E                           */
@tiny volatile  char     PORTF @0x09;        /* port F                           */
@tiny           char     DDRA  @0x04;        /* data direction port A            */
@tiny           char     DDRB  @0x05;        /* data direction port B            */
@tiny           char     DDRC  @0x06;        /* data direction port C            */
@tiny           char     DDRD  @0x07;        /* data direction port D            */
@tiny           char     DDRE  @0x0c;        /* data direction port E            */
@tiny           char     DDRF  @0x0d;        /* data direction port F            */

TIMER A

@tiny volatile  char     TASC   @0x0e;       /* timer A status/ctrl register     */
@tiny volatile  int      TACNT  @0x0f;       /* timer A counter register         */
@tiny volatile  char     TACNTH @0x0f;       /* timer A counter high             */
@tiny volatile  char     TACNTL @0x10;       /* timer A counter low              */
@tiny volatile  int      TAMOD  @0x11;       /* timer A modulo register          */
@tiny volatile  char     TAMODH @0x11;       /* timer A modulo high              */
@tiny volatile  char     TAMODL @0x12;       /* timer A modulo low               */
@tiny volatile  char     TASC0  @0x13;       /* timer A channel 0 status/ctrl    */
@tiny volatile  int      TACH0  @0x14;       /* timer A channel 0 register       */
@tiny volatile  char     TACH0H @0x14;       /* timer A channel 0 high           */
@tiny volatile  char     TACH0L @0x15;       /* timer A channel 0 low            */
@tiny volatile  char     TASC1  @0x16;       /* timer A channel 1 status/ctrl    */
@tiny volatile  int      TACH1  @0x17;       /* timer A channel 1 register       */
@tiny volatile  char     TACH1H @0x17;       /* timer A channel 1 high           */
@tiny volatile  char     TACH1L @0x18;       /* timer A channel 1 low            */
@tiny volatile  char     TASC2  @0x19;       /* timer A channel 2 status/ctrl    */
@tiny volatile  int      TACH2  @0x1a;       /* timer A channel 2 register       */
@tiny volatile  char     TACH2H @0x1a;       /* timer A channel 2 high           */
@tiny volatile  char     TACH2L @0x1b;       /* timer A channel 2 low            */
@tiny volatile  char     TASC3  @0x1c;       /* timer A channel 3 status/ctrl    */
@tiny volatile  int      TACH3  @0x1d;       /* timer A channel 3 register       */
```

```
@tiny volatilec har      TACH3H @0x1d;       /* timer A channel 3 high          */
@tiny volatile  char     TACH3L @0x1e;       /* timer A channel 3 low           */

//     OPTION REGISTER

@tiny         char     MOR   @0x1f;          /* Mask Option Write-Once Register  */

//     PWM

@tiny           char     PCTL1   @0x20;      /* PWM control register 1          */
@tiny           char     PCTL2   @0x21;      /* PWM control register 2          */

@tiny           char     FCR     @0x22;      /* Fault control register          */
@tiny volatile  char     FSR     @0x23;      /* Fault Status register           */
@tiny volatile  char     FTACK   @0x24;      /* Fault acknowledge register      */

@tiny           char     PWMOUT  @0x25;      /* PWM output control register     */
@tiny volatile  int      PCNT    @0x26;      /* PWM counter register            */
@tiny volatile  char     PCNTH   @0x26;      /* PWM counter register high       */
@tiny volatile  char     PCNTL   @0x27;      /* PWM counter register low        */
@tiny volatile  int      PMOD    @0x28;      /* PWM counter Modulo register     */
@tiny volatile  char     PMODH   @0x28;      /* PWM counter Modulo register high */
@tiny volatile  char     PMODL   @0x29;      /* PWM counter Modulo register low  */
@tiny volatile  int      PVAL1   @0x2a;      /* PWM 1 value register            */
@tiny volatile  char     PVAL1H  @0x2a;      /* PWM 1 value register high       */
@tiny volatile  char     PVAL1L  @0x2b;      /* PWM 1 value register low        */
@tiny volatile  int      PVAL2   @0x2c;      /* PWM 2 value register            */
@tiny volatile  char     PVAL2H  @0x2c;      /* PWM 2 value register high       */
@tiny volatile  char     PVAL2L  @0x2d;      /* PWM 2 value register low        */
@tiny volatile  int      PVAL3   @0x2e;      /* PWM 3 value register            */
@tiny volatile  char     PVAL3H  @0x2e;      /* PWM 3 value register high       */
@tiny volatile  char     PVAL3L  @0x2f;      /* PWM 3 value register low        */
@tiny volatile  int      PVAL4   @0x30;      /* PWM 4 value register            */
@tiny volatile  char     PVAL4H  @0x30;      /* PWM 4 value register high       */
@tiny volatile  char     PVAL4L  @0x31;      /* PWM 4 value register low        */
@tiny volatile  int      PVAL5   @0x32;      /* PWM 5 value register            */
@tiny volatile  char     PVAL5H  @0x32;      /* PWM 5 value register high       */
@tiny volatile  char     PVAL5L  @0x33;      /* PWM 5 value register low        */
@tiny volatile  int      PVAL6   @0x34;      /* PWM 6 value register            */
@tiny volatile  char     PVAL6H  @0x34;      /* PWM 6 value register high       */
@tiny volatile  char     PVAL6L  @0x35;      /* PWM 6 value register low        */

@tiny volatile  char     DEADTM  @0x36;      /* Dead Time Write-once register     */
@tiny volatile  char     DISMAP  @0x37;      /* PWM Disable Mapping Write-once reg.*/

//     SCI section

@tiny           char     SCC1    @0x38;      /* SCI control register 1          */
@tiny           char     SCC2    @0x39;      /* SCI control register 2          */
@tiny           char     SCC3    @0x3a;      /* SCI control register 3          */
@tiny volatile  char     SCS1    @0x3b;      /* SCI status register 1           */
@tiny volatile  char     SCS2    @0x3c;      /* SCI status register 2           */
@tiny volatile  char     SCDR    @0x3d;      /* SCI data register               */
```

```
@tiny           char    SCBR    @0x3e;      /* SCI baud rate                    */
//     INTERRUPT

@tiny volatile  char    ISCR    @0x3F;      /* IRQ control/status register      */

//     A/D

@tiny volatile  char    ADSCR   @0x40;      /* ADC status and control register  */
@tiny volatile  int     ADR     @0x41;      /* ADC data register                */
@tiny volatile  char    ADRH    @0x41;      /* ADC data register HIGH           */
@tiny volatile  char    ADRL    @0x42;      /* ADC data register LOW            */
@tiny           char    ADCLK   @0x43;      /* ADC clock register               */

//     SPI

@tiny           char    SPCR    @0x44;      /* SPI control register             */
@tiny volatile  char    SPSCR   @0x45;      /* SPI control/status register      */
@tiny volatile  char    SPDR    @0x46;      /* SPI data register                */

//     TIMER B

@tiny volatile  char    TBSC    @0x51;      /* timer B status/ctrl register     */
@tiny volatile  int     TBCNT   @0x52;      /* timer B counter register         */
@tiny volatile  char    TBCNTH  @0x52;      /* timer B counter high             */
@tiny volatile  char    TBCNTL  @0x53;      /* timer B counter low              */
@tiny volatile  int     TBMOD   @0x54;      /* timer B modulo register          */
@tiny volatile  char    TBMODH  @0x54;      /* timer B modulo high              */
@tiny volatile  char    TBMODL  @0x55;      /* timer B modulo low               */
@tiny volatile  char    TBSC0   @0x56;      /* timer B channel 0 status/ctrl    */
@tiny volatile  int     TBCH0   @0x57;      /* timer B channel 0 register       */
@tiny volatile  char    TBCH0H  @0x57;      /* timer B channel 0 high           */
@tiny volatile  char    TBCH0L  @0x58;      /* timer B channel 0 low            */
@tiny volatile  char    TBSC1   @0x59;      /* timer B channel 1 status/ctrl    */
@tiny volatile  int     TBCH1   @0x5a;      /* timer B channel 1 register       */
@tiny volatile  char    TBCH1H  @0x5a;      /* timer B channel 1 high           */
@tiny volatile  char    TBCH1L  @0x5b;      /* timer B channel 1 low            */

//      PLL

@tiny volatile  char    PCTL    @0x5c;      /* PLL control register             */
@tiny volatile  char    PBWC    @0x5d;      /* PLL bandwidth register           */
@tiny           char    PPG     @0x5e;      /* PLL programming register         */

//      SIM/FLASH/LVI/COP

@near  volatile  char   SBSR    @0xfe00;    /* SIM break status register        */
@near  volatile  char   SRSR    @0xfe01;    /* SIM reset status register        */
@near  volatile  char   SBFCR   @0xfe03;    /* SIM break control register       */
@near  volatile  char   FLCR    @0xfe08;    /* FLASH control register           */
@near  volatile  int    BRK     @0xfe0c;    /* SIM break address register       */
@near  volatile  char   BRKH    @0xfe0c;    /* SIM break address register high  */
@near  volatile  char   BRKL    @0xfe0d;    /* SIM break address register low   */
@near  volatile  char   BRKSCR  @0xfe0e;    /* SIM break stat and control register */
```

```
@near  volatile char    LVISCR  @0xfe0f;    /* LVI status register and control  */
@near  volatile  char   FLBPR   @0xff7e;    /* FLASH BLOCK PROTECT register     */
@near  volatile char    COPCTL  @0xffff;    /* COP control register             */
```

Assembler Include File for the MR32

I/O Definitions for 68HC908MR32

```
*              Bit labels:                                                  *
*                   BIT. is a mask for LDA, STA                             *
*                   BIT is a bit number for bset, bclr, brset, brclr        *
*                                                                           *
PORTA          equ      $00;                 ; I/O PORT A
PORTA7         equ      $7;                  ; PORTA BIT 7 I/O value
PORTA6         equ      $6;                  ; PORTA BIT 6 I/O value
PORTA5         equ      $5;                  ; PORTA BIT 5 I/O value
PORTA4         equ      $4;                  ; PORTA BIT 4 I/O value
PORTA3         equ      $3;                  ; PORTA BIT 3 I/O value
PORTA2         equ      $2;                  ; PORTA BIT 2 I/O value
PORTA1         equ      $1;                  ; PORTA BIT 1 I/O value
PORTA0         equ      $0;                  ; PORTA BIT 0 I/O value
;
PORTA7.        equ      $80;                 ; PORTA BIT 7 I/O value
PORTA6.        equ      $40;                 ; PORTA BIT 6 I/O value
PORTA5.        equ      $20;                 ; PORTA BIT 5 I/O value
PORTA4.        equ      $10;                 ; PORTA BIT 4 I/O value
PORTA3.        equ      $8;                  ; PORTA BIT 3 I/O value
PORTA2.        equ      $4;                  ; PORTA BIT 2 I/O value
PORTA1.        equ      $2;                  ; PORTA BIT 1 I/O value
PORTA0.        equ      $1;                  ; PORTA BIT 0 I/O value
;
PORTB          equ      $01;                 ; I/O PORT B
PORTB7         equ      $7;                  ; PORTB BIT 7 I/O value
PORTB6         equ      $6;                  ; PORTB BIT 6 I/O value
PORTB5         equ      $5;                  ; PORTB BIT 5 I/O value
PORTB4         equ      $4;                  ; PORTB BIT 4 I/O value
PORTB3         equ      $3;                  ; PORTB BIT 3 I/O value
PORTB2         equ      $2;                  ; PORTB BIT 2 I/O value
PORTB1         equ      $1;                  ; PORTB BIT 1 I/O value
PORTB0         equ      $0;                  ; PORTB BIT 0 I/O value
;
PORTB7.        equ      $80;                 ; PORTB BIT 7 I/O value
PORTB6.        equ      $40;                 ; PORTB BIT 6 I/O value
PORTB5.        equ      $20;                 ; PORTB BIT 5 I/O value
PORTB4.        equ      $10;                 ; PORTB BIT 4 I/O value
PORTB3.        equ      $8;                  ; PORTB BIT 3 I/O value
PORTB2.        equ      $4;                  ; PORTB BIT 2 I/O value
PORTB1.        equ      $2;                  ; PORTB BIT 1 I/O value
PORTB0.        equ      $1;                  ; PORTB BIT 0 I/O value
;
PORTC          equ      $02;                 ; I/O PORT C
PORTC6         equ      $6;                  ; PORTC BIT 6 Input value
PORTC5         equ      $5;                  ; PORTC BIT 5 Input value
```

```
PORTC4        equ      $4;                  ; PORTC BIT 4 Input value
PORTC3        equ      $3;                  ; PORTC BIT 3 Input value
PORTC2        equ      $2;                  ; PORTC BIT 2 Input value
PORTC1        equ      $1;                  ; PORTC BIT 1 Input value
PORTC0        equ      $0;                  ; PORTC BIT 0 Input value
;
PORTC6.       equ      $40;                 ; PORTC BIT 6 Input value
PORTC5.       equ      $20;                 ; PORTC BIT 5 Input value
PORTC4.       equ      $10;                 ; PORTC BIT 4 Input value
PORTC3.       equ      $8;                  ; PORTC BIT 3 Input value
PORTC2.       equ      $4;                  ; PORTC BIT 2 Input value
PORTC1.       equ      $2;                  ; PORTC BIT 1 Input value
PORTC0.       equ      $1;                  ; PORTC BIT 0 Input value
;
PORTD         equ      $03;                 ; I/O PORT D
PORTD6        equ      $6;                  ; PORTD BIT 6 I/O value
PORTD5        equ      $5;                  ; PORTD BIT 5 I/O value
PORTD4        equ      $4;                  ; PORTD BIT 4 I/O value
PORTD3        equ      $3;                  ; PORTD BIT 3 I/O value
PORTD2        equ      $2;                  ; PORTD BIT 2 I/O value
PORTD1        equ      $1;                  ; PORTD BIT 1 I/O value
PORTD0        equ      $0;                  ; PORTD BIT 0 I/O value
;
PORTD6.       equ      $40;                 ; PORTD BIT 6 I/O value
PORTD5.       equ      $20;                 ; PORTD BIT 5 I/O value
PORTD4.       equ      $10;                 ; PORTD BIT 4 I/O value
PORTD3.       equ      $8;                  ; PORTD BIT 3 I/O value
PORTD2.       equ      $4;                  ; PORTD BIT 2 I/O value
PORTD1.       equ      $2;                  ; PORTD BIT 1 I/O value
PORTD0.       equ      $1;                  ; PORTD BIT 0 I/O value
;
PORTE         equ      $08;                 ; I/O PORT E
PORTE7        equ      $7;                  ; PORTE BIT 7 I/O value
PORTE6        equ      $6;                  ; PORTE BIT 6 I/O value
PORTE5        equ      $5;                  ; PORTE BIT 5 I/O value
PORTE4        equ      $4;                  ; PORTE BIT 4 I/O value
PORTE3        equ      $3;                  ; PORTE BIT 3 I/O value
PORTE2        equ      $2;                  ; PORTE BIT 2 I/O value
PORTE1        equ      $1;                  ; PORTE BIT 1 I/O value
PORTE0        equ      $0;                  ; PORTE BIT 0 I/O value
;
PORTE7.       equ      $80;                 ; PORTE BIT 7 I/O value
PORTE6.       equ      $40;                 ; PORTE BIT 6 I/O value
PORTE5.       equ      $20;                 ; PORTE BIT 5 I/O value
PORTE4.       equ      $10;                 ; PORTE BIT 4 I/O value
PORTE3.       equ      $8;                  ; PORTE BIT 3 I/O value
PORTE2.       equ      $4;                  ; PORTE BIT 2 I/O value
PORTE1.       equ      $2;                  ; PORTE BIT 1 I/O value
PORTE0.       equ      $1;                  ; PORTE BIT 0 I/O value
;
PORTF         equ      $09;                 ; I/O PORT F
PORTF5        equ      $5;                  ; PORTF BIT 5 I/O value
PORTF4        equ      $4;                  ; PORTF BIT 4 I/O value
PORTF3        equ      $3;                  ; PORTF BIT 3 I/O value
PORTF2        equ      $2;                  ; PORTF BIT 2 I/O value
PORTF1        equ      $1;                  ; PORTF BIT 1 I/O value
```

```
PORTF0         equ      $0;                  ; PORTF BIT 0 I/O value
;
PORTE5.        equ      $20;                 ; PORTF BIT 5 I/O value
PORTE4.        equ      $10;                 ; PORTF BIT 4 I/O value
PORTE3.        equ      $8;                  ; PORTF BIT 3 I/O value
PORTE2.        equ      $4;                  ; PORTF BIT 2 I/O value
PORTE1.        equ      $2;                  ; PORTF BIT 1 I/O value
PORTE0.        equ      $1;                  ; PORTF BIT 0 I/O value
;
DDRA           equ      $04;                 ; PORTA DATA DIRECTION
DDRA7          equ      7;                   ; PORTA BIT 7 Data Direction
DDRA6          equ      6;                   ; PORTA BIT 6 Data Direction
DDRA5          equ      5;                   ; PORTA BIT 5 Data Direction
DDRA4          equ      4;                   ; PORTA BIT 4 Data Direction
DDRA3          equ      3;                   ; PORTA BIT 3 Data Direction
DDRA2          equ      2;                   ; PORTA BIT 2 Data Direction
DDRA1          equ      1;                   ; PORTA BIT 1 Data Direction
DDRA0          equ      0;                   ; PORTA BIT 0 Data Direction
;
DDRA7.         equ      $80;                 ; PORTA BIT 7 Data Direction
DDRA6.         equ      $40;                 ; PORTA BIT 6 Data Direction
DDRA5.         equ      $20;                 ; PORTA BIT 5 Data Direction
DDRA4.         equ      $10;                 ; PORTA BIT 4 Data Direction
DDRA3.         equ      $8;                  ; PORTA BIT 3 Data Direction
DDRA2.         equ      $4;                  ; PORTA BIT 2 Data Direction
DDRA1.         equ      $2;                  ; PORTA BIT 1 Data Direction
DDRA0.         equ      $1;                  ; PORTA BIT 0 Data Direction
;
DDRB           equ      $05;                 ; PORTB DATA DIRECTION
DDRB7          equ      7;                   ; PORTB BIT 7 Data Direction
DDRB6          equ      6;                   ; PORTB BIT 6 Data Direction
DDRB5          equ      5;                   ; PORTB BIT 5 Data Direction
DDRB4          equ      4;                   ; PORTB BIT 4 Data Direction
DDRB3          equ      3;                   ; PORTB BIT 3 Data Direction
DDRB2          equ      2;                   ; PORTB BIT 2 Data Direction
DDRB1          equ      1;                   ; PORTB BIT 1 Data Direction
DDRB0          equ      0;                   ; PORTB BIT 0 Data Direction
;
DDRB7.         equ      $80;                 ; PORTB BIT 7 Data Direction
DDRB6.         equ      $40;                 ; PORTB BIT 6 Data Direction
DDRB5.         equ      $20;                 ; PORTB BIT 5 Data Direction
DDRB4.         equ      $10;                 ; PORTB BIT 4 Data Direction
DDRB3.         equ      $8;                  ; PORTB BIT 3 Data Direction
DDRB2.         equ      $4;                  ; PORTB BIT 2 Data Direction
DDRB1.         equ      $2;                  ; PORTB BIT 1 Data Direction
DDRB0.         equ      $1;                  ; PORTB BIT 0 Data Direction
;

DDRC           equ      $06;                 ; PORTC DATA DIRECTION
DDRC6          equ      6;                   ; PORTC BIT 6 Data Direction
DDRC5          equ      5;                   ; PORTC BIT 5 Data Direction
DDRC4          equ      4;                   ; PORTC BIT 4 Data Direction
DDRC3          equ      3;                   ; PORTC BIT 3 Data Direction
DDRC2          equ      2;                   ; PORTC BIT 2 Data Direction
DDRC1          equ      1;                   ; PORTC BIT 1 Data Direction
DDRC0          equ      0;                   ; PORTC BIT 0 Data Direction
```

```
;
DDRC6.        equ      $40;                  ; PORTC BIT 6 Data Direction
DDRC5.        equ      $20;                  ; PORTC BIT 5 Data Direction
DDRC4.        equ      $10;                  ; PORTC BIT 4 Data Direction
DDRC3.        equ      $8;                   ; PORTC BIT 3 Data Direction
DDRC2.        equ      $4;                   ; PORTC BIT 2 Data Direction
DDRC1.        equ      $2;                   ; PORTC BIT 1 Data Direction
DDRC0.        equ      $1;                   ; PORTC BIT 0 Data Direction
;

DDRE          equ      $0c;                  ; PORTE DATA DIRECTION
DDRE7         equ      7;                    ; PORTE BIT 7 Data Direction
DDRE6         equ      6;                    ; PORTE BIT 6 Data Direction
DDRE5         equ      5;                    ; PORTE BIT 5 Data Direction
DDRE4         equ      4;                    ; PORTE BIT 4 Data Direction
DDRE3         equ      3;                    ; PORTE BIT 3 Data Direction
DDRE2         equ      2;                    ; PORTE BIT 2 Data Direction
DDRE1         equ      1;                    ; PORTE BIT 1 Data Direction
DDRE0         equ      0;                    ; PORTE BIT 0 Data Direction
;
DDRA7.        equ      $80;                  ; PORTE BIT 7 Data Direction
DDRA6.        equ      $40;                  ; PORTE BIT 6 Data Direction
DDRA5.        equ      $20;                  ; PORTE BIT 5 Data Direction
DDRA4.        equ      $10;                  ; PORTE BIT 4 Data Direction
DDRA3.        equ      $8;                   ; PORTE BIT 3 Data Direction
DDRA2.        equ      $4;                   ; PORTE BIT 2 Data Direction
DDRA1.        equ      $2;                   ; PORTE BIT 1 Data Direction
DDRA0.        equ      $1;                   ; PORTE BIT 0 Data Direction
;
DDRF          equ      $0d;                  ; PORTF DATA DIRECTION
DDRF5         equ      5;                    ; PORTF BIT 5 Data Direction
DDRF4         equ      4;                    ; PORTF BIT 4 Data Direction
DDRF3         equ      3;                    ; PORTF BIT 3 Data Direction
DDRF2         equ      2;                    ; PORTF BIT 2 Data Direction
DDRF1         equ      1;                    ; PORTF BIT 1 Data Direction
DDRF0         equ      0;                    ; PORTF BIT 0 Data Direction
;
DDRF5.        equ      $20;                  ; PORTF BIT 5 Data Direction
DDRF4.        equ      $10;                  ; PORTF BIT 4 Data Direction
DDRF3.        equ      $8;                   ; PORTF BIT 3 Data Direction
DDRF2.        equ      $4;                   ; PORTF BIT 2 Data Direction
DDRF1.        equ      $2;                   ; PORTF BIT 1 Data Direction
DDRF0.        equ      $1;                   ; PORTF BIT 0 Data Direction
;
;      TIMER A
TASC          equ      $0e;                  ; TIMER A Status/Ctrl Register
TATOF         equ      7;                    ; TIMER A Overflow Flag Bit
TATOIE        equ      6;                    ; TIMER A Overflow Interrupt Enable Bit
TATSTOP       equ      5;                    ; TIMER A Stop Bit
TATRST        equ      4;                    ; TIMER A Reset Bit
TAPS2         equ      2;                    ; TIMER A Prescaler Select Bit 2
TAPS1         equ      1;                    ; TIMER A Prescaler Select Bit 1
TAPS0         equ      0;                    ; TIMER A Prescaler Select Bit 0
;
TATOF.        equ      $80;                  ; TIMER A Overflow Flag Bit
TATOIE.       equ      $40;                  ; TIMER A Overflow Interrupt Enable Bit
```

```
TATSTOP.      equ      $20;                ; TIMER A Stop Bit
TATRST.       equ      $10;                ; TIMER A Reset Bit
TAPS2.        equ      $4;                 ; TIMER A Prescaler Select Bit 2
TAPS1.        equ      $2;                 ; TIMER A Prescaler Select Bit 1
TAPS0.        equ      $1;                 ; TIMER A Prescaler Select Bit 0
;
TACNT         equ      $0f;                ; TIMER A Counter Register
TACNTH        equ      $0f;                ; TIMER A Counter High
TACNTL        equ      $10;                ; TIMER A Counter Low
TAMOD         equ      $11;                ; TIMER A Modulo Register
TAMODH        equ      $11;                ; TIMER A Modulo High
TAMODL        equ      $12;                ; TIMER A Modulo Low
TASC0         equ      $13;                ; TIMER A Channel 0 Status/Control
TASC0CH0F     equ      7;                  ; Channel 0 Flag Bit
TASCC0H0IE    equ      6;                  ; Channel 0 Interrupt Enable Bit
TASC0MS0B     equ      5;                  ; Channel 0 Mode Select Bit B
TASC0MS0A     equ      4;                  ; Channel 0 Mode Select Bit A
TASC0ELS0B    equ      3;                  ; Channel 0 Edge/Level Select Bit
TASC0ELS0A    equ      2;                  ; Channel 0 Edge/Level Select Bit
TASC0TOV0     equ      1;                  ; Channel 0 Toggle-On-Overflow Bit
TASC0CH0MAX   equ      0;                  ; Channel 0 Maximum Duty Cycle Bit
;
TASC0CH0F.    equ      $80;                ; Channel 0 Flag Bit
TASC0CH0IE.   equ      $40;                ; Channel 0 Interrupt Enable Bit
TASC0MS0B.    equ      $20;                ; Channel 0 Mode Select Bit B
TASC0MS0A.    equ      $10;                ; Channel 0 Mode Select Bit A
TASC0ELS0B.   equ      $8;                 ; Channel 0 Edge/Level Select Bit
TASC0ELS0A.   equ      $4;                 ; Channel 0 Edge/Level Select Bit
TASC0TOV0.    equ      $2;                 ; Channel 0 Toggle-On-Overflow Bit
TASC0CH0MAX.  equ      $1;                 ; Channel 0 Maximum Duty Cycle Bit
;
TACH0         equ      $14;                ; TIMER A Channel 0 Register
TACH0H        equ      $14;                ; TIMER A Channel 0 High
TACH0L        equ      $15;                ; TIMER A Channel 0 Low
TASC1         equ      $16;                ; TIMER A Channel 1 Status/Control
TASC1CH1F     equ      7;                  ; Channel 1 Flag Bit
TASC1CH1IE    equ      6;                  ; Channel 1 Interrupt Enable Bit
TASC1MS1A     equ      4;                  ; Channel 1 Mode Select Bit A
TASC1ELS1B    equ      3;                  ; Channel 1 Edge/Level Select Bit
TASC1ELS1A    equ      2;                  ; Channel 1 Edge/Level Select Bit
TASC1TOV1     equ      1;                  ; Channel 1 Toggle-On-Overflow Bit
TASC1CH0MAX   equ      0;                  ; Channel 1 Maximum Duty Cycle Bit
;
TASC1CH1F.    equ      $80;                ; Channel 1 Flag Bit
TASC1CH1IE.   equ      $40;                ; Channel 1 Interrupt Enable Bit
TASC1MS1A.    equ      $10;                ; Channel 1 Mode Select Bit A
TASC1ELS1B.   equ      $8;                 ; Channel 1 Edge/Level Select Bit
TASC1ELS1A.   equ      $4;                 ; Channel 1 Edge/Level Select Bit
TASC1TOV1.    equ      $2;                 ; Channel 1 Toggle-On-Overflow Bit
TASC1CH1MAX.  equ      $1;                 ; Channel 1 Maximum Duty Cycle Bit
;
TACH1         equ      $17;                ; TIMER A Channel 1 Register
TACH1H        equ      $17;                ; TIMER A Channel 1 High
TACH1L        equ      $18;                ; TIMER A Channel 1 Low
TASC2         equ      $19;                ; TIMER A Channel 2 Status/Control
TASC2CH2F     equ      7;                  ; Channel 2 Flag Bit
```

```
TASC2CH2IE    equ      6;                   ; Channel 2 Interrupt Enable Bit
TASC2MS2B     equ      5;                   ; Channel 2 Mode Select Bit B
TASC2MS2A     equ      4;                   ; Channel 2 Mode Select Bit A
TASC2ELS2B    equ      3;                   ; Channel 2 Edge/Level Select Bit
TASC21ELS2A   equ      2;                   ; Channel 2 Edge/Level Select Bit
TASC2TOV2     equ      1;                   ; Channel 2 Toggle-On-Overflow Bit
TASC21CH2MAX  equ      0;                   ; Channel 2 Maximum Duty Cycle Bit
;
TASC2CH2F.    equ      $80;                 ; Channel 2 Flag Bit
TASC2CH2IE.   equ      $40;                 ; Channel 2 Interrupt Enable Bit
TASC2MS2B.    equ      $20;                 ; Channel 2 Mode Select Bit B
TASC2MS2A.    equ      $10;                 ; Channel 2 Mode Select Bit A
TASC2ELS2B.   equ      $8;                  ; Channel 2 Edge/Level Select Bit
TASC2ELS2A.   equ      $4;                  ; Channel 2 Edge/Level Select Bit
TASC2TOV2.    equ      $2;                  ; Channel 2 Toggle-On-Overflow Bit
TASC2CH2MAX.  equ      $1;                  ; Channel 2 Maximum Duty Cycle Bit
;
TACH2         equ      $1a;                 ; TIMER A Channel 2 Register
TACH2H        equ      $1a;                 ; TIMER A Channel 2 High
TACH2L        equ      $1b;                 ; TIMER A Channel 2 Low
TASC3         equ      $1c;                 ; TIMER A Channel 3 Status/Control
TASC3CH3F     equ      7;                   ; Channel 3 Flag Bit
TASC3CH3IE    equ      6;                   ; Channel 3 Interrupt Enable Bit
TASC3MS3A     equ      4;                   ; Channel 3 Mode Select Bit A
TASC3ELS3B    equ      3;                   ; Channel 3 Edge/Level Select Bit
TASC31ELS3A   equ      2;                   ; Channel 3 Edge/Level Select Bit
TASC3TOV3     equ      1;                   ; Channel 3 Toggle-On-Overflow Bit
TASC31CH3MAX  equ      0;                   ; Channel 3 Maximum Duty Cycle Bit
;
TASC3CH3F.    equ      $80;                 ; Channel 3 Flag Bit
TASC3CH3IE.   equ      $40;                 ; Channel 3 Interrupt Enable Bit
TASC3MS3A.    equ      $10;                 ; Channel 3 Mode Select Bit A
TASC3ELS3B.   equ      $8;                  ; Channel 3 Edge/Level Select Bit
TASC3ELS3A.   equ      $4;                  ; Channel 3 Edge/Level Select Bit
TASC3TOV3.    equ      $2;                  ; Channel 3 Toggle-On-Overflow Bit
TASC3CH3MAX.  equ      $1;                  ; Channel 3 Maximum Duty Cycle Bit
;
TACH3         equ      $1d;                 ; TIMER A Channel 3 Register
TACH3H        equ      $1d;                 ; TIMER A Channel 3 High
TACH3L        equ      $1e;                 ; TIMER A Channel 3 Low
;      OPTION REGISTER
MOR           equ      $1f;                 ; Mask Option Write-Once Register
MOREDGE       equ      7;                   ; Mask Option Edge-Align Enable Bit
MORBOTNEG     equ      6;                   ; Mask Option Bottom PWM Polarity Bit
MORTOPNEG     equ      5;                   ; Mask Option Top PWM Polarity Bit
MORINDEP      equ      4;                   ; Mask Option Independent Mode En. Bit
MORLVIRST     equ      3;                   ; Mask Option LVI Reset Enable Bit
MORLVIPWR     equ      2;                   ; Mask Option LVI Power Enable Bit
MORSTOPE      equ      1;                   ; Mask Option STOP Instruction Enable Bit
MORCOPD       equ      0;                   ; Mask Option COP Disable Bit
;
MOREDGE.      equ      $80;                 ; Mask Option Edge-Align Enable Bit
MORBOTNEG.    equ      $40;                 ; Mask Option Bottom PWM Polarity Bit
MORTOPNEG.    equ      $20;                 ; Mask Option Top PWM Polarity Bit
MORINDEP.     equ      $10;                 ; Mask Option Independent Mode En. Bit
MORLVIRST.    equ      $8;                  ; Mask Option LVI Reset Enable Bit
```

```
MORLVIPWR.     equ      $4;                  ; Mask Option LVI Power Enable Bit
MORSTOPE.      equ      $2;                  ; Mask Option STOP Instruction Enable Bit
MORCOPD.       equ      $1;                  ; Mask Option COP Disable Bit
;
;      PWM
PCTL1          equ      $20;                 ; PWM control register 1
PWMDISX        equ      7;                   ; PWM Software Disable Bank X Bit
PMWDISY        equ      6;                   ; PWM Software Disable Bank Y Bit
PWMINT         equ      5;                   ; PWM Interrupt Enable Bit
PWMF           equ      4;                   ; PWM Reload Flag
PWMISENS1      equ      3;                   ; PWM Current Sense Correction Bit 1
PMWISENS0      equ      2;                   ; PWM Current Sense Correction Bit 0
PWMLDOK        equ      1;                   ; PWM Load OK Bit
PWMEN          equ      0;                   ; PWM Module Enable Bit
;
PWMDISX.       equ      $80;                 ; PWM Software Disable Bank X Bit
PMWDISY.       equ      $40;                 ; PWM Software Disable Bank Y Bit
PWMINT.        equ      $20;                 ; PWM Interrupt Enable Bit
PWMF.          equ      $10;                 ; PWM Reload Flag
PWMISENS1.     equ      $8;                  ; PWM Current Sense Correction Bit 1
PMWISENS0.     equ      $4;                  ; PWM Current Sense Correction Bit 0
PWMLDOK.       equ      $2;                  ; PWM Load OK Bit
PWMEN.         equ      $1;                  ; PWM Module Enable Bit
;
PCTL2          equ      $21;                 ; PWM Control register 2
PWMLDFQ1       equ      7;                   ; PWM Load Frequency Bit 1
PWMLDFQ0       equ      6;                   ; PWM Load Frequency Bit 0
PWMIPOL1       equ      4;                   ; PWM Top/Bot Correction For PWM Pair 1
PWMIPOL2       equ      3;                   ; PWM Top/Bot Correction For PWM Pair 2
PWMIPOL3       equ      2;                   ; PWM Top/Bot Correction For PWM Pair 3
PWMPRSC1       equ      1;                   ; PWM Prescaler Bit 1
PWMPRSC0       equ      0;                   ; PWM Prescaler Bit 1
;
PWMLDFQ1.      equ      $80;                 ; PWM Load Frequency Bit 1
PWMLDFQ0.      equ      $40;                 ; PWM Load Frequency Bit 0
PWMIPOL1.      equ      $10;                 ; PWM Top/Bot Correction For PWM Pair 1
PWMIPOL2.      equ      $8;                  ; PWM Top/Bot Correction For PWM Pair 2
PWMIPOL3.      equ      $4;                  ; PWM Top/Bot Correction For PWM Pair 3
PWMPRSC1.      equ      $2;                  ; PWM Prescaler Bit 1
PWMPRSC0.      equ      $1;                  ; PWM Prescaler Bit 1
;

FCR            equ      $22;                 ; Fault Control Register
FCRFINT4       equ      7;                   ; Fault Pin 4 Interrupt Enable Bit
FCRFMODE4      equ      6;                   ; Fault Mode selection for Fault Pin 4
FCRFINT3       equ      5;                   ; Fault Pin 3 Interrupt Enable Bit
FCRFMODE3      equ      4;                   ; Fault Mode selection for Fault Pin 3
FCRFINT2       equ      3;                   ; Fault Pin 2 Interrupt Enable Bit
FCRFMODE2      equ      2;                   ; Fault Mode selection for Fault Pin 2
FCRFINT1       equ      1;                   ; Fault Pin 1 Interrupt Enable Bit
FCRFMODE1      equ      0;                   ; Fault Mode selection for Fault Pin 1
;
FCRFINT4.      equ      $80;                 ; Fault Pin 4 Interrupt Enable Bit
FCRFMODE4.     equ      $40;                 ; Fault Mode selection for Fault Pin 4
FCRFINT3.      equ      $20;                 ; Fault Pin 3 Interrupt Enable Bit
FCRFMODE3.     equ      $10;                 ; Fault Mode selection for Fault Pin 3
```

```
FCRFINT2.      equ      $8;                  ; Fault Pin 2 Interrupt Enable Bit
FCRFMODE2.     equ      $4;                  ; Fault Mode selection for Fault Pin 2
FCRFINT1.      equ      $2;                  ; Fault Pin 1 Interrupt Enable Bit
FCRFMODE1.     equ      $1;                  ; Fault Mode selection for Fault Pin 1
;
FSR            equ      $23;                 ; Fault Status register
FSRFPIN4       equ      7;                   ; Fault Pin 4 State
FSRFFLAG4      equ      6;                   ; Fault Event Flag 4
FSRFPIN3       equ      5;                   ; Fault Pin 3 State
FPFFLAG3       equ      4;                   ; Fault Event Flag 3
FSRFPIN2       equ      3;                   ; Fault Pin 2 State
FSRFFLAG2      equ      2;                   ; Fault Event Flag 2
FSRFPIN1       equ      1;                   ; Fault Pin 1 State
FSRFFLAG1      equ      0;                   ; Fault Event Flag 1
;
FSRFPIN4.      equ      $80;                 ; Fault Pin 4 State
FSRFFLAG4.     equ      $40;                 ; Fault Event Flag 4
FSRFPIN3.      equ      $20;                 ; Fault Pin 3 State
FPFFLAG3.      equ      $10;                 ; Fault Event Flag 3
FSRFPIN2.      equ      $8;                  ; Fault Pin 2 State
FSRFFLAG2.     equ      $4;                  ; Fault Event Flag 2
FSRFPIN1.      equ      $2;                  ; Fault Pin 1 State
FSRFFLAG1.     equ      $1;                  ; Fault Event Flag 1
;
FTACK          equ      $24;                 ; Fault acknowledge register
FTACK4         equ      6;                   ; Fault Acknowledge 4 Bit
DT6            equ      5;                   ; Dead Time Bit 6
FTACK3         equ      4;                   ; Fault Acknowledge 3 Bit
DT5            equ      4;                   ; Dead Time Bit 5
DT4            equ      3;                   ; Dead Time Bit 4
FTACK2         equ      2;                   ; Fault Acknowledge 2 Bit
DT3            equ      2;                   ; Dead Time Bit 3
DT2            equ      1;                   ; Dead Time Bit 2
FTACK1         equ      0;                   ; Fault Acknowledge 1 Bit
DT1            equ      0;                   ; Dead Time Bit 1
;
FTACK4.        equ      $40;                 ; Fault Acknowledge 4 Bit
DT6.           equ      $20;                 ; Dead Time Bit 6
FTACK3.        equ      $10;                 ; Fault Acknowledge 3 Bit
DT5.           equ      $10;                 ; Dead Time Bit 5
DT4.           equ      $8;                  ; Dead Time Bit 4
FTACK2.        equ      $4;                  ; Fault Acknowledge 2 Bit
DT3.           equ      $4;                  ; Dead Time Bit 3
DT2.           equ      $2;                  ; Dead Time Bit 2
FTACK1.        equ      $1;                  ; Fault Acknowledge 1 Bit
DT1.           equ      $1;                  ; Dead Time Bit 1
;
PWMOUT         equ      $25;                 ; PWM Output Control Register
OUTCTL         equ      6;                   ; PWM Output Control Enable
OUT6           equ      5;                   ; PWM 6 Active
OUT5           equ      4;                   ; PWM 5 Active
OUT4           equ      3;                   ; PWM 4 Active
OUT3           equ      2;                   ; PWM 3 Active
OUT2           equ      1;                   ; PWM 2 Active
OUT1           equ      0;                   ; PWM 1 Active
;
```

```
OUTCTL.        equ      $40;                 ; PWM Output Control Enable
OUT6.          equ      $20;                 ; PWM 6 Active
OUT5.          equ      $10;                 ; PWM 5 Active
OUT4.          equ      $8;                  ; PWM 4 Active
OUT3.          equ      $4;                  ; PWM 3 Active
OUT2.          equ      $2;                  ; PWM 2 Active
OUT1.          equ      $1;                  ; PWM 1 Active
;
PCNT           equ      $26;                 ; PWM counter register
PCNTH          equ      $26;                 ; PWM counter register high
PCNTL          equ      $27;                 ; PWM counter register low
PMOD           equ      $28;                 ; PWM counter Modulo register
PMODH          equ      $28;                 ; PWM counter Modulo register high
PMODL          equ      $29;                 ; PWM counter Modulo register low
PVAL1          equ      $2a;                 ; PWM 1 value register
PVAL1H         equ      $2a;                 ; PWM 1 value register high
PVAL1L         equ      $2b;                 ; PWM 1 value register low
PVAL2          equ      $2c;                 ; PWM 2 value register
PVAL2H         equ      $2c;                 ; PWM 2 value register high
PVAL2L         equ      $2d;                 ; PWM 2 value register low
PVAL3          equ      $2e;                 ; PWM 3 value register
PVAL3H         equ      $2e;                 ; PWM 3 value register high
PVAL3L         equ      $2f;                 ; PWM 3 value register low
PVAL4          equ      $30;                 ; PWM 4 value register
PVAL4H         equ      $30;                 ; PWM 4 value register high
PVAL4L         equ      $31;                 ; PWM 4 value register low
PVAL5          equ      $32;                 ; PWM 5 value register
PVAL5H         equ      $32;                 ; PWM 5 value register high
PVAL5L         equ      $33;                 ; PWM 5 value register low
PVAL6          equ      $34;                 ; PWM 6 value register
PVAL6H         equ      $34;                 ; PWM 6 value register high
PVAL6L         equ      $35;                 ; PWM 6 value register low
;
DEADTM         equ      $36;                 ; Dead Time Write-once register
DISMAP         equ      $37;                 ; PWM Disable Mapping Write-once reg.
;
;       SCI section
SCC1           equ      $38;                 ; SCI control register 1
LOOPS          equ      7;                   ; Loop Mode Select Bit
ENSCI          equ      6;                   ; Enable SCI Bit
TXINV          equ      5;                   ; Transmit Inversion Bit
SCC1M          equ      4;                   ; Mode (Character Length) Bit
WAKE           equ      3;                   ; Wakeup Condition Bit
ILTY           equ      2;                   ; Idle Line Type Bit
PEN            equ      1;                   ; Parity Enable Bit
PTY            equ      0;                   ; Parity Bit
;
LOOPS.         equ      $80;                 ; Loop Mode Select Bit
ENSCI.         equ      $40;                 ; Enable SCI Bit
TXINV.         equ      $20;                 ; Transmit Inversion Bit
SCC1M.         equ      $10;                 ; Mode (Character Length) Bit
WAKE.          equ      $8;                  ; Wakeup Condition Bit
ILTY.          equ      $4;                  ; Idle Line Type Bit
PEN.           equ      $2;                  ; Parity Enable Bit
PTY.           equ      $1;                  ; Parity Bit
;
```

```
SCC2           equ      $39;                 ; SCI Control Register 2
SCTIE          equ      7;                   ; SCI Transmit Enable Bit
TCIE           equ      6;                   ; SCI Transmission Complete Int. Enable
SCRIE          equ      5;                   ; SCI Receive Complete Int. Enable
ILIE           equ      4;                   ; SCI Idle Line Interrupt Enable Bit
SCITE          equ      3;                   ; SCI Transmitter Enable Bit
SCIRE          equ      2;                   ; SCI Receiver Enable Bit
RWU            equ      1;                   ; SCI Receiver Wakeup Bit
SBK            equ      0;                   ; SCI Receiver Break Bit
;
SCTIE.         equ      $80;                 ; SCI Transmit Enable Bit
TCIE.          equ      $40;                 ; SCI Transmission Complete Int. Enable
SCRIE.         equ      $20;                 ; SCI Receive Complete Int. Enable
ILIE.          equ      $10;                 ; SCI Idle Line Interrupt Enable Bit
SCITE.         equ      $8;                  ; SCI Transmitter Enable Bit
SCIRE.         equ      $4;                  ; SCI Receiver Enable Bit
RWU.           equ      $2;                  ; SCI Receiver Wakeup Bit
SBK.           equ      $1;                  ; SCI Receiver Break Bit
;
SCC3           equ      $3a;                 ; SCI Control Register 3
SCC3R8         equ      7;                   ; SCI Receive Bit 8
SCC3T8         equ      6;                   ; SCI Transmit Bit 8
ORIE           equ      3;                   ; SCI Receiver Overrun Interrupt En. Bit
NEIE           equ      2;                   ; SCI Receiver Noise Error Int. En. Bit
FEIE           equ      1;                   ; SCI Receiver Framing Error Int. En. Bit
PEIE           equ      0;                   ; SCI Receiver Parity Error Int. En. Bit
;
SCC3R8.        equ      $80;                 ; SCI Receive Bit 8
SCC3T8.        equ      $40;                 ; SCI Transmit Bit 8
ORIE.          equ      $8;                  ; SCI Receiver Overrun Interrupt En. Bit
NEIE.          equ      $4;                  ; SCI Receiver Noise Error Int. En. Bit
FEIE.          equ      $2;                  ; SCI Receiver Framing Error Int. En. Bit
PEIE.          equ      $1;                  ; SCI Receiver Parity Error Int. En. Bit
;
SCS1           equ      $3b;                 ; SCI Status Register 1
SCTE           equ      7;                   ; SCI Transmitter Empty Bit
SCS1TC         equ      6;                   ; SCI Transmission Complete Bit
SCRF           equ      5;                   ; SCI Receiver Full Bit
IDLE           equ      4;                   ; SCI Receiver Idle Bit
SCS1OR         equ      3;                   ; SCI Receiver Overrun Bit
SCSINF         equ      2;                   ; SCI Receiver Noise Flag Bit
SCS1FE         equ      1;                   ; SCI Receiver Framing Bit
SCS1PE         equ      0;                   ; SCI Receiver Parity Bit
;
SCTE.          equ      $80;                 ; SCI Transmitter Empty Bit
SCS1TC.        equ      $40;                 ; SCI Transmission Complete Bit
SCRF.          equ      $20;                 ; SCI Receiver Full Bit
IDLE.          equ      $10;                 ; SCI Receiver Idle Bit
SCS1OR.        equ      $8;                  ; SCI Receiver Overrun Bit
SCSINF.        equ      $4;                  ; SCI Receiver Noise Flag Bit
SCS1FE.        equ      $2;                  ; SCI Receiver Framing Bit
SCS1PE.        equ      $1;                  ; SCI Receiver Parity Bit
;
SCS2           equ      $3c;                 ; SCI Status Register 2
BKF            equ      1;                   ; SCI Break Flag Bit
RPF            equ      0;                   ; SCI Reception-in-Progress Bit
```

```
;
BKF.           equ      $2;                  ; SCI Break Flag Bit
RPF.           equ      $1;                  ; SCI Reception-in-Progress Bit
;
SCDR           equ      $3d;                 ; SCI Data Register
SCBR           equ      $3e;                 ; SCI baud rate
SCP1           equ      5;                   ; SCI Baud Rate Prescaler Bit 1
SCP0           equ      4;                   ; SCI Baud Rate Prescaler Bit 0
SCR2           equ      2;                   ; SCI Baud Rate Selection Bit 2
SCR1           equ      1;                   ; SCI Baud Rate Selection Bit 1
SCR0           equ      0;                   ; SCI Baud Rate Selection Bit 0
;
SCP1.          equ      $20;                 ; SCI Baud Rate Prescaler Bit 1
SCP0.          equ      $10;                 ; SCI Baud Rate Prescaler Bit 0
SCR2.          equ      $4;                  ; SCI Baud Rate Selection Bit 2
SCR1.          equ      $2;                  ; SCI Baud Rate Selection Bit 1
SCR0.          equ      $1;                  ; SCI Baud Rate Selection Bit 0
;
;      INTERRUPT
ISCR           equ      $3f;                 ; IRQ Control/Status Register
IRQ1F          equ      3;                   ; IRQ1 Flag
ACK1           equ      2;                   ; IRQ1 Interrupt Request Acknowledge Bit
IMASK1         equ      1;                   ; IRQ1 Interrupt Mask Bit
MODE1          equ      0;                   ; IRQ! Edge/Level Select Bit
;
IRQ1F.         equ      $8;                  ; IRQ1 Flag
ACK1.          equ      $4;                  ; IRQ1 Interrupt Request Acknowledge Bit
IMASK1.        equ      $2;                  ; IRQ1 Interrupt Mask Bit
MODE1.         equ      $1;                  ; IRQ! Edge/Level Select Bit
;
;      A/D

ADSCR          equ      $40;                 ; ADC Status and Control Register
COCO           equ      7;                   ; Conversions Complete Select Bit
AIEN           equ      6;                   ; ACD Interrupt Enable Bit
ADCO           equ      5;                   ; ADC Continuous Conversion Bit
ADCH4          equ      4;                   ; ADC Channel Select Bit 4
ADCH3          equ      3;                   ; ADC Channel Select Bit 3
ADCH2          equ      2;                   ; ADC Channel Select Bit 2
ADCH1          equ      2;                   ; ADC Channel Select Bit 1
ADCH0          equ      0;                   ; ADC Channel Select Bit 0
;
COCO.          equ      $80;                 ; Conversions Complete Select Bit
AIEN.          equ      $40;                 ; ACD Interrupt Enable Bit
ADCO.          equ      $20;                 ; ADC Continuous Conversion Bit
ADCH4.         equ      $10;                 ; ADC Channel Select Bit 4
ADCH3.         equ      $8;                  ; ADC Channel Select Bit 3
ADCH2.         equ      $4;                  ; ADC Channel Select Bit 2
ADCH1.         equ      $2;                  ; ADC Channel Select Bit 1
ADCH0.         equ      $1;                  ; ADC Channel Select Bit 0
;
ADR            equ      $41;                 ; ADC data register
ADRH           equ      $41;                 ; ADC data register HIGH
ADRL           equ      $42;                 ; ADC data register LOW
ADCLK          equ      $43;                 ; ADC clock register
ADIV2          equ      7;                   ; ADC Clock Prescaler Bit 2
```

```
ADIV1          equ     6;                    ; ADC Clock Prescaler Bit 1
ADIV0          equ     5;                    ; ADC Clock Prescaler Bit 0
ADICLK         equ     4;                    ; ADC Input Clock Select Bit
MODE1          equ     3;                    ; ADC Modes Of Result Justification Bit 1
MODE0          equ     2;                    ; ADC Modes Of Result Justification Bit 0
;
ADIV2.         equ     $80;                  ; ADC Clock Prescaler Bit 2
ADIV1.         equ     $40;                  ; ADC Clock Prescaler Bit 1
ADIV0.         equ     $20;                  ; ADC Clock Prescaler Bit 0
ADICLK.        equ     $10;                  ; ADC Input Clock Select Bit
MODE1.         equ     $8;                   ; ADC Modes Of Result Justification Bit 1
MODE0.         equ     $4;                   ; ADC Modes Of Result Justification Bit 0
;
;       SPI
SPCR           equ     $44;                  ; SPI Control Register
SPRIE          equ     7;                    ; SPI Receiver Interrupt Enable Bit
SPMSTR         equ     5;                    ; SPI Master Bit
CPOL           equ     4;                    ; SPI Clock Polarity Bit
CPHA           equ     3;                    ; SPI Clock Phase Bit
SPWOM          equ     2;                    ; SPI Wired-OR Mode Bit
SPE            equ     1;                    ; SPI Enable Bit
SPTIE          equ     0;                    ; SPI Transmit Interrupt Enable Bit
;
SPRIE.         equ     $80;                  ; SPI Receiver Interrupt Enable Bit
SPMSTR.        equ     $20;                  ; SPI Master Bit
CPOL.          equ     $10;                  ; SPI Clock Polarity Bit
CPHA.          equ     $8;                   ; SPI Clock Phase Bit
SPWOM.         equ     $4;                   ; SPI Wired-OR Mode Bit
SPE.           equ     $2;                   ; SPI Enable Bit
SPTIE.         equ     $1;                   ; SPI Transmit Interrupt Enable Bit
;
SPSR           equ     $45;                  ; SPI Control/Status Register
SPRF           equ     7;                    ; SPI Receiver full Bit
ERRIE          equ     6;                    ; SPI Error Interrupt Enable Bit
OVRF           equ     5;                    ; SPI Overflow Bit
MODF           equ     4;                    ; SPI Mode Fault Bit
SPTE           equ     3;                    ; SPI Transmitter Empty Bit
MODFEN         equ     2;                    ; SPI Mode Fault Enable Bit
SPR1           equ     1;                    ; SPI Baud Rate Select Bit 1
SPR0           equ     0;                    ; SPI Baud Rate Select Bit 0
;
SPRF.          equ     $80;                  ; SPI Receiver full Bit
ERRIE.         equ     $40;                  ; SPI Error Interrupt Enable Bit
OVRF.          equ     $20;                  ; SPI Overflow Bit
MODF.          equ     $10;                  ; SPI Mode Fault Bit
SPTE.          equ     $8;                   ; SPI Transmitter Empty Bit
MODFEN.        equ     $4;                   ; SPI Mode Fault Enable Bit
SPR1.          equ     $2;                   ; SPI Baud Rate Select Bit 1
SPR0.          equ     $1;                   ; SPI Baud Rate Select Bit 0
;
SPDR           equ     $46;                  ; SPI data register
;       TIMER B
TBSC           equ     $51;                  ; TIMER B Status/Ctrl Register
TBTOF          equ     7;                    ; TIMER B Overflow Flag Bit
TBTOIE         equ     6;                    ; TIMER B Overflow Interrupt Enable Bit
TBTSTOP        equ     5;                    ; TIMER B Stop Bit
```

```
TBTRST         equ      4;                   ; TIMER B Reset Bit
TBPS2          equ      2;                   ; TIMER B Prescaler Select Bit 2
TBPS1          equ      1;                   ; TIMER B Prescaler Select Bit 1
TBPS0          equ      0;                   ; TIMER B Prescaler Select Bit 0
;
TBTOF.         equ      $80;                 ; TIMER B Overflow Flag Bit
TBTOIE.        equ      $40;                 ; TIMER B Overflow Interrupt Enable Bit
TBTSTOP.       equ      $20;                 ; TIMER B Stop Bit
TBTRST.        equ      $10;                 ; TIMER B Reset Bit
TBPS2.         equ      $4;                  ; TIMER B Prescaler Select Bit 2
TBPS1.         equ      $2;                  ; TIMER B Prescaler Select Bit 1
TBPS0.         equ      $1;                  ; TIMER B Prescaler Select Bit 0
;
TBCNT          equ      $52;                 ; TIMER B Counter Register
TBCNTH         equ      $52;                 ; TIMER B Counter High
TBCNTL         equ      $53;                 ; TIMER B Counter Low
TBMOD          equ      $54;                 ; TIMER B Modulo Register
TBMODH         equ      $54;                 ; TIMER B Modulo High
TBMODL         equ      $55;                 ; TIMER B Modulo Low
TBSC0          equ      $56;                 ; TIMER B Channel 0 Status/Ctrl
TBSC0CH0F      equ      7;                   ; Channel 0 Flag Bit
TBSC0CH0IE     equ      6;                   ; Channel 0 Interrupt Enable Bit
TBSC0MS0B      equ      5;                   ; Channel 0 Mode Select Bit B
TBSC0MS0A      equ      4;                   ; Channel 0 Mode Select Bit A
TBSC0ELS0B     equ      3;                   ; Channel 0 Edge/Level Select Bit
TBSC0ELS0A     equ      2;                   ; Channel 0 Edge/Level Select Bit
TBSC0TOV0      equ      1;                   ; Channel 0 Toggle-On-Overflow Bit
TBSC0CH0MAX    equ      0;                   ; Channel 0 Maximum Duty Cycle Bit
;
TBSC0CH0F.     equ      $80;                 ; Channel 0 Flag Bit
TBSC0CH0IE.    equ      $40;                 ; Channel 0 Interrupt Enable Bit
TBSC0MS0B.     equ      $20;                 ; Channel 0 Mode Select Bit B
TBSC0MS0A.     equ      $10;                 ; Channel 0 Mode Select Bit A
TBSC0ELS0B.    equ      $8;                  ; Channel 0 Edge/Level Select Bit
TBSC0ELS0A.    equ      $4;                  ; Channel 0 Edge/Level Select Bit
TBSC0TOV0.     equ      $2;                  ; Channel 0 Toggle-On-Overflow Bit
TBSC0CH0MAX.   equ      $1;                  ; Channel 0 Maximum Duty Cycle Bit
;
TBCH0          equ      $57;                 ; TIMER B Channel 0 Register
TBCH0H         equ      $57;                 ; TIMER B Channel 0 High
TBCH0L         equ      $58;                 ; TIMER B Channel 0 Low
TBSC1          equ      $59;                 ; TIMER B Channel 1 Status/Ctrl
TBSC1CH1F      equ      7;                   ; Channel 1 Flag Bit
TBSC1CH1IE     equ      6;                   ; Channel 1 Interrupt Enable Bit
TBSC1MS1A      equ      4;                   ; Channel 1 Mode Select Bit A
TBSC1ELS1B     equ      3;                   ; Channel 1 Edge/Level Select Bit
TBSC1ELS1A     equ      2;                   ; Channel 1 Edge/Level Select Bit
TBSC1TOV1      equ      1;                   ; Channel 1 Toggle-On-Overflow Bit
TBSC1CH1MAX    equ      0;                   ; Channel 1 Maximum Duty Cycle Bit
;
TBSC1CH1F.     equ      $80;                 ; Channel 1 Flag Bit
TBSC1CH1IE.    equ      $40;                 ; Channel 1 Interrupt Enable Bit
TBSC1MS1A.     equ      $10;                 ; Channel 1 Mode Select Bit A
TBSC1ELS1B.    equ      $8;                  ; Channel 1 Edge/Level Select Bit
TBSC1ELS1A.    equ      $4;                  ; Channel 1 Edge/Level Select Bit
TBSC1TOV1.     equ      $2;                  ; Channel 1 Toggle-On-Overflow Bit
```

```
TBSC1CH1MAX.   equ      $1;                    ; Channel 1 Maximum Duty Cycle Bit
;
TBCH1          equ      $5a;                   ; TIMER B Channel 1 Register
TBCH1H         equ      $5a;                   ; TIMER B Channel 1 High
TBCH1L         equ      $5b;                   ; TIMER B Channel 1 Low

;      PLL
PCTL           equ      $5c;                   ; PLL Control Register
PLLIE          equ      7;                     ; PLL Interrupt Enable Bit
PLLF           equ      6;                     ; PLL Interrupt Flag
PLLON          equ      5;                     ; PLL On Bit
BCS            equ      4;                     ; PLL Base Clock Select Bit
;
PLLIE.         equ      $80;                   ; PLL Interrupt Enable Bit
PLLF.          equ      $40;                   ; PLL Interrupt Flag
PLLON.         equ      $20;                   ; PLL On Bit
BCS.           equ      $10;                   ; PLL Base Clock Select Bit
;
PBWC           equ      $5d;                   ; PLL Bandwidth Register
AUTO           equ      7;                     ; PLL Automatic Bandwidth Control Bit
LOCK           equ      6;                     ; PLL Lock Indicator Bit
ACQ            equ      5;                     ; PLL Acquisition Bit
XLD            equ      4;                     ; PLL Crystal Loss Detect Bit
;
AUTO.          equ      $80;                   ; PLL Automatic Bandwidth Control Bit
LOCK.          equ      $40;                   ; PLL Lock Indicator Bit
ACQ.           equ      $20;                   ; PLL Acquisition Bit
XLD.           equ      $10;                   ; PLL Crystal Loss Detect Bit
;
PPG            equ      $5e;                   ; PLL programming register
MUL7           equ      7;                     ; PLL Multiplier Select Bit 7
MUL6           equ      6;                     ; PLL Multiplier Select Bit 6
MUL5           equ      5;                     ; PLL Multiplier Select Bit 5
MUL4           equ      4;                     ; PLL Multiplier Select Bit 4
VRS7           equ      3;                     ; PLL VCO Range Bit 7
VRS6           equ      2;                     ; PLL VCO Range Bit 6
VRS5           equ      1;                     ; PLL VCO Range Bit 5
VRS4           equ      0;                     ; PLL VCO Range Bit 4
;
MUL7.          equ      $80;                   ; PLL Multiplier Select Bit 7
MUL6.          equ      $40;                   ; PLL Multiplier Select Bit 6
MUL5.          equ      $20;                   ; PLL Multiplier Select Bit 5
MUL4.          equ      $10;                   ; PLL Multiplier Select Bit 4
VRS7.          equ      $8;                    ; PLL VCO Range Bit 7
VRS6.          equ      $4;                    ; PLL VCO Range Bit 6
VRS5.          equ      $2;                    ; PLL VCO Range Bit 5
VRS4.          equ      $1;                    ; PLL VCO Range Bit 4
;
;      SIM/FLASH/LVI/COP

SBSR           equ      $fe00;                 ; SIM Break Status Register
SBSW           equ      1;                     ; SIM Break Stop/Wait Bit
;
SBSW.          equ      $2;                    ; SIM Break Stop/Wait Bit
;
SRSR           equ      $fe01;                 ; SIM Reset Status Register
```

```
POR            equ      7;                    ; SIM Power On Reset
PIN            equ      6;                    ; SIM External Reset Bit
COP            equ      5;                    ; SIM Computer Operating Prop. Reset Bit
ILOP           equ      4;                    ; SIM Illegal Opcode Reset Bit
ILAD           equ      3;                    ; SIM Illegal Address Reset Bit
LVI            equ      1;                    ; SIM Low-Voltage Inhibit Reset Bit
;
POR.           equ      $80;                  ; SIM Power On Reset
PIN.           equ      $40;                  ; SIM External Reset Bit
COP.           equ      $20;                  ; SIM Computer Operating Prop. Reset Bit
ILOP.          equ      $10;                  ; SIM Illegal Opcode Reset Bit
ILAD.          equ      $8;                   ; SIM Illegal Address Reset Bit
LVI.           equ      $2;                   ; SIM Low-Voltage Inhibit Reset Bit
;
SBFCR          equ      $fe03;                ; SIM Break Control Register
BCFE           equ      7;                    ; SIM Break Clear Flag Enable Bit
;
BCFE.          equ      $80;                  ; SIM Break Clear Flag Enable Bit
;
FLCR           equ      $fe08;                ; FLASH Control Register
HVEN           equ      3;                    ; FLASH High Voltage Enable Bit
MASS           equ      2;                    ; FLASH Mass Erase Control Bit
ERASE          equ      1;                    ; FLASH Erase Control Bit
PGM            equ      0;                    ; FLASH Program Control Bit
;
HVEN.          equ      $8;                   ; FLASH High Voltage Enable Bit
MASS.          equ      $4;                   ; FLASH Mass Erase Control Bit
ERASE.         equ      $2;                   ; FLASH Erase Control Bit
PGM.           equ      $1;                   ; FLASH Program Control Bit
;
BRKH           equ      $fe0c;                ; SIM Break Address Register High
BRKL           equ      $fe0d;                ; SIM Break Address Register Low
BRKSCR         equ      $fe0e;                ; SIM Break Status And Control Register
BRKE           equ      7;                    ; Break Enable Bit
BRKA           equ      6;                    ; Break Active Bit
;
BRKE.          equ      $80;                  ; Break Enable Bit
BRKA.          equ      $40;                  ; Break Active Bit
;
LVISCR         equ      $fe0f;                ; LVI Status Register And Control
LVIOUT         equ      7;                    ; LVI output Bit
TRPSEL         equ      5;                    ; LVI Trip Select Bit
;
LVIOUT.        equ      $80;                  ; LVI output Bit
TRPSEL.        equ      $20;                  ; LVI Trip Select Bit
;
FLBPR          equ      $ff7e;                ; FLASH Block Protect Register
COPCTL         equ      $ffff;                ; COP Control Register
```

Conclusion

The 68HC908MR32 is designed so that it can be used easily in place of the 68HC908MR24 with no hardware and minimal software changes. The block-protected FLASH memory facilitates easy in-circuit software upgrades.

How to Reach Us:

Home Page:
www.freescale.com

E-mail:
support@freescale.com

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
+1-800-521-6274 or +1-480-768-2130
support@freescale.com

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
support@freescale.com

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-441-2447 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.
Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters which may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals" must be validated for each customer application by customer's technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

freescale™
semiconductor