HC11

MC68HC11KG4

TECHNICAL
DATA

MOTOROLA

CUSTOMER FEEDBACK QUESTIONNAIRE (MC68HC11KG4/D)

Motorola wishes to continue to improve the quality of its documentation. We would welcome your feedback on the publication you have just received. Having used the document, please complete this card (or a photocopy of it, if you prefer).

1. How would you rate the quality of the document? Check one box in each category.

	Excellent			Poor		Excellent			Poor
Organization	☐	☐	☐	☐	Tables	☐	☐	☐	☐
Readability	☐	☐	☐	☐	Table of contents	☐	☐	☐	☐
Understandability	☐	☐	☐	☐	Index	☐	☐	☐	☐
Accuracy	☐	☐	☐	☐	Page size/binding	☐	☐	☐	☐
Illustrations	☐	☐	☐	☐	Overall impression	☐	☐	☐	☐

Comments: _____

2. What is your intended use for this document? If more than one option applies, please rank them (1, 2, 3).

Selection of device for new application ☐
System design ☐
Training purposes ☐
Other ☐ Please specify: _____

3. How well does this manual enable you to perform the task(s) outlined in question 2?

Completely ☐ ☐ ☐ ☐ Not at all

Comments: _____

4. How easy is it to find the information you are looking for?

Easy ☐ ☐ ☐ ☐ Difficult

Comments: _____

5. Is the level of technical detail in the following sections sufficient to allow you to understand how the device functions?

		Too little detail				Too much detail
SECTION 1	INTRODUCTION	☐	☐	☐	☐	☐
SECTION 2	PIN DESCRIPTIONS	☐	☐	☐	☐	☐
SECTION 3	CPU CORE AND INSTRUCTION SET	☐	☐	☐	☐	☐
SECTION 4	OPERATING MODES AND ON-CHIP MEMORY	☐	☐	☐	☐	☐
SECTION 5	RESETS AND INTERRUPTS	☐	☐	☐	☐	☐
SECTION 6	PARALLEL INPUT/OUTPUT	☐	☐	☐	☐	☐
SECTION 7	SERIAL COMMUNICATIONS INTERFACE	☐	☐	☐	☐	☐
SECTION 8	SERIAL PERIPHERAL INTERFACE	☐	☐	☐	☐	☐
SECTION 9	PULSE WIDTH MODULATION TIMER	☐	☐	☐	☐	☐
SECTION 10	TIMING SYSTEM	☐	☐	☐	☐	☐
SECTION 11	ANALOG-TO-DIGITAL CONVERTER	☐	☐	☐	☐	☐
SECTION 12	DIGITAL-TO-ANALOG CONVERTER	☐	☐	☐	☐	☐
SECTION A	ELECTRICAL SPECIFICATIONS	☐	☐	☐	☐	☐
SECTION B	MECHANICAL DATA AND ORDERING INFORMATION	☐	☐	☐	☐	☐
SECTION C	DEVELOPMENT SUPPORT	☐	☐	☐	☐	☐

Comments: _____

6. Have you found any errors? If so, please comment: _____

7. From your point of view, is anything missing from the document? If so, please say what: _____

– Cut along this line to remove –

8. How could we improve this document?

9. How would you rate Motorola's documentation?

	Excellent			Poor
– In general	☐	☐	☐	☐
– Against other semiconductor suppliers	☐	☐	☐	☐

10. Which semiconductor manufacturer provides the best technical documentation?
11. Which company (in any field) provides the best technical documentation?
12. How many years have you worked with microprocessors?

Less than 1 year ☐ 1–3 years ☐ 3–5 years ☐ More than 5 years ☐

– Second fold back along this line –

By air mail
Par avion

IBRS NUMBER PHQ-B/207/G
CCRI NUMERO PHQ-B/207/G

NE PAS AFFRANCHIR
NO STAMP REQUIRED

REPONSE PAYEE
GRANDE-BRETAGNE

Motorola Ltd.,
Colvilles Road,
Kelvin Industrial Estate,
EAST KILBRIDE,
G75 8BR.
GREAT BRITAIN.

F.A.O. Technical Publications Manager
(re: MC68HC11KG4/D)

MOTOROLA
Semiconductor Products Sector

– First fold back along this line –

– Cut along this line to remove –

– Third fold back along this line –

13. Currently there is some discussion in the semiconductor industry regarding a move towards providing data sheets in electronic form. If you have any opinion on this subject, please comment.

14. We would be grateful if you would supply the following information (at your discretion), or attach your card.

Name: Phone No:
Position: FAX No:
Department:
Company:
Address:

Thank you for helping us improve our documentation,
Technical Publications Manager, Motorola Ltd., Scotland.

– Finally, tuck this edge into opposite flap –

MC68HC11KG4

High-density complementary metal oxide semiconductor (HCMOS) microcontroller unit

Conventions

Where abbreviations are used in the text, an explanation can be found in the glossary, at the back of this manual. Register and bit mnemonics are defined in the paragraphs describing them.

An overbar is used to designate an active-low signal, eg: $\overline{\text{RESET}}$.

Unless otherwise stated, shaded cells in a register diagram indicate that the bit is either unused or reserved; 'u' is used to indicate an undefined state (on reset).

TABLE OF CONTENTS

3
CPU CORE AND INSTRUCTION SET

4
OPERATING MODES AND ON-CHIP MEMORY

5
RESETS AND INTERRUPTS

6
PARALLEL INPUT/OUTPUT

7
SERIAL COMMUNICATIONS INTERFACE

8
SERIAL PERIPHERAL INTERFACE

9
PULSE WIDTH MODULATION TIMER

10
TIMING SYSTEM

11
ANALOG-TO-DIGITAL CONVERTER

12
DIGITAL-TO-ANALOG CONVERTER

A
ELECTRICAL SPECIFICATIONS

B
MECHANICAL DATA AND ORDERING INFORMATION

C
DEVELOPMENT SUPPORT

LIST OF FIGURES

LIST OF TABLES

THIS PAGE LEFT BLANK INTENTIONALLY

1
INTRODUCTION

The MC68HC11KG4 is a member of the M68HC11 family of HCMOS 8-bit microcontrollers. In addition to the 24K bytes of user ROM, the MC68HC11KG4 contains 768 bytes of RAM and 640 bytes of EEPROM. This high-speed microcontroller makes use of a non-multiplexed bus and operates at 5V ±10% at a nominal bus speed of 4MHz; the fully static design allows operation at frequencies down to dc.

The MC68HC11KG4 is ideally suited to automotive and industrial applications, as three of the on-chip ports have the capability to sink and source large currents, which is ideal for driving LEDs. In addition, two of the nine I/O ports use 8.5V supply voltages. This enables them to drive high power peripherals, such as air core instruments of the type found in car dashboards.

1.1 Features

- Low power, high performance M68HC11 CPU core
- 24K bytes of user ROM (MC68HC11KG4)
- 768 bytes of RAM
- 640 bytes of byte-erasable EEPROM, with on-chip charge pump
- Non-multiplexed address and data buses, permitting direct access to the full 64K address map
- 16-bit timer with 3/4 input captures and 4/5 output compares; pulse accumulator, periodic interrupt and COP watchdog timer
- Enhanced COP (WCOP) watchdog timer, allowing refresh operation only within a time window
- Power saving PLL circuit
- Two NRZ type SCI subsystems; both enhanced to include a modulus prescaler
- SPI subsystem, with software selectable LSB/MSB first option
- 7-channel, 10-bit analog-to-digital converter
- 8-bit digital-to-analog converter

- Eight 9-bit PWM timer channels
- Available in the low profile 100-pin TQFP package
- RTI/SRTI
- Keyboard interrupt port (4 pins)
- Power driver short circuit detection

1.2 Mask options

- Security option (available/unavailable); see Section 4.4.3
- Window COP watchdog (enable/disable); see Section 5.1.3
- Synthesizer program register (enable/disable); see Section 2.5.4.2
- Low voltage reset (LVR) function (enable/disable); see Section 5.1.1 and Section A.5.8
- Ports H and K driven low in WAIT mode (enable/disable).

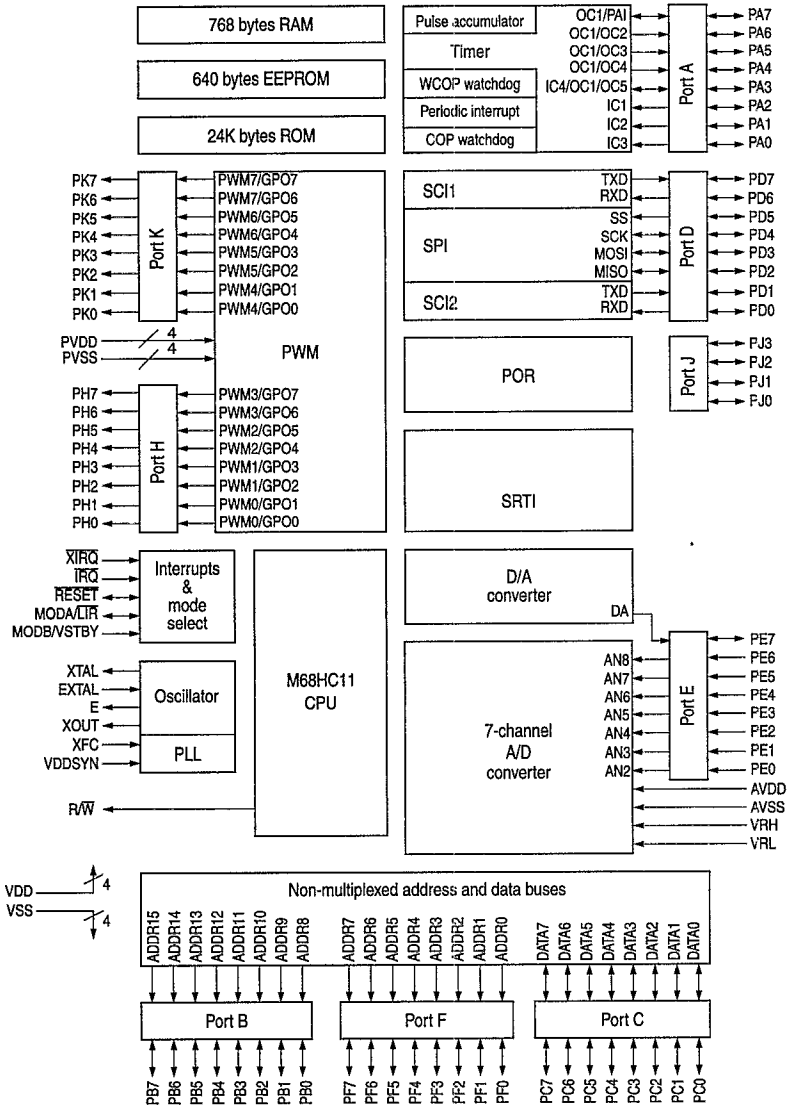


Figure 1-1 MC68HC11KG4 block diagram

THIS PAGE LEFT BLANK INTENTIONALLY

2
PIN DESCRIPTIONS

The MC68HC11KG4 is available in a 100-pin thin quad flat pack (TQFP). Most pins on this MCU serve two or more functions, as described in the following paragraphs. Refer to Figure 2-1, which shows the pin assignments for the 100-pin package.

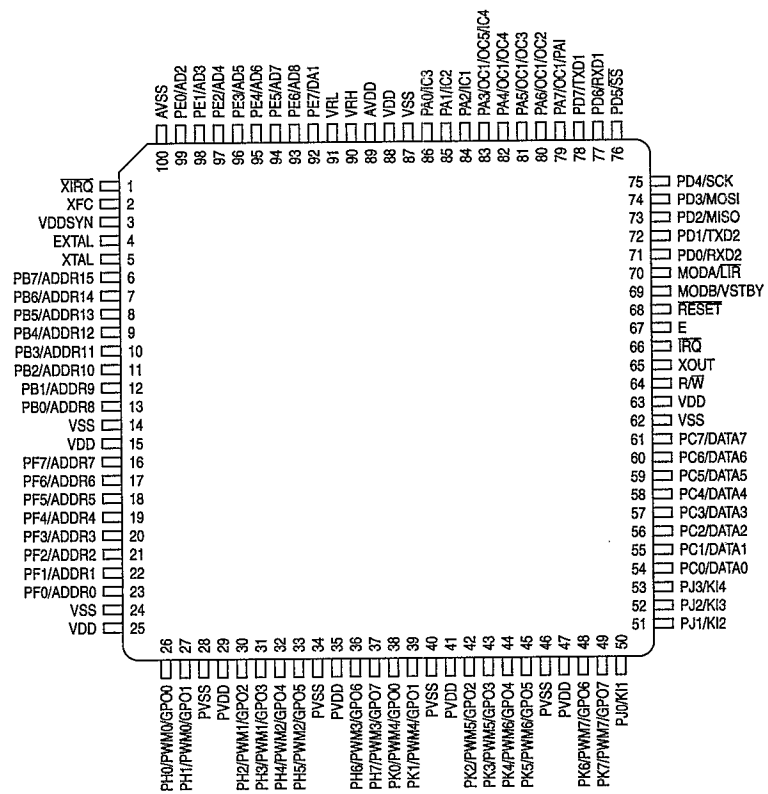


Figure 2-1 100-pin TQFP

2.1 VDD and VSS

Power is supplied to the microcontroller via these pins. VDD is the positive supply and VSS is ground. The MCU operates from a single 5V (nominal) power supply.

It is in the nature of CMOS designs that very fast signal transitions occur on the MCU pins. These short rise and fall times place very high short-duration current demands on the power supply. To prevent noise problems, special care must be taken to provide good power supply bypassing at the MCU. Bypass capacitors should have good high-frequency characteristics and be as close to the MCU as possible. Bypassing requirements vary, depending on how heavily the MCU pins are loaded.

The MC68HC11KG4 has five VDD pins and five VSS pins. One pair of these pins is reserved for supplying power to the analog-to-digital converter (AVDD, AVSS); the remaining pins are used for the internal logic, and to supply power to the port logic on either half of the chip. There are also four PVDD and four PVSS pins which are used to supply 8.5V to the ports H and K power driver output circuitry on the PWM timer module.

2.2 $\overline{\text{RESET}}$

An active low bidirectional control signal, $\overline{\text{RESET}}$, acts as an input to initialize the MCU to a known start-up state. It also acts as an open-drain output to indicate that an internal failure has been detected in either the clock monitor or the COP watchdog circuit. The CPU distinguishes between internal and external reset conditions by sensing whether the reset pin rises to a logic one in less than four E clock cycles after an internal reset has been released. It is therefore not advisable to connect an external resistor-capacitor (RC) power-up delay circuit to the reset pin of M68HC11 devices because the circuit charge time constant can cause the device to misinterpret the type of reset that occurred. Refer to Section 5 for further information.

Figure 2-2 illustrates a typical reset circuit that includes an external switch together with a low voltage inhibit circuit, to prevent power transitions or RAM or EEPROM corruption.

For the MC68HC11KG4, this external circuitry can be ommitted when the LVR circuitry is enabled.

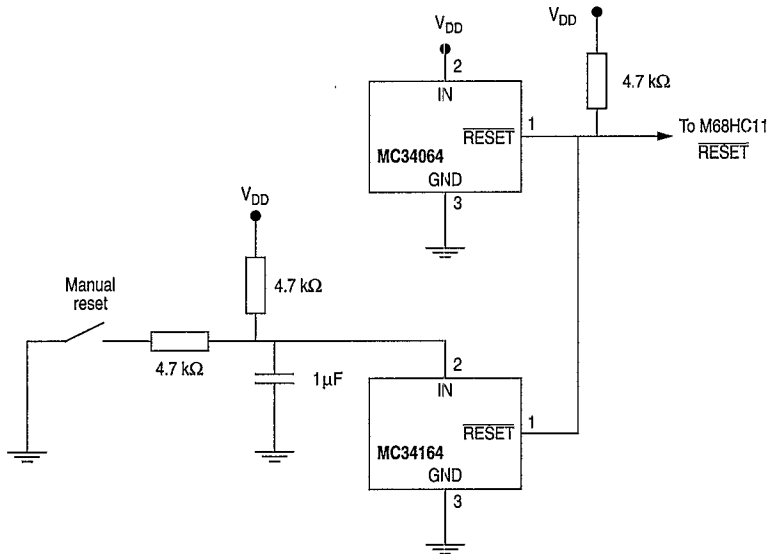


Figure 2-2 External reset circuitry

2.3 Crystal driver and external clock input (XTAL, EXTAL)

These two pins provide the interface for either a crystal or a CMOS compatible clock to control the internal clock generator circuitry. If the PLL circuit is not being used to provide the E clock, the frequency applied to these pins must be four times higher than the desired E clock rate. Figure 2-3 shows oscillator connections that should be used when the PLL is disabled, and Figure 2-4 shows the connections that should be used when the PLL is enabled.

The XTAL pin is normally left unconnected when an external CMOS compatible clock input is connected to the EXTAL pin. The XTAL output is normally intended to drive only a crystal. The XTAL output can be buffered with a high-impedance buffer, or it can be used to drive the EXTAL input of another M68HC11 family device (unless the PLL circuit is in use).

In all cases, use caution when designing circuitry associated with the oscillator pins.

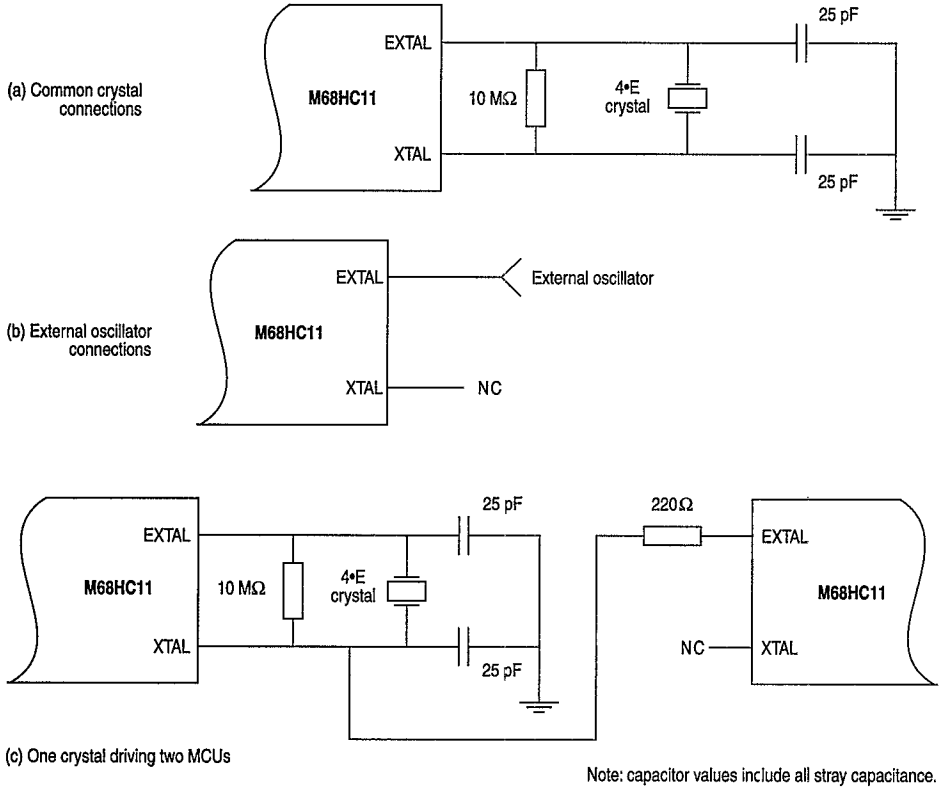
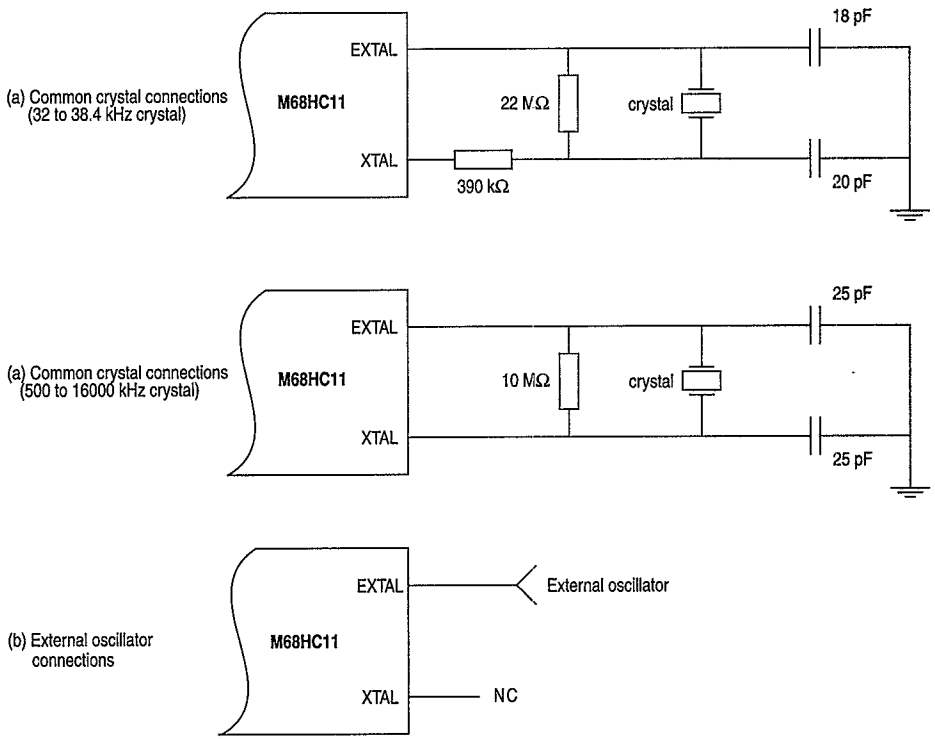


Figure 2-3 Oscillator connections (VDDSYN = 0, PLL disabled)



Note: capacitor values include all stray capacitance.
Note: all values of capacitance and resistance shown are approximate; exact values must be calculated knowing the crystal parameters and the expected voltage and temperature ranges.

Figure 2-4 Oscillator connections (VDDSYN = 1, PLL enabled)

2.4 E clock output (E)

E is the output connection for the internally generated E clock. The signal from E is used as a timing reference. The frequency of the E clock output is one quarter that of the input frequency at the XTAL and EXTAL pins (except when the PLL is used as the clock source). When E clock output is low, an internal process is taking place; when it is high, data is being accessed. All clocks, including the E clock, are halted when the MCU is in STOP mode. The E clock output can be turned off in single-chip modes to reduce the effects of RFI (see Section 4.3.2.5).

2.5 Phase-locked loop (XFC, VDDSYN)

The XFC and VDDSYN pins are the inputs for the on-chip PLL (phase-locked loop) circuitry. On reset, all system clocks are derived from the internal EXTAL signal (EXTALi). If enabled (VDDSYN high), the PLL uses the EXTALi frequency as a reference to generate a clock frequency that can be varied under software control. The user may choose to use the PLL output instead of EXTALi as the source clock for the system.

The PLL consists of a variable bandwidth loop filter, a voltage controlled oscillator (VCO), a feedback frequency divider and a digital phase detector. PLL functions are controlled by the PLLCR and SYNR registers. A block diagram of the PLL circuit is shown in Figure 2-5; refer also to Figure 10-1.

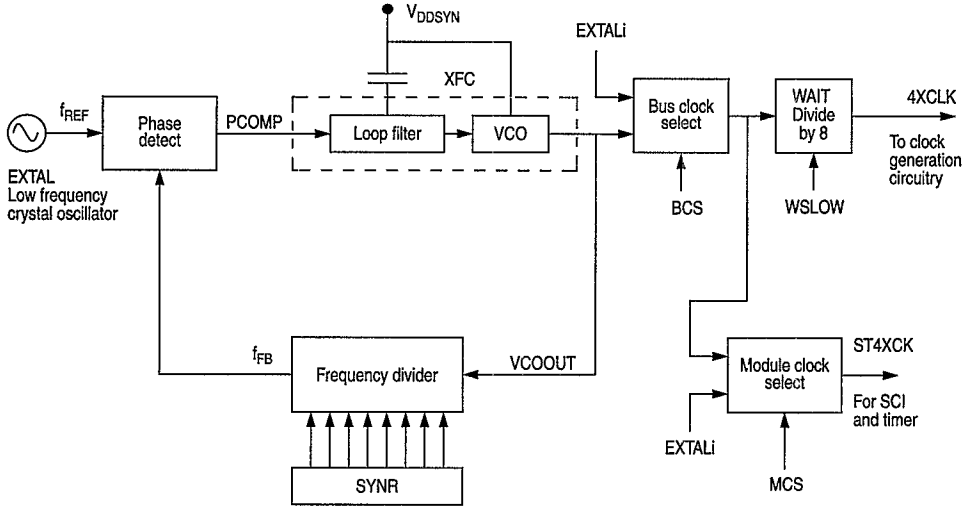


Figure 2-5 PLL circuit

2.5.1 PLL operation

The voltage controlled oscillator (VCO) generates the PLL output frequency VCOOUT. This signal is fed back through a frequency divider, which divides the signal frequency by a factor determined by the contents of the SYNR register, to produce the feedback signal f_{FB}. This signal is input to the phase detector along with the reference signal, f_{REF}. The phase detector generates a control signal (PCOMP) which is a function of the phase difference between f_{FB} and f_{REF}. PCOMP is then integrated, and the resultant dc voltage (visible on XFC) is applied to the VCO, modifying the output signal VCOOUT to lock it in phase with f_{REF}.

Note: Because the operation of the PLL depends on repeated adjustments to the voltage input to the VCO, a time t_{PLLS} is required for the stabilization of the output frequency.

The state of two bits in the PLLCR register, MCS and BCS, determine whether VCOOUT or EXTALi is used for the system clocks.

VDDSYN is the power supply pin for the PLL. Connecting it high enables the internal low frequency oscillator circuitry designed for the PLL. The PLL has been designed particularly for use with 4.2MHz crystals, though other values may be used. The maximum recommended crystal frequency for PLL operation is 4.2MHz. Above this frequency VDDSYN should be grounded to disable the PLL and enable the high frequency oscillator circuit; in this state EXTAL is designed for 16MHz operation and XFC may be left unconnected.

The PLL filter has two bandwidths that can be manually selected under control of the BWC bit in PLLCR. Whenever the PLL is first enabled, the wide bandwidth mode should be used, to enable the PLL frequency to ramp up quickly. After a time t_{PLLS} has elapsed, the filter can be switched to the narrow bandwidth mode, to make the final frequency more stable.

Caution: Bit 5 of the PLLCR (AUTO) must be cleared before an attempt is made to use BWC; manual bandwidth control should always be used.

2.5.2 Synchronization of PLL with subsystems

If the MCS bit in PLLCR is set, then the SCI and timer clocks run off the PLL output (4XCLK) as does the CPU. If MCS is cleared, then the timer and SCI subsystems operate off the EXTALi frequency, but are accessed by the CPU relative to the internal PH2 signal. In this case, although EXTALi is used as the reference for the PLL, the PH2 clock and the module clocks for the timer and the SCI are not synchronized. In order to ensure synchronized data, special circuitry has been incorporated into both subsystems.

2.5.3 Changing the PLL frequency

Changing the PLL frequency is possible in normal modes and in special modes on the MC68HC11KG4 (provided the SYNR mask option is enabled). The PLL output frequency can be changed by altering the contents of the SYNR register (see Section 2.5.4.2). To prevent possible bursts of high frequency operation during the reconfiguration of the PLL, the following sequence should be performed:

1) Switch to the low frequency bus rate (BCS = 0).
2) Disable the PLL (PLLON = 0).
3) Change the value in SYNR.
4) Enable the PLL (PLLON = 1).
5) Wait a time t_{PLLS} for the PLL frequency to stabilize.
6) Switch to the high frequency bus rate (BCS = 1).

2.5.4 PLL registers

Two registers are used to control the operation of the MC68HC11KG4 phase locked loop circuitry. These are the PLL control register and the synthesizer program register, each of which is described in the following paragraphs.

2.5.4.1 PLLCR — PLL control register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
PLL control (PLLCR)	$002E	PLLON	BCS	AUTO	BWC	VCOT	MCS	WSLOW	WEN	x011 1000

This read/write register contains two bits that are used to enable and disable the synthesizer and to switch from slow (EXTALi) to one of the fast speeds. Two other bits are used to control the filter bandwidth. The SCI, timer clock source and the slow clock for WAIT mode are also controlled by this register.

PLLON — PLL on

1 (set) – Switch PLL on.

0 (clear) – Switch PLL off.

This bit activates the synthesizer circuit without connecting it to the control circuit. This allows the circuit to stabilize before it drives the CPU clocks.

On reset, PLLON is forced low if the VDDSYN supply is low. If VDDSYN is at V_{DD}, PLLON is set by reset to allow the control loop to stabilize during power-up. PLLON cannot be cleared whilst using VCOOUT to drive the internal processor clock, i.e. when BCS is set.

BCS — Bus clock select

1 (set) – VCOOUT output drives the clock circuit (4XCLK).

0 (clear) – EXTALi drives the clock circuit (4XCLK).

This bit determines which signal drives the clock circuit generating the bus clocks. Once BCS has been altered it can take up to [1.5 EXTALi + 1.5 VCOOUT] cycles for the change in the clock to occur. Reset clears this bit.

Note: PLLON and BCS have built-in safeguards so that VCOOUT cannot be selected as the clock source (BCS = 1) if the PLL is off (PLLON = 0). Similarly, the PLL cannot be turned off (PLLON = 0) if it is on and in use (BCS = 1). Turning the PLL on and selecting VCOOUT as the clock source therefore requires two independent writes to PLLCR.

AUTO — Automatic bandwidth control (Test mode only)

1 (set) – Automatic bandwidth control selected.

0 (clear) – Manual bandwidth control selected.

Reset sets this bit.

Caution: This bit must be cleared before an attempt is made to use BWC; manual bandwidth control should always be used.

BWC — Bandwidth control

1 (set) – Wide (high and low) bandwidth control selected.

0 (clear) – Narrow (low) bandwidth control selected.

Bandwidth selection can only be controlled by BWC when AUTO is cleared. After the PLL is first enabled, or after a change in frequency, a delay of t_{PLLS} is required before clearing BWC. The low bandwidth driver is always enabled, so this bit determines whether the high bandwidth driver is on or off. Reset sets this bit.

VCOT — VCO test (Test mode only)

1 (set) – Loop filter operates as specified by AUTO and BWC.

0 (clear) – Low bandwidth mode of the PLL filter is disabled.

This bit is used to isolate the loop filter from the VCO for testing purposes. VCOT is always set in user modes. This bit is writeable only in bootstrap and test modes. Reset sets this bit.

MCS — Module clock select

1 (set) – 4XCLK is the source for the SCI and timer divider chain.

0 (clear) – EXTALi is the source for the SCI and timer divider chain.

Reset clears this bit.

WSLOW — Slow frequency in WAIT

1 (set) – After stacking for WAIT, divide core clock frequency by eight.

0 (clear) – Do not alter core clock frequency during WAIT.

This bit determines whether the internal clocks derived from 4XCLK are divided down by a factor of eight on entering WAIT mode. Reset clears this bit.

WEN — WAIT enable

1 (set) – Low-power WAIT mode selected (PLL set to 'idle' in WAIT mode).

0 (clear) – Do not alter 4XCLK during WAIT mode.

This bit determines whether the 4XCLK is disconnected from VCOOUT during WAIT and connected to EXTALi. Reset clears this bit.

When WEN is set, the CPU will respond to a WAIT instruction by first stacking the relevant registers, then by clearing BCS and setting the PLL to 'idle', with modulus = 1. BWC is set so that the wide bandwidth control is selected.

Any interrupt, any reset, or the assertion of RAF (receiver active flag) in the SCI will allow the PLL to resume operating at the frequency specified in the SYNR. The user must set BCS after the PLL has had time to adjust (t_{PLLS}). If the SCI RE bit (receiver enable bit) is clear, then RAF cannot become set, hence the PLL will not resume normal operation. For a description of RAF and RE, see Section 7.

2.5.4.2 SYNR — Synthesizer program register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Synthesizer program (SYNR)	$002F	SYNX1	SYNX0	SYNY5	SYNY4	SYNY3	SYNY2	SYNY1	SYNY0	0000 0001

The PLL frequency synthesizer multiplies the frequency of the input oscillator. The multiplication factor is software programmable via a loop divider, which consists of a six-bit modulo N counter, with a further two bit scaling factor.

Bits in SYNR can be read at any time, but can normally only be written to when SMOD = 1. There is however a mask option on the MC68HC11KG4 which determines whether or not the SYNR register can be programmed in normal modes. This option is programmed during manufacture, and must be stated on the order form.

On the MC68HC11KG4 the SYNR register can be written in normal as well as in special modes if the SYNR mask option is enabled.

The multiplication factor is given by $2(Y + 1)2^X$, where $0 \leq X \leq 3$ and $0 \leq Y \leq 63$.

Note: Exceeding recommended operating frequencies can result in indeterminate MCU operation.

SYNX[1:0]

These bits program the binary taps (divide by 1, 2, 4 and 8).

SYNY[5:0]

These bits program the six-bit modulo N (1 to 64) counter.

Note: The resolution of the multiplication factors decreases by a factor of two, as X increases:

X	Y	Possible multipliers
0	0 – 63	2, 4, 6, 8, ..., 128
1	0 – 63	4, 8, 12, 16, ..., 256
2	0 – 63	8, 16, 24, 32, ..., 512
3	0 – 63	16, 32, 48, 64, ..., 1024

2.6 Interrupt request ($\overline{IRQ}$)

The $\overline{IRQ}$ input provides a means of applying asynchronous interrupt requests to the MCU. Either falling-edge-sensitive triggering or level-sensitive triggering is program selectable (OPTION register). $\overline{IRQ}$ is always configured to level-sensitive-triggering at reset.

Note: Connect an external pull-up resistor, typically 4.7 kΩ, to V_{DD} when $\overline{IRQ}$ is used in a level sensitive wired-OR configuration. See also Section 2.7.

2.7 Nonmaskable interrupt ($\overline{XIRQ}$)

The $\overline{XIRQ}$ input provides a means of requesting a nonmaskable interrupt after reset initialization. During reset, the X bit in the condition code register (CCR) is set and any interrupt is masked until MCU software enables it. $\overline{XIRQ}$ is often used as a power loss detect interrupt.

$\overline{IRQ}$ must be configured for level sensitive operation if there is more than one source of interrupt. Whenever $\overline{XIRQ}$ or $\overline{IRQ}$ is used with multiple interrupt sources, each source must drive the interrupt input with an open-drain type of driver to avoid contention between outputs. There should be a single pull-up resistor near the MCU interrupt input pin (typically 4.7 kΩ). There must also be an interlock mechanism at each interrupt source so that the source holds the interrupt line low until

the MCU recognizes and acknowledges the interrupt request. If one or more interrupt source is still pending after the MCU services a request, the interrupt line will still be held low and the MCU will be interrupted again as soon as the interrupt mask bit in the MCU is cleared (normally upon return from an interrupt). Refer to Section 5.

2.8 MODA and MODB (MODA/$\overline{\text{LIR}}$ and MODB/VSTBY)

During reset, MODA and MODB select one of the four operating modes. Refer to Section 4.

After the operating mode has been selected, the $\overline{\text{LIR}}$ pin provides an open-drain output (driven low) to indicate that execution of an instruction has begun. In order to detect consecutive instructions in a high-speed application, this signal drives high for a short time to prevent false triggering. A series of E clock cycles occurs during execution of each instruction. The $\overline{\text{LIR}}$ signal goes low during the first E clock cycle of each instruction (opcode fetch). This output is provided for assistance in program debugging and its operation is controlled by the LIRDV bit in the OPT2 register.

The VSTBY pin is used to input RAM stand-by power. The MCU is powered from the VDD pin unless the difference between the level of VSTBY and VDD is greater than one MOS threshold (about 0.7 volts). When these voltages differ by more than 0.7 volts, the internal RAM and part of the reset logic are powered from VSTBY rather than VDD. This allows RAM contents to be retained without VDD power applied to the MCU. Reset must be driven low before V_{DD} is removed and must remain low until V_{DD} has been restored to a valid level.

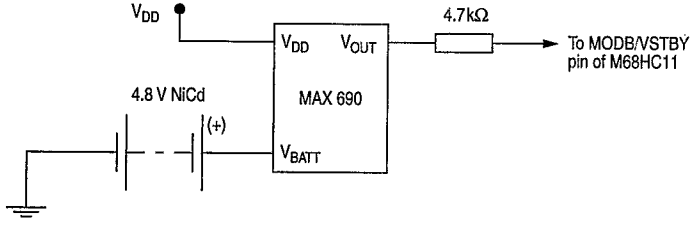


Figure 2-6 RAM stand-by connections

2.9 VRH and VRL

These pins provide the reference voltages for the analog-to-digital converter.

2.10 R/$\overline{W}$

In expanded and test modes, R/$\overline{W}$ performs the read/write function. R/$\overline{W}$ signals the direction of transfers on the external data bus. A high on this pin indicates that a read cycle is in progress.

In single chip mode the R/$\overline{W}$ signal is driven high.

2.11 XOUT

This pin outputs the buffered CLKX signal, if enabled by the XCLK bit in the CONFIG register. The frequency of CLKX can be selected using two bits in the OPT2 register (XDV1 and XDV0). On reset, CLKX has the same frequency as EXTAL. Refer to Section 4.

Note that the phase relationship between CLKX and EXTAL cannot be predicted.

2.12 Port signals

The MC68HC11KG4 includes 68 pins that are arranged into eight 8-bit ports (A, B, C, D, E, F, H and K) and one 4-bit port (J).

Ports A, B, C, D, F and J are fully bidirectional; port E pins are input only, except for PE7 which may be used as an output for the D/A converter, and ports H and K are output only. Each of the bidirectional ports serves a purpose other than I/O, depending on the operating mode or peripheral function selected. Note that ports B, C, and F are available for I/O functions only in single chip and bootstrap modes. Refer to Table 2-1 for details of the port signals' functions in different operating modes.

Note: When using the information about port functions, do not confuse pin function with the electrical state of the pin at reset. All general purpose I/O pins configured as inputs at reset are in a high-impedance state. Port data registers reflect the functional state of the port at reset. The pin function is mode dependent.

2.12.1 Port A

Port A is an 8-bit general purpose I/O port with a data register (PORTA) and a data direction register (DDRA). Port A pins share functions with the 16-bit timer system (see Section 9 for further information). PORTA can be read at any time and always returns the pin level. If written, PORTA stores the data in internal latches. The pins are driven only if they are configured as outputs. Writes to PORTA do not change the pin state when the pins are configured for timer output compares.

Table 2-1 Port signal functions

Port/bit	Single chip and bootstrap mode	Expanded multiplexed and special test mode
PA7	PA7/PAI and/or OC1	
PA6	PA6/OC2 and/or OC1	
PA5	PA5/OC3 and/or OC1	
PA4	PA4/OC4 and/or OC1	
PA3	PA3/OC5/IC4 and/or OC1	
PA2	PA2/IC1	
PA1	PA1/IC2	
PA0	PA0/IC3	
PB[7:0]	PB[7:0]	ADDR[15:8]
PC[7:0]	PC[7:0]	DATA[7:0]
PD7	PD7/TXD1	
PD6	PD6/RXD1	
PD5	PD5/$\overline{SS}$	
PD4	PD4/SCK	
PD3	PD3/MOSI	
PD2	PD2/MISO	
PD1	PD1/TXD2	
PD0	PD0/RXD2	
PE7	Input only/digital output	
PE[6:0]	Input only/analog inputs	
PF[7:0]	PF[7:0]	ADDR[7:0]
PH7	PH7/PWM3/GPO7	
PH6	PH6/PWM3/GPO6	
PH5	PH5/PWM2/GPO5	
PH4	PH4/PWM2/GPO4	
PH3	PH3/PWM1/GPO3	
PH2	PH2/PWM1/GPO2	
PH1	PH1/PWM0/GPO1	
PH0	PH0/PWM0/GPO0	
PJ3	PJ3/KI4	
PJ2	PJ2/KI3	
PJ1	PJ1/KI2	
PJ0	PJ0/KI1	
PK7	PK7/PWM7/GPO7	
PK6	PK6/PWM7/GPO6	
PK5	PK5/PWM6/GPO5	
PK4	PK4/PWM6/GPO4	
PK3	PK3/PWM5/GPO3	
PK2	PK2/PWM5/GPO2	
PK1	PK1/PWM4/GPO1	
PK0	PK0/PWM4/GPO0	

Out of reset, port A pins [7:0] are general purpose high-impedance inputs. When the functions associated with these pins are disabled, the bits in DDRA govern the I/O state of the associated pin. For further information, refer to Section 6 (I/O).

2.12.2 Port B

Port B is an 8-bit general purpose I/O port with a data register (PORTB) and a data direction register (DDRB). In single chip mode, port B pins are general purpose I/O pins (PB[7:0]). In expanded mode, port B pins act as the high-order address lines (ADDR[15:8]) of the address bus.

PORTB can be read at any time and always returns the pin level. If PORTB is written, the data is stored in internal latches. The pins are driven only if they are configured as outputs in single chip or bootstrap mode. For further information, refer to Section 6 (I/O).

Port B pins include on-chip pull-up devices which can be enabled or disabled via the port pull-up assignment register (PPAR).

2.12.3 Port C

Port C is an 8-bit general purpose I/O port with a data register (PORTC) and a data direction register (DDRC). In single chip mode, port C pins are general purpose I/O pins (PC[7:0]). In the expanded mode, port C pins are configured as data bus pins (DATA[7:0]).

PORTC can be read at any time; inputs return the pin level and outputs return the pin driver input level. If PORTC is written, the data is stored in internal latches. The pins are driven only if they are configured as outputs in single chip or bootstrap mode. Port C pins are general purpose inputs out of reset in single chip and bootstrap modes. In expanded and test modes, these pins are data bus lines out of reset.

The CWOM control bit in the OPT2 register disables port C's p-channel output drivers. Because the n-channel driver is not affected by CWOM, setting CWOM causes port C to become an open-drain-type output port suitable for wired-OR operation. In wired-OR mode (PORTC bits at logic level zero), the pins are actively driven low by the n-channel driver. When a port C bit is at logic level one, the associated pin is in a high impedance state as neither the n-channel nor the p-channel devices are active. It is customary to have an external pull-up resistor on lines that are driven by open-drain devices. Port C can only be configured for wired-OR operation when the MCU is in single chip mode. For further information, refer to Section 6 (I/O).

2.12.4 Port D

Port D, an 8-bit general purpose I/O port, has a data register (PORTD) and a data direction register (DDRD). The eight port D lines (PD[7:0]) can be used for general purpose I/O, for the serial

communications interfaces (SCI1, pins [7,6] and SCI2, pins [1,0]) and for the serial peripheral interface (SPI, pins [5:2]).

PORTD can be read at any time; inputs return the pin level and outputs return the pin driver input level. If PORTD is written, the data is stored in internal latches. The pins are driven only if port D is configured for general purpose output.

The DWOM bit in SPCR disables the p-channel output drivers of pins PD[5:2], and the WOMS bits in SCCR1 and S2CR2 disable those of pins PD[7,6] and PD[1,0] respectively. Because the n-channel driver is not affected by DWOM or WOMS, setting either bit causes the corresponding port D pins to become open-drain-type outputs suitable for wired-OR operation. In wired-OR mode (PORTD bits at logic level zero), the pins are actively driven low by the n-channel driver. When a port D bit is at logic level one, the associated pin is in a high impedance state as neither the n-channel nor the p-channel devices are active. It is customary to have an external pull-up resistor on lines that are driven by open-drain devices. Port D can be configured for wired-OR operation when the MCU is in single chip or expanded mode.

For further information, refer to Section 6 (I/O), Section 7 (SCI), and Section 8 (SPI).

2.12.5 Port E

Port E is an input-only port; pins [6:0] can also be used as the analog inputs for the analog-to-digital converter.

Pin PE7 can be used as the digital output of the digital-to-analog converter. When the digital-to-analog converter is not in use however, PE7 becomes an input-only line.

For further information, refer to Section 6 (I/O), Section 11(A/D) and Section 12(D/A).

2.12.6 Port F

Port F is an 8-bit general purpose I/O port with a data register (PORTF) and a data direction register (DDRF). In single chip mode, port F pins are general purpose I/O pins (PF[7:0]). In expanded mode, port F pins act as the low-order address lines (ADDR[7:0]) of the address bus.

PORTF can be read at any time and always returns the pin level. If PORTF is written, the data is stored in internal latches. The pins are driven only if they are configured as outputs in single chip or bootstrap mode.

Port F pins include on-chip pull-up devices that can be enabled or disabled via the port pull-up assignment register (PPAR). For further information, refer to Section 6 (I/O).

2.12.7 Port H

Port H is an 8-bit general purpose output port with a nominal 8.5V supply voltage and a data register (PORTH). Port H pins PH[7:0] can either be used as output pins or as pulse-width modulation channels, depending on the state of the corresponding bit in the PWEN register.

PORTH can be read at any time and always returns the port H data latch level. If PORTH is written, the data is stored in internal latches.

The port H mismatch register (PTHMS) indicates any short circuit cases on port H. PTHMS can be read at any time; a logic 1 at one of the register bits indicates a short circuit on the corresponding port H pin. PTHMS bits can be cleared by setting the corresponding bit in PTHMS. For further information, refer to Section 6 (I/O) and Section 9 (PWM Timer).

2.12.8 Port J

Port J is a 4-bit general purpose I/O port with a data register (PORTJ) and a data direction register (DDRJ).

PORTJ can be read at any time; inputs return the pin level and outputs return the pin driver input level. If PORTJ is written, the data is stored in internal latches. The pins are driven only if port J is configured for general purpose output.

The four port J pins can be configured to provide a keyboard interrupt function. This function is only active for port pins configured as inputs.

2.12.9 Port K

Port K is an 8-bit general purpose output port with a nominal 8.5V supply voltage and a data register (PORTK). Port K pins PK[7:0] can either be used as output pins or as pulse-width modulation channels, depending on the state of the corresponding bit in the PWEN register.

PORTK can be read at any time and always returns the port K data latch level. If PORTK is written, the data is stored in internal latches.

The port K mismatch register (PTKMS) indicates any short circuit cases on port K. PTKMS can be read at any time; a logic 1 at one of the register bits indicates a short circuit on the corresponding port K pin. PTKMS bits can be cleared by setting the corresponding bit in PTKMS. For further information, refer to Section 6(I/O) and Section 9(PWM Timer).

THIS PAGE LEFT BLANK INTENTIONALLY

3
CPU CORE AND INSTRUCTION SET

This section discusses the M68HC11 central processing unit (CPU) architecture, its addressing modes and the instruction set. For more detailed information on the instruction set, refer to the ***M68HC11 Reference Manual (M68HC11RM/AD)***.

The CPU is designed to treat all peripheral, I/O and memory locations identically, as addresses in the 64K byte memory map. This is referred to as memory-mapped I/O. There are no special instructions for I/O that are separate from those used for memory. This architecture also allows accessing an operand from an external memory location with no execution-time penalty.

3.1 Registers

M68HC11 CPU registers are an integral part of the CPU and are not addressed as if they were memory locations. The seven registers are shown in Figure 3-1 and are discussed in the following paragraphs.

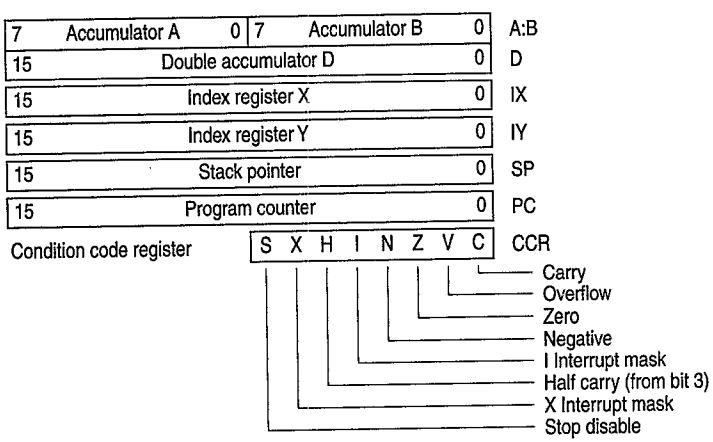


Figure 3-1 Programming model

3.1.1 Accumulators A, B and D

Accumulators A and B are general purpose 8-bit registers that hold operands and results of arithmetic calculations or data manipulations. For some instructions, these two accumulators are treated as a single double-byte (16-bit) accumulator called accumulator D. Although most operations can use accumulators A or B interchangeably, the following exceptions apply:

- The ABX and ABY instructions add the contents of 8-bit accumulator B to the contents of 16-bit register X or Y, but there are no equivalent instructions that use A instead of B.
- The TAP and TPA instructions transfer data from accumulator A to the condition code register, or from the condition code register to accumulator A, however, there are no equivalent instructions that use B rather than A.
- The decimal adjust accumulator A (DAA) instruction is used after binary-coded decimal (BCD) arithmetic operations, but there is no equivalent BCD instruction to adjust accumulator B.
- The add, subtract, and compare instructions associated with both A and B (ABA, SBA, and CBA) only operate in one direction, making it important to plan ahead to ensure the correct operand is in the correct accumulator.

3.1.2 Index register X (IX)

The IX register provides a 16-bit indexing value that can be added to the 8-bit offset provided in an instruction to create an effective address. The IX register can also be used as a counter or as a temporary storage register.

3.1.3 Index register Y (IY)

The 16-bit IY register performs an indexed mode function similar to that of the IX register. However, most instructions using the IY register require an extra byte of machine code and an extra cycle of execution time because of the way the opcode map is implemented. Refer to Section 3.3 for further information.

3.1.4 Stack pointer (SP)

The M68HC11 CPU has an automatic program stack. This stack can be located anywhere in the address space and can be any size up to the amount of memory available in the system. Normally the SP is initialized by one of the first instructions in an application program. The stack is configured as a data structure that grows downward from high memory to low memory. Each time a new byte is pushed onto the stack, the SP is decremented. Each time a byte is pulled from the stack, the SP is incremented. At any given time, the SP holds the 16-bit address of the next free location in the stack. Figure 3-2 is a summary of SP operations.

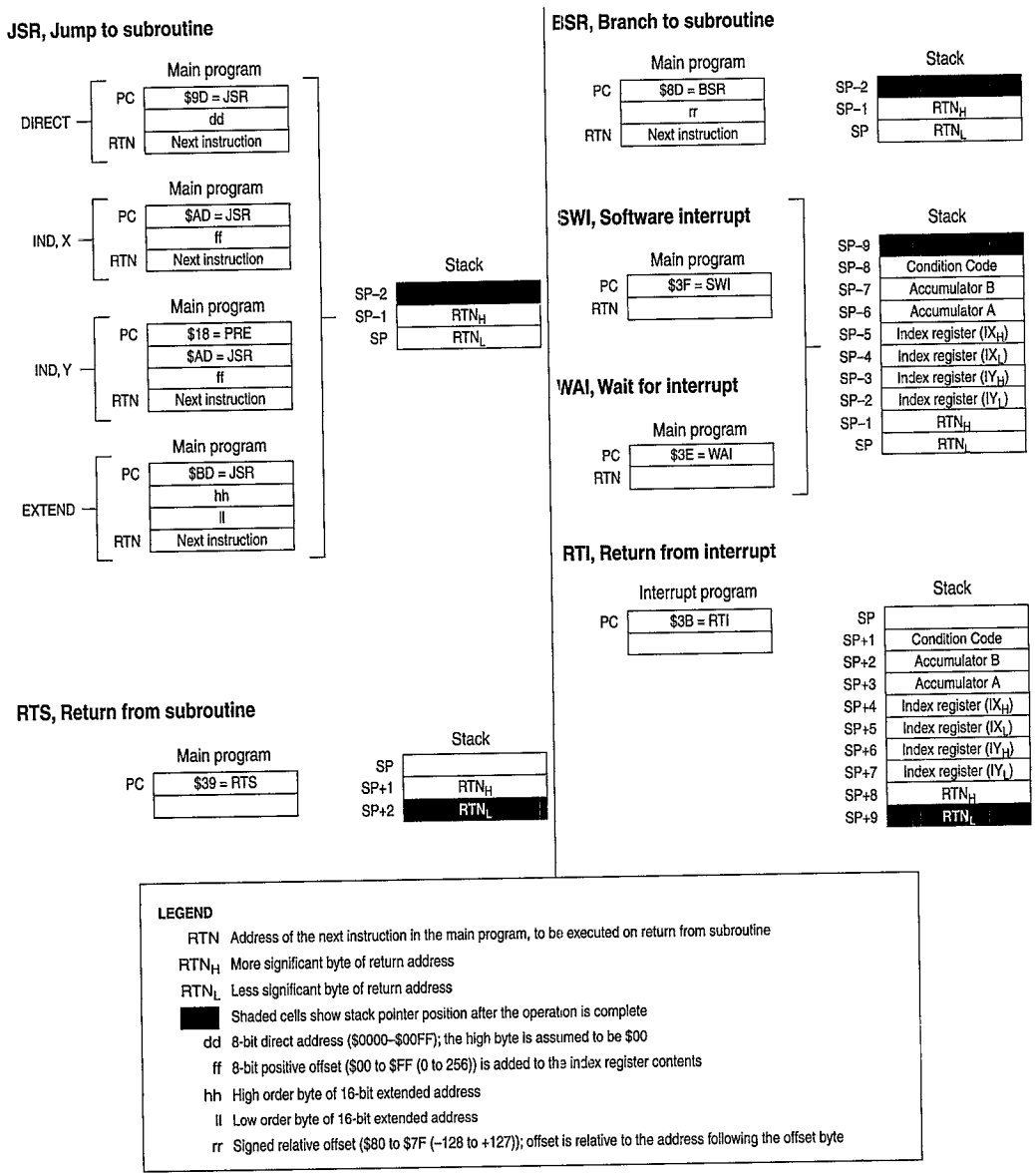


Figure 3-2 Stacking operations

When a subroutine is called by a jump to subroutine (JSR) or branch to subroutine (BSR) instruction, the address of the instruction after the JSR or BSR is automatically pushed onto the stack, less significant byte first. When the subroutine is finished, a return from subroutine (RTS) instruction is executed. The RTS pulls the previously stacked return address from the stack, and loads it into the program counter. Execution then continues at this recovered return address.

When an interrupt is recognized, the current instruction finishes normally, the return address (the current value in the program counter) is pushed onto the stack, all of the CPU registers are pushed onto the stack, and execution continues at the address specified by the vector for the interrupt. At the end of the interrupt service routine, an RTI instruction is executed. The RTI instruction causes the saved registers to be pulled off the stack in reverse order. Program execution resumes at the return address.

There are instructions that push and pull the A and B accumulators and the X and Y index registers. These instructions are often used to preserve program context. For example, pushing accumulator A onto the stack when entering a subroutine that uses accumulator A, and then pulling accumulator A off the stack just before leaving the subroutine, ensures that the contents of a register will be the same after returning from the subroutine as it was before starting the subroutine.

3.1.5 Program counter (PC)

The program counter, a 16-bit register, contains the address of the next instruction to be executed. After reset, the program counter is initialized from one of six possible vectors, depending on operating mode and the cause of reset.

Table 3-1 Reset vector comparison

	POR or $\overline{\text{RESET}}$ pin	Clock monitor	COP watchdog
Normal	$FFFE, $FFFF	$FFFC, $FFFD	$FFFA, $FFFB
Test or Boot	$BFFE, $BFFF	$BFFE, $BFFF	$BFFE, $BFFF

3.1.6 Condition code register (CCR)

This 8-bit register contains five condition code indicators (C, V, Z, N, and H), two interrupt masking bits, (IRQ and XIRQ) and a stop disable bit (S). In the M68HC11 CPU, condition codes are automatically updated by most instructions. For example, load accumulator A (LDAA) and store accumulator A (STAA) instructions automatically set or clear the N, Z, and V condition code flags. Pushes, pulls, add B to X (ABX), add B to Y (ABY), and transfer/exchange instructions do not affect the condition codes. Refer to Table 3-2, which shows the condition codes that are affected by a particular instruction.

3.1.6.1 Carry/borrow (C)

The C-bit is set if the arithmetic logic unit (ALU) performs a carry or borrow during an arithmetic operation. The C-bit also acts as an error flag for multiply and divide operations. Shift and rotate instructions operate with and through the carry bit to facilitate multiple-word shift operations.

3.1.6.2 Overflow (V)

The overflow bit is set if an operation causes an arithmetic overflow. Otherwise, the V-bit is cleared.

3.1.6.3 Zero (Z)

The Z-bit is set if the result of an arithmetic, logic, or data manipulation operation is zero. Otherwise, the Z-bit is cleared. Compare instructions do an internal implied subtraction and the condition codes, including Z, reflect the results of that subtraction. A few operations (INX, DEX, INY, and DEY) affect the Z-bit and no other condition flags. For these operations, only = and ≠ conditions can be determined.

3.1.6.4 Negative (N)

The N-bit is set if the result of an arithmetic, logic, or data manipulation operation is negative; otherwise, the N-bit is cleared. A result is said to be negative if its most significant bit (MSB) is set (MSB = 1). A quick way to test whether the contents of a memory location has the MSB set is to load it into an accumulator and then check the status of the N-bit.

3.1.6.5 Interrupt mask (I)

The interrupt request (IRQ) mask (I-bit) is a global mask that disables all maskable interrupt sources. While the I-bit is set, interrupts can become pending, but the operation of the CPU continues uninterrupted until the I-bit is cleared. After any reset, the I-bit is set by default and can only be cleared by a software instruction. When an interrupt is recognized, the I-bit is set after the registers are stacked, but before the interrupt vector is fetched. After the interrupt has been serviced, a return from interrupt instruction is normally executed, restoring the registers to the values that were present before the interrupt occurred. Normally, the I-bit is zero after a return from interrupt is executed. Although the I-bit can be cleared within an interrupt service routine, 'nesting' interrupts in this way should only be done when there is a clear understanding of latency and of the arbitration mechanism. Refer to Section 5.

3.1.6.6 Half carry (H)

The H-bit is set when a carry occurs between bits 3 and 4 of the arithmetic logic unit during an ADD, ABA, or ADC instruction. Otherwise, the H-bit is cleared. Half carry is used during BCD operations.

3.1.6.7 X interrupt mask (X)

The XIRQ mask (X) bit disables interrupts from the $\overline{\text{XIRQ}}$ pin. After any reset, X is set by default and must be cleared by a software instruction. When an $\overline{\text{XIRQ}}$ interrupt is recognized, the X- and I-bits are set after the registers are stacked, but before the interrupt vector is fetched. After the interrupt has been serviced, an RTI instruction is normally executed, causing the registers to be restored to the values that were present before the interrupt occurred. The X interrupt mask bit is set only by hardware $\overline{\text{RESET}}$ or $\overline{\text{XIRQ}}$ acknowledge). X is cleared only by program instruction (TAP, where the associated bit of A is 0; or RTI, where bit 6 of the value loaded into the CCR from the stack has been cleared). There is no hardware action for clearing X.

3.1.6.8 Stop disable (S)

Setting the STOP disable (S) bit prevents the STOP instruction from putting the M68HC11 into a low-power stop condition. If the STOP instruction is encountered by the CPU while the S-bit is set, it is treated as a no-operation (NOP) instruction, and processing continues to the next instruction. S is set by reset — STOP disabled by default.

3.2 Data types

The M68HC11 CPU supports the following data types:

- Bit data
- 8-bit and 16-bit signed and unsigned integers
- 16-bit unsigned fractions
- 16-bit addresses

A byte is eight bits wide and can be accessed at any byte location. A word is composed of two consecutive bytes with the most significant byte at the lower value address. Because the M68HC11 is an 8-bit CPU, there are no special requirements for alignment of instructions or operands.

3.3 Opcodes and operands

The M68HC11 family of microcontrollers uses 8-bit opcodes. Each opcode identifies a particular instruction and associated addressing mode to the CPU. Several opcodes are required to provide each instruction with a range of addressing capabilities. Only 256 opcodes would be available if the range of values were restricted to the number able to be expressed in 8-bit binary numbers.

A four-page opcode map has been implemented to expand the number of instructions. An additional byte, called a prebyte, directs the processor from page 0 of the opcode map to one of the other three pages. As its name implies, the additional byte precedes the opcode.

A complete instruction consists of a prebyte, if any, an opcode, and zero, one, two, or three operands. The operands contain information the CPU needs for executing the instruction. Complete instructions can be from one to five bytes long.

3.4 Addressing modes

Six addressing modes; immediate, direct, extended, indexed, inherent, and relative, detailed in the following paragraphs, can be used to access memory. All modes except inherent mode use an effective address. The effective address is the memory address from which the argument is fetched or stored, or the address from which execution is to proceed. The effective address can be specified within an instruction, or it can be calculated.

3.4.1 Immediate (IMM)

In the immediate addressing mode an argument is contained in the byte(s) immediately following the opcode. The number of bytes following the opcode matches the size of the register or memory location being operated on. There are two, three, and four (if prebyte is required) byte immediate instructions. The effective address is the address of the byte following the instruction.

3.4.2 Direct (DIR)

In the direct addressing mode, the low-order byte of the operand address is contained in a single byte following the opcode, and the high-order byte of the address is assumed to be $00. Addresses $00–$FF are thus accessed directly, using two-byte instructions. Execution time is reduced by eliminating the additional memory access required for the high-order address byte. In most applications, this 256-byte area is reserved for frequently referenced data. In M68HC11 MCUs, the memory map can be configured for combinations of internal registers, RAM, or external memory to occupy these addresses.

3.4.3 Extended (EXT)

In the extended addressing mode, the effective address of the argument is contained in two bytes following the opcode byte. These are three-byte instructions (or four-byte instructions if a prebyte is required). One or two bytes are needed for the opcode and two for the effective address.

3.4.4 Indexed (IND, X; IND, Y)

In the indexed addressing mode, an 8-bit unsigned offset contained in the instruction is added to the value contained in an index register (IX or IY) — the sum is the effective address. This addressing mode allows referencing any memory location in the 64Kbyte address space. These are two- to five-byte instructions, depending on whether or not a prebyte is required.

3.4.5 Inherent (INH)

In the inherent addressing mode, all the information necessary to execute the instruction is contained in the opcode. Operations that use only the index registers or accumulators, as well as control instructions with no arguments, are included in this addressing mode. These are one or two-byte instructions.

3.4.6 Relative (REL)

The relative addressing mode is used only for branch instructions. If the branch condition is true, an 8-bit signed offset included in the instruction is added to the contents of the program counter to form the effective branch address. Otherwise, control proceeds to the next instruction. These are usually two-byte instructions.

3.5 Instruction set

Refer to Table 3-2, which shows all the M68HC11 instructions in all possible addressing modes. For each instruction, the table shows the operand construction, the number of machine code bytes, and execution time in CPU E clock cycles.

Table 3-2 Instruction set (Page 1 of 6)

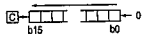
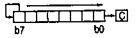
Mnemonic	Operation	Description	Addressing mode	Instruction: Opcode	Instruction: Operand	Instruction: Cycles	Condition codes: S	X	H	I	N	Z	V	C
ABA	Add accumulators	A + B ⇒ A	INH	1B	—	2	—	—	Δ	—	Δ	Δ	Δ	Δ
ABX	Add B to X	IX + (00:B) ⇒ IX	INH	3A	—	3	—	—	—	—	—	—	—	—
ABY	Add B to Y	IY + (00:B) ⇒ IY	INH	18 3A	—	4	—	—	—	—	—	—	—	—
ADCA (opr)	Add with carry to A	A + M + C ⇒ A	A IMM A DIR A EXT A IND, X A IND, Y	89 99 B9 A9 18 A9	ii dd hh ll ff ff	2 3 4 4 5	—	—	Δ	—	Δ	Δ	Δ	Δ
ADCB (opr)	Add with carry to B	B + M + C ⇒ B	B IMM B DIR B EXT B IND, X B IND, Y	C9 D9 F9 E9 18 E9	ii dd hh ll ff ff	2 3 4 4 5	—	—	Δ	—	Δ	Δ	Δ	Δ
ADDA (opr)	Add memory to A	A + M ⇒ A	A IMM A DIR A EXT A IND, X A IND, Y	8B 9B BB AB 18 AB	ii dd hh ll ff ff	2 3 4 4 5	—	—	Δ	—	Δ	Δ	Δ	Δ
ADDB (opr)	Add memory to B	B + M ⇒ B	B IMM B DIR B EXT B IND, X B IND. Y	CB DB FB EB 18 EB	ii dd hh ll ff ff	2 3 4 4 5	—	—	Δ	—	Δ	Δ	Δ	Δ
ADDD (opr)	Add 16-bit to D	D + (M:M+1) ⇒ D	IMM DIR EXT IND, X IND, Y	C3 D3 F3 E3 18 E3	jj kk dd hh ll ff ff	4 5 6 6 7	—	—	—	—	Δ	Δ	Δ	Δ
ANDA (opr)	AND A with memory	A • M ⇒ A	A IMM A DIR A EXT A IND, X A IND, Y	84 94 B4 A4 18 A4	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	0	—
ANDB (opr)	AND B with memory	B • M ⇒ B	B IMM B DIR B EXT B IND, X B IND, Y	C4 D4 F4 E4 18 E4	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	0	—
ASL (opr)	Arithmetic shift left		EXT IND, X IND, Y	78 68 18 68	hh ll ff ff	6 6 7	—	—	—	—	Δ	Δ	Δ	Δ
ASLA	Arithmetic shift left A		A INH	48	—	2	—	—	—	—	Δ	Δ	Δ	Δ
ASLB	Arithmetic shift left B		B INH	58	—	2	—	—	—	—	Δ	Δ	Δ	Δ
ASLD	Arithmetic shift left D		INH	05	—	3	—	—	—	—	Δ	Δ	Δ	Δ
ASR	Arithmetic shift right		EXT IND, X IND, Y	77 67 18 67	hh ll ff ff	6 6 7	—	—	—	—	Δ	Δ	Δ	Δ
ASRA	Arithmetic shift right A		A INH	47	—	2	—	—	—	—	Δ	Δ	Δ	Δ
ASRB	Arithmetic shift right B		B INH	57	—	2	—	—	—	—	Δ	Δ	Δ	Δ
BCC (rel)	Branch if carry clear	C = 0 ?	REL	24	rr	3	—	—	—	—	—	—	—	—
BCLR (opr) (msk)	Clear bit(s)	M • ($\overline{mm}$) ⇒ M	DIR IND, X IND, Y	15 1D 18 1D	dd mm ff mm ff mm	6 7 8	—	—	—	—	Δ	Δ	0	—
BCS (rel)	Branch if carry set	C = 1 ?	REL	25	rr	3	—	—	—	—	—	—	—	—
BEQ (rel)	Branch if equal to zero	Z = 1 ?	REL	27	rr	3	—	—	—	—	—	—	—	—
BGE (rel)	Branch if ≥ zero	N ⊕ V = 0 ?	REL	2C	rr	3	—	—	—	—	—	—	—	—
BGT (rel)	Branch if > zero	Z + (N ⊕ V) = 0 ?	REL	2E	rr	3	—	—	—	—	—	—	—	—
BHI (rel)	Branch if higher	C + Z = 0 ?	REL	22	rr	3	—	—	—	—	—	—	—	—

Table 3-2 Instruction set (Page 2 of 6)

Mnemonic	Operation	Description	Addressing mode	Instruction			Condition codes							
				Opcode	Operand	Cycles	S	X	H	I	N	Z	V	C
BHS (rel)	Branch if higher or same	C = 0 ?	REL	24	rr	3	—	—	—	—	—	—	—	—
BITA (opr)	Bit(s) test A with memory	A • M	A IMM A DIR A EXT A IND, X A IND, Y	85 95 B5 A5 18 A5	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	0	—
BITB (opr)	Bit(s) test B with memory	B • M	B IMM B DIR B EXT B IND, X B IND, Y	C5 D5 F5 E5 18 E5	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	0	—
BLE (rel)	Branch if ≤ zero	Z + (N ⊕ V) = 1 ?	REL	2F	rr	3	—	—	—	—	—	—	—	—
BLO (rel)	Branch if lower	C = 1 ?	REL	25	rr	3	—	—	—	—	—	—	—	—
BLS (rel)	Branch if lower or same	C + Z = 1 ?	REL	23	rr	3	—	—	—	—	—	—	—	—
BLT (rel)	Branch if < zero	N ⊕ V = 1 ?	REL	2D	rr	3	—	—	—	—	—	—	—	—
BMI (rel)	Branch if minus	N = 1 ?	REL	2B	rr	3	—	—	—	—	—	—	—	—
BNE (rel)	Branch if ≠ zero	Z = 0 ?	REL	26	rr	3	—	—	—	—	—	—	—	—
BPL(rel)	Branch if plus	N = 0 ?	REL	2A	rr	3	—	—	—	—	—	—	—	—
BRA (rel)	Branch always	1 = 1 ?	REL	20	rr	3	—	—	—	—	—	—	—	—
BRCLR(opr) (msk) (rel)	Branch if bit(s) clear	M • mm = 0 ?	DIR IND, X IND, Y	13 1F 18 1F	dd mm rr ff mm rr ff mm rr	6 7 8	—	—	—	—	—	—	—	—
BRN (rel)	Branch never	1 = 0 ?	REL	21	rr	3	—	—	—	—	—	—	—	—
BRSET(opr) (msk) (rel)	Branch if bit(s) set	$\overline{M}$ • mm = 0 ?	DIR IND, X IND, Y	12 1E 18 1E	dd mm rr ff mm rr ff mm rr	6 7 8	—	—	—	—	—	—	—	—
BSET (opr) (msk)	Set bit(s)	M + mm ⇒ M	DIR IND, X IND, Y	14 1C 18 1C	dd mm ff mm ff mm	6 7 8	—	—	—	—	Δ	Δ	0	—
BSR (rel)	Branch to subroutine	see Figure 3-2	REL	8D	rr	6	—	—	—	—	—	—	—	—
BVC (rel)	Branch if overflow clear	V = 0 ?	REL	28	rr	3	—	—	—	—	—	—	—	—
BVS (rel)	Branch if overflow set	V = 1 ?	REL	29	rr	3	—	—	—	—	—	—	—	—
CBA	Compare A with B	A – B	INH	11	—	2	—	—	—	—	Δ	Δ	Δ	Δ
CLC	Clear carry bit	0 ⇒ C	INH	0C	—	2	—	—	—	—	—	—	—	0
CLI	Clear interrupt mask	0 ⇒ I	INH	0E	—	2	—	—	—	0	—	—	—	—
CLR (opr)	Clear memory byte	0 ⇒ M	DIR IND, X IND, Y	7F 6F 18 6F	hh ll ff ff	6 6 7	—	—	—	—	0	1	0	0
CLRA	Clear accumulator A	0 ⇒ A	A INH	4F	—	2	—	—	—	—	0	1	0	0
CLRB	Clear accumulator B	0 ⇒ B	B INH	5F	—	2	—	—	—	—	0	1	0	0
CLV	Clear overflow flag	0 ⇒ V	INH	0A	—	2	—	—	—	—	—	—	0	—
CMPA (opr)	Compare A with memory	A – M	A IMM A DIR A EXT A IND, X A IND, Y	81 91 B1 A1 18 A1	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	Δ	Δ
CMPB (opr)	Compare B with memory	B – M	B IMM B DIR B EXT B IND, X B IND, Y	C1 D1 F1 E1 18 E1	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	Δ	Δ
COM (opr)	Ones complement memory byte	$FF – M ⇒ M	EXT IND, X IND, Y	73 63 18 63	hh ll ff ff	6 6 7	—	—	—	—	Δ	Δ	0	1
COMA	Ones complement A	$FF – A ⇒ A	A INH	43	—	2	—	—	—	—	Δ	Δ	0	1
COMB	Ones complement B	$FF – B ⇒ B	B INH	53	—	2	—	—	—	—	Δ	Δ	0	1

Table 3-2 Instruction set (Page 3 of 6)

Mnemonic	Operation	Description	Addressing mode	Instruction			Condition codes							
				Opcode	Operand	Cycles	S	X	H	I	N	Z	V	C
CPD (opr)	Compare D with memory (16-bit)	D – (M:M+1)	IMM DIR EXT IND, X IND, Y	1A 83 1A 93 1A B3 1A A3 CD A3	jj kk dd hh ll ff ff	5 6 7 7 7	—	—	—	—	Δ	Δ	Δ	Δ
CPX (opr)	Compare IX with memory (16-bit)	IX – (M:M+1)	IMM DIR EXT IND, X IND, Y	8C 9C BC AC CD AC	jj kk dd hh ll ff ff	4 5 6 6 7	—	—	—	—	Δ	Δ	Δ	Δ
CPY (opr)	Compare IY with memory (16-bit)	IY – (M:M+1)	IMM DIR EXT IND, X IND, Y	18 8C 18 9C 18 BC 1A AC 18 AC	jj kk dd hh ll ff ff	5 6 7 7 7	—	—	—	—	Δ	Δ	Δ	Δ
DAA	Decimal adjust A	adjust sum to BCD	INH	19	—	2	—	—	—	—	Δ	Δ	?	Δ
DEC (opr)	Decrement memory byte	M – 1 ⇒ M	EXT IND, X IND, Y	7A 6A 18 6A	hh ll ff ff	6 6 7	—	—	—	—	Δ	Δ	Δ	—
DECA	Decrement accumulator A	A – 1 ⇒ A	A INH	4A	—	2	—	—	—	—	Δ	Δ	Δ	—
DECB	Decrement accumulator B	B – 1 ⇒ B	B INH	5A	—	2	—	—	—	—	Δ	Δ	Δ	—
DES	Decrement stack pointer	SP – 1 ⇒ SP	INH	34	—	3	—	—	—	—	—	—	—	—
DEX	Decrement index register X	IX – 1 ⇒ IX	INH	09	—	3	—	—	—	—	—	Δ	—	—
DEY	Decrement index register Y	IY – 1 ⇒ IY	INH	18 09	—	4	—	—	—	—	—	Δ	—	—
EORA (opr)	Exclusive OR A with memory	A ⊕ M ⇒ A	A IMM A DIR A EXT A IND, X A IND, Y	88 98 B8 A8 18 A8	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	0	—
EORB (opr)	Exclusive OR B with memory	B ⊕ M ⇒ A	B IMM B DIR B EXT B IND, X B IND, Y	C8 D8 F8 E8 18 E8	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	0	—
FDIV	Fractional divide, 16 by 16	D / IX ⇒ IX; r ⇒ D	INH	03	—	41	—	—	—	—	—	Δ	Δ	Δ
IDIV	Integer divide, 16 by 16	D / IX ⇒ IX; r ⇒ D	INH	02	—	41	—	—	—	—	—	Δ	0	Δ
INC (opr)	Increment memory byte	M + 1 ⇒ M	EXT IND, X IND, Y	7C 6C 18 6C	hh ll ff ff	6 6 7	—	—	—	—	Δ	Δ	Δ	—
INCA	Increment accumulator A	A + 1 ⇒ A	A INH	4C	—	2	—	—	—	—	Δ	Δ	Δ	—
INCB	Increment accumulator B	B + 1 ⇒ B	B INH	5C	—	2	—	—	—	—	Δ	Δ	Δ	—
INS	Increment stack pointer	SP + 1 ⇒ SP	INH	31	—	3	—	—	—	—	—	—	—	—
INX	Increment index register X	IX + 1 ⇒ IX	INH	08	—	3	—	—	—	—	—	Δ	—	—
INY	Increment index register Y	IY + 1 ⇒ IY	INH	18 08	—	4	—	—	—	—	—	Δ	—	—
JMP (opr)	Jump	see Figure 3-2	EXT IND, X IND, Y	7E 6E 18 6E	hh ll ff ff	3 3 4	—	—	—	—	—	—	—	—
JSR (opr)	Jump to subroutine	see Figure 3-2	DIR EXT IND, X IND, Y	9D BD AD 18 AD	dd hh ll ff ff	5 6 6 7	—	—	—	—	—	—	—	—
LDAA (opr)	Load accumulator A	M ⇒ A	A IMM A DIR A EXT A IND, X A IND, Y	86 96 B6 A6 18 A6	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	0	—

Table 3-2 Instruction set (Page 4 of 6)

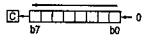
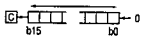
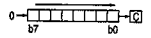
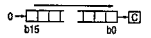
Mnemonic	Operation	Description	Addressing mode	Instruction			Condition codes							
				Opcode	Operand	Cycles	S	X	H	I	N	Z	V	C
LDAB (opr)	Load accumulator B	M ⇒ B	B IMM	C6	ii	2	—	—	—	—	Δ	Δ	0	—
			B DIR	D6	dd	3								
			B EXT	F6	hh ll	4								
			B IND, X	E6	ff	4								
			B IND, Y	18 E6	ff	5								
LDD (opr)	Load double accumulator D	M ⇒ A; M+1 ⇒ B	IMM	CC	jj kk	3	—	—	—	—	Δ	Δ	0	—
			DIR	DC	dd	4								
			EXT	FC	hh ll	5								
			IND, X	EC	ff	5								
			IND, Y	18 EC	ff	6								
LDS (opr)	Load stack pointer	M:M+1 ⇒ SP	IMM	8E	jj kk	3	—	—	—	—	Δ	Δ	0	—
			DIR	9E	dd	4								
			EXT	BE	hh ll	5								
			IND, X	AE	ff	5								
			IND, Y	18 AE	ff	6								
LDX (opr)	Load index register X	M:M+1 ⇒ IX	IMM	CE	jj kk	3	—	—	—	—	Δ	Δ	0	—
			DIR	DE	dd	4								
			EXT	FE	hh ll	5								
			IND, X	EE	ff	5								
			IND, Y	CD EE	ff	6								
LDY (opr)	Load index register Y	M:M+1 ⇒ IY	IMM	18 CE	jj kk	4	—	—	—	—	Δ	Δ	0	—
			DIR	18 DE	dd	5								
			EXT	18 FE	hh ll	6								
			IND, X	1A EE	ff	6								
			IND, Y	18 EE	ff	6								
LSL (opr)	Logical shift left		EXT	78	hh ll	6	—	—	—	—	Δ	Δ	Δ	Δ
			IND, X	68	ff	6								
			IND, Y	18 68	ff	7								
LSLA	Logical shift left A		A INH	48	—	2	—	—	—	—	Δ	Δ	Δ	Δ
LSLB	Logical shift Left B		B INH	58	—	2	—	—	—	—	Δ	Δ	Δ	Δ
LSLD	Logical shift left D		INH	05	—	3	—	—	—	—	Δ	Δ	Δ	Δ
LSR (opr)	Logical shift right		EXT	74	hh ll	6	—	—	—	—	0	Δ	Δ	Δ
			IND, X	64	ff	6								
			IND, Y	18 64	ff	7								
LSRA	Logical shift right A		A INH	44	—	2	—	—	—	—	0	Δ	Δ	Δ
LSRB	Logical shift right B		B INH	54	—	2	—	—	—	—	0	Δ	Δ	Δ
LSRD	Logical shift right D		INH	04	—	3	—	—	—	—	0	Δ	Δ	Δ
MUL	Multiply, 8 x 8	A * B ⇒ D	INH	3D	—	10	—	—	—	—	—	—	—	Δ
NEG (opr)	Twos complement memory byte	0 – M ⇒ M	EXT	70	hh ll	6	—	—	—	—	Δ	Δ	Δ	Δ
			IND, X	60	ff	6								
			IND, Y	18 60	ff	7								
NEGA	Twos complement A	0 – A ⇒ A	A INH	40	—	2	—	—	—	—	Δ	Δ	Δ	Δ
NEGB	Twos complement B	0 – B ⇒ B	B INH	50	—	2	—	—	—	—	Δ	Δ	Δ	Δ
NOP	No operation	no operation	INH	01	—	2	—	—	—	—	—	—	—	—
ORAA	OR accumulator A (inclusive)	A + M ⇒ A	A IMM	8A	ii	2	—	—	—	—	Δ	Δ	0	—
			A DIR	9A	dd	3								
			A EXT	BA	hh ll	4								
			A IND, X	AA	ff	4								
			A IND, Y	18 AA	ff	5								
ORAB	OR accumulator B (inclusive)	B + M ⇒ B	B IMM	CA	ii	2	—	—	—	—	Δ	Δ	0	—
			B DIR	DA	dd	3								
			B EXT	FA	hh ll	4								
			B IND, X	EA	ff	4								
			B IND, Y	18 EA	ff	5								
PSHA	Push A onto stack	A ⇒ Stack; SP = SP–1	A INH	36	—	3	—	—	—	—	—	—	—	—
PSHB	Push B onto stack	B ⇒ Stack; SP = SP–1	B INH	37	—	3	—	—	—	—	—	—	—	—
PSHX	Push IX onto stack (low first)	IX ⇒ Stack; SP = SP–2	INH	3C	—	4	—	—	—	—	—	—	—	—
PSHY	Push IY onto stack (low first)	IY ⇒ Stack; SP = SP–2	INH	18 3C	—	5	—	—	—	—	—	—	—	—

Table 3-2 Instruction set (Page 5 of 6)

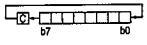
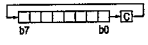
Mnemonic	Operation	Description	Addressing mode	Instruction			Condition codes							
				Opcode	Operand	Cycles	S	X	H	I	N	Z	V	C
PULA	Pull A from stack	SP = SP+1; Stack ⇒ A	A INH	32	—	4	—	—	—	—	—	—	—	—
PULB	Pull B from stack	SP = SP+1; Stack ⇒ B	B INH	33	—	4	—	—	—	—	—	—	—	—
PULX	Pull IX from stack (high first)	SP = SP+2; Stack ⇒ IX	INH	38	—	5	—	—	—	—	—	—	—	—
PULY	Pull IY from stack (high first)	SP = SP+2; Stack ⇒ IY	INH	18 38	—	6	—	—	—	—	—	—	—	—
ROL (opr)	Rotate left		EXT	79	hh ll	6	—	—	—	—	Δ	Δ	Δ	Δ
			IND, X	69	ff	6								
			IND, Y	18 69	ff	7								
ROLA	Rotate left A		A INH	49	—	2	—	—	—	—	Δ	Δ	Δ	Δ
ROLB	Rotate left B		B INH	59	—	2	—	—	—	—	Δ	Δ	Δ	Δ
ROR (opr)	Rotate right		EXT	76	hh ll	6	—	—	—	—	Δ	Δ	Δ	Δ
			IND, X	66	ff	6								
			IND, Y	18 66	ff	7								
RORA	Rotate right A		A INH	46	—	2	—	—	—	—	Δ	Δ	Δ	Δ
RORB	Rotate right B		B INH	56	—	2	—	—	—	—	Δ	Δ	Δ	Δ
RTI	Return from interrupt	see Figure 3-2	INH	3B	—	12	Δ	↓	Δ	Δ	Δ	Δ	Δ	Δ
RTS	Return from subroutine	see Figure 3-2	INH	39	—	5	—	—	—	—	—	—	—	—
SBA	Subtract B from A	A – B ⇒ A	INH	10	—	2	—	—	—	—	Δ	Δ	Δ	Δ
SBCA (opr)	Subtract with carry from A	A – M – C ⇒ A	A IMM	82	ii	2	—	—	—	—	Δ	Δ	Δ	Δ
			A DIR	92	dd	3								
			A EXT	B2	hh ll	4								
			A IND, X	A2	ff	4								
			A IND, Y	18 A2	ff	5								
SBCB (opr)	Subtract with carry from B	B – M – C ⇒ B	B IMM	C2	ii	2	—	—	—	—	Δ	Δ	Δ	Δ
			B DIR	D2	dd	3								
			B EXT	F2	hh ll	4								
			B IND, X	E2	ff	4								
			B IND, Y	18 E2	ff	5								
SEC	Set carry	1 ⇒ C	INH	0D	—	2	—	—	—	—	—	—	—	1
SEI	Set interrupt mask	1 ⇒ I	INH	0F	—	2	—	—	—	1	—	—	—	—
SEV	Set overflow flag	1 ⇒ V	INH	0B	—	2	—	—	—	—	—	—	1	—
STAA (opr)	Store accumulator A	A ⇒ M	A DIR	97	dd	3	—	—	—	—	Δ	Δ	0	—
			A EXT	B7	hh ll	4								
			A IND, X	A7	ff	4								
			A IND, Y	18 A7	ff	5								
STAB (opr)	Store accumulator B	B ⇒ M	B DIR	D7	dd	3	—	—	—	—	Δ	Δ	0	—
			B EXT	F7	hh ll	4								
			B IND, X	E7	ff	4								
			B IND, Y	18 E7	ff	5								
STD (opr)	Store accumulator D	A ⇒ M; B ⇒ M+1	DIR	DD	dd	4	—	—	—	—	Δ	Δ	0	—
			EXT	FD	hh ll	5								
			IND, X	ED	ff	5								
			IND, Y	18 ED	ff	6								
STOP	Stop internal clocks	—	INH	CF	—	2	—	—	—	—	—	—	—	—
STS (opr)	Store stack pointer	SP ⇒ M:M+1	DIR	9F	dd	4	—	—	—	—	Δ	Δ	0	—
			EXT	BF	hh ll	5								
			IND, X	AF	ff	5								
			IND, Y	18 AF	ff	6								
STX (opr)	Store index register X	IX ⇒ M:M+1	DIR	DF	dd	4	—	—	—	—	Δ	Δ	0	—
			EXT	FF	hh ll	5								
			IND, X	EF	ff	5								
			IND, Y	CD EF	ff	6								
STY (opr)	Store index register Y	IY ⇒ M:M+1	DIR	18 DF	dd	5	—	—	—	—	Δ	Δ	0	—
			EXT	18 FF	hh ll	6								
			IND, X	1A EF	ff	6								
			IND, Y	18 EF	ff	6								

Table 3-2 Instruction set (Page 6 of 6)

Mnemonic	Operation	Description	Addressing mode	Instruction			Condition codes							
				Opcode	Operand	Cycles	S	X	H	I	N	Z	V	C
SUBA (opr)	Subtract memory from A	A – M ⇒ A	A IMM A DIR A EXT A IND, X A IND, Y	80 90 B0 A0 18 A0	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	Δ	Δ
SUBB (opr)	Subtract memory from B	B – M ⇒ B	B IMM B DIR B EXT B IND, X B IND, Y	C0 D0 F0 E0 18 E0	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	Δ	Δ
SUBD (opr)	Subtract memory from D	D – M:M+1 ⇒ D	IMM DIR EXT IND, X IND, Y	83 93 B3 A3 18 A3	jj kk dd hh ll ff ff	4 5 6 6 7	—	—	—	—	Δ	Δ	Δ	Δ
SWI	Software interrupt	see Figure 3-2	INH	3F	—	14	—	—	—	1	—	—	—	—
TAB	Transfer A to B	A ⇒ B	INH	16	—	2	—	—	—	—	Δ	Δ	0	—
TAP	Transfer A to CC register	A ⇒ CCR	INH	06	—	2	Δ	↓	Δ	Δ	Δ	Δ	Δ	Δ
TBA	Transfer B to A	B ⇒ A	INH	17	—	2	—	—	—	—	Δ	Δ	0	—
TEST	Test (only in test modes)	address bus increments	INH	00	—	†	—	—	—	—	—	—	—	—
TPA	Transfer CC register to A	CCR ⇒ A	INH	07	—	2	—	—	—	—	—	—	—	—
TST (opr)	Test for zero or minus	M – 0	EXT IND, X IND, Y	7D 6D 18 6D	hh ll ff ff	6 6 7	—	—	—	—	Δ	Δ	0	0
TSTA	Test A for zero or minus	A – 0	A INH	4D	—	2	—	—	—	—	Δ	Δ	0	0
TSTB	Test B for zero or minus	B – 0	B INH	5D	—	2	—	—	—	—	Δ	Δ	0	0
TSX	Transfer stack pointer to X	SP + 1 ⇒ IX	INH	30	—	3	—	—	—	—	—	—	—	—
TSY	Transfer stack pointer to Y	SP + 1 ⇒ IY	INH	18 30	—	4	—	—	—	—	—	—	—	—
TXS	Transfer X to stack pointer	IX – 1 ⇒ SP	INH	35	—	3	—	—	—	—	—	—	—	—
TYS	Transfer Y to stack pointer	IY – 1 ⇒ SP	INH	18 35	—	4	—	—	—	—	—	—	—	—
WAI	Wait for interrupt	stack registers & WAIT	INH	3E	—	‡	—	—	—	—	—	—	—	—
XGDX	Exchange D with X	IX ⇒ D; D ⇒ IX	INH	8F	—	3	—	—	—	—	—	—	—	—
XGDY	Exchange D with Y	IY ⇒ D; D ⇒ IY	INH	18 8F	—	4	—	—	—	—	—	—	—	—

Operators

- ⇒ Is transferred to
- • Boolean AND
- + Arithmetic addition, except where used as an inclusive-OR symbol in Boolean formulae
- ⊕ Exclusive-OR
- * Multiply
- : Concatenation
- – Arithmetic subtraction, or negation symbol (Twos complement)

Operands

- dd 8-bit direct address ($0000–$00FF); the high byte is assumed to be zero
- ff 8-bit positive offset ($00 to $FF (0 to 256)) is added to the contents of the index register
- hh High order byte of 16-bit extended address
- ii One byte of immediate data
- jj High order byte of 16-bit immediate data
- kk Low order byte of 16-bit immediate data
- ll Low order byte of 16-bit extended address
- mm 8-bit mask (set bits to be affected)
- rr Signed relative offset ($80 to $7F (–128 to +127)); offset is relative to the address following the offset byte

Cycles

- † Infinite, or until reset occurs
- ‡ 12 cycles are used, beginning with the opcode fetch. A wait state is entered, which remains in effect for an integer number of MPU E clock cycles (n) until an interrupt is recognised. Finally, two additional cycles are used to fetch the appropriate interrupt vector. (14 + n, total).

Condition Codes

- — Bit not changed
- 0 Bit always cleared
- 1 Bit always set
- Δ Bit set or cleared, depending on the operation
- ↓ Bit can be cleared, but cannot become set
- ? Not defined

4

OPERATING MODES AND ON-CHIP MEMORY

This section contains information about the modes that define MC68HC11KG4 operating conditions, and about the on-chip memory that allows the MCU to be configured for various applications.

4.1 Operating modes

The values of the mode select inputs MODB and MODA during reset determine the operating mode (See Table 4-4). Single chip and expanded modes are the normal modes. In single chip mode only on-board memory is available. Expanded mode, however, allows access to external memory. Each of these two normal modes is paired with a special mode. Bootstrap, a variation of the single chip mode, is a special mode that executes a bootloader program in an internal bootstrap ROM. Test is a special mode that allows privileged access to internal resources.

4.1.1 Single chip operating mode

In single chip operating mode, the MC68HC11KG4 microcontroller has no external address or data bus. Ports B, C and F are available for general purpose parallel I/O.

4.1.2 Expanded operating mode

In expanded operating mode, the MCU can access a 64K byte physical address space. The address space includes the same on-chip memory addresses used for single chip mode, in addition to external memory and peripheral devices.

The expansion bus is made up of ports B, C, and F, and the R/$\overline{W}$ signal. In expanded mode, high order address bits are output on the port B pins, low order address bits on the port F pins, and the data bus on port C. The R/$\overline{W}$ pin signals the direction of data transfer on the port C bus.

To allow access to slow peripherals, off chip accesses can be extended by one E clock cycle, under control of the STRCH bit in the OPT2 register. The E clock stretches externally, but the internal clocks are not affected so that timers and serial systems are not corrupted. See Section 4.3.2.5.

A security feature can protect EEPROM data when in expanded mode; see Section 4.4.3.

4.1.3 Special test mode

4

Special test, a variation of the expanded mode, is used during Motorola's internal production testing, and is not intended or recommended for any other purpose. Its specification is subject to change without notice.

4.1.4 Special bootstrap mode

When the MCU is reset in special bootstrap mode, a small on-chip ROM is enabled at address $BE40–$BFFF. The ROM contains a reset vector and a bootloader program. The MCU fetches the reset vector, then executes the bootloader.

For normal use of the bootloader program, send a synchronization byte $FF to the SCI receiver at either E clock ÷256, or E clock ÷1664 (7812 or 1200 baud respectively, for an E clock of 2MHz). Then download up to 2048 bytes of program data (which is put into RAM starting at $0080). These characters are echoed through the transmitter. The bootloader program ends the download after a timeout of four character times or 2048 bytes. When loading is complete, the program jumps to location $0080 and begins executing the code. Use of an external pull-up resistor is required when using the SCI transmitter pin (TXD) because port D pins are configured for wired-OR operation by the bootloader. In bootstrap mode, the interrupt vectors point to RAM. This allows the use of interrupts through a jump table.

Further baud rate options are available on the MC68HC11KG4 by using a different value for the synchronization byte, as shown in Table 4-1.

Refer also to Motorola application note ***AN1060, M68HC11 Bootstrap Mode*** (the bootloader mode is similar to that used on the MC68HC11K4).

Table 4-1 Example bootloader baud rates

Sync. byte	Timeout delay	Baud rates for an E clock of:					
		2.00MHz	2.10MHz	3.00MHz	3.15MHz	4.00MHz	4.20MHz
$FF	4 char.	7812	8192	11718	12288	15624	16 406
$FF	4	1200	1260	1800	1890	2400	2524
$F0	4.9	9600	10080	14400	15120	19200	20192
$FD	17.3	5208	5461	7812	8192	10416	10 336
$FD	13	3906	4096	5859	6144	7812	8203

4.2 On-chip memory

The MC68HC11KG4 MCU includes 768 bytes of on-chip RAM, 24K bytes of ROM and 640 bytes of EEPROM. The bootloader ROM occupies a 448 byte block of the memory map. The CONFIG register is implemented as a separate EEPROM byte.

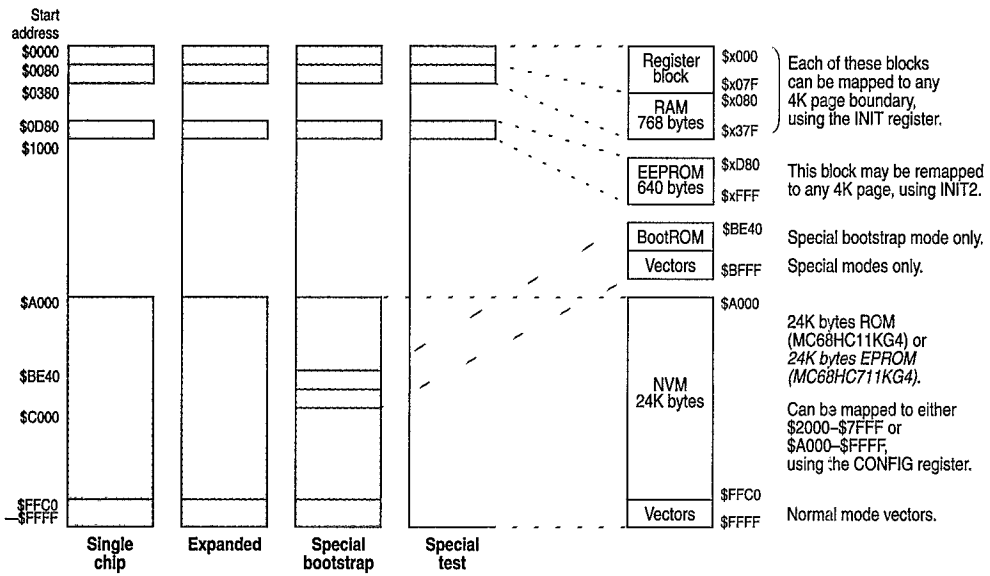


Figure 4-1 MC68HC11KG4 memory map

4.2.1 Mapping allocations

Memory locations for on-chip resources are the same for both expanded and single chip modes. The 128-byte register block originates at $0000 on reset and can be placed at any other 4K boundary ($x000) after reset by writing an appropriate value to the INIT register. Refer to Figure 4-1, which shows the memory map.

The on-board 768 byte RAM is initially located at $0080 after reset. The RAM is divided into two sections, of 128 bytes and 640 bytes. If RAM and registers are both mapped to the same 4K boundary, RAM starts at $x080 and 128 bytes are remapped at $x300–$x37F. Otherwise, RAM starts at $x000. See Figure 4-2.

Remapping is accomplished by writing appropriate values into the two nibbles of the INIT register. See Section 4.3.2.2.

The 640-byte EEPROM is initially located at $0D80 after reset, when EEPROM is enabled in the memory map by the CONFIG register. EEPROM can be placed in any other 4K page ($xD80) by writing to the INIT2 register.

The ROMAD and ROMON bits in the CONFIG register control the position and presence of ROM in the memory map. In special test mode, the ROMON bit is cleared so the ROM is removed from the memory map. In single chip mode, the ROMAD bit is set to one after reset, which enables the ROM at $A000–$FFFF. In expanded mode, the ROM may be enabled from $2000–$7FFF (ROMAD = 0) to allow an external memory to contain the interrupt vectors and initialization code.

In special bootstrap mode, a bootloader ROM is enabled at locations $BE40–$BFFF. The vectors for special bootstrap mode are contained in the bootloader program.

4.2.1.1 RAM

The MC68HC11KG4 has 768 bytes of fully static RAM that are used for storing instructions, variables and temporary data during program execution. RAM can be placed at any 4K boundary in the 64K byte address space by writing an appropriate value to the INIT register.

By default, RAM is initially located at $0080 in the memory map. Direct addressing mode can access the first 128 locations of RAM using a one-byte address operand. Direct mode accesses save program memory space and execution time. Registers can be moved to other boundaries to allow 256 bytes of RAM to be located in direct addressing space. See Figure 4-2.

The on-chip RAM is a fully static memory. RAM contents can be preserved during periods of processor inactivity by either of two methods, both of which reduce power consumption:

1) During the software-based STOP mode, MCU clocks are stopped, but the MCU continues to draw power from V_{DD}. Power supply current is directly related to operating frequency in CMOS integrated circuits and there is very little leakage when the clocks are stopped. These two factors reduce power consumption while the MCU is in STOP mode.
2) To reduce power consumption to a minimum, V_{DD} can be turned off, and the MODB/VSTBY pin can be used to supply RAM power from either a battery back-up or a second power supply. Although this method requires external hardware, it is very effective. Refer to Section 2 for information about how to connect the stand-by RAM power supply and to Section 5 for a description of low power operation.

4.2.1.2 ROM

The MCU has 24K bytes of ROM. The ROM array is enabled when the ROMON bit in the CONFIG register is set to one (erased). The ROMAD bit in CONFIG places the ROM at either $A000–$FFFF (ROMAD = 1) or at $2000–$7FFF (ROMAD = 0) when coming out of reset in expanded mode.

4.2.1.3 Bootloader ROM

The bootloader ROM is enabled at address $BE40–$BFFF during special bootstrap mode. The reset vector is fetched from this ROM and the MCU executes the bootloader firmware. In normal modes, the bootloader ROM is disabled.

4.2.2 Registers

In Table 4-2, a summary of registers and control bits, the registers are shown in ascending order within the 128-byte register block. The addresses shown are for default block mapping ($0000–$007F), however, the INIT register remaps the block to any 4K page ($x000–$x07F). See Section 4.3.2.2.

Table 4-2 Register and control bit assignments (Page 1 of 4)

Register name	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Port A data (PORTA)	$0000	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	undefined
Data direction A (DDRA)	$0001	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0	0000 0000
Data direction B (DDRB)	$0002	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	0000 0000
Data direction F (DDRF)	$0003	DDF7	DDF6	DDF5	DDF4	DDF3	DDF2	DDF1	DDF0	0000 0000
Port B data (PORTB)	$0004	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	undefined
Port F data (PORTF)	$0005	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0	undefined
Port C data (PORTC)	$0006	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	undefined
Data direction C (DDRC)	$0007	DDC7	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	0000 0000
Port D data (PORTD)	$0008	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0	undefined
Data direction D (DDRD)	$0009	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	0000 0000
Port E data (PORTE)	$000A	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0	undefined
Timer compare force (CFORC)	$000B	FOC1	FOC2	FOC3	FOC4	FOC5	0	0	0	0000 0000
Output compare 1 mask (OC1M)	$000C	OC1M7	OC1M6	OC1M5	OC1M4	OC1M3	0	0	0	0000 0000
Output compare 1 data (OC1D)	$000D	OC1D7	OC1D6	OC1D5	OC1D4	OC1D3	0	0	0	0000 0000
Timer count (TCNT) high	$000E	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	0000 0000
Timer count (TCNT) low	$000F	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	0000 0000
Timer input capture 1 (TIC1) high	$0010	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	undefined
Timer input capture 1 (TIC1) low	$0011	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	undefined
Timer input capture 2 (TIC2) high	$0012	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	undefined
Timer input capture 2 (TIC2) low	$0013	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	undefined
Timer input capture 3 (TIC3) high	$0014	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	undefined
Timer input capture 3 (TIC3) low	$0015	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	undefined
Timer output compare 1 (TOC1) high	$0016	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	1111 1111
Timer output compare 1 (TOC1) low	$0017	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111
Timer output compare 2 (TOC2) high	$0018	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	1111 1111
Timer output compare 2 (TOC2) low	$0019	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111
Timer output compare 3 (TOC3) high	$001A	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	1111 1111
Timer output compare 3 (TOC3) low	$001B	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111
Timer output compare 4 (TOC4) high	$001C	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	1111 1111
Timer output compare 4 (TOC4) low	$001D	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111
Capture 4/compare 5 (TI4/O5) high	$001E	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	1111 1111
Capture 4/compare 5 (TI4/O5) low	$001F	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111
Timer control 1 (TCTL1)	$0020	OM2	OL2	OM3	OL3	OM4	OL4	OM5	OL5	0000 0000
Timer control 2 (TCTL2)	$0021	EDG4B	EDG4A	EDG1B	EDG1A	EDG2B	EDG2A	EDG3B	EDG3A	0000 0000
Timer interrupt mask 1 (TMSK1)	$0022	OC1I	OC2I	OC3I	OC4I	I4/O5I	IC1I	IC2I	IC3I	0000 0000

Table 4-2 Register and control bit assignments (Page 2 of 4)

Register name	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer interrupt flag 1 (TFLG1)	$0023	OC1F	OC2F	OC3F	OC4F	I4/O5F	IC1F	IC2F	IC3F	0000 0000
Timer interrupt mask 2 (TMSK2)	$0024	TOI	RTII	PAOVI	PAII	0	0	PR1	PR0	0000 0000
Timer interrupt flag 2 (TFLG2)	$0025	TOF	RTIF	PAOVF	PAIF	0	0	0	0	0000 0000
Pulse accumulator control (PACTL)	$0026	0	PAEN	PAMOD	PEDGE	0	I4/O5	RTR1	RTR0	0000 0000
Pulse accumulator count (PACNT)	$0027	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	undefined
SPI control (SPCR)	$0028	SPIE	SPE	DWOM	MSTR	CPOL	CPHA	SPR1	SPR0	0000 01uu
SPI status (SPSR)	$0029	SPIF	WCOL	0	MODF	0	0	0	0	0000 0000
SPI data (SPDR)	$002A	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	undefined
Reserved	$002B									
Port pull-up assignment (PPAR)	$002C	0	0	0	0	JPPUE	0	FPPUE	BPPUE	0000 1011
Reserved	$002D									
PLL control (PLLCR)	$002E	PLLON	BCS	AUTO	BWC	VCOT	MCS	WSLOW	WEN	1011 1000
Synthesizer program (SYNR)	$002F	SYNX1	SYNX0	SYNY5	SYNY4	SYNY3	SYNY2	SYNY1	SYNY0	0000 0001
A/D control & status (ADCTL)	$0030	CCF	CONV8	SCAN	MULT	CD	CC	CB	CA	0000 0000
Reserved	$0031									
A/D frequency select (ADFRQ)	$0032	0	0	0	0	0	0	0	ADER	0000 0000
SRTI programming (RTREG)	$0033	SRTI3	SRTI2	SRTI1	SRTI0	0	0	0	0	1111 0000
SRTI control (RTCTL)	$0034	RTF	RTI	0	0	RTCK2	RTCK1	RTHF	RTLF	0000 0000
Block protect (BPROT)	$0035	BULKP	LVPEN	BPRT4	PTCON	BPRT3	BPRT2	BPRT1	BPRT0	1111 1111
Reserved	$0036									
EEPROM mapping (INIT2)	$0037	EE3	EE2	EE1	EE0	0	0	0	0	0000 0000
System config. options 2 (OPT2)	$0038	LIRDV	CWOM	STRCH	IRVNE	LSBF	SPR2	XDV1	XDV0	000x 0000
System config. options 1 (OPTION)	$0039	ADPU	CSEL	IRQE	DLY	CME	FCME	CR1	CR0	0001 0000
COP timer arm/reset (COPRST)	$003A	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	undefined
EEPROM programming (PPROG)	$003B	ODD	EVEN	LVPI	BYTE	ROW	ERASE	EELAT	EEPGM	0000 0000
Highest priority interrupt (HPRIO)	$003C	RBOOT	SMOD	MDA	PSEL4	PSEL3	PSEL2	PSEL1	PSEL0	xxx0 0110
RAM & I/O mapping (INIT)	$003D	RAM3	RAM2	RAM1	RAM0	REG3	REG2	REG1	REG0	0000 0000
Factory test (TEST1)	$003E	TILOP	PLTST	OCCR	CBYP	DISR	FCM	FCOP	DLVR	0000 x00x
Configuration control (CONFIG)	$003F	ROMAD	NWCOP	CLKX	PAREN	NOSEC	NOCOP	ROMON	EEON	xxxx xxxx
A/D result 1 (ADR1) high	$0040	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	undefined
A/D result 1 (ADR1) low	$0041	(bit 7)	(6)	0	0	0	0	0	0	uu00 0000
A/D result 2 (ADR2) high	$0042	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	undefined
A/D result 2 (ADR2) low	$0043	(bit 7)	(6)	0	0	0	0	0	0	uu00 0000
A/D result 3 (ADR3) high	$0044	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	undefined
A/D result 3 (ADR3) low	$0045	(bit 7)	(6)	0	0	0	0	0	0	uu00 0000

Table 4-2 Register and control bit assignments (Page 3 of 4)

Register name	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
A/D result 4 (ADR4) high	$0046	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	undefined
A/D result 4 (ADR4) low	$0047	(bit 7)	(6)	0	0	0	0	0	0	uu00 0000
A/D result 5 (ADR5) high	$0048	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	undefined
A/D result 5 (ADR5) low	$0049	(bit 7)	(6)	0	0	0	0	0	0	uu00 0000
A/D result 6 (ADR6) high	$004A	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	undefined
A/D result 6 (ADR6) low	$004B	(bit 7)	(6)	0	0	0	0	0	0	uu00 0000
A/D result 7 (ADR7) high	$004C	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	undefined
A/D result 7 (ADR7) low	$004D	(bit 7)	(6)	0	0	0	0	0	0	uu00 0000
A/D result 8 (ADR8) high	$004E	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	undefined
A/D result 8 (ADR8) low	$004F	(bit 7)	(6)	0	0	0	0	0	0	uu00 0000
SCI1 baud rate control high (SC1BDH)	$0050	BTST	BSPL	SYNC	SBR12	SBR11	SBR10	SBR9	SBR8	0000 0000
SCI1 baud rate control low (SC1BDL)	$0051	SBR7	SBR6	SBR5	SBR4	SBR3	SBR2	SBR1	SBR0	0000 0100
SCI1 control 1 (SC1CR1)	$0052	LOOPS	WOMS	0	M	WAKE	ILT	PE	PT	0000 0000
SCI1 control 2 (SC1CR2)	$0053	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK	0000 0000
SCI1 status 1 (SC1SR1)	$0054	TDRE	TC	RDRF	IDLE	OR	NF	FE	PF	1100 0000
SCI1 status 2 (SC1SR2)	$0055	0	0	0	0	0	0	0	RAF	0000 0000
SCI1 data high (SC1DRH)	$0056	R8	T8	0	0	0	0	0	0	undefined
SCI1 data low (SC1DRL)	$0057	R7T7	R6T6	R5T5	R4T4	R3T3	R2T2	R1T1	R0T0	undefined
Port J keyboard interrupt flag (JKFLAG)	$0058	0	0	0	0	KFLG3	KFLG2	KFLG1	KFLG0	0000 0000
Port J keyboard interrupt mask (JKMASK)	$0059	KEDG3	KEDG2	KEDG1	KEDG0	KMSK3	KMSK2	KMSK1	KMSK0	0000 0000
Port J data (PORTJ)	$005A	0	0	0	0	PJ3	PJ2	PJ1	PJ0	undefined
Data direction J (DDRJ)	$005B	0	0	0	0	DDJ3	DDJ2	DDJ1	DDJ0	0000 0000
Port H data (PORTH)	$005C	PH7	PH6	PH5	PH4	PH3	PH2	PH1	PH0	0000 0000
Port H mismatch (PTHMS)	$005D	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	0000 0000
Port K data (PORTK)	$005E	PK7	PK6	PK5	PK4	PK3	PK2	PK1	PK0	0000 0000
Port K mismatch (PTKMS)	$005F	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	0000 0000
PWM channel duty 0 (PWDTY0) high	$0060	0	0	0	0	0	0	MSB	(bit 8)	0000 0011
PWM channel duty 0 (PWDTY0) low	$0061	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111
PWM channel duty 1 (PWDTY1) high	$0062	0	0	0	0	0	0	MSB	(bit 8)	0000 0011
PWM channel duty 1 (PWDTY1) low	$0063	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111
PWM channel duty 2 (PWDTY2) high	$0064	0	0	0	0	0	0	MSB	(bit 8)	0000 0011
PWM channel duty 2 (PWDTY2) low	$0065	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111
PWM channel duty 3 (PWDTY3) high	$0066	0	0	0	0	0	0	MSB	(bit 8)	0000 0011
PWM channel duty 3 (PWDTY3) low	$0067	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111
PWM channel duty 4 (PWDTY4) high	$0068	0	0	0	0	0	0	MSB	(bit 8)	0000 0011

Table 4-2 Register and control bit assignments (Page 4 of 4)

Register name	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
PWM channel duty 4 (PWDTY4) low	$0069	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111
PWM channel duty 5 (PWDTY5) high	$006A	0	0	0	0	0	0	MSB	(bit 8)	0000 0011
PWM channel duty 5 (PWDTY5) low	$006B	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111
PWM channel duty 6 (PWDTY6) high	$006C	0	0	0	0	0	0	MSB	(bit 8)	0000 0011
PWM channel duty 6 (PWDTY6) low	$006D	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111
PWM channel duty 7 (PWDTY7) high	$006E	0	0	0	0	0	0	MSB	(bit 8)	0000 0011
PWM channel duty 7 (PWDTY7) low	$006F	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111
SCI2 baud rate high (SC2BDH)	$0070	BTST	BSPL	SYNC	SBR12	SBR11	SBR10	SBR9	SBR8	0000 0000
SCI2 baud rate low (SC2BDL)	$0071	SBR7	SBR6	SBR5	SBR4	SBR3	SBR2	SBR1	SBR0	0000 0100
SCI2 control 1 (SC2CR1)	$0072	LOOPS	WOMS	0	M	WAKE	ILT	PE	PT	0000 0000
SCI2 control 2 (SC2CR2)	$0073	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK	0000 0000
SCI2 status 1 (SC2SR1)	$0074	TDRE	TC	RDRF	IDLE	OR	NF	FE	PF	1100 0000
SCI2 status 2 (SC2SR2)	$0075	0	0	0	0	0	0	0	RAF	0000 0000
SCI2 data high (SC2DRH)	$0076	R8	T8	0	0	0	0	0	0	undefined
SCI2 data low (SC2DRL)	$0077	R7T7	R6T6	R5T5	R4T4	R3T3	R2T2	R1T1	R0T0	undefined
PWM control (PWCLK)	$0078	0	0	PCKC2	PCKC1	0	0	PWMF	PWMI	0000 0000
PWM channel enable (PWEN)	$0079	PWEN7	PWEN6	PWEN5	PWEN4	PWEN3	PWEN2	PWEN1	PWEN0	0000 0000
PWM channel polarity (PWPOL)	$007A	PPOL7	PPOL6	PPOL5	PPOL4	PPOL3	PPOL2	PPOL1	PPOL0	0000 0000
PWM channel counter high (PWCNT)	$007B	0	0	0	0	0	0	0	(bit 8)	0000 0000
PWM channel counter low (PWCNT)	$007C	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	0000 0000
D/A control (DACON)	$007D	0	0	0	0	0	0	0	DAEI	0000 0000
D/A data (DAI)	$007E	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	0000 0000
Reserved	$007F									

KEY

‡ Applies only to EPROM devices

x State on reset depends on mode selected

u State of bit on reset is undefined

4.3 System initialization

Registers and bits that control initialization and the basic operation of the MCU are protected against writes except under special circumstances. The following table lists registers that can be written only once after reset, or that must be written within the first 64 cycles after reset.

Table 4-3 Registers with limited write access

Operating Modes	Register address	Register name	Must be written in first 64 cycles	Write anytime
SMOD = 0	$x024	Timer interrupt mask register 2 (TMSK2)	Bits [3:0], once only	Bits [7:4]
	$x02F	Synthesizer program (SYNR)	All bits read-only	All bits read-only
	$x033	SRTI programming (RTREG)	No	All bits, once only
	$x034	SRTI control (RTCTL)	No	All bits (bits [3,2] once only)
	$x035	Block protect register (BPROT)	Clear bits once only	Set bits
	$x037	EEPROM mapping register (INIT2)	No	All bits, once only
	$x038	System configuration options register 2 (OPT2)	No	All bits (bit 4 once only)
	$x039	System configuration options register (OPTION)	Bits 5, 4, 2, 1, 0 once only	Bits 7, 6, 3
	$x03D	RAM and I/O map register (INIT)	All bits, once only	—
	$x059	Port J keyboard interrupt mask register (JKMASK)	No	All bits (bits [7:4] once only)
SMOD = 1	$x024	Timer interrupt mask register 2 (TMSK2)	—	All bits
	$x02F	Synthesizer program (SYNR)	—	All bits
	$x033	SRTI programming (RTREG)	—	All bits
	$x034	SRTI control (RTCTL)	—	All bits
	$x035	Block protect register (BPROT)	—	All bits
	$x037	EEPROM mapping register (INIT2)	—	All bits
	$x038	System configuration options register 2 (OPT2)	—	All bits
	$x039	System configuration options register (OPTION)	—	All bits
	$x03D	RAM and I/O map register (INIT)	—	All bits
	$x059	Port J keyboard interrupt mask register (JKMASK)	—	All bits

4.3.1 Mode selection

The four mode variations are selected by the logic states of the mode A (MODA) and mode B (MODB) pins during reset. The MODA and MODB logic levels determine the logic state of the special mode (SMOD) and mode A (MDA) control bits in the highest priority I-bit interrupt and miscellaneous (HPRIO) register.

After reset is released, the mode select pins no longer influence the MCU operating mode. In single chip operating mode, MODA pin is connected to a logic zero. In expanded mode, MODA is normally connected to V_{DD} through a pull-up resistor of 4.7 kΩ. The MODA pin also functions as the load instruction register ($\overline{LIR}$) pin when the MCU is not in reset. The open-drain active low $\overline{LIR}$ output pin drives low during the first E cycle of each instruction. The MODB pin also functions as

the stand-by power input (VSTBY), which allows the RAM contents to be maintained in the absence of V_{DD}.

Refer to Table 4-4, which is a summary of mode pin operation, the mode control bits and the four operating modes.

A normal mode is selected when MODB is logic one during reset. One of three reset vectors is fetched from address $FFFA–$FFFF, and program execution begins from the address indicated by this vector. If MODB is logic zero during reset, the special mode reset vector is fetched from addresses $BFFA–$BFFF and software has access to special test features. Refer to Section 5.

4

4.3.1.1 HPRIO — Highest priority I-bit interrupt & misc. register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Highest priority interrupt (HPRIO)	$003C	RBOOT	SMOD	MDA	PSEL4	PSEL3	PSEL2	PSEL1	PSEL0	xxx0 0110

Note: RBOOT, SMOD and MDA bits depend on the power-up initialization mode and can only be written in special modes when SMOD = 1. Refer to Table 4-4.

RBOOT — Read bootstrap ROM

1 (set) – Bootloader ROM enabled, at $BE40–$BFFF.

0 (clear) – Bootloader ROM disabled and not in map.

SMOD — Special mode select

1 (set) – Special mode variation in effect.

0 (clear) – Normal mode variation in effect.

Once cleared, cannot be set again.

MDA — Mode select A

1 (set) – Normal expanded or special test mode. (Expanded buses active.)

0 (clear) – Normal single chip or special bootstrap mode. (Ports active.)

PSEL[4:0] — Priority select bits (refer to Section 5)

4.3.2 Initialization

Because bits in the following registers control the basic configuration of the MCU, an accidental change of their values could cause serious system problems. The protection mechanism,

Table 4-4 Hardware mode select summary

Inputs		Mode	Control bits in HPRIO (latched at reset)		
MODB	MODA		RBOOT	SMOD	MDA
1	0	Single chip	0	0	0
1	1	Expanded	0	0	1
0	0	Special bootstrap	1	1	0
0	1	Special test	0	1	1

overridden in special operating modes, requires a write to the protected bits only within the first 64 bus cycles after any reset, or only once after each reset. See Table 4-3.

4.3.2.1 CONFIG — System configuration register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Configuration control (CONFIG)	$003F	ROMAD	NWCOP	CLKX	PAREN	NOSEC	NOCOP	ROMON	EEON	xxxx xxxx

CONFIG controls the presence and/or location of ROM and EEPROM in the memory map and enables the COP and WCOP watchdog systems. The CLKX bit is used to enable an output on the XOUT pin and the PAREN bit enables pull-ups on certain ports. A security feature that protects data in EEPROM and RAM is available, controlled by the NOSEC bit. Refer to Section 4.4.3.

CONFIG is made up of EEPROM cells and static working latches. The operation of the MCU is controlled directly by these latches and not the EEPROM byte. When programming the CONFIG register, the EEPROM byte is accessed. When the CONFIG register is read, the static latches are accessed.

These bits can be read at any time. The value read is the one latched into the register from the EEPROM cells during the last reset sequence. A new value programmed into this register is not readable until after a subsequent reset sequence.

Bits in CONFIG can be written at any time if SMOD = 1 (bootstrap or special test mode). If SMOD = 0 (single chip or expanded mode), they can only be written using the EEPROM programming sequence, and are neither readable nor active until latched via the next reset.

ROMAD — ROM mapping control

1 (set) – ROM addressed from $A000 to $FFFF.

0 (clear) – ROM addressed from $2000 to $7FFF (expanded mode only).

In single chip mode, reset sets this bit.

NWCOP — WCOP system disable (refer to Section 5)

1 (set) – WCOP system disabled.

0 (clear) – WCOP system enabled (forces reset on timeout).

CLKX — XOUT enable

1 (set) – XCLK signal driven out on the XOUT pin.

0 (clear) – XOUT pin disabled.

The frequency of the XCLK signal is controlled by two bits in the OPT2 register (see Section 4.3.2.5).

PAREN — Pull-up assignment register enable (refer to Section 6)

1 (set) – Pull-ups can be enabled using PPAR.

0 (clear) – All pull-ups disabled (not controlled by PPAR).

NOSEC — EEPROM security disabled (see Section 4.4.3)

1 (set) – Disable security.

0 (clear) – Enable security.

NOCOP — COP system disable (refer to Section 5)

1 (set) – COP system disabled.

0 (clear) – COP system enabled (forces reset on timeout).

ROMON — ROM enable

1 (set) – ROM included in the memory map.

0 (clear) – ROM excluded from the memory map.

In single chip mode, reset sets this bit. In special test mode, reset clears ROMON.

EEON — EEPROM enable

1 (set) – EEPROM included in the memory map.

0 (clear) – EEPROM excluded from the memory map.

4.3.2.2 INIT — RAM and I/O mapping register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
RAM & I/O mapping (INIT)	$003D	RAM3	RAM2	RAM1	RAM0	REG3	REG2	REG1	REG0	0000 0000

The internal registers used to control the operation of the MCU can be relocated on 4K boundaries within the memory space with the use of INIT. This 8-bit special-purpose register can change the default locations of the RAM and control registers within the MCU memory map. It can be written to only once within the first 64 E clock cycles after a reset. It then becomes a read-only register.

RAM[3:0] — RAM map position

These four bits, which specify the upper hexadecimal digit of the RAM address, control the position of the RAM in the memory map. The RAM can be positioned at the beginning of any 4K page in the memory map. Refer to Table 4-5.

REG[3:0] — 128-byte register block position

These four bits specify the upper hexadecimal digit of the address for the 128-byte block of internal registers. The register block is positioned at the beginning of any 4K page in the memory map. Refer to Table 4-5.

Table 4-5 RAM and register remapping

RAM[3:0]	Location
0000	$0000–$02FF
0001	$1000–$12FF
0010	$2000–$22FF
0011	$3000–$32FF
0100	$4000–$42FF
0101	$5000–$52FF
0110	$6000–$62FF
0111	$7000–$72FF
1000	$8000–$82FF
1001	$9000–$92FF
1010	$A000–$A2FF
1011	$B000–$B2FF
1100	$C000–$C2FF
1101	$D000–$D2FF
1110	$E000–$E2FF
1111	$F000–$F2FF

REG[3:0]	Location
0000	$0000–$007F
0001	$1000–$107F
0010	$2000–$207F
0011	$3000–$307F
0100	$4000–$407F
0101	$5000–$507F
0110	$6000–$607F
0111	$7000–$707F
1000	$8000–$807F
1001	$9000–$907F
1010	$A000–$A07F
1011	$B000–$B07F
1100	$C000–$C07F
1101	$D000–$D07F
1110	$E000–$E07F
1111	$F000–$F07F

When the memory map has the 128-byte register block mapped at the same location as RAM, the registers have priority and the RAM is relocated to the memory space immediately following the register block. This mapping feature keeps all the RAM available for use. Refer to Figure 4-2, which illustrates the overlap.

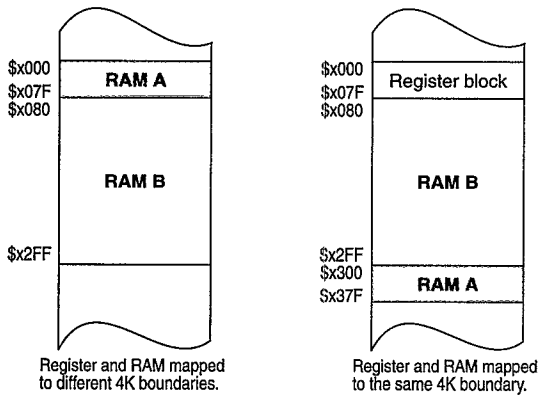


Figure 4-2 RAM and register overlap

4.3.2.3 INIT2 — EEPROM mapping register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
EEPROM mapping (INIT2)	$0037	EE3	EE2	EE1	EE0	0	0	0	0	0000 0000

This register determines the location of EEPROM in the memory map. INIT2 may be read at any time but bits 7–4 may be written only once after reset in normal modes.

EE[3:0] — EEPROM map position

EEPROM is located at $xD80–$xFFF, where x is the hexadecimal digit represented by EE[3:0]. Refer to Table 4-6.

Bits [3:0] — Not implemented; always read zero.

Table 4-6 EEPROM remapping

EE[3:0]	Location	EE[3:0]	Location
0000	$0D80–$0FFF	1000	$8D80–$8FFF
0001	$1D80–$1FFF	1001	$9D80–$9FFF
0010	$2D80–$2FFF	1010	$AD80–$AFFF
0011	$3D80–$3FFF	1011	$BD80–$BFFF
0100	$4D80–$4FFF	1100	$CD80–$CFFF
0101	$5D80–$5FFF	1101	$DD80–$DFFF
0110	$6D80–$6FFF	1110	$ED80–$EFFF
0111	$7D80–$7FFF	1111	$FD80–$FFFF

4.3.2.4 OPTION — System configuration options register 1

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
System config. options 1 (OPTION)	$0039	ADPU	CSEL	IRQE	DLY	CME	FCME	CR1	CR0	0001 0000

The 8-bit special-purpose OPTION register sets internal system configuration options during initialization. The time protected control bits IRQE, DLY, FCME and CR[1:0] can be written only once in the first 64 cycles after a reset and then they become read-only bits. This minimizes the possibility of any accidental changes to the system configuration. They may be written at any time in special modes.

ADPU — A/D power-up (refer to Section 11)

1 (set) – A/D system power enabled.

0 (clear) – A/D system disabled, to reduce supply current.

After enabling the A/D power, at least 100µs should be allowed for system stabilization.

CSEL — Clock select (refer to Section 11)

1 (set) – A/D and EEPROM use internal RC clock source (about 1.5MHz).

0 (clear) – A/D and EEPROM use system E clock (must be at least 1MHz).

This bit selects the clock source for the on-chip EEPROM and A/D charge pumps. The on-chip RC clock should be used when the E clock frequency falls below 1MHz.

IRQE — Configure $\overline{\text{IRQ}}$ for falling-edge-sensitive operation

1 (set) – Falling-edge-sensitive operation.

0 (clear) – Low-level-sensitive operation.

DLY — Enable oscillator start-up delay

1 (set) – A stabilization delay is imposed as the MCU is started up from STOP mode (or power-on reset).

0 (clear) – The oscillator start-up delay is bypassed and the MCU resumes processing within about four bus cycles. A stable external oscillator is required if this option is selected.

DLY is set on reset, so a delay is always imposed as the MCU is started up from power-on reset.

CME — Clock monitor enable (refer to Section 5)

1 (set) – Clock monitor enabled.

0 (clear) – Clock monitor disabled.

In order to use both STOP and clock monitor, the CME bit should be cleared before executing STOP, then set after recovering from STOP.

FCME — Force clock monitor enable (refer to Section 5)

1 (set) – Clock monitor enabled; cannot be disabled until next reset.

0 (clear) – Clock monitor follows the state of the CME bit.

When FCME is set, slow or stopped clocks will cause a clock failure reset sequence. To utilize STOP mode, FCME should always be cleared.

CR[1:0] — COP timer rate select bits (refer to Section 5)

These control bits determine a scaling factor for the watchdog timer.

4.3.2.5 OPT2 — System configuration options register 2

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
System config. options 2 (OPT2)	$0038	LIRDV	CWOM	STRCH	IRVNE	LSBF	SPR2	XDV1	XDV0	000x 0000

LIRDV — LIR driven

1 (set) – Enable $\overline{\text{LIR}}$ drive high pulse.

0 (clear) – $\overline{\text{LIR}}$ not driven high on MODA/$\overline{\text{LIR}}$ pin.

In single-chip and bootstrap modes, this bit has no meaning or effect. The $\overline{\text{LIR}}$ pin is driven low to indicate that execution of an instruction has begun. The $\overline{\text{LIR}}$ pin is normally configured for wired-OR operation (only pulls low). In order to detect consecutive instructions in a high-speed application, this signal can be made to drive high for a quarter of a cycle to prevent false triggering (LIRDV set).

CWOM — Port C wired-OR mode (refer to Section 6)

1 (set) – Port C outputs are open-drain.

0 (clear) – Port C operates normally.

STRCH — Stretch external accesses

1 (set) – Off-chip accesses are extended by one E clock cycle.

0 (clear) – Normal operation.

When this bit is set, off-chip accesses of addresses $8000 to $FFFF (with ROMAD = 0) or $0000 to $7FFF (with ROMAD = 1) are extended by one E clock cycle to allow access to slow peripherals. The E clock stretches externally, but the internal clocks are not affected, so that timers and serial systems are not corrupted.

Note: STRCH is cleared on reset; therefore a program cannot execute out of reset in a slow external ROM.

To use this feature, ROMON must be set on reset so that the device starts with internal ROM included in the memory map. STRCH should then be set.

STRCH has no effect in single chip and boot modes.

IRVNE — Internal read visibility/not E

IRVNE can be written once in any user mode. In **expanded modes**, IRVNE determines whether internal read visibility (IRV) is on or off, but has no meaning in user expanded secure mode, as IRV must be disabled. In special test modes, IRVNE is reset to one. In normal and bootstrap modes, IRVNE is reset to zero.

1 (set) – Data from internal reads is driven out of the external data bus.

0 (clear) – No visibility of internal reads on external bus.

In **single chip modes** this bit determines whether the E clock drives out from the chip.

1 (set) – E pin is driven low.

0 (clear) – E clock is driven out from the chip.

Refer to the following table for a summary of the operation immediately following reset.

Mode	IRVNE after reset	E clock after reset	IRV after reset	IRVNE affects only	IRVNE can be written
Single chip	0	On	Off	E	Once
Expanded	0	On	Off	IRV	Once
Boot	0	On	Off	E	Unlimited
Special test	1	On	On	IRV	Unlimited

LSBF — LSB-first enable (refer to Section 9)

1 (set) – Data is transferred LSB first.

0 (clear) – Data is transferred MSB first.

SPR2 — SPI clock rate select (refer to Section 9)

This bit adds a divide-by-four to the SPI clock chain.

4

XDV[1:0] — XOUT clock divide select

These two bits control the frequency of the XCLK signal, which is output on the XOUT pin if enabled by the CLKX bit in CONFIG. Table 4-7 shows some example frequencies. Once a clock rate has been selected, a maximum time of 16 E clock cycles should be allowed for the signal to stabilize. Note that on reset, both bits are cleared and the XCLK signal runs at the same frequency as EXTAL.

Note: The phase relationship between XOUT and EXTAL or E cannot be predicted.

Table 4-7 XOUT frequencies

XDV1	XDV0	EXTAL divided by:	XCLK if EXTAL = 8 MHz	XCLK if EXTAL = 8.4 MHz	XCLK if EXTAL = 12 MHz	XCLK if EXTAL = 12.6 MHz	XCLK if EXTAL = 16 MHz	XCLK if EXTAL = 16.8 MHz
0	0	1	8 MHz	8.4 MHz	12 MHz	12.6 MHz	16 MHz	16.8 MHz
0	1	4	2 MHz	2.1 MHz	3 MHz	3.15 MHz	4 MHz	4.2 MHz
1	0	6	1.33 MHz	1.4 MHz	2 MHz	2.1 MHz	2.7 MHz	2.8 MHz
1	1	8	1 MHz	1.05 MHz	1.5 MHz	1.57 MHz	2 MHz	2.1 MHz

4.3.2.6 BPROT — Block protect register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Block protect (BPROT)	$0035	BULKP	LVPEN	BPRT4	PTCON	BPRT3	BPRT2	BPRT1	BPRT0	1111 1111

BPROT prevents accidental writes to EEPROM and the CONFIG register. The bits in this register can be written to zero only once during the first 64 E clock cycles after reset in the normal modes; they can be set at any time. Once the bits are cleared, the EEPROM array and the CONFIG register can be programmed or erased. Setting the bits in the BPROT register to logic one protects the EEPROM and CONFIG register until the next reset. Refer to Table 4-8.

BULKP — Bulk erase of EEPROM protect

1 (set) – EEPROM cannot be bulk or row erased.

0 (clear) – EEPROM can be bulk erased normally.

LVPEN — EEPROM programming enable

The LVI is not activated on the MC68HC11KG4.

1 (set) – EEPROM programming is disabled.

0 (clear) – EEPROM programming is enabled.

Since the EEPROM low voltage detect circuitry (LVI) is not activated on this device, the LVPEN bit has only EEPROM programming enable and disable functionality.

PTCON — Protect for CONFIG register

1 (set) – CONFIG register cannot be programmed or erased.

0 (clear) – CONFIG register can be programmed or erased normally.

Note that, in special modes, CONFIG may be written regardless of the state of PTCON.

BPRT[4:0] — Block protect bits for EEPROM

1 (set) – Protection is enabled for associated block; it cannot be programmed or erased.

0 (clear) – Protection disabled for associated block.

Each of these four bits protects a block of EEPROM against writing or erasure, as follows:

Table 4-8 EEPROM block protect

Bit name	Block protected	Block size
BPRT0	$xD80–$xD9F	32 bytes
BPRT1	$xDA0–$xDDF	64 bytes
BPRT2	$xDE0–$xE5F	128 bytes
BPRT3	$xE60–$xF7F	288 bytes
BPRT4	$xF80–$xFFF	128 bytes

4.3.2.7 TMSK2 — Timer interrupt mask register 2

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer interrupt mask 2 (TMSK2)	$0024	TOI	RTII	PAOVI	PAII	0	0	PR1	PR0	0000 0000

PR[1:0] are time-protected control bits and can be changed only once and then only within the first 64 bus cycles after reset in normal modes.

Note: Bits in TMSK2 correspond bit for bit with flag bits in TFLG2. Ones in TMSK2 enable the corresponding interrupt sources.

TOI — Timer overflow interrupt enable (Refer to Section 10)

1 (set) – Interrupt requested when TOF is set.

0 (clear) – TOF interrupts disabled.

4

RTII — Real-time interrupt enable (Refer to Section 10)

1 (set) – Interrupt requested when RTIF set.

0 (clear) – RTIF interrupts disabled.

PAOVI — Pulse accumulator overflow interrupt enable (Refer to Section 10)

1 (set) – Interrupt requested when PAOVF set.

0 (clear) – PAOVF interrupts disabled.

PAII — Pulse accumulator interrupt enable (Refer to Section 10)

1 (set) – Interrupt requested when PAIF set.

0 (clear) – PAIF interrupts disabled.

Bits [3, 2] — Not implemented; always read zero.

PR[1:0] — Timer prescaler select

These two bits select the prescale rate for the main 16-bit free-running timer system. These bits can be written only once during the first 64 E clock cycles after reset in normal modes, or at any time in special modes. Refer to the following table:

PR[1:0]	Prescale factor
0 0	1
0 1	4
1 0	8
1 1	16

4.4 EEPROM and CONFIG register

4.4.1 EEPROM

The 640-byte on-board EEPROM is initially located from $0D80 to $0FFF after reset in all modes. It can be mapped to any other 4K page by writing to the INIT2 register. The EEPROM is enabled by the EEON bit in the CONFIG register. Programming and erasing are controlled by the PPROG register.

Unlike information stored in ROM, data in the 640 bytes of EEPROM can be erased and reprogrammed under software control. Because programming and erasing operations use an on-chip charge pump driven by V_{DD}, a separate external power supply is not required.

An internal charge pump supplies the programming voltage. Use of the block protect register (BPROT) prevents inadvertent writes to (or erases of) blocks of EEPROM (see Section 4.3.2.6). The CSEL bit in the OPTION register selects an on-chip oscillator clock for programming and erasing the EEPROM while operating at frequencies below 1 MHz.

In special modes there is one extra row of EEPROM, which is used for factory testing. Endurance and data retention specifications do not apply to these cells.

The erased state of each EEPROM byte is $FF.

4.4.1.1 PPROG — EEPROM programming control register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
EEPROM programming (PPROG)	$003B	ODD	EVEN	LVPI	BYTE	ROW	ERASE	EELAT	EEPGM	0000 0000

Note: Writes to EEPROM addresses are inhibited while EEPGM is one. A write to a different EEPROM location is prevented while a program or erase operation is in progress.

ODD — Program odd rows in half of EEPROM (Test)

EVEN — Program even rows in half of EEPROM (Test)

If both ODD and EVEN are set to one then all odd and even rows in half of the EEPROM will be programmed with the same data, within one programming cycle.

LVPI — EEPROM programming status

The LVI is not activated on the MC68HC11KG4.

1 (set) – EEPROM programming inhibited.

0 (clear) – EEPROM programming enabled.

This bit is read-only.

4

BYTE — EEPROM byte erase mode

1 (set) – Erase only one byte of EEPROM.

0 (clear) – Row or bulk erase mode used.

This bit may be read or written at any time.

ROW — EEPROM row/bulk erase mode (only valid when BYTE = 0)

1 (set) – Erase only one 16 byte row of EEPROM.

0 (clear) – Erase all 512 bytes of EEPROM.

This bit can be read or written at any time.

Table 4-9 Erase mode selection

Byte	Row	Action
0	0	Bulk erase (all 640 bytes)
0	1	Row erase (16 bytes)
1	0	Byte erase
1	1	Byte erase

ERASE — Erase/normal control for EEPROM

1 (set) – Erase mode.

0 (clear) – Normal read or program mode.

This bit can be read or written at any time.

EELAT — EEPROM latch control

1 (set) – EEPROM address and data bus set up for programming or erasing.

0 (clear) – EEPROM address and data bus set up for normal reads.

When the EELAT bit is cleared, the EEPROM can be read as if it were a ROM. The block protect register has no effect during reads. This bit can be read and written at any time.

EEPGM — EEPROM program command

1 (set) – Program or erase voltage switched on to EEPROM array.

0 (clear) – Program or erase voltage switched off to EEPROM array.

This bit can be read at any time but can only be written if EELAT = 1.

Note: If EELAT = 0 (normal operation) then EEPGM = 0 (programming voltage disconnected).

During EEPROM programming, the ROW and BYTE bits of PPROG are not used. If the frequency of the E clock is 1MHz or less, set the CSEL bit in the OPTION register. Remember that the EEPROM must be erased by a separate erase operation before programming. The following example of how to program an EEPROM byte assumes that the appropriate bits in BPROT have been cleared.

```
PROG    LDAB    #$02    EELAT=1
        STAB    $003B   Set EELAT bit
        STAA    $0D80   Store data to EEPROM address
        LDAB    #$03    EELAT=EEPGM=1
        STAB    $003B   Turn on programming voltage
        JSR     DLY10   Delay tEEPROG
        CLR     $003B   Turn off high voltage and set to READ mode
```

4.4.1.2 EEPROM bulk erase

To erase the EEPROM, ensure that the appropriate bits in the BPROT register are cleared, then complete the following steps using the PPROG register:

1) Write to PPROG with the ERASE, EELAT and appropriate BYTE and ROW bits set.
2) Write to the appropriate EEPROM address with any data. Row erase only requires a write to any location in the row. Bulk erase is accomplished by writing to any location in the array.
3) Write to PPROG with ERASE, EELAT, EEPGM and the appropriate BYTE and ROW bits set.
4) Delay for time t_{EEPROG}.
5) Clear the EEPGM bit in PPROG to turn off the high voltage.
6) Clear the PPROG register to reconfigure the EEPROM address and data buses for normal operation.

The following is an example of how to bulk erase the 640-byte EEPROM. The CONFIG register is not affected in this example.

```
BULKE   LDAB    #$06    EELAT=ERASE=1
        STAB    $003B   Set EELAT bit
```

```
        STAA    $0D80   Store data to any EEPROM address
        LDAB    #$07    EELAT=ERASE=EEPGM=1
        STAB    $003B   Turn on programming voltage
        JSR     DLY10   Delay tEEPROG
        CLR     $003B   Turn off high voltage and set to READ mode
```

4.4.1.3 EEPROM row erase

The following example shows how to perform a fast erase of 16 bytes of EEPROM:

```
ROWE    LDAB    #$0E    ROW=ERASE=EELAT=1
        STAB    $003B   Set to ROW erase mode
        STAB    0,X     Write any data to any address in ROW
        LDAB    #$0F    ROW=ERASE=EELAT=EEPGM=1
        STAB    $003B   Turn on high voltage
        JSR     DLY10   Delay tEEPROG
        CLR     $003B   Turn off high voltage and set to READ mode
```

4.4.1.4 EEPROM byte erase

The following is an example of how to erase a single byte of EEPROM:

```
BYTEE   LDAB    #$16    BYTE=ERASE=EELAT=1
        STAB    $003B   Set to BYTE erase mode
        STAB    0,X     Write any data to address to be erased
        LDAB    #$17    BYTE=ERASE=EELAT=EEPGM=1
        STAB    $003B   Turn on high voltage
        JSR     DLY10   Delay tEEPROG
        CLR     $003B   Turn off high voltage and set to READ mode
```

4.4.2 CONFIG register programming

Because the CONFIG register is implemented with EEPROM cells, use EEPROM procedures to erase and program this register. The procedure for programming is the same as for programming a byte in the EEPROM array, except that the CONFIG register address is used. CONFIG can be programmed or erased (including byte erase) while the MCU is operating in any mode, provided that PTCON in BPROT is clear. To change the value in the CONFIG register, complete the following procedure. Do not initiate a reset until the procedure is complete.

1) Erase the CONFIG register.
2) Program the new value to the CONFIG address.
3) Initiate reset.

CONFIG — System configuration register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Configuration control (CONFIG)	$003F	ROMAD	NWCOP	CLKX	PAREN	NOSEC	NOCOP	ROMON	EEON	xxxx xxxx

For a description of the bits contained in the CONFIG register refer to Section 4.3.2.1.

CONFIG is made up of EEPROM cells and static working latches. The operation of the MCU is controlled directly by these latches and not the EEPROM byte. When programming the CONFIG register, the EEPROM byte is accessed. When the CONFIG register is read, the static latches are accessed.

These bits can be read at any time. The value read is the one latched into the register from the EEPROM cells during the last reset sequence. A new value programmed into this register is not readable until after a subsequent reset sequence.

Bits in CONFIG can be written at any time if SMOD = 1 (bootstrap or special test mode). If SMOD = 0 (single chip or expanded mode), these bits can only be written using the EEPROM programming sequence, and none of the bits is readable or active until latched via the next reset.

4.4.3 RAM and EEPROM security

The optional security feature protects the contents of EEPROM and RAM from unauthorized access. Data, codes, keys, a program, or a key portion of a program, can be protected against access. To accomplish this, the protection mechanism restricts operation of protected devices to single-chip modes, and thus prevents the memory locations from being monitored externally (single-chip modes do not allow visibility of the internal address and data buses).Resident programs, however, have unlimited access to the internal EEPROM and RAM and can read, write, or transfer the contents of these memories.

Note: A mask option on the MC68HC11KG4 determines whether or not the security feature is available. If the feature is available, then the secure mode can be invoked by programming the NOSEC bit to zero. Otherwise, the NOSEC bit is permanently set to one, disabling security.

If the security feature is present and enabled and bootstrap mode is selected, then the following sequence is performed by the bootstrap program:

1) Output $FF on the SCI.
2) Turn block protect off. Clear BPROT register.
3) If EEPROM is enabled, erase it all.
4) Verify that the EEPROM is erased; if not, begin sequence again.
5) Write $FF to every RAM byte.
6) Erase the CONFIG register.

If all the above operations are successful, the bootloading process continues as if the device has not been secured.

CONFIG — System configuration register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Configuration control (CONFIG)	$003F	ROMAD	NWCOP	CLKX	PAREN	NOSEC	NOCOP	ROMCN	EEON	xxxx xxxx

For a description of the other bits contained in the CONFIG register refer to Section 4.3.2.1.

NOSEC — EEPROM security disabled

1 (set) – Disable security.

0 (clear) – Enable security.

With security enabled, selection of special test mode is prevented; single chip and user expanded modes may be accessed. If the MODA and MODB pins are configured for special test mode, the part will start in bootstrap mode.

5
RESETS AND INTERRUPTS

Resets and interrupt operations load the program counter with a vector that points to a new location from which instructions are to be fetched. A reset immediately stops execution of the current instruction and forces the program counter to a known starting address. Internal registers and control bits are initialized so that the MCU can resume executing instructions. An interrupt temporarily suspends normal program execution whilst an interrupt service routine is being executed. After an interrupt has been serviced, the main program resumes as if there had been no interruption.

5.1 Resets

There are four possible sources of reset. Power-on reset (POR) and external reset share the normal reset vector. The computer operating properly (COP)† reset and the clock monitor reset each has its own vector.

5.1.1 Power-on reset

A positive transition on V_{DD} generates a power-on reset (POR), which is used for power-up conditions. POR can also be used to detect drops in power supply voltages. A 4064 t_{CYC} delay is imposed which allows the clock generator to stabilize after the oscillator becomes active. If $\overline{\text{RESET}}$ is at logical zero at the end of this delay, the CPU remains in the reset condition until $\overline{\text{RESET}}$ goes to logical one.

The POR circuitry holds the $\overline{\text{RESET}}$ pin low whenever V_{DD} falls below the minimum operating level. Refer to Figure 2-2 and Figure 5-1. When V_{DD} increases above the voltage V_0, the CPU immediately starts coming out of reset, and when V_{DD} decreases below the voltage V_1, a normal reset is executed.

A reset is guaranteed if V_{DD} is greater than V_{MIN}. For the values of the voltages, refer to the electrical characteristics of the POR in Section A.5.8.

† Also applies to WCOP.

Note: In order to allow for future applications at lower operating voltages, a mask option on the MC68HC11KG4 allows the LVR circuitry to be permanently disabled. This reduces the current consumption in STOP mode to less than 100µA.

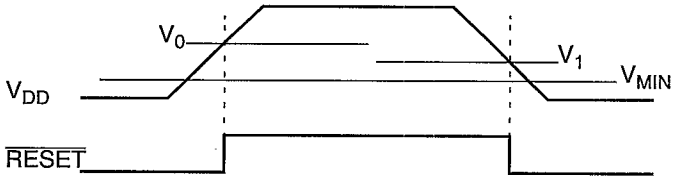


Figure 5-1 Power on reset

5.1.2 External reset ($\overline{\text{RESET}}$)

The CPU distinguishes between internal and external reset conditions by sensing whether the reset pin rises to a logic one in less than four E clock cycles after an internal device releases reset. When a reset condition is sensed, the $\overline{\text{RESET}}$ pin is driven low by an internal device for eight E clock cycles, then released. Four E clock cycles later it is sampled. If the pin is still held low, the CPU assumes that an external reset has occurred. If the pin is high, it indicates that the reset was initiated internally by either the COP system or the clock monitor. It is not advisable to connect an external resistor capacitor (RC) power-up delay circuit to the reset pin of M68HC11 devices because the circuit charge time constant can cause the device to misinterpret the type of reset that occurred. To guarantee recognition of an external reset, the $\overline{\text{RESET}}$ pin should be held low for at least 16 clock cycles.

5.1.3 COP and WCOP resets

The MC68HC11KG4 includes a computer operating properly (COP) system to help protect against software failures. For a greater system security however, a Window COP (WCOP) function is also included. The WCOP operation is similar to that of the COP, but the WCOP counter may only be reset by software within a predefined time window. If both systems are enabled, the WCOP has priority and the COP is practically redundant.

When the COP/WCOP is enabled, the software is responsible for keeping a free-running watchdog timer from timing out. When the software is no longer being executed in the intended sequence, a system reset is initiated.

The state of the NOCOP/NWCOP bit in the CONFIG register determines whether the COP/WCOP system is enabled or disabled. To change the enable status of the COP/WCOP system, change the contents of the CONFIG register and then perform a system reset. In the special test and

bootstrap operating modes, the COP and WCOP systems are initially inhibited by the disable resets (DISR) control bit in the TEST1 register. The DISR bit can subsequently be written to zero to enable COP/WCOP resets.

The COP system is clocked by ST4XCK/2^{17} and the WCOP system is clocked by ST4XCK/2^{14} (see Section 10). If the PLL circuit is active (VDDSYN = 1) and MCS and BCS are both set, then ST4XCK is equal to the output of the PLL circuit, VCOOUT. Otherwise, ST4XCK is the same as EXTALi. Refer to Figure 10-1.

The COP/WCOP timer rate control bits CR[1:0] in the OPTION register determine the COP/WCOP timeout period. ST4XCK/2^{17} or ST4XCK/2^{14}, depending on whether the COP or WCOP system is selected, is scaled by the factors shown in Table 5-1 and Table 5-1. After reset, these bits are zero, which selects the shortest timeout period. In normal operating modes, these bits can only be written once, within 64 bus cycles after reset.

Table 5-1 COP timer rate select

CR[1:0]	E/2^{15} divided by:	XTAL = 4 MHz: timeout(1)	XTAL = 8 MHz: timeout(1)	XTAL = 2^{23}: timeout(1)	XTAL = 12.6MHz: timeout(1)	XTAL = 2^{24}: timeout(1)
0 0	1	32.768 ms	16.384 ms	15.615 ms	10.42 ms	7.938 ms
0 1	4	131.07 ms	65.536 ms	62.5 ms	41.67 ms	31.75 ms
1 0	16	524.29 ms	262.14 ms	250 ms	166.7 ms	125 ms
1 1	64	2.1 s	1.049 s	1 s	667 ms	500 ms

Table 5-2 WCOP timer rate select

CR[1:0]	Divide ST4XCK/2^{14} by	ST4XCK = 4 MHz: timeout(1)	ST4XCK = 8 MHz: timeout(1)	ST4XCK = 2^{23}: timeout(1)	ST4XCK = 12.6 MHz: timeout(1)	ST4XCK = 2^{24}: timeout(1)
0 0	8	32.768 ms	16.384 ms	15.615 ms	10.42 ms	7.938 ms
0 1	32	131.07 ms	65.536 ms	62.5 ms	41.67 ms	31.75 ms
1 0	64	262.14 ms	131.07 ms	125 ms	83.2 ms	62.5 ms
1 1	128	524.28 ms	262.14 ms	250 s	166.7 ms	125 ms

(1) The timeout period has a tolerance of –0/+one cycle of the ST4XCK/2^{17} or ST4XCK/2^{14} clock due to the asynchronous implementation of the COP/WCOP circuitry. For example, with ST4XCK = 8 MHz, the uncertainty is –0/+16.384 ms. See also the ***M68HC11 Reference Manual, (M68HC11RM/AD).***

The WCOP has a programmable timeout period. The time range is from 8ms to 128ms with a 4MHz core frequency. As with the COP system, the timeout period is selected using the timer rate control bits CR[1:0] in the OPTION register. These bits are shared with the normal COP and apply to whichever COP system is enabled. If both systems are enabled the WCOP has priority and the COP is practically redundant. Under such circumstances, the COP cannot cause a system reset since the WCOP reset affects both COP and WCOP and the COP timeouts are longer.

The window size is fixed at 50% of the timeout period, as shown in Figure 5-2.

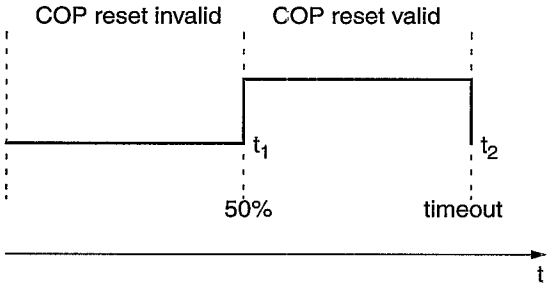


Figure 5-2 WCOP period

A software reset of the WCOP counter is performed by writing $55 followed by $AA to the COPRST register. The effect, which depends on the reset time within the WCOP period, is summarised in Table 5-3.

Table 5-3 Effect of a software reset at different stages of WCOP period

Period containing reset time t	Resulting effect
$0 < t < t_1$	System reset occurs
$t_1 < t < t_2$	WCOP counter restarts counting at zero
$t_2 < t$ (reset after timeout)	WCOP counter overflow produces system reset

If the WCOP reset does not occur within the valid window, the WCOP automatically produces a system reset. The WCOP is always enabled after reset except in special test mode.

On the MC68HC11KG4 it is possible to permanently enable or disable the WCOP with a mask option. If the WCOP mask option is enabled, the WCOP is always active in normal modes independent of the NWCOP bit in the CONFIG register. If the WCOP mask option is disabled, the WCOP activity depends on the NWCOP bit in the CONFIG register in normal modes. In special modes the WCOP activity always depends on NWCOP, independent of the mask option status.

The WCOP does not run in the WAIT state. There is no reset generated by the WCOP when the CPU is in WAIT state.

5.1.3.1 COPRST — Arm/reset COP/WCOP timer circuitry register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
COP timer arm/reset (COPRST)	$003A	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	not affected

This register is used to reset the COP and WCOP counters.

The following reset sequence will service the COP/WCOP timer. Write $55 to COPRST to arm the COP timer clearing mechanism. Then write $AA to COPRST to clear the COP/WCOP timer. Executing instructions between these two steps is possible as long as both steps are completed in the correct sequence before the timer times out.

5.1.4 Clock monitor reset

The clock monitor circuit is based on an internal RC time delay. If no MCU clock edges are detected within this RC time delay, the clock monitor can optionally generate a system reset. The clock monitor function is enabled or disabled by the CME control bit in the OPTION register. The presence of a timeout is determined by the RC delay, which allows the clock monitor to operate without any MCU clocks.

Clock monitor is used as a backup for the COP system. Because the COP needs a clock to function, it is disabled when the clocks stop. Therefore, the clock monitor system can detect clock failures not detected by the COP system.

Semiconductor wafer processing causes variations of the RC timeout values between individual devices. An E clock frequency below 10 kHz is detected as a clock monitor error. An E clock frequency of 200 kHz or more prevents clock monitor errors. Use of the clock monitor function when the E clock is below 200 kHz is not recommended.

Special considerations are needed when a STOP instruction is executed and the clock monitor is enabled. Because the STOP function causes the clocks to be halted, the clock monitor function generates a reset sequence if it is enabled at the time the STOP mode was initiated. Before executing a STOP instruction, clear the CME bit in the OPTION register to zero to disable the clock monitor. After recovery from STOP, set the CME bit to logic one to enable the clock monitor.

5.1.5 OPTION — System configuration options register 1

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
System config. options 1 (OPTION)	$0039	ADPU	CSEL	IRQE	DLY	CME	FCME	CR1	CR0	0001 0000

The special-purpose OPTION register sets internal system configuration options during initialization. The time protected control bits (IRQE, DLY, FCME and CR[1:0]) can be written to only once in the first 64 cycles after a reset and then they become read-only bits. This minimizes the possibility of any accidental changes to the system configuration. They may be written at any time in special modes.

5

ADPU — A/D power-up (Refer to Section 11)

1 (set) – A/D system power enabled.

0 (clear) – A/D system disabled, to reduce supply current.

CSEL — Clock select (Refer to Section 11)

1 (set) – A/D, *EPROM* and EEPROM use internal RC clock (about 1.5MHz).

0 (clear) – A/D, *EPROM* and EEPROM use system E clock (must be at least 1MHz).

IRQE — Configure $\overline{\text{IRQ}}$ for falling-edge-sensitive operation (Refer to Section 4)

1 (set) – Falling-edge-sensitive operation.

0 (clear) – Low-level-sensitive operation.

DLY — Enable oscillator start-up delay (Refer to Section 4)

1 (set) – A stabilization delay is imposed as the MCU is started up from STOP mode (or from power-on reset).

0 (clear) – The oscillator start-up delay is bypassed and the MCU resumes processing within about four bus cycles. A stable external oscillator is required if this option is selected.

Note: Because DLY is set on reset, a delay is always imposed as the MCU is started up from power-on reset.

CME — Clock monitor enable

1 (set) – Clock monitor enabled.

0 (clear) – Clock monitor disabled.

This control bit can be read or written at any time and controls whether or not the internal clock monitor circuit triggers a reset sequence when the system clock is slow or absent. When it is clear, the clock monitor circuit is disabled, and when it is set, the clock monitor circuit is enabled. Reset clears the CME bit.

In order to use both STOP and clock monitor, the CME bit should be cleared before executing STOP, then set after recovering from STOP.

FCME — Force clock monitor enable

1 (set) – Clock monitor enabled; cannot be disabled until next reset.

0 (clear) – Clock monitor follows the state of the CME bit.

When FCME is set, slow or stopped clocks will cause a clock failure reset sequence. To utilize STOP mode, FCME should always be cleared.

CR[1:0] — COP timer rate select bits

The COP/WCOP function is clocked by ST4XCK/2^{17}. ST4XCK can be either EXTALi or VCOOUT (see Section 5.1.3). These control bits determine a scaling factor for the watchdog timer period. See Table 5-1.

5.1.6 CONFIG — Configuration control register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Configuration control (CONFIG)	$003F	ROMAD	NWCOP	CLK4X	PAREN	NOSEC	NOCOP	ROMON	EEON	xxxx xxxx

Among other things, CONFIG controls the presence and location of EEPROM in the memory map and enables the COP/WCOP watchdog system. A security feature that protects data in EEPROM and RAM is available on mask programmed MCUs.

CONFIG is made up of EEPROM cells and static working latches. The operation of the MCU is controlled directly by these latches and not the EEPROM byte. When programming the CONFIG register, the EEPROM byte is accessed. When the CONFIG register is read, the static latches are accessed.

These bits can be read at any time. The value read is the one latched into the register from the EEPROM cells during the last reset sequence. A new value programmed into this register is not readable until after a subsequent reset sequence.

Bits in CONFIG can be written at any time if SMOD = 1 (bootstrap or special test mode). If SMOD = 0 (single chip or expanded mode), they can only be written using the EEPROM programming sequence, and are neither readable nor active until latched via the next reset.

ROMAD — ROM/EPROM mapping control (refer to Section 4)

1 (set) – ROM/*EPROM* addressed from $A000 to $FFFF.

0 (clear) – ROM/*EPROM* addressed from $2000 to $7FFF (expanded mode only).

In single chip mode, reset sets this bit.

NWCOP — WCOP system disable

1 (set) – WCOP system disabled.

0 (clear) – WCOP system enabled (forces reset on timeout).

5

CLKX — XOUT enable (refer to Section 4)

1 (set) – XCLK signal driven out on the XOUT pin

0 (clear) – XOUT pin disabled.

The frequency of the XCLK signal is controlled by two bits in the OPT2 register.

PAREN — Pull-up assignment register enable (refer to Section 6)

1 (set) – PPAR register enabled; pull-ups can be enabled using PPAR.

0 (clear) – All pull-ups disabled (not controlled by PPAR).

NOSEC — EEPROM security disabled (refer to Section 4)

1 (set) – Disable security.

0 (clear) – Enable security.

NOCOP — COP system disable

1 (set) – COP system disabled.

0 (clear) – COP system enabled (forces reset on timeout).

ROMON — ROM/*EPROM* enable (refer to Section 4)

1 (set) – ROM/*EPROM* included in the memory map.

0 (clear) – ROM/*EPROM* excluded from the memory map.

EEON — EEPROM enable (refer to Section 4)

1 (set) – EEPROM included in the memory map.

0 (clear) – EEPROM excluded from the memory map.

5.2 Effects of reset

When a reset condition is recognized, the internal registers and control bits are forced to an initial state. Depending on the cause of the reset and the operating mode, the reset vector can be fetched from any of six possible locations, as shown in Table 5-4.

Table 5-4 Reset cause, reset vector and operating mode

Cause of reset	Normal mode vector	Special test or bootstrap
POR or $\overline{\text{RESET}}$ pin	$FFFE, $FFFF	$BFFE, $BFFF
Clock monitor failure	$FFFC, $FFFD	$BFFC, $BFFD
COP/WCOP watchdog timeout	$FFFA, $FFFB	$BFFA, $BFFB

These initial states then control on-chip peripheral systems to force them to known start-up states, as described in the following paragraphs.

5.2.1 Central processing unit

After reset, the CPU fetches the restart vector from the appropriate address during the first three cycles, and begins executing instructions. The stack pointer and other CPU registers are indeterminate immediately after reset; however, the X and I interrupt mask bits in the condition code register (CCR) are set to mask any interrupt requests. Also, the S-bit in the CCR is set to inhibit the STOP mode.

5.2.2 Memory map

After reset, the INIT register is initialized to $00, putting the 768 bytes of RAM at locations $0080–$037F, and the control registers at locations $0000–$007F. The INIT2 register puts EEPROM at locations $0D80–$0FFF.

5.2.3 Parallel I/O

When a reset occurs in expanded operating modes, port B, C, and F pins used for parallel I/O are dedicated to the expansion bus. If a reset occurs during a single chip operating mode, all ports are configured as general purpose high-impedance inputs.

Note: Do not confuse pin function with the electrical state of the pin at reset. All general-purpose I/O pins configured as inputs at reset are in a high-impedance state. Port data registers reflect the port's functional state at reset. The pin function is mode dependent.

5.2.4 Timer

During reset, the timer system is initialized to a count of $0000. The prescaler bits are cleared, and all output compare registers are initialized to $FFFF. All input capture registers are indeterminate after reset. The output compare 1 mask (OC1M) register is cleared so that successful OC1 compares do not affect any I/O pins. The other four output compares are configured so that they do not affect any I/O pins on successful compares. All input capture edge-detector circuits are configured for capture disabled operation. The timer overflow interrupt flag and all eight timer function interrupt flags are cleared. All nine timer interrupts are disabled because their mask bits have been cleared.

The I4/O5 bit in the PACTL register is cleared to configure the I4/O5 function as OC5; however, the OM5:OL5 control bits in the TCTL1 register are clear so OC5 does not control the PA3 pin.

5.2.5 Real-time interrupt (RTI)

The real-time interrupt flag (RTIF) is cleared and automatic hardware interrupts are masked. The rate control bits are cleared after reset and can be initialized by software before the real-time interrupt (RTI) system is used.

5.2.6 Slow real-time interrupt (SRTI)

The slow real-time interrupt flag (RTF) is cleared, as are the SRTI clock prescaler bits RTCK1 and RTCK2, which can be initialized by software before the SRTI system is used.

5.2.7 Pulse accumulator

The pulse accumulator system is disabled at reset so that the pulse accumulator input (PAI) pin defaults to being a general-purpose input pin.

5.2.8 COP/WCOP system

The COP/WCOP watchdog system is enabled if the NOCOP/NWCOP control bit in the CONFIG register is cleared, and disabled if NOCOP/NWCOP is set. Both the COP and WCOP rate is set for the shortest duration timeout.

5.2.9 Serial communications interface (SCI)

The reset condition of the SCI system is independent of the operating mode. At reset, the SCI baud rate control register is initialized to $0004. All transmit and receive interrupts are masked and both the transmitter and receiver are disabled so the port pins default to being general purpose I/O lines. The SCI frame format is initialized to an 8-bit character size. The send break and receiver wake-up functions are disabled. The TDRE and TC status bits in the SCI status register are both set, indicating that there is no transmit data in either the transmit data register or the transmit serial shift register. The RDRF, IDLE, OR, NF, FE, PF, and RAF receive-related status bits are cleared.

Note: The foregoing paragraph also applies to SCI2.

5.2.10 Serial peripheral interface (SPI)

The SPI system is disabled by reset. Its associated port pins default to being general purpose I/O lines.

5.2.11 Pulse width modulation (PWM) timer

The PWEN channel enable register is set to $00 on reset, causing the port H and port K pins to be configured as general purpose outputs. The PWM channels are therefore disabled.

5.2.12 Analog-to-digital converter

The A/D converter configuration is indeterminate after reset. The ADPU bit is cleared by reset, which disables the A/D system. The conversion complete flag is cleared by reset.

5.2.13 Digital-to-analog converter

The digital-to-analog converter is disabled on reset and PE7 is therefore an input.

5.2.14 System

The EEPROM programming controls are disabled, so the memory system is configured for normal read operation. PSEL[4:0] are initialized with the binary value %00110, causing the external $\overline{IRQ}$ pin to have the highest I-bit interrupt priority. The $\overline{IRQ}$ and $\overline{XIRQ}$ pins are configured for level-sensitive operation (for wired-OR systems). The RBOOT, SMOD, and MDA bits in the HPRIO register reflect the status of the MODB and MODA inputs at the rising edge of reset. The DLY control bit is set to specify that an oscillator start-up delay is imposed upon recovery from STOP mode or power-on reset. The clock monitor system is disabled because CME and FCME are cleared.

5

5.3 Reset and interrupt priority

Resets and interrupts have a hardware priority that determines which reset or interrupt is serviced first when simultaneous requests occur. Any maskable interrupt can be given priority over other maskable interrupts.

The first six interrupt sources are not maskable by the I-bit in the CCR. The priority arrangement for these sources is fixed and is as follows:

1) POR or $\overline{RESET}$ pin
2) Clock monitor reset
3) COP/WCOP watchdog reset
4) Illegal opcode interrupt — see Section 5.4.3 for details of handling
5) Software interrupt (SWI) — see Section 5.4.4 for details of handling
6) $\overline{XIRQ}$ interrupt (SRTI interrupt wired to $\overline{XIRQ}$)

The maskable interrupt sources have the following priority arrangement:

7) $\overline{IRQ}$
8) Real-time interrupt
9) Timer input capture 1
10) Timer input capture 2
11) Timer input capture 3
12) Timer output compare 1
13) Timer output compare 2
14) Timer output compare 3
15) Timer output compare 4
16) Timer input capture 4/output compare 5
17) Timer overflow
18) PWM interrupt
19) Pulse accumulator overflow

20) Pulse accumulator input edge

21) SCI1 serial system

22) Port J keyboard interrupt

23) SPI transfer complete

24) SCI2 serial system

Any one of these maskable interrupts can be assigned the highest maskable interrupt priority by writing the appropriate value to the PSEL bits in the HPRIO register. Otherwise, the priority arrangement remains the same. An interrupt that is assigned highest priority is still subject to global masking by the I-bit in the CCR, or by any associated local bits. Interrupt vectors are not affected by priority assignment. To avoid race conditions, HPRIO can only be written while I-bit interrupts are inhibited.

5.3.1 HPRIO — Highest priority I-bit interrupt and misc. register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Highest priority interrupt (HPRIO)	$003C	RBOOT	SMOD	MDA	PSEL4	PSEL3	PSEL2	PSEL1	PSEL0	xxx0 0110

RBOOT, SMOD, and MDA bits depend on power-up initialization mode and can only be written in special modes when SMOD = 1. Refer to Table 4-4.

RBOOT — Read bootstrap ROM (refer to Section 4)

1 (set) – Bootloader ROM enabled, at $BE40–$BFFF.

0 (clear) – Bootloader ROM disabled and not in map.

SMOD — Special mode select (refer to Section 4)

1 (set) – Special mode variation in effect.

0 (clear) – Normal mode variation in effect.

MDA — Mode select A (refer to Section 4)

1 (set) – Normal expanded or special test mode in effect.

0 (clear) – Normal single chip or special bootstrap mode in effect.

PSEL[4:0] — Priority select bits

These bits select one interrupt source to be elevated above all other I-bit-related sources and can be written to only while the I-bit in the CCR is set (interrupts disabled). See Table 5-5.

Table 5-5 Highest priority interrupt selection

PSELx					Interrupt source promoted
4	3	2	1	0	
0	0	0	X	X	Reserved (default to $\overline{IRQ}$)
0	0	1	0	0	Reserved (default to $\overline{IRQ}$)
0	0	1	0	1	Reserved (default to $\overline{IRQ}$)
0	0	1	1	0	$\overline{IRQ}$ (external pin)
0	0	1	1	1	Real-time interrupt
0	1	0	0	0	Timer input capture 1
0	1	0	0	1	Timer input capture 2
0	1	0	1	0	Timer input capture 3
0	1	0	1	1	Timer output compare 1
0	1	1	0	0	Timer output compare 2
0	1	1	0	1	Timer output compare 3
0	1	1	1	0	Timer output compare 4
0	1	1	1	1	Timer output compare 5/input capture 4
1	0	0	0	0	Timer overflow
1	0	0	0	1	Pulse accumulator overflow
1	0	0	1	0	Pulse accumulator input edge
1	0	0	1	1	SPI serial transfer complete
1	0	1	0	0	SCI2 serial system
1	0	1	0	1	Reserved
1	0	1	1	0	Reserved (default to $\overline{IRQ}$)
1	0	1	1	1	PWM interrupt
1	1	0	0	0	SCI1 serial system
1	1	0	0	1	Port J keyboard interrupt
1	1	0	1	0	Reserved (default to $\overline{IRQ}$)
1	1	0	1	1	Reserved (default to $\overline{IRQ}$)
1	1	1	X	X	Reserved (default to $\overline{IRQ}$)

Table 5-6 Interrupt and reset vector assignments

Vector address	Interrupt source	CCR mask bit	Local mask
FFC0, C1 – FFCA, CB	Reserved	—	—
FFCC, CD	Port J keyboard interrupt	I	KMSK[3:0]
FFCE, CF	• SCI1 receive data register full • SCI1 receiver overrun • SCI1 transmit data register empty • SCI1 transmit complete • SCI1 idle line detect	I	RIE RIE TIE TCIE ILIE
FFD0 - FFD5	PWM interrupt	I	PWMI
FFD6, D7	• SCI2 receive data register full • SCI2 receiver overrun • SCI2 transmit data register empty • SCI2 transmit complete • SCI2 idle line detect	I	RIE RIE TIE TCIE ILIE
FFD8, D9	SPI serial transfer complete	I	SPIE
FFDA, DB	Pulse accumulator input edge	I	PAII
FFDC, DD	Pulse accumulator overflow	I	PAOVI
FFDE, DF	Timer overflow	I	TOI
FFE0, E1	Timer input capture 4/output compare 5	I	I4/O5I
FFE2, E3	Timer output compare 4	I	OC4I
FFE4, E5	Timer output compare 3	I	OC3I
FFE6, E7	Timer output compare 2	I	OC2I
FFE8, E9	Timer output compare 1	I	OC1I
FFEA, EB	Timer input capture 3	I	IC3I
FFEC, ED	Timer input capture 2	I	IC2I
FFEE, EF	Timer input capture 1	I	IC1I
FFF0, F1	Real-time interrupt	I	RTII
FFF2, F3	$\overline{\text{IRQ}}$ pin	I	None
FFF4, F5	$\overline{\text{XIRQ}}$ pin (SRTI interrupt wired to $\overline{\text{XIRQ}}$)	X	None
FFF6, F7	Software interrupt	None	None
FFF8, F9	Illegal opcode trap	None	None
FFFA, FB	COP failure WCOP failure	None None	NOCOP NWCOP
FFFC, FD	Clock monitor fail	None	CME
FFFE, FF	$\overline{\text{RESET}}$	None	None

5.4 Interrupts

Excluding reset type interrupts, the MC68HC11KG4 has 21 interrupt vectors that support 32 interrupt sources. The 18 maskable interrupts are generated by on-chip peripheral systems. These interrupts are recognized when the global interrupt mask bit (I) in the condition code register (CCR) is clear. The three nonmaskable interrupt sources are illegal opcode trap, software interrupt, and $\overline{\text{XIRQ}}$ pin. Refer to Table 5-6, which shows the interrupt sources and vector assignments for each source.

For some interrupt sources, such as the SCI interrupts, the flags are automatically cleared during the normal course of responding to the interrupt requests. For example, the RDRF flag in the SCI system is cleared by the automatic clearing mechanism consisting of a read of the SCI status register while RDRF is set, followed by a read of the SCI data register. The normal response to an RDRF interrupt request would be to read the SCI status register to check for receive errors, then to read the received data from the SCI data register. These two steps satisfy the automatic clearing mechanism without requiring any special instructions.

5

5.4.1 Interrupt recognition and register stacking

An interrupt can be recognized at any time after it is enabled by its local mask, if any, and by the global mask bit in the CCR. Once an interrupt source is recognized, the CPU responds at the completion of the instruction being executed. Interrupt latency varies according to the number of cycles required to complete the current instruction. When the CPU begins to service an interrupt, the contents of the CPU registers are pushed onto the stack in the order shown in Table 5-7. After the CCR value is stacked, the I-bit and the X-bit, if $\overline{\text{XIRQ}}$ is pending, are set to inhibit further interrupts. The interrupt vector for the highest priority pending source is fetched, and execution continues at the address specified by the vector. At the end of the interrupt service routine, the return from interrupt instruction is executed and the saved registers are pulled from the stack in reverse order so that normal program execution can resume. Refer to Section 3 for further information.

Table 5-7 Stacking order on entry to interrupts

Memory location	CPU registers
SP	PCL
SP – 1	PCH
SP – 2	IYL
SP – 3	IYH
SP – 4	IXL
SP – 5	IXH
SP – 6	ACCA
SP – 7	ACCB
SP – 8	CCR

5.4.2 Nonmaskable interrupt request ($\overline{\text{XIRQ}}$)

Nonmaskable interrupts are useful because they can always interrupt CPU operations. The most common use for such an interrupt is for serious system problems, such as program runaway or power failure. The $\overline{\text{XIRQ}}$ input is an updated version of the $\overline{\text{NMI}}$ (nonmaskable interrupt) input of earlier MCUs.

Upon reset, both the X-bit and I-bit of the CCR are set to inhibit all maskable interrupts and $\overline{\text{XIRQ}}$. After minimum system initialization, software can clear the X-bit by a TAP instruction, enabling $\overline{\text{XIRQ}}$ interrupts. Thereafter, software cannot set the X-bit. Thus, an $\overline{\text{XIRQ}}$ interrupt is a nonmaskable interrupt. Because the operation of the I-bit-related interrupt structure has no effect on the X-bit, the internal $\overline{\text{XIRQ}}$ pin remains unmasked. In the interrupt priority logic, the $\overline{\text{XIRQ}}$ interrupt has a higher priority than any source that is maskable by the I-bit. All I-bit-related interrupts operate normally with their own priority relationship.

When an I-bit-related interrupt occurs, the I-bit is automatically set by hardware after stacking the CCR byte. The X-bit is not affected. When an X-bit-related interrupt occurs, both the X and I bits are automatically set by hardware after stacking the CCR. A return from interrupt instruction restores the X and I bits to their pre-interrupt request state.

5.4.3 Illegal opcode trap

Because not all possible opcodes or opcode sequences are defined, the MCU includes an illegal opcode detection circuit, which generates an interrupt request. When an illegal opcode is detected and the interrupt is recognized, the current value of the program counter is stacked. After interrupt service is complete, the user should reinitialize the stack pointer to ensure that repeated execution of illegal opcodes does not cause stack underflow. Left uninitialized, the illegal opcode vector can point to a memory location that contains an illegal opcode. This condition causes an infinite loop that causes stack underflow. The stack grows until the system crashes.

The illegal opcode trap mechanism works for all unimplemented opcodes on all four opcode map pages. The address stacked as the return address for the illegal opcode interrupt is the address of the first byte of the illegal opcode. Otherwise, it would be almost impossible to determine whether the illegal opcode had been one or two bytes. The stacked return address can be used as a pointer to the illegal opcode, so that the illegal opcode service routine can evaluate the offending opcode.

5.4.4 Software interrupt

SWI is an instruction, and thus cannot be interrupted until complete. SWI is not inhibited by the global mask bits in the CCR. Because execution of SWI sets the I mask bit, once an SWI interrupt begins, other interrupts are inhibited until SWI is complete, or until user software clears the I bit in the CCR.

5.4.5 Maskable interrupts

The maskable interrupt structure of the MCU can be extended to include additional external interrupt sources through the $\overline{IRQ}$ pin. The default configuration of this pin is a low-level sensitive wired-OR network. When an event triggers an interrupt, a software accessible interrupt flag is set. When enabled, this flag causes a constant request for interrupt service. After the flag is cleared, the service request is released.

5.4.6 Reset and interrupt processing

The following flow diagrams illustrate the reset and interrupt process. Figure 5-3 and Figure 5-4 illustrate how the CPU begins from a reset and how interrupt detection relates to normal opcode fetches. Figure 5-5 to Figure 5-6 provide an expanded version of a block in Figure 5-3 and illustrate interrupt priorities. Figure 5-8 shows the resolution of interrupt sources within the SCI subsystem.

5.5 Low power operation

Both STOP and WAIT suspend CPU operation until a reset or interrupt occurs. The WAIT condition suspends processing and reduces power consumption to an intermediate level. The STOP condition turns off all on-chip clocks and reduces power consumption to an absolute minimum while retaining the contents of all bytes of the RAM.

5.5.1 WAIT

The WAI opcode places the MCU in the WAIT condition, during which the CPU registers are stacked and CPU processing is suspended until a qualified interrupt is detected. The interrupt can be an external $\overline{IRQ}$, an $\overline{XIRQ}$, or any of the internally generated interrupts, such as the timer or serial interrupts. The on-chip crystal oscillator remains active throughout the WAIT stand-by period.

The reduction of power in the WAIT condition depends on how many internal clock signals driving on-chip peripheral functions can be shut down. The CPU is always shut down during WAIT. While in the wait state, the address/data bus repeatedly runs read cycles to the address where the CCR contents were stacked. The MCU leaves the wait state when it senses any interrupt that has not been masked.

The PH2 clock to the free-running timer system is stopped if the I-bit is set and the COP/WCOP system is disabled by NOCOP/NWCOP being set. Several other systems can also be in a reduced power consumption state depending on the state of software-controlled configuration control bits. It is not possible to inhibit the SRTI in WAIT mode; the SRTI is therefore wired to the nonmaskable XIRQ interrupt line. Power consumption by the analog-to-digital (A/D) converter is not affected significantly by the WAIT condition. However, the A/D converter current can be eliminated by

writing the ADPU bit to zero and halting the RC clock (CSEL cleared). The SPI system is enabled or disabled by the SPE control bit, the SCI transmitter is enabled or disabled by the TE bit, and the SCI receiver is enabled or disabled by the RE bit (lowest power consumption is achieved when RE=TE=0). Power consumption is reduced if all the PWM enable bits (PWEN[4:1]) are cleared, thereby disabling every PWM channel. Setting the WEN bit in PLLCR will result in WAIT mode using a slower clock and hence less power (see Section 2.5). Depending on the state of the WSLOW bit in PLLCR, the internal clocks derived from 4XCLK are divided down by a factor of eight. Therefore the power consumption in WAIT is dependent on the particular application.

5.5.2 STOP

Executing the STOP instruction while the S-bit in the CCR is clear places the MCU in the STOP condition. If the S-bit is set, the STOP opcode is treated as a no-op (NOP). The STOP condition offers minimum power consumption because all clocks, including the crystal oscillator, are stopped while in this mode. To exit STOP and resume normal processing, a logic low level must be applied to one of the external interrupts ($\overline{IRQ}$ or $\overline{XIRQ}$) or to the $\overline{RESET}$ pin. A pending edge-triggered $\overline{IRQ}$ can also bring the CPU out of STOP.

Because all clocks are stopped in this mode, all internal peripheral functions also stop. The data in the internal RAM is retained as long as V_{DD} power is maintained. The CPU state and I/O pin levels are static and are unchanged by STOP. Therefore, when an interrupt comes to restart the system, the MCU resumes processing as if there were no interruption. If reset is used to restart the system a normal reset sequence results where all I/O pins and functions are also restored to their initial states.

To use the $\overline{IRQ}$ pin as a means of recovering from STOP, the I-bit in the CCR must be clear ($\overline{IRQ}$ not masked). The $\overline{XIRQ}$ pin can be used to wake up the MCU from STOP regardless of the state of the X-bit in the CCR, although the recovery sequence depends on the state of the X-bit. If X is clear ($\overline{XIRQ}$ not masked), the MCU starts up, beginning with the stacking sequence leading to normal service of the $\overline{XIRQ}$ request. If X is set ($\overline{XIRQ}$ masked or inhibited), then processing continues with the instruction that immediately follows the STOP instruction, and no $\overline{XIRQ}$ interrupt service is requested or pending.

Because the oscillator is stopped in STOP mode, a restart delay may be imposed to allow oscillator stabilization upon leaving STOP. If the internal oscillator is being used, this delay is required; however, if a stable external oscillator is being used, the DLY control bit can be used to bypass this start-up delay. The DLY control bit is set by reset and can be optionally cleared during initialization. If the DLY equal to zero option is used to avoid start-up delay on recovery from STOP, then reset should not be used as the means of recovering from STOP, as this causes DLY to be set again by reset, imposing the restart delay. This same delay also applies to power-on-reset, regardless of the state of the DLY control bit, but does not apply to a reset while the clocks are running. See Section 4.3.2.4.

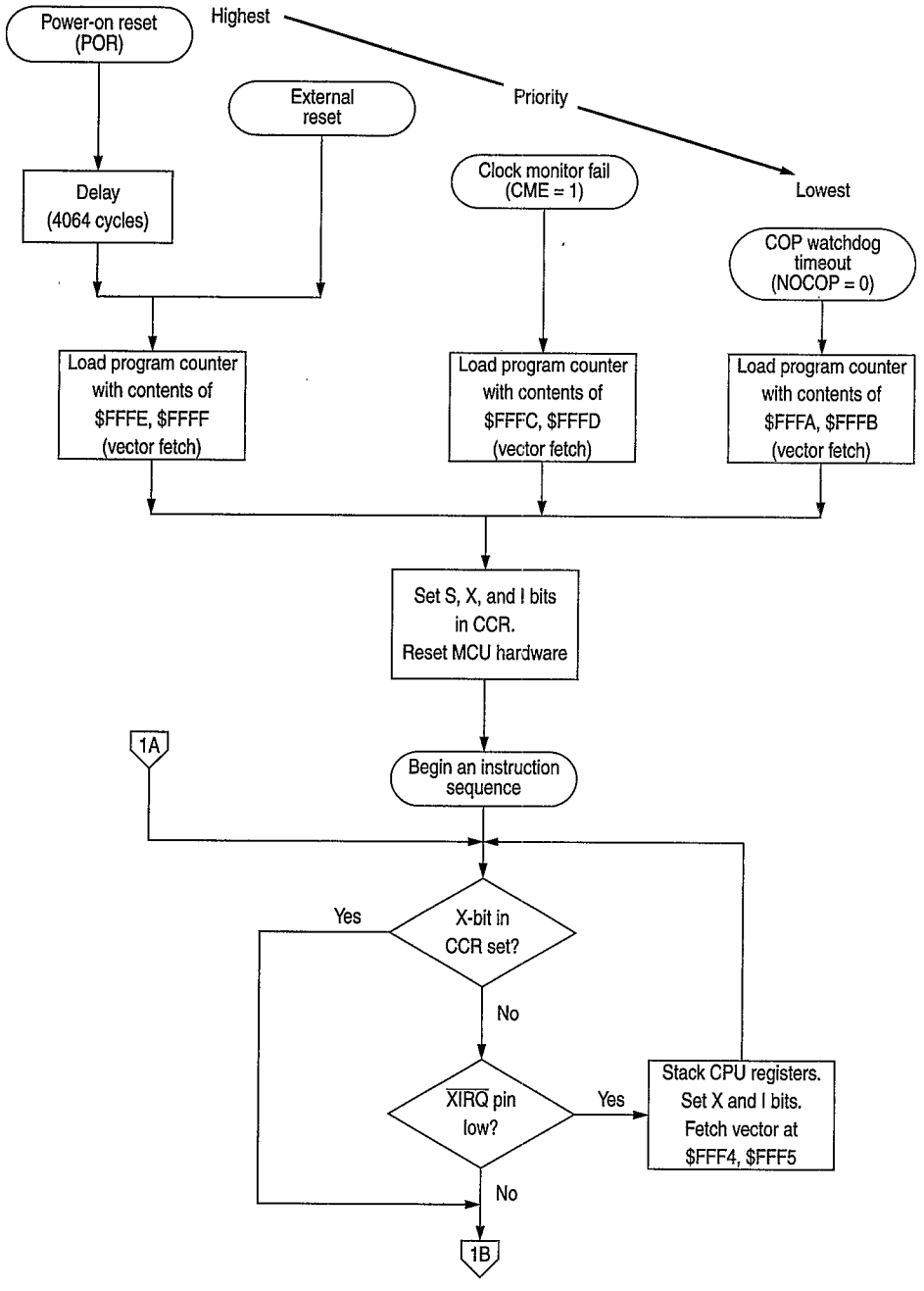


Figure 5-3 Processing flow out of reset (1 of 2)

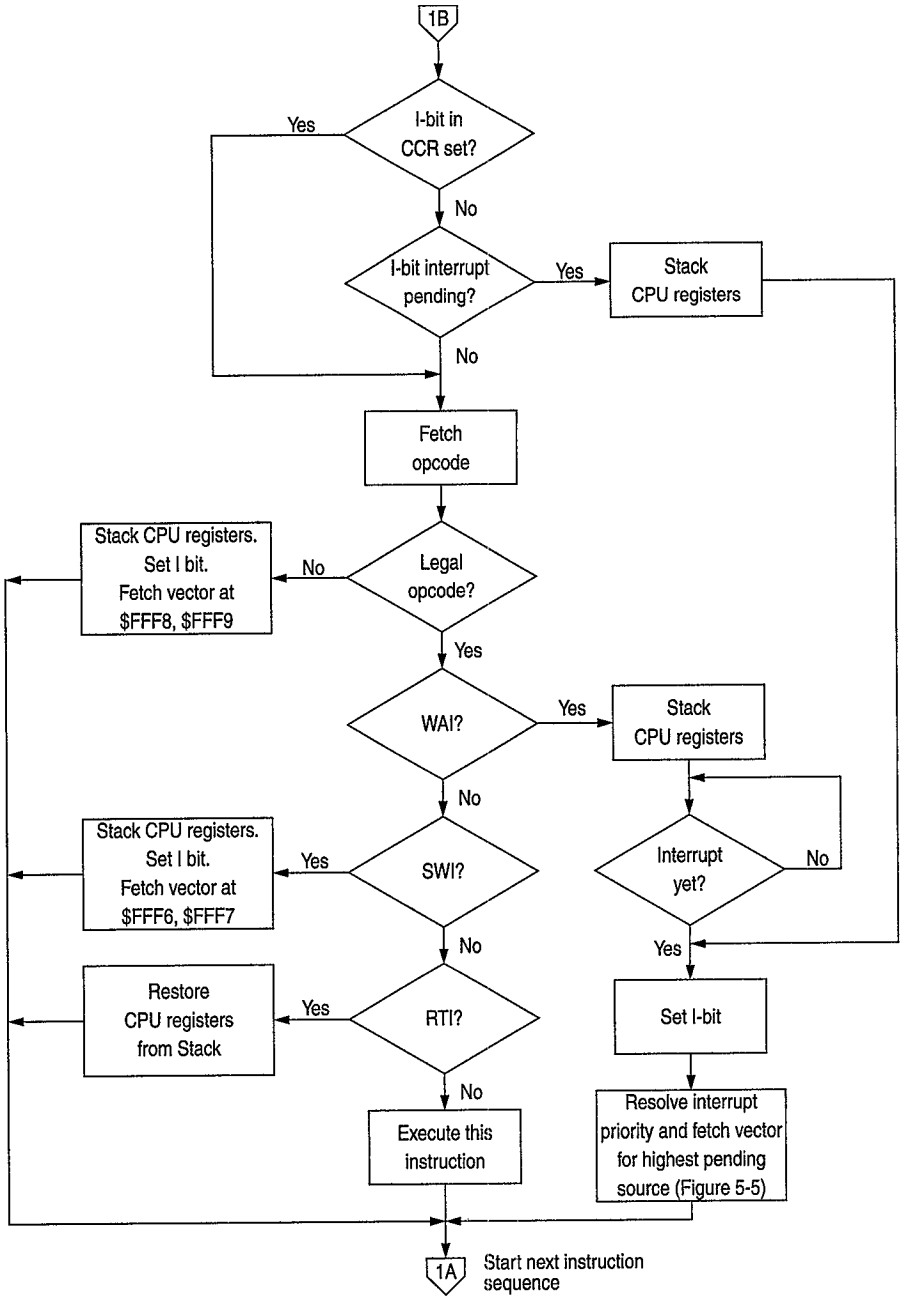


Figure 5-4 Processing flow out of reset (2 of 2)

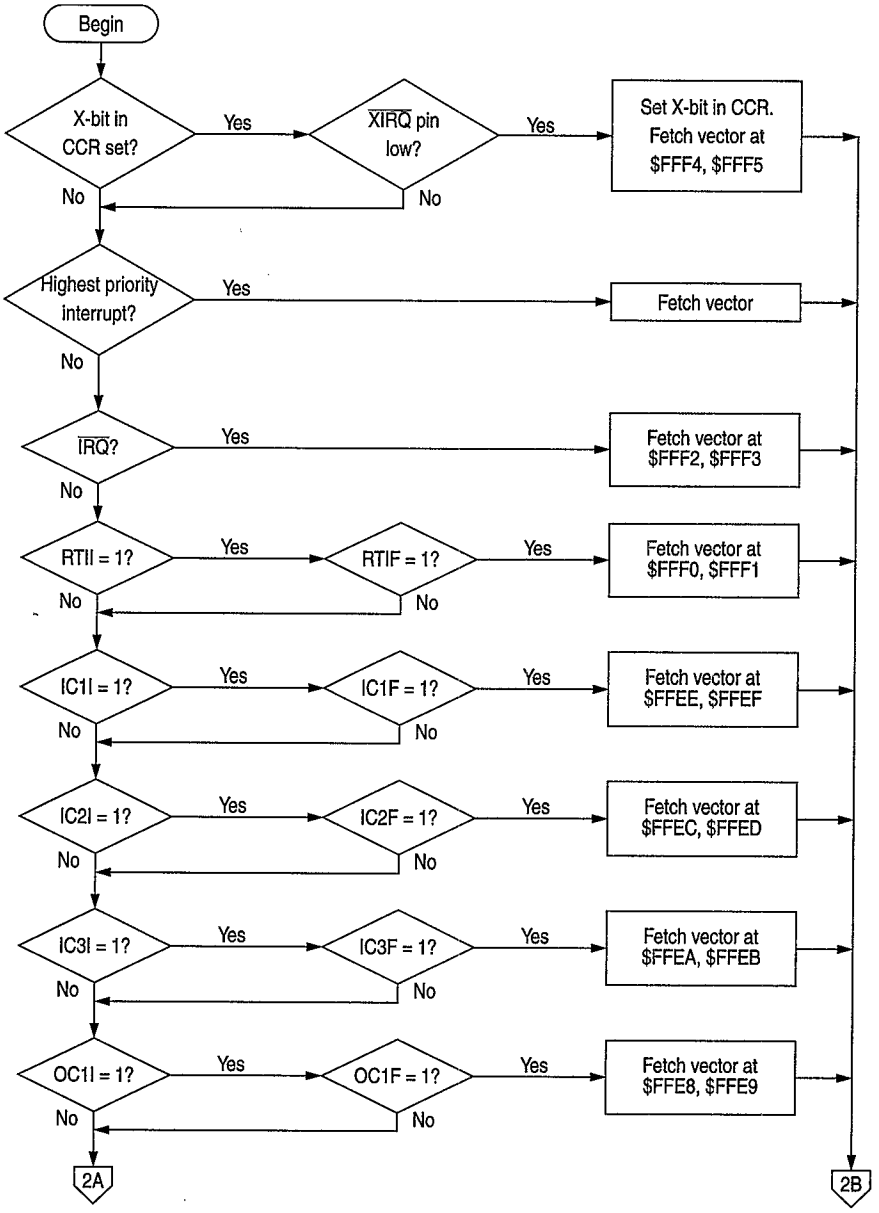


Figure 5-5 Interrupt priority resolution (1 of 3)

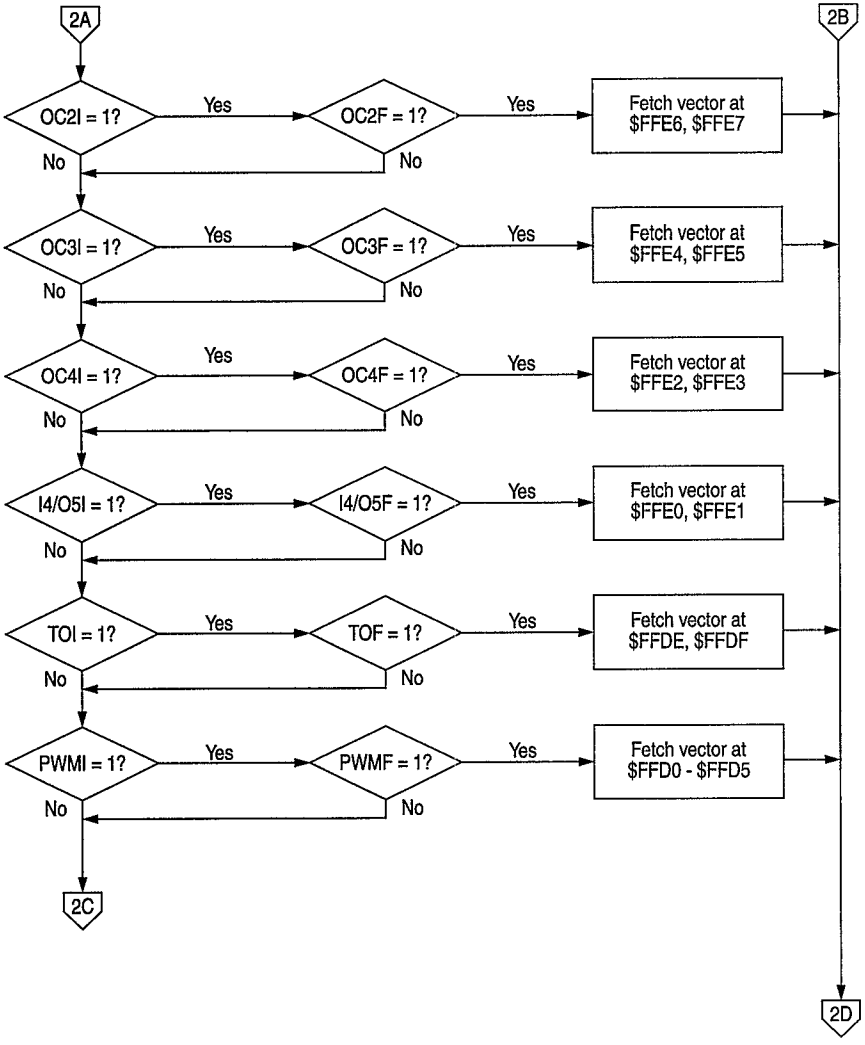
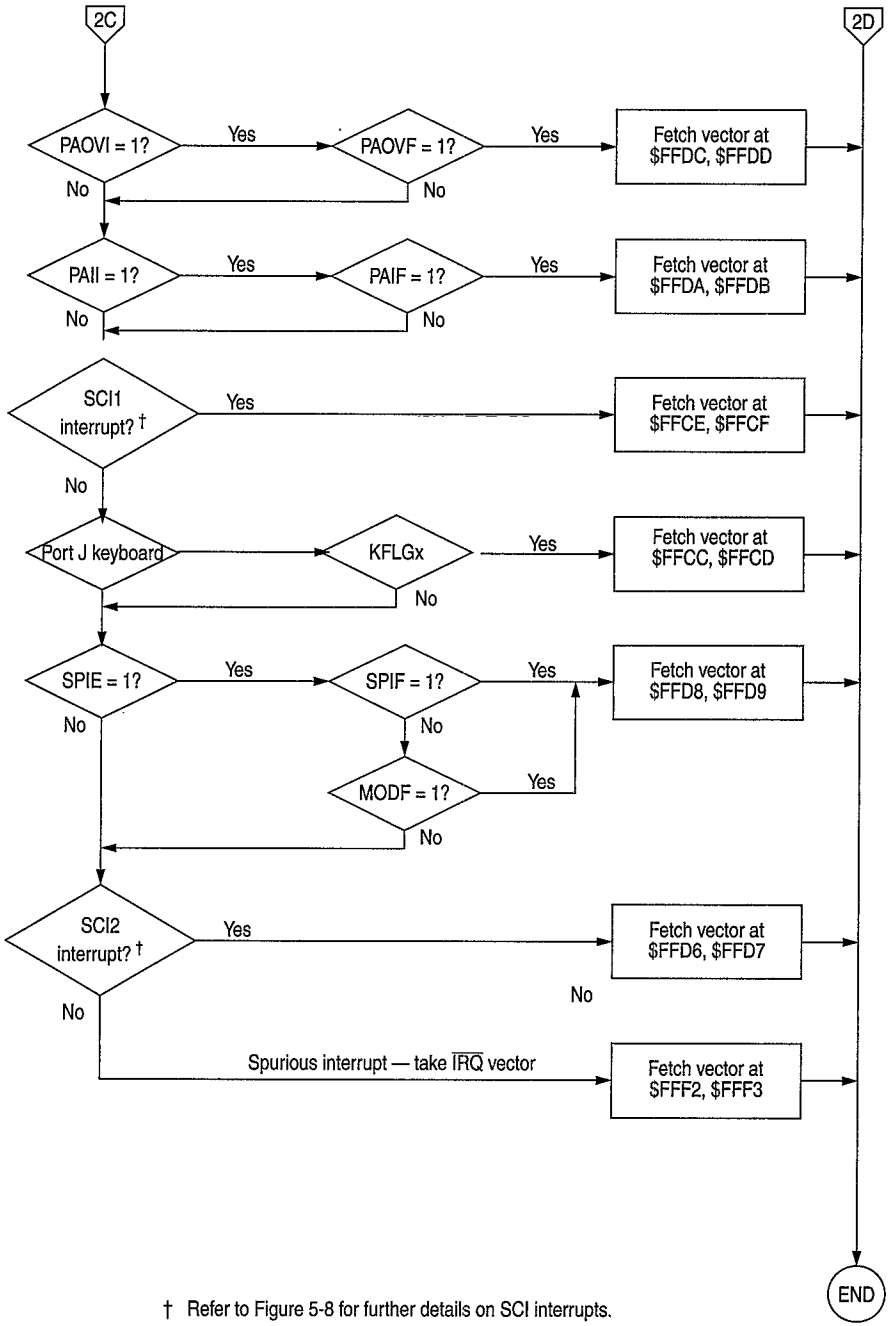


Figure 5-6 Interrupt priority resolution (2 of 3)



† Refer to Figure 5-8 for further details on SCI interrupts.

Figure 5-7 Interrupt priority resolution (3 of 3)

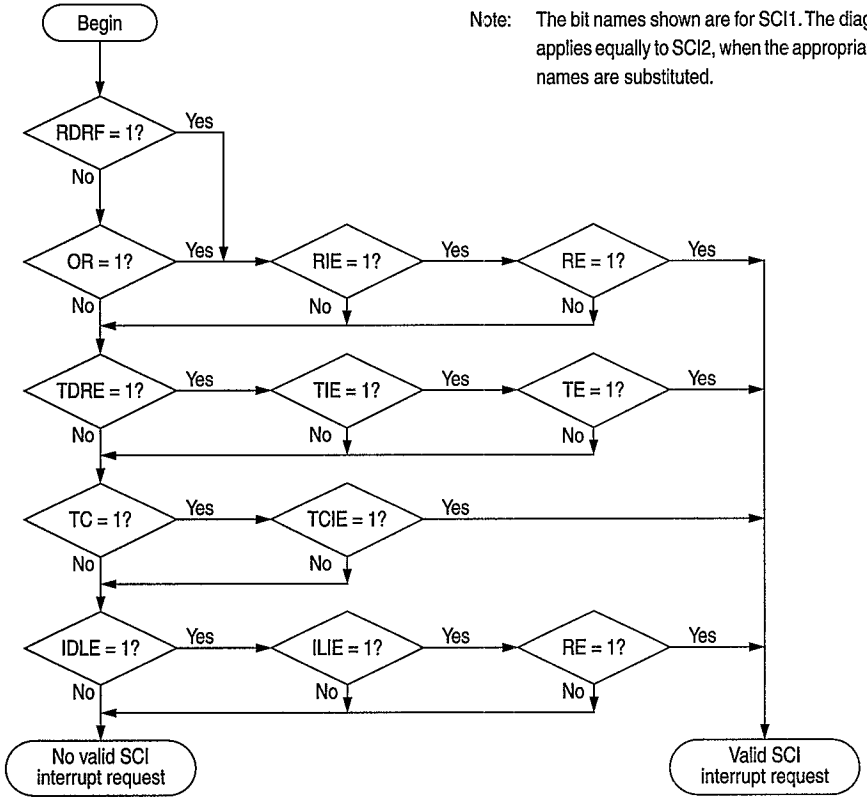


Figure 5-8 Interrupt source resolution within the SCI subsystem

THIS PAGE LEFT BLANK INTENTIONALLY

6
PARALLEL INPUT/OUTPUT

The MC68HC11KG4 has up to 44 input/output lines, 7 input-only lines and 17 output lines, depending on the operating mode. To enhance the I/O functions, the data bus of this microcontroller is non-multiplexed. The following table is a summary of the configuration and features of each port.

Table 6-1 Port configuration

Port	Input pins	Output pins	Bidirectional pins	Alternative functions
A	—	—	8	Timer
B	—	—	8	High order address
C	—	—	8	Data bus
D	—	—	8	SCI1, SCI2 and SPI
E	7	(1)	1(1)	A/D converter/ D/A converter(1)
F	—	—	8	Low order address
H	—	8	—	PWM
J	—	—	4	Keyboard interrupt
K	—	8	—	PWM

(1) Pin PE7 becomes an output when the D/A converter is in operation.

Note: Do not confuse pin function with the electrical state of that pin at reset. All general-purpose I/O pins that are configured as inputs at reset are in a high-impedance state and the contents of the port data registers are undefined; in port descriptions, a 'u' indicates this condition. The pin function is mode dependent.

6.1 Port A

Port A is an 8-bit bidirectional port, with both data and data direction registers. In addition to their I/O capability, port A pins are shared with timer functions, as shown in the following table.

Pin	Alternative function
PA0	IC3
PA1	IC2
PA2	IC1
PA3	OC5 and/or OC1, or IC4
PA4	OC4 and/or OC1
PA5	OC3 and/or OC1
PA6	OC2 and/or OC1
PA7	PAI and/or OC1

See Section 10 for more information.

6

On reset the pins are configured as general purpose high-impedance inputs.

6.1.1 PORTA — Port A data register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Port A data (PORTA)	$0000	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	undefined

This is a read/write register and is not affected by reset. The bits may be read and written at any time, but, when a pin is allocated to its alternative function, a write to the corresponding register bit has no effect on the pin state.

6.1.2 DDRA — Data direction register for port A

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Data direction A (DDRA)	$0001	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0	0000 0000

DDA[7:0] — Data direction for port A

1 (set) – The corresponding pin is configured as an output.

0 (clear) – The corresponding pin is configured as an input.

6.2 Port B

Port B is an 8-bit bidirectional port, with both data and data direction registers. In addition to their I/O capability, port B pins are used as the non-multiplexed high order address pins, as shown in the following table.

Pin	Alternative function
PB0	ADDR8
PB1	ADDR9
PB2	ADDR10
PB3	ADDR11
PB4	ADDR12
PB5	ADDR13
PB6	ADDR14
PB7	ADDR15

In expanded or test mode, the pins become the high order address lines and port B is not included in the memory map.

6

The state of the pins on reset is mode dependent. In single chip or bootstrap mode, port B pins are high-impedance inputs with selectable internal pull-up resistors (see Section 6.10). In expanded or test mode, port B pins are high order address outputs and PORTB/DDRB are not in the memory map.

6.2.1 PORTB — Port B data register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Port B data (PORTB)	$0004	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	undefined

The bits may be read and written at any time and are not affected by reset.

6.2.2 DDRB — Data direction register for port B

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Data direction B (DDRB)	$0002	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	0000 0000

DDB[7:0] — Data direction for port B

1 (set) – The corresponding pin is configured as an output.

0 (clear) – The corresponding pin is configured as an input.

6.3 Port C

Port C is an 8-bit bidirectional port, with both data and data direction registers. In addition to their I/O capability, port C pins are used as the non-multiplexed data bus pins, as shown in the following table.

Pin	Alternative function
PC0	DATA0
PC1	DATA1
PC2	DATA2
PC3	DATA3
PC4	DATA4
PC5	DATA5
PC6	DATA6
PC7	DATA7

In expanded or test mode, the pins become the data bus and port C is not included in the memory map.

6

The state of the pins on reset is mode dependent. In single chip or bootstrap mode, port C pins are high-impedance inputs. In expanded or test modes, port C pins are the data bus I/O and PORTC/DDRC are not in the memory map.

6.3.1 PORTC — Port C data register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Port C data (PORTC)	$0006	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	undefined

The bits may be read and written at any time and are not affected by reset.

6.3.2 DDRC — Data direction register for port C

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Data direction C (DDRC)	$0007	DDC7	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	0000 0000

DDC[7:0] — Data direction for port C

1 (set) – The corresponding pin is configured as an output.

0 (clear) – The corresponding pin is configured as an input.

6.4 Port D

Port D is a 8-bit bidirectional port, with both data and data direction registers. In addition to their I/O capability, port D pins are shared with SCI and SPI functions, as shown in the following table.

Pin	Alternative function	
PD0	RXD2	See Section 10 for more information.
PD1	TXD2	
PD2	MISO	See Section 8 for more information.
PD3	MOSI	
PD4	SCK	
PD5	$\overline{SS}$	
PD6	RXD1	See Section 8 for more information.
PD7	TXD1	

On reset the pins are configured as general purpose high-impedance inputs.

6.4.1 PORTD — Port D data register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Port D data (PORTD)	$0008	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0	undefined

This is a read/write register and is not affected by reset. The bits may be read and written at any time, but, when a pin is allocated to an alternative function, a write to the corresponding register bit has no effect on the pin state.

6.4.2 DDRD — Data direction register for port D

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Data direction D (DDRD)	$0009	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	0000 0000

DDD[7:0] — Data direction for port D

1 (set) – The corresponding pin is configured as an output.

0 (clear) – The corresponding pin is configured as an input.

6.5 Port E

PE[6:0] are input-only pins which are also shared with A/D functions, as shown in the following table. PE7 is normally an input, but is forced to be an output when the D/A converter is in use

Pin	Alternative function
PE0	AN2
PE1	AN3
PE2	AN4
PE3	AN5
PE4	AN6
PE5	AN7
PE6	AN8
PE7	DA1

See Section 10 and Section 11 for more information.

On reset, the pins are configured as general purpose high-impedance inputs.

6.5.1 PORTE — Port E data register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Port E data (PORTE)	$000A	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0	undefined

This is a read-only register and is not affected by reset. The bits may be read at any time.

Note: As port E shares pins with the A/D converter, a read of this register may affect any conversion currently in progress, if it coincides with the sample portion of the conversion cycle. Hence, normally port E should not be read during the sample portion of any conversion.

6.6 Port F

Port F is an 8-bit bidirectional port, with both data and data direction registers. In addition to their I/O capability, port F pins are used as the non-multiplexed low order address pins, as shown in the following table.

Pin	Alternative function
PF0	ADDR0
PF1	ADDR1
PF2	ADDR2
PF3	ADDR3
PF4	ADDR4
PF5	ADDR5
PF6	ADDR6
PF7	ADDR7

In expanded or test mode, the pins become the low order address and port F is not included in the memory map.

6

The state of the pins on reset is mode dependent. In single chip or bootstrap mode, port F pins are high-impedance inputs with selectable internal pull-up resistors (see Section 6.10). In expanded or test modes, port F pins are low order address outputs and PORTF/DDRF are not in the memory map.

6.6.1 PORTF — Port F data register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Port F data (PORTF)	$0005	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0	undefined

The bits may be read and written at any time and are not affected by reset.

6.6.2 DDRF — Data direction register for port F

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Data direction F (DDRF)	$0003	DDF7	DDF6	DDF5	DDF4	DDF3	DDF2	DDF1	DDF0	0000 0000

DDF[7:0] — Data direction for port F

1 (set) – The corresponding pin is configured as an output.

0 (clear) – The corresponding pin is configured as an input.

6.7 Port H

Port H is an 8-bit general purpose output port (GPO) with a data register. Port H pins are also used as PWM channels when selected by the corresponding bit(s) in the PWEN channel enable register, as shown in the following table.

Pin	Alternative function
PH0	PWM0
PH1	PWM0
PH2	PWM1
PH3	PWM1
PH4	PWM2
PH5	PWM2
PH6	PWM3
PH7	PWM3

See Section 9 for further information.

6

6.7.1 PORTH — Port H data register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Port H data (PORTH)	$005C	PH7	PH6	PH5	PH4	PH3	PH2	PH1	PH0	0000 0000

The bits may be read and written at any time; outputs return the port H data latch level, and are driven low on reset.

6.7.2 PTHMS — Port H mismatch register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Port H mismatch (PTHMS)	$005D	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(0)	0000 0000

A logic one at one of the register bits of the mismatch register indicates a short circuit at the corresponding port H bit.

6.8 Port J

Port J is a 4-bit general purpose I/O port with a data register (PORTJ) and a data direction register (DDRJ). The four port J pins, when configured as inputs, can be configured to provide a keyboard interrupt function. Each of the four port J lines has its own flag and mask bit as well as the usual data and data direction register bits. Each bit has a pullup transistor which can be enabled when that bit is operating as an input. The pullups can be globally disabled by writing to the JPPUE bit in the port pullup assignment register (PPAR). Reading from bits that are assigned as outputs reads the value back from the data register. The keyboard interrupt function can only be enabled for bits that are configured as inputs.

Pin	Alternative function
PJ0	KI1
PJ1	KI2
PJ2	KI3
PJ3	KI4

6.8.1 PORTJ — Port J data register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Port J data (PORTJ)	$005A	0	0	0	0	PJ3	PJ2	PJ1	PJ0	undefined

This is a read/write register and is not affected by reset. The bits may be read and written at any time, but, when a pin is allocated to an alternative function, a write to the corresponding register bit has no effect on the pin state.

6.8.2 DDRJ — Data direction register for port J

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Data direction J (DDRJ)	$005B	0	0	0	0	DDJ3	DDJ2	DDJ1	DDJ0	0000 0000

DDJ[3:0] — Data direction for port J

1 (set) – The corresponding pin is configured as an output.

0 (clear) – The corresponding pin is configured as an input.

6.8.3 JKFLAG — Port J keyboard interrupt flag register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Port J keyboard interrupt flag (JKFLAG)	$0058	0	0	0	0	KFLG3	KFLG2	KFLG1	KFLG0	0000 0000

This register contains the keyboard interrupt flags corresponding to port J bits [0:3] and can be read at any time. Bits in JKFLAG become set when an active edge is detected on the corresponding Port J pin. The active edge is selected by the KEDG[3:0] bits in JKMASK.

6.8.4 JKMASK — Port J keyboard interrupt mask register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Port J keyboard interrupt mask (JKMASK)	$0059	KEDG3	KEDG2	KEDG1	KEDG0	KMSK3	KMSK2	KMSK1	KMSK0	0000 0000

This register can be read at any time. Bits [0:3] can be written at any time, but bits [4:7] may be written only once after each reset, after which they become read-only. If SMOD = 1 however, writes to these bits are always permitted.

KEDG[3:0] — Keyboard interrupt active edge

These bits allow the port J interrupt active edges to be individually selected.

1 (set) – Interrupt on rising edges only.

0 (clear) – Interrupt on falling edges only.

KMSK[3:0] — Keyboard interrupt mask

These bits allow the port J inputs to be individually masked from the keyboard interrupt function.

1 (set) – Port J pin keyboard interrupt enabled.

0 (clear) – Port J pin keyboard interrupt disabled.

The port J keyboard interrupt function can be used to bring the MC68HC11KG4 out of STOP and WAIT mode. If the I-bit in the CCR is clear during STOP, then a port J keyboard interrupt initiates a jump to the keyboard interrupt vector. If however, the I-bit is set, then the interrupt vectors are disabled and the keyboard interrupt causes the device to continue with the next instruction following STOP.

A wakeup from WAIT results in a jump to the keyboard interrupt vector. The I-bit in the CCR needs to be cleared in order to receive the keyboard wakeup from WAIT.

6.9 Port K

Port K is an 8-bit general purpose output port with a data register. Port K pins are also used as PWM channels when selected by the corresponding bit(s) in the PWEN channel enable register. This is shown in the following table.

Pin	Alternative function	
PK0	PWM4	See Section 9 for further information.
PK1	PWM4	
PK2	PWM5	
PK3	PWM5	
PK4	PWM6	
PK5	PWM6	
PK6	PWM7	
PK7	PWM7	

6.9.1 PORTK — Port K data register .

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Port K data (PORTK)	$005E	PK7	PK6	PK5	PK4	PK3	PK2	PK1	PK0	0000 0000

The bits may be read and written at any time; outputs return the port K data latch level, and are driven low on reset.

6.9.2 PTKMS — Port K mismatch register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Port K mismatch (PTKMS)	$005F	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(0)	0000 0000

A logic one at one of the register bits of the mismatch register indicates a short circuit at the corresponding port K bit.

6.10 Internal pull-up resistors

Three of the ports (B, F and J) have internal, software selectable pull-up resistors under control of the port pull-up assignment register (PPAR).

6.10.1 PPAR — Port pull-up assignment register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Port pull-up assignment (PPAR)	$002C	0	0	0	0	JPPUE	0	FPPUE	BPPUE	0000 1011

Bits [7:4] — Not implemented; always read zero.

Bit 2 — Not implemented; always reads zero.

xPPUE — Port x pin pull-up enable

These bits control the on-chip pull-up devices connected to all the pins on I/O ports B, F and J. They are collectively enabled or disabled via the PAREN bit in the CONFIG register (see Section 6.11.2).

1 (set) – Port x pin on-chip pull-up devices enabled.

0 (clear) – Port x pin on-chip pull-up devices disabled.

Note: FPPUE and BPPUE have no effect in expanded mode since then ports F and B are dedicated address bus outputs.

6.11 System configuration

One bit in each of the following registers is directly concerned with the configuration of the I/O ports. For full details on the other bits in the registers, refer to the appropriate section.

6.11.1 OPT2 — System configuration options register 2

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
System config. options 2 (OPT2)	$0038	LIRDV	CWOM	STRCH	IRVNE	LSBF	SPR2	XDV1	XDV0	000x 0000

LIRDV — LIR driven (refer to Section 4)

1 (set) – Enable LIR drive high pulse.

0 (clear) – LIR not driven on MODA/$\overline{\text{LIR}}$ pin.

CWOM — Port C wired-OR mode

1 (set) – Port C outputs are open-drain.

0 (clear) – Port C operates normally.

STRCH — Stretch external accesses (refer to Section 4)

1 (set) – Off-chip accesses are extended by one E clock cycle.

0 (clear) – Normal operation.

IRVNE — Internal read visibility/not E (refer to Section 4)

1 (set) – Data from internal reads is driven out of the external data bus.

0 (clear) – No visibility of internal reads on external bus.

In **single chip mode** this bit determines whether the E clock drives out from the chip.

1 (set) – E pin is driven low.

0 (clear) – E clock is driven out from the chip.

LSBF — LSB first enable (refer to Section 9)

1 (set) – SPI data is transferred LSB first.

0 (clear) – SPI data is transferred MSB first.

SPR2 — SPI clock rate select (refer to Section 9)

XDV[1, 0] — XOUT clock divide select (refer to Section 4)

These two bits control the frequency of the XCLK signal, which is output on the XOUT pin if enabled by the CLKX bit in CONFIG.

6.11.2 CONFIG — System configuration register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Configuration control (CONFIG)	$003F	ROMAD	NWCOP	CLKX	PAREN	NOSEC	NOCOP	ROMON	EEON	xxxx xxxx

ROMAD — ROM/*EPROM* mapping control (refer to Section 4)

1 (set) – ROM/*EPROM* addressed from $A000 to $FFFF.

0 (clear) – ROM/*EPROM* addressed from $2000 to $7FFF (expanded mode only).

NWCOP — WCOP system disable (refer to Section 5)

1 (set) – WCOP system disabled.

0 (clear) – WCOP system enabled (forces reset on timeout).

CLKX — XOUT enable (refer to Section 4)

1 (set) – XCLK signal driven out on the XOUT pin

0 (clear) – XOUT pin disabled.

The frequency of the XCLK signal is controlled by two bits in the OPT2 register.

PAREN — Pull-up assignment register enable

1 (set) – Pull-ups can be enabled using PPAR register.

0 (clear) – All pull-ups disabled (not controlled by PPAR).

6

NOSEC — EEPROM security disabled (refer to Section 4)

1 (set) – Disable security.

0 (clear) – Enable security.

NOCOP — COP system disable (refer to Section 5)

1 (set) – COP system disabled.

0 (clear) – COP system enabled (forces reset on timeout).

ROMON — ROM/*EPROM* enable (refer to Section 4)

1 (set) – ROM/*EPROM* included in the memory map.

0 (clear) – ROM/*EPROM* excluded from the memory map.

In single chip mode, reset sets this bit. In special test mode, reset clears ROMON.

EEON — EEPROM enable (refer to Section 4)

1 (set) – EEPROM included in the memory map.

0 (clear) – EEPROM excluded from the memory map.

7
SERIAL COMMUNICATIONS INTERFACE†

The serial communications interface (SCI) is a universal asynchronous receiver transmitter (UART). It has a non-return to zero (NRZ) format (one start, eight or nine data, and one stop bit) that is compatible with standard RS-232 systems.

The SCI shares I/O with two of port D's pins:

Pin	Alternative function
PD6	RXD1
PD7	TXD1

The SCI transmit and receive functions are enabled by TE and RE respectively, in SC1CR2.

The SCI features enabled on this MCU include: 13-bit modulus prescaler, idle line detect, receiver-active flag, transmitter and receiver hardware parity. A block diagram of the enhanced baud rate generator is shown in Figure 7-1. See Table 7-1 for example baud rate control values.

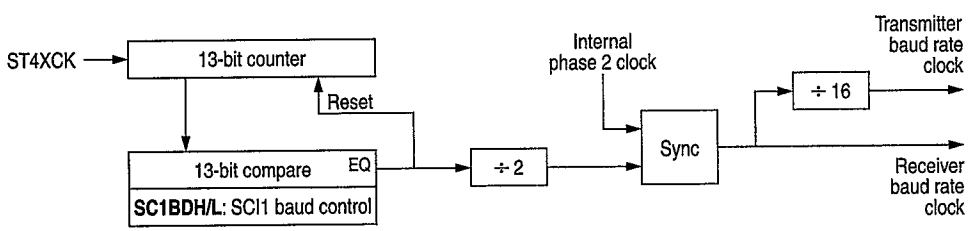


Figure 7-1 SCI1 baud rate generator circuit diagram

† The MC68HC11KG4 contains two serial communications interfaces, both having similar operation. For ease of reference, a full description of SCI1 (PD6/RXD1, PD7/TXD1) is given first, followed by a summary of SCI2 (Section 7.8), detailing its differences.

7.1 Data format

The serial data format requires the following conditions:

- An idle-line condition before transmission or reception of a message.
- A start bit, logic zero, transmitted or received, that indicates the start of each character.
- Data that is transmitted and received least significant bit (LSB) first.
- A stop bit, logic one, used to indicate the end of a frame. (A frame consists of a start bit, a character of eight or nine data bits, and a stop bit.)
- A break (defined as the transmission or reception of a logic zero for some multiple number of frames).

Selection of the word length is controlled by the M bit of SC1CR1.

7.2 Transmit operation

The SCI transmitter includes a parallel data register (SC1DRH/SC1DRL) and a serial shift register. The contents of the shift register can only be written through the parallel data register. This double buffered operation allows a character to be shifted out serially while another character is waiting in the parallel data register to be transferred into the shift register. The output of the shift register is applied to TXD as long as transmission is in progress or the transmit enable (TE) bit of serial communication control register 2 (SC1CR2) is set. The block diagram, Figure 7-2, shows the transmit serial shift register and the buffer logic at the top of the figure.

7.3 Receive operation

During receive operations, the transmit sequence is reversed. The serial shift register receives data and transfers it to the parallel receive data registers (SC1DRH/SC1DRL) as a complete word. This double buffered operation allows a character to be shifted in serially while another character is still in the serial data registers. An advanced data recovery scheme distinguishes valid data from noise in the serial data stream. The data input is selectively sampled to detect receive data, and majority sampling logic determines the value and integrity of each bit.

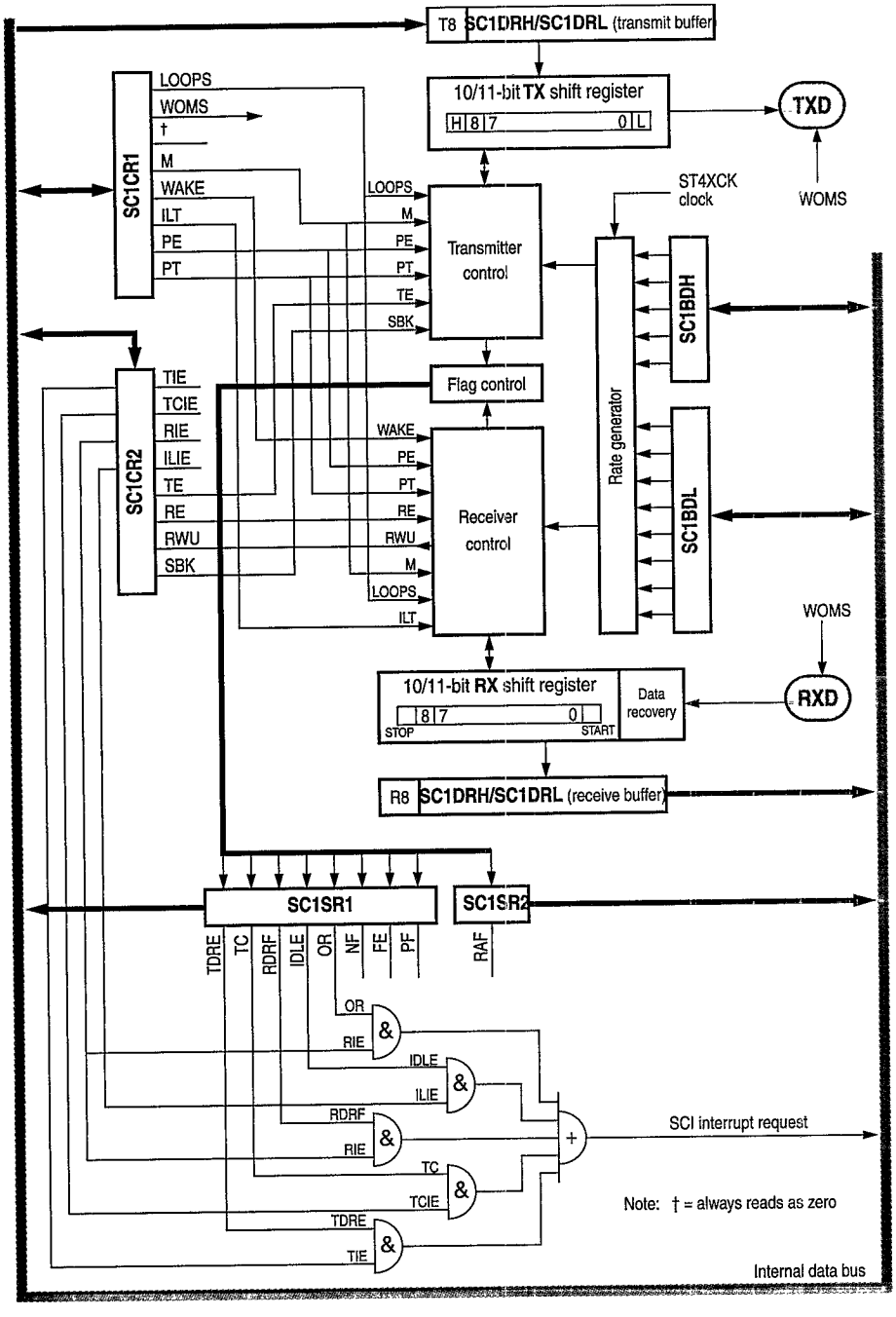


Figure 7-2 SCI1 block diagram

7.4 Wake-up feature

The wake-up feature reduces SCI service overhead in multiple receiver systems. Software for each receiver evaluates the first character or frame of each message. All receivers are placed in wake-up mode by writing a one to the RWU bit in the SC1CR2 register. When RWU is set, the receiver-related status flags (RDRF, IDLE, OR, NF, FE, and PF) are inhibited (cannot be set). Although RWU can be cleared by a software write to SC1CR2, to do so would be unusual. Normally RWU is set by software and is cleared automatically with hardware. Whenever a new message begins, logic alerts the dormant receivers to wake up and evaluate the initial character of the new message.

Two methods of wake-up are available: idle-line wake-up and address mark wake-up. During idle-line wake-up, a dormant receiver activates as soon as the RXD line becomes idle. In the address mark wake-up, logic one in the most significant bit (MSB) of a character activates all sleeping receivers. To use either receiver wake-up method, establish a software addressing scheme to allow the transmitting devices to direct messages to individual receivers or to groups of receivers. This addressing scheme can take any form as long as all transmitting and receiving devices are programmed to understand the same scheme.

7.4.1 Idle-line wake-up

Clearing the WAKE bit in SC1CR1 register enables idle-line wake-up mode. In idle-line wake-up mode, all receivers are active (RWU bit in SC1CR2 = 0) when each message begins. The first frames of each message are addressing frames. Each receiver in the system evaluates the addressing frames of a message to determine if the message is intended for that receiver. When a receiver finds that the message is not intended for it, it sets the RWU bit. Once set, the RWU control bit disables all but the necessary receivers for the remainder of the message, thus reducing software overhead for the remainder of that message. As soon as an idle line is detected by receiver logic, hardware automatically clears the RWU bit so that the first frames of the next message can be evaluated by all receivers in the system. This type of receiver wake-up requires a minimum of one idle frame time between messages, and no idle time between frames within a message.

7.4.2 Address-mark wake-up

Setting the WAKE bit in SC1CR1 register enables address-mark wake-up mode. The address-mark wake-up method uses the MSB of each frame to differentiate between address information (MSB = 1) and actual message data (MSB = 0). All frames consist of seven information bits (eight bits if M bit in SC1CR1 = 1) and an MSB which, when set to one, indicates an address frame. The first frames of each message are addressing frames. Receiver logic evaluates these marked frames to determine the receivers for which that message is intended. When a receiver finds that the message is not intended for it, it sets the RWU bit. Once set, the RWU control bit disables all but the necessary receivers for the remainder of the message, thus reducing software

overhead for the remainder of that message. When the next message begins, its first frame will have the MSB set which will automatically clear the RWU bit and indicate that this is an addressing frame. This frame is always the first frame received after wake-up because the RWU bit is cleared before the stop bit for the first frame is received. This method of wake-up allows messages to include idle times, however, there is a loss in efficiency due to the extra bit time required for the address bit in each frame.

7.5 SCI error detection

Four error conditions can occur during SCI1 operation. These error conditions are: serial data register overrun, received bit noise, framing, and parity error. Four bits (OR, NF, FE, and PF) in serial communications status register 1 (SC1SR1) indicate if one of these error conditions exists.

The overrun error (OR) bit is set when the next byte is ready to be transferred from the receive shift register to the serial data registers (SC1DRH/SC1DRL) and the registers are already full (RDRF bit is set). When an overrun error occurs, the data that caused the overrun is lost and the data that was already in serial data registers is not disturbed. The OR is cleared when the SC1SR is read (with OR set), followed by a read of the SCI data registers.

The noise flag (NF) bit is set if there is noise on any of the received bits, including the start and stop bits. The NF bit is not set until the RDRF flag is set. The NF bit is cleared when the SC1SR is read (with FE equal to one) followed by a read of the SCI1 data registers.

When no stop bit is detected in the received data character, the framing error (FE) bit is set. FE is set at the same time as the RDRF. If the byte received causes both framing and overrun errors, the processor only recognizes the overrun error. The framing error flag inhibits further transfer of data into the SCI1 data registers until it is cleared. The FE bit is cleared when the SC1SR is read (with FE equal to one) followed by a read of the SCI1 data registers.

The parity error flag (PF) is set if received data has incorrect parity. The flag is cleared by a read of SC1SR1 with PE set, followed by a read of SC1DR.

7.6 SCI registers

There are eight addressable registers in the SCI1. SC1BDH, SC1BDL, SC1CR1, and SC1CR2 are control registers. The contents of these registers control functions and indicate conditions within the SCI1. The status registers SC1SR1 and SC1SR2 contain bits that indicate certain conditions within the SCI1. SC1DRH and SC1DRL are SCI data registers. These double buffered registers are used for the transmission and reception of data, and are used to form the 9-bit data word for the SCI1. If the SCI1 is being used with 7 or 8-bit data, only SC1DRL needs to be accessed. Note that if 9-bit data format is used, the upper register should be written first to ensure that it is transferred to the transmitter shift register with the lower register.

7.6.1 SC1BDH, SC1BDL — SCI1 baud rate control registers

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
SCI1 baud rate high (SC1BDH)	$0050	BTST	BSPL	SYNC	SBR12	SBR11	SBR10	SBR9	SBR8	0000 0000
SCI1 baud rate low (SC1BDL)	$0051	SBR7	SBR6	SBR5	SBR4	SBR3	SBR2	SBR1	SBR0	0000 0100

The contents of this register determine the baud rate of the SCI1.

BTST — Baud register test (Test mode only)

BSPL — Baud rate counter split (Test mode only)

SYNC — Baud rate counter synchronization (Test mode only)

SBR[12:0] — SCI1 baud rate selects

Use the following formula to calculate SCI1 baud rate. Refer to the table of baud rate control values for example rates:

$$\text{SCI1 baud rate} = \frac{\text{ST4XCK}}{16 \times (2\text{BR})}$$

where the baud rate control value (BR) is the contents of SC1BDH/L (BR = 1, 2, 3,... 8191).

For example, to obtain a baud rate of 1200 with an EXTAL frequency of 12MHz, the baud register (SC1BDH/L) should contain $0138 (see Table 7-1).

Table 7-1 Example SCI1 baud rate control values

Target baud rate	ST4XCK frequency					
	8 MHz		12 MHz		16 MHz	
	Dec value	Hex value	Dec value	Hex value	Dec value	Hex value
110	2272	$08E0	3409	$0D51	4545	$11C1
150	1666	$0682	2500	$09C4	3333	$0D05
300	833	$0341	1250	$04E2	1666	$0682
600	416	$01A0	625	$0271	833	$0341
1200	208	$00D0	312	$0138	416	$01A0
2400	104	$0068	156	$009C	208	$00D0
4800	52	$0034	78	$004E	104	$0068
9600	26	$001A	39	$0027	52	$0034
19200	13	$000D	20	$0014	26	$001A
38400					13	$000D

The clock rate generator is disabled if BR = 0, or if neither the receiver nor transmitter is enabled (both RE and TE in SC1CR2 are cleared).

Writes to the baud rate registers will only be successful if the last (or only) byte written is SC1BDL. The use of an STD instruction is recommended as it guarantees that the bytes are written in the correct order.

Note: ST4XCK may be the output of the PLL circuit or it may be the EXTAL input of the MCU (see Section 2.5 and Figure 10-1).

7.6.2 SC1CR1 — SCI1 control register 1

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
SCI1 control 1 (SC1CR1)	$0052	LOOPS	WOMS	0	M	WAKE	ILT	PE	PT	0000 0000

The SC1CR1 register provides the control bits that determine word length and select the method used for the wake-up feature.

LOOPS — SCI1 loop mode enable

1 (set) – SCI1 transmit and receive are disconnected from TXD and RXD pins, and transmitter output is fed back into the receiver input.

0 (clear) – SCI1 transmit and receive operate normally.

Both the transmitter and receiver must be enabled to use the LOOP mode. When the LOOP mode is enabled, the TXD pin is driven high (idle line state) if the transmitter is enabled.

WOMS — Wired-OR mode for SCI1 pins (PD7, PD6)

1 (set) – TXD and RXD are open drains if operating as outputs.

0 (clear) – TXD and RXD operate normally.

Bit 5 — Not implemented; always reads zero

M — Mode (select character format)

1 (set) – Start bit, 9 data bits, 1 stop bit.

0 (clear) – Start bit, 8 data bits, 1 stop bit.

WAKE — Wake-up by address mark/idle

1 (set) – Wake-up by address mark (most significant data bit set).

0 (clear) – Wake-up by IDLE line recognition.

ILT — Idle line type

1 (set) – Long (SCI1 counts ones only after stop bit).

0 (clear) – Short (SCI1 counts consecutive ones after start bit).

This bit determines which of two types of idle line detection method is used by the SCI1 receiver. In short mode the stop bit and any bits that were ones before the stop bit will be considered as part of that string of ones, possibly resulting in erroneous or premature detection of an idle line condition. In long mode the SCI1 system does not begin counting ones until a stop bit is received.

PE — Parity enable

1 (set) – Parity enabled.

0 (clear) – Parity disabled.

PT — Parity type

1 (set) – Parity odd (an odd number of ones causes parity bit to be zero, an even number of ones causes parity bit to be one).

0 (clear) – Parity even (an even number of ones causes parity bit to be zero, an odd number of ones causes parity bit to be one).

7

7.6.3 SC1CR2 — SCI1 control register 2

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
SCI1 control 2 (SC1CR2)	$0053	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK	0000 0000

The SC1CR2 register provides the control bits that enable or disable individual SCI1 functions.

TIE — Transmit interrupt enable

1 (set) – SCI1 interrupt requested when TDRE status flag is set.

0 (clear) – TDRE interrupts disabled.

TCIE — Transmit complete interrupt enable

1 (set) – SCI1 interrupt requested when TC status flag is set.

0 (clear) – TC interrupts disabled.

RIE — Receiver interrupt enable

1 (set) – SCI1 interrupt requested when RDRF flag or the OR status flag is set.

0 (clear) – RDRF and OR interrupts disabled.

ILIE — Idle line interrupt enable

1 (set) – SCI1 interrupt requested when IDLE status flag is set.

0 (clear) – IDLE interrupts disabled.

TE — Transmitter enable

1 (set) – Transmitter enabled.

0 (clear) – Transmitter disabled.

RE — Receiver enable

1 (set) – Receiver enabled.

0 (clear) – Receiver disabled.

RWU — Receiver wake-up control

1 (set) – Wake-up enabled and receiver interrupts inhibited.

0 (clear) – Normal SCI1 receiver.

SBK — Send break

1 (set) – Break codes generated as long as SBK is set.

0 (clear) – Break generator off.

7.6.4 SC1SR1 — SCI1 status register 1

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
SCI1 status 1 (SC1SR1)	$0054	TDRE	TC	RDRF	IDLE	OR	NF	FE	PF	1100 0000

The bits in SC1SR1 indicate certain conditions in the SCI1 hardware and are automatically cleared by special acknowledge sequences.

TDRE — Transmit data register empty flag

1 (set) – SC1DR empty.

0 (clear) – SC1DR busy.

This flag is set when SC1DR is empty. Clear the TDRE flag by reading SC1SR1 with TDRE set and then writing to SC1DR.

TC — Transmit complete flag

1 (set) – Transmitter idle.

0 (clear) – Transmitter busy.

This flag is set when the transmitter is idle (no data, preamble, or break transmission in progress). Clear the TC flag by reading SC1SR1 with TC set and then writing to SC1DR.

RDRF — Receive data register full flag

1 (set) – SC1DR full.

0 (clear) – SC1DR empty.

Once cleared, IDLE is not set again until the RXD line has been active and becomes idle again. RDRF is set if a received character is ready to be read from SC1DR. Clear the RDRF flag by reading SC1SR1 with RDRF set and then reading SC1DR.

IDLE — Idle line detected flag

1 (set) – RXD line is idle.

0 (clear) – RXD line is active.

This flag is set if the RXD line is idle. Once cleared, IDLE is not set again until the RXD line has been active and becomes idle again. The IDLE flag is inhibited when RWU = 1. Clear IDLE by reading SC1SR1 with IDLE set and then reading SC1DR.

OR — Overrun error flag

1 (set) – Overrun detected.

0 (clear) – No overrun.

OR is set if a new character is received before a previously received character is read from SC1DR. Clear the OR flag by reading SC1SR1 with OR set and then reading SC1DR.

NF — Noise error flag

1 (set) – Noise detected.

0 (clear) – Unanimous decision.

NF is set if the majority sample logic detects anything other than a unanimous decision. Clear NF by reading SC1SR1 with NF set and then reading SC1DR.

FE — Framing error

1 (set) – Zero detected.

0 (clear) – Stop bit detected.

FE is set when a zero is detected where a stop bit was expected. Clear the FE flag by reading SC1SR1 with FE set and then reading SC1DR.

PF — Parity error flag

1 (set) – Incorrect parity detected.

0 (clear) – Parity correct.

PF is set if received data has incorrect parity. Clear PF by reading SC1SR1 with PE set and then reading SC1DR.

7.6.5 SC1SR2 — SCI1 status register 2

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
SCI1 status 2 (SC1SR2)	$0055	0	0	0	0	0	0	0	RAF	0000 0000

In the SC1SR2 only bit 0 is used, to indicate receiver active. The other seven bits always read zero.

Bits [7:1] — Not implemented; always read zero

RAF — Receiver active flag (read only)

1 (set) – A character is being received.

0 (clear) – A character is not being received.

7.6.6 SC1DRH, SC1DRL — SCI1 data high/low registers

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
SCI1 data high (SC1DRH)	$0056	R8	T8	0	0	0	0	0	0	undefined
SCI1 data low (SC1DRL)	$0057	R7T7	R6T6	R5T5	R4T4	R3T3	R2T2	R1T1	R0T0	undefined

SC1DRH/SC1DRL is a parallel register that performs two functions. It is the receive data register when it is read, and the transmit data register when it is written. Reads access the receive data buffer and writes access the transmit data buffer. Data received or transmitted is double buffered.

If the SCI1 is being used with 7 or 8-bit data, only SC1DRL needs to be accessed. Note that if 9-bit data format is used, the upper register should be written first to ensure that it is transferred to the transmitter shift register with the lower register.

R8 — Receiver bit 8

Ninth serial data bit received when SCI1 is configured for a nine data bit operation

T8 — Transmitter bit 8

Ninth serial data bit transmitted when SCI1 is configured for a nine data bit operation

Bits [5:0] — Not implemented; always read zero

R/T[7:0] — Receiver/transmitter data bits [7:0]

SCI1 data is double buffered in both directions.

7.7 Status flags and interrupts

The SCI1 transmitter has two status flags. These status flags can be read by software (polled) to tell when certain conditions exist. Alternatively, a local interrupt enable bit can be set to enable each of these status conditions to generate interrupt requests. Status flags are automatically set by hardware logic conditions, but must be cleared by software. This provides an interlock mechanism that enables logic to know when software has noticed the status indication. The software clearing sequence for these flags is automatic — functions that are normally performed in response to the status flags also satisfy the conditions of the clearing sequence.

TDRE and TC flags are normally set when the transmitter is first enabled (TE set to one). The TDRE flag indicates there is room in the transmit queue to store another data character in the transmit data register. The TIE bit is the local interrupt mask for TDRE. When TIE is zero, TDRE must be polled. When TIE and TDRE are one, an interrupt is requested.

The TC flag indicates the transmitter has completed the queue. The TCIE bit is the local interrupt mask for TC. When TCIE is zero, TC must be polled; when TCIE is one and TC is one, an interrupt is requested.

Writing a zero to TE requests that the transmitter stop when it can. The transmitter completes any transmission in progress before shutting down. Only an MCU reset can cause the transmitter to stop and shut down immediately. If TE is cleared when the transmitter is already idle, the pin reverts to its general purpose I/O function (synchronized to the bit-rate clock). If anything is being transmitted when TE is cleared, that character is completed before the pin reverts to general purpose I/O, but any other characters waiting in the transmit queue are lost. The TC and TDRE flags are set at the completion of this last character, even though TE has been disabled.

7.7.1 Receiver flags

The SCI1 receiver has seven status flags, three of which can generate interrupt requests. The status flags are set by the SCI1 logic in response to specific conditions in the receiver. These flags can be read (polled) at any time by software. Refer to Figure 7-3, which shows SCI1 interrupt arbitration.

When an overrun takes place, the new character is lost, and the character that was in its way in the parallel receive data register (RDR) is undisturbed. RDRF is set when a character has been

received and transferred into the parallel RDR. The OR flag is set instead of RDRF if overrun occurs. A new character is ready to be transferred into the RDR before a previous character is read from the RDR.

The NF, FE and PF flags provide additional information about the character in the RDR, but do not generate interrupt requests.

The receiver active flag (RAF) indicates that the receiver is busy.

The last receiver status flag and interrupt source come from the IDLE flag. The RXD line is idle if it has constantly been at logic one for a full character time. The IDLE flag is set only after the RXD line has been busy and becomes idle. This prevents repeated interrupts for the time RXD remains idle.

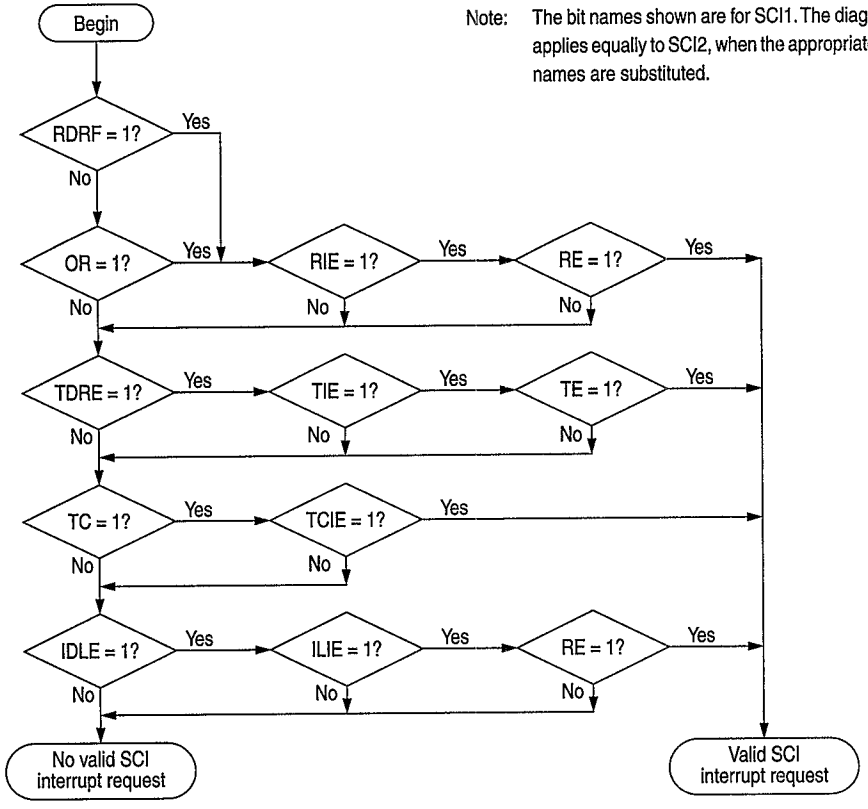


Figure 7-3 Interrupt source resolution within SCI1

7.8 SCI2

In addition to the subsystem described in the preceding paragraphs, (SCI1), the MC68HC11KG4 has another, similar, SCI module (SCI2). This system is identical to SCI1, with the following exceptions:

- SCI2 shares I/O with the following port D pins:

Pin	Alternative function
PD0	RXD2
PD1	TXD2

- The SCI2 transmit and receive functions are enabled by TE and RE respectively, in SC2CR2
- SCI1 functions and data are handled by a register block at $0050–$0057. The corresponding registers for SCI2 are at addresses $0070–$0077, as described in the following sections.
- The SCI2 baud rate register SC2BDH/SC2BDL is at address $0070/71.

7

7.8.1 SC2BDH, SC2BDL — SCI2 baud rate control registers

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
SCI2 baud rate high (SC2BDH)	$0070	BTST	BSPL	SYNC	SBR12	SBR11	SBR10	SBR9	SBR8	0000 0000
SCI2 baud rate low (SC2BDL)	$0071	SBR7	SBR6	SBR5	SBR4	SBR3	SBR2	SBR1	SBR0	0000 0100

The contents of this register determine the baud rate of SCI2. For details of the bits and the corresponding baud rates, see Section 7.6.1.

7.8.2 SC2CR1 — SCI2 control register 1

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
SCI2 control 1 (SC2CR1)	$0072	LOOPS	WOMS	0	M	WAKE	ILT	PE	PT	0000 0000

The SC2CR1 register provides the control bits that determine word length and select the method used for the wake-up feature. Bit 6 is described below; for details of the other bits see Section 7.6.2.

WOMS — Wired-OR mode for SCI2 pins (PD1, PD0)

1 (set) – TXD2 and RXD2 are open drains if operating as outputs.

0 (clear) – TXD2 and RXD2 operate normally.

7.8.3 SC2CR2 — SCI2 control register 2

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
SCI2 control 2 (SC2CR2)	$0073	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK	0000 0000

The SC2CR2 register provides the control bits that enable or disable individual SCI2 functions. For details of the bits see Section 7.6.3.

7.8.4 SC2SR1 — SCI2 status register 1

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
SCI2 status 1 (SC2SR1)	$0074	TDRE	TC	RDRF	IDLE	OR	NF	FE	PF	1100 0000

The bits in SC2SR1 indicate certain conditions in the SCI2 hardware and are automatically cleared by special acknowledge sequences. For details of the bits see Section 7.6.4

7.8.5 SC2SR2 — SCI2 status register 2

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
SCI2 status 2 (SC2SR2)	$0075	0	0	0	0	0	0	0	RAF	0000 0000

In the SC2SR2 only bit 0 is used, to indicate receiver active. The other seven bits always read zero.

7.8.6 SC2DRH, SC2DRL — SCI2 data high/low registers

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
SCI2 data high (SC2DRH)	$0076	R8	T8	0	0	0	0	0	0	undefined
SCI2 data low (SC2DRL)	$0077	R7T7	R6T6	R5T5	R4T4	R3T3	R2T2	R1T1	R0T0	undefined

SC2DRH/SC2DRL is a parallel register that performs two functions. It is the receive data register when it is read, and the transmit data register when it is written. Reads access the receive data buffer and writes access the transmit data buffer. Data received or transmitted is double buffered. See Section 7.6.6 for more details.

8
SERIAL PERIPHERAL INTERFACE

The serial peripheral interface (SPI), an independent serial communications subsystem, allows the MCU to communicate synchronously with peripheral devices, such as transistor-transistor logic (TTL) shift registers, liquid crystal (LCD) display drivers, analog-to-digital converter subsystems, and other microprocessors. The SPI is also capable of inter-processor communication in a multiple master system. The SPI system can be configured as either a master or a slave device, with data rates as high as one half of the E clock rate when configured as a master and as fast as the E clock rate when configured as a slave.

The SPI shares I/O with four of port D's pins and is enabled by SPE in the SPCR.

Pin	Alternative function
PD2	MISO
PD3	MOSI
PD4	SCK
PD5	$\overline{SS}$

8.1 Functional description

The central element in the SPI system is the block containing the shift register and the read data buffer (see Figure 8-1). The system is single buffered in the transmit direction and double buffered in the receive direction. This means that new data for transmission cannot be written to the shifter until the previous transfer is complete; however, received data is transferred into a parallel read data buffer so the shifter is free to accept a second serial character. As long as the first character is read out of the read data buffer before the next serial character is ready to be transferred, no overrun condition occurs. A single MCU register address is used for reading data from the read data buffer and for writing data to the shifter.

The SPI status block represents the SPI status functions (transfer complete, write collision, and mode fault) performed by the serial peripheral status register (SPSR). The SPI control block represents those functions that control the SPI system through the serial peripheral control register (SPCR).

8.2 SPI transfer formats

During an SPI transfer, data is simultaneously transmitted and received. A serial clock line synchronizes shifting and sampling of the information on the two serial data lines. A slave select line allows individual selection of a slave SPI device; slave devices that are not selected do not interfere with SPI bus activities. On a master SPI device, the select line can optionally be used to indicate a multiple master bus contention. Refer to Figure 8-2.

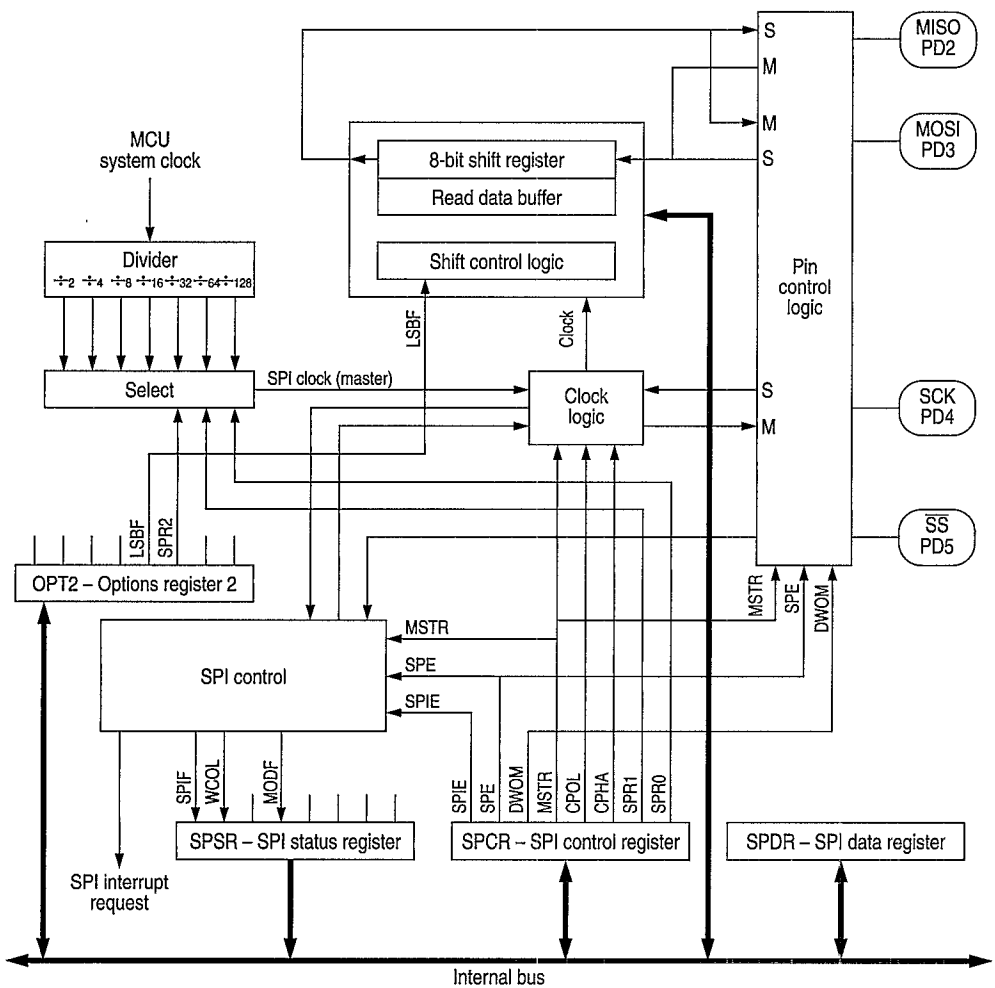
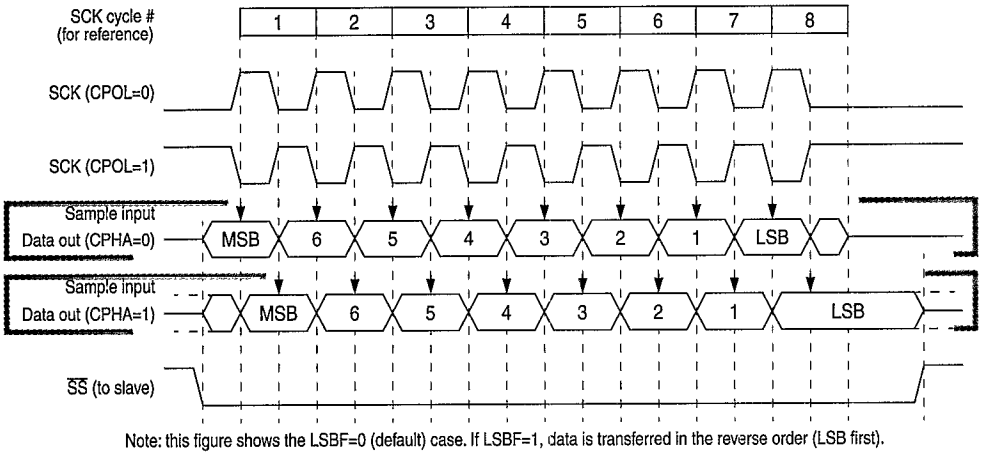


Figure 8-1 SPI block diagram



Note: this figure shows the LSBF=0 (default) case. If LSBF=1, data is transferred in the reverse order (LSB first).

Figure 8-2 SPI transfer format

8.2.1 Clock phase and polarity controls

Software can select one of four combinations of serial clock phase and polarity using two bits in the SPI control register (SPCR). The clock polarity is specified by the CPOL control bit, which selects an active high or active low clock, and has no significant effect on the transfer format. The clock phase (CPHA) control bit selects one of two different transfer formats. The clock phase and polarity should be identical for the master SPI device and the communicating slave device. In some cases, the phase and polarity are changed between transfers to allow a master device to communicate with peripheral slaves having different requirements.

When CPHA equals zero, the $\overline{SS}$ line must be deasserted and reasserted between each successive serial byte. Also, if the slave writes data to the SPI data register (SPDR) while $\overline{SS}$ is low, a write collision error results.

When CPHA equals one, the $\overline{SS}$ line can remain low between successive transfers.

8.3 SPI signals

The following paragraphs contain descriptions of the four SPI signals: master in slave out (MISO), master out slave in (MOSI), serial clock (SCK), and slave select ($\overline{SS}$).

Any SPI output line must have its corresponding data direction bit in DDRD register set. If the DDR bit is clear, that line is disconnected from the SPI logic and becomes a general-purpose input. All SPI input lines are forced to act as inputs regardless of the state of the corresponding DDR bits in DDRD register.

8.3.1 Master in slave out

MISO is one of two unidirectional serial data signals. It is an input to a master device and an output from a slave device. The MISO line of a slave device is placed in the high-impedance state if the slave device is not selected.

8.3.2 Master out slave in

The MOSI line is the second of the two unidirectional serial data signals. It is an output from a master device and an input to a slave device. The master device places data on the MOSI line a half-cycle before the clock edge that the slave device uses to latch the data.

8.3.3 Serial clock

SCK, an input to a slave device, is generated by the master device and synchronizes data movement in and out of the device through the MOSI and MISO lines. Master and slave devices are capable of exchanging a byte of information during a sequence of eight clock cycles.

There are four possible timing relationships that can be chosen by using control bits CPOL and CPHA in the serial peripheral control register (SPCR). Both master and slave devices must operate with the same timing. The SPI clock rate select bits of the master device, SPR[2:0], select the clock rate. SPR[1:0] are found in the SPCR register and SPR2 is in the OPT2 register. In a slave device, SPR[2:0] have no effect on the operation of the SPI.

8.3.4 Slave select

The slave select $\overline{SS}$ input of a slave device must be externally asserted before a master device can exchange data with the slave device. $\overline{SS}$ must be low before data transactions begin and must stay low for the duration of the transaction.

The $\overline{SS}$ line of the master must be held high. If it goes low, a mode fault error flag (MODF) is set in the serial peripheral status register (SPSR). To disable the mode fault circuit, write a one in bit 5 of the port D data direction register. This sets the $\overline{SS}$ pin to act as a general-purpose output, rather than a dedicated input to the slave select circuit, thus inhibiting the mode fault flag. The other three lines are dedicated to the SPI whenever the serial peripheral interface is on.

The state of the master and slave CPHA bits affects the operation of $\overline{SS}$. CPHA settings should be identical for master and slave. When CPHA = 0, the shift clock is the OR of $\overline{SS}$ with SCK. In this clock phase mode, $\overline{SS}$ must go high between successive characters in an SPI message. When CPHA = 1, $\overline{SS}$ can be left low between successive SPI characters. In cases where there is only one SPI slave MCU, its $\overline{SS}$ line can be tied to V_{SS} as long as only CPHA = 1 clock mode is used.

8.4 SPI system errors

Two kinds of system errors can be detected by the SPI system. The first type of error arises in a multiple-master system when more than one SPI device simultaneously tries to be a master. This error is called a mode fault. The second type of error, write collision, indicates that an attempt was made to write data to the SPDR while a transfer was in progress.

When the SPI system is configured as a master and the $\overline{SS}$ input line goes to active low, a mode fault error has occurred — usually because two devices have attempted to act as master at the same time. In the case where more than one device is concurrently configured as a master, there is a chance of contention between two pin drivers. For push-pull CMOS drivers, this contention can cause permanent damage. The mode fault detection circuitry attempts to protect the device by disabling the drivers. The MSTR control bit in the SPCR and all four DDRD control bits associated with the SPI are cleared and an interrupt is generated (subject to masking by the SPIE control bit and the I bit in the CCR).

Other precautions may need to be taken to prevent driver damage. If two devices are made masters at the same time, the mode fault detector does not help protect either one unless one of them selects the other as slave. The amount of damage possible depends on the length of time both devices attempt to act as master.

A write collision error occurs if the SPDR is written while a transfer is in progress. Because the SPDR is not double buffered in the transmit direction, writes to SPDR cause data to be written directly into the SPI shift register. Because this write corrupts any transfer in progress, a write collision error is generated. The transfer continues undisturbed, and the write data that caused the error is not written to the shifter.

A write collision is normally a slave error because a slave has no control over when a master initiates a transfer. A master knows when a transfer is in progress, so there is no reason for a master to generate a write-collision error, although the SPI logic can detect write collisions in both master and slave devices.

The SPI configuration determines the characteristics of a transfer in progress. For a master, a transfer begins when data is written to SPDR and ends when SPIF is set. For a slave with CPHA equal to zero, a transfer starts when $\overline{SS}$ goes low and ends when $\overline{SS}$ returns high. In this case, SPIF is set at the middle of the eighth SCK cycle when data is transferred from the shifter to the parallel data register, but the transfer is still in progress until $\overline{SS}$ goes high. For a slave with CPHA equal to one, transfer begins when the SCK line goes to its active level, which is the edge at the beginning of the first SCK cycle. The transfer ends when SPIF is set, for a slave in which CPHA=1.

8.5 SPI registers

The three SPI registers, SPCR, SPSR, and SPDR, provide control, status, and data storage functions. Refer to the following information for a description of how these registers are organized.

8.5.1 SPCR — SPI control register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
SPI control (SPCR)	$0028	SPIE	SPE	DWOM	MSTR	CPOL	CPHA	SPR1	SPR0	0000 01uu

This register can be read at any time. It can be written at any time except when the MBSP bit in the CONFIG register is set, upon which the SPCR bits are forced into their reset state and the SPI is disabled.

SPIE — Serial peripheral interrupt enable

1 (set) – A hardware interrupt sequence is requested each time SPIF or MODF is set.

0 (clear) – SPI interrupts are inhibited.

8

Set the SPIE bit to a one to request a hardware interrupt sequence each time the SPIF or MODF status flag is set. SPI interrupts are inhibited if this bit is clear or if the I bit in the condition code register is one.

SPE — Serial peripheral system enable

1 (set) – Port D [5:2] is dedicated to the SPI.

0 (clear) – Port D has its default I/O functions and the clock generator is stopped.

When the SPE bit is set, the port D pins 2, 3, 4, and 5 are dedicated to the SPI functions and lose their general purpose I/O functions. When the SPI system is enabled and expects any of PD[4:2] to be inputs then those pins will be inputs regardless of the state of the associated DDRD bits. If any of PD[4:2] are expected to be outputs then those pins will be outputs only if the associated DDRD bits are set. However, if the SPI is in the master mode, DDD5 determines whether PD5 is an error detect input (DDD5 = 0) or a general-purpose output (DDD5 = 1).

DWOM — Port D wired-OR mode

1 (set) – Port D [5:2] buffers configured for open-drain outputs.

0 (clear) – Port D [5:2] buffers configured for normal CMOS outputs.

MSTR — Master mode select

1 (set) – Master mode

0 (clear) – Slave mode

CPOL — Clock polarity

1 (set) – SCK is active low.

0 (clear) – SCK is active high.

When the clock polarity bit is cleared and data is not being transferred, the SCK pin of the master device has a steady state low value. When CPOL is set, SCK idles high. Refer to Figure 8-2 and Section 8.2.1.

CPHA — Clock phase

The clock phase bit, in conjunction with the CPOL bit, controls the clock-data relationship between master and slave. The CPHA bit selects one of two different clocking protocols. Refer to Figure 8-2 and Section 8.2.1.

SPR1 and SPR0 — SPI clock rate selects

These two bits select the SPI clock rate, as shown in Table 8-1. Note that SPR2 is located in the OPT2 register, and that its state on reset is zero.

Table 8-1 SPI clock rates

SPR[2:0]	E clock divide ratio	SPI clock frequency (≡ baud rate) for:			
		E = 2MHz	E = 3MHz	E = 4MHz	E = 4.2MHz
0 0 0	2	1.0 MHz	1.5 MHz	2.0 MHz	2.1MHz
0 0 1	4	500 kHz	750kHz	1.0 MHz	1.05MHz
0 1 0	16	125 kHz	187.5 kHz	250 kHz	262.5kHz
0 1 1	32	62.5 kHz	93.7 kHz	125 kHz	131.25kHz
1 0 0	8	250 kHz	375 kHz	500 kHz	525kHz
1 0 1	16	125 kHz	187.5 kHz	250 kHz	262.5kHz
1 1 0	64	31.3 kHz	46.9 kHz	62.5 kHz	65.625kHz
1 1 1	128	15.6 kHz	23.4 kHz	31.3 kHz	32.8kHz

8.5.2 SPSR — SPI status register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
SPI status (SPSR)	$0029	SPIF	WCOL	0	MODF	0	0	0	0	0000 0000

This register can be read at any time, but writing to it has no effect.

SPIF — SPI interrupt complete flag

1 (set) – Data transfer to external device has been completed.

0 (clear) – No valid completion of data transfer.

SPIF is set upon completion of data transfer between the processor and the external device. If SPIF goes high, and if SPIE is set, a serial peripheral interrupt is generated. To clear the SPIF bit, read the SPSR with SPIF set, then access the SPDR. Unless SPSR is read (with SPIF set) first, attempts to write SPDR are inhibited.

WCOL — Write collision

1 (set) – Write collision.

0 (clear) – No write collision.

8

Clearing the WCOL bit is accomplished by reading the SPSR (with WCOL set) followed by an access of SPDR. Refer to Section 8.3.4 and Section 8.4.

MODF — Mode fault

1 (set) – Mode fault.

0 (clear) – No mode fault.

To clear the MODF bit, read the SPSR (with MODF set), then write to the SPCR. Refer to Section 8.3.4 and Section 8.4.

Bits [5, 3, 2, 1, 0] — Not implemented; always read zero.

8.5.3 SPDR — SPI data register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
SPI data (SPDR)	$002A	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	undefined

The SPDR is used when transmitting or receiving data on the serial bus. Only a write to this register initiates transmission or reception of a byte, and this only occurs in the master device. At the completion of transferring a byte of data, the SPIF status bit is set in both the master and slave devices.

A read of the SPDR is actually a read of a buffer. To prevent an overrun and the loss of the byte that caused the overrun, the first SPIF must be cleared by the time a second transfer of data from the shift register to the read buffer is initiated.

SPI is double buffered in and single buffered out.

8.5.4 OPT2 — System configuration options register 2

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
System config. options 2 (OPT2)	$0038	LIRDV	CWOM	STRCH	IRVNE	LSBF	SPR2	XDV1	XDV0	000x 0000

LIRDV — LIR driven (refer to Section 4)

1 (set) – Enable LIR drive high pulse.

0 (clear) – LIR not driven on MODA/$\overline{\text{LIR}}$ pin.

CWOM — Port C wired-OR mode (refer to Section 6)

1 (set) – Port C outputs are open-drain.

0 (clear) – Port C operates normally.

STRCH — Stretch external accesses (refer to Section 4)

1 (set) – Off-chip accesses are extended by one E clock cycle.

0 (clear) – Normal operation.

IRVNE — Internal read visibility/not E (refer to Section 4)

1 (set) – Data from internal reads is driven out of the external data bus.

0 (clear) – No visibility of internal reads on external bus.

In single chip mode this bit determines whether the E clock drives out from the chip.

1 (set) – E pin is driven low.

0 (clear) – E clock is driven out from the chip.

LSBF — LSB first enable

1 (set) – SPI data is transferred LSB first.

0 (clear) – SPI data is transferred MSB first.

If this bit is set, data, which is usually transferred MSB first, is transferred LSB first. LSBF does not affect the position of the MSB and LSB in the data register. Reads and writes of the data register always have MSB in bit 7.

SPR2 — SPI clock rate select

When set, SPR2 adds a divide-by-4 prescaler to the SPI clock chain. With the two bits in the SPCR, this bit specifies the SPI clock rate. Refer to Table 8-1.

XDV[1, 0] — XOUT clock divide select (refer to Section 4)

These two bits control the frequency of the XCLK signal, which is output on the XOUT pin if enabled by the CLKX bit in CONFIG.

9
PULSE WIDTH MODULATION TIMER

The PWM timer subsystem provides up to eight 9-bit pulse-width modulated waveforms on the port H and port K pins. A 16kHz repetition frequency is achieved by programming the PWM with a 2 E clock frequency with E = 4MHz.

Pin	Alternative function
PH[0,1]	PWM0
PH[2,3]	PWM1
PH[4,5]	PWM2
PH[6,7]	PWM3
PK[0,1]	PWM4
PK[2,3]	PWM5
PK[4,5]	PWM6
PK[6,7]	PWM7

Three control registers configure the PWM outputs — PWCLK, PWPOL and PWEN. The PWCLK register selects the prescale value for the PWM clock sources and indicates when a PWM interrupt has occurred. The PWPOL register determines each channel's polarity and the PWEN register enables the PWM channels.

The eight 9-bit channels share a counter and there is a separate duty cycle register for each channel. The duty cycle registers are double buffered so that if they are changed while the channel is enabled, the change does not take effect until the counter rolls over or the channel is disabled. A new duty cycle can be forced into effect immediately by writing to the duty cycle register and then writing to the counter.

The 9-bit counter runs at the rate of the selected clock source as shown. The value of this counter is compared with that of a duty register. When a match occurs, a flip-flop changes state, causing the counter output to change state also. When the counter rolls over it resets the flip-flop and the output changes back. The select bit PPOL allows each channel to independently create a signal which is going from low-high or from high-low. Setting a bit in the PPOL register has immediate effect on the output signal.

The MSB in each duty register is used to select whether the left or the right port line (i.e. the left or right path of an H-bridge) is enabled for the PWM signal. The opposite port line drives the

general purpose output value. The MSB bit also inverts the PWM signal; this is done because the duty values are assumed to be 2s complement values within the range -512 to +511. A change in duty value from $03FF to $0000 (–1 to 0) causes a change in current direction within the corresponding H-bridge. This is done by changing the polarity of the PWM signal and exchanging PWM and output port signal within the H-bridge.

9.1 PWM control registers

9.1.1 PWCLK — PWM control register with clock control bits

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Pulse width control register(PWCLK)	$0078	0	0	PCKC2	PCKC1	0	0	PWMF	PWMI	0000 0000

This register contains bits for determining the rate of clock C, as well as the PWM interrupt flag.

Bits[7,6] — Not implemented; always read zero.

PCKC[2:1] — Prescaler for clock C

These two bits determine the rate of clock C, the clock source for each channel, as shown in the following table:

PCKC[2:1]	Value of clock C
0 0	2E
0 1	E
1 0	E / 2
1 1	E / 4

Bits[3,2] — Not implemented; always read zero.

PWMF — PWM interrupt flag

1 (set) – PWM interrupt has occurred.

0 (clear) – PWM interrupt has not occurred.

The PWM interrupt flag bit indicates when interrupt conditions have occurred. On-chip hardware sets this bit as a result of a PWM interrupt. To clear this bit in the PWCLK register, a write to PWCLK must be performed, where the PWMF bit to be cleared by the write is set to '1'.

PWMI — PWM interrupt mask

1 (set) – PWM interrupt enabled.

0 (clear) – PWM interrupt disabled.

9.1.2 PWPOL — PWM timer polarity register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Pulse width polarity register(PWPOL)	$007A	PPOL7	PPOL6	PPOL5	PPOL4	PPOL3	PPOL2	PPOL1	PPOL0	0000 0000

PPOL[7:0] — Pulse width channel x polarity

1 (set) – PWM channel x output is high at the beginning of the clock cycle and goes low when the duty count is reached.

0 (clear) – PWM channel x output is low at the beginning of the clock cycle and goes high when duty count is reached.

Each channel has a polarity bit that allows a cycle to start with either a high or a low level. This is shown on the block diagram, Figure 9-1, as a selection of either the Q output or the $\overline{Q}$ output of the PWM output flip flop. When one of the bits in the PWPOL register is set, the associated PWM channel output is high at the beginning of the clock cycle, then goes low when the duty count is reached. Setting a bit in PWPOL has immediate effect on the output.

9.1.3 PWEN — PWM timer enable register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Pulse width enable (PWEN)	$0079	PWEN7	PWEN6	PWEN5	PWEN4	PWEN3	PWEN2	PWEN1	PWEN0	0000 0000

When a bit of the PWEN register is zero the associated port H or port K lines are general purpose outputs. When a bit is at logic one, the associated port H or port K line becomes a PWM output, enabling the PWM channel.

PWEN[7:0] — Pulse width channel x enable

1 (set) – Channel enabled on the associated port pin.

0 (clear) – Channel disabled.

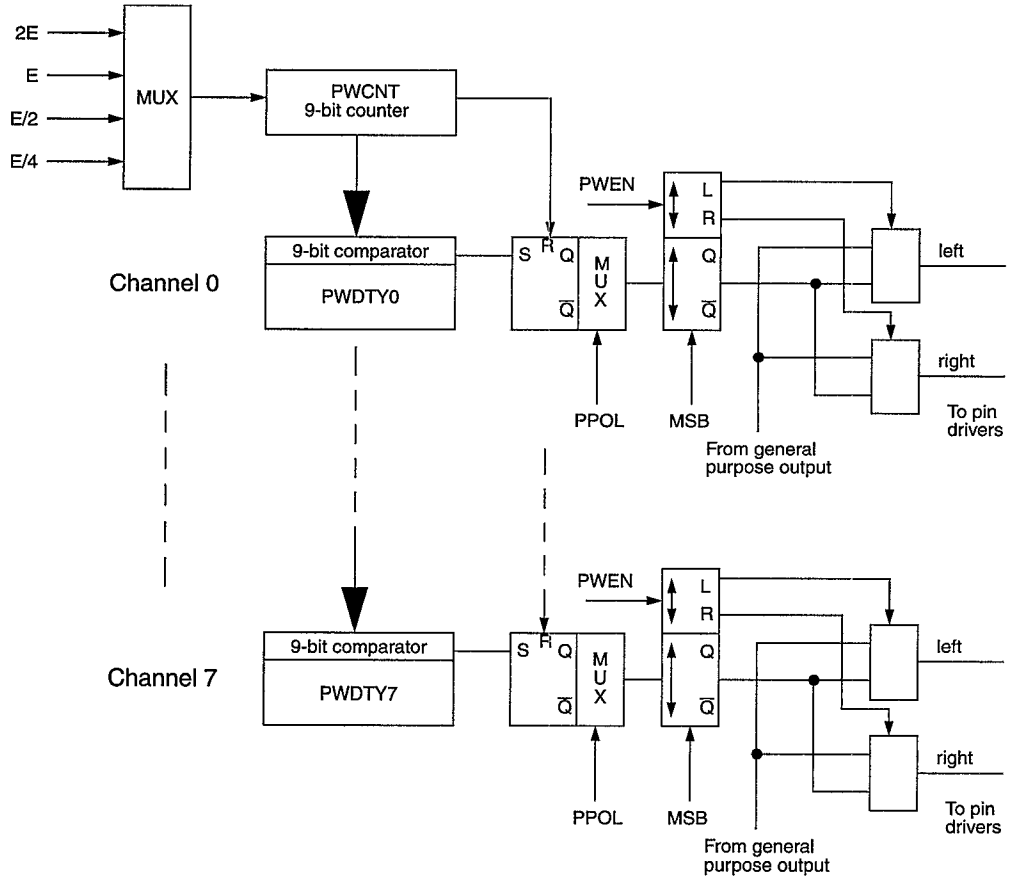


Figure 9-1 PWM timer block diagram

9.2 PWM counter register

The eight 9-bit channels share a counter which may be read at any time without affecting the count or the operation of the PWM channels. Writes to a counter cause the counter to be reset to $0000. This is generally done before the counter is enabled; however the counter can be written to whilst enabled at the risk of a truncated PWM period. Note that if the 2E clock is selected the LSB of the counter will always read zero, since it is changing at twice the rate of the CPU bus. The counter will always count to $01FF (9 bits) before rolling over to $0000.

9.2.1 PWCNT — PWM timer counter register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Pulse width count (PWCNT) high	$007B	0	0	0	0	0	0	0	(bit 8)	0000 0000
Pulse width count (PWCNT) low	$007C	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	0000 0000

PWCNT can be read at any time; a read of the high order byte latches the low order byte for one cycle so that double reads relate to the same count. A write to this register causes it to be reset to $0000.

9.3 PWM channel duty registers

There is one duty register for each channel. The value in this register determines the duty of the associated PWM timer channel. The duty value is compared to the counter and if it is equal to the counter value, the output goes to the state defined by the associated polarity bit. If the register is written while the channel is enabled, then the new value is held in a buffer until the counter rolls over or the channel is disabled.

Because this is a 9-bit PWM, a two byte duty register must be used for each channel.

9.3.1 PWDTY — PWM timer duty cycle register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Pulse width duty high (PWDTY)	$006x	0	0	0	0	0	0	MSB	(bit 8)	0000 0011
Pulse width duty low (PWDTY)	$006x	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111

MSB

Most significant bit extension of the 9-bit duty value, in order to achieve a 10-bit 2s complement value. The MSB controls the current direction in the H-bridge.

Bit 0 – bit 8

This 9-bit value determines the duty cycle of the associated PWM channel.

Reads of this register return the most recent value written.

Depending on the polarity bit PPOL, the duty registers may contain the count of either the high time or the low time. If the polarity bit is one, the output starts high and then goes low when the duty count is reached, so the duty registers contain a count of the high time. Conversely, if the polarity bit is zero,

the output starts low and then goes high when the duty count is reached, so the duty registers contain a count of the low time.

If polarity = 1, then:

Duty cycle = (PWDTYx/512) x 100%

If polarity = 0:

Duty cycle = ((512 – PWDTYx)/512) x 100%

Note: If the duty register is set to $0000, then the output will always be in the state which would normally be the state changed to at the duty change of state (see also Section 9.3.2).

9.3.2 Boundary cases

The following boundary conditions apply to the values stored in the PWDTYx register and the PPOLx bits:

- If PWDTYx = $0000 and PPOLx = 0 then the output is always high.
- If PWDTYx = $0000 and PPOLx = 1 then the output is always low.

9

9.4 Clock scheme for different PWM channels

Switching more than one PWM channel simultaneously would cause large currents in the corresponding power drivers of the H-bridges. Clocking of different channels has to be delayed so that only one channel switches at a time. In order to achieve this delay, microshifts are introduced as shown in Figure 9-2.

The start of the period time t_{period} is shifted from the E clock incrementally by a quarter of an E clock cycle ($t_E/4$) for channels 0 - 7. Inevitable large currents due to switching of the power drivers are thus distributed over a number of time slots, ensuring that the power supply is not overloaded.

9.5 PWM interrupt

The PWM system generates an interrupt every 64 periods of the 1/16kHz period time when the PWM is programmed with the 2E clock frequency. The counter register is extended by six further bits, thus enabling the interrupt to be synchronized with the PWM period. Overflow of the extended counter is the condition for the PWM interrupt. Should the counter be reset via a write access on

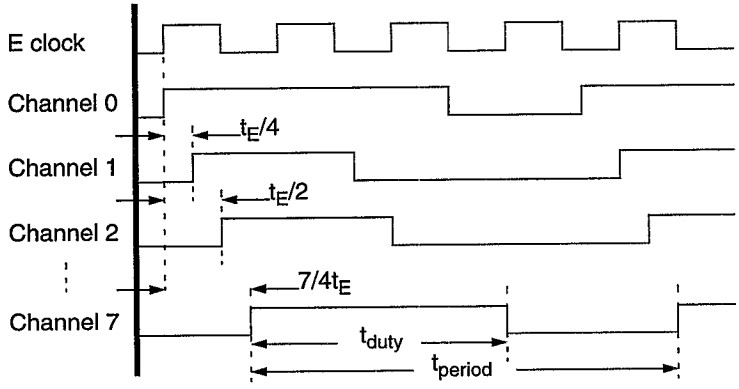


Figure 9-2 Switching delay by microshifts

the PWCNT register, then the extended counter bits will be reset also, ensuring that the PWM interrupt is always synchronized with the PWM period.

9.6 PWM test feature

For test purposes, the extended counter bits for the PWM interrupt can be disconnected from the PWM counter. The extended bits are then clocked with the PWM clock which is selected using the PCKC1/PCKC2 bits in the PWCLK register.

This clock bypass is achieved by setting the CBYP bit in the TEST1 register. A write access to the counter register PWCNT forces a reset of both the PWM counter and the PWM interrupt counter.

9.7 PWM operation in STOP and WAIT modes

When the CPU enters WAIT mode, the PWM clock is stopped, thus saving current. When the CPU enters STOP mode, all clocks are stopped. The PWM registers remain the same as they were before entering STOP or WAIT mode.

9.8 PWM H-bridge system

Two PWM H-bridge circuits are used to drive a system with cross-coupled coils for a dashboard instrument. A single bridge is formed by one PWM channel in combination with a general purpose output channel. This combination is capable of driving one of the two coils. The pulse-width ratio in one of the two PWM channels corresponds to the average value of the current through the coil.

The vector (I1, I2) of the currents through the coils in the cross coupled system is determined by the pulse widths (PW1, PW2). It also determines the elevation angle of the whole system.

The cross-coupled coils and the current vectors are shown in Figure 9-3.

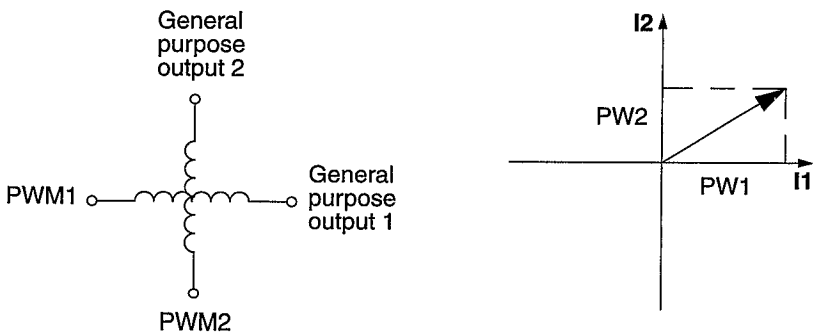


Figure 9-3 Driving cross coupled coils

9

9.8.1 H-bridge power driver

Special power drivers in the H-bridge must be used to drive the system with coils, because high voltages which occur at the driver outputs due to switching the coils would damage the circuits if the drivers are not protected against this. The H-bridge driver circuit is shown in Figure 9-4.

In order to avoid large switching currents through the NMOS and PMOS driver transistors, both devices are not active at the same time. However, the switching delay between the NMOS and PMOS transistors must be short, because the driver has to supply the coil circuit with a continuous current during the commutation. If the commutation time is not short enough, there would be diode currents into the bulk due to voltages below PV_{SS} or greater than PV_{DD} on the driver outputs. Low voltage drops on the device transistors are also necessary to avoid these diode currents. This results in large power devices for the H-bridge.

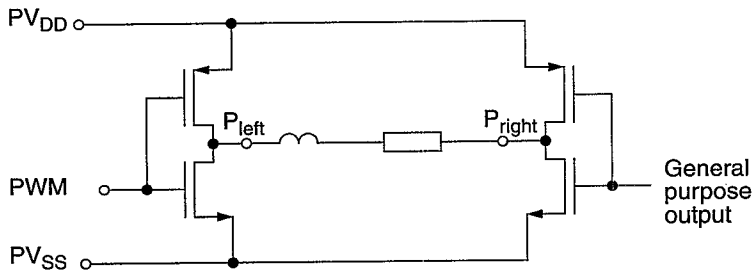


Figure 9-4 H-bridge driver circuit

9.8.2 Power driver circuit

A short switching delay and small switching currents are achieved by the circuit shown in Figure 9-5.

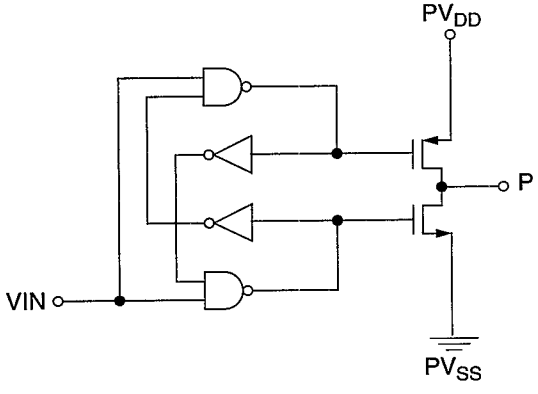


Figure 9-5 Power driver circuit

A high-low transition at the input VIN produces a low-high transition at the driver PMOS gate prior to the corresponding H-bridge low-high transition at the driver NMOS gate. The NOR gate blocks the transition at VIN as long as the upper inverter in Figure 9-5 produces a low level. The result is that both driver devices are not active at the same time. If the NMOS transistor width in the upper inverter is large compared to that of its PMOS transistor, then this results in a fast propagation of

the required low-level signal which unblocks the NOR gate.The switching delay is thus minimised, and the lower inverter in the figure realises a fast low-high transition.

9.8.3 Software short circuit protection

The short circuit protection method is a combined hardware/software solution. The pin value of a single power driver output is compared with the set level of the actual power driver. A difference between the two levels indicates a short circuit case at the actual power driver. The corresponding bit in the mismatch register PTHMS or PTKMS is then set. The short circuit protection circuitry is shown in Figure 9-6.

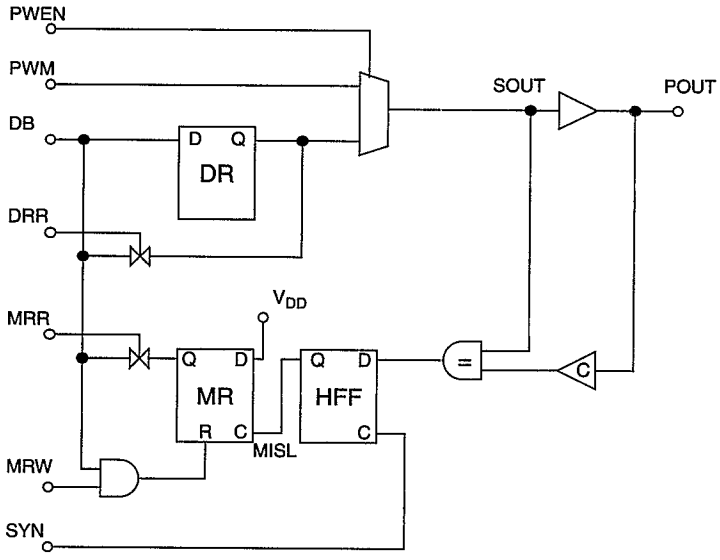


Figure 9-6 Short circuit protection circuitry

Each output bit of ports H and K can either be used as a general purpose output or as a PWM channel, depending on the state of the corresponding bit in the PWEN register. The actual output level POUT of a single power driver is compared with the set output level SOUT via the EXOR gate in the circuitry. The output of buffer 'C' goes low if the high set output is significantly lower than PVDD. Similarly, the buffer output goes high if the low set output is significantly higher than PVSS. The comparison result is latched with the appropriate signal SYN. The timing of the signal SYN

depends on the amount of microshifts for the actual PWM channel. The SYN signal occurs 1/4 of an E-clock cycle later than the start of the PWM period. This ensures that the PWM signal on the output POUT is stable. A short circuit on the output is indicated by a mismatch between the levels SOUT and POUT; this mismatch results in a high signal latched by the HFF. The latched mismatch signal MISL is now stored in the corresponding bit MR of the mismatch register. The mismatch register PTHMS or PTKMS can be polled in a proper time period like an interrupt flag register. A high on the mismatch register bit has to be handled like an interrupt flag. It can be cleared by writing a logic 1 back to this register. Since the maximum time a short circuit can be withstood by a single power is 20 ms, the mismatch registers should be polled at least every 10ms.

9.8.4 Driver states

A single H-bridge controls the direction of current through the coils. Three states are necessary for the required H-bridge operations for the 360° aircore instruments.

- Forward state - M1, M2 off; M0, M3 on.
- Backward state - M0, M3 off; M1, M2 on.
- Off state - M0, M2 off; M1, M3 on.

M0 - M3 are the driver transistors which form the H-bridge. The circuits in Figure 9-7 show how the bridge can operate in the different states. The driver transistors are shown as switches for simplification.

In each state, the output driver holds the output P_{LEFT} or P_{RIGHT} (left or right port line) at voltage $PVDD-V_{drop}$ while the PWM driver switches the opposite part of the H-bridge. The PWM driver switches between FORWARD state and OFF state or BACKWARD state and OFF state. The average current I is determined by the pulse width ratio at the PWM driver.

Current direction is changed by switching between FORWARD and BACKWARD state. The output and the PWM functionality has also been changed between the two port lines when switching current direction.

Holding the general purpose output at $PVDD-V_{drop}$ is done by initially writing a '1' to the PORTH and PORTK data register bits which control the H-bridge.

9.8.5 Port H and port K configurations

Each of the ports H and K incorporates two power driver blocks. A single power driver block consists of two H-bridges each with a left and right power driver. Each of the power driver blocks has a separate voltage supply. A port configuration which controls two 360° aircore instruments or stepper motors is shown in Figure 9-8. Note that a single instrument should not be controlled with H-bridges from different power driver blocks. The minimum voltage drop mismatch is only ensured within a single block. .

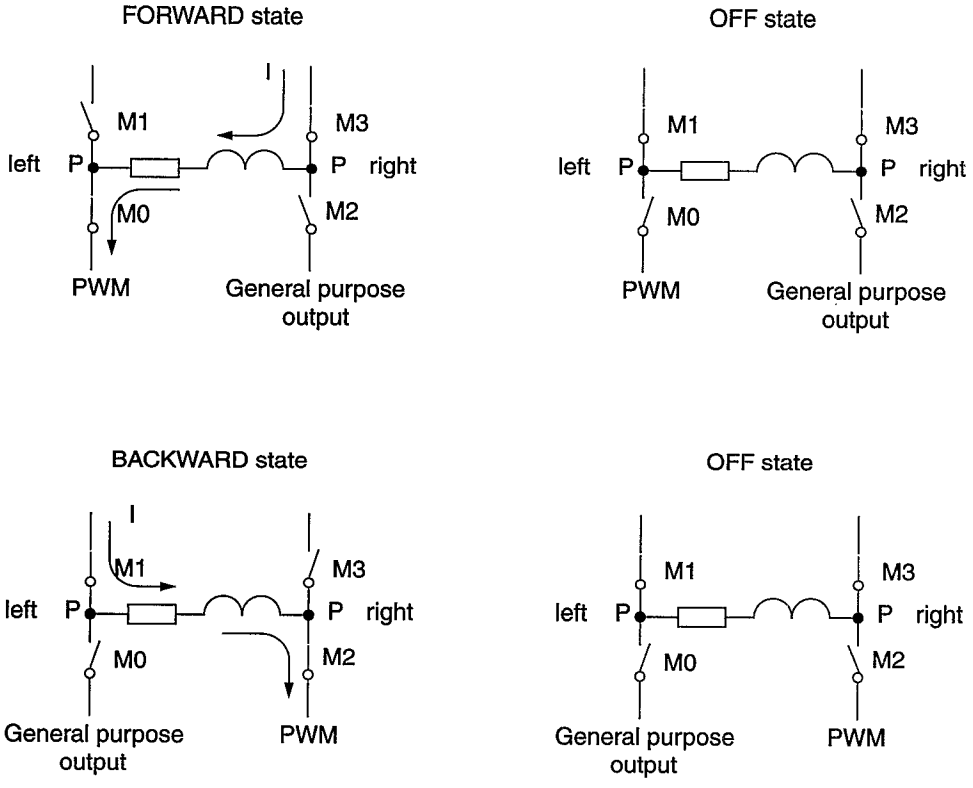


Figure 9-7 H-bridge states

Another configuration with four 90° small angle instruments is shown in Figure 9-9. The small angle instrument does not need a switching between two quadrants; it can be controlled by a single power driver and its PWM.

Note: The MSB in the duty register may not be changed during the operation with the 90° instruments. This would exchange the functionality of PWM and general purpose output between the left and the right port bit.

A minimum voltage drop mismatch is not ensured for the large angle instrument in this configuration.

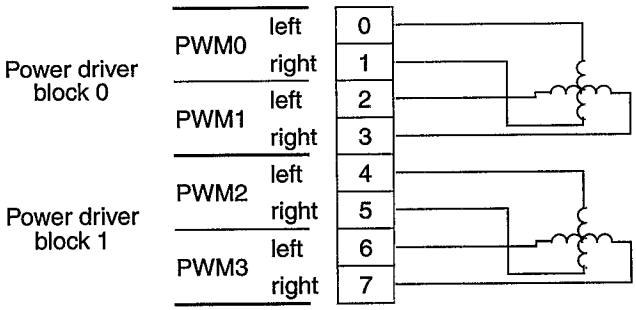


Figure 9-8 Configuration for two 360° instruments

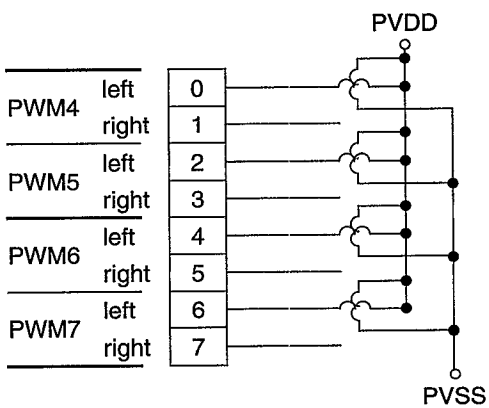


Figure 9-9 Configuration for four 90° instruments

9.8.6 H-bridge control with the PWM

The current in the PWM can be adjusted within the range of $-I_{MAX} * 512/512$ and $+I_{MAX} * 511/512$. The current is a linear function of the 10-bit 2's complement value in the duty register of the PWM channel. The MSB bit controls the current direction within the H-bridge. This will be done by

exchanging the PWM and general purpose output functionality and inverting the PWM signal in the H-bridge when switching the MSB bit.

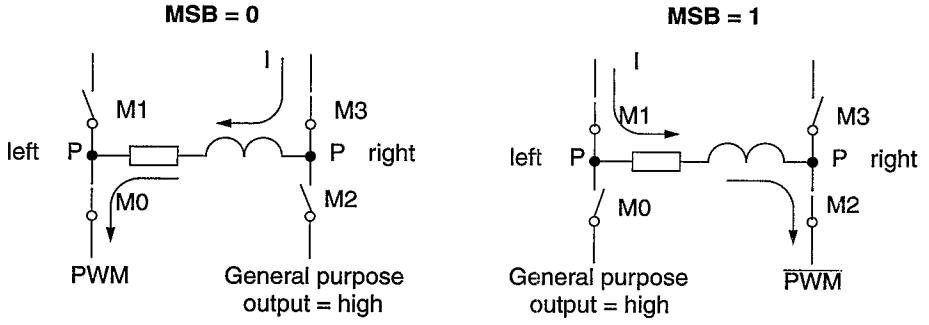


Figure 9-10 H-bridge control with the PWMs

Figure 9-11 shows the correspondence between the 2's complement values in the duty register and the required PWM signal in the H-bridge. The figure shows that the PWM signal has to be inverted for the negative values in order to obtain the correct H-bridge signal which is drawn in the figure. (Compare the periods for the values -1 and 7 in Figure 9-11.)

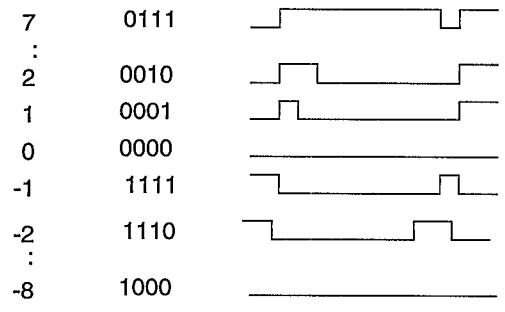


Figure 9-11 Correspondence between duty values and required PWM values

9.8.7 Operation in WAIT mode

When the CPU goes into WAIT state the outputs of ports H and K are set low by hardware. This is independent of the PWM output level and of the values in the output registers. The reason for this is to avoid a current flow within the H-bridge when in the WAIT state.

This feature can be disabled by a mask option on the MC68HC11KG4. In this case, when going into WAIT or STOP, port H and port K work as normal output ports which hold the last data register value or the last PWM value if this was enabled.

THIS PAGE LEFT BLANK INTENTIONALLY

10
TIMING SYSTEM

The M68HC11 timing system is composed of several clock divider chains. The main clock divider chain includes a 16-bit free-running counter, which is driven by a programmable prescaler. The main timer's programmable prescaler provides one of the four clocking rates to drive the 16-bit counter. Two prescaler control bits select the prescale rate. The prescaler output divides the system clock by 1, 4, 8, or 16. Taps from this main clocking chain drive circuitry are used to generate the slower clocks used by the pulse accumulator, the real-time interrupt (RTI) and the computer operating properly (COP) and window COP (WCOP) watchdog subsystems. Refer to Figure 10-1. A separate clock divider chain, which uses the XTAL clock directly, is used for the slow real-time interrupt (SRTI).

10.1 Timer operation

All main timer system activities are referenced to the free-running counter. The counter begins incrementing from $0000 as the MCU comes out of reset, and continues to the maximum count, $FFFF. At the maximum count, the counter rolls over to $0000, sets an overflow flag and continues to increment. As long as the MCU is running in a normal operating mode, there is no way to reset, change or interrupt the counting. The capture/compare subsystem features three input capture channels, four output compare channels and one channel that can be selected to perform either input capture or output compare. Each of the input capture functions has its own 16-bit input capture register (time capture latch) and each of the output compare functions has its own 16-bit compare register. All timer functions, including the timer overflow and RTI, have their own interrupt controls and separate interrupt vectors. See Table 10-1 for related frequencies and periods.

Clocks for the pulse accumulator, real time interrupt and COP/WCOP functions are derived from the internal ST4XCK signal. If the PLL circuit is active (VDDSYN = 1) and the MCS and BCS bits in PLLCR are both set, then ST4XCK is equal to the output of the PLL circuit, VCOOUT. Otherwise, ST4XCK is the same as EXTALi. Refer to Figure 10-1 and Section 2.

The pulse accumulator contains an 8-bit counter and edge select logic. The pulse accumulator can operate in either event counting mode or gated time accumulation mode. During event counting mode, the pulse accumulator's 8-bit counter increments when a specified edge is detected on an input pin. During gated time accumulation mode, an internal clock source ($ST4XCK/2^8$) increments the 8-bit counter while an input signal has a predetermined logic level. See Section 10.1.8.

The real-time interrupt (RTI) is a programmable periodic interrupt circuit that permits pacing of the execution of software routines by selecting one of four interrupt rates. It is clocked by the 16-bit timer (ST4XCK/2^{15}); see Section 10.1.4.

The slow real-time interrupt (SRTI) can provide slower interrupt rates than the RTI, the slowest interrupt rate having a 1s interval. The SRTI is clocked directly by the EXTALi oscillator frequency. (See Section 10.1.5).

The COP watchdog clock input is tapped off from the free-running counter chain (ST4XCK/2^{17}); see Section 10.1.6. The COP automatically times out unless it is serviced within a specific time by a program reset sequence. If the COP is allowed to time out, a reset is generated, which drives the $\overline{\text{RESET}}$ pin low to reset the MCU and the external system (see Section 5).

Table 10-1 Timer resolution and capacity

	Clock						
	4.0MHz	8.0MHz	12.0MHz	16.0MHz		ST4XCK	Crystal(1)
Control bits	1.0MHz	2.0MHz	3.0MHz	4.0MHz	4.2MHz	ST4XCK/4	Clock
PR[1:0]	1000ns	500ns	333ns	250ns	238ns	4/ST4XCK	Period
00	1.0µs	500ns	333ns	250ns	238ns	4/ST4XCK	– resolution
	65.536ms	32.768ms	21.845ms	16.384ms	15.604ms	2^{18}/ST4XCK	– overflow
01	4.0µs	2.0µs	1.333µs	1.0µs	0.952µs	16/ST4XCK	– resolution
	262.14ms	131.07ms	87.381ms	65.536ms	62.415ms	2^{20}/ST4XCK	– overflow
10	8.0µs	4.0µs	2.667µs	2.0µs	1.905µs	32/ST4XCK	– resolution
	524.29ms	262.14ms	174.76ms	131.07ms	124.83ms	2^{21}/ST4XCK	– overflow
11	16.0µs	8.0µs	5.333µs	4.0µs	3.81µs	64/ST4XCK	– resolution
	1049ms	524.29ms	349.53ms	262.14ms	249.7ms	2^{22}/ST4XCK	– overflow

(1) Crystal frequencies are valid only if the PLL is not active.

10

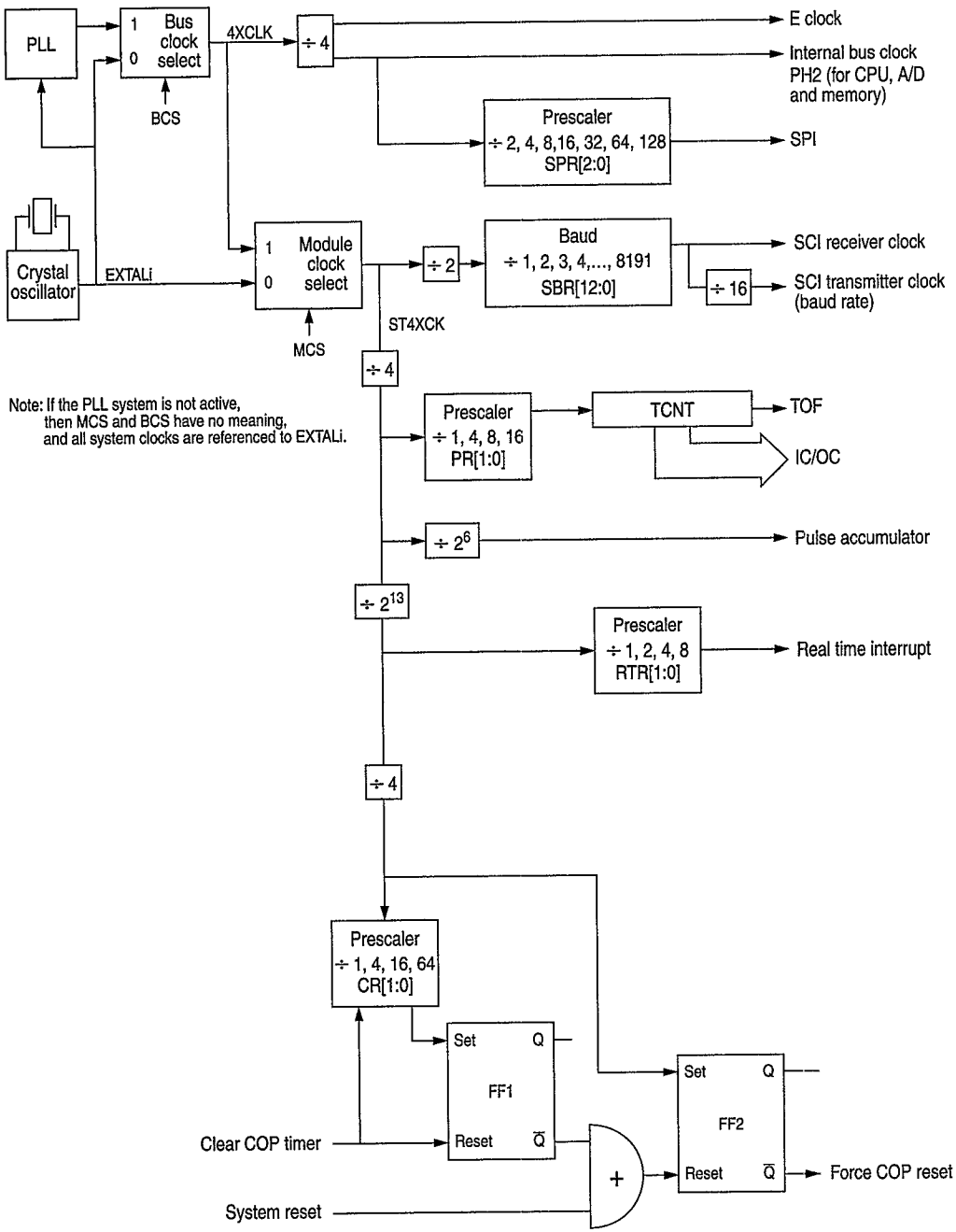


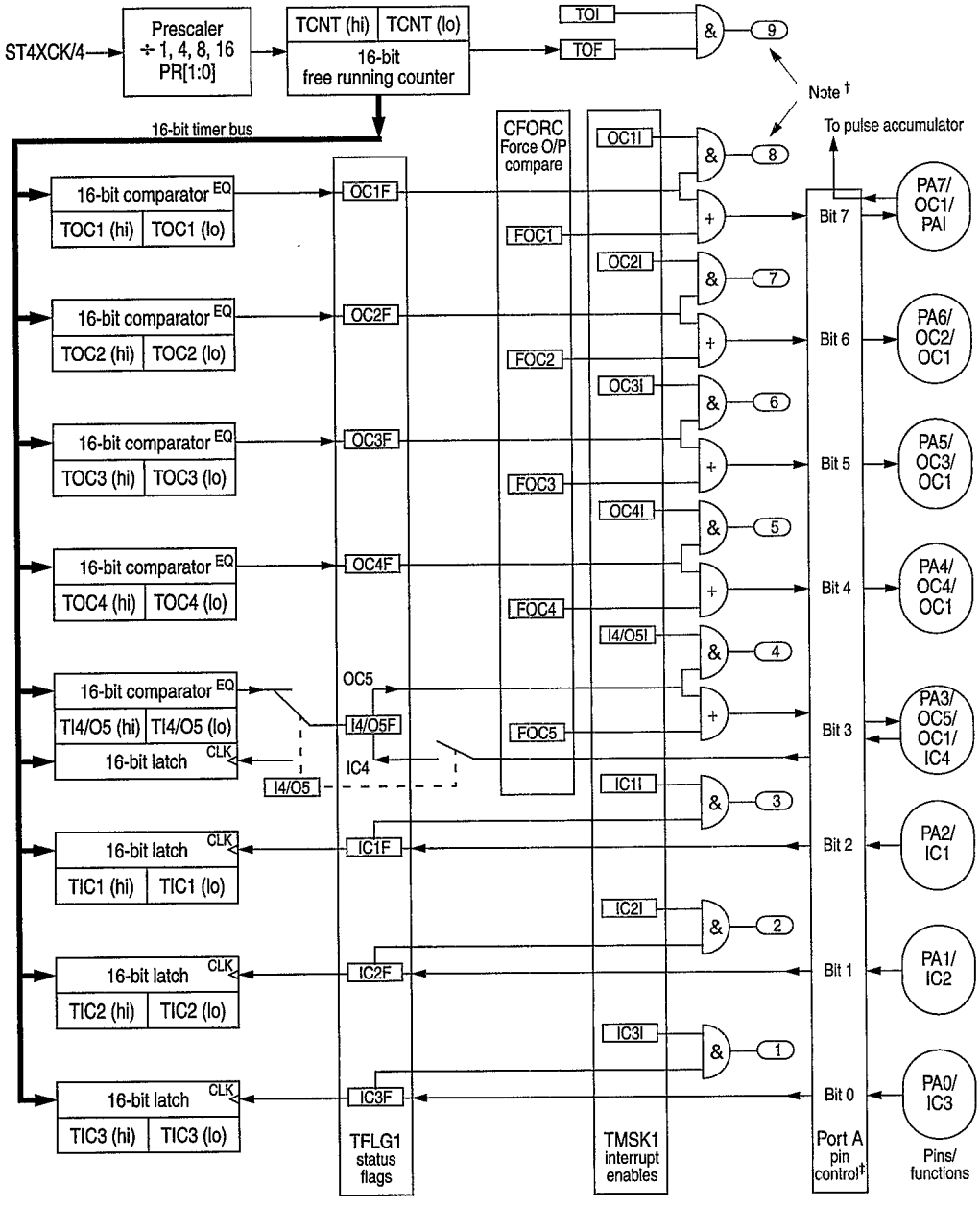
Figure 10-1 Timer clock divider chains

10.1.1 Timer structure

The timer functions share I/O with all eight pins of port A:

Pin	Alternative function
PA0	IC3
PA1	IC2
PA2	IC1
PA3	OC5 and/or OC1, or IC4
PA4	OC4 and/or OC1
PA5	OC3 and/or OC1
PA6	OC2 and/or OC1
PA7	PAI and/or OC1

Figure 10-2 shows the capture/compare system block diagram. The port A pin control block includes logic for timer functions and for general-purpose I/O. For pins PA3, PA2, PA1 and PA0, this block contains both the edge-detection logic and the control logic that enables the selection of which edge triggers an input capture. The digital level on PA[3:0] can be read at any time (read PORTA register), even if the pin is being used for the input capture function. Pins PA[6:3] are used either for general-purpose I/O, or as output compare pins. When one of these pins is being used for an output compare function, it cannot be written directly as if it were a general-purpose output. Each of the output compare functions (OC[5:2]) is related to one of the port A output pins. Output compare 1 (OC1) has extra control logic, allowing it optional control of any combination of the PA[7:3] pins. The PA7 pin can be used as a general-purpose I/O pin, as an input to the pulse accumulator or as an OC1 output pin.



† Interrupt requests 1–9 (these are further qualified by the I-bit in the CCR)

‡ Port A pin actions are controlled by OC1M, OC1D, PACTL, TCTL1 and TCTL2 registers

Figure 10-2 Capture/compare block diagram

10.1.2 Input capture

The input capture function records the time an external event occurs by latching the value of the free-running counter when a selected edge is detected at the associated timer input pin. Software can store latched values and use them to compute the periodicity and duration of events. For example, by storing the times of successive edges of an incoming signal, software can determine the period and pulse width of a signal. To measure period, two successive edges of the same polarity are captured. To measure pulse width, two alternative polarity edges are captured.

In most cases, input capture edges are asynchronous with respect to the internal timer counter, which is clocked relative to an internal clock (PH2). These asynchronous capture requests are synchronized with PH2 so that latching occurs on the opposite half cycle of PH2 from when the timer counter is being incremented. This synchronization process introduces a delay from when the edge occurs to when the counter value is detected. Because these delays cancel out when the time between two edges is being measured, the delay can be ignored. When an input capture is being used with an output compare, there is a similar delay between the actual compare point and when the output pin changes state.

The control and status bits that implement the input capture functions are contained in the PACTL, TCTL2, TMSK1, and TFLG1 registers.

To configure port A bit 3 as an input capture, clear the DDA3 bit of the DDRA register. Note that this bit is cleared out of reset. To enable PA3 as the fourth input capture, set the I4/O5 bit in the PACTL register. Otherwise, PA3 is configured as a fifth output compare out of reset, with bit I4/O5 being cleared. If the DDA3 bit is set (configuring PA3 as an output), and IC4 is enabled, then writes to PA3 cause edges on the pin to result in input captures. Writing to TI4/O5 has no effect when the TI4/O5 register is acting as IC4.

10.1.2.1 TCTL2 — Timer control register 2

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer control 2 (TCTL2)	$0021	EDG4B	EDG4A	EDG1B	EDG1A	EDG2B	EDG2A	EDG3B	EDG3A	0000 0000

Use the control bits of this register to program input capture functions to detect a particular edge polarity on the corresponding timer input pin. Each of the input capture functions can be independently configured to detect rising edges only, falling edges only, any edge (rising or falling), or to disable the input capture function. The input capture functions operate independently of each other and can capture the same TCNT value if the input edges are detected within the same timer count cycle.

EDGxB and EDGxA — Input capture edge control

There are four pairs of these bits. Each pair is cleared by reset and must be encoded to configure the corresponding input capture edge detector circuit. IC4 functions only if the I4/O5 bit in the PACTL register is set.

EDGxB	EDGxA	Configuration
0	0	ICx disabled
0	1	ICx captures on rising edges only
1	0	ICx captures on falling edges only
1	1	ICx captures on any edge

10.1.2.2 TIC1–TIC3 — Timer input capture registers

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer input capture 1 (TIC1) high	$0010	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	undefined
Timer input capture 1 (TIC1) low	$0011	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	undefined
Timer input capture 2 (TIC2) high	$0012	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	undefined
Timer input capture 2 (TIC2) low	$0013	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	undefined
Timer input capture 3 (TIC3) high	$0014	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	undefined
Timer input capture 3 (TIC3) low	$0015	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	undefined

When an edge has been detected and synchronized, the 16-bit free-running counter value is transferred into the input capture register pair as a single 16-bit parallel transfer. Timer counter value captures and timer counter incrementing occur on opposite half-cycles of the phase 2 clock so that the count value is stable whenever a capture occurs. Input capture values can be read from a pair of 8-bit read-only registers. A read of the high-order byte of an input capture register pair inhibits a new capture transfer for one bus cycle. If a double-byte read instruction, such as LDD, is used to read the captured value, coherency is assured. When a new input capture occurs immediately after a high-order byte read, transfer is delayed for an additional cycle but the value is not lost.

The TICx registers are not affected by reset.

10

10.1.2.3 TI4/O5 — Timer input capture 4/output compare 5 register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Capture 4/compare 5 (TI4/O5) high	$001E	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	1111 1111
Capture 4/compare 5 (TI4/O5) low	$001F	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111

Use TI4/O5 as either an input capture register or an output compare register, depending on the function chosen for the PA3 pin. To enable it as an input capture pin, set the I4/O5 bit in the pulse accumulator control register (PACTL) to logic level one. To use it as an output compare register, set the I4/O5 bit to a logic level zero. Refer to Section 10.1.8.1.

The TI4/O5 register pair resets to ones ($FFFF).

10.1.3 Output compare

Use the output compare (OC) function to program an action to occur at a specific time — when the 16-bit counter reaches a specified value. For each of the five output compare functions, there is a separate 16-bit compare register and a dedicated 16-bit comparator. The value in the compare register is compared to the value of the free-running counter on every bus cycle. When the compare register matches the counter value, an output compare status flag is set. The flag can be used to initiate the automatic actions for that output compare function.

To produce a pulse of a specific duration, write a value to the output compare register that represents the time the leading edge of the pulse is to occur. The output compare circuit is configured to set the appropriate output either high or low, depending on the polarity of the pulse being produced. After a match occurs, the output compare register is reprogrammed to change the output pin back to its inactive level at the next match. A value representing the width of the pulse is added to the original value, and then written to the output compare register. Because the pin state changes occur at specific values of the free-running counter, the pulse width can be controlled accurately at the resolution of the free-running counter, independent of software latency. To generate an output signal of a specific frequency and duty cycle, repeat this pulse-generating procedure.

There are four 16-bit read/write output compare registers: TOC1, TOC2, TOC3, and TOC4, and the TI4/O5 register, which functions under software control as either IC4 or OC5. Each of the OC registers is set to $FFFF on reset. A value written to an OC register is compared to the free-running counter value during each E clock cycle. If a match is found, the particular output compare flag is set in timer interrupt flag register 1 (TFLG1). If that particular interrupt is enabled in the timer interrupt mask register 1 (TMSK1), an interrupt is generated. In addition to an interrupt, a specified action can be initiated at one or more timer output pins. For OC[5:2], the pin action is controlled by pairs of bits (OMx and OLx) in the TCTL1 register. The output action is taken on each successful compare, regardless of whether or not the OCxF flag in the TFLG1 register was previously cleared.

OC1 is different from the other output compares in that a successful OC1 compare can affect any or all five of the OC pins. The OC1 output action taken when a match is found is controlled by two 8-bit registers with three bits unimplemented: the output compare 1 mask register, OC1M, and the output compare 1 data register, OC1D. OC1M specifies which port A outputs are to be used, and OC1D specifies what data is placed on these port pins.

10.1.3.1 TOC1–TOC4 — Timer output compare registers

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer output compare 1 (TOC1) high	$0016	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	1111 1111
Timer output compare 1 (TOC1) low	$0017	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111
Timer output compare 2 (TOC2) high	$0018	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	1111 1111
Timer output compare 2 (TOC2) low	$0019	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111
Timer output compare 3 (TOC3) high	$001A	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	1111 1111
Timer output compare 3 (TOC3) low	$001B	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111
Timer output compare 4 (TOC4) high	$001C	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	1111 1111
Timer output compare 4 (TOC4) low	$001D	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111

All output compare registers are 16-bit read-write. Each is initialized to $FFFF at reset. If an output compare register is not used for an output compare function, it can be used as a storage location. A write to the high-order byte of an output compare register pair inhibits the output compare function for one bus cycle. This inhibition prevents inappropriate subsequent comparisons. Coherency requires a complete 16-bit read or write. However, if coherency is not needed, byte accesses can be used.

For output compare functions, write a comparison value to output compare registers TOC1–TOC4 and TI4/O5. When TCNT value matches the comparison value, specified pin actions occur.

All TOCx register pairs reset to ones ($FFFF).

10.1.3.2 CFORC — Timer compare force register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer compare force (CFORC)	$000B	FOC1	FOC2	FOC3	FOC4	FOC5	0	0	0	0000 0000

The CFORC register allows forced early compares. FOC[1:5] correspond to the five output compares. These bits are set for each output compare that is to be forced. The action taken as a result of a forced compare is the same as if there were a match between the OCx register and the free-running counter, except that the corresponding interrupt status flag bits are not set. The forced channels trigger their programmed pin actions to occur at the next timer count transition after the write to CFORC.

The CFORC bits should not be used on an output compare function that is programmed to toggle its output on a successful compare because a normal compare that occurs immediately before or after the force can result in an undesirable operation.

FOC[1:5] — Force output compares

1 (set) – A forced output compare action will occur on the specified pin.

0 (clear) – No action.

Bits [2:0] — Not implemented; always read zero

10.1.3.3 OC1M — Output compare 1 mask register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Output compare 1 mask (OC1M)	$000C	OC1M7	OC1M6	OC1M5	OC1M4	OC1M3	0	0	0	0000 0000

Use OC1M with OC1 to specify the bits of port A that are affected by a successful OC1 compare. The bits of the OC1M register correspond to PA7–PA3.

OC1M[7:3] — Output compare masks for OC1

1 (set) – OC1 is configured to control the corresponding pin of port A.

0 (clear) – OC1 will not affect the corresponding port A pin.

Bits [2:0] — Not implemented; always read zero.

10.1.3.4 OC1D — Output compare 1 data register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Output compare 1 data (OC1D)	$000D	OC1D7	OC1D6	OC1D5	OC1D4	OC1D3	0	0	0	0000 0000

Use this register with OC1 to specify the data that is to be written to the affected pin of port A after a successful OC1 compare. When a successful OC1 compare occurs, a data bit in OC1D is written to the corresponding pin of port A for each bit that is set in OC1M.

OC1D[7:3] — Output compare data for OC1

If OC1Mx is set, data in OC1Dx is output to port A pin x on successful OC1 compares.

Bits [2:0] — Not implemented; always read zero

10.1.3.5 TCNT — Timer counter register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer count (TCNT) high	$000E	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	0000 0000
Timer count (TCNT) low	$000F	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	0000 0000

The 16-bit read-only TCNT register contains the prescaled value of the 16-bit timer. A full counter read addresses the more significant byte (MSB) first. A read of this address causes the less significant byte (LSB) to be latched into a buffer for the next CPU cycle so that a double-byte read returns the full 16-bit state of the counter at the time of the MSB read cycle.

TCNT resets to $0000.

10.1.3.6 TCTL1 — Timer control register 1

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer control 1 (TCTL1)	$0020	OM2	OL2	OM3	OL3	OM4	OL4	OM5	OL5	0000 0000

The bits of this register specify the action taken as a result of a successful OCx compare.

OM[5:2] — Output mode
OL[5:2] — Output level

OMx	OLx	Action taken on successful compare
0	0	Timer disconnected from OCx pin logic
0	1	Toggle OCx output line
1	0	Clear OCx output line to 0
1	1	Set OCx output line to 1

These control bit pairs are encoded to specify the action taken after a successful OCx compare. OC5 functions only if the I4/O5 bit in the PACTL register is clear.

10.1.3.7 TMSK1 — Timer interrupt mask register 1

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer interrupt mask 1 (TMSK1)	$0022	OC1I	OC2I	OC3I	OC4I	I4/O5I	IC1I	IC2I	IC3I	0000 0000

Use this 8-bit register to enable or inhibit the timer input capture and output compare interrupts.

Note: Bits in TMSK1 correspond bit for bit with flag bits in TFLG1. Ones in TMSK1 enable the corresponding interrupt sources.

OC1I–OC4I — Output compare x interrupt enable

1 (set) – OCx interrupt is enabled.

0 (clear) – OCx interrupt is disabled.

If the OCxI enable bit is set when the OCxF flag bit is set, a hardware interrupt sequence is requested.

I4/O5I — Input capture 4/output compare 5 interrupt enable

1 (set) – IC4/OC5 interrupt is enabled.

0 (clear) – IC4/OC5 interrupt is disabled.

When I4/O5 in PACTL is set, I4/O5I is the input capture 4 interrupt enable bit.
When I4/O5 in PACTL is zero, I4/O5I is the output compare 5 interrupt enable bit.

IC1I–IC3I — Input capture x interrupt enable

1 (set) – ICx interrupt is enabled.

0 (clear) – ICx interrupt is disabled.

If the ICxI enable bit is set when the ICxF flag bit is set, a hardware interrupt sequence is requested.

10.1.3.8 TFLG1 — Timer interrupt flag register 1

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer interrupt flag 1 (TFLG1)	$0023	OC1F	OC2F	OC3F	OC4F	I4/O5F	IC1F	IC2F	IC3F	0000 0000

Bits in this register indicate when timer system events have occurred. Coupled with the bits of TMSK1, the bits of TFLG1 allow the timer subsystem to operate in either a polled or interrupt driven system. Clear flags by writing a one to the corresponding bit position(s).

Note: Bits in TFLG1 correspond bit for bit with flag bits in TMSK1. Ones in TMSK1 enable the corresponding interrupt sources.

OC1F–OC4F — Output compare x flag

1 (set) – Counter has reached the preset output compare x value.

0 (clear) – Counter has not reached the preset output compare x value.

These flags are set each time the counter matches the corresponding output compare x values.

I4/O5F — Input capture 4/output compare 5 flag

Set by IC4 or OC5, depending on the function enabled by I4/O5 bit in PACTL

IC1F–IC3F — Input capture x flag

1 (set) – Selected edge has been detected on corresponding port pin.

0 (clear) – Selected edge has not been detected on corresponding port pin.

These flags are set each time a selected active edge is detected on the ICx input line

10.1.3.9 TMSK2 — Timer interrupt mask register 2

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer interrupt mask 2 (TMSK2)	$0024	TOI	RTII	PAOVI	PAII	0	0	PR1	PR0	0000 0000

Use this 8-bit register to enable or inhibit timer overflow and real-time interrupts. The timer prescaler control bits are included in this register.

Note: Bits in TMSK2 correspond bit for bit with flag bits in TFLG2. Ones in TMSK2 enable the corresponding interrupt sources.

TOI — Timer overflow interrupt enable

1 (set) – Timer overflow interrupt requested when TOF is set.

0 (clear) – TOF interrupts disabled.

RTII — Real-time interrupt enable (see Section 10.1.4)

1 (set) – Real time interrupt requested when RTIF is set.

0 (clear) – Real time interrupts disabled.

PAOVI — Pulse accumulator overflow interrupt enable (see Section 10.1.8)

PAII — Pulse accumulator input edge interrupt enable (see Section 10.1.8)

Bits [3, 2] — Not implemented; always read zero.

PR[1:0] — Timer prescaler select

These bits are used to select the prescaler divide-by ratio. In normal modes, PR[1:0] can only be written once, and the write must be within 64 cycles after reset. See Table 10-1 for specific timing values.

PR[1:0]	Prescaler
0 0	1
0 1	4
1 0	8
1 1	16

10.1.3.10 TFLG2 — Timer interrupt flag register 2

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer interrupt flag 2 (TFLG2)	$0025	TOF	RTIF	PAOVF	PAIF	0	0	0	0	0000 0000

Bits in this register indicate when certain timer system events have occurred. Coupled with the four high-order bits of TMSK2, the bits of TFLG2 allow the timer subsystem to operate in either a polled or interrupt driven system. Clear flags by writing a one to the corresponding bit position(s).

Note: Bits in TFLG2 correspond bit for bit with flag bits in TMSK2. Ones in TMSK2 enable the corresponding interrupt sources.

TOF — Timer overflow interrupt flag

1 (set) – TCNT has overflowed from $FFFF to $0000.

0 (clear) – No timer overflow has occurred.

RTIF — Real time (periodic) interrupt flag (see Section 10.1.4)

1 (set) – RTI period has elapsed.

0 (clear) – RTI flag has been cleared.

PAOVF — Pulse accumulator overflow interrupt flag (see Section 10.1.8)

PAIF — Pulse accumulator input edge interrupt flag (see Section 10.1.8.)

Bits [3:0] — Not implemented; always read zero

10.1.4 Real-time interrupt

The real-time interrupt (RTI) feature, used to generate hardware interrupts at a fixed periodic rate, is clocked by the 16-bit free-running counter (ST4XCK/2^{15}). See Figure 10-1. The RTI clock rate is configured by the RTR1 and RTR0 bits in the pulse accumulator control register, PACTL. The different rates available are a product of the source frequency and the value of bits RTR[1:0]. The source frequency, ST4XCK/2^{15}, can be divided by 1, 2, 4 or 8. Refer to Table 10-2 which shows examples of periodic real-time interrupt rates. The RTII bit in the TMSK2 register enables the interrupt capability.

Table 10-2 RTI periodic rates

RTR[1:0]	ST4XCK = 16MHz	ST4XCK = 8MHz	ST4XCK = 4MHz	ST4XCK = 4.2MHz	ST4XCK = xMHz
0 0	2.048ms	4.096ms	8.192ms	7.802ms	2^{15}/ST4XCK
0 1	4.096ms	8.192ms	16.384ms	15.604ms	2^{16}/ST4XCK
1 0	8.192ms	16.384ms	32.768ms	31.208ms	2^{17}/ST4XCK
1 1	16.384 ms	32.768ms	65.536ms	62.415ms	2^{18}/ST4XCK

The clock source for the RTI function is free-running clock that cannot be stopped or interrupted except by reset. This causes the time between successive RTI timeouts to be a constant that is independent of the software latency associated with flag clearing and service. For this reason, an RTI period starts from the previous timeout, not from when RTIF is cleared.

Every timeout causes the RTIF bit in TFLG2 to be set, and if RTII is set, an interrupt request is generated. After reset, one entire RTI period elapses before the RTIF flag is set for the first time. Refer to the TMSK2, TFLG2, and PACTL registers.

10.1.4.1 TMSK2 — Timer interrupt mask register 2

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer interrupt mask 2 (TMSK2)	$0024	TOI	RTII	PAOVI	PAII	0	0	PR1	PR0	0000 0000

This register contains the real-time interrupt enable bit.

Note: Bits in TMSK2 correspond bit for bit with flag bits in TFLG2. Ones in TMSK2 enable the corresponding interrupt sources.

TOI — Timer overflow interrupt enable (see Section 10.1.3.9)

1 (set) – Timer overflow interrupt requested when TOF is set.

0 (clear) – TOF interrupts disabled.

RTII — Real-time interrupt enable

1 (set) – Real time interrupt requested when RTIF is set.

0 (clear) – Real time interrupts disabled.

PAOVI — Pulse accumulator overflow interrupt enable (see Section 10.1.8)

PAII — Pulse accumulator input edge (see Section 10.1.8)

Bits[3, 2] — Not implemented; always reads zero

PR[1, 0] — Timer prescaler select (see Section 10.1.3.9)

10.1.4.2 TFLG2 — Timer interrupt flag register 2

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer interrupt flag 2 (TFLG2)	$0025	TOF	RTIF	PAOVF	PAIF	0	0	0	0	0000 0000

Bits of this register indicate the occurrence of timer system events. Coupled with the four high-order bits of TMSK2, the bits of TFLG2 allow the timer subsystem to operate in either a polled or interrupt driven system. Clear flags by writing a one to the corresponding bit position(s).

Note: Bits in TFLG2 correspond bit for bit with flag bits in TMSK2. Ones in TMSK2 enable the corresponding interrupt sources.

TOF — Timer overflow interrupt flag (see Section 10.1.3.10)

1 (set) – TCNT has overflowed from $FFFF to $0000.

0 (clear) – No timer overflow has occurred.

RTIF — Real-time interrupt flag

1 (set) – RTI period has elapsed.

0 (clear) – RTI flag has been cleared.

The RTIF status bit is automatically set at the end of every RTI period.

PAOVF — Pulse accumulator overflow interrupt flag (see Section 10.1.8)

PAIF — Pulse accumulator input edge interrupt flag (see Section 10.1.8)

Bits [3:0] — Not implemented; always read zero

10.1.4.3 PACTL — Pulse accumulator control register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Pulse accumulator control (PACTL)	$0026	0	PAEN	PAMOD	PEDGE	0	I4/O5	RTR1	RTR0	0000 0000

The RTR[1:0] bits in this register select the rate for the RTI system. The remaining bits control the pulse accumulator and IC4/OC5 functions.

Bits [7, 3] — Not implemented; always read zero

PAEN — Pulse accumulator system enable (see Section 10.1.8)

1 (set) – Pulse accumulator enabled.

0 (clear) – Pulse accumulator disabled.

PAMOD — Pulse accumulator mode (see Section 10.1.8)

1 (set) – Gated time accumulation mode.

0 (clear) – Event counter mode.

PEDGE — Pulse accumulator edge control (see Section 10.1.8)

This bit has different meanings depending on the state of the PAMOD bit.

I4/O5 — Input capture 4/output compare 5 (see Section 10.1.8)

1 (set) – Input capture 4 function is enabled (no OC5).

0 (clear) – Output compare 5 function is enabled (no IC4).

RTR[1:0] — RTI interrupt rate select

These two bits determine the rate at which the RTI system requests interrupts. The RTI system is driven by the ST4XCK/2^{15} clock rate that is compensated so it is independent of the timer prescaler. These two control bits select an additional division factor. Refer to Table 10-2.

10.1.5 Slow real-time interrupt

The slow real-time interrupt (SRTI) generates real-time interrupts with intervals of up to 1s. The SRTI is clocked by the EXTAL oscillator frequency, and is generated by setting two bits in the RTCTL register in order to select the nominal divide rate and the corresponding interrupt rate. The divide ratio can be adjusted between -27 and +31 ppm. of the selected value by means of the four bits in RTREG. The SRTI can therefore be adjusted with an approximate accuracy of better than 4ppm. For further information see Section 10.1.5.3.

The SRTI real-time interrupt is hardwired to the nonmaskable XIRQ interrupt line. Depending on the mode there are two configurations:

- RUN mode: The SRTI interrupt is masked by the RTI mask bit in the RTCTL register. The SRTI interrupt can be inhibited depending on this bit.
- WAIT/STOP mode: It is not possible to inhibit the SRTI interrupt with the RTI mask bit.

The following points should be considered when using or inhibiting the SRTI interrupt:

- When using the SRTI interrupt and also WAIT and RUN mode in the application, it should be enabled by setting the RTI bit in the RTCTL register, and the XIRQ should be enabled by clearing the X bit in the CCR. The SRTI always generates an XIRQ interrupt which must be serviced by the interrupt service routine.
- When the SRTI interrupt is not used in the application which uses WAIT and RUN mode, it should be disabled by leaving the X-bit in the CCR set high (XIRQ not enabled) when coming out of reset.
- Precautions *always* need to be taken when going into STOP mode. When an SRTI interrupt is pending (RTF flag is set) at the same time as STOP instruction occurs, the MCU will not go into STOP mode. This is due firstly to the fact that the XIRQ interrupt always does a recovery from STOP mode even if the X-bit in the CCR register is set, and the SRTI interrupt is not maskable by the RTI bit in STOP and WAIT modes. *Before going into STOP mode, it should always be ensured that the RTF flag is not set, and that there is no pending SRTI interrupt.*

Masking is performed by setting the interrupt mask RTI in the RTCTL register. The SRTI interrupt flag bit indicates when interrupt conditions have occurred. On-chip hardware sets this bit as a result of the occurrence of the corresponding condition. To clear this bit in the RTCTL register, a write operation must be performed to the RTCTL register where the RTF bit to be cleared by the write is set to one.

The SRTI clock, which is directly derived from the EXTAL oscillator frequency, is the clock source for the SRTI counter. The RTCK[2,1] bits in the RTCTL register determine the nominal divide rate and the interrupt rate for the 4.2MHz EXTAL frequency. Refer to Table 10-3. In normal modes, these bits can be written to once only. After setting these bits, the SRTI down counter will be reset. It starts counting with its programmed value. The operation of the SRTI down counter chain is shown in Figure 10-3. The SRTI high and low part counter overflow flag bits RTHF and RTLF indicate when an overflow condition after the 4th and 12th bit of the counter chain has occurred.

On-chip hardware sets this bit as a result of the occurrence of the corresponding condition. To clear these bits in the RTCTL register, a write operation must be performed to the RTCTL register where the RTHF and RTLF bits to be cleared by the write are set to one. The main purpose of these bits is testing the SRTI counter.

Each time the counter reaches the value zero an interrupt RTI occurs. In addition, the least significant four bits of the counter are preset again with the adjustment value in the RTREG register. If the RTREG value is positive, the low and the high counter part is preset with $4000, and if it is negative it will be preset with $3FFF.

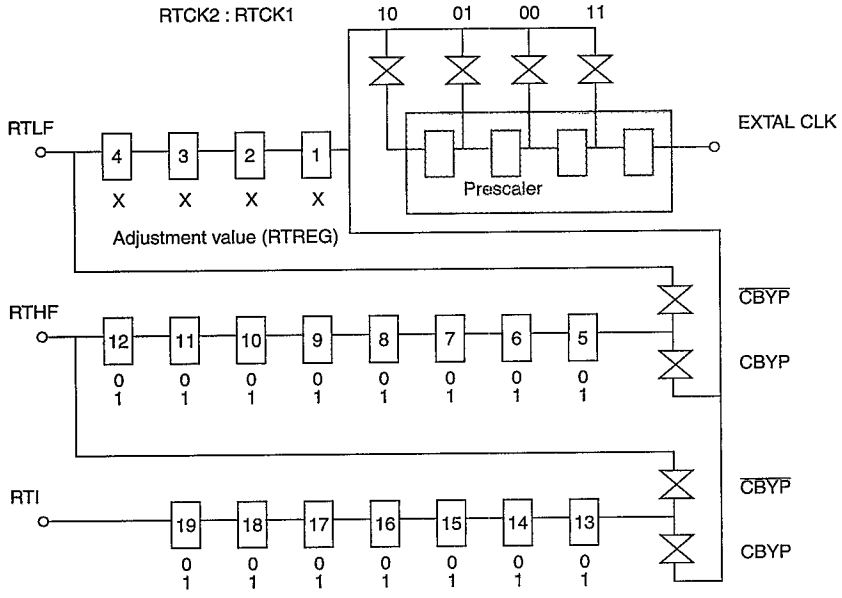


Figure 10-3 SRTI down counter chain

10.1.5.1 SRTI test features

In special modes the SRTI counter will be tested by setting the CBYP bit in the TEST1 register. The 23-bit down counter of the SRTI is divided into one 4-bit counter, one 8-bit counter and one 7-bit counter. The remaining four bits are used for the prescaler. The prescaled SRTI clock drives the three counters directly. An overflow of each counter part sets the corresponding flag in the

RTCTL register. In special modes the RTREG programming register and the prescaler bits in the RTCTL register can be written any time. Each write access to the RTREG register causes a preset of the SRTI counter with the adjustment value given in this register. When the CBYP bit is set in the TEST1 register in special mode, the SRTI counter will not be reset after an SRTI interrupt occurs. This ensures that all the counter bits will be activated during test.

10.1.5.2 RTCTL — SRTI control register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
SRTI programming register (RTREG)	$0034	RTF	RTI	0	0	RTCK2	RTCK1	RTHF	RTLF	0000 0000

RTF — Slow real time interrupt flag

1 (set) – The SRTI period has elapsed.

0 (clear) – The SRTI period has not elapsed since flag RTF flag was last cleared.

To clear this bit, a write operation must be performed to the RTCTL register where the RTF bit to be cleared by the write is set to one.

RTI — Slow real time interrupt enable

1 (set) – Slow real time interrupt requested when RTF is set.

0 (clear) – Slow real time interrupt disabled.

RTCK[2:1] — Prescaler for SRTI clock

These two bits determine the nominal divide rate and the interrupt rate for the 4.194304 crystal frequency, as shown in Table 10-3. The nominal divide rate can be adjusted using the SRTI[3:0] bits in the RTREG register. (see Section 10.1.5.3).

Table 10-3 SRTI interval divide and interrupt rates

RTCK2	RTCK1	Nominal divide rate	Interrupt rate (s)
0	0	2 097 152	0.5
0	1	1 048 576	0.25
1	0	524 288	0.125
1	1	4 194 304	1.0

RTHF — SRTI high part counter overflow flag

1 (set) – An overflow condition at the 12th bit of the counter chain has occurred.

0 (clear) – An overflow condition at the 12th bit of the counter chain has not occurred.

The SRTI high part counter overflow flag bit indicates when an overflow condition at the 12th bit of the counter chain has occurred. To clear this bit, a write operation must be performed on the RTCL register, where the RTHF bit to be cleared by the write is set to one.

RTLF — SRTI low part counter overflow flag

1 (set) – An overflow condition after the 4th bit of the counter chain has occurred.

0 (clear) – An overflow condition after the 4th bit of the counter chain has not occurred.

The SRTI low part counter overflow flag bit indicates when an overflow condition after the 4th bit of the counter chain has occurred. To clear this bit, a write operation must be performed on the RTCL register, where the RTLF bit to be cleared by the write is set to one.

10.1.5.3 RTREG — Slow real-time interrupt programming register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
SRTI programming register (RTREG)	$0033	SRTI3	SRTI2	SRTI1	SRTI0	0	0	0	0	1111 0000

The four bit value in this register determines the division rates for the SRTI. The division rates can be adjusted within the range of -27ppm and +31ppm of the nominal value. The nominal value is determined by the clock select bits RTCK1 and RTCK2 in RTCTL. Refer to Table 10-4.

10

10.1.6 Computer operating properly (COP) watchdog function

The clocking chain for the COP function is tapped off from the main timer divider chain (ST4XCK/2^{17}). The CR[1:0] bits in the OPTION register and the NOCOP bit in the CONFIG register control and configure the COP function. One additional register, COPRST, is used to arm and clear the COP watchdog reset system. Refer to Section 5 for a more detailed discussion of the COP function.

Table 10-4 SRTI periodic rates

RTREG	RTCK2 : RTCK1				Deviation (ppm)
	1 : 1	0 : 0	0 : 1	1 : 0	
$8x	4 194 192	2 097 096	1 048 548	524 274	-27
$9x	4 194 208	2 097 104	1 048 552	524 276	-23
$Ax	4 194 224	2 097 112	1 048 556	524 278	-19
$Bx	4 194 240	2 097 120	1 048 560	524 280	-15
$Cx	4 194 256	2 097 128	1 048 564	524 282	-11
$Dx	4 194 272	2 097 136	1 048 568	524 284	-8
$Ex	4 194 288	2 097 144	1 048 572	524 286	-4
$Fx	4 194 304	2 097 152	1 048 576	524 288	0
$0x	4 194 320	2 097 160	1 048 580	524 290	+4
$1x	4 194 336	2 097 168	1 048 584	524 292	+8
$2x	4 194 352	2 097 176	1 048 588	524 294	+11
$3x	4 194 368	2 097184	1 048 592	524 296	+15
$4x	4 194 384	2 097 192	1 048 596	524 298	+19
$5x	4 194 400	2 097 200	1 048 600	524 300	+23
$6x	4 194 416	2 097 208	1 048 604	524 302	+27
$7x	4 194 432	2 097 216	1 048 608	524 304	+31

10.1.7 Window COP (WCOP) watchdog function

The WCOP operation is similar to that of the COP function, but a greater system security is provided due to the fact that the WCOP counter may only be reset by software within a predefined time window. A counter reset outside of this window immediately causes a system reset. The CR[1:0] bits in the OPTION register and the NWCOP bit in the CONFIG register control and configure the WCOP function. Refer to Section 5 for a more detailed discussion of the WCOP function.

10.1.8 Pulse accumulator

The MC68HC11KG4 has an 8-bit counter that can be configured to operate either as a simple event counter, or for gated time accumulation, depending on the state of the PAMOD bit in the PACTL register. Refer to the pulse accumulator block diagram, Figure 10-4.

In the event counting mode, the 8-bit counter is clocked to increasing values by an external pin. The maximum clocking rate for the external event counting mode is the E clock divided by two. In gated time accumulation mode, a free-running ST4XCK/2^8 signal drives the 8-bit counter, but only while the external PAI pin is activated. Refer to Table 10-5. The pulse accumulator counter can be read or written at any time.

Table 10-5 Pulse accumulator timing

ST4XCK	ST4XCK/4 clock	Cycle time	2^8/ST4XCK	PACNT overflow
4.0 MHz	1.0 MHz	1000 ns	64 µs	16.384 ms
8.0 MHz	2.0 MHz	500 ns	32 µs	8.192 ms
12.0 MHz	3.0 MHz	333 ns	21.33 µs	5.461 ms
16.0 MHz	4.0 MHz	250 ns	16.0 µs	4.096 ms
16.8MHz	4.2MHz	238ns	15.24µs	3.901ms

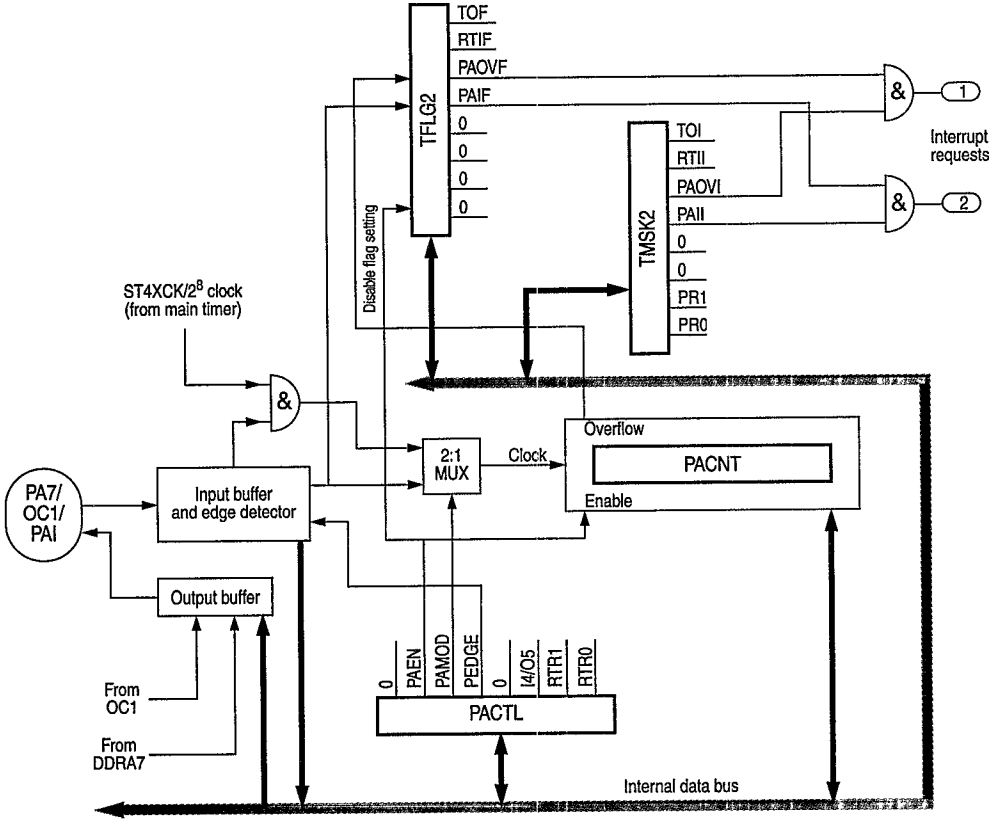


Figure 10-4 Pulse accumulator block diagram

Pulse accumulator control bits are located within the PACTL, TMSK2 and TFLG2 registers, as described in the following paragraphs.

10.1.8.1 PACTL — Pulse accumulator control register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Pulse accumulator control (PACTL)	$0026	0	PAEN	PAMOD	PEDGE	0	I4/O5	RTR1	RTR0	0000 0000

Four of this register's bits control an 8-bit pulse accumulator system. Another bit enables either the OC5 function or the IC4 function, while two other bits select the rate for the real-time interrupt system.

Bits [7, 3] — Not implemented; always read zero

PAEN — Pulse accumulator system enable

1 (set) – Pulse accumulator enabled.

0 (clear) – Pulse accumulator disabled.

PAMOD — Pulse accumulator mode

1 (set) – Gated time accumulation mode.

0 (clear) – Event counter mode.

PEDGE — Pulse accumulator edge control

This bit has different meanings depending on the state of the PAMOD bit, as shown:

PAMOD	PEDGE	Action of clock
0	0	PAI falling edge increments the counter.
0	1	PAI rising edge increments the counter.
1	0	A zero on PAI inhibits counting.
1	1	A one on PAI inhibits counting.

I4/O5 — Input capture 4/output compare 5

1 (set) – Input capture 4 function is enabled (no OC5).

0 (clear) – Output compare 5 function is enabled (no IC4)

RTR[1:0] — RTI interrupt rate selects (see Section 10.1.4)

10.1.8.2 PACNT — Pulse accumulator count register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Pulse accumulator count (PACNT)	$0027	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	undefined

This 8-bit read/write register contains the count of external input events at the PAI input, or the accumulated count. In gated time accumulation mode, PACNT is readable even if PAI is not active. The counter is not affected by reset and can be read or written at any time. Counting is synchronized to the internal PH2 clock so that incrementing and reading occur during opposite half cycles.

10.1.8.3 Pulse accumulator status and interrupt bits

The pulse accumulator control bits, PAOVI and PAII, PAOVF and PAIF are located within timer registers TMSK2 and TFLG2.

10.1.8.4 TMSK2 — Timer interrupt mask 2 register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer interrupt mask 2 (TMSK2)	$0024	TOI	RTII	PAOVI	PAII	0	0	PR1	PR0	0000 0000

10.1.8.5 TFLG2 — Timer interrupt flag 2 register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer interrupt flag 2 (TFLG2)	$0025	TOF	RTIF	PAOVF	PAIF	0	0	0	0	0000 0000

PAOVI and PAOVF — Pulse accumulator interrupt enable and overflow flag

The PAOVF status bit is set each time the pulse accumulator count rolls over from $FF to $00. To clear this status bit, write a one in the corresponding data bit position (bit 5) of the TFLG2 register. The PAOVI control bit allows the pulse accumulator overflow to be configured for polled or interrupt-driven operation and does not affect the state of PAOVF. When PAOVI is zero, pulse accumulator overflow interrupts are inhibited, and the system operates in a polled mode, which requires that PAOVF be polled by user software to determine when an overflow has occurred. When the PAOVI control bit is set, a hardware interrupt request is generated each time PAOVF is set. Before leaving the interrupt service routine, software must clear PAOVF.

PAII and PAIF — Pulse accumulator input edge interrupt enable and flag

The PAIF status bit is automatically set each time a selected edge is detected at the PA7/PAI/OC1 pin. To clear this status bit, write to the TFLG2 register with a one in the corresponding data bit position (bit 4). The PAII control bit allows the pulse accumulator input edge detect to be configured for polled or interrupt-driven operation but does not affect setting or clearing the PAIF bit. When PAII is zero, pulse accumulator input interrupts are inhibited, and the system operates in a polled mode. In this mode, the PAIF bit must be polled by user software to determine when an edge has occurred. When the PAII control bit is set, a hardware interrupt request is generated each time PAIF is set. Before leaving the interrupt service routine, software must clear PAIF.

11
ANALOG-TO-DIGITAL CONVERTER

The analog-to-digital converter system consists of a single 10-bit successive approximation type converter and a 16-channel multiplexer. Seven of the channels are connected to pins on the MC68HC11KG4 (port E), five are unused and the remaining four channels are dedicated to internal reference points or test functions. The A/D converter shares input pins with port E as follows:

Pin	Alternative function
PE0	AN2
PE1	AN3
PE2	AN4
PE3	AN5
PE4	AN6
PE5	AN7
PE6	AN8

There are eight 10-bit result registers, and control logic allows for four or eight consecutive conversions before stopping, or for conversions to continue with the newest conversion overwriting the oldest result register. Also, conversions can be performed several times on a single selected channel, or consecutively on a selected group of four channels. In addition, the control logic can convert eight channels and then stop, or convert continuously.

Two dedicated lines (VRH and VRL) are provided for the reference voltage inputs. These pins may be connected to a separate or isolated power supply to ensure full accuracy of the A/D conversion. Furthermore, the A/D converter supply pins AVDD and AVSS allow the user to isolate the A/D voltage supply from the main VDD/VSS supply lines. This reduces the effects of noise from the CPU core in the A/D converter, thus improving accuracy.

The 10-bit A/D converter accepts analog inputs ranging from V_{RL} to V_{RH}. Smaller input ranges can also be obtained by adjusting V_{RL} and V_{RH} to the desired upper and lower limits. Conversion is specified and tested for V_{RL} = 0 volts and V_{RH} = 5 volts. The A/D system can be operated with V_{RL} below V_{DD} and/or V_{RL} above V_{SS} as long as V_{RH} is above V_{RL} by enough to support the conversions (2.5 to 5.0 volts).

Each set of four conversions takes 144 cycles of the E-clock, provided that E is greater than or equal to 750 kHz. If E is less than 750 kHz, an internal R-C oscillator, which is nominally 1.5 MHz, must be used for the A/D conversion clock. When the internal R-C oscillator is being used as the conversion clock, the conversion complete flag (CCF) must be used to determine when a conversion sequence has been completed. When using the internal R-C oscillator for A/D conversions, the sample and conversion process runs at the nominal 1.5 MHz rate; however, the conversion results must be transferred to the MCU result registers synchronously with the MCU E-clock, so conversion time is limited to a maximum of one channel per E-clock cycle. Alternatively, if the R-C oscillator is not being used and E is greater than 1.5 MHz, the conversion frequency can be halved to E/2 under control of the ADER bit in the ADFRQ register. Note that in this operating mode, each set of four conversions takes 288 cycles of the E clock.

Two control bits in the OPTION register control the basic configuration of the A/D system. The A/D power-up bit (ADPU) allows the system to be disabled, resulting in reduced power consumption when the A/D system is not being used. Any conversion which is in process when ADPU is written to zero will be aborted. A delay of typically 100 microseconds is required after turning on the A/D (by writing ADPU from 0 to 1) for the analog and comparator sections to stabilize. The CSEL bit is used to select either the internal R-C oscillator or the MCU E-clock as the A/D system clock source.

11.1 Conversion process

The A/D converter is ratiometric. One LSB is equal to (V_{RH}–V_{RL})/1024. An input voltage which is lower than V_{RH} and higher than (V_{RH} – (1.5 * LSB)) converts ideally to $FFC0 (full scale), and an input voltage which is less than (0.5 * LSB) converts ideally to $0000. The maximum analog input voltage should not exceed (1.125*VRH). Also, the six least significant bits should always read zero. For ratiometric conversions, the source of each analog input should use V_{RH} as the supply voltage and be referenced to V_{RL}.

The A/D reference inputs are applied to a precision internal digital-to-analog converter. Control logic drives this D/A and the analog output is successively compared to the selected analog input which was sampled at the beginning of the conversion time. The conversion process is monotonic with no missing codes.

11.2 Channel assignments

A multiplexer allows the single A/D converter to select one of sixteen analog signals. Seven of these channels are supported on Port E input pins. Of the nine other channels, five are reserved for future use and four are for internal reference points and testing purposes. Table 11-1 shows the signals selected by the channel select bits (CD, CC, CB, CA) in the ADCTL register. The CONV8 bit selects either four or eight conversions. All "reserved" channels are connected to V_{RL}.

Table 11-1 Channel assignments

CD	CC	CB	CA	Channel signal	CONV8 = 0 Result in register if		CONV8 = 1 Result in register if	
					MULT = 1	MULT = 0	MULT = 1	MULT = 0
0	0	0	0	AN2 (on PE0)	ADR5	ADR5 – 8	ADR1	ADR1 – 8
0	0	0	1	AN3 (on PE1)	ADR6	ADR5 – 8	ADR2	ADR1 – 8
0	0	1	0	AN4 (on PE2)	ADR7	ADR5 – 8	ADR3	ADR1 – 8
0	0	1	1	AN5 (on PE3)	ADR8	ADR5 – 8	ADR4	ADR1 – 8
0	1	0	0	AN6 (on PE4)	ADR5	ADR5 – 8	ADR5	ADR1 – 8
0	1	0	1	AN7 (on PE5)	ADR6	ADR5 – 8	ADR6	ADR1 – 8
0	1	1	0	AN8 (on PE6)	ADR7	ADR5 – 8	ADR7	ADR1 – 8
0	1	1	1	Reserved	ADR8	ADR5 – 8	ADR8	ADR1 – 8
1	0	0	0	Reserved	ADR5	ADR5 – 8	ADR1	ADR1 – 8
1	0	0	1	Reserved	ADR6	ADR5 – 8	ADR2	ADR1 – 8
1	0	1	0	Reserved	ADR7	ADR5 – 8	ADR3	ADR1 – 8
1	0	1	1	Reserved	ADR8	ADR5 – 8	ADR4	ADR1 – 8
1	1	0	0	VRH(1)	ADR5	ADR5 – 8	ADR5	ADR1 – 8
1	1	0	1	VRL(1)	ADR6	ADR5 – 8	ADR6	ADR1 – 8
1	1	1	0	VRH /2(1)	ADR7	ADR5 – 8	ADR7	ADR1 – 8
1	1	1	1	Test (reserved)(1)	ADR8	ADR5 – 8	ADR8	ADR1 – 8

(1) Used for factory testing.

11.3 Single channel operation

Single channel operation is selected by writing a zero to the MULT bit in the A/D control and status register (ADCTL). This mode has four variations, which can be selected using the CONV8 and SCAN bits in the ADCTL register. In the first two variations, the CONV8 bit is clear and the single selected channel is converted four consecutive times. In the second two variations, the CONV8 bit is set and the single selected channel is converted eight consecutive times. The state of the SCAN bit determines whether continuous or single scanning is selected. The channel is selected by the CD – CA bits in the ADCTL register.

11.3.1 4-conversion, single scan

Whichever port E pin is selected, the first result will be stored in the ADR5 result register and the fourth result will be stored in the ADR8 register. After the fourth conversion is complete all conversion activity is halted until a new conversion command is written to the ADCTL control register.

11.3.2 4-conversion, continuous scan

Conversions continue to be performed on the selected channel with the fifth conversion being stored in the ADR5 register (overwriting the first conversion result), the sixth conversion overwrites ADR6, the seventh overwrites ADR7, and so on continuously. Using this variation, the data in any result register is at most four conversion times old.

11.3.3 8-conversion, single scan

The result of the first conversion will be placed in result register ADR1, while the result of the eighth conversion will be placed in result register ADR8. After the eighth conversion is complete all conversion activity is halted until a new conversion command is written to the ADCTL control register.

11.3.4 8-conversion, continuous scan

Conversions continue to be performed on the selected channel with the ninth conversion being stored in the ADR1 register (overwriting the first conversion result), the tenth conversion overwrites ADR2, the eleventh overwrites ADR3, and so on continuously. Using this variation, the data in any result register is at most eight conversion times old.

11.4 Multiple channel operation

Multiple channel operation is selected by writing a one to the MULT bit in the A/D control and status register (ADCTL). This mode has four variations, which can be selected using the CONV8 and SCAN bits in the ADCTL register. In the first two variations, the CONV8 bit is clear and a group of four channels is selected. (In this multiple channel mode, only the two most significant bits of the channel address (CD and CC) are decoded.) In the second two variations, the CONV8 bit is set and a group of eight channels is converted, depending on the state of the CD bit. The state of the SCAN bit determines whether continuous or single scanning is selected.

11.4.1 4-channel single scan

A group of four channels is selected by the CD and CC bits in the ADCTL register. For details of channel assignments, see Table 11-1. The first result is stored in the ADR5 result register, the second in ADR6, the third in ADR7 and the fourth in the ADR8 register. After the fourth conversion is complete, all conversion activity is halted until a new conversion command is written to the ADCTL control register.

11.4.2 4-channel continuous scan

Conversions continue to be performed on the selected group of channels with the fifth conversion being stored in the ADR5 register (replacing the earlier conversion result for the first channel in the group), the sixth conversion overwrites ADR6, the seventh overwrites ADR7, and so on, continuously. Using this second variation the data in any result register is, at most, four conversion times old.

11.4.3 8-channel single scan

When CONV8 is set and MULT is set, then a group of eight channels is converted. The group is selected by the CD bit. Each of the channels is converted and the result is placed in a separate result register. Port E pin 0 uses result register ADR1, Port E pin 1 uses result register ADR2 and so on. Each channel is converted once, then all conversion activity is halted until a new conversion command is written to the ADCTL control register.

11.4.4 8-channel continuous scan

Conversions continue to be performed on the eight channels with the ninth conversion being stored in the ADR1 register (replacing the earlier conversion result for the first channel in the group), the tenth conversion overwrites ADR2, the eleventh overwrites ADR3, and so on, continuously. Using this second variation the data in any result register is, at most, eight conversion times old.

11.5 Power-up and clock select

A/D power up is controlled by the ADPU bit in the OPTION register. When ADPU is cleared, power to the A/D system is removed. When ADPU is set, the A/D system is enabled. A delay of 100 microseconds is required after turning on the A/D converter, to allow the analog bias voltages to stabilize.

Clock select is controlled by the CSEL bit in the OPTION register. When CSEL is cleared, the A/D system uses the system E-clock. When CSEL is set, the A/D system uses an internal R-C clock source, nominally 1.5 MHz, in which case the R-C internal clock should be selected. A delay of 10 milliseconds is required, after changing CSEL from zero to one, to allow the R-C oscillator to start and internal bias voltages to settle.

When the A/D system is operating with the MCU E-clock, all switching and comparator operations are synchronized with the MCU clock. This allows the comparator results to be sampled at quiet clock times to minimise the effect of internal switching noise. As the internal R-C oscillator is

asynchronous with respect to the MCU clock, internal switching noise is more likely to affect the overall accuracy of the A/D results, when using this oscillator, than when using the E-clock.

11.6 Operation in STOP and WAIT modes

If a conversion sequence is still in process when the MC68HC11KG4 enters the STOP or WAIT mode, the conversion of the current channel is suspended. When the MCU resumes normal operation, that channel is re-sampled and the conversion sequence resumes. As the MCU exits the WAIT mode, the A/D circuits are stable and valid results can be obtained on the first conversion. However, in STOP mode the comparator, charge pump and R-C oscillator are turned off. If the MC68HC11KG4 exits the STOP mode with a delay (as is normal), there will automatically be enough time for these circuits to stabilize before the first conversion. If the MC68HC11KG4 exits the STOP mode with no delay (DLY bit in OPTION register equal to zero) and a stable external clock supplied, the user must allow about 100 microseconds for the A/D circuitry to stabilize and to avoid invalid results.

11.7 Registers

11.7.1 ADCTL — A/D control and status register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
A/D control & status register (ADCTL)	$0030	CCF	CONV8	SCAN	MULT	CD	CC	CB	CA	Undefined

This register can be read and written at any time. Note that a write to this register will always clear the CCF bit.

11

CCF — Conversions complete flag

This flag bit is set automatically after an A/D conversion cycle (four or eight conversions, depending on which conversion mode is selected). If a continuous scan mode is selected, the CCF flag will become set after the first time all four (or eight) registers have been updated, and it will remain set until the ADCTL register is again written. Each time the ADCTL register is written, this bit is automatically cleared, any current conversion is aborted and a new conversion sequence is started.

CONV8 — Convert 8/convert 4 select bit

1 (set) – Convert 8 channels or one channel 8 times, (uses all 8 result registers).

0 (clear) – Convert 4 channels or one channel 4 times (uses 4 result registers).

SCAN — Continuous scan control

1 (set) – Convert continuously.

0 (clear) – Perform selected number of conversions (4 or 8) and stop.

MULT — Multiple channel/single channel control

1 (set) – Convert the channels in the selected group.

0 (clear) – Convert single channel selected.

CD, CC, CB, CA — Channel select bits

When 4-conversion (CONV8 = 0) and multiple channel (MULT=1) modes are selected, the CB and CA bits have no meaning or effect, and the CD and CC bits specify which of four groups of four channels are to be converted. When 8-conversion (CONV8 = 1) and multiple channel (MULT=1) modes are selected, the CC, CB and CA bits have no meaning or effect. Refer to Table 11-1 for a list of the A/D channel assignments.

11.7.2 ADFRQ — A/D converter frequency select register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
A/D frequency select register (ADFRQ)	$0032	0	0	0	0	0	0	0	ADER	0000 0000

This register can be read and written at any time.

Bits [7:1] — Not implemented; always read zero.

ADER — A/D frequency select

1 (set) – E/2 clock is used for A/D conversions.

0 (clear) – E clock is used for A/D conversions.

11.7.3 ADR1 — ADR8 A/D result registers

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
A/D result 1 (ADR1) high	$0040	(Bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(8)	undefined
A/D result 1 (ADR1) low	$0041	(7)	(6)	0	0	0	0	0	0	uu00 0000
A/D result 2 (ADR2) high	$0042	(Bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(8)	undefined
A/D result 2 (ADR2) low	$0043	(7)	(6)	0	0	0	0	0	0	uu00 0000
A/D result 3 (ADR3) high	$0044	(Bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(8)	undefined
A/D result 3 (ADR3) low	$0045	(7)	(6)	0	0	0	0	0	0	uu00 0000
A/D result 4 (ADR4) high	$0046	(Bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(8)	undefined
A/D result 4 (ADR4) low	$0047	(7)	(6)	0	0	0	0	0	0	uu00 0000
A/D result 5 (ADR5) high	$0048	(Bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(8)	undefined
A/D result 5 (ADR5) low	$0049	(7)	(6)	0	0	0	0	0	0	uu00 0000
A/D result 6 (ADR6) high	$004A	(Bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(8)	undefined
A/D result 6 (ADR6) low	$004B	(7)	(6)	0	0	0	0	0	0	uu00 0000
A/D result 7 (ADR7) high	$004C	(Bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(8)	undefined
A/D result 7 (ADR7) low	$004D	(7)	(6)	0	0	0	0	0	0	uu00 0000
A/D result 8 (ADR8) high	$004E	(Bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(8)	undefined
A/D result 8 (ADR8) low	$004F	(7)	(6)	0	0	0	0	0	0	uu00 0000

The eight 10-bit result registers are read-only; they can be read at any time, but a write will have no meaning or effect. In each result register, the eight high order bits are in one address location and the remaining two low order bits are in bit locations 6 and 7 of the following address. The six unused bits will always read as zeros. This allows a double byte read to be performed without having to adjust the result.

12
DIGITAL-TO-ANALOG CONVERTER

The digital-to-analog converter system converts an 8-bit binary value into a monotonic analog voltage level which is output on the PE7 pin. The converter is a resistive ladder type which sources very low current (0.1µA at 5V) to maintain accuracy.

The converter creates an analog voltage level between AVSS and AVDD. This voltage is proportional to the digital values between $00 and $FF.

12.1 DA1 — Digital to analog data register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
DA1 (Digital to analog data)	$007E	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	0000 0000

The value written to this register is converted into an analog voltage such that $00=AVSS and $FF=(AVDD–AVDD/256); a total of 256 incremental voltages represents the inputs from $00 to $FF. The worst case settling time following a write to this register is 5µs. Note that during STOP mode the D/A output is open to prevent additional current drain.

12.2 DACON — Digital to analog control register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
DACON (Digital-to-analog control)	$007D	0	0	0	0	0	0	0	DAE1	0000 0000

This register may be read and written at any time.

DAE1 — Digital to analog enable

1 (set) – An analog value proportional to the digital value in the DA1 register is output on PE7.

0 (clear) – The digital-to-analog converter is disabled and PE7 is a general purpose input.

When DAE1 is set, pin PE7 is forced to be an output.

A
ELECTRICAL SPECIFICATIONS

This section contains the electrical specifications and associated timing information for the standard supply voltage (V_{DD} = 5V ± 5%) MC68HC11KG4 variants.

A.1 Maximum ratings

Rating	Symbol	Min.	Typ.	Max.	Unit
Supply voltage, power driver parts(1)	PV_{DD}	V_{DD}	8.5	9.5	V
Supply voltage, logic parts(2)	V_{DD}	– 0.5	5	7	V
Input voltage	V_{IN}	V_{SS} – 0.5	—	V_{DD} + 0.5	V
Operating temperature range	T_A	– 40	—	+85	°C
Junction temperature	T_J	—	—	+115	°C
Storage temperature range	T_{STG}	– 55	—	+150	°C
Current drain per pin (excluding V_{DD} and V_{SS} and port H and K pins)	I_D	—	—	25	mA
Current drain per pin for ports H and port K pins (T_A = – 40 C)	$I_{D, HK}$	—	—	50	mA
Current drain per pin for ports H and port K pins (T_A = 25 C)	$I_{D, HK}$	—	—	42	mA
Current drain per pin for port H and port K pins (T_A = 85 C)	$I_{D, HK}$	—	—	37	mA
Total power consumption for ports H and K (power drivers) and ports B, C, F (LED ports) (T_A = 85 C)	$P_{H, K, B, C, F}$	—	—	256	mW

(1) The maximum supply voltage for the power driver parts is under normal operating conditions as well as under short circuit conditions.

(2) All voltages are with respect to V_{SS}

Note: This device contains circuitry designed to protect against damage due to high electrostatic voltages or electric fields. However, it is recommended that normal precautions be taken to avoid the application of any voltages higher than those given in the maximum ratings table to this high impedance circuit. For maximum reliability all unused inputs should be tied to either V_{SS} or V_{DD}.

A.2 Thermal characteristics and power considerations

The average chip junction temperature, T_J, in degrees Celsius can be obtained from the following equation:

$$T_J = T_A + (P_D \bullet \theta_{JA}) \quad [1]$$

where:

T_A = Ambient temperature (°C)

θ_{JA} = Package thermal resistance, junction-to-ambient (°C/W)

P_D = Total power dissipation = $P_{INT} + P_{I/O}$ (W) + $P_{H,K}$

P_{INT} = Internal chip power = $I_{DD} \bullet V_{DD}$ (W)

$P_{I/O}$ = Power dissipation on input and output pins (user determined) excluding ports H and K.

An approximate relationship between P_D and T_J (if $P_{I/O}$ is neglected) is:

$$P_D = \frac{K}{T_J + 273} \quad [2]$$

Solving equations [1] and [2] for K gives:

$$K = P_D \bullet (T_A + 273) + \theta_{JA} \bullet P_D^2 \quad [3]$$

where K is a constant for a particular part. K can be determined by measuring P_D (at equilibrium) for a known T_A. Using this value of K, the values of P_D and T_J can be obtained for any value of T_A, by solving the above equations. The package thermal characteristics are shown below:

Characteristics	Symbol	Value	Unit
Thermal resistance – 100-pin TQFP package	θ_{JA}	54	°C/W

$P_{H,K}$ = Ports H and K power dissipation (user determined). The power dissipation of a single port bit can be calculated using:

$$P_{H,K} = I_L^2 \bullet R_{ONP} \quad [4]$$

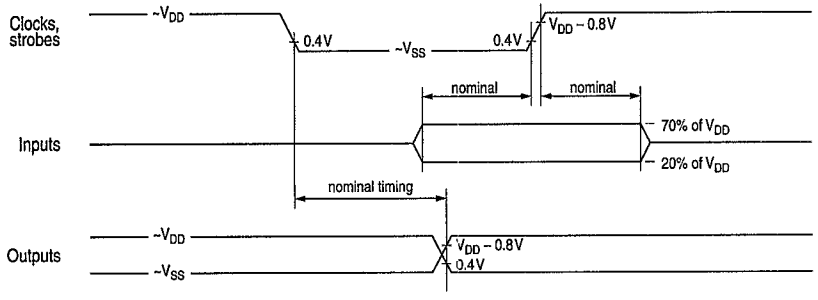
for the PMOS transistor, and:

$$P_{H,K} = I_L^2 \bullet R_{ONN} \quad [5]$$

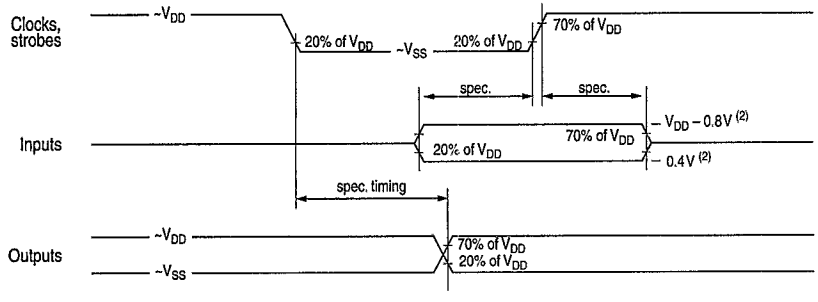
for the NMOS transistor. [6]

The 'on' resistances R_{ONP} and R_{ONN} can be obtained from the port H and K electrical characteristics. [7]

A.3 Test methods



(b) DC testing



(c) AC testing

Notes:

(1) Full test loads are applied during all DC electrical tests and AC timing measurements.

(2) During AC timing measurements, inputs are driven to 0.4V and V_{DD} – 0.8V; timing measurements are taken at the 20% and 70% of V_{DD} points.

Figure A-1 Test methods

A

A.4 DC electrical characteristics

(V_{DD} = 5.0 Vdc ± 5%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, unless otherwise noted)

Characteristic	Symbol	Min.	Max.	Unit
Output voltage[1] (I_{LOAD} = ± 10 µA):				
All outputs	V_{OL}	—	0.1	V
All outputs except $\overline{\text{RESET}}$ & MODA	V_{OH}	V_{DD} – 0.1	—	V
Output high voltage[1] (I_{LOAD} = –0.8mA, V_{DD} =4.5V):				
All outputs except XTAL, $\overline{\text{RESET}}$ & MODA	V_{OH}	V_{DD} – 0.8	—	V
Output low voltage[1] (I_{LOAD} = +1.6mA):				
All outputs except XTAL	V_{OL}	—	0.4	V
Input high voltage:				
All inputs except $\overline{\text{RESET}}$	V_{IH}	0.7V_{DD}	V_{DD}	V
Input low voltage, I_{LOAD} = 1.6mA	V_{IL}	—	0.2V_{DD}	V
I/O ports three-state leakage ($V_{IN} = V_{IH}$ or V_{IL})	I_{OZ}	—	±10	µA
Input leakage[2] ($V_{IN} = V_{DD}$ or V_{SS}): $\overline{\text{IRQ}}$, $\overline{\text{XIRQ}}$	I_{IN}	—	±1	
(except port E[6:0]) MODB/V_{STBY}		—	±10	µA
Input current with pullup resistors ($V_{IN} = V_{IL}$; ports B, F, J)	I_{IPR}	100	500	µA
RAM stand-by voltage (power down)	V_{SB}	2.0	V_{DD}	V
RAM stand-by current (power down)	I_{SB}	—	20	µA
Input capacitance PE[7:0], $\overline{\text{IRQ}}$, $\overline{\text{XIRQ}}$, EXTAL	C_{IN}	—	8	pF
Ports A, B, C, D, F, J, MODA/$\overline{\text{LIR}}$, RESET		—	12	
Output load capacitance All outputs except PD[4:1],	CL			
XOUT, XTAL, MODA/LIR		—	90	pF
PD[4:1]		—	200	pF

(1) V_{OH} specification for $\overline{\text{RESET}}$ and MODA is not applicable as they are open-drain pins.
V_{OH} specification is not applicable to port C and port D in wired-OR mode.
Refer to Table A.5.6 and Table A.5.7 for port H and port K electrical characteristics.

(2) Refer to A/D specification for the leakage current on A/D inputs.

A

A.4.1 Electrical characteristics for the ports

(V_{DD} = 5.0V ±5%, V_{SS} = 0V, T_A = T_L to T_H unless otherwise noted)

Characteristic	Symbol	Min.	Typ.	Max.	Unit
Output current[1]					
Ports B, C, F: V_{OUT} = V_{OH}	I_{OUT}	—	0.8	5.0	mA
Ports B, C, F: V_{OUT} = V_{OL}	I_{OUT}	—	–1.6	–16	mA
Ports A, D, J: V_{OUT} = V_{OH}	I_{OUT}	—	0.8	—	mA
Ports A, D, J: V_{OUT} = V_{OL}	I_{OUT}	—	–1.6	—	mA
Output low voltage I_{OUT} = –1.6mA, V_{DD} = 5V	V_{OL}	—	—	0.4	V
Output low voltage I_{OUT} = –10mA, V_{DD} = 5V	V_{OL}	—	—	1.2	V
Output low voltage I_{OUT} = –16mA, V_{DD} = 5V	V_{OL}	—	1.9	2.3	V
Output high voltage I_{OUT} = 1.6mA, V_{DD} = 5V	V_{OH}	V_{DD} – 0.8	—	—	V
Output high voltage I_{OUT} = 5mA, V_{DD} = 5V	V_{OH}	V_{DD} – 0.9	—	—	V
Schmitt trigger inputs Threshold voltage on rising edge of input voltage (ports A, D, J)	$V_{th,LH}$	—	3.0	—	V
Schmitt trigger inputs Threshold voltage on falling edge of input voltage (ports A, D, J)	$V_{th,HL}$	—	1.9	—	V

(1) Only 8 I/O pins may be used with its maximum value of either 16mA sink or 5mA source current.

A

A.4.2 DC electrical characteristics — modes of operation

(V_{DD} = 5.0 Vdc ±5%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, unless otherwise noted)

Characteristic	Symbol	4.2MHz	Unit
Maximum total supply current (including PLL)(1):			
RUN:	IDD		
Single chip mode		40	mA
Expanded mode		50	mA
WAIT: (All peripheral functions shut down)	WIDD		
Single chip mode		23	mA
Expanded mode		25	mA
WAIT: (only SRTI running, PLL off, WEN=1, WSLOW=1)		1.5	mA
STOP: (no clocks)	SIDD		
Single chip mode		200	µA
Maximum power dissipation:	P_D		
Single chip mode		210	mW
Expanded mode		262.5	mW

(1) All current measurements taken with suitable decoupling capacitors across the power supply to suppress the transient switching currents inherent in CMOS designs.
EXTAL is driven with a square wave, with t_{CYC} = 238ns for 4.2kHz devices;
All ports (except H, K and the LED driver ports) are configured as inputs; bus frequency f_0 = 4.2MHz; oscillator frequency = 4.2MHz (PLL – 16.8MHz), $V_{IL} \leq 0.2V$; $V_{IH} \geq VDD–0.2V$, no dc loads

A

A.5 Control timing

(V_{DD} = 5.0 Vdc ±5%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H unless otherwise noted)					
Characteristic[1]	**Symbol**	**Min.**	**Typ.**	**Max.**	**Unit**
External oscillator frequency Crystal option (PLL on) External clock option (PLL off)	 f_{XTAL} $4f_0$	 — DC	 4.1943 —	 4.2 16.8	 MHz MHz
Internal operating frequency PLL frequency Crystal (f_{XTAL}/4) External clock	 f_{4XCLK} f_0 f_0	 — — DC	 16.7772 4.1943 	 4.2	 MHz MHz MHz
Cycle time	t_{CYC}	238	—	—	ns
STOP recovery startup time DLY = 0 DLY = 1	 t_{SRS} t_{SRS}	 — —	 — —	 4 4064	 t_{CYC} t_{CYC}
WAIT recovery startup time	t_{WRS}	—	—	4	t_{CYC}
Reset input pulse width[1][2] (To guarantee external reset vector) (Minimum input time; may be pre-empted by internal reset)	PW_{RSTL}	 16 1	 — —	 — —	t_{CYC}
Mode programming set-up time hold time	 t_{MPS} t_{MPH}	 2 10	 — —	 — —	 t_{CYC} ns
Interrupt pulse width, $PW_{IRQ} = t_{CYC}$ +20ns IRQ edge sensitive mode	PW_{IRQ}	258	—	—	ns
Interrupt pulse period	t_{ILIL}	[3]	—	—	t_{CYC}
PLL stabilization time[4]	t_{PLLS}	—	—	1	ms
Processor control setup time	t_{PCSU}	110	—	—	ns
Timer pulse width	PW_{TIM}	258	—	—	ns

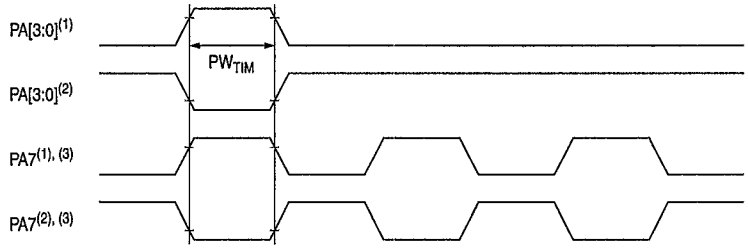
(1) All timing is given with respect to 20% and 70% of VDD, unless otherwise stated.

(2) Reset is recognized during the first clock cycle in which it is held low. Internal circuitry then drives the pin low for eight clock cycles, releases the pin and samples the pin level four cycles later to determine the source of the interrupt. (See Section 5.)

(3) The minimum period t_{ILIL} should not be less than the number of cycles it takes to execute the interrupt service plus 21 t_{CYC}

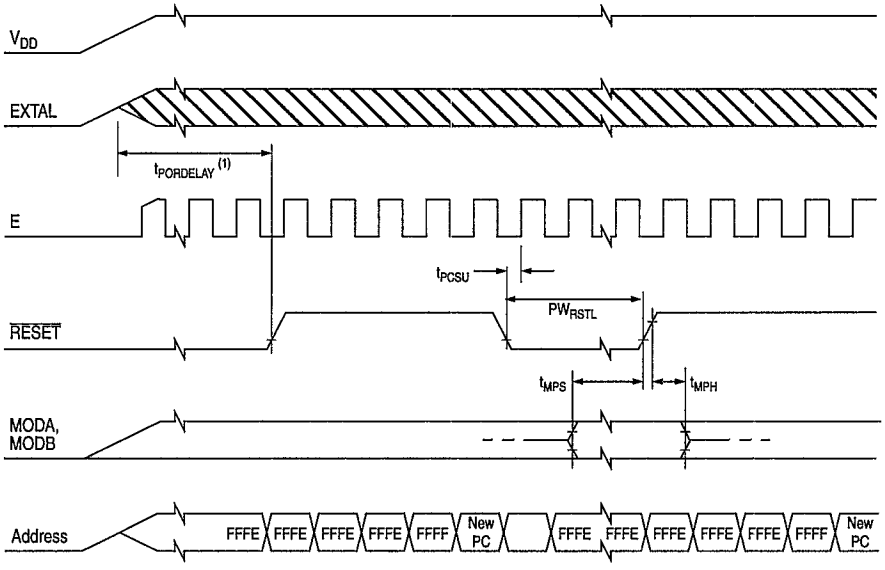
(4) The maximum stabilization time t_{PLLS} is valid for an XFC capacitance of 4.7nF or less

A



Notes
(1) Rising edge sensitive input.
(2) Falling edge sensitive input.
(3) Maximum pulse accumulator clocking rate is E clock frequency divided by two (E/2).

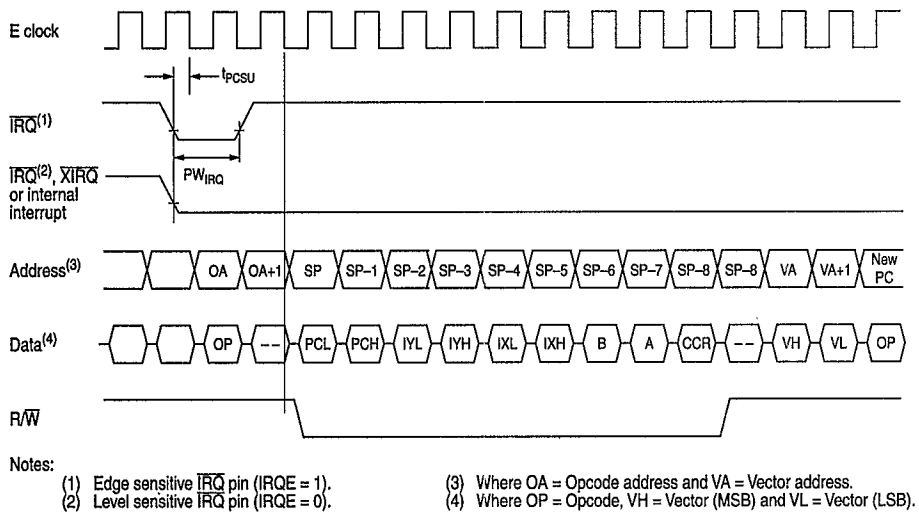
Figure A-2 Timer inputs



(1) $t_{PORDELAY} = 4064\ t_{CYC}$

Figure A-3 Reset timing

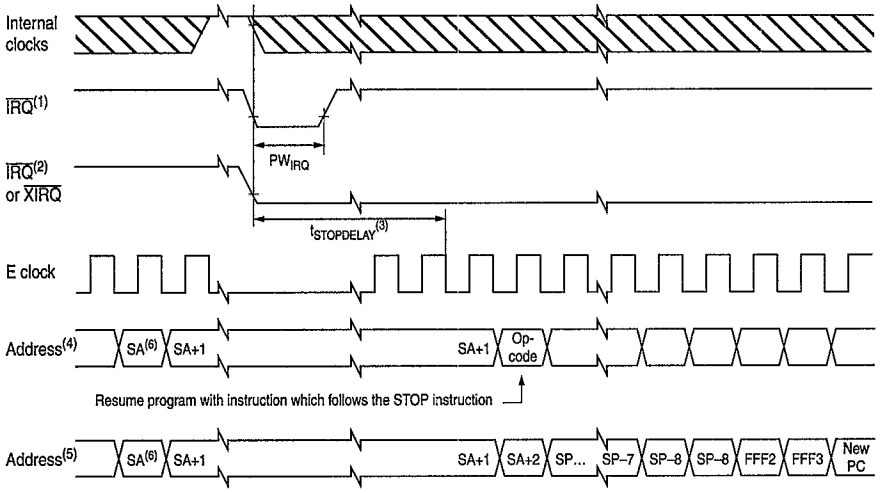
A



Notes:
(1) Edge sensitive $\overline{IRQ}$ pin (IRQE = 1).
(2) Level sensitive $\overline{IRQ}$ pin (IRQE = 0).
(3) Where OA = Opcode address and VA = Vector address.
(4) Where OP = Opcode, VH = Vector (MSB) and VL = Vector (LSB).

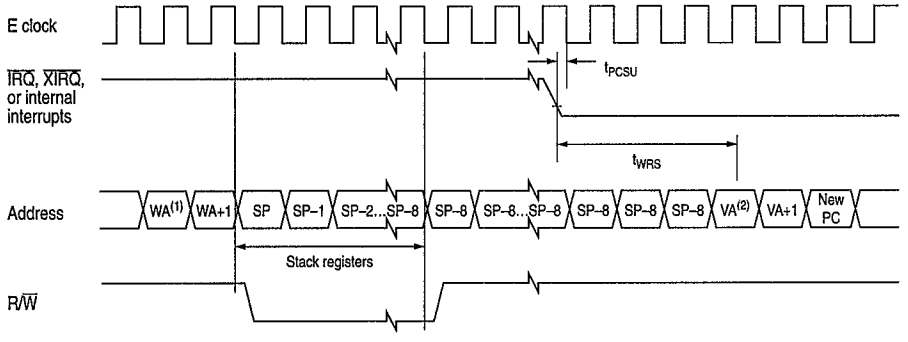
Figure A-4 Interrupt timing

A



Notes:
(1) Edge sensitive $\overline{IRQ}$ pin (IRQE = 1).
(2) Level sensitive $\overline{IRQ}$ pin (IRQE = 0).
(3) If DLY = 1: $t_{STOPDELAY} = 4064\ t_{CYC}$
If DLY = 0: $t_{STOPDELAY} = 4\ t_{CYC}$
(4) $\overline{XIRQ}$ with X-bit in CCR = 1.
(5) $\overline{IRQ}$ (or $\overline{XIRQ}$, with X-bit = 0; in this case vector fetch will be $FFF4/5).
(6) SA = STOP address.

Figure A-5 STOP recovery timing



Notes:
$\overline{RESET}$ also causes recovery from WAIT. (1) WA = WAIT address. (2) VA = Vector address.

Figure A-6 WAIT recovery timing

A.5.1 Peripheral port timing

(V_{DD} = 5.0 Vdc ± 5%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)

Characteristic (1)	Symbol	4.2MHz		Unit
		Min.	Max.	
Frequency of operation (E clock frequency)	f_{OP}	0	4.2	MHz
E clock period	t_{CYC}	238	—	ns
Peripheral data set-up time, all ports (2)	t_{PDSU}	100	—	ns
Peripheral data hold time, all ports (2)	t_{PDH}	50	—	ns
Delay time, peripheral data write	t_{PWD}			ns
MCU write to port A, B , H or K		—	200	
MCU write to port C, D, F or J ($t_{PWD} = t_{CYC}/4 + 100$ns)		—	160	

(1) All timing is given with respect to 20% and 70% of V_{DD}, unless otherwise noted.

(2) Port C and D timing is valid for active drive (CWOM, DWOM, and WOMS bits clear).

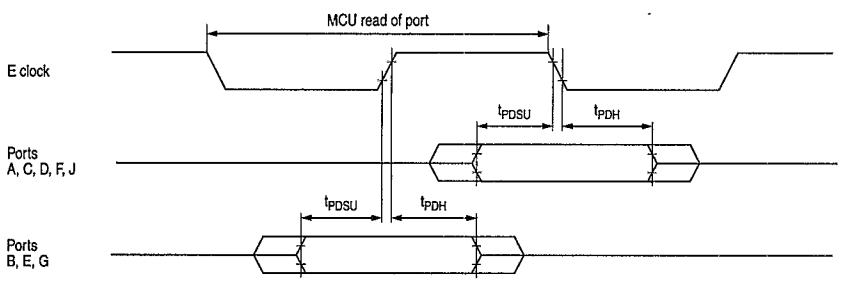


Figure A-7 Port read timing diagram

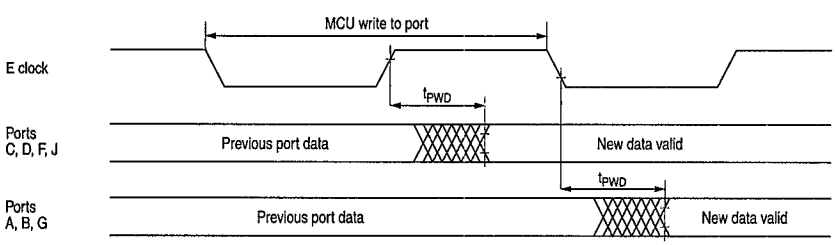


Figure A-8 Port write timing diagram

A

A.5.2 PLL control timing

(VDD = 5.0Vdc ± 5%, VSS = 0Vdc, $T_A = T_L$ to T_H unless otherwise noted)

Characteristic	Symbol	Min.	Typ	Max	Units
PLL reference frequency	f_{REF}	2000	4194.3	4200	kHz
System frequency	f_{SYS}	dc	—	4.2	MHz
PLL output frequency	f_{VCOOUT}	2	—	16.8	
External clock operation	f_{XTAL}	dc	—	16	
Capacitor on pin XFC	C_{XFC}	—	—	4.7	nF
PLL stabilization time(1)	t_{PLLS}	—		1	ms

(1) Assumes that stable VDDSYN is applied and that the crystal oscillator is stable. Stabilization time is measured from power-up to $\overline{\text{RESET}}$ release. This specification also applies to the period required for PLL stabilization after changing the X and Y frequency control bits in the synthesizer control register (SYNR) while PLL is running, and to the period required for the clock to stabilize after WAIT with WEN = 1.

A

A.5.3 Pulse width modulation characteristics

Characteristic	Symbol	Min.	Max.	Unit
Repetition frequency	f_P	16	—	kHz
Period time	t_{PERIOD}	—	1/16	ms
Duty time	t_{DUTY}	0/512 * t_{PERIOD}	511/512 * t_{PERIOD}	s

A.5.4 Analog-to-digital converter characteristics

(V_{DD} = 5.0 Vdc ± 5%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H. 750kHz ≤ E ≤ 4MHz, unless otherwise noted)(1)

Characteristic	Parameter	Min.	Typ.	Max.	Unit
Resolution	Number of bits resolved by ADC	10	10	—	bits
Non-linearity(2)	Maximum deviation from the best fitting ADC transfer characteristic	—	±0.25	±1.0	LSB
Offset error(2)	Maximum offset from ADC transfer characteristic	—	±0.25	±1.0	LSB
Total unadjusted error	Maximum deviation from the ideal ADC transfer characteristic	—	±0.5	±1.5	LSB
Quantization error	Uncertainty due to converter resolution	—	—	±0.5	LSB
Absolute accuracy	Difference between the actual input voltage and the full-scale weighted equivalent of the binary output code, including all error sources	—	±1.0	±2.0	LSB
Conversion range	Analog input voltage range	V_{RL}	—	V_{RH}	V
Analog input voltage(3)	Maximum and minimum analog input voltage	V_{RL}–0.3	—	1.125V_{RH}	V
V_{RH}	Maximum analog reference voltage	V_{RL}	—	V_{DD}+0.1	V
V_{RL}	Minimum analog reference voltage	V_{SS}–0.1	—	V_{RH}	V
ΔV_R	Minimum difference between V_{RH} and V_{RL}	4.5	—	—	V
Monotonicity	Conversion result never decreases with an increase in input voltage and has no missing codes		Guaranteed		
Zero input reading	Conversion result when $V_{in} = V_{RL}$	0000	—	—	Hex
Full scale reading	Conversion result when $V_{in} = V_{RH}$	—	—	FFC0	Hex
Input leakage(4)	Input leakage on A/D pins: PE[6:0]	—	—	400	nA
Current consumption	Vrh to Vrl divider chain current consumption	—	—	300	µA

(1) Input clock duty cycles other than 50% will affect the A/D accuracies.

(2) Non-linearity and offset error are characterised but not tested.

(3) Minimum analogue input voltage should not go below V_{RL}–0.3V.
Maximum analogue input voltage should not exceed 1.125V_{RH}

(4) Source impedance should equal approximately 1kΩ. Source impedances greater than 1kΩ affect accuracy adversely because of input leakage.

A

A.5.5 Digital-to-analog converter characteristics

(V_{DD} = 5.0 Vdc ±5%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H. 750kHz ≤ E ≤ 4MHz, unless otherwise noted)

Characteristic	Parameter	Min.	Typ.	Max.	Unit
Resolution	Number of bits resolved by the D/A	8		—	Bits
Absolute accuracy, 0.1μA source current on port E7	Difference between the output code in the D/A data register and the equivalent output voltage	—	—	±1	LSB

A

A.5.6 Port H and port K DC electrical characteristics

Characteristic	Symbol	Min.	Max.	Unit
Output voltage, T=–40°C, PV_{DD}=8V I_{OUT}=47mA I_{OUT}=–47mA	V_{oH} V_{oL}	PV_{DD} – 0.5 —	— 0.35	V V
Output voltage, T=25°C, PV_{DD}=8V I_{OUT}=39mA I_{OUT}=–39mA	V_{oH} V_{oL}	PV_{DD} – 0.5 —	— 0.35	V V
Output voltage, T=85°C, PV_{DD}=8V I_{OUT}=35mA I_{OUT}=–35mA	V_{oH} V_{oL}	PV_{DD} – 0.5 —	— 0.35	V V
Output voltage, T=–40°C, PV_{DD}=8.5V I_{OUT}=48mA I_{OUT}=–48mA	V_{oH} V_{oL}	PV_{DD} – 0.5 —	— 0.35	V V
Output voltage, T=25°C, PV_{DD}=8.5V I_{OUT}=41mA I_{OUT}=–41mA	V_{oH} V_{oL}	PV_{DD} – 0.5 —	— 0.35	V V
Output voltage, T=85°C, PV_{DD}=8.5V I_{OUT}=36mA I_{OUT}=–36mA	V_{oH} V_{oL}	PV_{DD} – 0.5 —	— 0.35	V V
Output voltage, T=–40°C, PV_{DD}=9V I_{OUT}=50mA I_{OUT}=–50mA	V_{oH} V_{oL}	PV_{DD} – 0.5 —	— 0.35	V V
Output voltage, T=25°C, PV_{DD}=9V I_{OUT}=42mA I_{OUT}=–42mA	V_{oH} V_{oL}	PV_{DD} – 0.5 —	— 0.35	V V
Output voltage, T=85°C, PV_{DD}=9V I_{OUT}=37mA I_{OUT}=–37mA	V_{OH} V_{oL}	PV_{DD} – 0.5 —	— 0.35	V V
Voltage drop mismatch between different drivers I_{OUT} = 30mA	V_{ddr}	—	50	mV

Note: The worst case on-resistances R_{ONP} and R_{ONN} are obtained from the port H and port K characteristics. The given PMOS and NMOS voltage drops have to be divided by the specified output currents. The appropriate voltage drops to be used for the calculation are $V_{dP} = PV_{DD} - V_{oH}$ and $V_{dN} = V_{oL}$ for the PMOS and NMOS transistor respectively. The on-resistances are calculated with: $R_{ONP} = V_{dP}/I_{OUT}$ for the PMOS transistor and $R_{ONN} = V_{dN}/I_{OUT}$ for the NMOS transistor.

A

A.5.7 Port H and port K maximum ratings

Characteristic	Symbol	PV_{DD}=8V		PV_{DD}=8.5V		PV_{DD}=9V		Unit
		Min.	Max.	Min.	Max.	Min.	Max.	
Maximum allowed output current (DC)(1)								
T=–40°C	I_{omax}	—	±47	—	±48	—	±50	mA
T=25°C	I_{omax}	—	±39	—	±41	—	±42	mA
T=85°C	I_{omax}	—	±35	—	±36	—	±37	mA
Maximum allowed reverse output current (DC)(2)								
T=–40°C	I_{rmax}	—	–47	—	–48	—	–50	mA
T=25°C	I_{rmax}	—	–39	—	–41	—	–42	mA
T=85°C	I_{rmax}	—	–29	—	–30	—	–31	mA
Output high voltage (DC)(3)								
T=–40°C	V_{outh}	$PV_{DD}-0.5$	$PV_{DD}+0.5$	$PV_{DD}-0.5$	$PV_{DD}+0.5$	$PV_{DD}-0.5$	$PV_{DD}+0.5$	V
T=25°C	V_{outh}	$PV_{DD}-0.5$	$PV_{DD}+0.5$	$PV_{DD}-0.5$	$PV_{DD}+0.5$	$PV_{DD}-0.5$	$PV_{DD}+0.5$	V
T=85°C	V_{outh}	$PV_{DD}-0.5$	$PV_{DD}+0.4$	$PV_{DD}-0.5$	$PV_{DD}+0.4$	$PV_{DD}-0.5$	$PV_{DD}+0.4$	V
Output low voltage (DC)(3)	V_{outl}	0	0.5	0	0.5	0	0.5	V
Short circuit withstand time	t_{SC}	—	20	—	20	—	20	ms

(1) These are the maximum output currents when using resistive loads. The output voltage is lower than PV_{DD} and higher than PV_{SS} in this case. In any case, the output voltage must not be lower than $V_{outh} = PV_{DD} - 0.5V$, and it must not be higher than $V_{outl} = 0.5V$. The maximum positive output currents are given for the worst case minimum allowed output voltage $V_{outh} = PV_{DD} - 0.5V$.

(2) These are the maximum output currents when using inductive loads. In the flyback state, the output voltage could be higher than PV_{DD}. The output voltage must not exceed the maximum values V_{outh} at the given temperatures. The maximum reverse output currents are given for the worst case maximum allowed output voltage $V_{outh} = PV_{DD} + 0.5V$ (T = –40°C, T = 25°C) and $V_{outh} = PV_{DD} + 0.4V$ (T = 85°C).

(3) The output voltage V_{oH} must not exceed its maximum specified value in the flyback state when driving inductive loads. It must not be lower than the minimum specified value when driving inductive loads as well as only resistive loads. The output voltage V_{oL} must not be lower than $V_{outl} = 0V$ and it must not be higher than $V_{outl} = 0.5V$ for resistive as well as for inductive loads.

A

A.5.8 Power on reset characteristics

(V_{DD} = 5V ± 5%, f_0 = 4.2 MHz)

Characteristic	Min.	Typ.	Max.	Unit
Supply voltage, V_{DD}	4.5	5	5.5	V
Power on reset voltage, V_0 (1)	> V_1	4.3	4.75	V
Power off reset voltage, V_1 (1)	3.8	4.2	< V_0	V
Minimum reset voltage, V_{MIN}	1.5	—	—	V
Supply voltage rise and fall time	—	—	1000	ms
LVR internal delay	—	5	—	ms

(1) The functionality of the parts will be ensured at supply voltages lower than 5V down to those levels where the LVR reset becomes active.

A

A.5.9 Serial peripheral interface timing

(V_{DD} = 5.0 Vdc ±5%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)

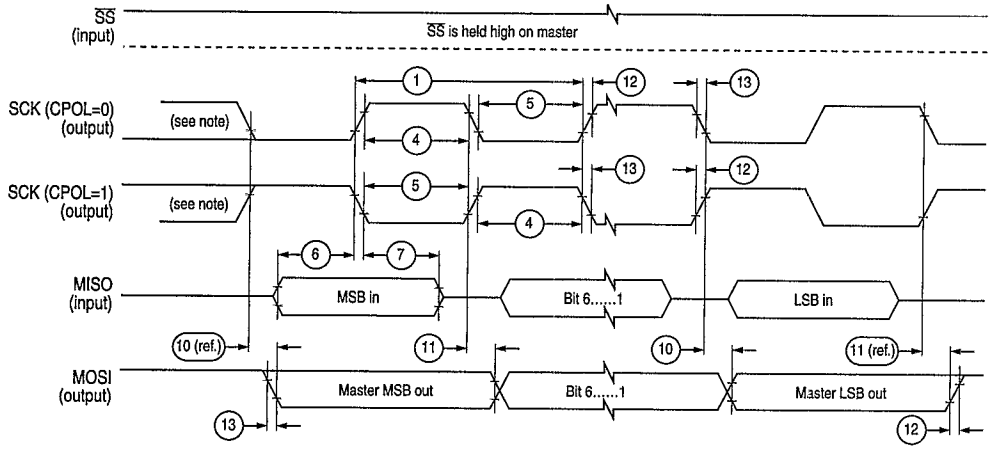
Num	Characteristic [1]		Symbol	4.2MHz Min.	4.2MHz Max.	Unit
	Operating frequency	Master Slave	$f_{OP(M)}$ $f_{OP(S)}$	0 0	0.5 4.0	f_{OP} MHz
1	Cycle time	Master Slave	$t_{CYC(M)}$ $t_{CYC(S)}$	2.0 250	— —	t_{CYC} ns
2	Enable lead time [2]	Master Slave	$t_{LEAD(M)}$ $t_{LEAD(S)}$	— 200	— —	ns
3	Enable lag time [2]	Master Slave	$t_{LAG(M)}$ $t_{LAG(S)}$	— 200	— —	ns
4	Clock (SCK) high time	Master Slave	$t_{W(SCKH)M}$ $t_{W(SCKH)S}$	130 85	— —	ns
5	Clock (SCK) low time	Master Slave	$t_{W(SCKL)M}$ $t_{W(SCKL)S}$	130 85	— —	ns
6	Input data set-up time	Master Slave	$t_{SU(M)}$ $t_{SU(S)}$	100 100	— —	ns
7	Input data hold time	Master Slave	$t_{H(M)}$ $t_{H(S)}$	100 100	— —	ns
8	Access time (from high-z to data active)	Slave	t_A	0	120	ns
9	Disable time (hold time to high-z state)	Slave	t_{DIS}	—	125	ns
10	Data valid (after enable edge) [3]		$t_{V(S)}$	—	125	ns
11	Output data hold time (after enable edge)		t_{HO}	0	—	ns
12	Rise time [3] SPI outputs (SCK, MOSI and MISO) SPI inputs (SCK, MOSI, MISO and $\overline{SS}$)		 t_{RM} t_{RS}	 — —	 100 2.0	 ns µs
13	Fall time [3] SPI outputs (SCK, MOSI and MISO) SPI inputs (SCK, MOSI, MISO and $\overline{SS}$)		 t_{FM} t_{FS}	 — —	 100 2.0	 ns µs

(1) All timing is given with respect to 20% and 70% of V_{DD}, unless otherwise noted.

(2) Signal production depends on software.

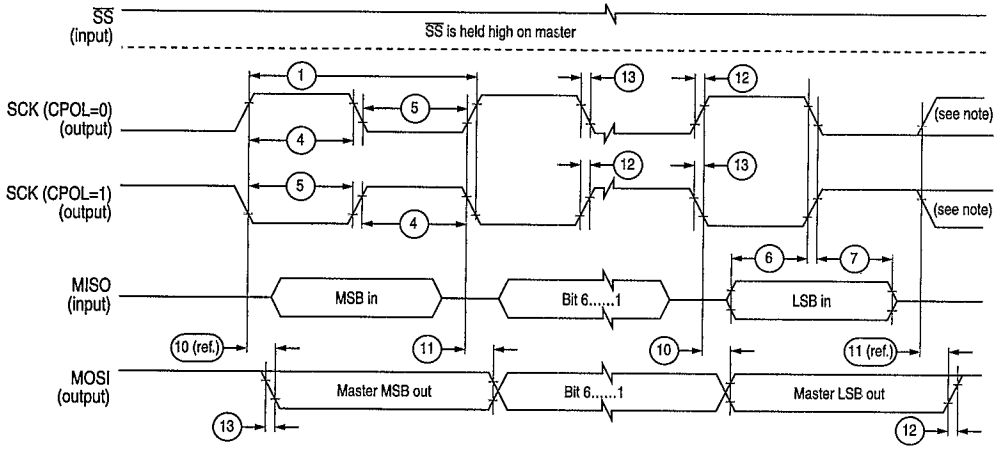
(3) Assumes 200pF load on all SPI pins.

A



Note: This first clock edge is generated internally, but is not seen at the SCK pin.

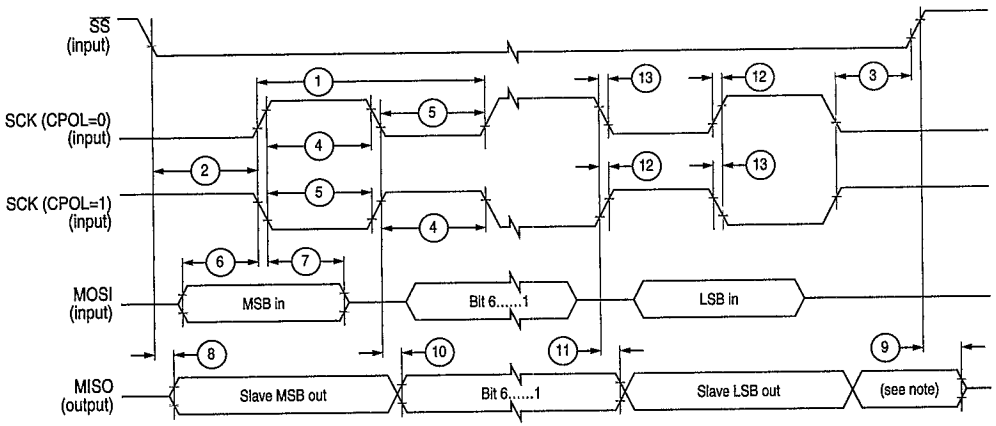
Figure A-9 SPI master timing (CPHA = 0)



Note: This last clock edge is generated internally, but is not seen at the SCK pin.

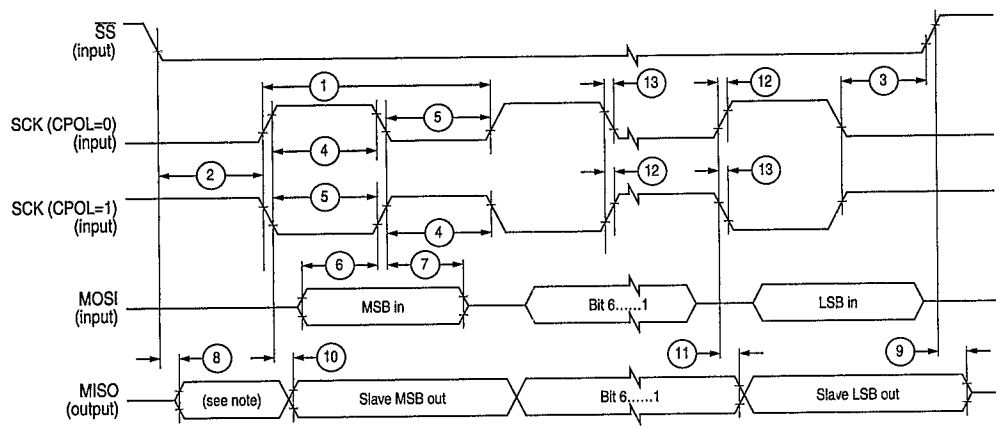
Figure A-10 SPI master timing (CPHA = 1)

A



Note: Not defined, but normally the MSB of character just received.

Figure A-11 SPI slave timing (CPHA = 0)



Note: Not defined, but normally the LSB of character last transmitted.

Figure A-12 SPI slave timing (CPHA = 1)

A

A.5.10 Non-multiplexed expansion bus timing

(V_{DD} = 5.0 Vdc ±5%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)

Num	Characteristic [1]	Symbol	4.2MHz		Unit
			Min.	Max.	
	Frequency of operation (E clock frequency)	f_{OP}	0	4.2	MHz
1	E clock period	t_{CYC}	238	—	ns
2	Pulse width, E low [2], [3]	PW_{EL}	100	—	ns
3	Pulse width, E high [2], [3]	PW_{EH}	95	—	ns
4A	E clock rise time	t_R	—	20	ns
4B	E clock fall time	t_F		15	
9	Address hold time [3]	t_{AH}	20	—	ns
11	Address delay time [3]	t_{AD}	—	70	ns
12	Address valid to E rise time [3]	t_{AV}	30	—	ns
17	Read data set-up time	t_{DSR}	20	—	ns
18	Read data hold time	t_{DHR}	0	—	ns
19	Write data delay time	t_{DDW}	—	40	ns
21	Write data hold time [3]	t_{DHW}	30	—	ns
29	MPU address access time [3]	t_{ACCA}	135	—	ns
39	Write data set-up time [3]	t_{DSW}	55	—	ns
57	Address valid to data three-state time	t_{AVDZ}	—	10	ns

(1) All timing is given with respect to 20% and 70% of V_{DD}, unless otherwise noted.

(2) Input clock duty cycles other than 50% will affect the bus performance.

(3) For $f_{OP} \leq 2$MHz the following formulae may be used to calculate parameter values:

$PW_{EL} = t_{CYC}/2 - 20ns$ $\quad PW_{EH} = t_{CYC}/2 - 25ns$

$t_{AH} = t_{CYC}/8 - 10ns$ $\quad t_{AD} = t_{CYC}/8 + 40ns$

$t_{AV} = PW_{EL} - t_{AD}$ $\quad t_{DHW} = t_{CYC}/8$

$t_{ACCA} = t_{CYC} - t_F - t_{DSR} - t_{AD}$ $\quad t_{DSW} = PW_{EH} - t_{DDW}$

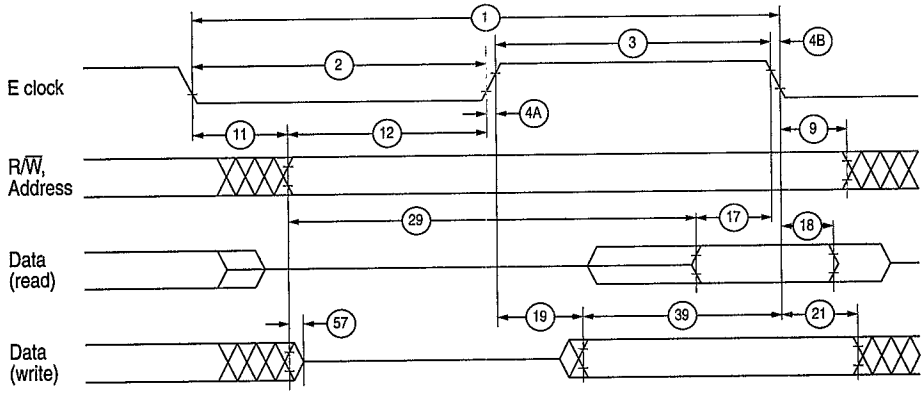


Figure A-13 Expansion bus timing

A.6 EEPROM characteristics

Characteristic	Temperature range −40 to +85°C	Unit
Programming time, t_{EEPROG} [1]		
<1MHz, RCO enabled	10	ms
1–2MHz, RCO disabled	20	ms
≥2MHz & whenever RCO enabled	10	ms
Erase time: byte, row and bulk [1]	10	ms
Write/erase endurance [2]	10000	cycles
Data retention [2]	10	years

(1) The RC oscillator (RCO) must be enabled (by setting the CSEL bit in the OPTION register) for EEPROM programming and erasure when the E clock frequency is less than 1.0MHz.

(2) Refer to the current issue of Motorola's quarterly ***Reliability Monitor Report*** for the latest failure rate information.

A

THIS PAGE LEFT BLANK INTENTIONALLY

B

MECHANICAL DATA AND ORDERING INFORMATION

The MC68HC11KG4 is available packaged in a 100-pin thin quad flat pack (TQFP).

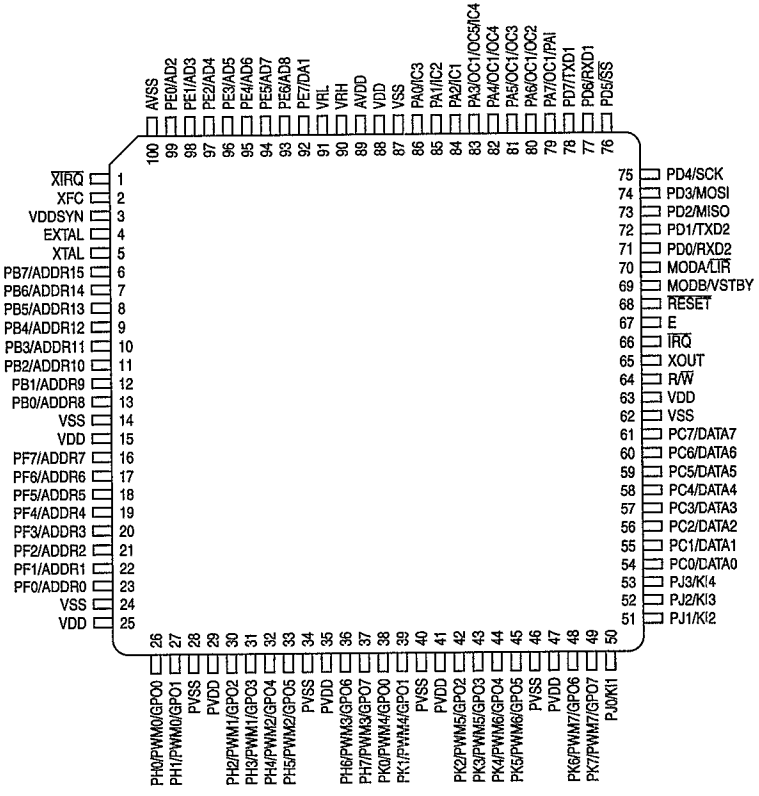
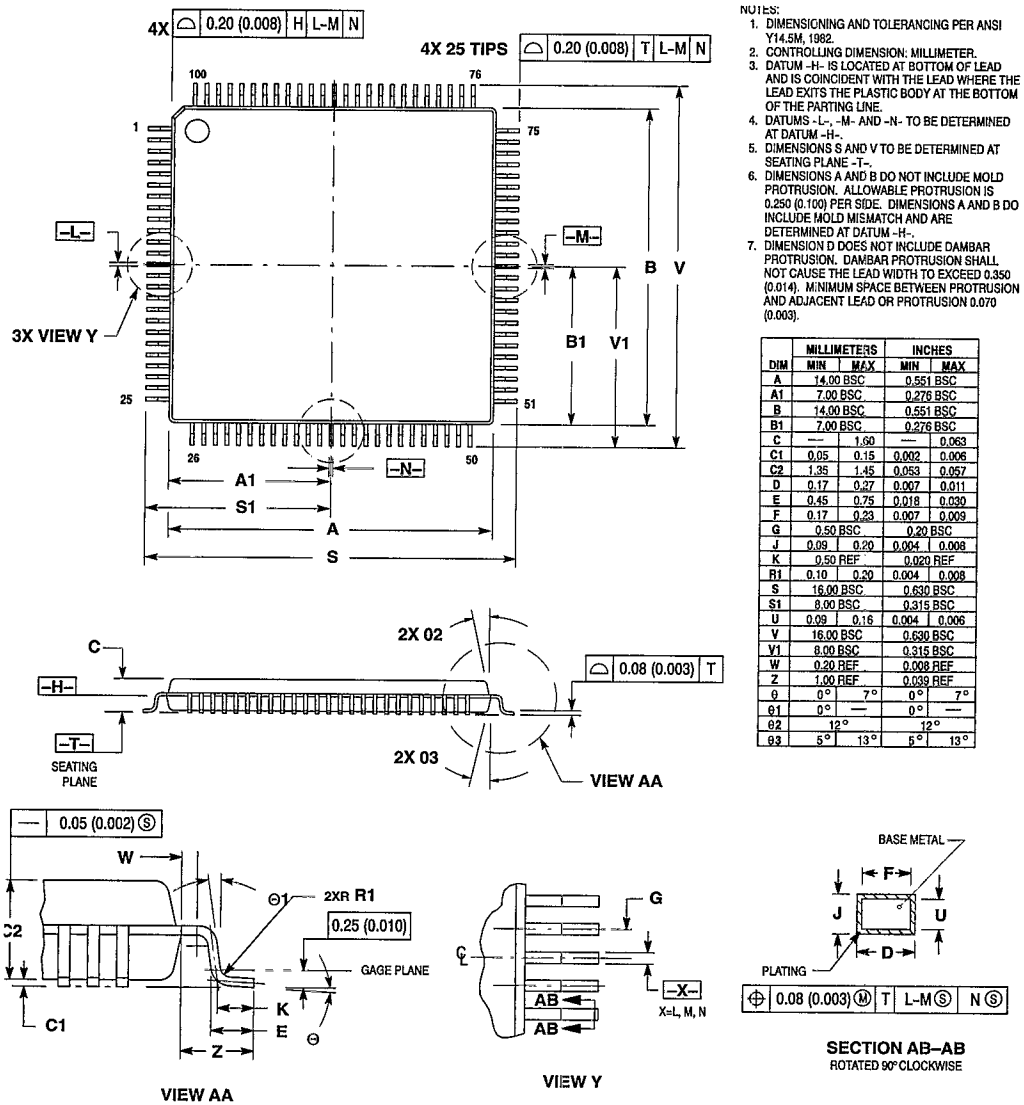


Figure B-1 100-pin TQFP



NOTES:
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: MILLIMETER.
3. DATUM -H- IS LOCATED AT BOTTOM OF LEAD AND IS COINCIDENT WITH THE LEAD WHERE THE LEAD EXITS THE PLASTIC BODY AT THE BOTTOM OF THE PARTING LINE.
4. DATUMS -L-, -M- AND -N- TO BE DETERMINED AT DATUM -H-.
5. DIMENSIONS S AND V TO BE DETERMINED AT SEATING PLANE -T-.
6. DIMENSIONS A AND B DO NOT INCLUDE MOLD PROTRUSION. ALLOWABLE PROTRUSION IS 0.250 (0.100) PER SIDE. DIMENSIONS A AND B DO INCLUDE MOLD MISMATCH AND ARE DETERMINED AT DATUM -H-.
7. DIMENSION D DOES NOT INCLUDE DAMBAR PROTRUSION. DAMBAR PROTRUSION SHALL NOT CAUSE THE LEAD WIDTH TO EXCEED 0.350 (0.014). MINIMUM SPACE BETWEEN PROTRUSION AND ADJACENT LEAD OR PROTRUSION 0.070 (0.003).

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	14.00 BSC		0.551 BSC	
A1	7.00 BSC		0.276 BSC	
B	14.00 BSC		0.551 BSC	
B1	7.00 BSC		0.276 BSC	
C	—	1.60	—	0.063
C1	0.05	0.15	0.002	0.006
C2	1.35	1.45	0.053	0.057
D	0.17	0.27	0.007	0.011
E	0.45	0.75	0.018	0.030
F	0.17	0.23	0.007	0.009
G	0.50 BSC		0.20 BSC	
J	0.09	0.20	0.004	0.008
K	0.50 REF		0.020 REF	
R1	0.10	0.20	0.004	0.008
S	16.00 BSC		0.630 BSC	
S1	8.00 BSC		0.315 BSC	
U	0.09	0.16	0.004	0.006
V	16.00 BSC		0.630 BSC	
V1	8.00 BSC		0.315 BSC	
W	0.20 REF		0.008 REF	
Z	1.00 REF		0.039 REF	
θ	0°	7°	0°	7°
θ1	0°	—	0°	—
θ2	12°		12°	
θ3	5°	13°	5°	13°

Case 983-01

Figure B-2 100-pin TQFP mechanical dimensions

B

B.1 Ordering information

Use the information in the following table to specify the appropriate device when placing an order.

Table B-1 Custom ROM device ordering information

Package	Temperature	Description	Frequency	Source device
100-pin TQFP	–40 to +85°C	Custom ROM	4 MHz	MC68HC11KG4CPU4

To specify a custom ROM device, first select a standard source device, then complete a custom ROM device order form. The order form can be obtained from your local Motorola sales office or distributer.

B

THIS PAGE LEFT BLANK INTENTIONALLY

C
DEVELOPMENT SUPPORT

The following information provides a reference to development tools for the M68HC11 family of microcontrollers. For more detailed information please refer to the appropriate system manual.

Table C-1 M68HC11 development tools

Devices	Evaluation boards	Evaluation modules	Evaluation systems/kits	Programmer boards
MC68HC11KG4	—	M68EM11KG4	—	M68SPGMR11

Note: Target cables for the evaluation module should be ordered separately.

C.1 EVS — Evaluation system

The EVS is an economical tool for designing, debugging and evaluating target systems based on the MC68HC11KG4 device type. The two printed circuit boards that comprise the EVS are the M68EM11KG4 emulator module and the M68PFB11KIT platform board. The main features of the EVS are as follows:

- Monitor/debugger firmware
- Single-line assembler/disassembler
- Host computer download capability
- Dual memory maps:
  - 64Kbyte monitor map that includes 16Kbytes of monitor EPROM
- MCU extension I/O port for single chip, expanded and special test operating modes
- RS-232C terminal and host I/O ports
- Logic analyser connector

C.2 MMDS11 — Motorola modular development system

The MMDS11 is an emulator system that provides a bus state analyser and real-time memory windows. The unit's integrated design environment includes an editor, an assembler, user interface and source-level debug. A complete MMDS11 consists of:

- A station module — the metal MMDS11 enclosure, containing the control board and the internal power supply. Most system cables connect to the MMDS11 station module. (The cable to an optional target system, however, runs through an aperture in the station module enclosure to connect directly to the emulator module).
- An emulator module (EM) — such as the EM11KG4: a printed circuit board that enables system functionality for a specific set of MCUs. The EM fits into the station module through a sliding panel in the enclosure top. The EM has a connector for the target cable.
- Two logic clip cable assemblies — twisted pair cables that connect the station module to your target system, a test fixture, a clock or any other circuitry useful for evaluation or analysis. One end of each cable assembly has a moulded connector, which fits into station module pod A or pod B. Leads at the other end of the cable terminate in female probe tips. Ball clips come with the cables.
- A 9-lead RS-232 serial cable — the cable that connects the station module to the host computer's RS-232 port.

C.3 SPGMR11 — Serial programmer system

The SPGMR11 is an economical tool for programming M68HC11 MCUs. The system consists of the M68SPGMR11 unit and a programming module which adapts the SPGMR11 to the appropriate MCU and package type. The programming module can be ordered as M68PA11KG4PU100 for 100-pin TQFP packaged devices.

GLOSSARY

This section contains abbreviations and specialist words used in this data sheet and throughout the industry. Further information on many of the terms may be gleaned from Motorola's ***M68HC11 Reference Manual, M68HC11RM/AD,*** or from a variety of standard electronics text books.

$xxxx	The digits following the '$' are in hexadecimal format.
%xxxx	The digits following the '%' are in binary format.
A/D, ADC	Analog-to-digital (converter).
Bootstrap mode	In this mode the device automatically loads its internal memory from an external source on reset and then allows this program to be executed.
Byte	Eight bits.
CCR	Condition codes register; an integral part of the CPU.
CERQUAD	A ceramic package type, principally used for EPROM and high temperature devices.
Clear	'0' — the logic zero state; the opposite of 'set'.
CMOS	Complementary metal oxide semiconductor. A semiconductor technology chosen for its low power consumption and good noise immunity.
COP	Computer operating properly. *aka* 'watchdog'. This circuit is used to detect device runaway and provide a means for restoring correct operation.
CPU	Central processing unit.
D/A, DAC	Digital-to-analog (converter).
EEPROM	Electrically erasable programmable read only memory. *aka* 'EEROM'.
EPROM	Erasable programmable read only memory. This type of memory requires exposure to ultra-violet wavelengths in order to erase previous data. *aka* 'PROM'.
ESD	Electrostatic discharge.
Expanded mode	In this mode the internal address and data bus lines are connected to external pins. This enables the device to be used in much more complex systems, where there is a need for external memory for example.

EVS	Evaluation system. One of the range of platforms provided by Motorola for evaluation and emulation of their devices.
HCMOS	High-density complementary metal oxide semiconductor. A semiconductor technology chosen for its low power consumption and good noise immunity.
I/O	Input/output; used to describe a bidirectional pin or function.
Input capture	(IC) This is a function provided by the timing system, whereby an external event is 'captured' by storing the value of a counter at the instant the event is detected.
Interrupt	This refers to an asynchronous external event and the handling of it by the MCU. The external event is detected by the MCU and causes a predetermined action to occur.
$\overline{\text{IRQ}}$	Interrupt request. The overline indicates that this is an active-low signal format.
K byte	A kilo-byte (of memory); 1024 bytes.
LCD	Liquid crystal display.
LSB	Least significant byte.
M68HC11	Motorola's family of advanced 8-bit MCUs.
MCU	Microcontroller unit.
MI BUS	Motorola interconnect bus. A single wire, medium speed serial communications protocol.
MSB	Most significant byte.
Nibble	Half a byte; four bits.
NRZ	Non-return to zero.
Opcode	The opcode is a byte which identifies the particular instruction and operating mode to the CPU. See also: prebyte, operand.
Operand	The operand is a byte containing information the CPU needs to execute a particular instruction. There may be from 0 to 3 operands associated with an opcode. See also: opcode, prebyte.
Output compare	(OC) This is a function provided by the timing system, whereby an external event is generated when an internal counter value matches a predefined value.
PLCC	Plastic leaded chip carrier package.
PLL	Phase-locked loop circuit. This provides a method of frequency multiplication, to enable the use of a low frequency crystal in a high frequency circuit.
Prebyte	This byte is sometimes required to qualify an opcode, in order to fully specify a particular instruction. See also: opcode, operand.

Pull-down, pull-up	These terms refer to resistors, sometimes internal to the device, which are permanently connected to either ground or V_{DD}.
PWM	Pulse width modulation. This term is used to describe a technique where the width of the high and low periods of a waveform is varied, usually to enable a representation of an analog value.
QFP	Quad flat pack package.
RAM	Random access memory. Fast read and write, but contents are lost when the power is removed.
RFI	Radio frequency interference.
RTI	Real-time interrupt.
ROM	Read-only memory. This type of memory is programmed during device manufacture and cannot subsequently be altered.
RS-232C	A standard serial communications protocol.
SAR	Successive approximation register.
SCI	Serial communications interface.
Set	'1' — the logic one state; the opposite of 'clear'.
Silicon glen	An area in the central belt of Scotland, so called because of the concentration of semiconductor manufacturers and users found there.
Single chip mode	In this mode the device functions as a self contained unit, requiring only I/O devices to complete a system.
SPI	Serial peripheral interface.
Test mode	This mode is intended for factory testing.
TTL	Transistor-transistor logic.
UART	Universal asynchronous receiver transmitter.
VCO	Voltage controlled oscillator.
Watchdog	*see* 'COP'.
Wired-OR	A means of connecting outputs together such that the resulting composite output state is the logical OR of the state of the individual outputs.
Word	Two bytes; 16 bits.
$\overline{\text{XIRQ}}$	Non-maskable interrupt request. The overline indicates that this has an active-low signal format.

THIS PAGE LEFT BLANK INTENTIONALLY

INDEX

In this index numeric entries are placed first; page references in *italics* indicate that the reference is to a figure.

D

E

F

G

H

I

J

K

L

M

N

O

P

R

S

T

U

V

W

X

Z

THIS PAGE LEFT BLANK INTENTIONALLY

How to reach us:

USA/EUROPE: Motorola Literature Distribution; P.O. Box 5405, Denver, Colorado 80217. 1-303-675-2140

Mfax: RMFAX0@email.sps.mot.com – TOUCHTONE 1- 602-244-6609, http://sps.motorola.com/mfax

US & CANADA ONLY: http://sps.motorola.com/mfax

HOME PAGE: http://motorola.com/sps/

JAPAN: Nippon Motorola Ltd.; Tatsumi-SPD-JLDC, 6F Seibu-Butsuryu-Center, 3-14-2 Tatsumi Koto-Ku, Tokyo 135, Japan. 81-3-3521-8315

HONG KONG: Motorola Semiconductors H.K. Ltd.; 8B Tai Ping Industrial Park, 51 Ting Kok Road, Tai Po, N.T., Hong Kong. 852-26629298

CUSTOMER FOCUS CENTER: 1-800-521-6274

MOTOROLA

MC68HC11KG4/D

1ATX45289-0 PRINTED IN USA 6/98 IMPERIAL LITHO 36342 1000 LITCSG2