

## AN1792

## Using an MC68HC908MR24 in Place of an MC68HC708MP16

By Bill Lucas and Denise Younger  
Freescale Microprocessor Division  
Austin, Texas

## Introduction

---

This application note documents the differences between the MC68HC708MP16 (MP16) and the MC68HC908MR24 (MR24). The information here is intended to help users migrate from the MC68HC708MP16 to the MC68HC908MR24.

The new features of the MR24 and differences between the two are:

- MR24 has a 24-k FLASH memory, replacing MP16's 16-k EPROM.
- MR24's RAM size increased to 768 bytes, starting at \$60, from 512 bytes on the MP16.
- MR24's low-voltage inhibit (LVI) is selectable with 5 percent or 10 percent tolerance.
- MR24's 10-bit A/D (analog-to-digital) converter has an 8-bit truncation mode, allowing compatibility with the MP16's 8-bit A/D converter. Quantization error, however, is affected in the truncation mode. Refer to the *MC68HC908MR24 General Release Specification*, Freescale document order number HC908MR24GRS/D, for more information.

- MR24's 10-bit A/D converter has a finite current draw of 1.6 mA, whereas on the MP16,  $V_{DDAD}/V_{DDREF}$  has negligible current draw.
- MP16's 8-bit A/D converter is replaced with a 10-bit module, selectable in 8-bit and 10-bit modes.
- Load OK bit (LDOK) in PWM control register 1 is now read-write, and the manner in which it is used has changed.
- Page zero I/O (input/output) module register addresses have changed.
- An external  $V_{REFH}$  pin has been added to the A/D converter.

## Background

---

The MC68HC908MR24 is an improved version of the MC68HC708MP16 and was launched to create a new family of products for motor control applications.

The MR24 is designed to be a mechanical drop-in replacement for the MC68HC708MP16, with one exception: Redefinition of pin 12 on the MP16 from  $V_{ADCAP}$  to  $V_{REFH}$  on the MR24.

Also, the 2-channel timer A on the MP16 is now named timer B on the MR24. And, the 4-channel timer B on the MP16 is now named timer A on the MR24.

From a software point of view, there are a number of address changes. The address and timer differences between the MP16 and MR24 are summarized in [Table 1](#).

**Table 1. MP16 and MR24 Address and Timer Differences**

Function	MC68HC708MP16	MC68HC908MR24
Pin 12	$V_{ADCAP}$	$V_{REFH}$
Timers	Pins 32–39 Timer A	Pins 32–39 Becomes timer B
I/O	\$0000–\$004f	\$0000–\$005f
RAM	\$0050–\$024f 512 bytes	\$0060–\$035f 768 bytes
A/D	8-bit mode	8- or 10-bit modes
	CONFIG (configuration register)	MOR (mask option register)

## From a Hardware Point of View

---

The physical footprint of the MR24 is the same as the MP16 except for the renaming of timer A and timer B on the MR24 and  $V_{ADCAP}$  on the MP16 being changed to  $V_{REFH}$  on the MR24.

The  $V_{ADCAP}$  to  $V_{REFH}$  change (pin 12) requires a printed circuit board change to move from the MP16 to the MR24. **Figure 1** shows typical usage of  $V_{ADCAP}$  on the MP16. **Figure 2** shows typical usage of  $V_{REFH}$  on the MR24.

**Figure 3** is an example of circuitry with a jumper that will accommodate both the MP16 and MR24.

Application Note

Freescale Semiconductor, Inc.

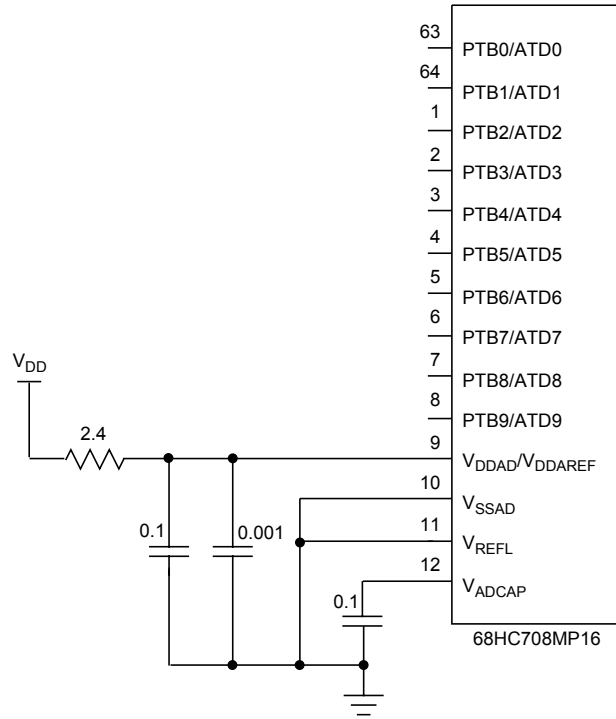


Figure 1. MC68HC708MP16 A/D Example

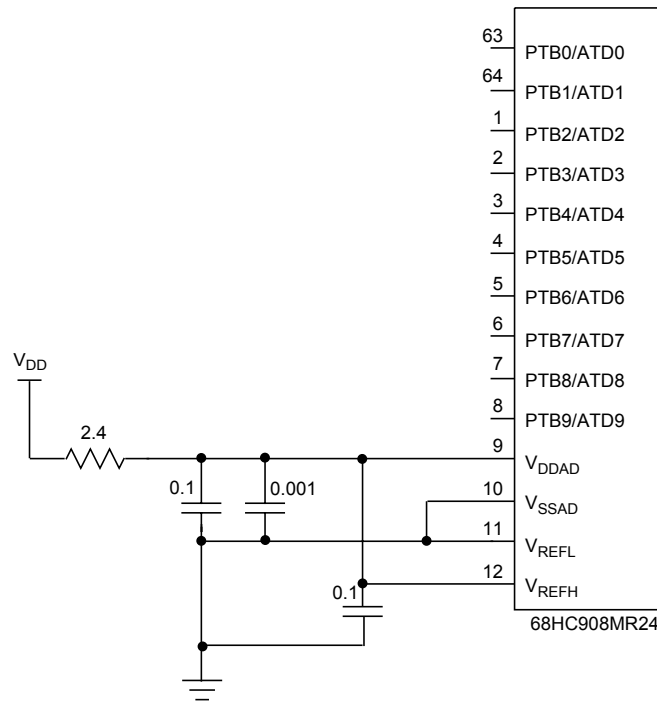
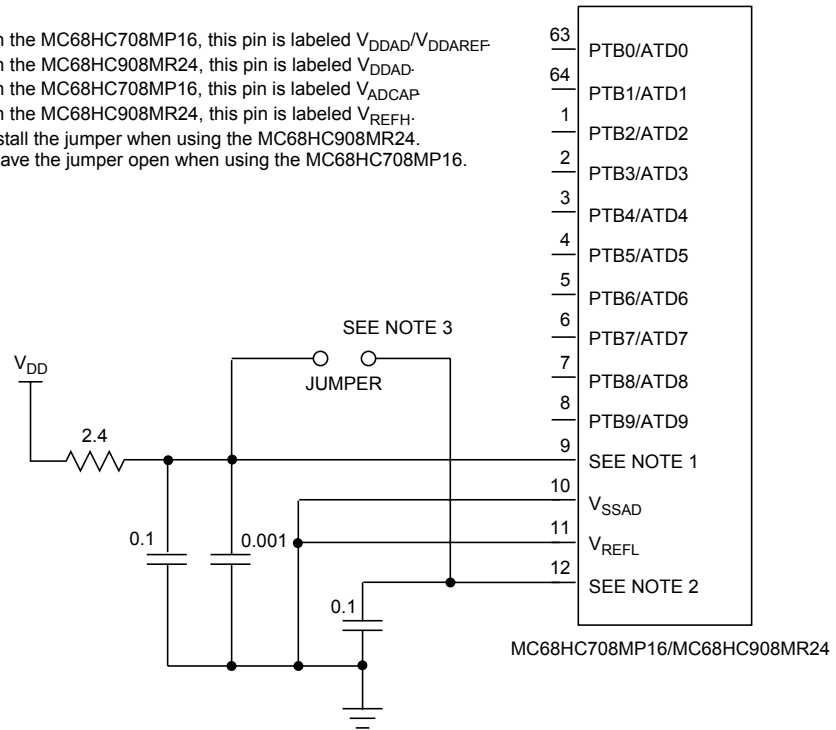


Figure 2. MC68HC908MR24 A/D Example

Notes:

1. On the MC68HC708MP16, this pin is labeled  $V_{DDAD}/V_{DDAREF}$   
On the MC68HC908MR24, this pin is labeled  $V_{DDAD}$ .
2. On the MC68HC708MP16, this pin is labeled  $V_{ADCAP}$   
On the MC68HC908MR24, this pin is labeled  $V_{REFH}$ .
3. Install the jumper when using the MC68HC908MR24.  
Leave the jumper open when using the MC68HC708MP16.



**Figure 3. MC68HC708MP16/MC68HC908MR24 A/D Example**

**From a Software Point of View**

A number of control, status, and data registers' addresses have changed from the MP16 to the MR24.

For instance, the I/O register map on the MP16 started at \$0000 and ended at \$004f. Now on the MR24, the I/O register map starts at \$0000 and ends at \$005f. Also, RAM on the MP16 started at \$0050 and ended at \$024f (512 bytes), whereas RAM on the MR24 starts at \$0060 and ends at \$035f (768 bytes).

The CONFIG (configuration register), located at address \$001f on the MP16, has been renamed to MOR (mask option register) on the MR24. However, its address is the same and the control bits within the register remain the same.

The LDOK bit, in PWM control register 1 (PCTL1) located at address \$0020, is used to enable values to be loaded into buffered PWM control and data registers. The intended use of LDOK is to prevent partial reloads of PWM parameters from occurring until all of the PWM parameters have been calculated, stored in the PWM registers, and it is now "OK" to use those values. The registers affected by the LDOK bit are the PWM modulus, PWM prescaler, and PWM value registers. When the LDOK bit is set, the calculated values are loaded into a second set of registers and passed to the PWM generator's registers at the beginning of the next PWM reload cycle. After the values have been loaded into the PWM generator, the LDOK bit is then set to a logic 0 by the PWM hardware.

On the MR24, the LDOK bit is now implemented as a read/write bit. Therefore, it requires an interlocking mechanism to ensure that inadvertent setting of LDOK does not occur when other bits in PCTL1 are written. On the MP16, LDOK was a write-only bit and is always read as a logic 0. To set the LDOK bit on the MR24, LDOK must first be read as a logic 0 before it can be set to a logic 1.

An example of an inadvertent setting of LDOK, without the interlock, would be if you set the LDOK bit in PCTL1 and before the load of the PWM generator occurs some other routine executes this instruction:

```
bset 5,PCTL1; or any other bit in PCTL1
```

The bit set (bset) instruction is a read/modify/write instruction and can be broken down into three steps:

1. The contents of PCTL1 are read.
2. Bit 5 (in this example) is set to a logic 1.
3. The original contents of PCTL1 (with bit 5 set to a 1) are written back to PCTL1.

When operation 1 is performed, LDOK was set to a logic 1. Some time after step 1 and before step 3, the PWM hardware performs a reload, thus clearing LDOK in PCTL1. Step 3 is then performed and a logic 1 is rewritten to LDOK, thus re-arming the PWM module for another reload when a reload was not requested. This was not an issue with the MP16, as LDOK is always read as a logic 0, and a write of 0 had no effect on

LDOK. The interlock on the MR24 solves the issue of inadvertent writes to LDOK, as it is required to read LDOK as a logic 0 before it may be set to a logic 1.

Because LDOK is a read/write bit on the MR24, a load operation can be aborted after the LDOK bit is set and before the actual PWM generator load occurs by clearing LDOK when it is set to a logic 1. This permits multiple, asynchronous software routines (for example, a Hall effect commutation ISR and a servo PWM update ISR) to interface properly with the PWM module without corrupting each other's data.

For convenience, I/O include files for the MR24 for both C language and assembler language are included in this application note.

## Address Changes

---

The addresses listed here have changed from the MP16 to the MR24, while the remainder of the I/O addresses stayed the same. The bits within these registers also remain the same.

The new locations for I/O addresses on the MR24 are listed here.

### I/O Ports

\$0004	Data Direction Register A (DDRA)
\$0005	Data Direction Register B (DDRB)
\$0006	Data Direction Register C (DDRC)
\$0007	Data Direction Register D (DDRD)
\$0008	Port E Data Register (PTE)
\$0009	Port F Data Register (PTF)
\$000C	Data Direction Register E (DDRE)
\$000D	Data Direction Register F (DDRF)

**Application Note**
**4-Channel Timer  
Interface Module**

**NOTE:** *The 4-channel timer on the MR24 is now named timer A.*

```

$000E Timer A Status and Control Register (TASC)
$000F Timer A Counter Register High (TACNTH)
$0010 Timer A Counter Register Low (TACNTL)
$0011 Timer A Modulo Register High (TAMODH)
$0012 Timer A Modulo Register Low (TAMODL)
$0013 Timer A Channel 0 Status and Control Register (TASCO)
$0014 Timer A Channel 0 Register High (TACH0H)
$0015 Timer A Channel 0 Register Low (TACH0L)
$0016 Timer A Channel 1 Status and Control Register (TASC1)
$0017 Timer A Channel 1 Register High (TACH1H)
$0018 Timer A Channel 1 Register Low (TACH1L)
$0019 Timer A Channel 2 Status and Control Register (TASC2)
$001A Timer A Channel 2 Register High (TACH2H)
$001B Timer A Channel 2 Register Low (TACH2L)
$001C Timer A Channel 3 Status and Control Register (TASC3)
$001D Timer A Channel 3 Register High (TACH3H)
$001E Timer A Channel 3 Register Low (TACH3L)

```

**2-Channel Timer  
Interface Module**

**NOTE:** *The 2-channel timer on the MR24 is now named timer B.*

```

$0051 Timer B Status and Control Register (TBSC)
$0052 Timer B Counter Register High (TBCNTH)
$0053 Timer B Counter Register Low (TBCNTL)
$0054 Timer B Modulo Register High (TBMODH)
$0055 Timer B Modulo Register Low (TBMODL)
$0056 Timer B Channel 0 Status and Control Register (TBSCO)
$0057 Timer B Channel 0 Register High (TBCH0H)
$0058 Timer B Channel 0 Register Low (TBCH0L)
$0059 Timer B Channel 1 Status and Control Register (TBSC1)
$005A Timer B Channel 1 Register High (TBCH1H)
$005B Timer B Channel 1 Register Low (TBCH1L)

```

**IRQ**

```
$003F IRQ Status and Control Register (ISCR)
```

**ADC**

```
$0040 ADC Status and control Register (ADSCR)
```

**NOTE:** *The analog-to-digital converter in the MR24 has been changed to a 10-bit module. Changes to the data register and the clock register are discussed later.*

**SPI**

\$0044	SPI Control Register (SPCR)
\$0045	SPI Status and Control Register (SPSCR)
\$0046	SPI Data Register (SPDR)

**CGM**

\$005C	PLL Control Register (PCTL)
\$005D	PLL Bandwidth Control Register (PBWC)
\$005E	PLL Programming Register (PPG)

**NOTE:** *The following registers on the MR24 have changed their addresses and/or had bits within the registers changed. New registers also have been added. Refer to the MR24's general release specification for more detailed information.*

**ADC**

The 8-bit ADR on the MP16 has been replaced with two registers, ADRH and ADRL, to support 8-bit and 10-bit conversions.

\$0041	ADC Data Register (ADRH)
\$0042	ADC Data Register (ADRL)
\$0043	ADC Clock Register (ADCLK)

Three bits have been added to the MR24 in bit positions 1–3. MODE1 (bit 3) and MODE0 (bit 2) allow the user to choose the conversion's data results justification of the analog-to-digital converter. Bit 1 will be used in future products.

**FLASH**

\$FE07	Reserved Test Register (FLTCR)
--------	--------------------------------

FLTCR is used only for production testing. Accessing this register can have unpredictable effects in normal MCU operation.

\$FE08	Flash Control Register (FLCR)
--------	-------------------------------

FLCR takes the place of the EPROM control register on the MP16.

\$FF80	Flash Block Protect Register (FLBPR)
--------	--------------------------------------

When the BPRX bits are set, a range of addresses is protected from being programmed or erased.

## Application Note

**LVI**                    \$FE0F   LVI Status and Control Register (LVISCR)

Bit 5 is now LVI trip selection (TPRSEL). The user can now select between 5 percent and 10 percent tolerance when monitoring the 5.0 V<sub>dc</sub>-V<sub>CC</sub> power supply. The default is set at 10 percent.

**Vector Addresses**    The vectors for timer A and timer B in the MP16 have been renamed in the MR24. The rest of the vectors remain the same between the two devices.

### C Include Files for the MC68HC908MR24

---

#### I/O Definitions for the MR24

These are for use with COSMIC Software, Inc.'s MC68HC08 compiler. The user should be aware that different compilers may require slight syntactical changes.

##### 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 volatile char 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 */
    
```

**Application Note**

```

@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

@near volatile   char    SBSR    @0xfe00; /* SIM break status register */
@near volatile   char    SRSR    @0xfe01; /* SIM reset status register */
@near            char    SBFCR   @0xfe03; /* SIM break control register */
@near            char    FLCR    @0xfe08; /* FLASH control register */
@near volatile   char    LVISCR  @0xfe0f; /* LVI status register and control */
@near            char    FLBPR   @0xff80; /* FLASH BLOCK PROTECT register */
@near volatile   char    COPCTL  @0xffff; /* COP control register */
    
```

## Assembler Include File for the MR24

### *I/O Definitions for MC68HC908MR24*

```

PORTA            equ    $00;           ; port A
PORTB            equ    $01;           ; port B
PORTC            equ    $02;           ; port C
PORTD            equ    $03;           ; port D
PORTE            equ    $08;           ; port E
PORTF            equ    $09;           ; port F
DDRA             equ    $04;           ; data direction port A
DDRB             equ    $05;           ; data direction port B
DDRC             equ    $06;           ; data direction port C
DDRD             equ    $07;           ; data direction port D
DDRF             equ    $0d;           ; data direction port F

;      TIMER A

TASC             equ    $0e;           ; timer A status/ctrl register
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/ctrl
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/ctrl
TACH1            equ    $17;           ; timer A channel 1 register
TACH1H           equ    $17;           ; timer A channel 1 high
    
```

**Application Note**

```

TACH1L      equ      $18;           ; timer A channel 1 low
TASC2       equ      $19;           ; timer A channel 2 status/ctrl
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/ctrl
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

;      PWM

PCTL1       equ      $20;           ; PWM control register 1
PCTL2       equ      $21;           ; PWM control register 2

FCR         equ      $22;           ; Fault control register
FSR         equ      $23;           ; Fault Status register
FTACK       equ      $24;           ; Fault acknowledge register

PWMOUT      equ      $25;           ; PWM output control register
PCNT        equ      $26;           ; PWM counter register
PCNTH       equ      $26;           ; PWM counter register high
PCNTL       equ      $27;           ; PWM counter register low
PMD         equ      $28;           ; PWM counter Modulo register
PMDH        equ      $28;           ; PWM counter Modulo register high
PMDL        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
            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
SCC2      equ      $39;          ; SCI control register 2
SCC3      equ      $3a;          ; SCI control register 3
SCS1      equ      $3b;          ; SCI status register 1
SCS2      equ      $3c;          ; SCI status register 2
SCDR      equ      $3d;          ; SCI data register
SCBR      equ      $3e;          ; SCI baud rate

;      INTERRUPT

ISCR      equ      $3f;          ; IRQ control/status register

;      A/D

ADSCR     equ      $40;          ; ADC status and control register
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

;      SPI

SPCR      equ      $44;          ; SPI control register
          equ      $45;          ; SPI control/status register
SPDR      equ      $46;          ; SPI data register

;      TIMER B

TBSC      equ      $51;          ; timer B status/ctrl register
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
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
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
PBWC      equ      $5d;          ; PLL bandwidth register
@ PPG     equ      $5e;          ; PLL programming register
    
```

## Application Note

```

;      SIM

SBSR      equ      $fe00;          ; SIM break status register
SRSR      equ      $fe01;          ; SIM reset status register
SBFCR     equ      $fe03;          ; SIM break control register
FLCR      equ      $fe08;          ; FLASH control register
LVISCR    equ      $fe0f;          ; LVI status register and control
FLBPR     equ      $ff80;          ; FLASH BLOCK PROTECT register
COPCTL    equ      $ffff;          ; COP control register

```

## Conclusion

---

The MC68HC908MR24 is designed to be used in place of the MC68HC708MP16 with minimal hardware and software changes. The inclusion of the 10-bit analog-to-digital converter and increased memory size over the MP16 make this processor an attractive alternative. The block-protected FLASH memory facilitates easy in-circuit software upgrades.

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.

