MOTOROLA

MC3870

Advance Information

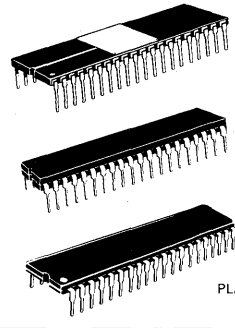
SINGLE-CHIP MICROCONTROLLER

The MC3870 is a monolithic 8-bit microcomputer utilizing ion-implanted, N-channel silicon-gate technology and advanced circuit design techniques. The single-chip MC3870 offers maximum cost effectiveness in a wide range of control and logic replacement applications.

- Software Compatible with F8 Family
- 2048 Byte Mask Programmable ROM
- 64 Byte Scratchpad RAM
- 32 Bits (4 Ports) TTL-Compatible I/O
- Programmable Binary Timer
  - Interval Timer Mode
  - Pulse Width Measurement Mode
  - Event Counter Mode
- External Interrupt
- Crystal, LC, RC, External
- Low Power (275 mW Typ.)
- Single +5 Volt ± 10% Power Supply

MOS
(N-CHANNEL, SILICON-GATE DEPLETION LOAD)

SINGLE-CHIP MICROCONTROLLER



L SUFFIX
CERAMIC PACKAGE
CASE 715

S SUFFIX
CERDIP PACKAGE
CASE 734

P SUFFIX
PLASTIC PACKAGE
CASE 711

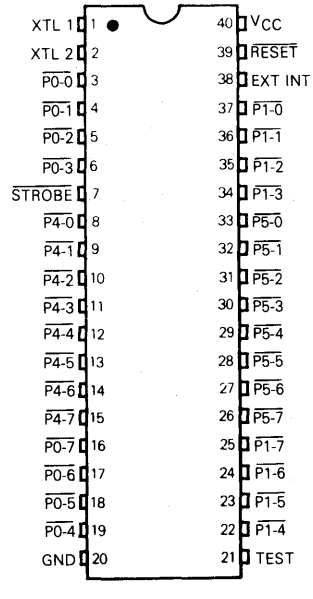
3

ABSOLUTE MAXIMUM RATINGS*

	Operating Temperature	
	0 to 70°C	−40 to +85°C
Temperature Under Bias	−20°C to +85°C	−50°C to +100°C
Storage Temperature	−65°C to +150°C	−65°C to +150°C
Voltage on any Pin with Respect to Ground (Except open-drain pins and TEST)	−1.0 V to +7 V	−1.0 V to +7 V
Voltage on TEST with Respect to Ground	−1.0 V to +9 V	−1.0 V to +9 V
Voltage to Open-Drain Pins with Respect to Ground	−1.0 V to +13.5 V	−1.0 V to +13.5 V
Power Dissipation	1.5 W	1.5 W
Power Dissipation by any One I/O Pin	60 mW	60 mW
Power Dissipation by All I/O Pins	600 mW	600 mW

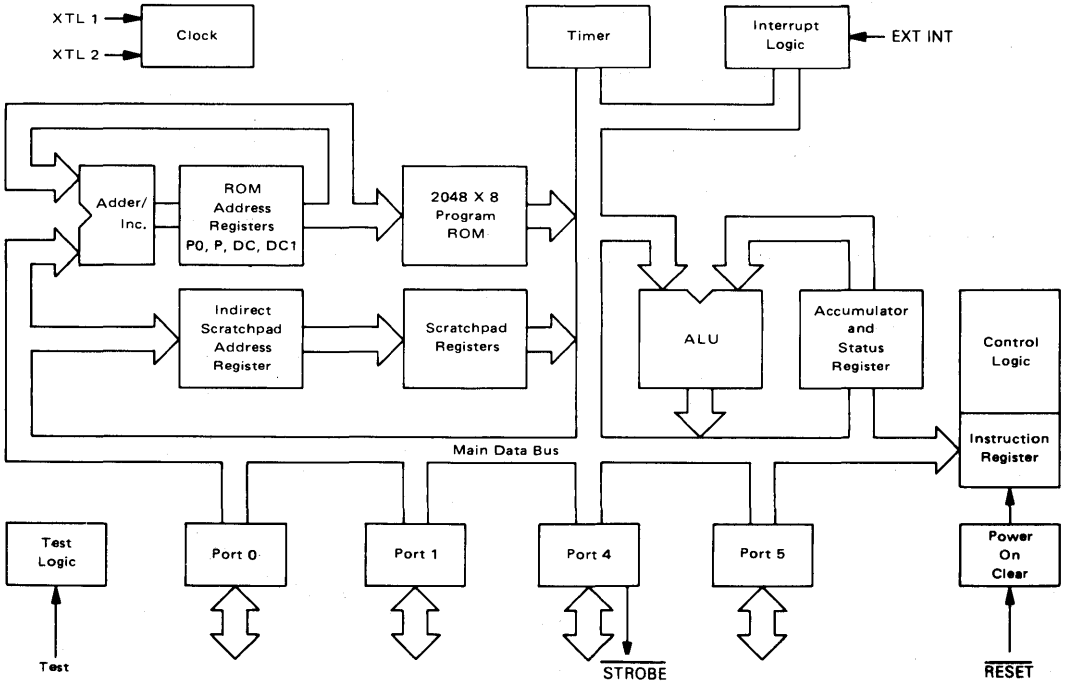
*Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

PIN ASSIGNMENT



This is advance information and specifications are subject to change without notice.
Published by Motorola Inc. with permission of Mostek, Inc.

FIGURE 1 — BLOCK DIAGRAM



AC CHARACTERISTICS

Signal	Symbol	Parameter	0 to 70°C		−40° to +85°C		Unit	Notes
			Min	Max	Min	Max		
XTL1 XTL2	to	Time Base Period, all clock modes	250	1000	250	500	ns	
	tex(H)	External clock pulse width high	90	700	100	390	ns	
	tex(L)	External clock pulse width low	100	700	110	390	ns	
ϕ	tϕ	Internal ϕ clock	2t$_0$			2t$_0$		
WRITE	tw	Internal WRITE Clock period	4tϕ 6tϕ			4tϕ 6tϕ		Short Cycle Long Cycle
I/O	tdI/O	Output delay from internal WRITE clock	0	1000	0	1200	ns	50 pF plus one TTL load
	tsI/O	Input setup time to internal WRITE clock	1000		1200		ns	
$\overline{\text{STROBE}}$	t$_{I/O-s}$	Output valid to $\overline{\text{STROBE}}$ delay	3tϕ − 1000	3tϕ + 250	3tϕ − 1200	3tϕ + 300	ns	I/O load = 50 pF + 1 TTL load
	t$_{sL}$	$\overline{\text{STROBE}}$ low time	8tϕ − 250	12tϕ + 250	8tϕ − 300	12tϕ + 300	ns	$\overline{\text{STROBE}}$ load = 50 pF + 3 TTL loads
$\overline{\text{RESET}}$	t$_{RH}$	$\overline{\text{RESET}}$ hold time, low	6tϕ + 750		6tϕ + 1000		ns	
	t$_{RPOC}$	$\overline{\text{RESET}}$ hold time, low for power clear	power supply rise time + 0.1		power supply rise time + 0.15		ms	
EXT INT	t$_{EH}$	EXT INT hold time in active and inactive state	6tϕ + 750		6tϕ + 1000		ns	To trigger interrupt
			2tϕ		2tϕ		ns	To trigger timer

DC CHARACTERISTICS (I/O Power Dissipation ≤ 100 mW) (Note 2)

Symbol	Parameter	0 to 70 + C		−40 to +85°C		Unit	Conditions
		Min	Max	Min	Max		
V_{CC}	Power Supply Voltage	4.5	5.5	4.75	5.25	V	
V_{IHEX}	External Clock Input High Level	2.4	V_{CC}	2.4	V_{CC}	V	
V_{ILEX}	External Clock Input Low Level	−0.3	0.6	−0.3	0.6	V	
I_{IHEX}	External Clock Input High Current	—	100	—	130	μA	$V_{IHEX}=2.40$
I_{ILEX}	External Clock Input Low Current	—	−100	—	−130	μA	$V_{ILEX}=0.60$
$V_{IHI/O}$	Input High Level, I/O Pins	2.0	V_{CC}	2.2	V_{CC}	V	Standard Pullup
		2.0	13.2	2.2	13.2	V	Open Drain (1)
V_{IHR}	Input High Level, $\overline{RESET}$	2.0	V_{CC}	2.2	V_{CC}	V	
V_{IHEI}	Input High Level, EXT INT	2.0	V_{CC}	2.2	V_{CC}		
V_{IL}	Input Low Level	−0.3	0.8	−0.3	0.7	V	(1)
I_{IL}	Input Low Current, All Pins with Standard Pullup Resistor	—	−1.6	—	−1.9	mA	$V_{IN}=0.4$ V
I_L	Input Leakage Current, Open Drain Pins,	—	+10	—	+18	μA	$V_{IN}=13.2$ V
	and Inputs with No Pullup Resistor	—	−5	—	−8		$V_{IN}=0.2$ V
I_{OH}	Output High Current Pins with Standard Pullup Resistor	−100	—	−90	—	μA	$V_{OH}=2.4$ V
I_{OHDD}	Output High Current Direct Drive Pins	−1.5	—	−1.3	—	mA	$V_{OH}=1.5$ V
		—	−8.5	—	−11		$V_{OH}=0.7$ V
I_{OHS}	STROBE Output High Current	−300	—	−270	—	μA	$V_{OH}=2.4$ V
I_{OL}	Output Low Current	1.8	—	1.65	—	mA	$V_{OL}=0.4$ V
I_{OLS}	STROBE Output Low Current	5.0	—	4.5	—	mA	$V_{OL}=0.4$ V
I_{CC}	Power Supply Current	—	85	—	110	mA	Outputs Open
P_D	Power Dissipation	—	400	—	525	mW	Outputs Open

1. $\overline{RESET}$ and EXT INT have internal Schmitt triggers giving minimum 0.2 V hysteresis.
2. Power dissipation for I/O pins is calculated by $\Sigma(V_{CC}-V_{IL})(|I_{IL}|)=\Sigma(V_{CC}-V_{OH})(|I_{OH}|)=\Sigma(V_{OL})(I_{OL})$

TIMER AC CHARACTERISTICS

Definitions:
Error = Indicated time value − actual time value
$t_{psc} = t\phi \times$ Prescale Value

Interval Timer Mode:

Single interval error, free running (Note 3) ..... $\pm 6t\phi$
Cumulative interval error free running (Note 3) 0
Error between two Timer reads (Note 2) ..... $\pm (t_{psc}+t\phi)$
Start Timer to stop Timer error (Notes 1, 4) ..... $+t\phi$ to $-(t_{psc}+t\phi)$
Start Timer to read Timer error (Notes 1, 2) ..... $-5t\phi$ to $-(t_{psc}+7t\phi)$
Start Timer to interrupt request error (Notes 1, 3) ..... $-2t\phi$ to $-8t\phi$
Load Timer to Stop Timer error (Note 1) ..... $+t\phi$ to $-(t_{pcs}+2t\phi)$
Load Timer to read Timer error (Notes 1, 2) ..... $-5t\phi \pm$ to $-(t_{psc}+8t\phi)$
Load Timer to interrupt request error (Notes 1, 3) ..... $-2t\phi$ to $-9t\phi$

Pulse Width Measurement Mode:

Measurement accuracy (Note 4) ..... $+t\phi$ to $-(t_{psc}+2t\phi)$
Minimum pulse width of EXT INT pin ..... $2t\phi$

Event Counter Mode:

Minimum active time of EXT INT pin ..... $2t\phi$
Minimum inactive time of EXT INT pin ..... $2t\phi$

NOTES:
1. All times which entail loading, starting, or stopping the Timer are referenced from the end of the last machine cycle of the OUT or OUTS instruction.
2. All times which entail reading the Timer are referenced from the end of the last machine cycle of the IN or INS instruction.
3. All times which entail the generation of an interrupt request are referenced from the start of the machine cycle in which the appropriate interrupt request latch is set. Additional time may elapse if the interrupt request occurs during a privileged or multicycle instruction.
4. Error may be cumulative if operation is repetitively performed.

FIGURE 2 — $\overline{\text{STROBE}}$ SOURCE CAPABILITY
(TYPCIAL AT V_{CC} = 5 V, T_A = 25°C)

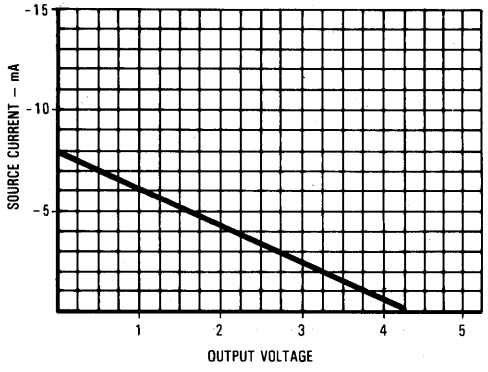
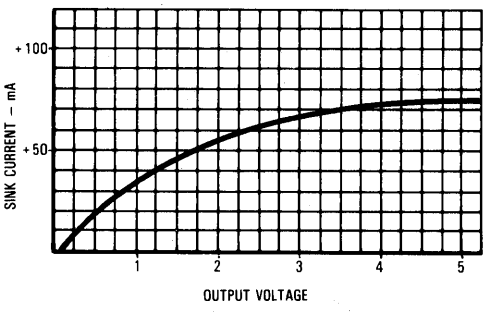


FIGURE 3 — $\overline{\text{STROBE}}$ SINK CAPABILITY
(TYPICAL AT V_{CC} = 5 V, T_A = 25°C)



3

FIGURE 4 — STANDARD I/O PORT SOURCE CAPABILITY
(TYPICAL AT V_{CC} = 5 V, T_A = 25°C)

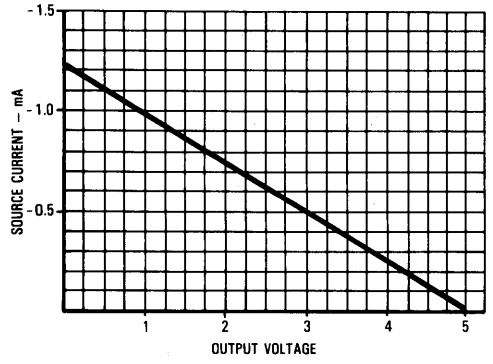


FIGURE 5 — DIRECT DRIVE I/O PORT SOURCE CAPABILITY
(TYPICAL AT V_{CC} = 5 V, T_A = 25°C)

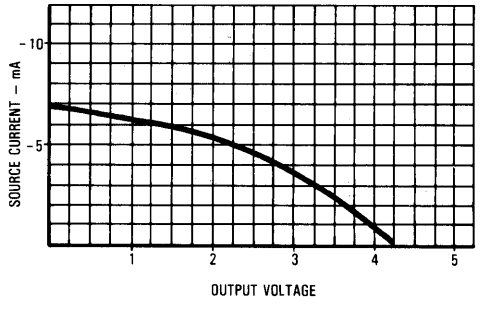


FIGURE 6 — I/O PORT SINK CAPABILITY
(TYPICAL AT V_{CC} = 5 V, T_A = 25°C)

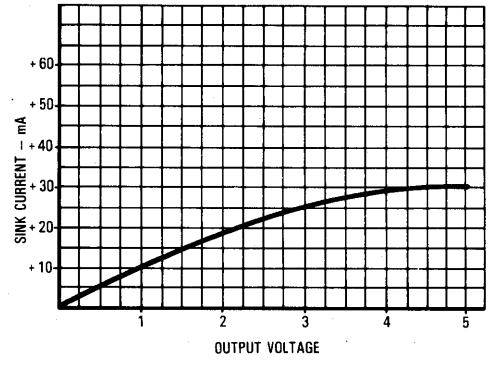


FIGURE 7 — MAXIMUM OPERATING TEMPERATURE
vs I/O POWER DISSIPATION

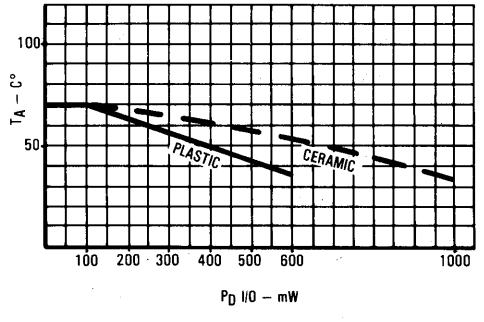


FIGURE 8 — MC3870 I_{DD} vs TEMPERATURE (V_{CC}=5 V)

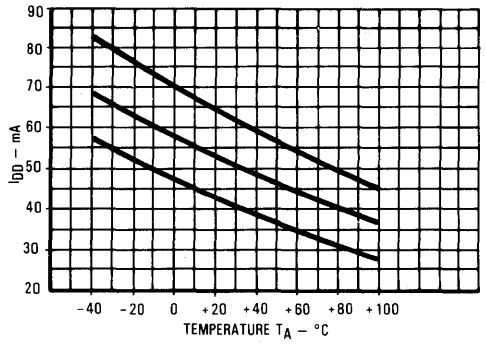
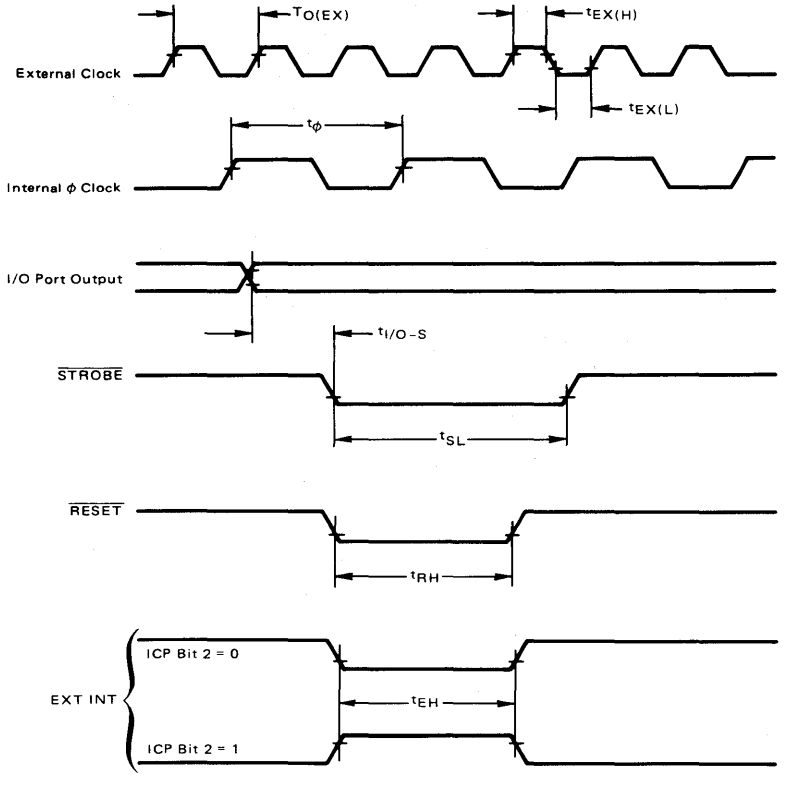
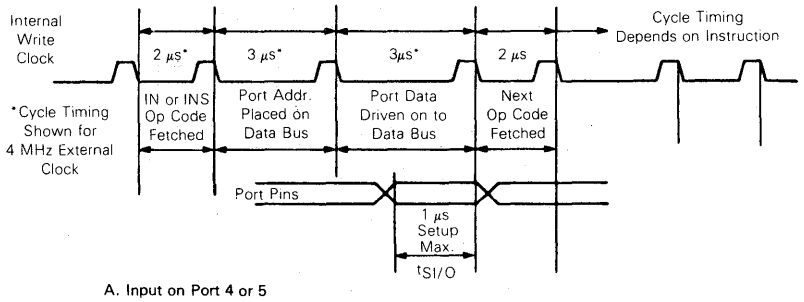


FIGURE 9 — ac TIMING DIAGRAM

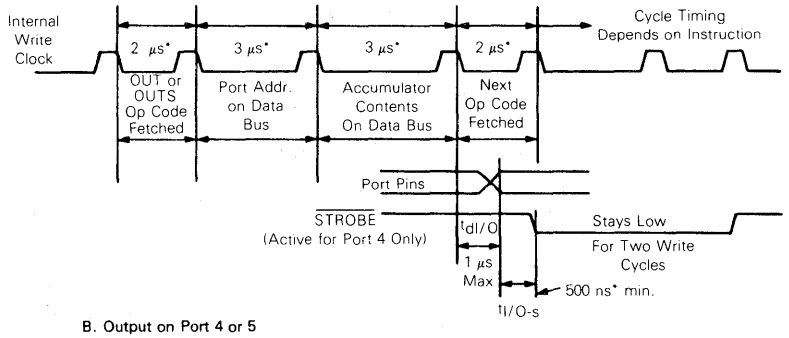


NOTE: All measurements are referenced to V_{IL} max., V_{IH} min., V_{OL} max., or V_{OH} min.

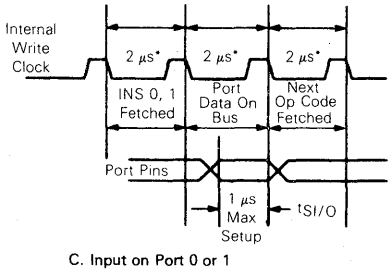
FIGURE 10 — INPUT/OUTPUT ac TIMING



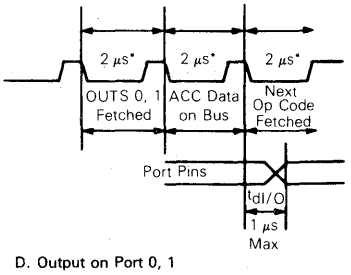
A. Input on Port 4 or 5



B. Output on Port 4 or 5



C. Input on Port 0 or 1



D. Output on Port 0, 1

MC3870 CLOCKS

The time base for the MC3870 may originate from one of four sources. The four configurations are shown in Figure 12. There is an internal 26 pF capacitor between XTL 1 and GND and an internal 26 pF capacitor between XTL 2 and GND. Thus, external capacitors are not necessarily required. In all external clock modes the external time-base frequency is divided by two to form the internal ϕ clock. The external clock frequency is divided by eight during short instruction cycles and is divided by twelve during long instruction cycles as given per instruction in the instruction set towards the end of this data sheet. To get the total instruction cycle time, divide the external clock frequency by eight, invert the number, then multiply by the short number of cycles. Then divide the external clock frequency by twelve and invert the number (Yx) then multiply by the number of long cycles. Add these two numbers to get the number of microseconds per instruction for a given clock frequency.

CRYSTAL SELECTION

The use of a crystal as the time base is highly recommended as the frequency stability and reproducability from system-to-system is unsurpassed. The 3870 has an internal divide-by-two to allow the use of inexpensive and widely available TV Color Burst Crystals (3.58 MHz). The following crystal parameters are suggested for 3870 applications:

a) Parallel Resonance, Fundamental Mode AT-Cut, HC-33/μ holder

b) Frequency Tolerance measured with 18 pF load (0.1% accuracy) – drive level 10 mW

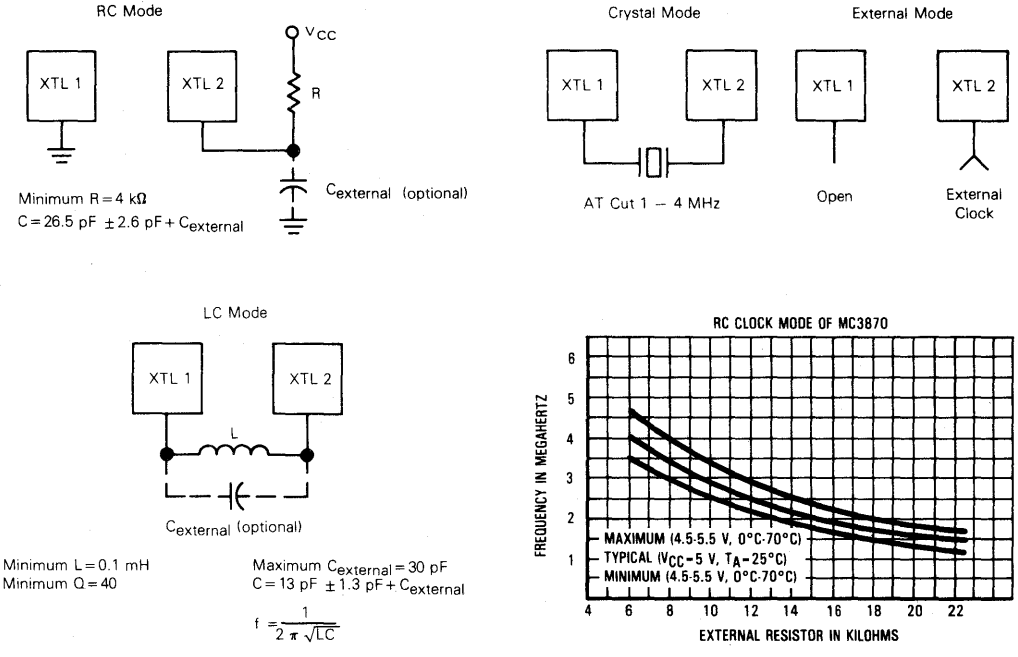
c) Shunt capacitance (Co) = 7 pF max

d) Series resistance (Rs)

f = 1 MHz	Rs = 550 ohms max
f = 2 MHz	Rs = 300 ohms max
f = 3 MHz	Rs = 100 ohms max
f = 3.58 MHz	Rs = 100 ohms max
f = 4 MHz	Rs = 100 ohms max

3

FIGURE 11 – CLOCK CONFIGURATION



FUNCTIONAL PIN DESCRIPTION

$\overline{P0\text{-}0}$ - $\overline{P0\text{-}7}$ AND $\overline{P1\text{-}0}$ - $\overline{P1\text{-}7}$

Ports 1 and 2 are 16 lines which can be individually used as standard TTL-type inputs or latched outputs.

$\overline{P4\text{-}0}$ - $\overline{P4\text{-}7}$ AND $\overline{P5\text{-}0}$ - $\overline{P5\text{-}7}$

Ports 4 and 5 are 16 lines which can be individually used as standard, open drain, or direct drive type latched outputs or inputs. Refer to Figure 14 for more information on port options.

$\overline{STROBE}$

This output, which is normally high, provides a single low pulse after valid data is present on port 4 ($\overline{P4\text{-}0}$ - $\overline{P4\text{-}7}$) during an output instruction.

$\overline{RESET}$

This active low input is used to reset the internal state of the microcomputer. When allowed to go high, program execution begins at $000.

EXT/INT

This input is an external interrupt. Its active state is software programmable. The input is also used in conjunction with the timer for pulse width measurement and event counting.

XTL 1 AND XTL 2

These two inputs interface a crystal (1 to 4 MHz), LC network, RC network, or an external single-phase clock to the microcomputer.

TEST

TEST is an input used only in testing the MC3870. For normal circuit functionality, this pin is left unconnected or may be grounded.

V_{CC}

This is the power supply input (+5 V ±10%).

Pin Name	Description	Type
$\overline{P0\text{-}0}$ - $\overline{P0\text{-}7}$	I/O Port 0	Bidirectional
$\overline{P1\text{-}0}$ - $\overline{P1\text{-}7}$	I/O Port 1	Bidirectional
$\overline{P4\text{-}0}$ - $\overline{P4\text{-}7}$	I/O Port 4	Bidirectional
$\overline{P5\text{-}0}$ - $\overline{P5\text{-}7}$	I/O Port 5	Bidirectional
$\overline{STROBE}$	Ready Strobe	Output
EXT INT	External Interrupt	Input
$\overline{RESET}$	External Reset	Input
TEST	Test Line	Input
XTL 1, XTL 2	Time Base	Input
V_{CC}, GND	Power Supply Lines	Input

MC3870 ARCHITECTURE

This section describes the basic functional elements of the MC3870 as shown in the block diagram of Figure 1. A programming model is shown in Figure 12.

MAIN CONTROL LOGIC

The Instruction Register (IR) receives the operation code (OP code) or the instruction to be executed from the program ROM via the data bus. During all OP code fetches eight bits are latched into the IR. Some instructions are completely specified by the upper four bits of the OP code. In those instructions the lower four bits are an immediate register address or an immediate 4-bit operand. Once latched into the IR the main control logic decodes the instruction and provides the necessary control gating signals to all circuit elements.

ROM ADDRESS REGISTERS

There are four 11-bit registers associated with the 2K × 8 ROM. These are the Program Counter (P0), the Stack Register (P), the Data Counter (DC), and the Auxiliary Data Counter (DC1). The Program Counter is used to address instructions or immediate operands. P is used to save the contents of P0 during an interrupt or subroutine call. Thus, P contains the return address at which processing is to resume upon completion of the subroutine or the interrupt routine.

The Data Counter (DC) is used to address data tables. This register is auto-incrementing. Of the two data counters only DC can access the ROM. However, the XDC instruction allows DC and DC1 to be exchanged.

Associated with the address registers is an 11-bit Adder/Incrementer. This logic element is used to increment P0 or DC when required and is also used to add displacements to P0 on relative branches or to add the data bus contents to DC in the ADC (Add Data Counter) instruction.

2048 × 8 ROM

The microcomputer program and data constants are stored in the program ROM. When a ROM access is required, the appropriate address register (P0 or DC) is gated onto the ROM address bus and the ROM output is gated onto the main data bus. The first byte in the ROM is location zero.

SCRATCHPAD AND IS

The scratchpad provides 64 8-bit registers which may be used as general purpose RAM memory. The Indirect Scratchpad Address Register (IS) is a 6-bit register used to address the 64 registers. All 64 registers may be accessed using IS. In addition, the lower order 12 registers may also be directly addressed.

IS can be visualized as holding two octal digits. This division of IS is important since a number of instructions increment or decrement only the least-significant three bits of IS when referencing scratchpad bytes via IS. This makes it easy to reference a buffer consisting of contiguous scratchpad bytes. For example, when the low order octal digit is incremented or decremented IS is incremented from octal 27 (0'27') to 0'20' or is decremented from 0'20' to 0'27'. This feature of the IS is very useful in many program sequences. All six bits of IS may be loaded at one time or either half may be loaded independently.

Scratchpad registers 9 through 15 (decimal) are given mnemonic names (J, H, K, and Q) because of special linkages between these registers and other registers such as

the Stack Register. These special linkages facilitate the implementation of multi-level interrupts and subroutine nesting. For example, the instruction LR K,P stores the lower eight bits of the Stack Register into register 13 (K lower or KL) and stores the upper three bits of P into register 12 (K upper or KU).

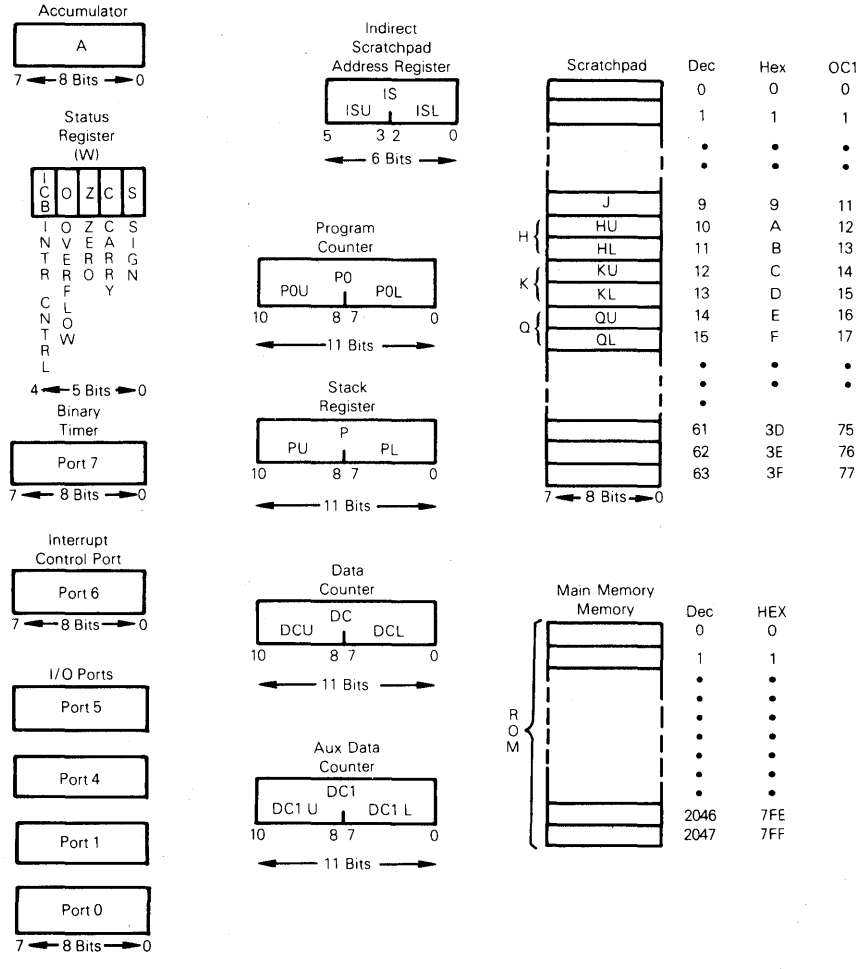
ARITHMETIC AND LOGIC UNIT (ALU)

After receiving commands from the main control logic, the ALU performs the required arithmetic or logic operations (using the data presented on the two input buses) and provides the result on the result bus. The arithmetic operations that can be performed in the ALU are binary add, decimal adjust, add with carry, decrement, and increment. The logic operations that can be performed are AND, OR, EXCLUSIVE OR, "1's" complement, shift right, and shift left. Besides providing the result on the result bus, the ALU also provides four signals representing the status of the result. These signals, stored in the Status Register (W), represent CARRY, OVERFLOW, SIGN, and ZERO condition of the result of the operation.

ACCUMULATOR (A)

The Accumulator (A) is the principal register for data manipulation within the 3870. The A serves as one input to the ALU for arithmetic or logical operations. The result of ALU operations are stored in the A.

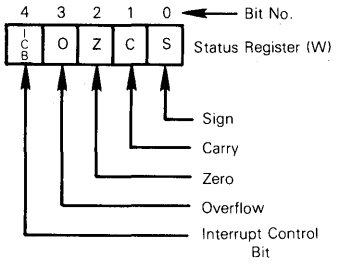
FIGURE 12 — MC3870 PROGRAMMABLE REGISTERS, PORTS, AND MEMORY MAP



THE STATUS REGISTER (W)

The Status Register (also called the W register) holds five status flags as shown in Figure 13.

FIGURE 13 — STATUS REGISTER (W)



Summary of Status Bits

OVERFLOW = $Carry_7 \oplus CARRY_6$

ZERO = $\overline{ALU_7} \wedge \overline{ALU_6} \wedge \overline{ALU_5} \wedge \overline{ALU_4} \wedge \overline{ALU_3} \wedge \overline{ALU_2} \wedge \overline{ALU_1} \wedge \overline{ALU_0}$

CARRY = $CARRY_7$

SIGN = $\overline{ALU_7}$

INTERRUPT CONTROL BIT (ICB)

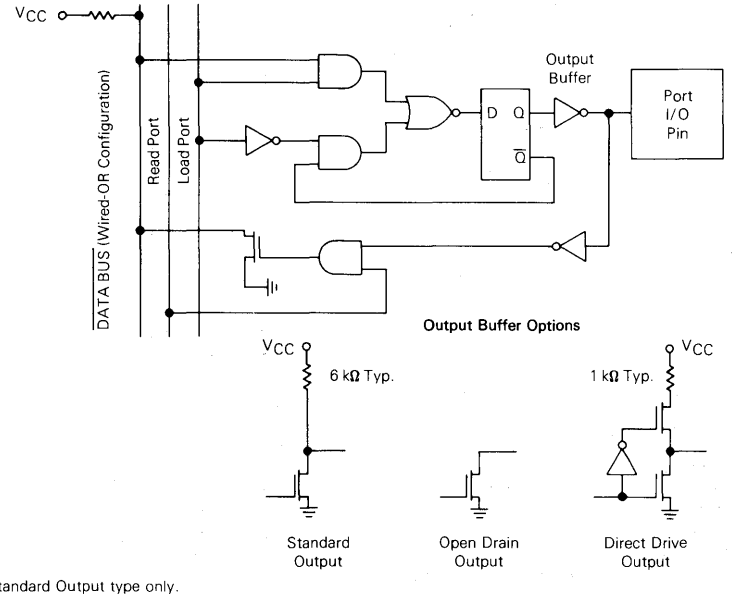
The ICB may be used to allow or disallow interrupts in the MC3870. This bit is not the same as the two interrupt enable bits in the Interrupt Control Port (ICP). If the ICB is set and the MC3870 interrupt logic communicates an interrupt request to the CPU section, the interrupt will be acknowledged and processed upon completion of the first non-privileged instruction. If the ICB is cleared an interrupt request will not be acknowledged or processed until the ICB is set.

I/O PORTS

The MC3870 provides four complete bidirectional Input/Output ports. These are ports 0, 1, 4, and 5. In addition, the Interrupt Control Port is addressed as port 6 and the binary timer is addressed as port 7. An output instruction (OUT or OUTS) causes the contents of A to be latched into the addressed port. An input instruction (IN or INS) transfers the contents of the port to A (port 6 is an exception which is described later). The schematic of an I/O pin and available output drive options are shown in Figure 14.

An output ready strobe is associated with port 4. This flag may be used to signal a peripheral device that the MC3870 has just completed an output of new data to port 4. The strobe provides a single low pulse shortly after the output operation is completely finished, so either edge may be used to signal the peripheral. $\overline{STROBE}$ may also be used as an input strobe simply by doing a dummy output of H '00' strobe to port 4 after completing the input operation.

FIGURE 14 — I/O PIN CONCEPTUAL DIAGRAM WITH OUTPUT BUFFER OPTIONS



Ports 0 and 1 are Standard Output type only.

Ports 4 and 5 may both be any of the three output options (programmable bit-by-bit).

The $\overline{STROBE}$ output is always configured similar to a Direct Drive Output except that it is capable of driving 3 TTL loads.

$\overline{RESET}$ and EXT INT may have standard 6 kΩ (typical) pullup or may have no pullup. These two inputs have Schmitt trigger inputs with a minimum of 0.2 volts of hysteresis.

TIMER AND INTERRUPT CONTROL PORT

The Timer is an 8-bit binary down counter which is software programmable to operate in one of three modes: the Interval Timer Mode, the Pulse Width Measurement Mode, or the Event Counter Mode. As shown in Figure 15, associated with the Timer are an 8-bit register called the interrupt control port, a programmable prescaler, and an 8-bit modulo-N register. A functional logic diagram is shown in Figure 16.

INTERRUPT CONTROL PORT (PORT 6)

The desired timer mode, prescale value, starting and stopping the timer, active level of the EXT INT pin, and local enabling or disabling of interrupts are selected by outputting the proper bit configuration from the Accumulator to the Interrupt Control Port (port 6) with an OUT or OUTS instruction. Bits within the Interrupt Control Port are defined as follows:

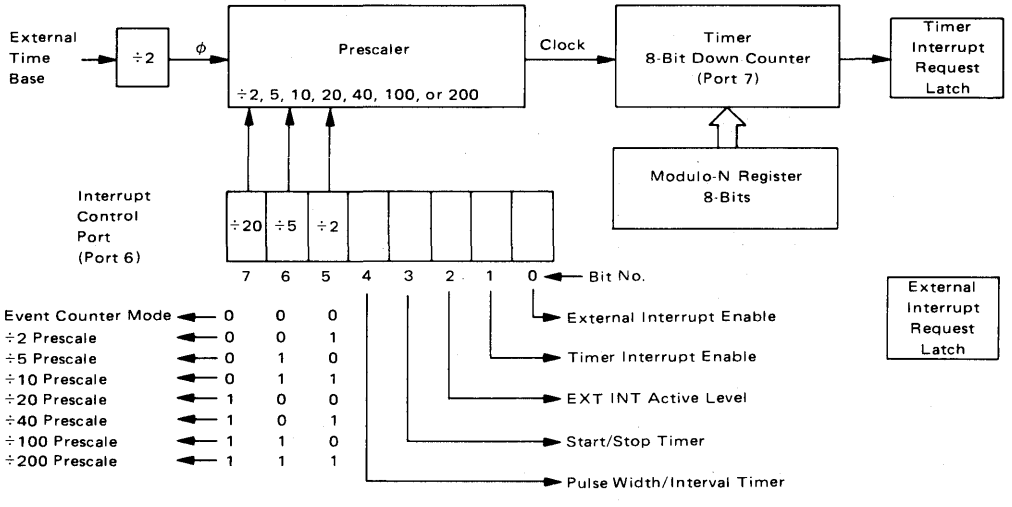
Bit 0 — External Interrupt Enable
Bit 1 — Timer Interrupt Enable
Bit 2 — EXT INT Active Level
Bit 3 — Start/Stop Timer
Bit 4 — Pulse Width/Interval Timer
Bit 5 — ÷2 Prescale
Bit 6 — ÷5 Prescale
Bit 7 — ÷20 Prescale

A special situation exists when reading the Interrupt Control Port (with IN or INS instruction). The Accumulator is *not* loaded with the content of the ICP; instead, Accumulator bits 0 through 6 are loaded with "0's" while bit 7 is loaded with the logic level being applied to the EXT INT pin, thus allowing the status of EXT INT to be determined without the necessity of servicing an external interrupt request. When reading the Interrupt Control Port (port 6) bit 7 of the Accumulator is loaded with the actual logic level being applied to the EXT INT pin, regardless of the status of ICP bit 2 (the EXT INT Active Level bit); that is, if EXT INT is a *+5 V bit 7* of the Accumulator is set to a logic "1", but if EXT INT is at GND then Accumulator bit 7 is reset to logic "0". This capability is useful in establishing a high speed polled handshake procedure or for using EXT INT as an extra input pin if external interrupts are not required and the Timer is used only in the Interval Timer Mode. However, if it is desirable to read the contents of the ICP then one of the 64 scratchpad registers or one byte of RAM may be used to save a copy of whatever is written to the ICP.

The rate at which the timer is clocked in the External Timer Mode is determined by the frequency of an internal ϕ clock and by the division value selected for the prescaler. (The internal ϕ clock operates at one-half the external time-base frequency). If ICP bit 5 is set and bits 6 and 7 are cleared, the prescaler divides ϕ by 2. Likewise, if bit 6 or 7 is individually set, the prescaler divides ϕ by 5 or 20 respectively. Combinations of bits 5, 6, and 7 may also be selected. For example, if bits 5 and 7 are set while 6 is cleared the prescaler will divide by 40. Thus, possible prescaler values are ÷2, ÷5, ÷10, ÷20, ÷40, ÷100, and ÷200.

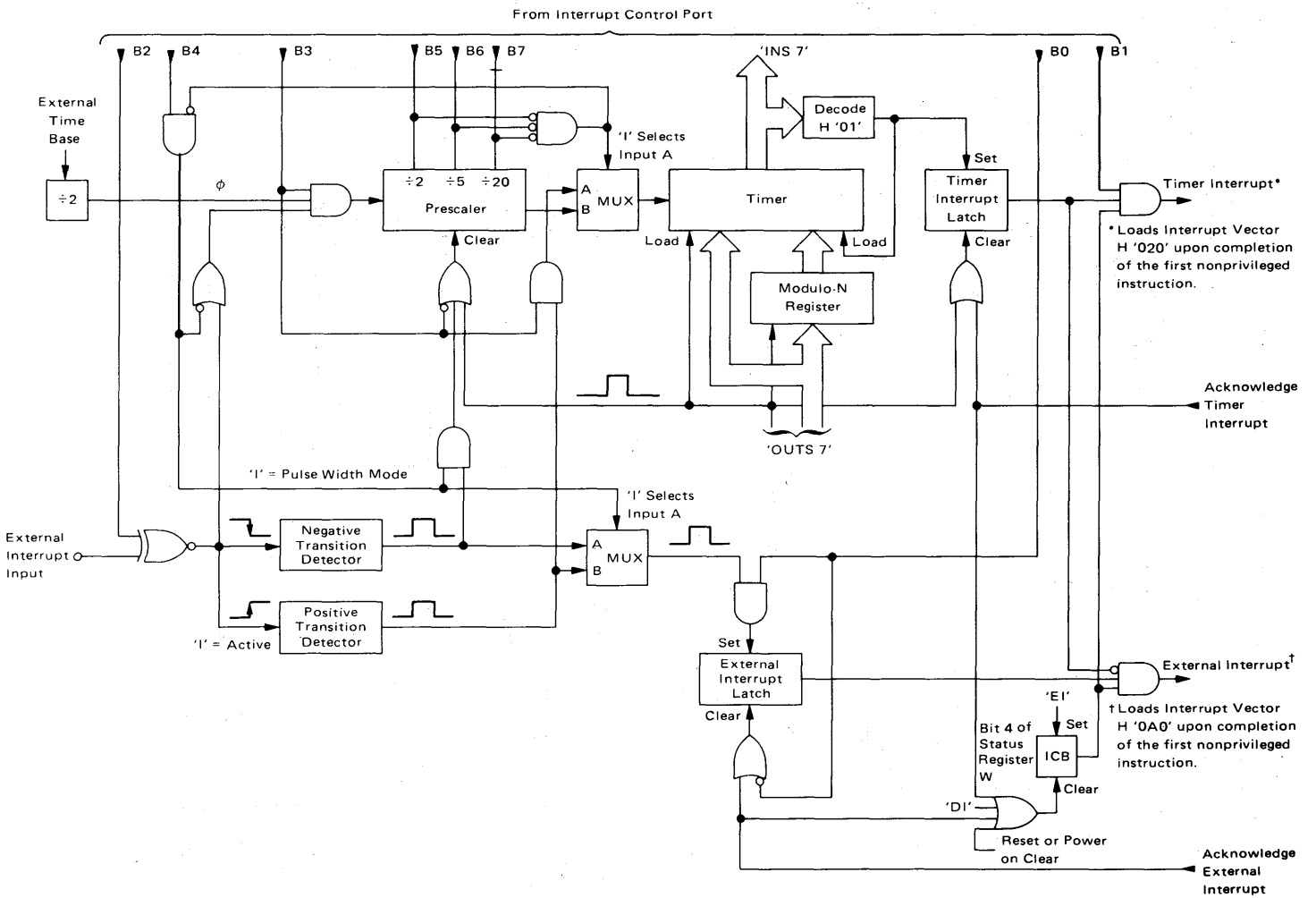
Any of three conditions will cause the prescaler to be reset: whenever the timer is stopped by clearing ICP bit 3, execution of an output instruction to Port 7, (the timer is assigned port address 7), or on the trailing edge transition of the EXT INT pin when in the Pulse Width Measurement Mode. These last two conditions are explained in more detail below.

FIGURE 15 — TIMER AND CONTROL PORT BLOCK DIAGRAM



NOTE: See Figure 17 for a more detailed functional diagram.

FIGURE 16 — MC3870 TIMER/INTERRUPT FUNCTIONAL DIAGRAM



An OUT or OUTS instruction to Port 7 will load the content of the Accumulator to both the Timer and the 8-bit modulo-N register, reset the prescaler, and clear any previously stored timer interrupt request. As previously noted, the Timer is an 8-bit down counter which is clocked by the prescaler in the Interval Timer mode and in the Pulse Width Measurement Mode. The prescaler is not used in the Event Counter Mode. The modulo-N register is a buffer whose function is to save the value which was most recently outputted to Port 7. The modulo-N register is used in all three timer modes.

Interval Timer Mode – When ICP bit 4 is cleared (logic 0) and at least one prescale bit is set, the Timer operates in the Interval Timer Mode. when bit 3 of the ICP is set the Timer will start counting down from the modulo-N value. After counting down to H '01', the Timer returns to the modulo-N value at the next count. On the transition from H '01' to H 'N' the Timer sets a timer interrut request latch. Note that the interrupt request latch is set by the transition to H 'N' and not be the presence of H 'N' in the timer, thus allowing a full 256 counts if the modulo-N register is preset to H '00'. If bit 1 of the ICP is set, the interrupt request is passed on to the CPU section of the MC3870. However, if bit 1 of the ICP is a logic 0 the interrupt request is not passed on to the CPU section but the interrupt request latch remains set. If ICP bit 1 is subsequently set, the interrupt request will then be passed on to the CPU section. (Recall from the discussion of the Status Register's Interrupt Control Bit that the interrupt request will be acknowledged by the CPU section only if ICB is set.) Only two events can reset the timer interrupt request latch: when the timer interrupt request latch is acknowledged by the CPU section, or when a new load of the modulo-N register is performed.

Consider an example in which the modulo-N register is loaded with H '64' (decimal 100). The timer interrupt request latch will be set at the 100th count following the timer start and the timer interrupt request latch will repeatedly be set on precise 100 counter intervals. If the prescaler is set at ÷40 the timer interrupt request latch will be set every 4000 ϕ clock periods. For a 2 MHz ϕ clock (4 MHz time-base frequency) this will produce 2 millisecond intervals.

The range of possible intervals is from 2 to 51,200 ϕ clock periods (1 μs to 25.6 ms for a 2 MHz clock). However, approximately 50 ϕ periods is a practical minimum because the time between setting the interrupt request latch and the execution of the first instruction of the interrupt service routine is at least 29 ϕ periods (the response time is dependent upon how many privileged instructions are encountered when the request occurs); 29 is based on the timer interrupt occuring at the beginning of a non-privileged short instruction. To establish time intervals greater than 51,200 ϕ clock periods is a simple matter of using the timer interrupt service routine to count the number of interrupts, saving the result in one or more of the scratchpad registers until the desired interval is achieved. With this technique virtually any time interval, or several time intervals, may be generated.

The Timer may be read at any time and in any mode using an input instruction (IN 7 or INS 7) and may take place "on the fly" without interfering with normal timer operation. Also, the Timer may be stopped at any time by clearing bit 3 of the ICP. The Timer will hold its current contents indefinitely and will resume counting when bit 3 is again set. Recall however that the prescaler is reset whenever the Timer is stopped; thus a series of starting and stopping will result in a cumulative truncation error.

A summary of other timer errors is given in the timing section of their specification. For a free running timer in the Interval Timer Mode the time interval between any two interrupt requests may be in error by $\pm 6\,\phi$ clock periods although the cumulative error over many intervals is zero. The prescaler and Timer generate precise intervals for setting the timer interrupt request latch but the time out may occur at any time within a machine cycle. (There are two types of machine cycles; short cycles which consist of 4 ϕ clock periods and long cycles which consist of 6 ϕ clock periods.) Interrupt requests are synchronized with the internal machine clock thus, giving rise to the possible $\pm 6\,\phi$ error. Additional errors may arise due to the interrupt request occuring while a privileged instruction or multicycle instruction is being executed. Nevertheless, for most applications all of the above errors are neglibible, especially if the desired time interval is greater than 1 ms.

Pulse Width Measurement Mode – When ICP bit 4 is set (logic 1) and at least one prescale bit is set, the Timer operates in the Pulse Width Measurement Mode. This mode is used for accurately measuring the duration of a pulse applied to the EXT INT pin. The Timer is stopped and the prescaler is reset whenever EXT INT is at its inactive level. The active level of EXT INT is defined by ICP bit 2; if cleared, EXT INT is active low; if set, EXT INT is active high. If ICP bit 3 is set, the prescaler and Timer will start counting when EXT INT transitions to the active level. When EXT INT returns to the inactive level the Timer then stops, the prescaler resets, and *if ICP bit 0* is set an external interrupt request latch is set. (Unlike timer interrupts, external interrupts are not latched if the ICP Interrupt Enable bit is not set.)

As in the Interval Timer Mode, the Timer may be read at any time, may be stopped at any time by clearing ICP bit 3, the prescaler and ICP bit 1 function as previously described, and the Timer still functions as an 8-bit binary down counter with the timer interrupt request latch being set on the Timer's transition from H '01' to H 'N'. Note that the EXT INT pin has nothing to do with loading the Timer; its action is that of automatically starting and stopping the Timer and of generating external interrupts. Pulse widths longer than the prescale value times the modulo-N value are easily measured by using the timer interrupt service routine to store the number of timer interrupts in one or more scratchpad registers.

As for accuracy, the actual pulse duration is typically slightly longer than the measured value because the status of the prescaler is not readable and is reset when the Timer is stopped. Thus, for maximum accuracy, it is advisable to use a small division setting for the prescaler.

Event Counter Mode – When ICP bit 4 is cleared and all prescale bits (ICP bits 5, 6, and 7) are cleared, the Timer operates in the Event Counter Mode. This mode is used for counting pulses applied to the EXT INT pin. If ICP bit 3 is set, the Timer will decrement on each transition from the inactive level to the active level of the EXT INT pin. The prescaler is not used in this mode; but as in the other two timer modes,

the Timer may be read at any time, may be stopped at any time by clearing ICP bit 3, ICP bit 1 functions previously described, and the timer interrupt request latch is set on the Timer's transition from H '01' to H 'N'.

Normally ICP bit 0 should be kept cleared in the Event Counter Mode; otherwise, external interrupts will be generated on the transition from the inactive level to the active level of the EXT INT pin.

For the Event Counter Mode, the minimum pulse width required on EXT INT is 2 ϕ clock periods and the minimum inactive time is 2 ϕ clock periods; therefore, the maximum repetition rate is 500 kHz.

External Interrupts – When the timer is in the Interval Timer Mode the EXT INT pin is available for non-timer related interrupts. If ICP bit 0 is set, an external interrupt request latch is set when there is a transition from the inactive level to the active level of EXT INT. (EXT INT is an edge-triggered input.) The interrupt request is latched until either acknowledged by the CPU section or until ICP bit 0 is cleared (unlike timer interrupt requests which remain latched even when ICP bit 1 is cleared). External interrupts are handled in the same fashion when the Timer is in the Pulse Width Measurement Mode or in the Event Counter Mode, except that only in the Pulse Width Measurement Mode the external interrupt request latch is set on the trailing edge of EXT INT; that is, on the transition from the active level to the inactive level.

INTERRUPT HANDLING

When either a timer or an external interrupt request is communicated to the CPU section of the MC3870, it will be acknowledged and processed at the completion of the first non-privileged instruction if the Interrupt Control Bit of the Status Register is set. If the Interrupt Control Bit is not set, the interrupt request will continue until either the Interrupt Control Bit is set and the CPU section acknowledges the interrupt or until the interrupt request is cleared as previously described.

If there is both a timer interrupt request and an external interrupt request when the CPU section starts to process the requests, the timer interrupt is handled first.

When an interrupt is allowed the CPU section will request that the interrupting element pass its interrupt vector address to the Program Counter via the data bus. The vector address for a timer interrupt is H '020'. The vector address for external interrupts is H '0A0'. After the vector address is passed to the Program Counter, the CPU section sends an acknowledge signal to the appropriate interrupt request latch which clears that latch. The execution of the interrupt service routine will then commence. The return address of the original program is automatically saved in the Stack Register, P.

The Interrupt Control Bit of W (Status Register) is automatically reset when an interrupt request is acknowledged. It is then the programmer's responsibility to determine when ICB will again be set (by executing an EI instruction). This action prevents an interrupt service routine from being interrupted unless the programmer so desires.

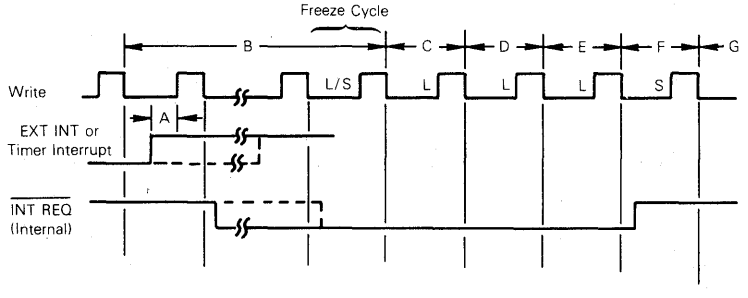
Figure 17 details the interrupt sequence which occurs whether the interrupt request is from an external source via EXT INT or from the MC3870's internal timer. Events are labeled with the letters A through G and are described below.

Event A – An interrupt request must satisfy a hold time requirement as specified in the AC Characteristics in order to guarantee that it is valid on the rising edge of the WRITE clock.

Event B – Event B represents the instruction being executed when the interrupt occurs. The last cycle of B is normally the instruction fetch for the next cycle. However, if B is not a privileged instruction and the CPU's Interrupt Control Bit is set, then the last cycle becomes a "freeze" cycle rather than a fetch. At the end of the freeze cycle the interrupt request latches are inhibited from altering the interrupt daisy-chain so that sufficient time will be allowed for the daisy-chain to settle. (If B is a privileged instruction, the instruction fetch is not replaced by a freeze cycle; instead, the fetch is performed and the next instruction is executed. Although unlikely to be encountered, a series of privileged instructions will be sequentially executed without interrupt. One more instruction, called a 'protected' instruction, will always be executed after the last privileged instruction. The last cycle of the protected instruction then performs the freeze.)

The dashed lines on EXT INT illustrate the last opportunity for EXT INT to cause the last cycle of a non-protected instruction to become a freeze cycle.

FIGURE 17 – INTERRUPT SEQUENCE



The freeze cycle is a short cycle (4 ϕ clock periods) in all cases except where B is the Decrement Scratchpad instruction, in which case the freeze cycle is a long cycle (6 ϕ clock periods).

INT REQ goes low on the next negative edge of WRITE if both PRI IN is low and the appropriate interrupt enable bit of the Interrupt Control Port is set. Both INT REQ and WRITE are internal signals.

Event C – A NO-OP long cycle to allow time for the internal priority chain to settle.

Event D – The Program Counter (PO) is pushed to the stack register (P) in order to save the return address. The interrupt circuitry places the lower 8 bits of the interrupt vector address onto the data bus. This is always a long cycle.

Event E – A long cycle in which the interrupt circuitry places the upper 8 bits of the interrupt vector address onto the data bus.

Event F – A short cycle in which the interrupting interrupt request latch is cleared. Also, the CPU's Interrupt Control Bit is cleared, thus disabling interrupts until an EI instruction is performed. The fetch of the next instruction from the interrupt address.

Event G – Begin execution of the first instruction of the interrupt service routine.

SUMMARY OF INTERRUPT SEQUENCE

For the MC3870 the interrupt response time is defined as the time elapsed between the occurrence of EXT INT going active (or the Timer transitioning to H 'N') and the beginning of execution of the first instruction of the interrupt service routine. The interrupt response time is a variable dependent upon what the microprocessor is doing when the interrupt request occurs. As shown in Figure 17, the minimum interrupt response time is 3 long cycles plus 2 short cycles plus one WRITE clock pulse width plus a setup time of EXT INT prior to the leading edge of the WRITE pulse – a total of 27 ϕ clock periods plus the setup time. At a 2 MHz ϕ this is 14.25 μs. Although the maximum could theoretically be infinite, a practical maximum is 35 μs (based on the interrupt request occurring near the beginning of a PI and LR K, P sequence).

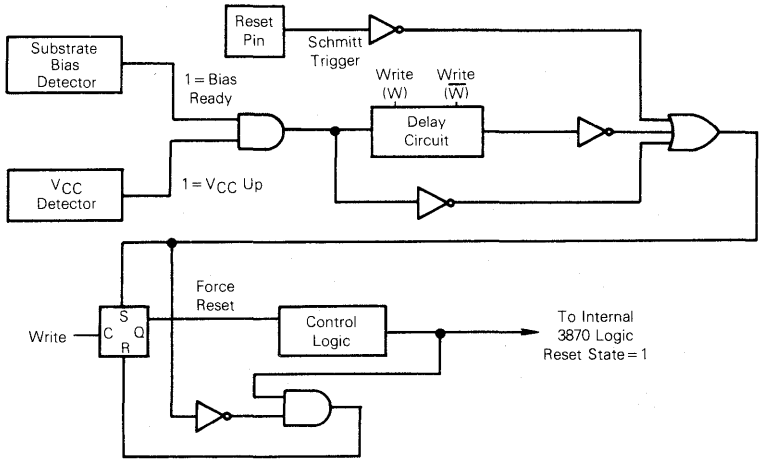
POWER-ON RESET

The intent of the Power-On Reset circuitry on the MC3870 is to automatically reset the device following a typical power-up situation, thus saving external reset circuitry in many applications. This circuitry is not guaranteed to sense a "Brown Out" (low voltage) condition nor is it guaranteed to operate under all possible power-on situations.

Three conditions are required before the MC3870 will leave the reset state and begin operation. Refer to Figure 18 as an aid to the following descriptions. The On-Chip V_{CC} detector senses a minimum value of V_{CC} before it will allow the MC3870 to operate. The threshold of this detector is set by analog circuitry because a stable voltage reference is not available with n-channel MOS processing. Processing variations will cause this threshold to vary from a low of 3.0 volts to a high of 4.3 volts with 3.5 volts being typical.

The MC3870 uses a substrate bias as a technique to provide improved performances versus power consumption relative to conventional grounded substrate approaches. This bias generator may start operating as low as $V_{CC}=3$ volts on some devices while others may require $V_{CC}=4$ volts in order to get adequate substrate bias. Until the substrate reaches the proper bias, the MC3870 will not be released from the reset state. The final condition required is that the clocks of the MC3870 must be functioning. Typically the clocks will start to function at V_{CC} equal to 3 to 3.5 volts but since the part is tested at 4.5 volts, Motorola can not guarantee any operation below 4.5 volts. The output of the delay circuit in Figure 18 will stay low until the clocks

FIGURE 18 – POWER ON RESET BLOCK DIAGRAM



start to function. If the input to the delay circuit is high, typically after 100 cycles of the WRITE clock (800 cycles of the external clock) the output of the delay circuit will go high allowing the MC3870 to begin execution.

If V_{CC} falls to ground for at least a few hundred nanoseconds the output of the delay circuit will go low immediately and the MC3870 will reset.

The internal logic may detect a valid V_{CC}, bias and clocks at $V_{CC} = 3.5$ volts and allow the MC3870 to start executing after the time delay. With a slowly rising power supply, the part may start running before V_{CC} is above 4.5 volts which is below the guaranteed voltage range. When power-on-clear is required with a slowly rising power supply, an external capacitor must be used on the $\overline{\text{RESET}}$ pin to hold it below 0.8 volts until V_{CC} is stable above 4.5 volts. (Note: The option to disconnect the internal pullup resistor on $\overline{\text{RESET}}$ is available which allows the use of a larger external pullup resistor and a small capacitor on $\overline{\text{RESET}}$.)

In many applications it is desirable if the unit does an automatic power-on-clear, but not mandatory. The unit will have a RESET push button and if the unit does not power-up correctly or malfunctions because of some disturbance on the V_{CC} line, the operator will simply press RESET and restore normal operation. It is for these applications that the internal power-on-clear circuitry was designed.

In some applications it is required that the microcomputer continue to run properly without operator intervention after brown-outs, power line disturbances, electrical noise, computer malfunction due to a programming bug, or any other disturbance except a catastrophic failure of some component.

One concept used to keep computers running is that of the "WATCHDOG TIMER". The computer is programmed to periodically reset the watchdog timer during the normal execution of its program (this is easily done in the MC3870 as its normal application is in some control function which is typically periodic). As long as the computer continues to execute its program the watchdog timer is continually reset and never times out. Should the computer stop executing its program for whatever reason, the watchdog timer will time out producing a RESET pulse to the CPU re-starting execution. This is a very positive way to assure that the computer is doing its job, i.e., executing the program. It is important that the software driving the watchdog timer test as many functional blocks (timer, ALU, scratchpad RAM, and ports) of the MC3870 as possible before resetting the watchdog timer. This is because operation of the MC3870, with an out of specification power supply, may allow some of the functions to operate correctly while other functions are not operable.

Motorola can guarantee correct operation of the MC3870 only while the V_{CC} voltage remains within its specified limits. If proper operation of the MC3870 must be guaranteed after a disturbance on the V_{CC} line, then an external circuit must be used to monitor the V_{CC} line and produce $\overline{\text{RESET}}$ to the MC3870 whenever V_{CC} is out of the specified limits.

A related characteristic to power-on-clear is the startup time of the basic timing element. The LC and RC oscillators begin to function almost immediately once V_{CC} is high enough to allow the on-board oscillator to operate ($V_{CC} = 3.5$ V). Operation with a crystal is partly mechanical and some start time is required to get the mass of the crystal into vibrational motion. This time is basically dependent on the frequency (mass) of the crystal. 4 MHz crystals typically require about 2-3 ms to start while 1 MHz crystals require 60-70 ms to start oscillating. Of course, this time may vary greatly from crystal to crystal and is also a function of the power supply rise time characteristic, however, the high-frequency crystals start faster and are definitely recommended (i.e., 3-4 MHz).

The condition of the port pins during the power-in-clear sequence is often asked. The port pins or the STROBE line cannot be specified until V_{CC} reaches 4.5 V and the MC3870 enters the RESET state. Before this, the port pins may stay at V_{SS}, may track V_{CC} as it rises, or they may track V_{CC} part way up then return to V_{SS} (ports 4 and 5 will go to V_{CC} once the clocks are running and the MC3870 has sufficient V_{CC} to properly operate the internal control logic and I/O ports).

EXTERNAL RESET

When $\overline{\text{RESET}}$ is taken low, the content of the Program Counter is pushed to the Stack Register and then the Program Counter and the ICB bit of the W Status Register are cleared. The original Stack Register content is lost. Ports 4, 5, 6, and 7 are loaded with H '00'. The contents of all other registers and ports are unchanged or undefined. When RESET is taken high, the first program instruction is fetched from ROM location H '000'. When an external reset of the MC3870 occurs, P0 is pushed into P and the old contents of P are lost. It must be noted that an external reset is recognized at the start of a machine cycle and not necessarily at the end of an instruction. Thus, if the MC3870 is executing a multi-cycle instruction, that instruction is not completed and the contents of P upon reset may not necessarily be the address of the instruction that would have been executed next. It may, for example, point to an immediate operand if the reset occurred during the second cycle of an LI or CI instruction. Additionally, several instructions (JMP, PI, PI, LR, P0, Q) as well as the interrupt acknowledge sequence modify P0 in parts. That is, they alter P0 by first loading one part then the other and the entire operation takes more than one cycle. Should reset occur during this modification process the value pushed into P will be part of the old P0 (the as yet unmodified part) and part of the new P0 (already modified part). Thus, care should be taken (perhaps by external gating) to insure that reset does not occur at an undesirable time if any significance is to be given to the contents of P after a reset occurs.

V_{CC} DECOUPLING

The MC3870 family devices have dynamic circuitry internally which requires a good high frequency decoupling capacitor to surpress noise on the V_{CC} line. A 0.01 μF or 0.1 μF ceramic capacitor should be placed between V_{CC} and ground, located physically close to the MC3870 device. This will reduce noise generated by the MC3870 to about 70-100 mV on the V_{CC} line.

TEST LOGIC

Special test logic is implemented to allow access to the internal main data bus for test purposes.

In normal operation, the TEST pin is unconnected or is connected to GND. When TEST is placed at a TTL level (2.0 V to 2.6 V) port 4 becomes an output of the internal data

bus. The data appearing on the port 4 pins is logically true whereas input data forced on port 5 must be logically false. When TEST is placed at a high level (6.0 V to 7.0 V), the ports act as above and additionally the 2K × 8 program ROM is prevented from driving the data bus. In this mode, operands and instructions may be forced externally through port 5 instead of being accessed from the program ROM. When TEST is in either the TTL state or the high state, STROBE ceases its normal function and becomes a machine cycle clock (identical to the F8 multi-chip system WRITE clock except inverted).

Timing complexities render the capabilities associated with the TEST pin impractical for use in a user's application, but these capabilities are thoroughly sufficient to provide a rapid method for thoroughly testing the MC3870.

SUPPLEMENTARY NOTES

The Interrupt Control Bit of the W Status Register is automatically reset when an interrupt request is acknowledged. It is then the programmer's responsibility to determine when ICB will again be set (by execution an EI instruction). This action prevents an interrupt service routine from being interrupted unless the programmer so desires.

When reading the Interrupt Control Port (port 6), bit 7 of the Accumulator is loaded with the actual logic level being applied to the EXT INT pin, regardless of the status of ICP bit 2 (the EXT INT Active Level bit). This is, if EXT INT is at +5 V, bit 7 of the Accumulator is set to a logic "1"; but, if EXT INT is at GND, then Accumulator bit 7 is reset to logic "0".

In the MC3870 (F8 COMPATIBLE) INSTRUCTION SET summary, the number of cycles shown are "nominal" machine cycles. A nominal machine cycle is defined as 4 ϕ clock periods, thus, requiring 2 μs for a 2 MHz ϕ clock frequency (4 MHz external time-base frequency).

Also, the summary uses an older nomenclature for register names. The translation is as follows:

PC0 = P0	Program Counter
PC1 = P	Stack Register
DC0 = DC	Data Counter
DC1 = DC1	Auxiliary Data Counter

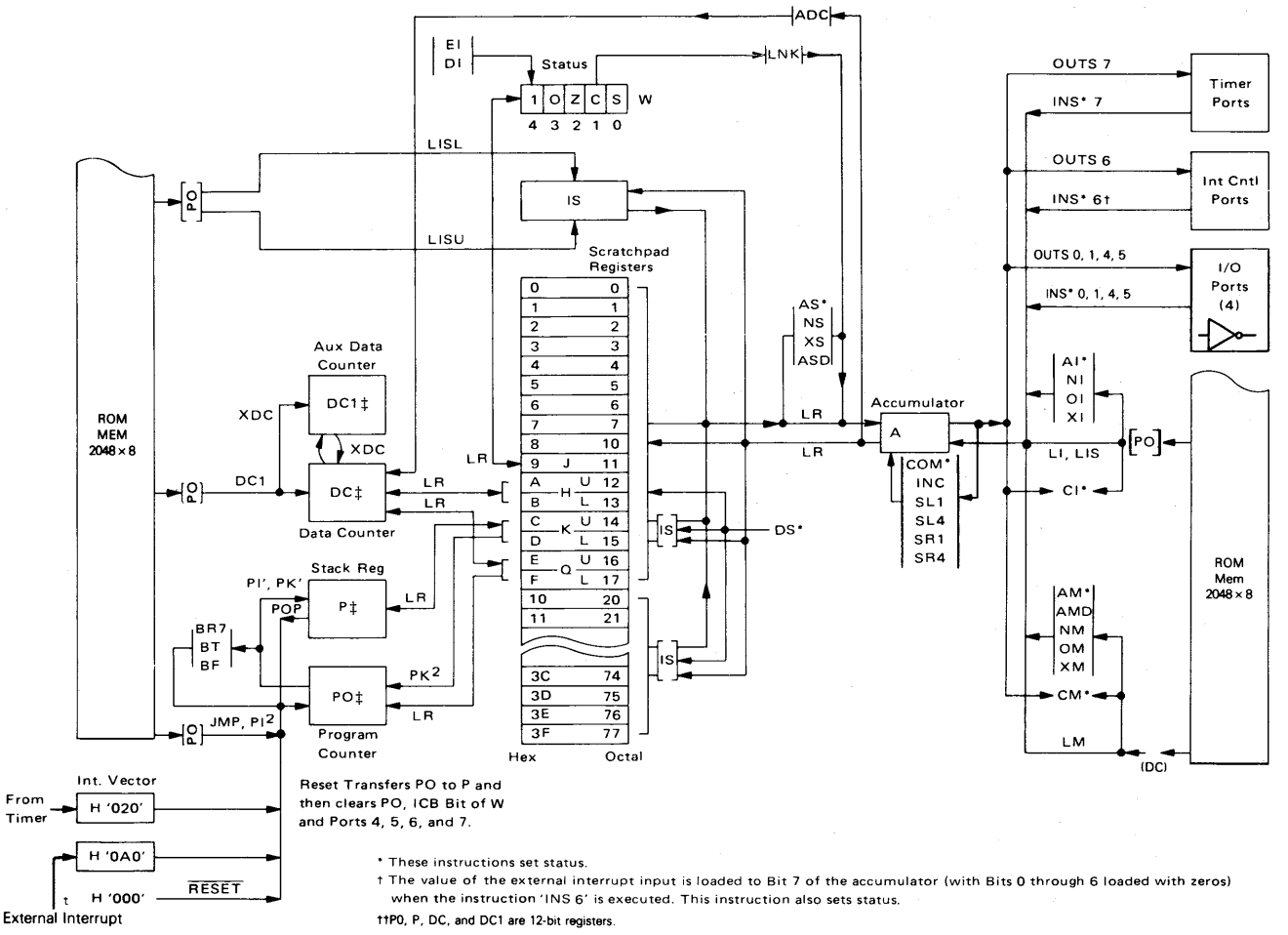
The nomenclature is used in order to be consistent with the assembly language mnemonics.

For the MC3870, execution of an INS or OUTS instruction requires 2 machine cycles for ports 0 and 1, whereas ports 4 and 5 require 4 machine cycles.

INSTRUCTION EXECUTION

This section details the timing and execution of the MC3870 instruction set. Refer to Figure 19 for a MC3870 Programming Model.

FIGURE 19 — MC3870 PROGRAMMING MODEL



* These instructions set status.

† The value of the external interrupt input is loaded to Bit 7 of the accumulator (with Bits 0 through 6 loaded with zeros) when the instruction 'INS 6' is executed. This instruction also sets status.

††P0, P, DC, and DC1 are 12-bit registers.

NOTE: The Instructions PI and PK are shown in two sequential parts. (PI^1, PI^2 and PK^1, PK^2).

MC3870 INSTRUCTION SET

ACCUMULATOR GROUP INSTRUCTIONS

Operation	Mnemonic Op Code	Operand	Function	Machine Code	Bytes	Cycles Short	Cycles Long	(2 MHzφ)	Status Bits OVR	Status Bits ZERO	Status Bits CRY	Status Bits SIGN
Add Carry	LNK		A←(A)+CRY	19	1	1		2	1/0	1/0	1/0	1/0
Add Immediate	AI	ii	A←(A)+H'ii'	24ii	2	1	1	5	1/0	1/0	1/0	1/0
And Immediate	NI	ii	A←(A)ΔH'ii'	21ii	2	1	1	5	0	1/0	0	1/0
Clear	CLR		A←H'00'	70	1	1		2	–	–	–	–
Compare Immediate	CI	ii	H'ii' +(A)+1	25ii	2	1	1	5	1/0	1/0	1/0	1/0
Complement	COM		A←(A)+H'FF'	18	1	1		2	0	1/0	0	1/0
Exclusive or Immediate	XI	ii	A←(A)+H'ii'	23ii	2	1	1	5	0	1/0	0	1/0
Increment	INC		A←(A)+1	1F	1	1		2	1/0	1/0	1/0	1/0
Load Immediate	LI	ii	A←H'ii'	20ii	2	1	1	5	–	–	–	–
Load Immediate Short	LIS	i	A←H'0i'	7i	1	1		2				
OR Immediate	OI	ii	A←(A)vH'ii'	22ii	2	1	1	5	0	1/0	0	1/0
Shift Left One	SL	1	Shift Left 1	13	1	1		2	0	1/0	0	1/0
Shift Left Four	SL	4	Shift Left 4	15	1	1		2	0	1/0	0	1/0
Shift Right One	SR	1	Shift Right 1	12	1	1		2	0	1/0	0	1
Shift Right Four	SR	4	Shift Right 4	14	1	1		2	0	1/0	0	1

BRANCH INSTRUCTIONS In all conditional branches P0←(P0)+2 if the test condition is not met. Execution is complete in 3 short cycles.

Operation	Mnemonic Op Code	Operand	Function	Machine Code	Bytes	Cycles Short	Cycles Long	(2 MHz φ)	Status Bits OVR	Status Bits ZERO	Status Bits CRY	Status Bits SIGN
Branch on Carry	BC	aa	P0←(P0)+1+H'aa' if CRY=1	82aa	2	2	1	7	–	–	–	–
Branch on Positive	BP	aa	P0←(P0)+1+H'aa' if SIGN=1	81aa	2	2	1	7	–	–	–	–
Branch on Zero	BZ	aa	P0←(P0)+1+'Haa' if Zero=1	84aa	2	2	1	7	–	–	–	–
Branch on True	BT	taa	P0←(P0)+1+'Haa' if any test is true	8taa	2	2	1	7	–	–	–	–

TEST CONDITION

2^2	2	2^0
ZERO	CRY	SIGN

Branch if Negative	BM	aa	P0←(P0)+1+H'aa' if SIGN=0	91aa	2	2	1	7	–	–	–	–
Branch if No Carry	BNC	aa	P0←(P0)+1+H'aa' if CARRY=0	92aa	2	2	1	7	–	–	–	–
Branch if No Overflow	BNO	aa	P0←(P0)+1+H'aa' if OVR=0	98aa	2	2	1	7	–	–	–	–
Branch if Not Zero	BNZ	aa	P0←(P0)+1+H'aa' if ZERO=0	94aa	2	2	1	7	–	–	–	–
Branch if False Test	BF	taa	P0←(P0)+1+H'aa' if all false test bits	9taa	2	2	1	7	–	–	–	–

TEST CONDITION

2^7	2^6	2^5	2^4
OVF	ZERO	CRY	SIGN

Branch if ISAR (Lower) ≠7	BR7	aa	P0←(P0)+1+H'aa' ISARL≠7	8Faa	2	2	1	5	–	–	–	–
			P0←(P0)+2 if ISARL=7		2	2		4	–	–	–	–
Branch Relative	BR	aa	P0←(P0)+1+H'aa'	90aa	2	2	1	7	–	–	–	–
Jump*	JMP	aaaa	P0←H'aaaa'	29aaaa	3	1	3	11	–	–	–	–

*Privileged instruction, accumulator contents altered during execution JMP.

MEMORY REFERENCE INSTRUCTIONS In all Memory Reference Instructions, the Data Counter is incremented DC←(DC)+1.

Operation	Mnemonic Op Code	Operand	Function	Machine Code	Bytes	Cycles Short	Cycles Long	(2 MHz φ)	Status Bits OVR	Status Bits ZERO	Status Bits CRY	Status Bits SIGN
Add Binary	AM		A←(A)+[(DC)]	88	1	1	1	5	1/0	1/0	1/0	1/0
Add Decimal	AMD		A←(A)+[(DC)]• BCD Adjust	89	1	1	1	5	1/0	1/0	1/0	1/0
AND	NM		A←(A)Λ[(DC)]	8A	1	1	1	5	0	1/0	0	1/0
Compare	CM		[(DC)]+(A)+1	8D	1	1	1	5	1/0	1/0	1/0	1/0
Exclusive OR	XM		A←(A)⊕[(DC)]	8C	1	1	1	5	0	1/0	0	1/0
Load	LM		A←[(DC)]	16	1	1	1	5	–	–	–	–
Logical OR	OM		A←(A)v[(DC)]	8B	1	1	1	5	0	1/0	0	1/0
Store	ST		A←[(DC)]	17	1	1	1	5	–	–	–	–

ADDRESS REGISTER GROUP INSTRUCTIONS

Operation	Mnemonic Op Code	Operand	Function	Machine Code	Bytes	Cycles Short	Cycles Long	(2 MHz φ)	Status Bits OVR	Status Bits ZERO	Status Bits CRY	Status Bits SIGN
Add to Data Counter	ADC		DC←(DC)+(A)	8E	1	1	1	5	–	–	–	–
Call to Subroutine*	PK		P0U←(r12); P0L←(r13), P←(P0)	0C	1	1	2	8	–	–	–	–
Call to Subroutine Immediate*	PI	aaaa	P←(P0), P0←H'aaaa	28aaaa	3	2	3	13	–	–	–	–
Exchange DC	XDC		(DC)⇄(DC1)	2C	1	2		4	–	–	–	–
Load Data Counter	LR	DC, Q	DCU←(r14), DCL←(r15)	0F	1	1	2	8	–	–	–	–
Load Data Counter	LR	DC, H	DCU←(r10), DCL←(r11)	10	1	1	2	8	–	–	–	–
Load DC Immediate	DCI	aaaa	DC H'aaaa'	2Aaaaa	3	3	2	12	–	–	–	–
Load Program Counter	LR	P0, Q	P0U←(r14), P0L←(r15)	0D	1	1	2	8	–	–	–	–
Load Stack Register	LR	P, K	PU←(r12), PL←r13)	09	1	1	2	8	–	–	–	–
Return from Subroutine*	POP		P0←(P)	1C	1	2		4	–	–	–	–
Store Data Counter	LR	Q, DC	r14←(DCU), r15←(DCL)	0E	1	1	2	8	–	–	–	–
Store Data Counter	LR	H, DC	r10←DCU, r11←(DCL)	11	1	1	2	8	–	–	–	–
Store Stack Register	LR	K, P	r12←(PU), r13←(PL)	08	1	1	2	8	–	–	–	–

SCRATCHPAD REGISTER INSTRUCTIONS (Refer to Scratchpad Addressing Modes)

Operation	Mnemonic Op Code	Operand	Function	Machine Code	Bytes	Cycles Short	Cycles Long	(2 MHz φ)	Status Bits OVR	Status Bits ZERO	Status Bits CRY	Status Bits SIGN
Add Binary	AS	r	A←(A)+(r)	Cr	1	1		2	1/0	1/0	1/0	1/0
Add Decimal	ASD	r	A←(A)+(r)	Dr	1	2		4	1/0	1/0	1/0	1/0
Decrement	DS	r	r←(r)+H'FF'	3r	1		1	3	1/0	1/0	1/0	1/0
Load	LR	A, r	A←(r)	4r	1	1		2	–	–	–	–
Load	LR	A, KU	A←(r12)	00	1	1		2	–	–	–	–
Load	LR	A, KL	A←(r13)	01	1	1		2	–	–	–	–
Load	LR	A, QU	A←(r14)	02	1	1		2	–	–	–	–
Load	LR	A, QL	A←(r15)	03	1	1		2	–	–	–	–
Load	LR	r, A	r←(A)	5r	1	1		2	–	–	–	–
Load	LR	KU, A	r12←(A)	04	1	1		2	–	–	–	–
Load	LR	KL, A	r13←(A)	05	1	1		2	–	–	–	–
Load	LR	QU, A	r14←(A)	06	1	1		2	–	–	–	–
Load	LR	QL, A	r15←(A)	07	1	1		2	–	–	–	–
AND	NS	r	A←(A)Λ(r)	Fr	1	1		2	0	1/0	0	1/0
Exclusive OR	XS	r	A←(A)+(r)	Er	1	1		2	0	1/0	0	1/0

*Privileged instruction, accumulator contents altered during execution of PI instruction.

MISCELLANEOUS INSTRUCTIONS

Operation	Mnemonic Op Code	Operand	Function	Machine Code	Bytes	Cycles Short	Cycles Long	(2 MHz φ)	Status Bits OVR	Status Bits ZERO	Status Bits CRY	Status Bits SIGN
Disable Interrupt	DI		Reset ICB	1A	1	1		2	–	–	–	–
Enable Interrupt*	EI		Set ICB	1B	1	1		2	–	–	–	–
Input	IN	04,05,06,07	A←(Input Port aa)	26aa	2	1	2	8	0	1/0	0	1/0
Input Short	INS	0, 1	A←(Input Port 0 or 1)	A0,A1	1	2		4	0	1/0	0	1/0
Input Short	INS	4,5,6,7	A←(Input Port a)	Aa	1	1	2	8	0	1/0	0	1/0
Load ISAR	LR	IS,A	IS←(A)	0B	1	1		2	–	–	–	–
Load ISAR Lower	LISL	bbb	ISL←bbb	6(1bbb)**	1	1		2	–	–	–	–
Load ISAR Upper	LISU	bbb	ISU←bbb	6(0bbb)**	1	1		2	–	–	–	–
Load Status Register*	LR	W,J	W←(r9)	1D	1	2		4	1/0	1/0	1/0	1/0
No Operation	NOP		P0←(P0)+1	2B	1	1		2	–	–	–	–
Output*	OUT	04,05,06,07	Output Port aa←(A)	27aa	2	1	2	8	–	–	–	–
Output Short	OUTS	0, 1	Output Port 0 or 1←(A)	B0,B1	1	2		4	–	–	–	–
Output Short	OUTS	4,5,6,7	Output Port a←(A)	Ba	1	1	2	8	–	–	–	–
Store ISAR	LR	A,IS	A←(IS)	0A	1	1		2	–	–	–	–
Store Status Reg	LR	J,W	r9←(W)	1E	1	1		2	–	–	–	–

*Privileged instruction
**b = 1-bit immediate operand

3

NOTES

Lower case denotes variables specified by programmer

Function	Definitions
←	is replaced by
( )	the contents of
( )	Binary "1s" complement of
+	Arithmetic Add (Binary or Decimal)
⊕	Logical "OR" exclusive
Λ	Logical "AND"
v	Logical "OR" inclusive
H"	Hexadecimal digit
[( )]	Contents of memory specified by ( )
a	Address Variable (four bits)
A	Accumulator
b	One bit immediate operand
DC	Data Counter (Indirect Address Register)
DC1	Data Counter 1 (Auxiliary Data Counter)
DCL	Least significant 8 bits of Data Counter Addressed
DCU	Most significant 8 bits of Data Counter Addressed
H	Scratchpad Register 10 and 11
i	Immediate operand (four bits)
ICB	Interrupt Control Bit
IS	Indirect Scratchpad Address Register
ISL	Least Significant 3 bits of ISAR
ISU	Most Significant 3 bits of ISAR
J	Scratchpad Register 9
K	Registers 12 and 13
KL	Register 13
KU	Register 12
P0	Program Counter
P0L	Least Significant 8 Bits of Program Counter
P0U	Most Significant 8 bits of Program Counter
P	Stack Register
PL	Least Significant 8 bits of Program Counter
PU	Most Significant 8 bits of Active Stack Register
Q	Registers 14 and 15
QL	Reigster 15
QU	Register 14
r	Scratchpad Register (any address 0 through B) (See Below)
W	Status Register

Scratchpad Addressing Modes Using IS. (r≠0 through B)

r=H'C'	Register Addressed by IS is (Unmodified)
r=H'D'	Register Addressed by IS is Incremented
r=H'E'	Register Addressed by IS is Decremented
r=H'F'	Illegal OP Code

Status Register

–	No change in condition
1/0	is set to "1" or "0" depending on conditions
CRY	Carry Flag
OVR	Overflow Flag
SIGN	Sign of Result Flag
ZERO	Zero Flag

ORDERING INFORMATION

The following information is required when ordering a custom MCU. This information may be transmitted to Motorola in the following media:

PROM(s) MCM2716s or MCM2708s

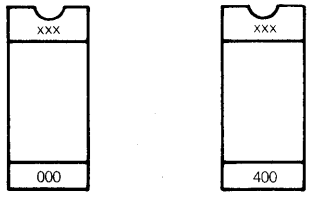
MDOS disk file

To initiate a ROM pattern for the MCU it is necessary to first contact your local field service office, local sales person, or your local Motorola representative.

PROMs – The MCM2708 or MCM2716 type PROMs, programmed with the customer program (positive logic sense for address and data), may be submitted for pattern generation. The PROMs must be clearly marked to indicate which PROM corresponds to which address space (000-3FF HEX), (400-7FF) or (000-7FF). See Figure 21 for recommended marking procedure.

After the PROM(s) are marked they should be placed in conductive IC carriers and securely packed. Do not use styrofoam.

FIGURE 21 – PROM MARKING



xxx = Customer ID

VERIFICATION MEDIA

All original pattern media (PROMs or Floppy Disk) are filed for contractual purposes and are not returned. A computer listing of the ROM code will be generated and returned along with a listing verification form. The listing should be thoroughly checked and the verification form completed, signed, and returned to Motorola. The signed verification form constitutes the contractual agreement for creation of the customer mask. If desired, Motorola will program a blank 2716 EPROM (supplied by the customer) from the data file used to create the custom mask to aid in the verification process.

ROM VERIFICATION UNITS

Ten MC3870s containing the customer's ROM pattern will be sent for program verification. These units will have been made using the custom mask but are for the purpose of ROM verification only. For expediency they are usually unmarked, packaged in ceramic, and tested only at room temperature and 5 volts. These RVUs are included in the mask charge and are not production parts.

FLEXIBLE DISKS

The disk media submitted must be single-sided, single-density, 8-inch, MDOS compatible floppies. The customer must write the binary file name and company name on the disk with a felt-tip pen. The floppies are not to be returned by Motorola as they are used for archival storage. The minimum MDOS system files must be on the disk as well as the absolute binary object file (filename .LO type of file) from the MC3870 cross assembler. An object file made from a memory dump using the ROLLOUT command is also admissable. Consider submitting a source listing as well as the following files: filename .LX (EXORciser® loadable format) and filename .SA (ASCII Source Code). These files will of course be kept confidential and are used 1) to speed up the process in house if any problems arise, and 2) to speed up our customer to factory interface if a user finds any software errors and needs assistance quickly from the factory representatives.

MDOS is Motorola's Disk Operating System available on development systems such as EXORcisers, or EXORsets, etc.

MC3870 ORDERING INFORMATION

Date _____ Customer PO Number _____

Customer Name _____

Address _____

City _____ State _____ Zip _____

Country _____

Phone _____ Extension _____

Contact _____

Customer Part Number _____

Options:

	Reset	Pullup ☐	No Pullup ☐
	External Interrupt	Pullup ☐	No Pullup ☐

Port Options:

	Standard TTL	Open Drain	Direct Drive
P4-0	☐	☐	☐
P4-1	☐	☐	☐
P4-2	☐	☐	☐
P4-3	☐	☐	☐
P4-4	☐	☐	☐
P4-5	☐	☐	☐
P4-6	☐	☐	☐
P4-7	☐	☐	☐
P5-0	☐	☐	☐
P5-1	☐	☐	☐
P5-2	☐	☐	☐
P5-3	☐	☐	☐
P5-4	☐	☐	☐
P5-5	☐	☐	☐
P5-6	☐	☐	☐
P5-7	☐	☐	☐

Pattern Media

☐ PROMs (MCM2716 or MCM2708)
(Customer can send in two extra PROMs, Motorola will program the customer's code on these PROMs for code verification.

☐ Floppy Disk

☐ Other _____

Clock Mode _____ ☐ XTAL ☐ RC ☐ LC ☐ External

Clock Freq _____

Temp Range _____ ☐ 0-70°C ☐ −40-+85°C

Marking Information (12 Characters Maximum)

_____

NOTE: All other media requires prior factory approval.