CONFIG Register Programming for EEPROM-based M68HC11 Microcontrollers

Introduction

To guarantee proper operation of EEPROM-based M68HC11 devices, the CONFIG register must be correctly programmed. A CONFIG register verification and reprogramming routine should be included at the beginning of critical M68HC11 programs.

Code Listing

The following example code shows how to verify and reprogram the EEPROM CONFIG register to ensure proper operation. The same results can be accomplished with less generic, user-specified code. **Table 1** shows M68HC11 devices with EEPROM-based CONFIG registers. Use **Table 1** when customizing the source code. Refer to the appropriate M68HC11 Technical Data Book or Technical Summary for CONFIG register control bit definitions

The code will execute properly in single-chip or expanded operating modes on all EEPROM-based M68HC11 microcontrollers except for devices in the A Series. The CONFIG register In A Series devices can only be programmed in special test or bootstrap operating modes. Users devices in the A Series may choose to provide hardware support for special test or bootstrap mode operation. The code can be used as written in these modes if a proper starting address is selected. See **SECTION 3 CONFIGURATION AND MODES OF OPERATION** of the *M68HC11 Reference Manual* (M68HC11RM/AD) for more information.

```
* FILENAME: config.asm
*
* DESCRIPTION: This code checks the CONFIG register on an EEPROM-based
* HC11 device and reprograms it with the proper value if necessary.
*
```

Refer to Table 1. Fill in the blank that follows with the register base address for the device being used.

```
REGBASE     equ     $____               ;beginning of HC11 registers
* Offsets from the beginning of the register block.
TCNT        equ     $0E
TOC4        equ     $1C
TFLG1       equ     $23
BPROT       equ     $35
OPTION      equ     $39
PPROG       equ     $3B
CONFIG      equ     $3F
CSCSTR      equ     $5A
* The following register bit constants are needed.
OC4F        equ     $10
PTCON       equ     $10
CME         equ     $08
BYTE        equ     $10
ERASE       equ     $04
EELAT       equ     $02
EEPGM       equ     $01
```

Fill in the blank that follows with the desired CONFIG register value.

```
* Other user constants should follow, including:
MY_CONFIG  equ     $__
```

Fill in the program starting address in the following blank.

```
START      org     $____                ;program starts here
```

The next line is only needed for derivatives in the K Series that are running in expanded mode.

```
           clr     CSCSTR               ;disable clock stretching on K-series
           lds     #$00FF               ;set a valid stack pointer
           ldx     #REGBASE             ;set beginning of register block
           ldaa    CONFIG,X             ;read CONFIG
           cmpa    #MY_CONFIG           ;check for valid CONFIG
           beq     NORMAL               ;if CONFIG is OK, go on as usual
```

At this point, 49 cycles remain for modifications to be made to the time protected registers on all HC11 devices except for devices in the K Series that are running in expanded mode. On these devices, 37 cycles remain because program chip-select clock stretching is enabled in expanded mode, effectively doubling the execution time of all instructions until stretching is disabled.

```
           bclr    BPROT,X,PTCON        ;clear CONFIG protect bit
* CONFIG erase sequence.
           ldaa    #{BYTE + ERASE + EELAT}
           staa    PPROG,X
```

The EEPROM erase sequence requires that some data be stored to the byte being erased. The actual data stored and instructions used are irrelevant; it is only necessary to complete a memory write cycle to the location in question.

```
           staa    CONFIG,X             ;store something to CONFIG
           ldaa    #{BYTE + ERASE + EELAT + EEPGM}
           staa    PPROG,X
           jsr     EEDELAY              ;wait 10 ms
           clr     PPROG,X              ;finish erase sequence
* CONFIG program sequence.
           ldaa    #EELAT
           staa    PPROG,X
           ldaa    #MY_CONFIG           ;desired CONFIG value
           staa    CONFIG,X
           ldaa    #{EELAT + EEPGM}
           staa    PPROG,X
           jsr     EEDELAY              ;wait 10 ms
           clr     PPROG,X              ;finish program sequence
* Now allow clock monitor to reset the HC11 and latch the new CONFIG register value.
           bset    OPTION,X,CME         ;enable clock monitor reset
           tpa                          ;get condition code register
           anda    #$7F                 ;enable STOP mode
           tap
           nop
           stop                         ;enter STOP mode and allow reset
* User program resumes here if CONFIG does not need to be reprogrammed.
NORMAL     etc.
* This delay subroutine may be used for any EEPROM programming/erase operation.
EEDELAY    ldd     TCNT,X               ;get current time
```

Fill in the following blank with the delay term used for program and erase operations. DELAY = ECLK/100, and typical values are 40000 at 4 MHz, 20000 at 2 MHz, and 10000 at 1 MHz.

```
           addd    #_____               ;add delay
           std     TOC4,X               ;allow match at end of delay
           ldaa    #OC4F                ;clear last output compare match
           staa    TFLG1,X
* Wait for OC4 match (end of 10 ms delay) to occur.
DELAYLOOP  brclr   TFLG1,X,OC4F,DELAYLOOP
           rts                          ;end of delay loop
```

Table 1 M68HC11 Devices with EEPROM-Based CONFIG Registers

Device	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Base
MC68HC11A0	—	—	—	—	NOSEC	NOCOP	ROMON	EEON	$1000
MC68HC11A1	—	—	—	—	NOSEC	NOCOP	ROMON	EEON	$1000
MC68HC11A7	—	—	—	—	NOSEC	NOCOP	ROMON	EEON	$1000
MC68HC11A8	—	—	—	—	NOSEC	NOCOP	ROMON	EEON	$1000
MC68L11A0	—	—	—	—	NOSEC	NOCOP	ROMON	EEON	$1000
MC68L11A1	—	—	—	—	NOSEC	NOCOP	ROMON	EEON	$1000
MC68L11A7	—	—	—	—	NOSEC	NOCOP	ROMON	EEON	$1000
MC68L11A8	—	—	—	—	NOSEC	NOCOP	ROMON	EEON	$1000
MC68HC11E0	—	—	—	—	NOSEC	NOCOP	ROMON	EEON	$1000
MC68HC11E1	—	—	—	—	NOSEC	NOCOP	ROMON	EEON	$1000
MC68HC11E8	—	—	—	—	NOSEC	NOCOP	ROMON	EEON	$1000
MC68HC11E9	—	—	—	—	NOSEC	NOCOP	ROMON	EEON	$1000
MC68L11E0	—	—	—	—	NOSEC	NOCOP	ROMON	EEON	$1000
MC68L11E1	—	—	—	—	NOSEC	NOCOP	ROMON	EEON	$1000
MC68L11E8	—	—	—	—	NOSEC	NOCOP	ROMON	EEON	$1000
MC68L11E9	—	—	—	—	NOSEC	NOCOP	ROMON	EEON	$1000
MC68HC711E9	—	—	—	—	NOSEC	NOCOP	ROMON	EEON	$1000
MC68S711E9	—	—	—	—	NOSEC	NOCOP	ROMON	EEON	$1000
MC68HC11E20	—	—	—	—	NOSEC	NOCOP	ROMON	EEON	$1000
MC68HC711E20	—	—	—	—	NOSEC	NOCOP	ROMON	EEON	$1000
MC68HC811E2	EE3	EE2	EE1	EE0	NOSEC	NOCOP	ROMON	EEON	$1000
MC68HC11EA9	—	—	—	—	NOSEC	NOCOP	ROMON	EEON	$1000
MC68HC711EA9	—	—	—	—	NOSEC	NOCOP	ROMON	EEON	$1000
MC68HC11F1	EE3	EE2	EE1	EE0	NOSEC	NOCOP	ROMON	EEON	$1000
MC68L11F1	EE3	EE2	EE1	EE0	NOSEC	NOCOP	ROMON	EEON	$1000
MC68HC11K0	ROMAD	—	CLK4X	PAREN	NOSEC	NOCOP	ROMON	EEON	$0000
MC68HC11K1	ROMAD	—	CLK4X	PAREN	NOSEC	NOCOP	ROMON	EEON	$0000
MC68HC11K3	ROMAD	—	CLK4X	PAREN	NOSEC	NOCOP	ROMON	EEON	$0000
MC68HC11K4	ROMAD	—	CLK4X	PAREN	NOSEC	NOCOP	ROMON	EEON	$0000
MC68L11K0	ROMAD	—	CLK4X	PAREN	NOSEC	NOCOP	ROMON	EEON	$0000
MC68L11K1	ROMAD	—	CLK4X	PAREN	NOSEC	NOCOP	ROMON	EEON	$0000
MC68L11K3	ROMAD	—	CLK4X	PAREN	NOSEC	NOCOP	ROMON	EEON	$0000
MC68L11K4	ROMAD	—	CLK4X	PAREN	NOSEC	NOCOP	ROMON	EEON	$0000
MC68HC711K4	ROMAD	—	CLK4X	PAREN	NOSEC	NOCOP	ROMON	EEON	$0000
MC68HC11KA0	ROMAD	—	CLKX	PAREN	NOSEC	NOCOP	ROMON	EEON	$0000
MC68HC11KA1	ROMAD	—	CLKX	PAREN	NOSEC	NOCOP	ROMON	EEON	$0000
MC68HC11KA3	ROMAD	—	CLKX	PAREN	NOSEC	NOCOP	ROMON	EEON	$0000
MC68HC11KA4	ROMAD	—	CLKX	PAREN	NOSEC	NOCOP	ROMON	EEON	$0000
MC68HC711KA4	ROMAD	—	CLKX	PAREN	NOSEC	NOCOP	ROMON	EEON	$0000
MC68HC11KA2	ROMAD	—	CLKX	PAREN	NOSEC	NOCOP	ROMON	EEON	$0000
MC68HC711KA2	ROMAD	—	CLKX	PAREN	NOSEC	NOCOP	ROMON	EEON	$0000
MC68HC11L0	—	—	—	—	NOSEC	NOCOP	ROMON	EEON	$1000
MC68HC11L1	—	—	—	—	NOSEC	NOCOP	ROMON	EEON	$1000
MC68HC11L5	—	—	—	—	NOSEC	NOCOP	ROMON	EEON	$1000

Table 1 M68HC11 Devices with EEPROM-Based CONFIG Registers (Continued)

Device	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Base
MC68HC11L6	—	—	—	—	NOSEC	NOCOP	ROMON	EEON	$1000
MC68L11L0	—	—	—	—	NOSEC	NOCOP	ROMON	EEON	$1000
MC68L11L1	—	—	—	—	NOSEC	NOCOP	ROMON	EEON	$1000
MC68L11L5	—	—	—	—	NOSEC	NOCOP	ROMON	EEON	$1000
MC68L11L6	—	—	—	—	NOSEC	NOCOP	ROMON	EEON	$1000
MC68HC711L6	—	—	—	—	NOSEC	NOCOP	ROMON	EEON	$1000
MC68HC11P2	ROMAD	—	—	PAREN	NOSEC	NOCOP	ROMON	EEON	$0000
MC68HC711P2	ROMAD	—	—	PAREN	NOSEC	NOCOP	ROMON	EEON	$0000

Home Page:
www.freescale.com
email:
support@freescale.com
USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
(800) 521-6274
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
+81 2666 8080
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
(800) 441-2447
303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor
@hibbertgroup.com