68HC05E16
68HC705E24

SPECIFICATION

Rev. 0.8

(General Release)

Microcontroller Design Center
Munich, Germany

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

1.1 GENERAL

The 68HC705E24/68HC05E16 HCMOS Microcomputer are members of the M68HC05 Family, specially suitable in telecom applications. This 8 bit microcomputer unit (MCU) contains an oscillator system specially designed for telecom applications, CPU, RAM, (EP)ROM, Bootloader-ROM, EEPROM, parallel I/O, a Core timer, a 16-bit programmable timer, COP, a multiplexed dual I^2C Bus and a 2 channel A/D Converter.

1.2 FEATURES

- 64 QFP Package (a 44 pin QFP Package is available for the 68HC05E16 for cost and space sensitive applications)
- 16240 Bytes of User ROM +128 Bytes Bootloader ROM + 16 Bytes User Vectors
- *24064 Bytes of User EPROM +1024 Bytes Bootloader ROM + 16 Bytes User Vectors (EPROM)*
- 352 Bytes of RAM
- 320 Bytes EEPROM
- 16 Bit Timer with two input captures and two output compares
- 15 stage multi-functional core timer with overflow, real time interrupt and watchdog
- COP Watchdog Timer Mask Option / *Write once register*
- 49 dedicated I/O Lines (29 on 44 Pin QFP)
- Power Saving STOP and WAIT Modes
- Edge-Sensitive or Edge and Level-Sensitive Interrupt Trigger
- Keyboard Interrupt
- Multiplexed Dual I^2C Bus
- 2 Channel A/D Converter
- Low Voltage Indicator
- Phase-Locked-Loop (PLL) Synthesizer with programmable speed and buffered crystal output
- Ceramic Resonator for fast start-up
- Internal RC-Oscillator to replace ceramic resonator in cost sensitive applications and fast start-up times

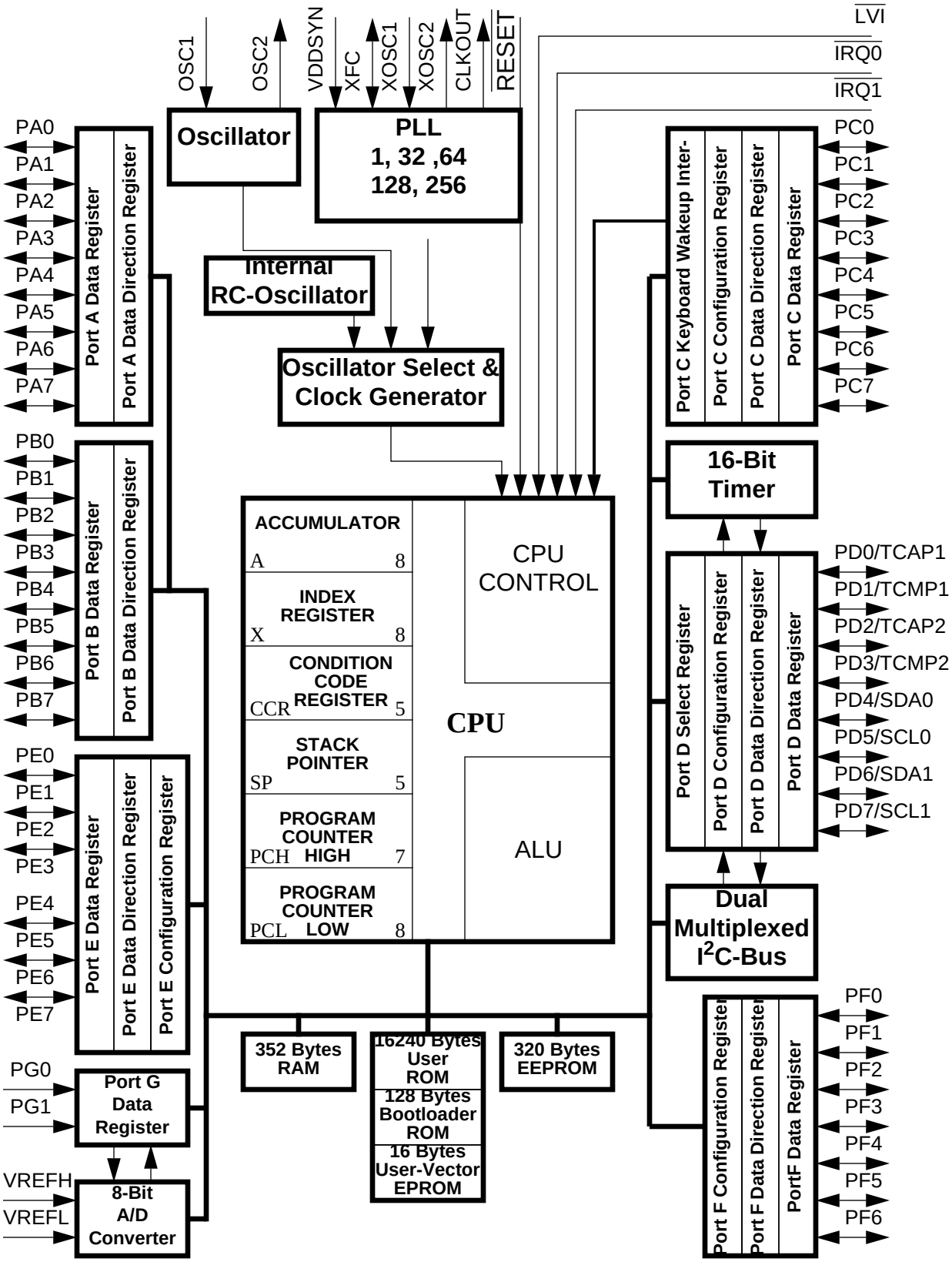


Figure 1-1 Block Diagram of the 68HC05E16

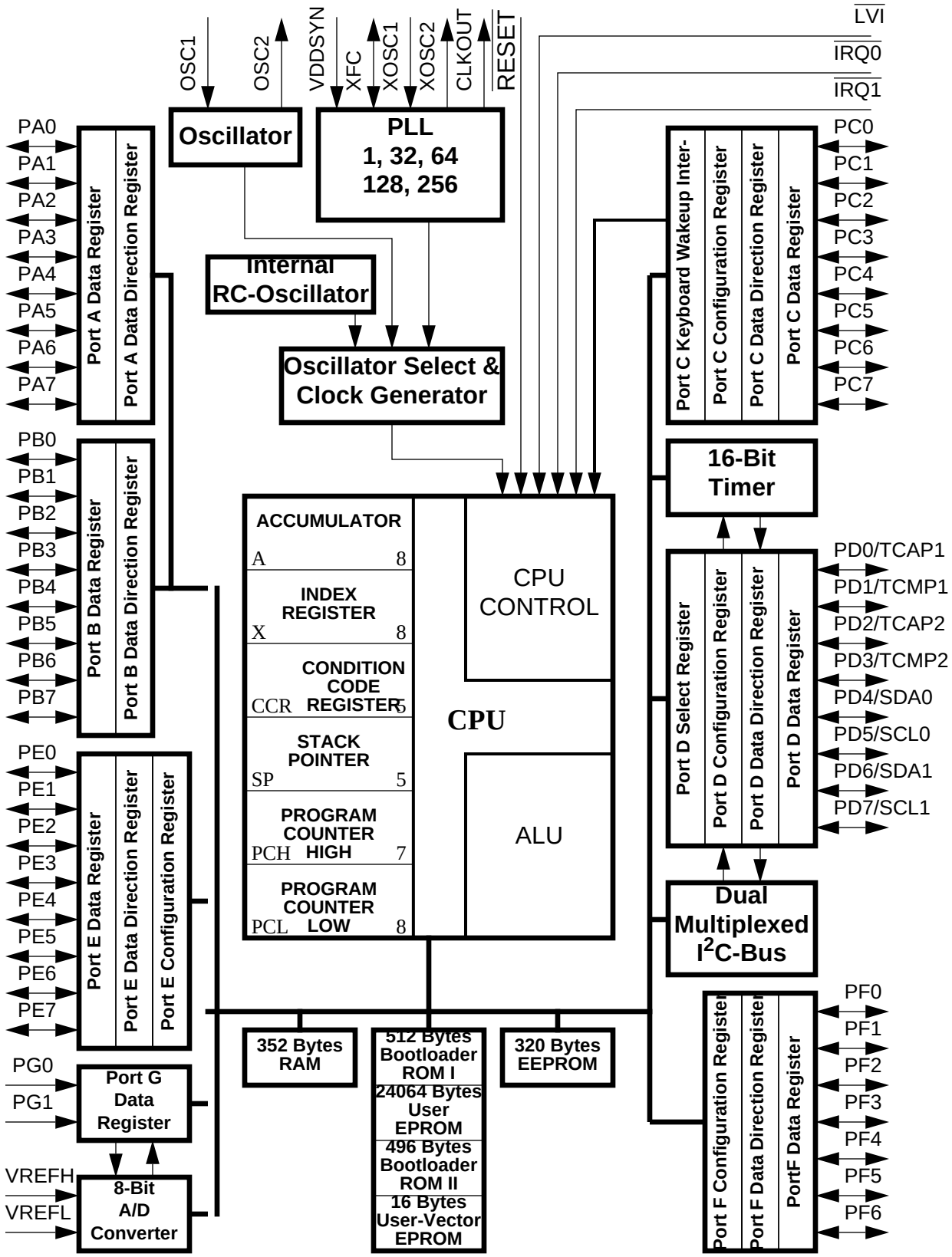


Figure 1-2 ***Block Diagram of the 68HC705E24***

1.3 MASK OPTIONS

There is the following mask options on the 68HC705E24/68HC05E16: Stop Instruction(enable/disable). For the 68HC05E16 the COP enable/disable is also a mask option, while for the 68HC705E24 the COP is enabled/disabled using a write once register. These mask options are programmed during fabrication.

1.4 SIGNAL DESCRIPTION

Active Low signals like $\overline{\text{RESET}}$ will be denoted with either an asterisk or an overline.

1.4.1 VDD AND VSS

Power is supplied to the microcontroller using these two pins. VDD is the positive supply and VSS is ground.

1.4.2 $\overline{\text{IRQ0}}$

The interrupt triggering sensitivity of this pin can be programmed as falling edge or low level. The $\overline{\text{IRQ0}}$ pin contains an internal Schmitt trigger as part of its input to improve noise immunity. Refer to **Section 3.3, INTERRUPTS** for more detail.

1.4.3 $\overline{\text{IRQ1}}$

The interrupt triggering sensitivity of this additional interrupt pin can be programmed as falling edge, rising edge, low level and high level. The $\overline{\text{IRQ1}}$ pin contains an internal Schmitt trigger as part of its input to improve noise immunity. In addition to the $\overline{\text{IRQ0}}$ there is an enable/disable mask-bit and an interrupt flag available for $\overline{\text{IRQ1}}$. Refer to **Section 3.3, INTERRUPTS** for more detail.

1.4.4 $\overline{\text{LVI}}$

This additional interrupt pin provides an interrupt when the supply voltage to the chip increases above a low voltage threshold. The $\overline{\text{LVI}}$ pin contains an internal Schmitt trigger as part of its input to improve noise immunity. Refer to **Section 3.3, INTERRUPTS** for more detail.

1.4.5 OSC1, OSC2

OSC1, OSC2 provide a control input for a ceramic resonator connected to these pins. This oscillator can be chosen instead of the 32kHz/PLL frequency generator if faster start-up times are required.

1.4.6 XOSC1, XOSC2, CLKOUT

XOSC1, XOSC2 provide control input for an on-chip clock oscillator/pll circuit. A crystal, a ceramic resonator (usually a 32.768kHz watch crystal), or an external signal connects to these pins providing a system clock. CLKOUT provides a buffered output of the crystal oscillator.

1.4.7 VDDSYN

This pin provides a separate power connection to the PLL synthesizer which should be at the same potential as VDD.

1.4.8 XFC

This pin provides a means for connecting an external filter capacitor to the synthesizer phase-locked loop filter. Refer to **CHAPTER 9, CORE TIMER** for additional information concerning this capacitor.

1.4.9 $\overline{\text{RESET}}$

This active low pin is used to reset the MCU to a known start-up state by pulling $\overline{\text{RESET}}$ low. The $\overline{\text{RESET}}$ pin contains an internal Schmitt trigger as part of its input to improve noise immunity.

1.4.10 PA0-PA7 / PB0-PB7

These sixteen I/O lines comprise ports A and B. The state of any pin is software programmable. All port A and B lines are configured as inputs during power-on or reset with pull-down resistors on port A and pull-up resistors on Port B. Refer to **Section 5.8, INPUT/OUTPUT PROGRAMMING** for a detailed description of I/O programming.

1.4.11 PC0-PC7 / Keyboard Interrupt

The eight I/O lines of Port C are shared with the keyboard interrupt function. In addition, a configuration register allows the lines to be defined as inputs with pull up resistors, or open drain outputs. Refer to **Section 5.8, INPUT/OUTPUT PROGRAMMING** for a detailed description of I/O programming.

1.4.12 PD0-PD7, TCAP1, TCMP1, TCAP2, TCMP2, SDA0, SCL0, SDA1, SCL1

These eight port lines are shared with other functions to give added flexibility. In addition, a configuration register allows the lines to be defined as inputs with pull up resistors, or open drain outputs. Port pins PD0 and PD1 share pins with Timer input capture(TCAP1) and Timer output compare (TCMP1). Port pins PD2 and PD3 share pins with Timer input capture(TCAP2) and Timer output compare (TCMP2). Refer to **CHAPTER 6, 16 BIT PROGRAMMABLE TIMER** for more information. Port pins PD4 and PD5 share pins with SDA0 and SCL0, the bidirectional data and clock lines for one I^2C bus connection. Port pins PD6 and PD7 share pins with SDA1 and SCL1, the bidirectional data and clock lines for the other I^2C bus connection. Internally the two I^2C buses are multiplexed into one I^2C bus module. Refer to **CHAPTER 7, MULTI-MASTER M-BUS** for more information.

1.4.13 PE0-PE7, PF0-PF6

These fifteen I/O lines comprise Port E and Port F and do not share their pins with any other functions. In addition, a configuration register allows the lines to be defined as inputs with pull up resistors, or open drain outputs. Refer to **Section 5.8, INPUT/OUTPUT PROGRAMMING** for a detailed description of I/O programming.

1.4.14 PG0-PG1, AD0/AD1

These two Port lines are shared with the 8-bit A/D Converter. These lines can also be used as 2 digital inputs. Refer to **CHAPTER 8, A/D CONVERTER** for more information.

1.4.15 VREFH, VREFL

These pins provide the reference voltages for the A/D converter.

CHAPTER 2
OPERATING MODES

The MCU has 4 modes of operation: Single-chip, Test, Bootloader and Emulation. All modes except Single-chip are for factory use only.

2.1 SINGLE-CHIP MODE

In single-chip mode, the address and data buses are not available externally, but there are five 8-bit I/O ports, one 7-bit I/O port and one 2-bit input-only port. Some Port bits are shared with the Timer, I^2C Bus and A/D converter pins. This mode allows the MCU to function as a self-contained microcontroller, with maximum use of the pins for on-chip peripheral functions. All address and data activity occurs within the MCU.

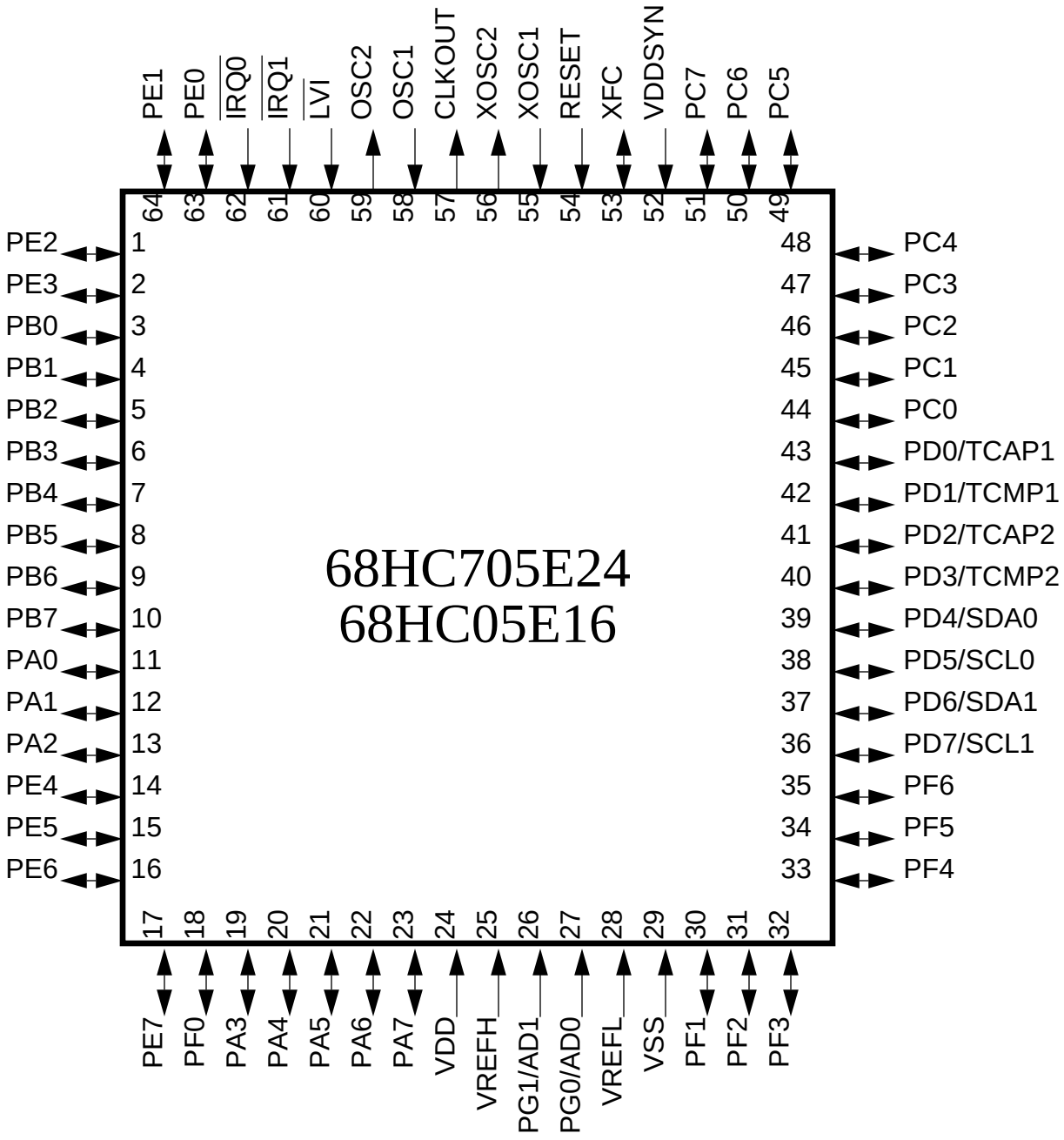


Figure 2-1 Single Chip Mode Pinout of the 68HC705E24/68HC05E16

NOTE: For the 44 Pin QFP- Version (available only for the 68HC05E16) the following pins will be omitted: PE7-PE0, PF6-PF0, PC7-PC3.

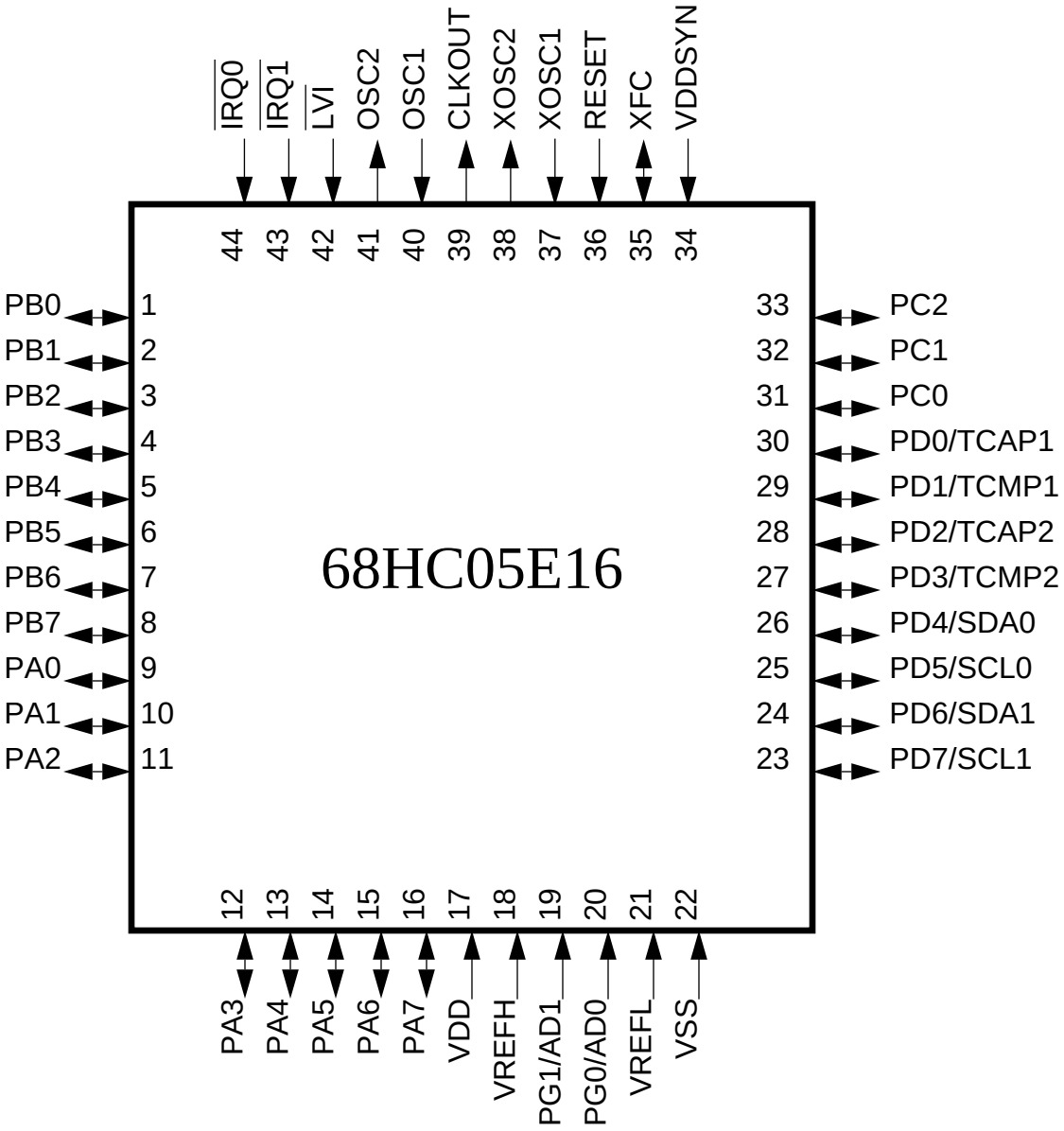


Figure 2-2 Single Chip Mode Pinout of the 68HC05E16 QFP-44

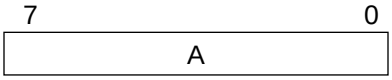
CHAPTER 3
CPU CORE

3.1 REGISTERS

The MCU contains the registers described in the following paragraphs.

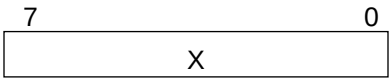
3.1.1 ACCUMULATOR (A)

The accumulator is a general purpose 8-bit register used to hold operands and results of arithmetic calculations or data manipulations.



3.1.2 INDEX REGISTER (X)

The index register is an 8-bit register used for the indexed addressing value to create an effective address. The index register may also be used as a temporary storage area.



3.1.3 CONDITION CODE REGISTER (CCR)

The CCR is a 5-bit register in which the H, N, Z, and C bits are used to indicate the results of the instruction just executed, and the I bit is used to enable interrupts. These bits can be individually tested by a program, and specific actions can be taken as a result of their state. Each bit is explained in the following paragraphs.

3.1.3.1 Half Carry (H)

This bit is set during ADD and ADC operations to indicate that a carry occurred between bits 3 and 4.

3.1.3.2 Interrupt (I)

When this bit is set, the timer and external interrupt is masked (disabled). If an interrupt occurs while this bit is set, the interrupt is latched and processed as soon as the I bit is cleared.

3.1.3.3 Negative (N)

When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was negative.

3.1.3.4 Zero (Z)

When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was zero.

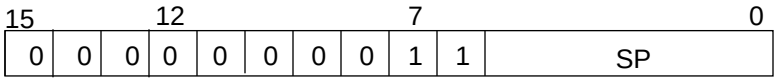
3.1.3.5 Carry/Borrow (C)

When set, this bit indicates that a carry or borrow out of the arithmetic logical unit (ALU) occurred during the last arithmetic operation. This bit is also affected during bit test and branch instructions and during shifts and rotates.

3.1.4 STACK POINTER (SP)

The stack pointer contains the address of the next free location on the stack. During an MCU reset or the reset stack pointer (RSP) instruction, the stack pointer is set to location $00FF. The stack pointer is then decremented as data is pushed onto the stack and incremented as data is pulled from the stack.

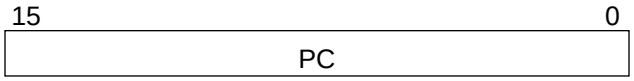
When accessing memory, the ten most significant bits are permanently set to 0000000011. These ten bits are appended to the six least significant register bits to produce and address within the range of $00FF to $00C0. Subroutines and interrupts may use up to 64 (decimal) locations. If 64 locations are exceeded, the stack pointer wraps around and loses the previously stored information. A subroutine call occupies two locations on the stack; an interrupt uses five locations.



3.1.5 PROGRAM COUNTER (PC)

The program counter is a 16-bit register that contains the address of the next byte to be fetched.

NOTE: The HC05 CPU core is capable of addressing 16-bit locations. For this implementation, however, the addressing registers are limited to a 32k byte memory map.



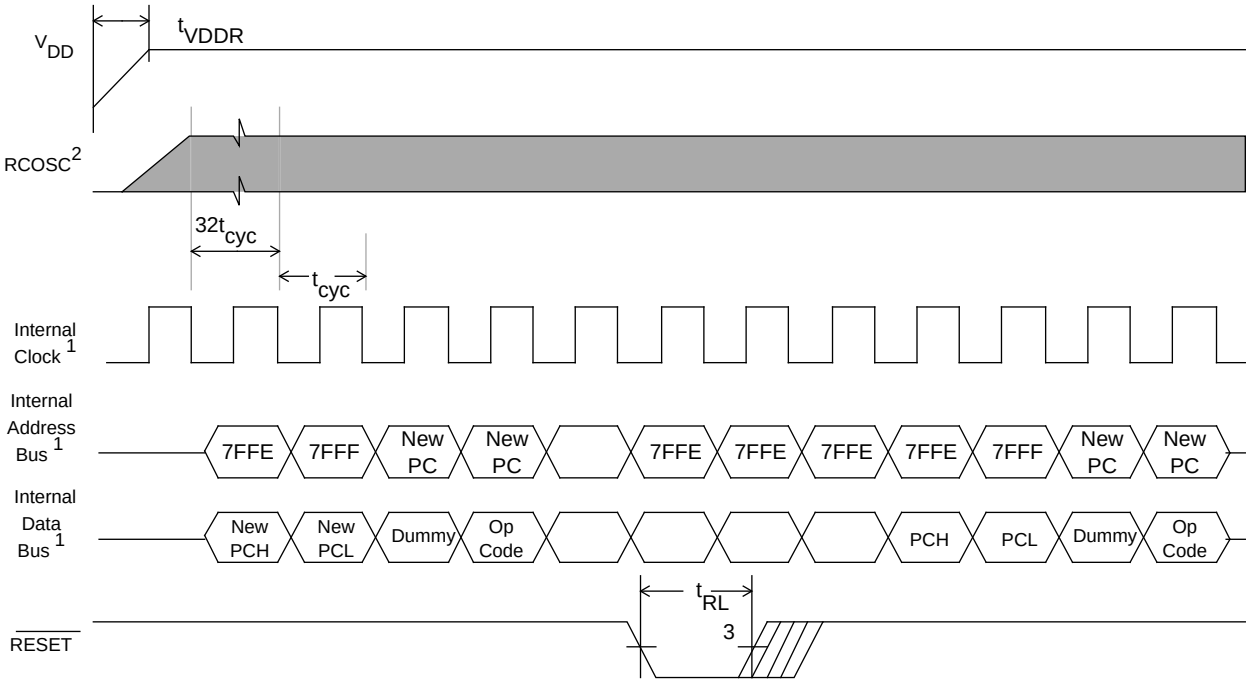
3.2 RESETS

The MCU can be reset four ways: by the initial power-on reset function, by an active low input to the $\overline{\text{RESET}}$ pin, by a COP watchdog-timer reset, and by an opcode fetch from an illegal address.

3.2.1 POWER-ON RESET (POR)

An internal reset is generated on power-up to allow the internal clock generator to stabilize. The power-on reset is strictly for power turn-on conditions and should not be used to detect a drop in the power supply voltage. There is a 32 internal processor clock cycle (t_{cyc}) oscillator stabilization delay after the internal RC-oscillator becomes active. At the end of this 32 cycle delay, if the RESET pin is low, the MCU will remain in the reset condition until RESET goes high (i.e. if the pin is being used as an input).

.



NOTES:
1. Internal timing signal and bus information not available externally.
2. RC-OSC line is not meant to represent frequency. It is only used to represent time.
3. The next rising edge of the internal processor clock following the rising edge of RESET initiates the reset sequence.

Figure 3-1 Power-On Reset and RESET

3.2.2 RESET PIN

The MCU is reset when a logic zero is applied to the RESET input for a period of one and one-half machine cycles (t_{cyc}).

3.2.3 COMPUTER OPERATING PROPERLY (COP) RESET

The MCU contains a watchdog timer that automatically times out if not reset (cleared) within a specific time by a program reset sequence. If the COP watchdog timer is allowed to time-out, an internal reset is generated to reset the MCU. Because the internal reset signal is used, the MCU comes out of a COP reset in the same operating mode it was in when the COP time-out was generated.

3.2.4 ILLEGAL ADDRESS RESET

When an opcode fetch occurs from an address which is not implemented in the RAM ($0040 - $019F), EEROM ($1C0 - $2FF) or EPROM/BOOTROM ($1E00-$7FFF), the part is automatically reset.

NOTE: No RTS or RTI instruction should be placed at the end of a memory block (i.e. at addresses $19F and $2FF) since this results in an ILLEGAL ADDRESS RESET.

3.3 INTERRUPTS

The MCU can be interrupted in eight different ways: the seven maskable hardware interrupts ($\overline{\text{IRQ0}}$, $\overline{\text{IRQ1}}$, $\overline{\text{LVI}}$, core timer, keyboard-wakeup and 16-bit Timer and M-Bus) and the nonmaskable software interrupt instruction (SWI).

Interrupts cause the processor to save register contents on the stack and to set the interrupt mask (I bit) to prevent additional interrupts. The RTI instruction causes the register contents to be recovered from the stack and normal processing to resume.

Unlike $\overline{\text{RESET}}$, hardware interrupts do not cause the current instruction execution to be halted, but are considered pending until the current instruction is complete.

NOTE: The current instruction is the one already fetched and being operated on.

When the current instruction is complete, the processor checks all pending hardware interrupts. If interrupts are not masked (CCR I bit clear) and the corresponding interrupt enable bit is set, the processor proceeds with interrupt processing; otherwise, the next instruction is fetched and executed.

If both an external interrupt and a timer interrupt are pending at the end of an instruction execution, the external interrupt is serviced first. The SWI is executed the same as any other instruction, regardless of the I-bit state.

Register	Flag Name	Interrupts	CPU Interrupt	Vector Address
N/A	N/A	Reset	$\overline{\text{RESET}}$	$7FFE-$7FFF
N/A	N/A	Software Interrupt	SWI	$7FFC-$7FFD
N/A	N/A	External Interrupt 0	$\overline{\text{IRQ0}}$	$7FFA-$7FFB
CTCSR	TOF	Core Timer Overflow	CTIMER	$7FF8-$7FF9
CTCSR	RTI	Real Time Interrupt	CTIMER	$7FF8-$7FF9
CPICSR	CPIF	Custom Periodic Int.	CTIMER	$7FF8-$7FF9
LVICSR	LVIINT	Low Voltage Int.	$\overline{\text{LVI}}$	$7FF6-$7FF7
TSR	ICF1	Timer Input Capture 1	TIMER	$7FF4-$7FF5
TSR	OCF1	Timer Output Compare 1	TIMER	$7FF4-$7FF5
TSR	ICF2	Timer Input Capture 2	TIMER	$7FF4-$7FF5
TSR	OCF2	Timer Output Compare 2	TIMER	$7FF4-$7FF5
TSR	TOF	Timer Overflow	TIMER	$7FF4-$7FF5
MSR	MIF	M-Bus I/O completed	MBUS	$7FF2-$7FF3
IRQ1KWI	IRQ1F	External Interrupt 1	$\overline{\text{IRQ1}}$	$7FF0-$7FF1
IRQ1KWI	KWIF	Keyboard Wakeup Int.	$\overline{\text{IRQ1}}$	$7FF0-$7FF1

Table 3-1 Vector Address for Interrupts and Reset

3.3.1 HARDWARE CONTROLLED INTERRUPT SEQUENCE

The following three functions ($\overline{\text{RESET}}$, STOP, and WAIT) are not in the strictest sense an interrupt; however, they are acted upon in a similar manner. See **Figure 3-2,Interrupt Processing Flowchart** and **Figure 3-9, STOP/WAIT Flowcharts**. A discussion is provided below.

1. $\overline{\text{RESET}}$ - A low input on the $\overline{\text{RESET}}$ input pin causes the program to vector to its starting address which is specified by the contents of memory locations $7FFE and $7FFF. The I bit in the condition code register is also set. Much of the MCU is configured to a known state during this type of reset as previously described in **Section 3.2, RESETS**.
2. STOP - The STOP instruction causes the oscillators to be turned off and the processor to “sleep” until an external interrupt ($\overline{\text{IRQ0}}$, $\overline{\text{IRQ1}}$, $\overline{\text{LVI}}$ or Keyboard) or reset occurs.

3. WAIT - The WAIT instruction causes all processor clocks to stop, but leaves the timer clock running. This “rest” state of the processor can be cleared by reset, an external interrupt (IRQ0, IRQ1, LVI or Keyboard), or internally by core timer, 16-Bit timer or MBUS in slave operation interrupt. There are no special wait vectors for these individual interrupts.

3.3.2 SOFTWARE INTERRUPT (SWI)

The SWI is an executable instruction and a non-maskable interrupt: it is executed regardless of the state of the I bit in the CCR. If the I bit is zero (interrupts enabled), SWI executes after interrupts which were pending when the SWI was fetched, but before interrupts generated after the SWI was fetched. The interrupt service routine address is specified by the contents of memory locations $7FFC and $7FFD.

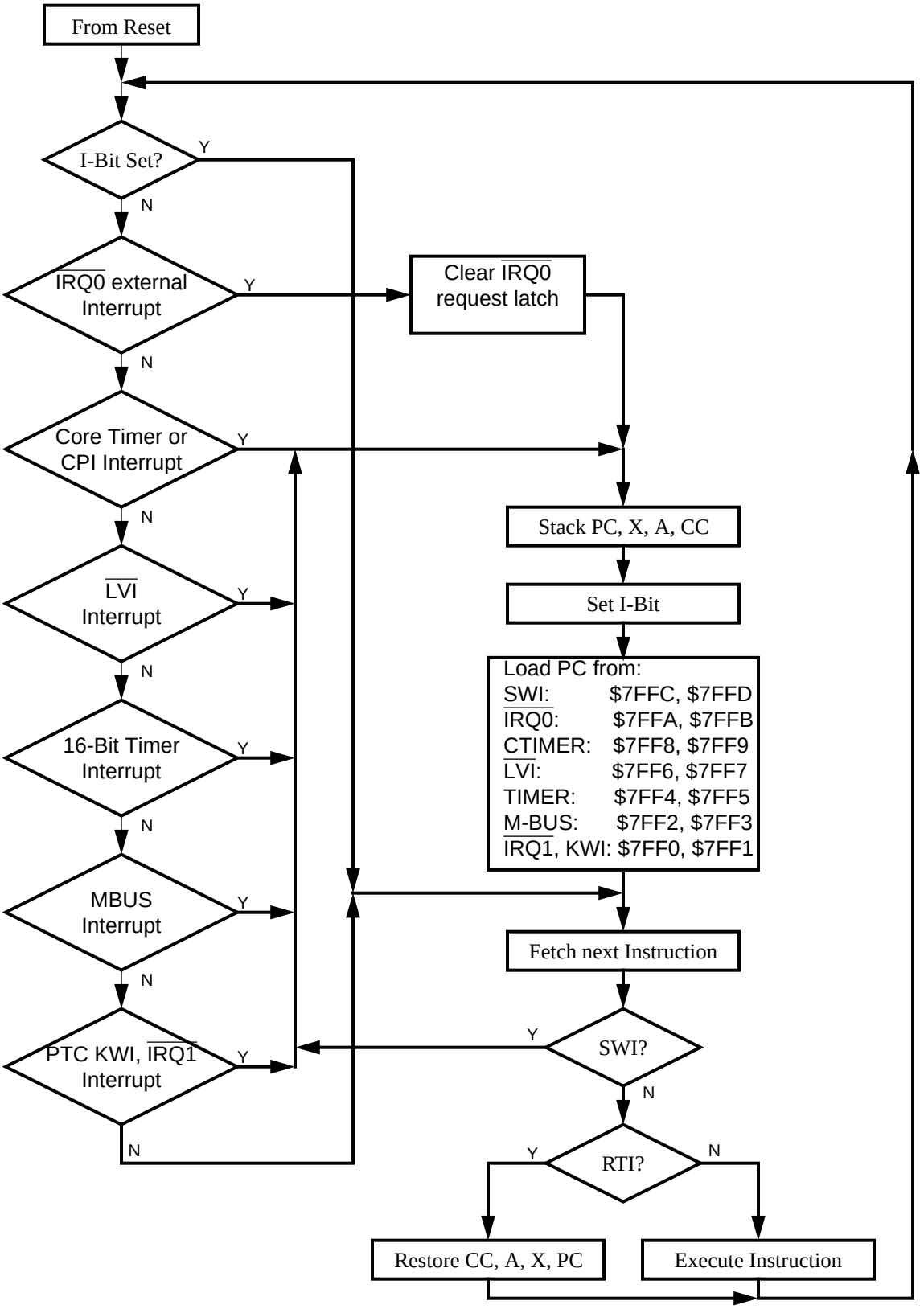


Figure 3-2 Interrupt Processing Flowchart

3.3.3 EXTERNAL INTERRUPT ($\overline{\text{IRQ0}}$)

If the interrupt mask bit (I bit) of the CCR is set, all maskable interrupts (internal and external) are disabled. Clearing the I bit enables interrupts. The interrupt request is latched immediately following the falling edge of $\overline{\text{IRQ0}}$. It is then synchronized internally and serviced by the interrupt service routine located at the address specified by the contents of $7FFA and $7FFB.

	7	6	5	4	3	2	1	0
$0F	LVIINT	LVIVAL	LVIRST	LVIENA	0	0	OPTCOP	OPTIRQ0
RESET	0	-	Wonly 0	0	-	-	Wonce0	0

Figure 3-3 $\overline{\text{LVI}}$/OPTIONS Status/Control Register

Either a level-sensitive and edge-sensitive trigger, or an edge-sensitive-only trigger is available by modifying Bit 0 of register $0F (OPTIRQ0). At reset this bit is cleared so that the $\overline{\text{IRQ0}}$ interrupt is edge-sensitive only. This read/write bit must be set by software to get an level-sensitive and edge-sensitive trigger.

The OPTCOP enables/disables the COP. This bit is only available on the 68HC705E24. This bit is cleared by reset thus enabling the COP. Writing a 1 disables the COP. This bit can be written only once, so that the COP state can not be changed after the first write to this register. It is recommended to write to this register as soon as possible after reset in order to lock the state of the COP.

This bit is not available on the 68HC05E16 and reads always as 0. The COP is enabled/ disabled by an mask option programmed during fabrication.

NOTE: The internal interrupt latch is cleared in the first part of the interrupt service routine; therefore, one external interrupt pulse could be latched and serviced as soon as the I bit is cleared.

3.3.4 CORE TIMER (CTIMER) INTERRUPT

There are three different Core timer interrupt flags that cause a CTIMER interrupt whenever they are set and enabled. The interrupt flags and enable bits are located in the CTIMER Control and Status Register (CTCSR). Any of these interrupts will vector to the same interrupt service routine, located at the address specified by the contents of memory location $7FF8 and $7FF9.

3.3.5 CUSTOM PERIODIC (CPI) INTERRUPT

There is a timer interrupt flag that cause a CPI interrupt whenever set and enabled. The interrupt flag and enable bits are located in the CPI Control and Status Register (CPICSR). An interrupt will vector to the same interrupt service routine as the core timer interrupts, located at the address specified by the contents of memory location $7FF8 and $7FF9.

3.3.6 LOW VOLTAGE INTERRUPT

For the following description see also **Figure 3-4, LVI Power-On Sequence**, **Figure 3-5, LVI Power-Down Sequence** and **Figure 3-6, LVI Recover Sequence**. This interrupt has a dedicated interrupt vector at $7FF6 and $7FF7. It is generated by different conditions of the $\overline{\text{LVI}}$ input pin. The $\overline{\text{LVI}}$ input is used as a “Low Voltage Indicator”. For proper function it should be driven low if the supply voltage is below a useful operating level. The main difference to a normal reset is that a $\overline{\text{LVI}}$ becoming low does not reset the CPU but can be detected by the user program. The level of the $\overline{\text{LVI}}$ input can be tested by reading bit 6 of register $0F (LVIVAL). After detecting the low power condition the user program should save all useful information for a proper shutdown. An high to low transition of the $\overline{\text{LVI}}$ pin can also cause an interrupt if enabled by setting the LVIENA bit of register $0F. The occurrence of a valid interrupt can be monitored by reading the flag bit LVIINT. The interrupt service routine must reset the interrupt by writing a 1 to the LVIRST bit (This bit always reads as 0). If the supply voltage stays above the data retention level and then rises again to the normal operating range along with a low to high transition of $\overline{\text{LVI}}$ an interrupt wakes up the CPU. The system clock is restarted if it was halted using the STOP instruction. (All interrupts which should not wake up the CPU should be disabled prior to entering the STOP-mode).
If the CPU performs a power-on-reset due to a supply voltage below the power-on trip level, no interrupt will be performed and the CPU start will be delayed until $\overline{\text{LVI}}$ becomes high. (To disable the $\overline{\text{LVI}}$ feature, connect a resistor from $\overline{\text{LVI}}$ to VDD).

.

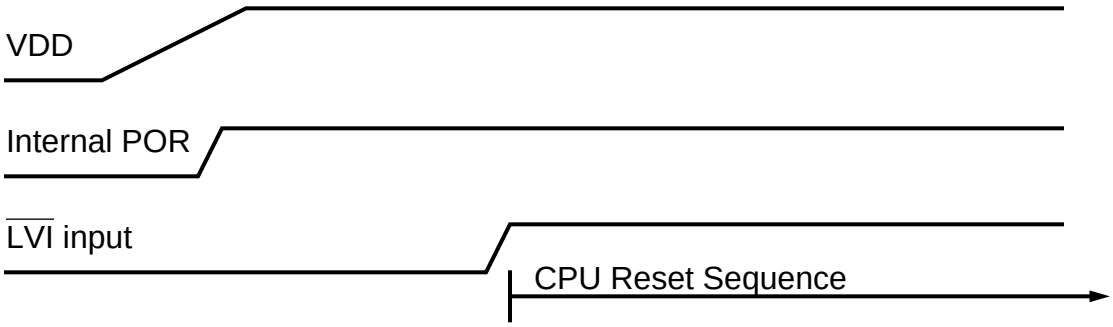


Figure 3-4 $\overline{\text{LVI}}$ Power-On Sequence

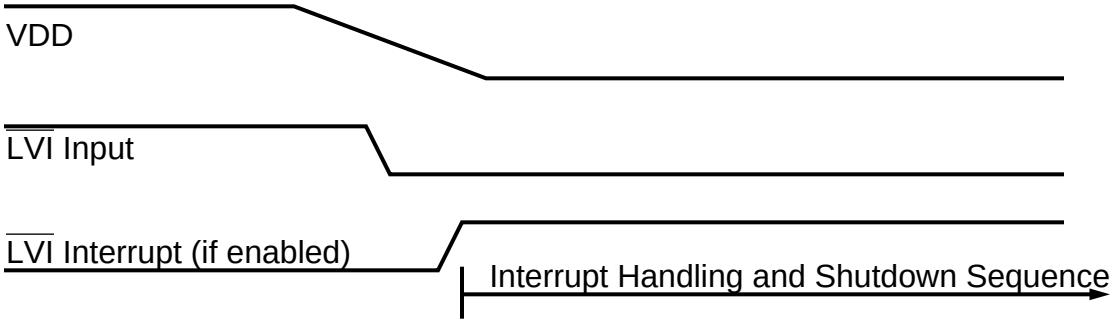


Figure 3-5 $\overline{\text{LVI}}$ Power-Down Sequence

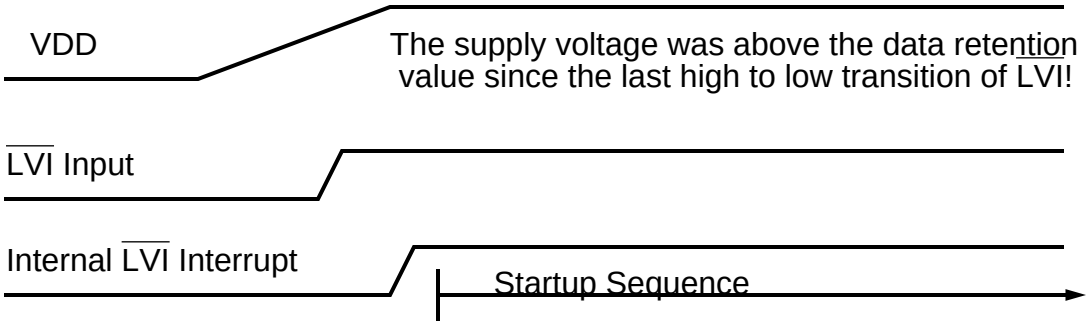


Figure 3-6 LVI Recover Sequence

3.3.7 TIMER (16 BIT) INTERRUPTS

There are three different timer interrupt flags for the 16-bit timer that cause a timer interrupt whenever they are set and enabled. The timer interrupt enable bits are located in the timer control registers (TCR1 and TCR2) and the timer interrupt flags are located in the timer status registers (TSR). Either of these interrupts will vector to the same service routine, located at the address specified by the contents of memory locations $7FF4 and $7FF5.

3.3.8 M-BUS INTERRUPTS

There is an interrupt flag together with three status flags for the M-BUS that cause an M-BUS interrupt whenever it is set and enabled. These interrupts will vector to the service routine located at the address specified by the contents of memory locations $7FF2 and $7FF3.

3.3.9 $\overline{\text{IRQ1}}$ INTERRUPT

In contrary to the $\overline{\text{IRQ0}}$ interrupt the $\overline{\text{IRQ1}}$ interrupt is much more flexible since it can be enabled/disabled independently and there is a selection of falling edge, falling edge & low level, rising edge and rising edge & high level sensitivity. The interrupt vector location for this interrupt is at $7FF0, $7FF1. For the following discussion refer to **Figure 3-3, LVI/ OPTIONS Status/Control Register** and **Table 3-2, IRQ1 Interrupt Sensitivity**. The interrupt is enabled by setting bit 4 of register $1B and disabled by clearing this bit. The $\overline{\text{IRQ1}}$ interrupt vector is shared with the Keyboard Interrupt. In order to distinguish between these two interrupts there is an interrupt flag, bit 5 of register $1B. To clear the interrupt a 1 should be written to the IRQ1RST which always reads as 0.The current status of the $\overline{\text{IRQ1}}$ pin can be monitored by reading the IRQ1VAL of register $1B. The bits IRQ1LV and IRQ1EDG are used to select different interrupt sensitivities according to the following table.

IRQ1LV	IRQ1EDG	interrupt sensitivity
0	0	falling edge
0	1	rising edge
1	0	falling edge & low level
1	1	rising edge & high level

Table 3-2 $\overline{\text{IRQ1}}$ Interrupt Sensitivity

Bit	7	6	5	4	3	2	1	0
$1B	KWIINT	KWIENA	IRQ1INT	IRQ1ENA	IRQ1LV	IRQ1EDG	IRQ1RST	IRQ1VAL
RESET	0	0	0	0	0	0	Wonly 0	-

Figure 3-7 $\overline{\text{IRQ1}}$/KWI Status/Control Register.

3.3.10 KEYBOARD WAKEUP INTERRUPT

This interrupt shares the vector with the $\overline{\text{IRQ1}}$ interrupt at $7FF0, $7FF1. All Port C lines configured as an input contribute to this wired-or type interrupt. An high to low transition causes the interrupt. This interrupt can be enabled/disabled by setting/clearing bit 6 of register $1B. The resulting interrupt flag can be tested as bit 7 of register $1B. The interrupt is cleared by reading port C.

3.4 LOW-POWER MODES

3.4.1 STOP

The STOP instruction places the MCU in its lowest power consumption mode. In STOP mode, the internal oscillator is turned off, halting all internal processing, including timer (and COP Watchdog timer) operation.

The I bit in the CCR is cleared to enable external interrupts. All other registers, including the remaining bits in the CTCSR, and memory remain unaltered. All input/output lines remain unchanged. The processor can be brought out of the STOP mode only by an external interrupt, ($\overline{LVI}$, $\overline{IRQ0}$, $\overline{IRQ1}$ or Keyboard-wakeup) or $\overline{RESET}$.

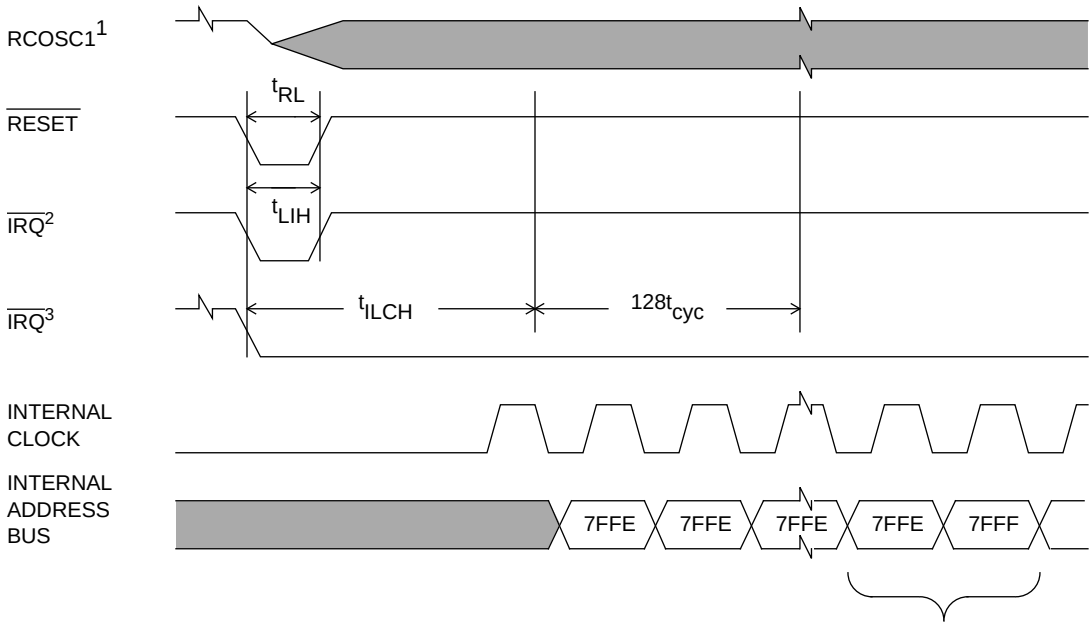
Refer also to **CHAPTER 12, OSCILLATOR SYSTEM** for the behaviour of the different oscillators during stop mode and stop recovery.

The STOP instruction can be disabled by a mask option. When disabled, the STOP instruction is executed as a NOP.

3.4.2 STOP RECOVERY

The processor can be brought out of the STOP mode only by an external interrupt ($\overline{LVI}$, $\overline{IRQ0}$, $\overline{IRQ1}$ or keyboard-wakeup) or $\overline{RESET}$. See **Figure 3-8, Stop Recovery Timing**.

.



Notes:
1. Represents the internal rating of the RCOSC output.
2. $\overline{IRQ}$ pin edge-sensitive selection.
3. $\overline{IRQ}$ pin level and edge sensitive selection.
4. $\overline{IRQ}$ can denote either $\overline{IRQ0}$, $\overline{IRQ1}$ or $\overline{LVI}$.

Figure 3-8 Stop Recovery Timing

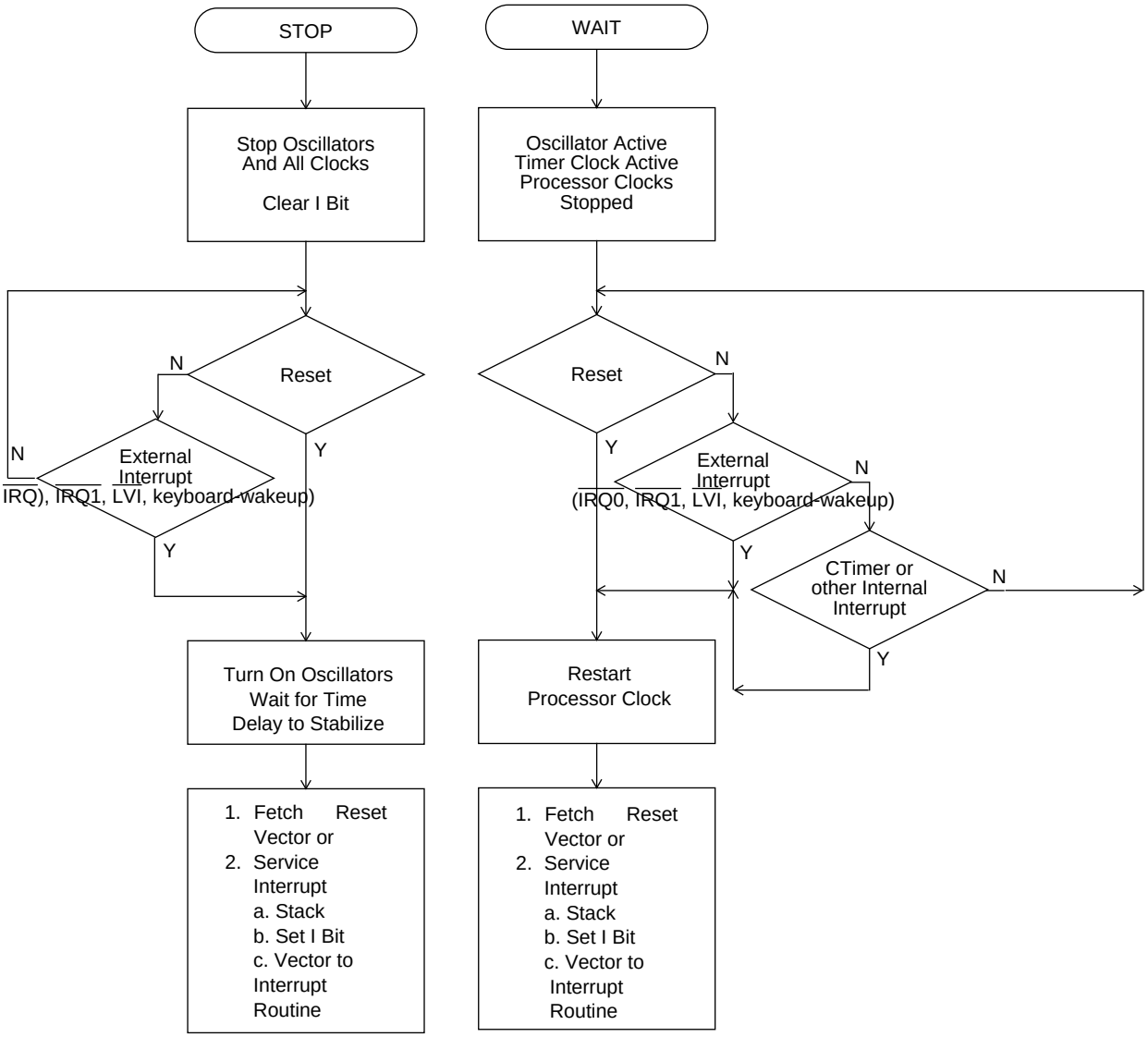


Figure 3-9 STOP/WAIT Flowcharts

3.4.3 WAIT

The WAIT instruction places the MCU in a low-power consumption mode, but the WAIT mode consumes more power than the STOP mode. All CPU action is suspended, but the Core timer and the 16-Bit timer remain active. An interrupt from the core timer or the 16-Bit timer will cause the MCU to exit the WAIT mode if enabled.

During the WAIT mode, the I bit in the CCR is cleared to enable interrupts. All other registers, memory, and input/output lines remain in their previous state. The Core timer or 16 Bit Timer may be enabled to allow a periodic exit from the WAIT mode. MBUS in slave mode or Custom Periodic Interrupt may also cause the device to exit WAIT mode.

3.4.4 DATA-RETENTION MODE

The contents of RAM and CPU registers are retained at supply voltages as low as 2.0Vdc. This is called the data retention mode where the data is held, but the device is not guaranteed to operate. $\overline{\text{RESET}}$ must be held low during data-retention mode.

3.5 INSTRUCTION SET

The MCU has a set of 62 basic instructions. They can be divided into five different types: register/memory, read-modify-write, branch, bit manipulation, and control. The following paragraphs briefly explain each type. For more information on the instruction set, refer to the M6805 Family User's Manual (M6805UM/AD2) or the MC68HC05C4 Data Sheet (MC68HC05C4/D).

3.5.1 REGISTER/MEMORY INSTRUCTIONS

Most of these instructions use two operands. One operand is either the accumulator or the index register. The other operand is obtained from memory using one of the addressing modes. The jump unconditional (JMP) and jump to subroutine (JSR) instructions have no register operand. Refer to the following instruction list

.

Function	Mnemonic
Load A from Memory	LDA
Load X from Memory	LDX
Store A in Memory	STA
Store X in Memory	STX
Add Memory to A	ADD
Add Memory and Carry to A	ADC
Subtract Memory	SUB
Subtract Memory from A with Borrow	SBC
AND Memory to A	AND
OR Memory with A	ORA
Exclusive OR Memory with A	EOR
Arithmetic compare A with Memory	CMP
Arithmetic Compare X with Memory	CPX
Bit Test Memory with A (Logical Compare)	BIT
Jump Unconditional	JMP
Jump to Subroutine	JSR
Multiply	MUL

3.5.2 READ-MODIFY-WRITE INSTRUCTIONS

These instructions read a memory location or a register, modify or test its contents, and write the modified value back to memory or to the register. The test for negative or zero (TST) instruction is an exception to the read-modify-write sequence since it does not modify the value. Do not use these read-modify-write instructions on write-only locations. Refer to the following list of instructions.

Function	Mnemonic
Increment	INC
Decrement	DEC
Clear	CLR
Complement	COM
Negate (Twos Complement)	NEG
Rotate Left Thru Carry	ROL
Rotate Right Thru Carry	ROR
Logical Shift Left	LSL
Logical Shift Right	LSR
Arithmetic Shift Right	ASR
Test for Negative or Zero	TST

3.5.3 BRANCH INSTRUCTIONS

This set of instruction branches if a particular condition is met; otherwise, no operation is performed. Branch instructions are two-byte instructions. Refer to the following list for branch instructions.

Function	Mnemonic
Branch Always	BRA
Branch Never	BRN
Branch if Higher	BHI
Branch if Lower or Same	BLS
Branch if Carry Clear	BCC
Branch if Higher or Same	BHS
Branch if Carry Set	BCS
Branch if Lower	BLO
Branch if Not Equal	BNE
Branch if Equal	BEQ
Branch if Half Carry Clear	BHCC
Branch if Half Carry Set	BHCS
Branch if Plus	BPL
Branch if Minus	BMI
Branch if Interrupt Mask Bit is Clear	BMC
Branch if Interrupt Mask Bit is Set	BMS
Branch if Interrupt Line is Low	BIL
Branch if Interrupt Line is High	BIH
Branch to Subroutine	BSR

3.5.4 BIT MANIPULATION INSTRUCTIONS

The MCU is capable of setting or clearing any writable bit which resides in the first 256 bytes of the memory space where all port registers, port DDRs, Ctimer counter, Ctimer control, timer registers and part of on-chip RAM reside. An additional feature allows the software to test and branch on the state of any bit within these 256 locations. The bit set, bit clear and bit test and branch functions are all implemented with a single instruction. For test and branch instructions, the value of the bit tested is also placed in the carry bit of the condition code register. These instructions are also read-modify-write instructions. Do not bit manipulate write-only locations. Refer to the following list for bit manipulation instructions.

Function	Mnemonic
Branch if Bit n is Set	BRSET n (n = 0..7)
Branch if bit n is Clear	BRCLR n (n = 0..7)
Set Bit n	BSET n (n = 0...7)
Clear Bit n	BCLR n (n = 0...7)

3.5.5 CONTROL INSTRUCTIONS

These instructions are register reference instructions and are used to control processor operation during program execution. Refer to the following list for control instructions.

Function	Mnemonic
Transfer A to X	TAX
Transfer X to A	TXA
Set Carry Bit	SEC
Clear Carry Bit	CLC
Set Interrupt Mask Bit	SEI
Clear Interrupt Mask Bit	CLI
Software Interrupt	SWI
Return from Subroutine	RTS
Return from Interrupt	RTI
Reset Stack Pointer	RSP
No-Operation	NOP
Stop	STOP
Wait	WAIT

3.6 ADDRESSING MODES

The MCU uses ten different addressing modes to provide the programmer with an opportunity to optimize the code for all situations. The various indexed addressing modes make it possible to locate data tables, code conversion tables, and scaling tables anywhere in the memory space. Short indexed accesses are single byte instructions; the longest instructions (three bytes) permit accessing tables throughout memory. Short and long absolute addressing is also included. One- or two-byte direct addressing instructions access all data bytes in most applications. Extended addressing permits jump instructions to reach all memory.

The term “effective address” (EA) is used in describing the various addressing modes. Effective address is defined as the address from which the argument for an instruction is fetched or stored.

3.6.1 IMMEDIATE

In the immediate addressing mode, the operand is contained in the byte immediately following the opcode. The immediate addressing mode is used to access constants that do not change during program execution (e.g., a constant used to initialize a loop counter).

3.6.2 DIRECT

In the direct addressing mode, the effective address of the argument is contained in a single byte following the opcode byte. Direct addressing allows the user to directly address the lowest 256 bytes in memory with a single two-byte instruction.

3.6.3 EXTENDED

In the extended addressing mode, the effective address of the argument is contained in the two bytes following the opcode byte. Instructions with extended addressing mode are capable of referencing arguments anywhere in memory with a single three-byte instruction. When using the Motorola assembler, the user need not specify whether an instruction uses direct or extended addressing. The assembler automatically selects the shortest form of the instruction.

3.6.4 RELATIVE

The relative addressing mode is only used in branch instructions. In relative addressing, the contents of the 8-bit signed offset byte, which is the last byte of the instruction, is added to the PC if, and only if, the branch conditions are true. Otherwise control proceeds to the next instruction. The span of relative addressing is from -128 to +127 from the address of the next opcode. The programmer need not calculate the offset when using the Motorola assembler, since it calculates the proper offset and checks to see that it is within the span of the branch.

3.6.5 INDEXED, NO OFFSET

In the indexed, no offset addressing mode, the effective address of the argument is contained in the 8-bit index register. This addressing mode can access the first 256 memory locations. These instructions are only one byte long. This mode is often used to move a pointer through a table or to hold the address of a frequently referenced RAM or I/O location.

3.6.6 INDEXED, 8-BIT OFFSET

In the indexed, 8-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the unsigned byte following the opcode. The addressing mode is useful for selecting the K^{th} element in an n element table. With this two-byte instruction, K would typically be in X with the address of the beginning of the table in the instruction. As such, tables may begin anywhere within the first 256 addressable locations and could extend as far as location 510 ($1FE). This is the last location which can be accessed in this way.

3.6.7 INDEXED, 16-BIT OFFSET

In the indexed, 16-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the two unsigned bytes following the opcode. This address mode can be used in a manner similar to indexed, 8-bit offset except that this three-byte instruction allows tables to be anywhere in memory. As with direct and extended addressing, the Motorola assembler determines the shortest form of indexed addressing.

3.6.8 BIT SET/CLEAR

In the bit set/clear addressing mode, the bit to be set or cleared is part of the opcode, and the byte following the opcode specifies the direct addressing of the byte in which the specified bit is to be set or cleared. Any read/write bit in the first 256 locations of memory, including I/O, can by selectively set or cleared with a single two-byte instruction.

3.6.9 BIT TEST AND BRANCH

The bit test and branch addressing mode is a combination of direct addressing and relative addressing. The bit that is to be tested and its condition (set or clear), is included in the opcode. The address of the byte to be tested is in the single byte immediately following the opcode byte. The signed relative 8-bit offset in the third byte is added to the PC if the specified bit is set or cleared in the specified memory location. This single three-byte instruction allows the program to branch based on the condition of any readable bit in the first 256 locations of memory. The span of branching is from -128 to +127 from the address of the next opcode. The state of the tested bit is also transferred to the carry bit of the condition code register.

3.6.10 INHERENT

In the inherent addressing mode, all the information necessary to execute the instruction is contained in the opcode. Operations specifying only the index register and/or accumulator as well as the control instruction with no other arguments are included in this mode. These instructions are one byte long.

CHAPTER 4
MEMORY

The 68HC705E24/68HC05E16 has a 32K byte memory map, consisting of user (EP)ROM, user RAM, Bootloader-ROM, EEPROM, and I/O. See **Figure 4-1, Memory Map of the 68HC05E16** and **Figure 4-2, Memory Map of the 68HC705E24**.

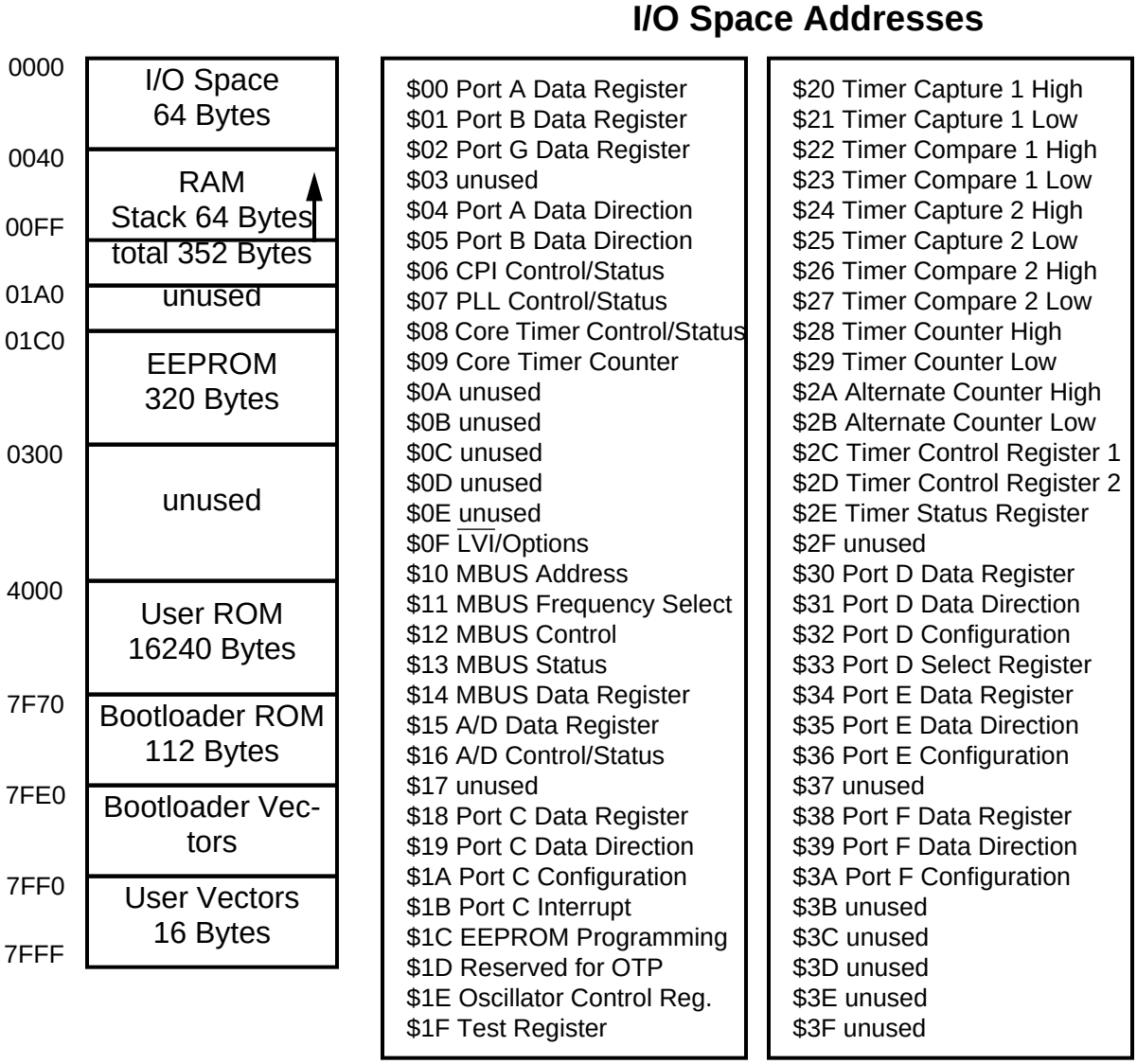


Figure 4-1 Memory Map of the 68HC05E16

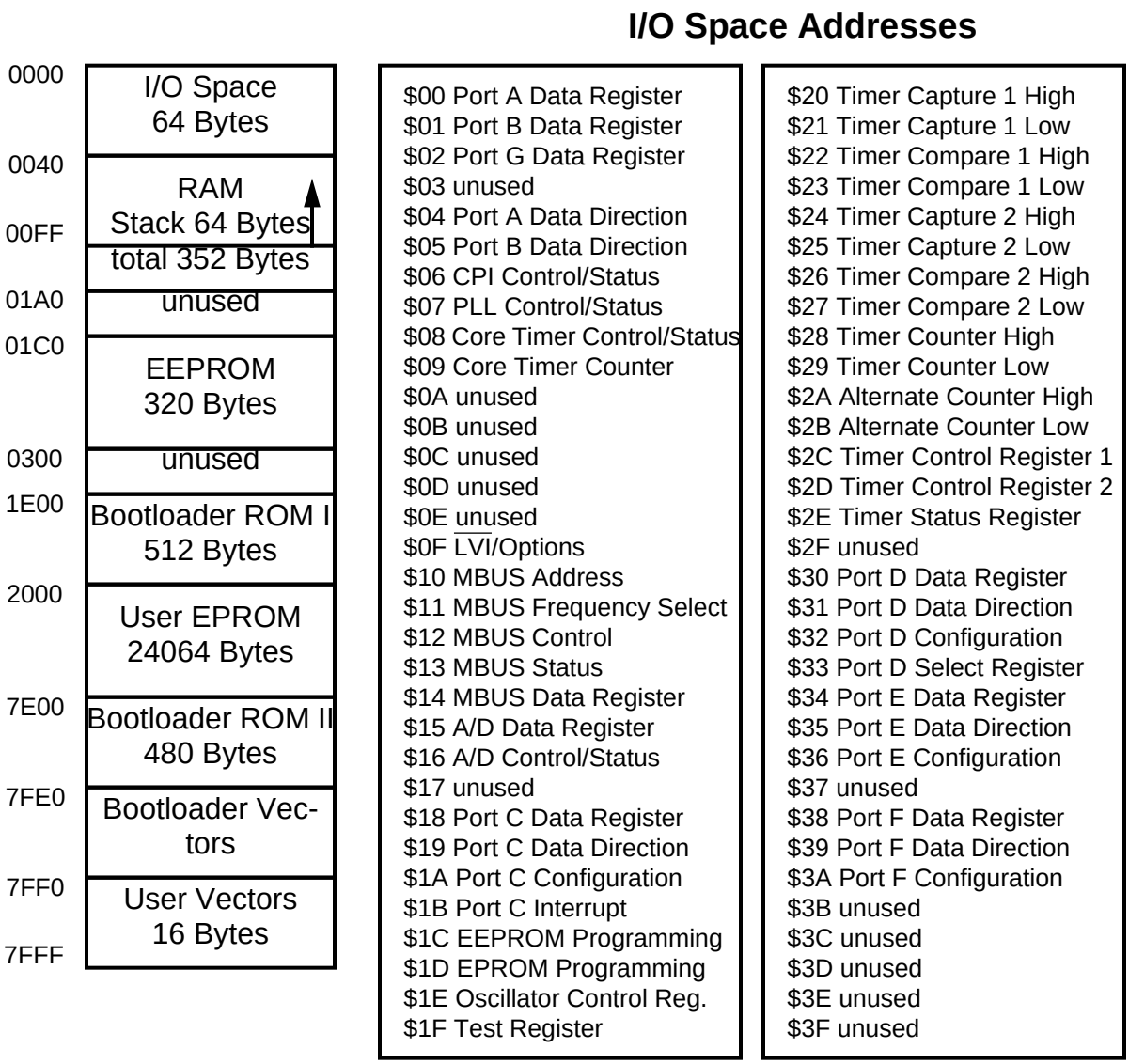


Figure 4-2 Memory Map of the 68HC705E24

4.1 ROM (ONLY 68HC05E16)

The 16384 bytes of mask programmable ROM are located from $4000 to $7FFF, 128 bytes ranging from $7F70 to $7FEF are reserved factory test purposes. The 16 bytes of user vector range from $7FF0 to $7FFF.

4.2 EPROM (ONLY 68HC705E24)

The 24064 bytes of EPROM are located from $2000 to $7DFF, with 16 bytes of user vectors from $7FF0 to $7FFF.

4.2.1 EPROM PROGRAMMING REGISTER $1D.

Bit	7	6	5	4	3	2	1	0
$1D	0	0	0	TS1	TS0	ELATCH	0	EPGM
RESET	0	0	0	0	0	0	0	0

Figure 4-3 EPROM Programming Register

This register is used to program the EPROM array. Only the LATCH and EPGM bits are available in user mode. To program a byte of EPROM, apply the programming voltage V_{pp} to the $\overline{\text{IRQ0}}$ pin, set LATCH, then write data to the desired address and set EPGM for time t_{epgm}. If there is a multibyte programming facility which allows to program up to 16 bytes at a time. After setting the ELATCH bit up to 16 bytes can be written to the desired addresses. The condition which allows multibyte programming is, that the address bits A15-A4 must NOT change, i.e all bytes must be located within the same 16 bytes address block.

4.2.1.1 ELATCH

When set, ELATCH configures the EPROM address and data bus for programming. When ELATCH is set writes to the EPROM array cause the data bus and address bus to be latched. This bit is readable and writable, but reads from the array are inhibited if the ELATCH bit is set and a write to the EPROM space has taken place. When clear address and data bus are configured for normal operation. Reset clears this bit.

4.2.1.2 EPGM

EPGM must be written to enable or disable the EPGM function. When set, EPGM turns on the internal programming voltage switch starts programming the EPROM. When clear the power is switched off. If ELATCH is not set, then EPGM cannot be set. Reset clears this bit.

4.2.1.3 TS1, TS0

These bits are only available in test or bootloader mode. In single-chip mode they are forced to 0.They are intended for different factory tests.

4.3 RAM

The user RAM consists of 352 bytes ranging from $0040 to $019F of a shared stack area. The stack begins at address $00FF. The stack pointer can access 64 bytes of RAM in the range $00FF to $00C0. See also **Section 3.1.4, STACK POINTER (SP)**

4.4 EEPROM

The EEPROM on this device is 320 bytes long and is located at address $01C0. Programming the EEPROM can be done by the user on a single-byte basis by manipulating the Programming Register, located at address $001C.

4.4.1 PROGRAMMING REGISTER $1C

The contents and use of the programming register are discussed below.

Bit	7	6	5	4	3	2	1	0
$1C	0	CPEN	0	ER1	ER0	EELATCH	EERC	EEPGM
RESET	0	0	0	0	0	0	0	0

Figure 4-4 Programming Register

4.4.1.1 CPEN - Charge Pump Enable

When set, CPEN enables the charge pump which produces the internal programming voltage. This bit should be set with the EELATCH bit. The programming voltage will not be available until EEPGM is set. The charge pump should be disabled when not in use. CPEN is readable and writable and is cleared by reset.

4.4.1.2 ER1:ER0 - Erase Select Bits

ER1 and ER0 form a 2-bit field which is used to select one of three erase modes: byte, block, or bulk. **Table 4-1, Erase Mode Select**, shows the modes selected for each bit configuration. These bits are readable and writable and are cleared by reset.

In byte erase mode, only the selected byte is erased. In block mode, a 64-byte block of EEPROM is erased. The EEPROM memory space is divided into five 64-byte blocks ($01C0-$01FF, $0200-$023F, $0240-$027F, $0280-$02BF and $02C0-$02FF) Doing a block erase to any address within a block will erase the entire block. In bulk erase mode, the entire 320 byte EEPROM array is erased.

ER1	ER0	MODE
0	0	No Erase
0	1	Byte Erase
1	0	Block Erase
1	1	Bulk Erase

Table 4-1 Erase Mode Select

4.4.1.3 EELATCH

When set, EELATCH configures the EEPROM address and data bus for programming. When EELATCH is set, writes to the EEPROM array cause the data bus and the address bus to be latched. This bit is readable and writable, but reads from the array are inhibited if the EELATCH bit is set and a write to the EEPROM space has taken place. When clear, address and data buses are configured for normal operation. Reset clears this bit.

4.4.1.4 EERC - EEPROM RC Oscillator Control

When this bit is set, the EEPROM section uses the internal RC oscillator instead of the CPU clock. After setting the EERC bit, delay a time t_{RCON} to allow the RC oscillator to stabilize. This bit is readable and writable and should be set by the user when the internal bus frequency falls below 1.5 MHz. Reset clears this bit.

4.4.1.5 EEPGM - EEPROM Programming Power Enable

EEPGM must be written to enable (or disable) the EEPGM function. When set, EEPGM turns on the charge pump and enables the programming (or erasing) power to the EEPROM array. When clear, this power is switched off. This will enable pulsing of the programming voltage to be controlled internally. This bit can be read at any time, but can only be written to if EELATCH=1. If EELATCH is not set, then EEPGM cannot be set. Reset clears this bit.

4.4.2 PROGRAMMING/ERASING PROCEDURES

To program a byte of EEPROM, set EELATCH = CPEN = 1, set ER1 = ER0 = 0, write data to the desired address and then set EEPGM for a time t_{EPGM}.

NOTE: Any bit should be erased before it is programmed.

To erase a **byte** of EEPROM set EELATCH = 1, CPEN = 1, ER1 = 0 and ER0 = 1, write to the address to be erased, and set EEPGM for a time t_{EBYT}.

To erase a **block** of EEPROM set EELATCH = 1, CPEN = 1, ER1 = 1 and ER0 = 0, write to any address in the block, and set EEPGM for a time t_{EBLOCK}.

For a **bulk** erase set EELATCH = 1, CPEN = 1, ER1 = 1, and ER0 = 1, write to any address in the array, and set EEPGM for a time t_{EBULK}.

To terminate the programming or erase sequence, clear EEPGM, delay for a time t_{FPV} to allow the program voltage to fall, and then clear EELATCH and CPEN to free up the buses. Following each erase or programming sequence, clear all programming control bits.

The following program is an example of the EEPROM programming sequence using the timer to implement the required delay and assuming a 1 MHz bus frequency.

```
TCSR    EQU     $0008           TIMER CONTROL/STATUS REGISTER
TCNT    EQU     $0009           TIMER COUNTER REGISTER
```

```
TOF       EQU     7               TOF BIT OF TCSR
PROG      EQU     $001C           PROGRAM REGISTER
CPEN      EQU     6               CHARGE PUMP ENABLE
ER1       EQU     4               ERASE SELECT 1
ER0       EQU     3               ERASE SELECT 0
EELATCH   EQU     2               LATCH BIT
EERC      EQU     1               RC/OSC SELECTOR
EEPGM     EQU     0               EE PROGRAM BIT
EESTART   EQU     $01C0           STARTING ADDRESS OF EEPROM
SUMPIN    EQU     $FF             DUMMY DATA

          ORG     $4000
START     EQU     *
          BSET    EERC,PROG       SELECT RC OSC
          BSR     DELAY           RC OSC STABILIZATION
          BSET    CPEN,PROG       TURN ON CHARGE PUMP
          BSET    EELATCH,PROG    ENABLE LATCH BIT
          BCLR    ER1,PROG        ENSURE PROGRAM (NOT ERASE)
          BCLR    ER0,PROG        ENSURE PROGRAM (NOT ERASE)
OK        LDA     #SUMPIN         GET DATA
          STA     EESTART
          BSET    EEPGM, PROG     ENABLE PROGRAMMING POWER
          JSR     DELAY           WAIT FOR PROGRAM TIME
          BCLR    EEPGM,PROG      CLEAR EEPGM
          JSR     DELAY           WAIT FOR PROG VOLTAGE TO FALL
          BCLR    EELATCH,PROG    CLEAR LATCH
          BCLR    CPEN,PROG       DISABLE CHARGE PUMP
          CMP     EESTART         VERIFY
          BNE     OUT1
          CLC                     CLEAR CARRY BIT FOR NO ERROR
OUT       RTS
OUT1      SEC                     FLAG AN ERROR
          RTS

* THIS ROUTINE GIVES A 15 MS (+/- 1 MS) DELAY AT A 1Mhz BUS.
* THE SAME DELAY ROUTINE IS USED IN THIS EXAMPLE FOR SIMPLICITY,
* USING THE LONGEST DELAY TIME. USERS WILL WANT TO TRITE SHORTER
* DELAY ROUTINES FOR APPLICATIONS IN WHICH SPEED IS IMPORTANT.

DELAY     EQU     *
          LDX     #15             COUNT OF 15
TIMLP     BCLR    TOF,TCSR        CLEAR TOF
          BRCLR   TOF,TCSR,*      WAIT FOR TOF FLAG
          DECX
          BNE     TIMLP           COUNT DOWN TO 0
          RTS
```

CHAPTER 5
INPUT/OUTPUT PORTS

In single-chip mode there are 49 lines arranged as four 8-bit I/O ports, one 7-bit I/O port, one 8-bit port which shares various I/O configurations with the TIMER and MBUS subsystems and one 2-bit input-only port shared with the A/D converter. The I/O ports are programmable for different functions as explained in the remainder of this chapter.

NOTE: To avoid a glitch on the output pins, write data to the I/O Port Data Register before writing a one to the corresponding Data Direction Register.

5.1 PORT A

Port A is an 8-bit bidirectional port which does not share any of its pins with other subsystems. The port A data register is at $0000 and the data direction register (DDR) is at $0004. Reset does not affect the data register, but clears the data direction register, thereby returning the ports to inputs. Writing a one to a DDR bit sets the corresponding port bit to output mode. There are fixed pull-down resistors on this port if configured as input.

5.2 PORT B

Port B is an 8-bit bidirectional port which does not share any of its pins with other subsystems. The port B data register is at $0001 and the data direction register (DDR) is at $0005. Reset does not affect the data register, but clears the data direction register, thereby returning the ports to inputs. Writing a one to a DDR bit sets the corresponding port bit to output mode. There are fixed pull-up resistors on this port if configured as input.

5.3 PORT C

Port C is an 8-bit bidirectional port which is equipped with a keyboard interrupt and does not share any of its pins with other subsystems. There are three read/write registers associated to the port selecting the different functions. The register locations are as shown in **Table 5-1, Port C Register Addresses**.

Data Register	PTCIO	$18
Data Direction Register	PTCDDR	$19
Configuration Register	PTCCFR	$1A
Keyboard-wakeup	PTCIRQ	$1B

Table 5-1 Port C Register Addresses

All registers except the Data Register are cleared by reset thereby turning all ports to normal inputs with pull-up resistor.

The Data Direction Register along with the ConFiguration Register determines the behaviour of the I/O Port according to **Table 5-2, I/O Port Configuration Functions**.

DDR	CFR	Function
0	0	input with pull-up
0	1	input without pull-up
1	0	push-pull output
1	1	open-drain output

Table 5-2 I/O Port Configuration Functions

Every Port C pin configured as an input contributes to the wired or keyboard interrupt. So a single pin can be disabled by configuring it as an output.

5.3.1 KEYBOARD INTERRUPT

For the following discussion refer also to **Section 3.3, INTERRUPTS**, **Section 3.3.9, IRQ1 INTERRUPT** and **Section 3.3.10, KEYBOARD WAKEUP INTERRUPT**. The keyboard interrupt is enabled by setting the keyboard interrupt enable bit (Bit 6, $1B), These input lines have an internal pull-up. This pull-up can be switched off setting the corresponding Configuration Register bit to 1. Once a High to Low transition is sensed on any of the lines of PC0-PC7 configured as input, provided the interrupt mask bit of the condition code register is cleared, a keyboard interrupt is generated and the keyboard status flag (Bit 7, $1B) is set. The interrupt service routine is specified by the contents of the memory locations $7FF0 and $7FF1. The keyboard interrupt vector address is shared with the $\overline{\text{IRQ1}}$, external interrupt vector thus to distinguish between a keyboard interrupt and an $\overline{\text{IRQ1}}$ interrupt the service routine must check the keyboard status flag (bit 7,$1B). The interrupt is cleared by reading the Port C data register thus obtaining the current values of the I/O pins. The keyboard interrupt is edge and level sensitive. A keyboard interrupt will force the processor out of STOP or WAIT mode.

NOTE: The keyboard interrupt enable and status flags are bit 6 and 7 of the KWI/$\overline{\text{IRQ1}}$ Status/Control Register located at address $1B together with the $\overline{\text{IRQ1}}$ enable and status flags.

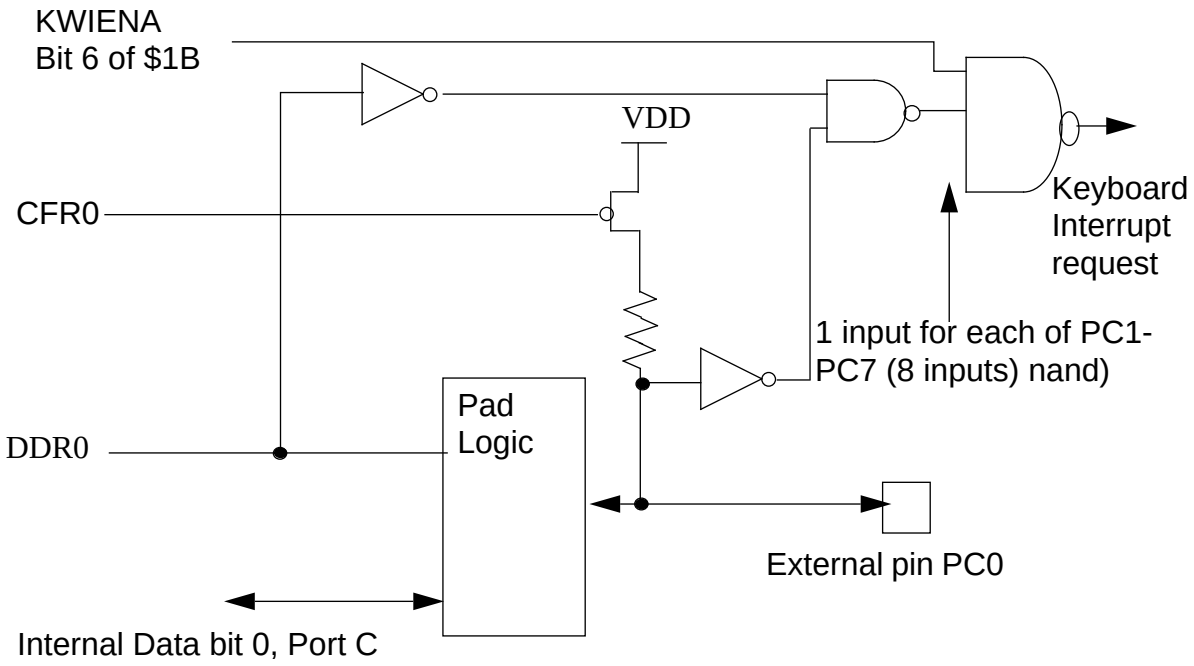


Figure 5-1 Keyboard Interrupt Configuration

5.4 PORT D

Port D is an 8-bit port which shares its pins with the TIMER and M-BUS subsystems. There are four read/write registers associated to the port selecting the different functions. All the ports in Port D can be configured as input/output pins or can be used by other systems within the MCU. The register locations are as shown in **Table 5-1, Port C Register Addresses**.

Data Register	PTDIO	$30
Data Direction Register	PTDDDR	$31
Configuration Register	PTDCFR	$32
Function Select Register	PTDSEL	$33

Table 5-3 Port D Register Addresses

All registers except the Data Register are cleared by reset thereby turning all ports to normal inputs with pull-up resistor.

The Data Direction Register and the ConFiguration Register determine the behaviour of the I/O Port according to **Table 5-2, I/O Port Configuration Functions**. Setting bits 7-0

in the Port D select register to logical “1” configures the pin to be dedicated to the TIMER or M-BUS subsystems.

NOTE: This select bit overwrites the corresponding Data Direction Bit!

Functional description of each bit is as follows:

5.4.1 PD0/TCAP1 - PORTD BIT 0

This pin is set as port D bit-0 I/O when the bit-0 in the PortD select register is ’0’ and becomes TIMER input capture 1 input when the select bit is set to ’1’. Setting the configuration bit-0 to 1 switches off the pull-up resistor.

5.4.2 PD1/TCMP1 - PORTD BIT 1

This pin is set as port D bit-1 I/O when the bit-1 in the PortD select register is ’0’ and is configured as TIMER output compare 1 output when the corresponding select bit is set to ’1’. Setting the configuration bit-0 to 1 makes it an open-drain output.

5.4.3 PD2/TCAP2 - PORTD BIT 2

This pin is set as port D bit-2 I/O when the bit-2 in the PortD select register is ’0’ and becomes TIMER input capture input when the select bit is set to ’1’. Setting the configuration bit-2 to 1 switches off the pull-up resistor.

5.4.4 PD3/TCMP2 - PORTD BIT 3

This pin is set as port D bit-3 I/O when the bit-3 in the PortD select register is ’0’ and is configured as TIMER output compare 2 output when the corresponding select bit is set to ’1’. Setting the configuration bit-3 to 1 makes it an open-drain output.

5.4.5 PD4/SDA0 - PORTD BIT 4

This pin is set as port D bit-4 I/O when the bit-4 in the PortD select register is ’0’ and is configured as M-BUS 0 Data pin when the corresponding select bit is set to ’1’. In this case it is always an open-drain I/O.

5.4.6 PD5/SCL0 - PORTD BIT 5

This pin is set as port D bit-5 I/O when the bit-5 in the PortD select register is ’0’ and is configured as M-BUS 0 Clock pin when the corresponding select bit is set to ’1’. In this case it is always an open-drain I/O.

5.4.7 PD6/SDA1 - PORTD BIT 6

This pin is set as port D bit-6 I/O when the bit-6 in the PortD select register is ’0’ and is configured as M-BUS 1 Data pin when the corresponding select bit is set to ’1’. In this case it is always an open-drain I/O.

5.4.8 PD7/SCL1 - PORTD BIT 7

This pin is set as port D bit-7 I/O when the bit-7 in the PortD select register is ’0’ and is configured as M-BUS 1 Clock pin when the corresponding select bit is set to ’1’. In this case it is always an open-drain I/O.

5.5 PORT E

Port E is an 8-bit bidirectional port which does not share any of its pins with other subsystems. There are three read/write registers associated to the port selecting the different functions. The register locations are shown in **Table 5-1, Port C Register Addresses**.

Data Register	PTEIO	$34
Data Direction Register	PTEDDR	$35
Configuration Register	PTECFR	$36

Table 5-4 Port E Register Addresses

All registers except the Data Register are cleared by reset thereby turning all ports to normal inputs with pull-up resistor.

The Data Direction Register along with the ConFiguration Register determines the behaviour of the I/O Port according to **Table 5-2, I/O Port Configuration Functions**.

5.6 PORT F

Port F is an 7-bit bidirectional port which does not share any of its pins with other subsystems. There are three read/write registers associated to the port selecting the different functions. The register locations are shown in **Table 5-1, Port C Register Addresses**.

Data Register	PTFIO	$38
Data Direction Register	PTFDDR	$39
Configuration Register	PTFCFR	$3A

Table 5-5 Port F Register Addresses

All registers except the Data Register are cleared by reset thereby turning all ports to normal inputs with pull-up resistor. Unused Bits read as 0.

The Data Direction Register along with the ConFiguration Register determines the behaviour of the I/O Port according to **Table 5-2, I/O Port Configuration Functions**.

5.7 PORT G

Port G is an 2-bit fixed input-only port (Bits 1-0) which can be read at any time; it reads the two analog inputs to the A/D converter, when it is enabled. Bits 7-2 read as zero.

Port G can still be read during an A/D conversion sequence, but this may inject noise on the analog inputs, resulting in reduced accuracy of the A/D. Furthermore performing a digital read of Port-G with levels other than VDD or VSS on the Port-G pins will result in greater power dissipation during the read cycles.
Since Port-G is input only there is no Data Direction Register (DDR) associated with it. The Data Register is located at $02.

5.8 INPUT/OUTPUT PROGRAMMING

Bidirectional port lines may be programmed as an input or an output under software control. The direction of the pins is determined by the state of the corresponding bit in the port data direction register (DDR). Any I/O port pin is configured as an output if its corresponding DDR bit is set to a logic one. A pin is configured as an input if its corresponding DDR bit is cleared to a logic zero.

At power-on or reset, all DDRs are cleared, which configure all port pins as inputs. The data direction registers are capable of being written to or read by the processor. During the programmed output state, a read of the data register actually reads the value of the output data latch and not the I/O pin. Refer to **Figure 5-2, Port I/O Circuitry** and **Table 5-6, I/O Pin functions**.

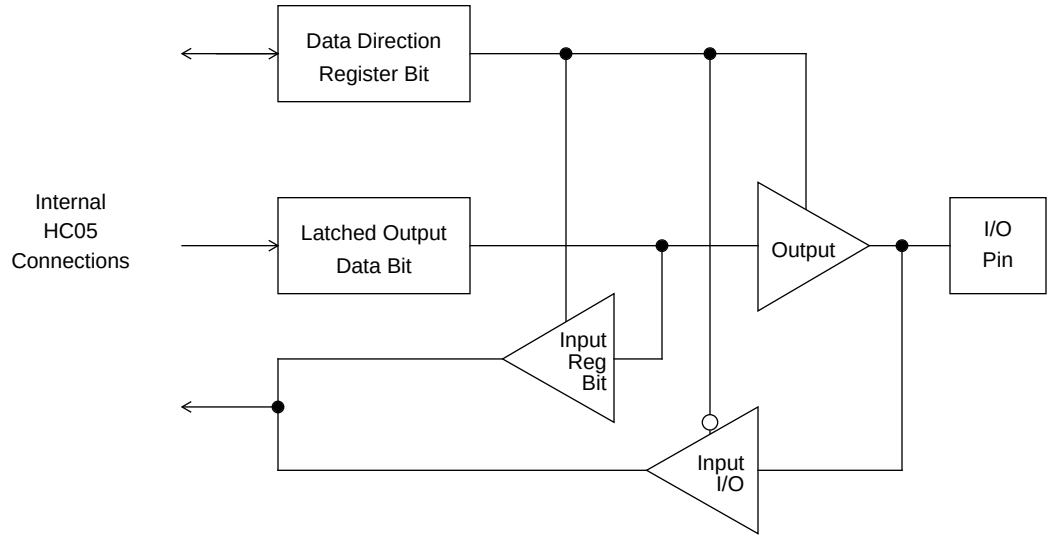


Figure 5-2 Port I/O Circuitry

R/$\overline{W}$[a]	DDR	I/O Pin Function
0	0	The I/O Pin is input. Data is written to the output data latch
0	1	Data is written into the output latch and output to the I/O Pin.
1	0	The State of the I/O Pin is read.
1	1	The I/O Pin is in output mode. The output data latch is read.

Table 5-6 I/O Pin functions

a. R/$\overline{W}$ is an internal signal.

CHAPTER 6
16 BIT PROGRAMMABLE TIMER

6.1 INTRODUCTION

The timer consists of a 16-bit, free running counter driven by a fixed divide-by-four prescaler. This timer can be used for many purposes, including input waveform measurements while simultaneously generating an output waveform. Pulse widths can vary from several microseconds to many seconds. Refer to **Figure 6-1, Timer Block Diagram**.

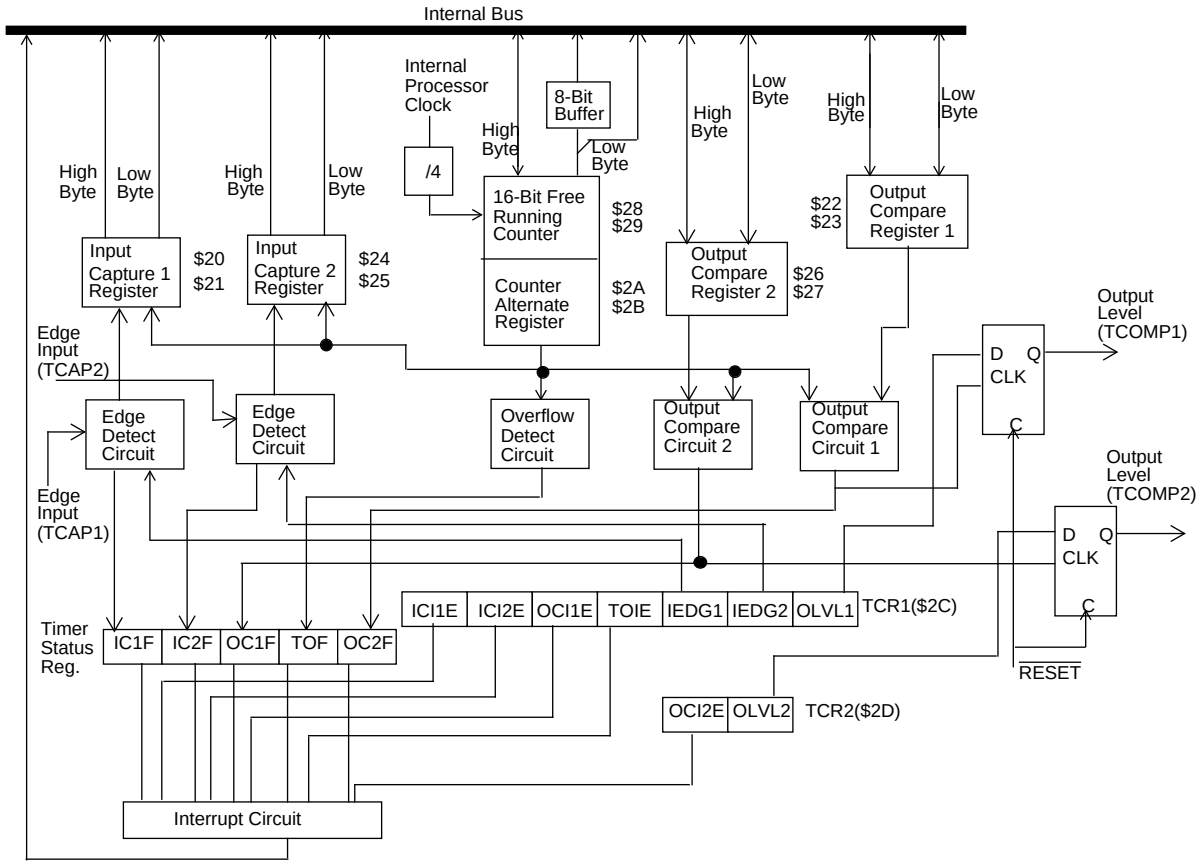


Figure 6-1 Timer Block Diagram

Because the timer has a 16-bit architecture, each specific functional segment (capability) is represented by two registers. These registers contain the high and low byte of that functional segment. Generally, accessing the low byte of a specific timer function allows full control of that function; however, an access of the high byte inhibits that specific timer function until the low byte is also accessed.

NOTE: The I bit in the CCR should be set while manipulating both the high and low byte register of a specific timer function to ensure that an interrupt does not occur.

6.1.1 TIMER REGISTER

Register Name	Address
Capture register 1 HIGH byte	$20
Capture register 1 LOW byte	$21
Compare register 1 HIGH byte	$22
Compare register 1 LOW byte	$23
Capture register 2 HIGH byte	$24
Capture register 2 LOW byte	$25
Compare register 2 HIGH byte	$26
Compare register 2 LOW byte	$27
Counter register HIGH byte	$28
Counter register LOW byte	$29
Alternate counter HIGH byte	$2A
Alternate counter LOW byte	$2B
Timer control register 1	$2C
Timer control register 2	$2D
Timer status register	$2E

Table 6-1 16-Bit Timer Register Addresses

6.2 COUNTER

The key element in the programmable timer is a 16-bit, free-running counter or counter register, preceded by a prescaler that divides the internal processor clock by four. The prescaler gives the timer a resolution of 2.0 microseconds if the internal bus clock is 2.0 MHz. The counter is incremented during the low portion of the internal bus clock. Software can read the counter at any time without affecting its value.

The double-byte, free-running counter can be read from either of two locations, $28-$29 (counter register) or $2A-$2B (counter alternate register). A read from only the least significant byte (LSB) of the free-running counter ($29, $2B) receives the count value at the time of the read. If a read of the free-running counter or counter alternate register first addresses the most significant byte ($28, $2A), the LSB ($29, $2B) is transferred to a buffer. This buffer value remains fixed after the first MSB read, even if the user reads the MSB several times. This buffer is accessed when reading the free-running counter or counter alternate register LSB ($29 or $2B) and thus completes a read sequence of the total counter value. In reading either the free-running counter or counter alternate register, if the MSB is read, the LSB must also be read to complete the sequence.

The counter alternate register differs from the counter register in one respect: a read of the counter register MSB can clear the timer overflow flag (TOF). Therefore, the counter alternate register can be read at any time without the possibility of missing timer overflow interrupts due to clearing of the TOF.

6.3 OUTPUT COMPARE REGISTERS

There are two output compare registers: Output compare register 1 and output compare register 2. Output compare registers can be used for several purposes such as controlling an output waveform or indicating when a period of time has elapsed. All bits are readable and writable and are not altered by the timer hardware or reset. If the compare function is not needed, the two bytes of the output compare register can be used as storage locations.

6.3.1 OUTPUT COMPARE REGISTER 1

The 16-bit output compare register 1 is made up of two 8-bit registers at locations $22 (MSB) and $23 (LSB). The output compare register contents are compared with the contents of the free-running counter once every four internal processor clock cycles. If a match is found, the output compare flag OC1F (bit 5 of the timer status register $2E) is set and the corresponding output level OLVL1 bit is clocked to TCMP1 pin.

The output compare register values and the output level bit should be changed after each successful comparison to establish a new elapsed timeout. An interrupt can also accompany a successful output compare provided the corresponding interrupt enable bit (OCI1E) is set.

After a processor write cycle to the output compare register 1 containing the MSB ($22), the output compare function is inhibited until the LSB ($23) is also written. The user must write both bytes (locations) if the MSB is written first. A write made only to the LSB ($23) will not inhibit the compare function. The free-running counter is updated every four internal bus clock cycles. The minimum time required to update the output compare register is a function of the program rather than the internal hardware.

The processor can write to either byte of the output compare register 1 without affecting the other byte. The output level (OLVL1) bit is clocked to the output level register regardless of whether the output compare flag (OC1F) is set or clear.

Because the output compare flag OC1F and the output compare register 1 are undetermined at power-on, and are not affected by external reset care must be exercised when initializing the output compare function. The following procedure is recommended:

- Write the high byte to the compare register 1 to inhibit further compares until the low byte is written.
- Read the Status register to arm the OC1F if it is already set.
- Write the output compare register 1 low byte to enable the output compare 1 function with the flag clear.

The purpose of this procedure is to prevent the OC1F bit from being set between the time it is read and the write to the corresponding output compare register

6.3.2 OUTPUT COMPARE REGISTER 2

The 16-bit output compare register 2 is made up of two 8-bit registers at locations $26 (MSB) and $27 (LSB). The output compare register contents are compared with the contents of the free-running counter once every four internal processor clock cycles. If a match is found, the output compare flag OC2F (bit 1 of the timer status register $2E) is set and the corresponding output level OLVL2 bit (bit 0 of control register 2,$2D) is clocked to TCMP2 pin.

The output compare register values and the output level bit should be changed after each successful comparison to establish a new elapsed timeout. An interrupt can also accompany a successful output compare provided the corresponding interrupt enable bit (OCI2E, bit 5 of timer control register 2 at $2D) is set.

After a processor write cycle to the output compare register 2 containing the MSB ($26), the output compare function is inhibited until the LSB ($27) is also written. The user must write both bytes (locations) if the MSB is written first. A write made only to the LSB ($26) will not inhibit the compare function. The free-running counter is updated every four internal bus clock cycles. The minimum time required to update the output compare register is a function of the program rather than the internal hardware.

The processor can write to either byte of the output compare register 2 without affecting the other byte. The output level (OLVL2) bit is clocked to the output level register regardless of whether the output compare flag (OC2F) is set or clear.

Because the output compare flag OC2F and the output compare register 2 are undetermined at power-on, and are not affected by external reset care must be exercised when initializing the output compare function. A procedure as recommended for compare register 1 should be followed.

NOTE: As the TCMP1 and TCMP2 pins are shared with PORT D, the output compare 1 & 2 can not be used for compare, when the pins are selected to be input, but the register can still be used as a temporary store.

6.4 INPUT CAPTURE REGISTERS

There are two identical input capture registers: Input capture register 1 and input capture register 2. The two following sections describe these two registers.

6.4.1 INPUT CAPTURE REGISTER 1

Two 8-bit registers, which make up the 16-bit input capture register 1, are read-only and are used to latch the value of the free-running counter after the corresponding input capture edge detector senses a defined transition on the TCAP1 pin. The level transition which triggers the counter transfer is defined by the corresponding input edge bit (IEDG1). Reset does not affect the contents of the input capture register except when exiting stop mode.

IEDG1 = 0: CAPTURE ON NEGATIVE EDGE
IEDG1 = 1: CAPTURE ON POSITIVE EDGE

An interrupt can also accompany a capture provided the corresponding interrupt enable bit, ICI1E (bit 7 of the timer control register 1 $2C) is set.

The result obtained by an input capture will be one more than the value of the free-running counter on the rising edge of the internal bus clock preceding the external transition. This delay is required for internal synchronization. Resolution is one count of the free-running counter, which is four internal bus clock cycles.

The free-running counter contents are transferred to the input capture register on each proper signal transition regardless of whether the input capture flag (ICF1F) is set or clear. The input capture register always contains the free-running counter value that corresponds to the most recent input capture.

After a read of the input capture register most significant byte($20), the counter transfer is inhibited until the least significant byte ($21) is also read. This characteristic causes the time used in the input capture software routine and its interaction with the main program to determine the minimum pulse period.

A read of the input capture register LSB ($21) does not inhibit the free-running counter transfer since they occur on opposite edges of the internal bus clock.

6.4.2 INPUT CAPTURE REGISTER 2

Input capture register B is identical to Input capture Register 2. Two 8-bit registers, which make up the 16-bit input capture register 2, are read-only and are used to latch the value of the free-running counter after the corresponding input capture edge detector senses a defined transition on the TCAP2 pin. The level transition which triggers the counter transfer is defined by the corresponding input edge bit (IEDG2). Reset does not affect the contents of the input capture register except when exiting stop mode.

IEDG2 = 0: CAPTURE ON NEGATIVE EDGE
IEDG2 = 1: CAPTURE ON POSITIVE EDGE

An interrupt can also accompany a capture provided the corresponding interrupt enable bit, ICI2E (bit 6 of the timer control register 1 $2D) is set

The result obtained by an input capture will be one more than the value of the free-running counter on the rising edge of the internal bus clock preceding the external transition. This delay is required for internal synchronization. Resolution is one count of the free-running counter, which is four internal bus clock cycles.

The free-running counter contents are transferred to the input capture register on each proper signal transition regardless of whether the input capture flag (ICF2F) is set or clear. The input capture register always contains the free-running counter value that corresponds to the most recent input capture.

After a read of the input capture register most significant byte($24), the counter transfer is inhibited until the least significant byte ($25) is also read. This characteristic causes the time used in the input capture software routine and its interaction with the main program to determine the minimum pulse period.

A read of the input capture register LSB ($25) does not inhibit the free-running counter transfer since they occur on opposite edges of the internal bus clock.

6.5 TIMER CONTROL REGISTERS (TCR1 & TCR2) $2C,$2D

The two timer control registers TCR1($2C) and TCR2($2D) are read/write registers. Five bits control interrupts associated with the timer status register flags IC1F, IC2F,OC1F,OC2F and TOF, two bits control which edge is significant to the input capture 1 and 2 edge detectors. The control registers are illustrated below followed by a definition of each bit.

TCR1 $2C	IC1IE	IC2IE	OC1IE	TOIE	CO1E	IEDG1	IEDG2	OLVL1
RESET:	0	0	0	0	0	U	U	U
TCR2 $2D	0	0	OC2IE	0	CO2E	0	0	OLVL2
RESET:	0	0	0	0	0	0	0	U

Figure 6-2 Timer Control Registers

IC1IE - Input Capture 1 Interrupt Enable
1 = Interrupt enabled
0 = Interrupt disabled

IC2IE - Input Capture 2 Interrupt Enable
1 = Interrupt enabled
0 = Interrupt disabled

OC1IE - Output Compare 1 Interrupt Enable
1 = Interrupt enabled
0 = Interrupt disabled

TOIE - Timer Overflow Interrupt Enable
1 = Interrupt enabled
0 = Interrupt disabled

CO1E - Timer Compare 1 Output Enable
1 = Output of Timer Compare 1 is enabled
0 = Output of Timer Compare 1 is disabled, i.e. held low
Reset clears this bit.

IEDG1 - Input Edge
Value of input edge determines which level transition on TCAP1 pins will trigger the transfer of the free-running counter to the input capture register 1.
1 = Positive edge
0 = Negative edge
Reset does not affect the IEDG1 bit (U=unaffected).

IEDG2 - Input Edge
Value of input edge determines which level transition on TCAP2 pins will trigger the transfer of the free-running counter to the input capture register 2.
1 = Positive edge
0 = Negative edge
Reset does not affect the IEDG2 bit (U=unaffected).

OLVL1 - Output Level
Value of output level is clocked into output level register by the next successful output compare 1,and will appear on the TCOMP1 pins.
1 = High output
0 = Low output

OCI2E - Output Compare 2 Interrupt Enable
1 = Interrupt enabled
0 = Interrupt disabled

CO2E - Timer Compare 2 Output Enable
1 = Output of Timer Compare 2 is enabled
0 = Output of Timer Compare 2 is disabled, i.e. held low
Reset clears this bit.

OLVL2 - Output Level
Value of output level is clocked into output level register by the next successful output compare B, and will appear on the TCOMP2 pins.
1 = High output
0 = Low output

Bits 1,2,4,6,AND 7 OF TRC2 - Not used
Always read zero

6.6 TIMER STATUS REGISTER (TSR) $2E

The TSR is a read-only register containing timer status flag bits.

$2E	IC1F	IC2F	OC1F	TOF	TCAP1	TCAP2	OC2F	0
RESET:	U	U	U	0	0	0	U	0

Figure 6-3 Timer Status Register

IC1F - Input Capture 1 Flag
1 = Flag set when selected polarity edge is sensed by input capture 1 edge detector.
0 = Flag cleared when TSR and input capture 1 registers LOW byte is accessed.

IC2F - Input Capture 2 Flag
1 = Flag set when selected polarity edge is sensed by input capture 2 edge detector.
0 = Flag cleared when TSR and input capture 2 registers LOW byte is accessed.

OC1F - Output Compare 1 Flag
1 = Flag set when output compare register 1 contents match the free-running counter contents.
0 = Flag cleared when TSR and output compare register 1 low byte are accessed.

TOF - Timer Overflow Flags
1 = Flag set when free-running counter transition from $FFFF to $0000 occurs.
0 = Flag cleared when TSR and counter low register are accessed
Bits 0,2, and 3 - Not used.
Always read zero.

TCAP1 - Timer Capture 1
This bit reflects the current state of the Timer Capture 1 Input.

TCAP2 - Timer Capture 2
This bit reflects the current state of the Timer Capture2 Input.

OC2F - Output Compare 2 Flag

1 = Flag set when output compare register 2 contents match the free-running counter contents.

0 = Flag cleared when TSR and output compare register 2 low byte are accessed.

Accessing the timer status registers satisfies the first condition required to clear status bits. The remaining step is to access the registers corresponding to the status bit.

A problem can occur when using the timer overflow function and reading the free-running counter at random times to measure an elapsed time. Without incorporating the proper precautions into software, the timer overflow flag could unintentionally be cleared if:

1) The timer status register is read or written when TOF is set, and
2) The LSB of the free-running counter is read but not for the purpose of servicing the flag.

The counter alternate register at address $2A and $2B contains the same value as the free-running counter (at address $28 and $29); therefore, this alternate register can be read at any time without affecting the timer overflow flag in the timer status register.

6.7 TIMER DURING WAIT MODE

The CPU clock halts during the WAIT mode, but the timer keeps on running.

6.8 TIMER DURING STOP MODE

In the STOP mode, the timer stops counting and holds the last count value if STOP is exited by an interrupt. If $\overline{\text{RESET}}$ is used, the counters are forced to $FFFC. During STOP, if at least one valid input capture edge occurs at the TCAP pins, the input capture detect circuit is armed. This does not set any timer flags nor wake up the MCU, but when the MCU does wake up, there is an active input capture flag and data from the first valid edge that occurred during the STOP mode. If $\overline{\text{RESET}}$ is used to exit STOP mode, then no input capture flag or data remains, even if a valid input capture edge occurred.

CHAPTER 7
MULTI-MASTER M-BUS

7.1 INTRODUCTION

Motorola bus (in short: M-Bus) is a two wire, bidirectional serial bus which provides a simple, efficient way for data exchange between devices. It is fully compatible to I^2C bus standard. This two wire bus minimizes the interconnection between the devices and eliminates the need of address decoder, so that less PCB traces and economic hardware structure resulted. In this implementation two pairs of I^2C bus wires are supported by multiplexing logic.

This bus is suitable of the applications which need frequent, occasional communications in a short distance among a number of devices. It also provides a flexibility that allows additional devices to be “hanged” on it in further expansion for system developing.The maximum data rate is limited on 100Kbit/s, the maximum communication length and number of devices that can be connected are limited by a maximum bus capacitance of 400 pF.

M-Bus system is a true multi-master bus including collision detection and arbitration to prevent data corruption if two or more masters intend to control the bus simultaneously. This feature provides the capability for complex applications with multi-processor control. It may also be used for rapid testing and alignment of end products via external connections to an assembly-line computer.

7.2 M-BUS INTERFACE FEATURES

- Fully compatible to I^2C Bus standard
- Multi-Master operation
- Software programmable for one of 32 different serial clock frequencies
- Software selectable acknowledge bit
- Interrupt driven byte by byte data transfer
- Arbitration lost driven interrupt with automatic mode switching from master to slave
- Calling address identification interrupt
- Generate/detect the start or stop signal
- Repeated START signal generation
- Generate/recognize the acknowledge bit
- Bus busy detection

7.3 M-BUS SYSTEM CONFIGURATION

M-Bus system uses a serial data line (SDA) and a serial clock line (SCL) for data transfer. All devices connected on it must have open drain or open collector output, logic “and” function is exercised in both lines with two pulling up resistors.

7.4 M-BUS PROTOCOL

Normally a standard communication is composed of four parts, START signal, Slave address transmission, Data transfer and STOP signal. They are described briefly in the following sections and illustrated in **Figure 7-1, M-BUS Transmission Signal Diagram**.

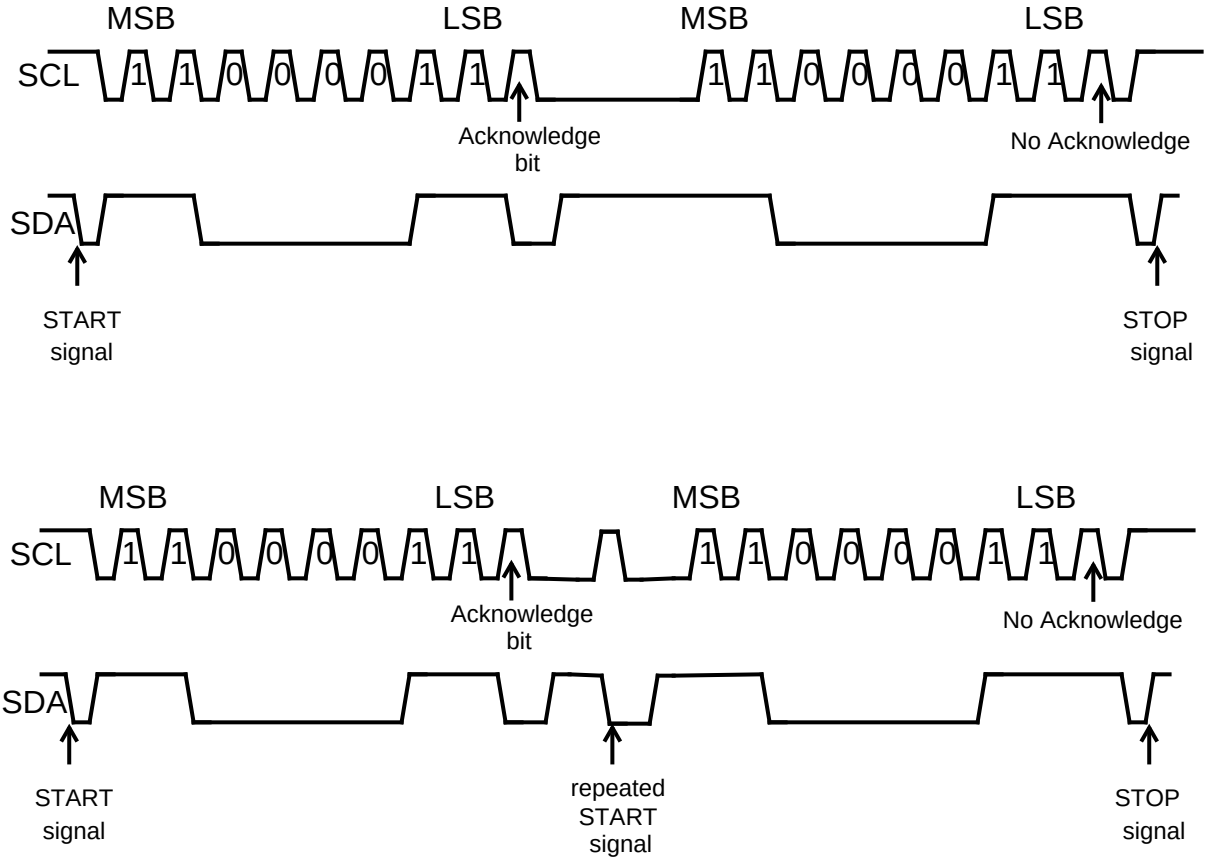


Figure 7-1 M-BUS Transmission Signal Diagram

7.4.1 START SIGNAL

When the bus is free, i.e., no master device is engaging the bus (both SCL and SDA lines are at logical high, a master may initiate communication by sending a START signal. As shown in **Figure 7-1, M-BUS Transmission Signal Diagram**, a START signal is defined as a high to low transition of SDA while SCL is high. This signal denotes the beginning of a new data transfer (each data transfer may contain several bytes of data) and wake up all slaves.

7.4.2 SLAVE ADDRESS TRANSMISSION

The first byte of data transfer immediately after the START signal is the slave address transmitted by the Master. This is a seven bits long calling address followed by a R/W-bit. The R/W-bit tells the slave the desired direction of data transfer.

Only the slave with matched address will respond by sending back an acknowledge bit which is signalled from the receiving device by pulling the SDA low at the 9th clock (see **Figure 7-1, M-BUS Transmission Signal Diagram**.)

7.4.3 DATA TRANSFER

Once a successful slave addressing is achieved, the data transfer can proceed byte by byte in a direction specified by the R/W-bit sent by the calling master.

Each data byte is 8 bit long. Data can be changed only during SCL is low and must be held stable during SCL is high as shown in **Figure 7-1, M-BUS Transmission Signal Diagram**. One clock pulse is for one bit data transferring, MSB is transferred first. Each byte data has be followed by an acknowledge bit, which is signalled from the receiving device by pulling the SDA low at the 9th clock. So one complete data byte transferring needs 9 clock pulses.

If the slave receiver does not acknowledge the master, the SDA line should be left high by the slave, the master can then generate a stop signal to abort the data transfer or a start signal (repeated start) to commence a new calling.

If the master receiver does not acknowledge the slave transmitter after one byte transmission, it means an 'end of data' to the slave, so that the slave shall release the SDA line for the master to generate 'stop' or 'start' signal.

7.4.4 REPEATED START SIGNAL

As shown in **Figure 7-1, M-BUS Transmission Signal Diagram**, a repeated START signal is to generate a START signal without first generating a STOP signal to terminate the communication. This is used by the master to communicate with another slave or with the same slave in different mode (transmit/receive mode) without releasing the bus.

7.4.5 STOP SIGNAL

The master can terminate the communication by generating a STOP signal to free the bus. However, the master may generate a START signal followed by a calling command without generating a STOP signal first. This is called repeat start. A STOP signal is defined as a low to high transition of SDA while SCL at logical high. (**Figure 7-1, M-BUS Transmission Signal Diagram**).

7.4.6 ARBITRATION PROCEDURE

This interface circuit is a true multi-master system which allows more than one master to be connected on it. If two or more masters try to control the bus at the same time, a clock synchronization procedure determines the bus clock, for which the low period is equal to the longest clock low period and the high is equal to the shortest one among the masters. A data arbitration procedure determines the priority, the masters will lose arbitration if they transmit logic “1” while the other transmit logic “0” the losing masters will immediately switch over to slave receive mode and stop its data and clock outputs. The transition from master to slave mode will not generate a stop condition in this case. Meanwhile a software bit will be set by hardware to indicate loss of arbitration.

7.4.7 CLOCK SYNCHRONIZATION

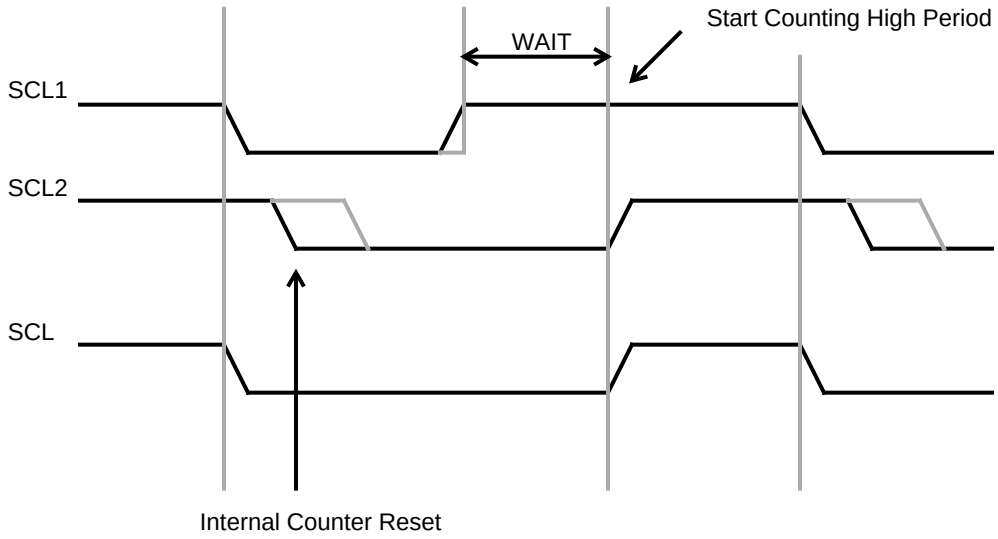


Figure 7-2 Clock Synchronization

Since wired-AND logic is performed on SCL line, a high to low transition on SCL line will affect the devices connected on the bus. The devices start counting their low period and once a device's clock has gone low, it will hold the SCL line low until the clock high state is reached. However, the change of low to high in this device clock may not change the state of the SCL line, if another device clock is still within its low period. Therefore synchronized clock SCL will be held low by the device with the longest low period. Devices with shorter low periods enter a high wait state during this time (see **Figure 7-2, Clock Synchronization**). When all devices concerned have counted off their low period, the synchronized clock SCL line will be released and go high. There will then be no difference between the device clocks and the state of the SCL line and all of them will start counting their high periods. The first device to complete its high period will again pull the SCL line low.

7.4.8 HANDSHAKING

The clock synchronization mechanism can be used as a handshake in data transfer. Slave device may hold the SCL low after completion of one byte transfer (9 bits). In such case, it will halt the bus clock and force the master clock in a wait state until the slave release the SCL line.

7.5 REGISTERS

There are five different registers used in the M-Bus interface and the internal configuration of these registers is discussed in the following paragraphs.

7.5.1 M-BUS ADDRESS REGISTER (MADR)

Bit	7	6	5	4	3	2	1	0
$10	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	0
RESET	0	0	0	0	0	0	0	0

Figure 7-3 M-BUS Address Register (MADR)

Bit 1 to bit 7 contain its own specific slave address. This register is cleared upon reset.

7.5.2 M-BUS FREQUENCY DIVIDER REGISTER (FDR)

Bit	7	6	5	4	3	2	1	0
$11	-	-	-	MBC4	MBC3	MBC2	MBC1	MBC0
RESET	-	-	-	0	0	0	0	0

Figure 7-4 M-BUS Frequency Divider Register (FDR)

Bit 0 to bit 4 are used for clock rate selection. The serial bit clock frequency is equal to the CPU clock divided by the divider shown in the following **Table 7-1, M-BUS Prescaler**. This register is cleared upon reset.

For a 4 MHz external crystal operation (2 MHz internal operating frequency), the serial bit clock frequency of M-Bus ranges from 460 Hz to 90909 Hz.

MCB4-MCB0	DIVIDER	MCB4-MCB0	DIVIDER
00000	22	10000	352
00001	24	10001	384
00010	28	10010	448
00011	34	10011	544
00100	44	10100	704
00101	48	10101	768
00110	56	10110	896
00111	68	10111	1088
01000	88	11000	1408
01001	96	11001	1536
01010	112	11010	1792
01011	136	11011	2176
01100	176	11100	2816
01101	192	11101	3072
01110	224	11110	3584
01111	272	11111	4352

Table 7-1 M-BUS Prescaler

7.5.3 M-BUS CONTROL REGISTER (MCR)

Bit	7	6	5	4	3	2	1	0
$12	MEN	MIEN	MSTA	MTX	TXAK	MMUX	0	0
RESET	0	0	0	0	0	0	0	0

Figure 7-5 M-BUS Control Register (MCR)

The MCR provides six control bits. MCR is cleared upon reset.

7.5.3.1 MEN

If the M-Bus enable bit (MEN) is set, the M-Bus interface system is enabled. If MEN is cleared, the interface is reset and disabled.The MEN bit must be set first before any bits of MCR being set.

7.5.3.2 MIEN

If the M-Bus interrupt enable bit (MIEN) is set, the M-Bus interrupt occurs provided the MIF in the status register is set. If MIEN is cleared, the M-Bus interrupt is disabled.

7.5.3.3 MSTA

Master/slave mode select bit:

1 = Master
0 = Bit

Upon reset, this bit is cleared. When this bit is changed from 0 to 1, a START signal is generated on the bus and the master mode is selected. When this bit is changed from 1 to 0, a STOP signal is generated and the operation mode changes from master to slave. In master mode, a bit clear immediately followed by a bit set of this bit generates a repeated START signal (See **Figure 7-1, M-BUS Transmission Signal Diagram**) without generating a STOP signal.

7.5.3.4 MTX

Transmit/Receive mode select bit:

1 = Transmit
0 = Receive

7.5.3.5 TXAK

If the transmit acknowledge enable bit (TXAK) is cleared, an acknowledge signal will be sent out to the bus at the 9th clock bit after receiving one byte data. When TXAK is set, no acknowledge signal response (i.e. acknowledge bit = 1).

7.5.3.6 MMUX

Multiplex bit:

0 = Select SDA0,SCL0
1 = Select SDA1,SCL1.

7.5.4 M-BUS STATUS REGISTER (MSR)

This status register is software readable only with exception of bit 1 (MIF) and bit 4 (MAL), which are software clearable. All bits are cleared upon reset except bit 7 (MCF) and bit 0 (RXAK).

Bit	7	6	5	4	3	2	1	0
$13	MCF	MAAS	MBB	MAL	0	SRW	MIF	RXAK
RESET	1	0	0	0	0	0	0	1

Figure 7-6 M-BUS Status Register (MSR)

7.5.4.1 MCF

Data transferring bit. When one byte of data is being transferred, this bit is cleared.

7.5.4.2 MAAS

Addressed as a slave bit

1 = Addressed as a slave
0 = Not addressed

When its own specific address (M-Bus Address Register) is matched with the calling address, this bit is set. The CPU is interrupted provided the MIEN is set. Then CPU needs to check the SRW bit and set its TX/RX mode accordingly.Writing to the M-Bus Control Register clears this bit.

7.5.4.3 MBB

Bus busy bit

1 = Bus busy
0 = Bus idle

This bit indicates the status of the bus. When a start signal is detected, the MBB is set. If a STOP signal is detected, it is cleared.

7.5.4.4 MAL

The arbitration lost bit (MAL) is set by hardware when the arbitration procedure is lost during a master transmission mode. This bit must be cleared by software.

7.5.4.5 SRW

When MAAS is set, the R/W- command bit of the calling address sent from master is latched into the R/W-command bit (SRW). Checking this bit, the CPU can select slave transmit/receive mode according to the command of master.

7.5.4.6 MIF

The M-Bus interrupt bit (MIF) is set when there is a pending interrupt, which will cause a M-Bus interrupt request provided MIEN is set. This bit is set when one of the following event occurs:

1. Complete one byte of data transfer, it is set at the falling edge of the 9th clock.
2. Receive a calling address which matches its own specific address in slave receive mode.
3. Arbitration lost.

This bit must be cleared by software in the interrupt routine.

7.5.4.7 RXAK

If the received acknowledge bit (RXAK) is low, it indicates an acknowledge signal has been received after the completion of 8 bits data transmission on the bus. If RXAK is high, it means no acknowledge signal is detected at the 9th clock. It is set upon reset.

7.5.5 M-BUS DATA I/O REGISTER (MDR)

Bit	7	6	5	4	3	2	1	0
$14	TRXD7	TRXD6	TRXD5	TRXD4	TRXD3	TRXD2	TRXD1	TRXD0
RESET	-	-	-	-	-	-	-	-

Figure 7-7 M-BUS Data I/O Register (MDR)

In master transmit mode, data writing into this register is sent to the bus automatically, most significant bit is sent first. In master receive mode, reading from this register initiates next byte data receiving. In slave mode, the same function is available after it is addressed.

7.6 PROGRAMMING CONSIDERATION

7.6.1 INITIALIZATION

Reset will put the M-Bus Control Register to its default status. Before the interface can be used to transfer serial data, an initialization procedure must be carried out.

1. Update Frequency Divider Register (FDR) to select a SCL frequency.
2. Update M-Bus Address Register (MADR) to define its own slave address.
3. Set MEN bit of M-Bus Control Register (MCR) to enable the M-Bus interface system.
4. Modify the bits of M-Bus Control Register (MCR) to select Master/Slave mode, Transmit/Receive mode, interrupt enable or not.

7.6.2 GENERATION OF A START SIGNAL AND THE FIRST BYTE OF DATA TRANSFER

After completion of the initialization procedure, serial data can be transmitted by selecting the 'master transmitter' mode. If the device is connected to a multi-master bus system, the state of the M-Bus busy bit (MBB) must be tested to check whether the serial bus is free. If the bus is free (MBB = 0), the start condition and the first byte (the slave address) can be sent. An example of a program which generates the START signal and transmits the first byte of data (slave address) is shown below:

```
                SEI                         ; DISABLE INTERRUPT
CHFLAG          BRSET     5,MSR,CHFLAG      ; CHECK THE MBB BIT OF THE
                                            ; STATUS REGISTER.
                                            ; IF IT IS SET, WAIT UNTIL IT IS
                                            ; CLEAR.
TXSTART         BSET       4,MCR            ; SET TRANSMIT MODE
                BSET      5,MCR             ; SET MASTER MODE
                                            ; i.e. GENERATE START
                                            ; CONDITION
                LDA       CALLING           ; GET THE CALLING ADDRESS
                STA       MDR               ; TRANSMIT THE CALLING
                                            ; ADDRESS
                CLI                         ; ENABLE INTERRUPT
```

7.6.3 SOFTWARE RESPONSES AFTER TRANSMISSION OR RECEPTION OF A BYTE

Transmission or reception of a byte will set the data transferring bit (MCF) to 1, which indicates one byte communication is finished. Also the M-Bus interrupt bit (MIF) is set to generate an M-Bus interrupt if the interrupt function is enable in initialization. Software must clear the MIF bit in the interrupt routine first. The MCF bit will be cleared by reading from the M-BUS DATA I/O Register (MDR) in receive mode or writing to MDR in transmit mode. Software may serve the M-Bus I/O in the main program by monitoring the MIF bit if the interrupt function is disabled. The following is an example of a software response by a 'master transmitter' in the interrupt routine.

```
ISR         BCLR    1,MSR           ; CLEAR THE MIF LFAG
            BRCLR   5, MCR,SLAVE    ; CHECK THE MSTA FLAG,
                                    ; BRANCH IF SLAVE MODE
            BRCLR   4, MCR,RECEIVE  ; CHECK THE MODE FLAG
            BRSET   0,MSR,END       ;CHECK ACKNOWLEDGEMENT
                                    ; FROM RECEIVER,
                                    ; IF NO ACKNOWL'MENT, END
                                    ; OF TRANSMISSION
TRANSMIT    LDA     DATABUF         ; GET THE NEXT BYTE OF DATA
```

7.6.4 GENERATION OF THE STOP SIGNAL

A data transfer ends with a STOP signal generated by the 'master' ' device. A master transmitter can simply generate a STOP signal after all the data has been transmitted. The following is an example showing how a stop condition is generated by a master transmitter:

```
MASTXBRSET0,MSR,END; IF NO ACKNOWLEDGEMENT,
                                    ; BRANCH TO END,
            LDA     TXCNT           ; GET VALUE FROM THE
                                    ; TRANSMITTING COUNTER
            BEQ     END             ; IF NO MORE DATA, BRANCH
                                    ; TO END
            LDA     DATABUF         ; GET NEXT BYTE OF DATA
            STA     MDR             ; TRANSMIT THE DATA
            DEC     TXCNT           ; DECREASE THE TXCNT
            BRA     EMASTX          ; EXIT
END         BCLR    5,MCR           ; GENERATE A STOP
                                    ; CONDITION
EMASTX      RTI                     ; RETURN FROM INTERRUPT
```

If a master receiver wants to terminate a data transfer, it must inform the slave transmitter by not acknowledging the last byte of data which can be done by setting the transmit acknowledge bit (TXAK) before reading the 2nd last byte of data. Before reading the last byte of data, a STOP signal must be generated first. The following is an example showing how a STOP signal is generated by a master receiver.

```
MASR        DEC     RXCNT
            BEQ     ENMASR      ; LAST BYTE TO BE READ
            LDA     RXCNT
            DECA                ; CHECK LAST SECOND BYTE
                                ; TO BE READ
            BNE     NXMAR       ; NOT LAST ONE OR LAST
                                ; SECOND
LAMAR       BSET    3,MCR       ; LAST SECOND, DISABLE
                                ; ACKNOWLEDGEMENT
                                ; TRANSMITTING
            BRA     NXMAR       ; NXMAR
ENMASR      BCLR    5,MCR       ; LAST ONE, GENERATE 'STOP'
                                ; SIGNAL
NXMAR       LDA     MDR         ; READ DATA AND STORE
            STA     RXBUF
            RTI
```

7.6.5 GENERATION OF A REPEATED START SIGNAL

At the end of data transfer, if the master still want to communicate on the bus, it can generate another START signal followed by another slave address without first generate a STOP signal. A program example is as shown.

```
RESTART     BCLR    5,MCR       ; ANOTHER START (RESTART)
                                ; IS GENERATED BY
            BSET    5,MCR       ; THESE TWO CONSEQUENCE
                                ; INSTRUCTION
            LDA     CALLING     ; GET THE CALLING ADDRESS
            STA     MDR         ; TRANSMIT THE CALLING
                                ; ADDRESS
```

7.6.6 SLAVE MODE

In slave service routine, the master addressed as slave bit (MAAS) should be tested for checking if a calling of its own address has just received. If MAAS is set, software should set the transmit/receive mode select bit (MTX BIT OF MCR) according to the R/W-command bit (SRW). Writing to the MCR clears the MAAS automatically. Then a data transfer may be initiated by writing information to MDR or dummy reading from MDR.

In slave transmitter routine, the received acknowledge bit (RXAK) must be tested before transmitting next byte of data. RXAK is set means 'end of data' signal from the master receiver, then it must switch from transmitter mode to receiver mode by software and a dummy read followed to release the SCL line so that the master can generate a stop signal.

7.6.7 ARBITRATION LOST

If more than one master want to engage the bus simultaneously, only one master wins and the others lost arbitration. The arbitration lost devices immediately switch to slave receive mode by hardware. Their data output to the SDA line is stopped but internal transmitting clock still run until the end of the byte transmitting. An interrupt occurs when this dummy 'byte' transmitting is accomplished with MAL=1 and MSTA=0. If one master attempt to start transmission while the bus is being engaged by another master, the hardware will inhibit the transmission; switch the MSTA bit from 1 to 0 without generating STOP conditions; generate an interrupt to CPU and set the MAL to indicate that the attempt to engage the bus is failed. Consideration of these cases, the slave service routine should test the MAL first, software should clear the MAL bit if it is set.

NOTE: **Figure 7-8, Flowchart of MBUS Interrupt Routine** shows an example flow of the MBUS-Interrupt routine.

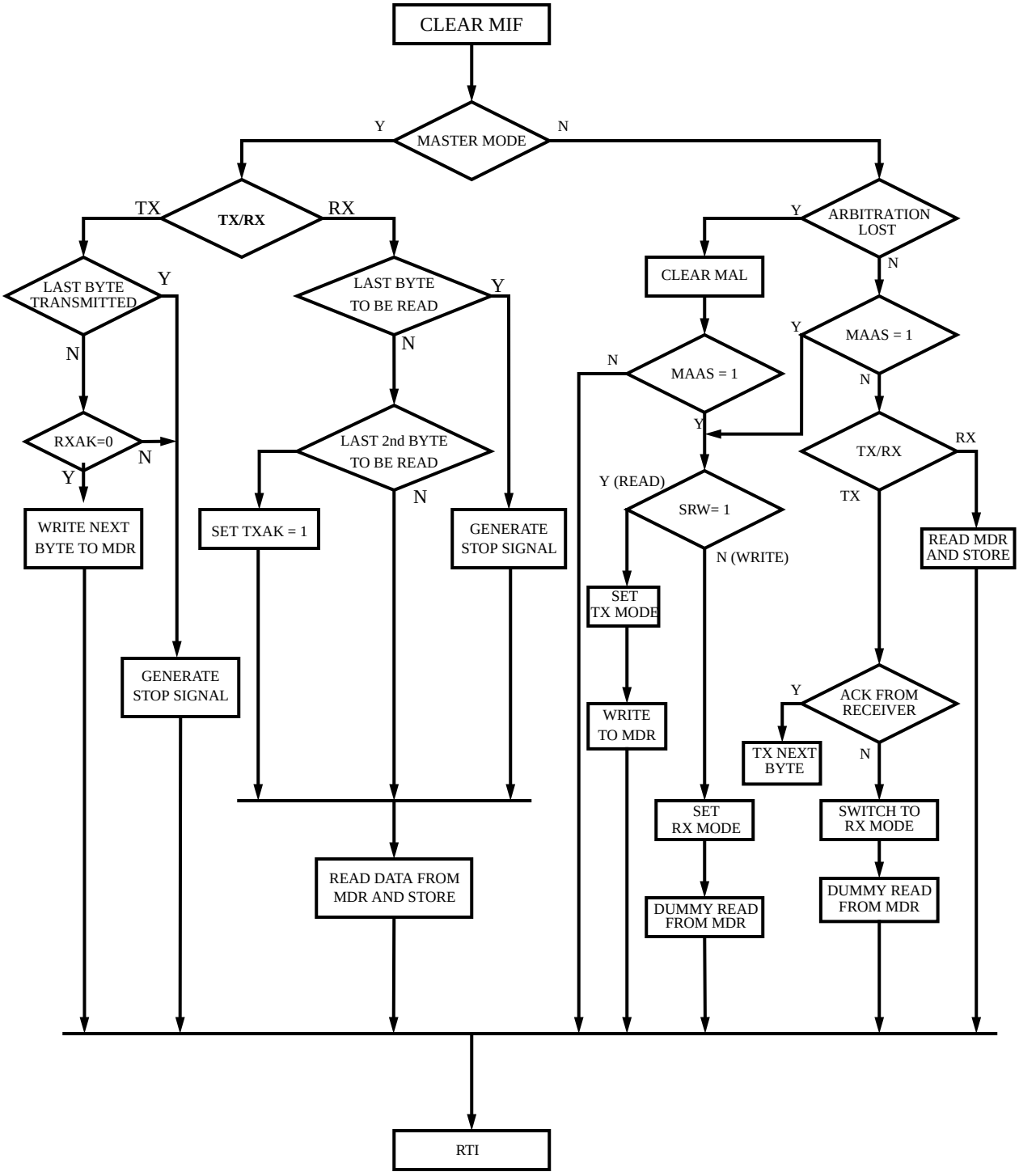


Figure 7-8 Flowchart of MBUS Interrupt Routine

7.7 OPERATION DURING WAIT MODE

During WAIT mode the MBUS block is idle, but wakes up on receiving a valid start condition, if in slave mode. If the interrupt is enabled the CPU will come out of WAIT mode after the end of a byte transmission.

7.8 OPERATION DURING STOP MODE

In Stop Mode the whole block is disabled.

CHAPTER 8
A/D CONVERTER

The 68HC705E24/68HC05E16 includes a 2-channel, 8-bit, multiplexed input, successive approximation A/D converter, with two of the inputs available on external pins and four additional internal channels for test purposes.

8.1 ANALOG SECTION

8.1.1 RATIOMETRIC CONVERSION

The A/D is ratiometric, with two dedicated pins supplying the reference voltages (V_{REFH} and V_{REFL}). An input voltage equal to V_{REFH} converts to $FF (full scale) and an input voltage equal to V_{REFL} converts to $00. An input voltage greater than V_{REFH} will convert to $FF with no overflow indication. For ratiometric conversions, the source of each analog input should use V_{REFH} as the supply voltage and be referenced to V_{REFL}.

8.1.2 V_{REFH} AND V_{REFL}

The reference supply for the converter uses two dedicated pins rather than being driven by the system power supply lines because the voltage drops in the bonding wires of those heavily loaded pins would degrade the accuracy of the A/D conversion. V_{REFH} and V_{REFL} can be any voltage between V_{SS} and V_{DD}, as long as $V_{REFH} > V_{REFL}$. However, the accuracy of conversions is tested and guaranteed only for V_{REFL} = VSS and V_{REFH} = VDD.

8.1.3 ACCURACY

The 8-bit conversions are accurate to within +/- 1.5 LSB including quantization.

8.2 CONVERSION PROCESS

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 and has no missing codes.

8.3 DIGITAL SECTION

8.3.1 CONVERSION TIMES

Each channel of conversion takes 32 clock cycles, which must be at a frequency equal to or greater than 1 MHz.

A multiplexer allows the single A/D converter to select one of six analog signals.Two pins of port G are input signals to the multiplexer.

8.3.2 INTERNAL VS. MASTER OSCILLATOR

If the MCU bus (E clock) frequency is less than 1.0 MHz, an internal RC oscillator (nominally 1.5 MHz) must be used for the A/D conversion clock. This selection is made by setting the ADRC bit in the A/D Status and Control Register to 1.

When the internal RC oscillator is being used as the conversion clock three limitations apply:

- The conversion complete flag (COCO) must be used to determine when a conversion sequence has been completed, due to the frequency tolerance of the RC oscillator and its asynchronism with regard to the MCU E clock.
- The conversion process runs at the nominal 1.5 MHz rate but 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 cycle.
- If the system clock is running faster than the RC oscillator, the RC oscillator should be turned off, and the system clock used as the conversion clock.

8.3.3 MULTI-CHANNEL OPERATION

A multiplexer allows the single A/D converter to select one of two analog signals. The two pins of Port G are input signals to the multiplexer.

8.4 A/D STATUS AND CONTROL REGISTER (ADSCR) $16

The following paragraphs describe the function of the A/D Status and Control Register.

Bit	7	6	5	4	3	2	1	0
$16	COCO	ADRC	ADON	0	0	CH2	CH1	CH0
RESET	0	-	0	0	0	-	-	-

Figure 8-1 A/D Status and Control Register

8.4.1 COCO - CONVERSIONS COMPLETE

This read-only status bit is set when a conversion is completed, indicating that the A/D Data Register contains valid results. This bit is cleared whenever the A/D Status and Control Register is written and a new conversion automatically started, or whenever the A/D Register is read. Once a conversion has been started by writing to the A/D Status and Control Register, conversions of the selected channel will continue every 32 cycles until the A/D Status and Control Register is written again. In this continuous conversion mode the A/D Data Register will be filled with new data, and the COCO bit set, every 32 cycles. Data from the previous conversion will be overwritten regardless of the state of the COCO bit prior to writing.

8.4.2 ADRC-RC OSCILLATOR ON

When ADRC is set, the A/D section runs on the internal RC oscillator instead of the CPU clock. The RC oscillator requires a time t_{RCON} to stabilize, and results can be inaccurate during this time. See **Section 8.3.2, INTERNAL VS. MASTER OSCILLATOR**.

8.4.3 ADON - A/D ON

When the A/D is turned on (ADON = 1), it requires a time t_{ADON} for the current sources to stabilize, and results can be inaccurate during this time. This bit turns on the charge pump.

8.4.4 CH2:0 - CHANNEL SELECT BIT

CH2, CH1 and CH0 form a three bit field which is used to select one of two A/D channels. Channel 0 and Channel 1 correspond to Port G0 and G1 resp. input pin on the MCU. Channels 4-7 are used for internal reference points. In User Mode, channel 7 is reserved and converts to $00. The following table shows the signals selected by the channel select field.

Using a Port G pin as both an analog and digital input simultaneously is not recommended to prevent excess power dissipation. To digitally read Port G pins simultaneously the A/D should be disabled (ADON=0).

CH2	CH1	CH0	CHANNEL	SIGNAL
0	0	0	0	AD0/Port G Bit 0
0	0	1	1	AD1/Port G Bit 1
0	1	0	2	unused
0	1	1	3	unused
1	0	0	4	V_{REFH}
1	0	1	5	$(V_{REFH}+V_{REFL})/2$
1	1	0	6	V_{REFL}
1	1	1	7	FACTORY TEST

Table 8-1 A/D Channel Assignments

8.5 A/D DATA REGISTER ($15)

One 8-bit result register is provided. This register is updated each time COCO is set.

8.6 A/D DURING WAIT MODE

The A/D continues normal operation during WAIT mode. To decrease power consumption during WAIT, it is recommended that both the ADON and ADRC bits in the A/D Status and Control Register be cleared if the A/D converter is not being used. If the A/D converter is in use and the system clock rate is above 1.0 MHz, it is recommended that the ADRC bit be cleared.

8.7 A/D DURING STOP MODE

In STOP mode the comparator and charge pump are turned off and the A/D ceases to function. Any pending conversion is aborted. When the clocks begin oscillation upon leaving the STOP mode, a finite amount of time passes before the A/D circuits stabilize enough to provide conversions to the specified accuracy. Normally, the delays built into the 68HC705E24/68HC05E16 when coming out of STOP mode are sufficient for this purpose, therefore no explicit delays need to be built into the software.

CHAPTER 9
CORE TIMER

9.1 CORE TIMER

The Core Timer for this device is a 15-stage multi-functional ripple counter. The features include Timer Over Flow, Power-On Reset (POR), Real Time Interrupt, and COP Watchdog Timer

.

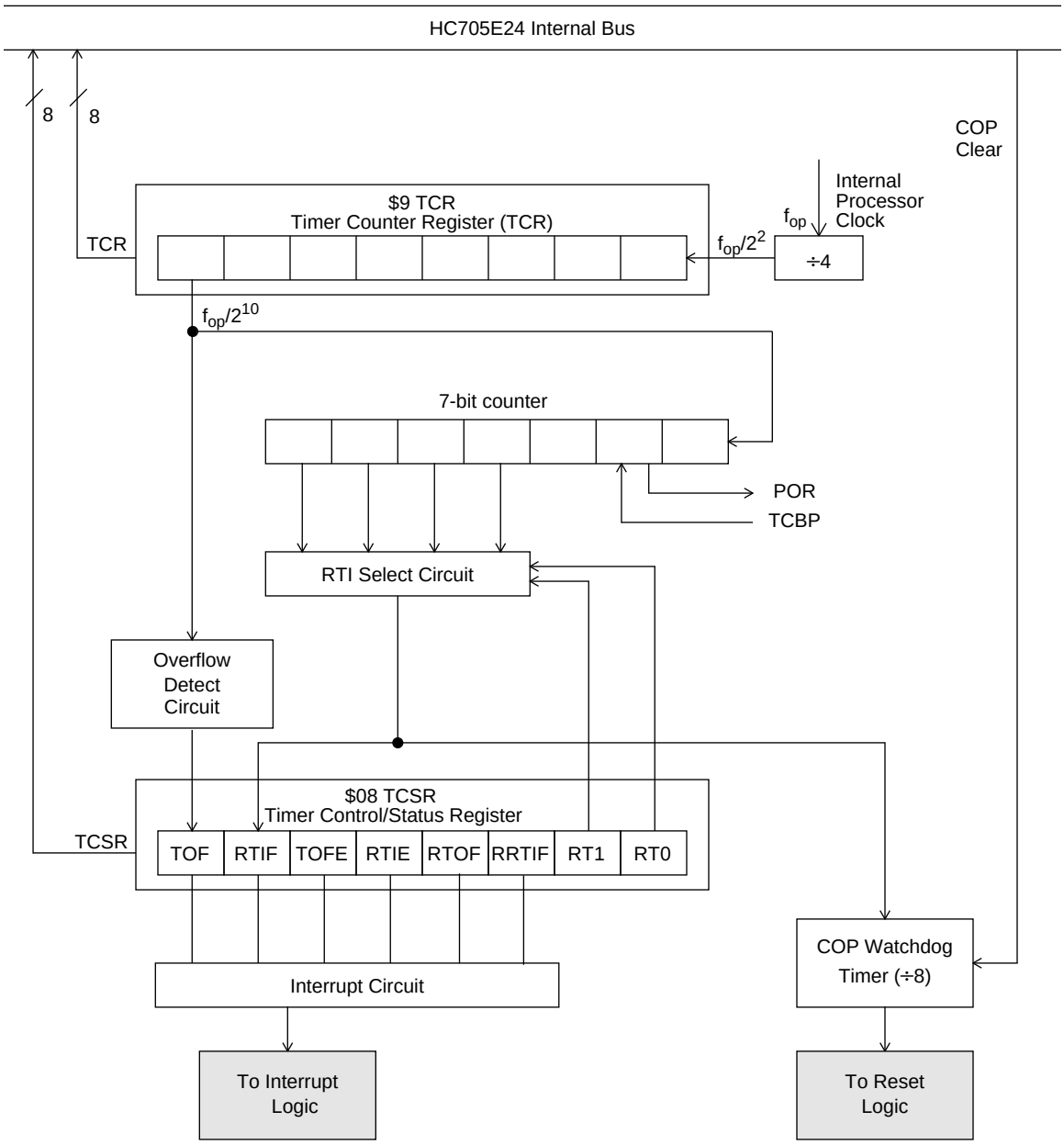


Figure 9-1 Core Timer Block Diagram

As seen in **Figure 9-1, Core Timer Block Diagram**, the Timer is driven by the output of the clock select circuit followed by a fixed divide by four prescaler. This signal drives an 8-bit ripple counter. The value of this 8-bit ripple counter can be read by the CPU at any time by accessing the Timer Counter Register (TCR) at address $09. A timer overflow function is implemented on the last stage of this counter, giving a possible interrupt at the rate of $f_{op}/1024$. Additional taps produce the POR function at $f_{op}/32$. The Timer Counter Bypass circuitry (available only in Test Mode) is at this point in the timer chain. This circuit is followed by two more stages, with the resulting clock ($f_{op}/16384$) driving the Real Time Interrupt circuit. The RTI circuit consists of three divider stages with a 1 of 4 selector. The output of the RTI circuit is further divided by eight to drive the mask optional COP Watchdog Timer circuit. The RTI rate selector bits, and the RTI and TOF enable bits and flags are located in the Timer Control and Status Register at location $08.

9.1.1 TIMER CONTROL AND STATUS REGISTER (TCSR) $08

The TCSR contains the timer interrupt flag, the timer interrupt enable bits, and the real time interrupt rate select bits. **Figure 9-4, Core Timer Counter Register (CTCSR)** shows the value of each bit in the TCSR when coming out of reset.

Bit	7	6	5	4	3	2	1	0
$08	TOF	RTIF	TOFE	RTIE	RTOF	RRTIF	RT1	RT0
RESET	0	0	0	0	W only0	W only0	1	1

Figure 9-2 Core Timer Control and Status Register (CTCSR)

9.1.1.1 TOF - Timer Over Flow

TOF is a read-only status bit and is set when the 8-bit ripple counter rolls over from $FF to $00. A CPU interrupt request will be generated if TOFE is set. Reset clears TOF.

9.1.1.2 RTIF - Real Time Interrupt Flag

The Real Time Interrupt circuit consists of a three stage divider and a 1 of 4 selector. The clock frequency that drives the RTI circuit is $f_{op}/2^{13}$ (or $f_{op}/8192$) with three additional divider stages giving a maximum interrupt period of 4 seconds at a crystal frequency of 32.768 kHz. RTIF is a read-only status bit and is set when the output of the chosen (1 of 4 selection) stage goes active. A CPU interrupt request will be generated if RTIE is set. Reset clears RTIF.

9.1.1.3 TOFE - Timer Over Flow Enable

When this bit is set, a CPU interrupt request is generated when the TOF bit is set. Reset clears this bit.

9.1.1.4 RTIE - Real Time Interrupt Enable

When this bit is set, a CPU interrupt request is generated when the RTIF bit is set. Reset clears this bit.

9.1.1.5 RTOF

This bit reads always as ’0’. Writing a ’1’ to this bits clears the timer overflow flag (TOF). Writing a zero to this bit has no effect.

9.1.1.6 RRTIF

This bit reads always as ’0’. Writing a ’1’ to this bits clears the real time interrupt flag (RTIF). Writing a zero to this bit has no effect.

9.1.1.7 RT1:RT0 - Real Time Interrupt Rate Select

These two bits select one of four taps from the Real Time Interrupt circuit. **Table 9-1, RTI Rates** shows the available interrupt rates with several f_{op} values. Reset sets these RT0 and RT1, selecting the lowest periodic rate and therefore the maximum time in which to alter these bits if necessary. Care should be taken when altering RT0 and RT1 if the time-out period is imminent or uncertain. If the selected tap is modified during a cycle in which the counter is switching, an RTIF could be missed or an additional one could be generated. To avoid problems, the COP should be cleared before changing RTI taps.

	RTI RATES AT f_{OP} FREQUENCY SPECIFIED:					
RT1:RT0	**16.384kHz**	**524kHz**	**1.049MHz**	**2.097MHz**	**4.194MHz**	**RATIO**
00	1s	31.3ms	15.6ms	7.8ms	3.9ms	$2^{14}/f_{op}$
01	2s	62.5ms	31.3ms	15.6ms	7.8ms	$2^{15}/f_{op}$
10	4s	125ms	62.5ms	31.3ms	15.6ms	$2^{16}/f_{op}$
11	8s	250ms	125.1ms	62.5ms	31.3ms	$2^{17}/f_{op}$

Table 9-1 RTI Rates

9.1.2 COMPUTER OPERATING PROPERLY (COP) WATCHDOG RESET

The COP watchdog timer function is implemented on this device by using the output of the RTI circuit and further dividing it by eight. The minimum COP reset rates are listed in **Table 9-2, Minimum COP Reset Times**. If the COP circuit times out, an internal reset is generated and the normal reset vector is fetched. Preventing a COP time-out is done by writing a ’0’ to bit 0 of address $7FF0. When the COP is cleared, only the final divide by eight stage (output of the RTI) is cleared.

Bit	7	6	5	4	3	2	1	0
$0F	LVIINT	LVIVAL	LVIRST	LVIENA	0	0	OPTCOP	OPTIRQ0
RESET	0	-	W only0	0	-	-	Wonce0	0

Figure 9-3 $\overline{\text{LVI}}$/OPTIONS Status/Control Register

The OPTCOP enables/disables the COP. This bit is cleared by reset thus enabling the COP. Writing a 1 disables the COP. This bit can be written only once, so that the COP state can not be changed after the first write to this register. It is recommended to write to this register as soon as possible after reset in order to lock the state of the COP.

	MINIMUM COP RESET AT f_{OP} FREQUENCY SPECIFIED:					
RT1:RT0	**16.384kHz**	**524kHz**	**1.049MHz**	**2.097MHz**	**4.194MHz**	**RATIO**
00	7s	218.8ms	109.4ms	54.7ms	27.3ms	$7*2^{14}/f_{op}$
01	14s	437.5ms	218.8ms	109.4ms	54.7ms	$7*2^{15}/f_{op}$
10	28s	875.0ms	437.5ms	218.8ms	109.4ms	$7*2^{16}/f_{op}$
11	56s	1.75s	875.0ms	437.5ms	218.8ms	$7*2^{17}/f_{op}$

Table 9-2 Minimum COP Reset Times

9.1.3 TIMER COUNTER REGISTER (TCR) $09

The Timer Counter Register is a read-only register which contains the current value of the 8-bit ripple counter at the beginning of the timer chain. This counter is clocked at f_{op} divided by 4 and can be used for various functions including a software input capture. Extended time periods can be attained using the TOF function to increment a temporary RAM storage location thereby simulating a 16-bit (or more) counter.

Bit	7	6	5	4	3	2	1	0
$09	CCNT7	CCNT6	CCNT5	CCNT4	CCNT3	CCNT2	CCNT1	CCNT0
RESET	0	0	0	0	0	0	0	0

Figure 9-4 Core Timer Counter Register (CTCSR)

The power-on cycle clears the entire counter chain and begins clocking the counter. After 32 cycles, the power-on reset circuit is released which again clears the counter chain and allows the device to come out of reset. At this point, if $\overline{\text{RESET}}$ is not asserted, the timer will start counting up from zero and normal device operation will begin. When $\overline{\text{RESET}}$ is asserted anytime during operation (other than POR), the counter chain will be cleared.

CHAPTER 10
CUSTOM PERIODIC INTERRUPT

10.1 CUSTOM PERIODIC INTERRUPT

The custom periodic interrupt is software programmable to a 0.25 second, 0.5 second, or 1 second interrupt. The interrupt is generated from the 32 kHz XOSC1 input by a 15-bit counter. This interrupt is under the control of the Custom Periodic Interrupt Control/Status Register located at $06.

10.1.1 CUSTOM PERIODIC INTERRUPT CONTROL/STATUS REGISTER ($06).

The CPICSR contains the CPI flag and enable bits. **Figure 10-1, Custom Periodic Interrupt Control/Status Register (CPICSR)** shows the location of these bits and their values after reset.

Bit	7	6	5	4	3	2	1	0
$06	0	CPIF	0	CPIE	0	0	CPI1	CPI0
RESET	0	0	0	0		0	0	0

Figure 10-1 Custom Periodic Interrupt Control/Status Register (CPICSR)

10.1.1.1 CPIF - Custom Periodic Interrupt Flag

CPIF is a clearable, read-only status bit and is set when the 15-bit counter changes from $7FFF to $0000. A CPU interrupt request will be generated if CPIE is set. Clearing the CPIF is done by writing a ’0’ to it. Writing a ’1’ to CPIF has no effect on the bit’s value. Reset clears CPIF.

10.1.1.2 CPIE - Custom Periodic Interrupt Enable

When this bit is cleared, the counter is cleared and CPI interrupts are disabled. When this bit is set, the counter starts from $0000 and a CPU interrupt request is generated when the CPIF bit is set. Reset clears this bit.

10.1.1.3 CPI1 and CPI0 Custom Periodic Interrupt Rate Select

These two read/write bits select one of three taps from the 15-stage counter to provide an interrupt rate according to **Table 10-1, CPI Rates (32.768kHz crystal)**. Reset clears these bits selecting the lowest periodic rate. Care should be taken when altering CPI1 and CPI0 if the timeout period is imminent or uncertain. If the selected tap is modified during a cycle in which the counter is switching, a CPIF could be missed or an additional one could be generated.

CPI1	CPI0	CPI Interrupt Rate
0	0	1s (Reset Condition)
0	1	1s (reserved)
1	0	0.5s
1	1	0.25s

Table 10-1 CPI Rates (32.768kHz crystal)

10.2 OPERATION DURING STOP MODE

The timer system is cleared and the CPI counter is halted when going into STOP mode. When STOP is exited by an external interrupt or an external RESET, the internal oscillator will resume, followed by a 1024 internal processor oscillator stabilization delay. The timer system counter is then cleared and operation resumes. The CPI will continue counting once the oscillator resumes and does not wait for the oscillator to stabilize.

10.3 OPERATION DURING WAIT MODE

The CPU clock halts during the WAIT mode, but the timer and CPI remain active. If interrupts are enabled, a timer interrupt or custom periodic interrupt will cause the processor to exit the WAIT mode.

CHAPTER 11
PHASE-LOCKED LOOP

11.1 PHASE-LOCKED LOOP SYNTHESIZER

The phase-locked loop (PLL) consists of a variable bandwidth loop filter, a voltage controlled oscillator (VCO), a feedback frequency divider, and a digital phase detector. A small external capacitor (typically 0.1µF) is used by the loop filter. V_{DDSyn} is the supply source for the PLL and may be bypassed to minimize noise.

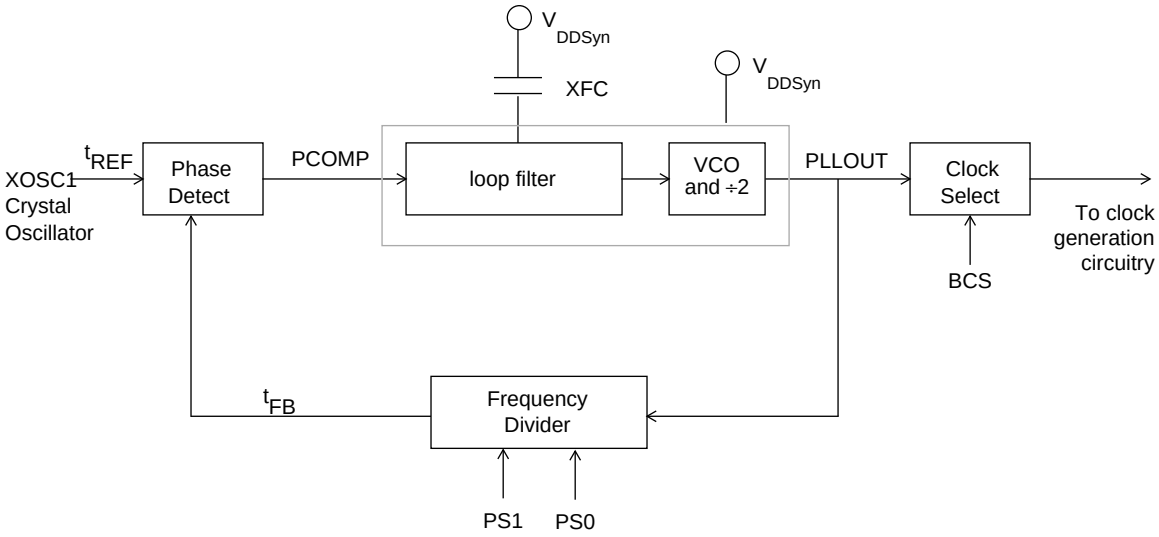


Figure 11-1 PLL Circuit

The phase detector compares the frequency and phase of the feedback frequency (t_{FB}) and the crystal oscillator reference frequency (t_{REF}) and generates the output, PCOMP, as shown in **Figure 11-1, PLL Circuit**. The output wave-form is then integrated and amplified. The resultant dc voltage is applied to the voltage controlled oscillator. The output of the VCO is divided by a variable frequency divider of 256, 128, 64, or 32 to provide the feedback frequency for the phase detector.

To change PLL frequencies, follow the procedure outlined below to modify the bits in the PLL control register:

1. Clear BCS to enable the low frequency bus rate,
2. Clear PLLON to disable the PLL,
3. Select the speed using PS1 and PS0,
4. Set PLLON to enable the PLL,

5. Wait for the BWC bit to go low,
6. Set BCS to switch to the high frequency bus rate.

The user should not switch among the high speeds with the BCS bit set. Following the procedure above will prevent possible bursts of high frequency operation during the re-configuration of the PLL. Using a 32KHz oscillator, the start-up time of the PLL is less than 10 milliseconds.

The PLL loop filter has two bandwidths which are automatically selected by the PLL if AUTO=1. Whenever the PLL is first enabled, the wide bandwidth mode is used. This enables the PLL frequency to ramp up quickly. When the output frequency is near the desired frequency, the filter is switched to the narrow bandwidth mode to make the final frequency more stable. Manual control can be done by clearing AUTO in the PLLCR and setting the appropriate value for BWC.

The PLL requires an external filter capacitor connected between XFC and VDDSyn. This capacitor should be located as close to the chip as possible to minimize noise. A typical value for this capacitor is 0.1 μF.

11.1.1 PHASE-LOCKED LOOP CONTROL REGISTER (PLLCR) $07

This read/write register contains the control bits to select the PLL frequency and enable/ disable the synthesizer.

Bit	7	6	5	4	3	2	1	0
$07	0	BCS	AUTO	BWC	PLLON	VCOTST	PS1	PS0
RESET	0	0	1	0	1	1	0	0

Figure 11-2 Phase-Locked Loop Control Register (PLLCR)

11.1.1.1 BCS - Bus Clock Select

When this bit is set, the output of the PLL is used to generate the internal processor clock. When clear, the internal bus clock is driven by the crystal (XOSC1÷2). Once BCS has been changed, it may take up to 1.5 XOSC1 cycles + 1.5 PLLOUT cycles to make the transition. During the transition, the clock select output will be held low and all CPU and timer activity will cease until the transition is complete. Reset clears this bit.

11.1.1.2 AUTO

When set, this bit selects the automatic bandwidth circuitry in the Phase detect block. When clear, manual bandwidth control is selected. Reset sets this bit.

NOTE: This bit is intended for use by Motorola to test and characterize the PLL. The user should always have this bit set to 1.

11.1.1.3 BWC - Bandwidth Control

This bit selects high bandwidth control when set, and low bandwidth control when clear. The low bandwidth driver is always enabled, so this bit determines whether the high bandwidth driver is on or off. Bandwidth control is under manual control only when the AUTO bit is clear. When the AUTO bit is set, BWC acts as a read-only status bit to indicate which mode has been selected by the internal circuit. On PLL start-up in automatic mode (AUTO=1), the high bandwidth driver is enabled (BWC=1) by internal circuitry until the PLL has locked onto the specified frequency. The high bandwidth driver is then disabled and BWC is cleared by internal circuitry. The user can poll BWC to determine when the PLL output is stable and BCS can be set. Reset clears this bit. See **Table 11-1, Loop Filter Bandwidth Control**.

AUTO	BWC	VCOTST	HIGH BANDWIDTH	LOW BANDWIDTH
0	0	0	OFF	OFF
0	0	1	OFF	ON
0	1	0	ON	OFF
0	1	1	ON	ON
1	X	1	AUTO	ON

Table 11-1 Loop Filter Bandwidth Control

NOTE: This bit is intended for use by Motorola to test and characterize the PLL. The user should always have this bit set to 0.

11.1.1.4 PLLON - PLL On

This bit activates the synthesizer circuit without connecting it to the control circuit. This allows the synthesizer to stabilize before it can drive the CPU clocks. When this bit is cleared, the PLL is shut off. Reset sets this bit.

NOTE: PLLON should not be cleared while using the PLL to drive the internal processor clock, i.e. when BCS is high. If the internal processor clock is driven by the PLL, clearing the PLLON bit would cause the internal processor clock to stop. Exercise caution when using these bits.

11.1.1.5 VCOTST - VCO Test

This bit is used to isolate the loop filter from the VCO in order to facilitate testing. When clear, the low bandwidth mode of the PLL filter is disabled. When set, the loop filter operates as indicated by the values of AUTO and BWC. This bit is always set when AUTO=1 as security when running in automatic mode. Reset sets this bit.

NOTE: This bit is intended for use by Motorola to test and characterize the PLL. The user should always have this bit set to 1.

11.1.1.6 PS1:PS0 - PLL Synthesizer Speed Select

These two bits select one of four taps from the PLL to drive the CPU clocks. These bits are used in conjunction with PLLON and BCS bits in the PLL Control Register. These bits should not be written if BCS in the PLLCR is at a logic high. Reset clears PS1 and PS0, choosing a bus clock frequency of 524kHz.

PS1	PS0	CPU BUS CLOCK FREQUENCY (f_{OP})
0	0	524kHz (Reset Condition)
0	1	1.049MHz
1	0	2.097MHz
1	1	4.194MHz (to be determined)

Table 11-2 PS1 and PS0 Speed Selects (32.768kHz crystal)

11.1.2 NOISE IMMUNITY

The MCU should be insulated as much as possible from noise in the system. We recommend the following steps be taken to help prevent problems due to noise injection.

1. The application environment should be designed so that the MCU is not near signal traces which switch often, such as a clock signal,
2. The oscillator circuit for the MCU should be placed as close as possible to the XOSC1 and XOSC2, resp. OSC1 and OSC2 pins on the MCU, and
3. All power pins should be filtered (to minimize noise on these signals) by using bypass capacitors placed as close as possible to the MCU.

See the Application Note Designing for Electromagnetic Compatibility (EMC) with HCMOS Microcontrollers, available through the Motorola Literature Distribution Center, document number AN1050/D.

CHAPTER 12
OSCILLATOR SYSTEM

12.1 INTRODUCTION

This chip has three different clock sources which all have their individual advantages and disadvantages. The user is able to select the oscillator which fits his needs under software control. The oscillators not needed, should be stopped in order to reduce power consumption. **Figure 12-1, Block Diagram of the Oscillator System** shows an overview of the oscillator system.

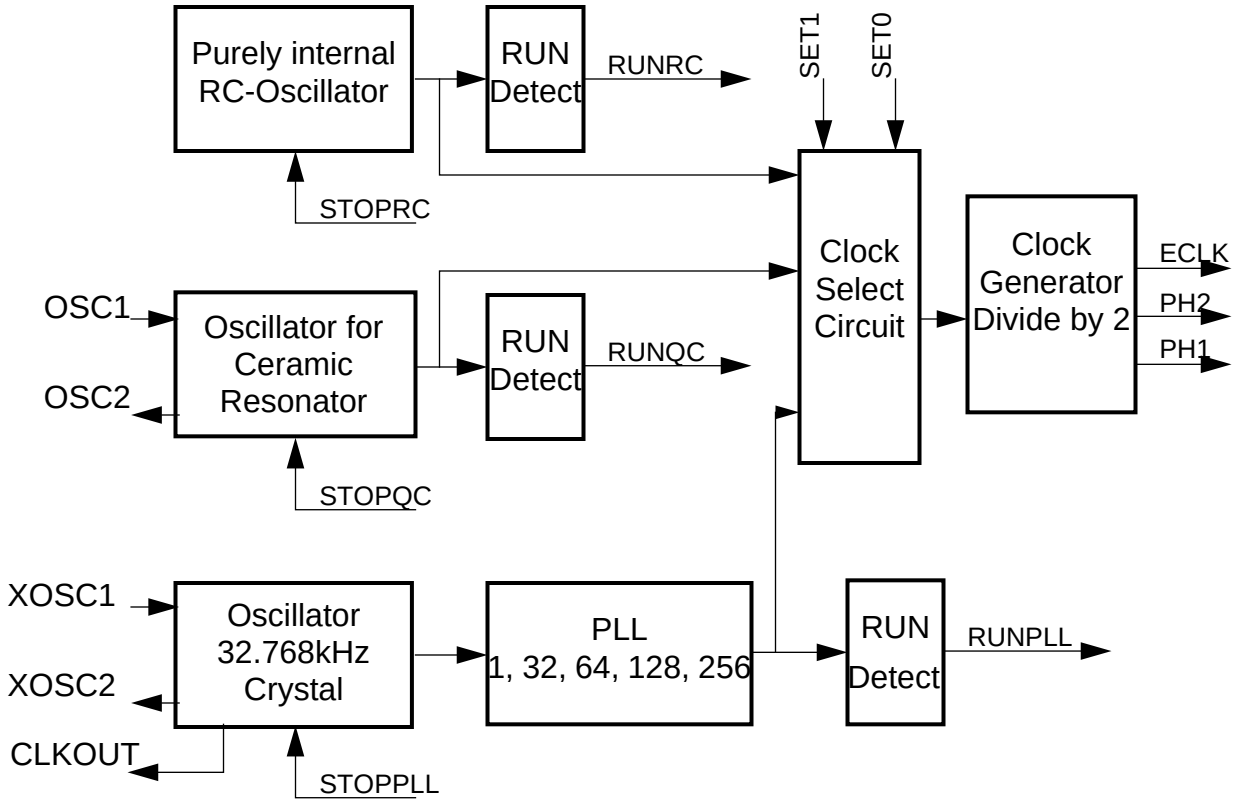
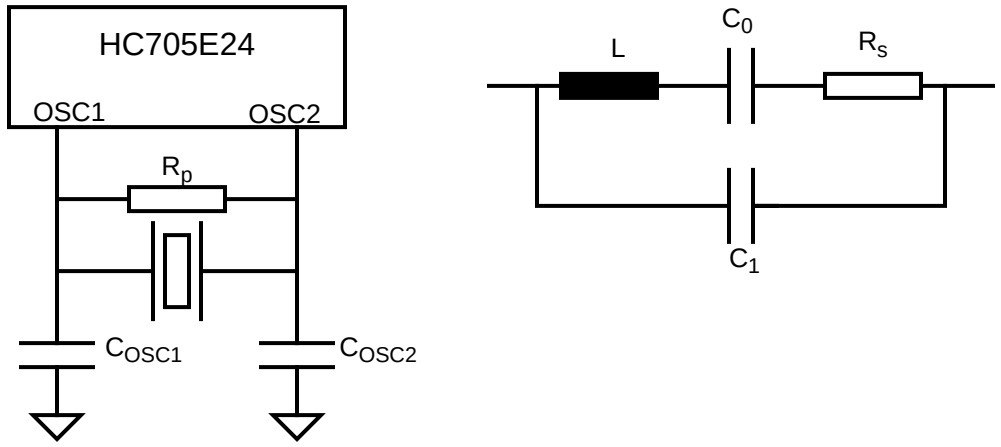


Figure 12-1 Block Diagram of the Oscillator System

12.2 HIGH FREQUENCY OSCILLATOR

The circuit shown in **Figure 12-2, Crystal/Ceramic-Resonator Oscillator Connections** is recommended when using a crystal or a ceramic resonator. The internal oscillator is designed to interface with an AT-cut crystal or ceramic resonator in the frequency range specified by f_{osc}. Use of an external CMOS oscillator connected to OSC1 is recommended when crystals outside the specified ranges are used. In this case OSC2 should be unconnected with a maximum load of 20pF. The crystal and components should be mounted as close as possible to the OSC1, OSC2 pins to minimize output distortion and start-up stabilization time. If short start-up times are needed the use of a ceramic resonator is recommended, whereas a crystal is more appropriate if high accuracy is required. If this oscillator is not used OSC1 must be tied to VSS.



Oscillator Connections Equivalent Crystal Circuit

Figure 12-2 Crystal/Ceramic-Resonator Oscillator Connections

Recommended values for the crystal/ceramic resonator are shown in **Table 12-1, Crystal/Ceramic Resonator Parameters**.

	Crystal		Ceramic	
	2 MHz	4 MHz	2-4 MHz	Unit
R_{Smax}	400	75	10	Ohms
C_0	5	7	40	pF
C_1	0.008	0.012	4.3	pF
C_{OSC1}	15-40	15-30	30	pF
C_{OSC2}	15-30	15-25	30	pF
R_p	10	10	1-10	MOhms
Q	30000	40000	1250	1

Table 12-1 Crystal/Ceramic Resonator Parameters

12.3 LOW FREQUENCY OSCILLATOR

The circuit shown in **Figure 12-2, Crystal/Ceramic-Resonator Oscillator Connections** is recommended when using a typical 32.768kHz watch crystal. An external CMOS oscillator connected to XOSC1 can also be used to drive the oscillator circuit. In this case XOSC2 should be unconnected with a maximum load of 5pF. The crystal and components should be mounted as close as possible to the XOSC1, XOSC2 pins to minimize output distortion and start-up stabilization time. Special care should also be taken for the VSS connection of the external filter capacitor and the decoupling of the VDDSYN power. If this oscillator is not used, XOSC1 must be tied to VSS.

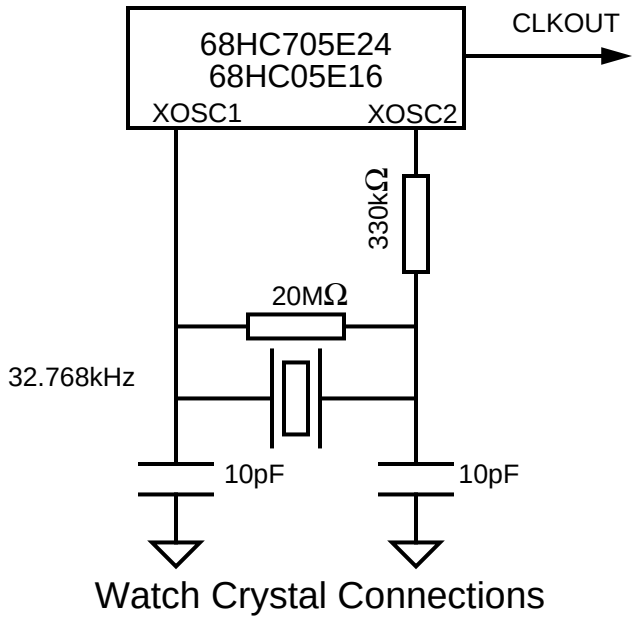


Figure 12-3 Recommended Watch Crystal Connections

12.4 INTERNAL RC-OSCILLATOR

This purely internal RC-Oscillator is used in order to provide a very fast start-up. Since the accuracy of this oscillator is very poor (25%), the user should switch to one of the other oscillators as soon as possible to achieve better accuracy, after having completed the simple software initialization procedures.

12.5 OSCILLATOR SELECT REGISTER ($1E)

This read/write register contains the control bits to start/stop the oscillators, to select the active oscillator and to obtain status information about the oscillators.

Bit	7	6	5	4	3	2	1	0
1E$	STOPPLL	STOPQC	STOPRC	SET1	SET0	RUNPLL	RUNQC	RUNRC
RESET	0	0	0	0	0	0	0	0

Figure 12-4 Oscillator Select Register (OSCREG)

12.5.1 STOPPLL

When this read/write bit is set, the oscillator connected to the pins XOSC1, XOSC2 providing the clock for the PLL and the CPI is stopped. It is not possible to set this bit (i.e. stop the oscillator), if the PLL clock is selected as active clock. Reset and Stop clear this bit, so that this oscillator is started recovering from Reset or Stop.

12.5.2 STOPQC

When this read/write bit is set, the oscillator connected to the pins OSC1, OSC2 is stopped. It is not possible to set this bit (i.e. stop the oscillator), if the ceramic resonator clock is selected as active clock. Reset and Stop clear this bit, so that this oscillator is started recovering from Reset or Stop.

12.5.3 STOPRC

When this read/write bit is set, the internal RC-oscillator is stopped. It is not possible to set this bit (i.e. stop the oscillator), if the RC-oscillator clock is selected as active clock. Reset and Stop clear this bit, so that this oscillator is started recovering from Reset or Stop.

12.5.4 SET1, SET0

These two read/write bits select the current active oscillator acting as a clock source for the whole system. An oscillator not running cannot be selected as active oscillator. The current active oscillator will not be suspended. The bits SET1, SET0 always reflect the current active oscillator. Reset and Stop clear these bits, so that the internal RC-oscillator is the active one recovering from Reset or Stop.

SET1	SET0	Selected Oscillator
0	0	Internal RC-oscillator
0	1	High Frequency Oscillator
1	0	Low Frequency Oscillator
1	1	**Illegal**

Table 12-2 Oscillator Selection

12.5.5 RUNPLL

This read-only bit is set if the PLL-oscillator is properly stabilized. The stabilization period for this oscillator is 1024 cycles. Reset and Stop clear this bit.

12.5.6 RUNQC

This read-only bit is set if the high frequency oscillator is properly stabilized. The stabilization period for this oscillator is 1024 cycles. Reset and Stop clear this bit.

12.5.7 RUNRC

This read-only bit is set if the internal RC-oscillator is properly stabilized. The stabilization period for this oscillator is 128 cycles. Reset and Stop clear this bit.

CHAPTER 13
ELECTRICAL SPECIFICATIONS

13.1 MAXIMUM RATINGS

Ratings	Symbol	Value	Unit
Supply Voltage	V_{DD}	-0.3 to +7.0	V
Input Voltage	V_{in}	V_{SS} - 0.3 to V_{DD} + 0.3	V
Boot-Loader Mode ($\overline{IRQ0}$ Pin only)	V_{in}	V_{SS} - 0.3 to 2 x V_{DD} + 0.3	V
Current Drain per Pin excl. V_{DD} and V_{SS}	I	25	mA
Operating Temperature Range (standard plastic package)	T_A	0 to +70	˚C
Storage Temperature Range	T_{STG}	-65to +150	˚C

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum-rated voltages to this high-impedance circuit. For proper operation, it is recommended that V_{in} and V_{out} be constrained to the range $V_{SS} \le (V_{in}$ or $V_{out}) \le V_{DD}$. Reliability of operation is enhanced if unused inputs are connected to an appropriate logic voltage level (e.g., either V_{SS} or V_{DD}).

13.2 THERMAL CHARACTERISTICS

Characteristics	Symbol	Value	Unit
Thermal resistance Plastic QFP64	Θ_{JA}	TBD	˚C/W

13.3 DC ELECTRICAL CHARACTERISTICS

(V_{DD} = 5.0 V_{dc} ±10%, V_{SS} = 0 V_{dc}, T_A = 0°C to +70°C, unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Output Voltage I_{load} 10µA	V_{OL} V_{OH}	- V_{DD}-0.1	- -	0.1 -	V V
Output High Voltage (I_{Load}=0.8mA) P(A - F)F	V_{OH}	V_{DD} - 0.8	V_{DD} - 0.4	-	V
Output Low Voltage (I_{Load}= 1.6mA) P(A - F), CLKOUT	V_{OL}	-	0.1	0.4	V
Input High Voltage P(A - F), PG, OSC1, RESET, IRQ0, IRQ1, LVI	V_{IH}	0.7 x V_{DD}	-	V_{DD}	V
Input Low Voltage P(A - F), PG, OSC1, RESET, IRQ0, IRQ1, LVI	V_{IL}	V_{SS}	-	0.2 x V_{DD}	V
XFC Wide Bandwidth Source Sink	I_{OH} I_{OL}	-20 20	-41 41	- -	µµA µµA
XFC Narrow Bandwidth Source Sink	I_{OH} I_{OL}	-1 1	-4 4	- -	µµA µµA
Pull-up Source Current (V_{OH}=4V) Port B and C, D, E, F if enabled	I_{PU}	40	-	100	µµA
Pull-down Sink Current (V_{OL}=0.8V) Port A	I_{PD}	40	-	100	µµA
I/O Ports Hi-Z Leakage Current PC, PD, PE, PF, RESET, IRQ0, IRQ1, LVI	I_{OZ}	-	0.2 ±		±1µA
Capacitance All I/Os	C_{OUT}	-	-	12	pF
Hysteresis RESET, IRQ0/1, LVI, TCAP1/2, Port C, SDA0/1, SCL0/1	V_{HYST}	-	1.0	-	V

Characteristic	Symbol	Min	Typ	Max	Unit
Supply Current (See Notes)					
RUN (@ 2MHz f_{BUS})	I_{DD}	-	5	12	mA
RUN (@ 16kHz f_{BUS})	I_{DD}	-	100	200	µµA
WAIT (@ 2MHz f_{BUS})	I_{DD}	-	1	4	mA
WAIT (@16kHz f_{BUS})	I_{DD}	-	30	50	µµA
STOP (Oscillators off) (0 - 70 ˚C)	I_{DD}	-	1	8	µµA

NOTES:

1. All values shown reflect average measurements.
2. Typical values at midpoint of voltage range, i.e. exactly V_{DD} = 5V, 25°C only.
3. Wait I_{DD}: Only timer system active.
4. Run (Operating) I_{DD}, Wait I_{DD}: Measured using external square wave clock source (f_{osc} = 4.2 MHz), all inputs 0.2 V from rail; no dc loads, less than 50pF on all outputs, C_L = 20 pF on OSC2, C_L = 5pF on XOSC2.
5. Wait, Stop I_{DD}: All ports configured as inputs, V_{IL} = 0.2 V, V_{IH} = V_{DD}-0.2 V.
6. Stop I_{DD} measured with XOSC1, OSC1 = V_{SS}.
7. Wait I_{DD} is affected linearly by the XOSC2, OSC2 capacitance.

13.4 A/D CONVERTER CHARACTERISTICS

(V_{DD} = 5.0 V_{dc} ±10%, V_{SS} = 0 V_{dc}, T_A = 0°C to +70°C, unless otherwise noted)

Characteristic	Parameter	Min	Max	Unit
Resolution	Number of Bits resolved	8	-	Bit
Non-Linearity	Max deviation from the best straight line through A/D transfer characteristics (V_{RH}=V_{DD}, V_{RL}=V_{SS})	-	1/2	± LSB
Quantization Error	Uncertainly due to converter resolution	-	1/2	± LSB
Absolute Accuracy	Difference between the actual input Voltage and the full scale equivalent of the binary output code for all errors	-	1	±LSB
Conversion Range	Analog input voltage range	V_{RL}	V_{RH}	V
V_{RH}	Max. analog refer. voltage	V_{RL}	V_{DD}+0.1	V
V_{RL}	Min. analog reference voltage	V_{SS}-0.1	V_{RH}	V
Conversion Time	a) External clock b) Internal RC oscillator	- -	32 32	T_{CYC} µµs
Monotonicity	Conversion result never decreases with an increase in input voltage and has no missing codes	GUARANTEED		
Zero Input Reading	Conversion result V_{IN}=V_{RL}	00	-	Hex
Full Scale Reading	Conversion result V_{IN}=V_{RH}	-	FF	Hex
Sample Acquisition Time (See Note 1)	a) External clock b) Internal RC oscillator	- -	12 12	T_{CYC} µµs
Sample/Hold Cap.	Input capacitance on AD(0-1)		12	pF
Input Leakage	Input leakage on AD(0-1) Input leakage on V_{RL}, V_{RH}	- -	1 1	µµA µµA

NOTES:
1. Source impedances greater than 10k ohm will adversely affect internal RC charging time during input sampling.
2. The external system error caused by input leakage current is approximately equal to the product of R source and input current
3. A/D accuracy may decrease as V_{RH} is reduced below 4V.

13.5 CONTROL TIMING

(V_{DD} = 5.0 V_{dc} ±10%, V_{SS} = 0 V_{dc}, T_A = 0°C to +70°C, unless otherwise noted)

Characteristic	Symbol	Min	Max	Unit
Frequency of Operation Crystal Option External	 f_{OSC} f_{OSC}	 - dc	 32.768 4.2	 kHz MHz
Internal Operating Frequency Crystal Option External	 f_{OP} f_{OP}	 - dc	 16.384 2.1	 kHz MHz
Cycle Time	t_{CYC}	480	-	ns
Ceramic Resonator Start-up Time	t_{OCOV}	-	10	ms
Ceramic Resonator Stop Recovery Time	t_{ICCH}	-	10	ms
Crystal Oscillator Start-up Time (32kHz)	t_{OXOV}	-	1000	ms
Stop Recovery Start-up Time (32kHz))	t_{ILCH}	-	1000	ms
PLL Start-up Time	t_{PLLS}	-	10	ms
PLL Stability (V_{DD}=2.7V-6.0V, f=1.049MHz)		3.0	3.3	%
OSC1 Pulse Width	t_{OH}, t_{OL}	90	-	ns
RESET Input Pulse Width	t_{RL}	1.5	-	t_{CYC}
EEPROM Byte Erase Time (0 - 70˚C)	t_{eera}		10	ms
EEPROM Block Erase Time (0 - 70˚C)	t_{eera}		100	ms
EEPROM Bulk Erase Time (0 - 70˚C)	t_{eera}		400	ms
EEPROM Byte Program Time (0 - 70˚C)	t_{eepgm}		10	ms
RC oscillator stabilization time (AD)	t_{RCON}		5	µµs
A/D on current stabilization time	t_{ADON}		100	µµs
16 Bit Timer Resolution (note 4) Input Capture Pulse Width Input Capture Pulse Period	 t_{resl} t_{TLTH} t_{TLTL}	 4 250 note 3	 - - -	 t_{CYC} ns t_{CYC}
Interrupt Pulse Width Low (Edge triggered)	t_{ILIH}	250	-	ns
Interrupt Pulse Period	t_{ILIL}	note 2	-	t_{CYC}

NOTES:

1. For Bus frequencies less than 1.0MHz the internal RC oscillator should be used when programming the EEPROM.
2. The minimum period t_{ILIL} should not be less than the number of cycles it takes to execute the interrupt service routine plus 21 t_{CYC}.
3. The minimum period $t_{tlt l}$ should not be less than the number of cycles it takes to execute the capture interrupt service routine plus 24 t_{CYC}.
4. Since a 2-bit prescaler in the timer must count four external cycles (t_{CYC}) this is the limiting factor in determining the timer resolution.

How to reach us:

MFAX: RMFAX0@email.sps.mot.com - TOUCHTONE (602) 244-6609

INTERNET: http://Design-net.com

USA/EUROPE: Motorola Literature Distribution; P.O. Box 20912; Phoenix, Arizona 85036. 1-800-441-2447

JAPAN: Nippon Motorola Ltd.; Tatsumi-SPD-JLDC, Toshikatsu Otsuki, 6F Seibu-Butsuryu-Center, 3-14-3 Tatsumi Koto-Ku, Tokyo 135, Japan. 03-3521-8315

HONG KONG: Motorola Semiconductors H.K. Ltd.; 8B Tai Ping Industrial Park, 51 Ting Tok Road, Tai Po, N.T., Hong Kong. 852-26629298

MOTOROLA

HC05E16GRS/D