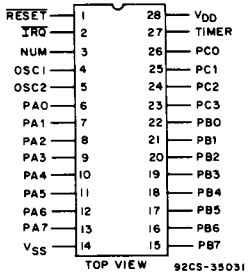
CDP6805F2, CDP6805F2C



TERMINAL ASSIGNMENT

CMOS High-Performance Silicon-Gate 8-Bit Microcomputer

Hardware Features:

- *Typical full speed operating power of 10 mW at 5 V*
- *Typical WAIT mode power of 3 mW*
- *Typical STOP mode power of 5 μW*
- *64 bytes of on-chip RAM*
- *1089 bytes of on-chip ROM*
- *16 bidirectional I/O lines*
- *4 input-only lines*
- *Internal 8-bit timer with software programmable 7-bit prescaler*
- *External timer input*
- *External and timer interrupts*
- *Master reset and power-on reset*
- *Single 3 to 6 volt supply*
- *On-chip oscillator*
- *1 μs cycle time*

The CDP6805F2 Microcomputer Unit (MCU) belongs to the CDP6805 Family of CMOS Microcomputers. This 8-bit MCU contains on-chip oscillator CPU, RAM, ROM, I/O, and Timer. Fully static design allows operation at frequencies down to DC, further reducing its already low-power consumption. It is a low-power processor designed for low-end to mid-range applications in the consumer, automotive, industrial, and communications markets where very low power consumption constitutes an important factor.

Software Features:

- *Versatile interrupt handling*
- *True bit manipulation*
- *10 addressing modes*
- *Efficient instruction set*
- *Memory-mapped I/O*
- *User-callable self-check routines*
- *Two power-saving standby modes*

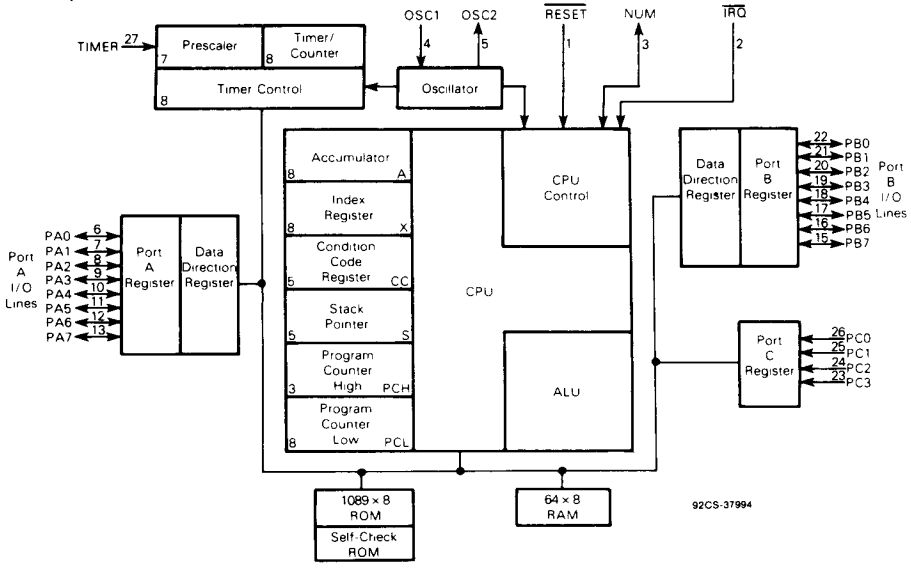


Fig. 1 - CDP6805F2 CMOS microcomputer block diagram.

File Number **1369**

CDP6805F2, CDP6805F2C

MAXIMUM RATINGS (Voltages Referenced to V_{SS})

Ratings	Symbol	Value	Unit
Supply Voltage	V_{DD}	−0.3 to +8	V
All Input Voltages Except OSC1	V_{in}	V_{SS} −0.5 to V_{DD} +0.5	V
Current Drain per Pin Excluding V_{DD} and V_{SS}	I	10	mA
Operating Temperature Range CDP6805F2 CDP6805F2C	T_A	T_L to T_H 0 to 70 −40 to +85	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C

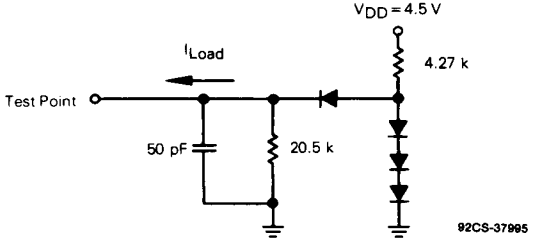


Fig. 2 - Equivalent test load.

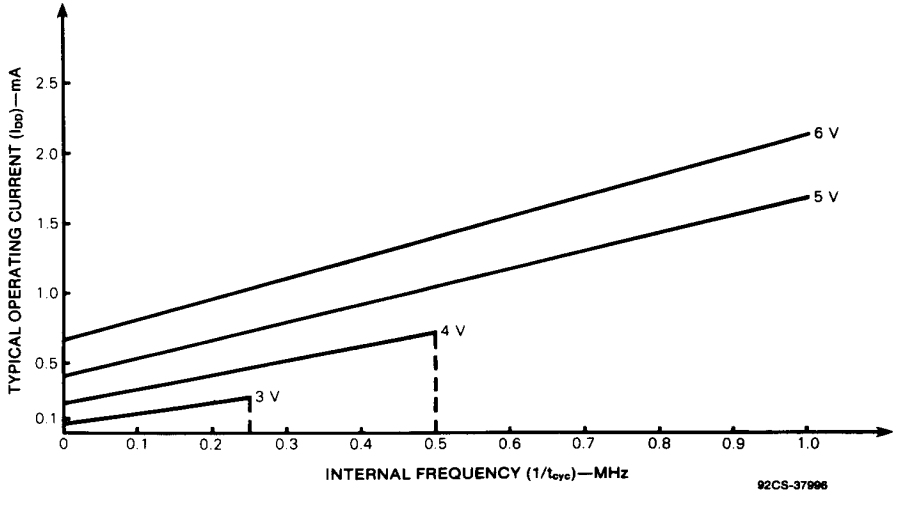


Fig. 3 - Typical operating current vs. internal frequency.

DC ELECTRICAL CHARACTERISTICS (V_{DD} = 5 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, unless otherwise noted) (See Note 1)

Characteristics	Symbol	Min	Max	Unit
Output Voltage, $I_{Load} \leq 10.0\ \mu A$	V_{OL} V_{OH}	– $V_{DD}-0.1$	0.1 –	V
Output High Voltage ($I_{Load} = -200\ \mu A$) PA0-PA7, PB0-PB7	V_{OH}	4.1	–	V
Output Low Voltage, ($I_{Load} = 800\ \mu A$) PA0-PA7, PB0-PB7	V_{OL}	–	0.4	V
Input High Voltage Ports PA0-PA7, PB0-PB7, PC0-PC3 TIMER, $\overline{IRQ}$, $\overline{RESET}$ OSC1	V_{IH}	$V_{DD}-2$ $V_{DD}-0.8$ $V_{DD}-1.5$	V_{DD} V_{DD} V_{DD}	V
Input Low Voltage, All Inputs	V_{IL}	V_{SS}	0.8	V
Total Supply Current (C_L = 50 pF on Ports, No dc Loads, t_{cyc} = 1 µs) RUN (Measured During Self-Check, V_{IL} = 0.2 V, $V_{IH} = V_{DD}-0.2$ V) WAIT (See Note 2) STOP (See Note 2)	I_{DD}	– – –	**4** **1.5** **150**	mA mA µA
I/O Ports Input Leakage – PA0-PA7, PB0-PB7	I_{IL}	–	± 10	µA
Input Current – $\overline{RESET}$, $\overline{IRQ}$, TIMER, OSC1, PC0-PC3	I_{in}	–	± 1	µA
Output Capacitance – Ports A and B	C_{out}	–	12	pF
Input Capacitance – $\overline{RESET}$, $\overline{IRQ}$, TIMER, OSC1, PC0-PC3	C_{in}	–	8	pF

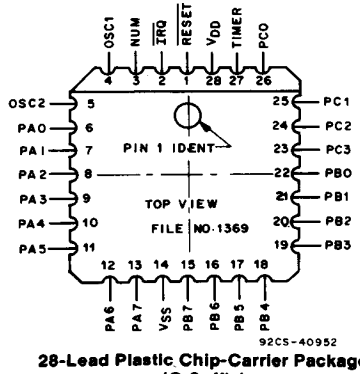
NOTES:
1. Electrical Characteristics for V_{DD} = 3 V available soon.
2. Test Conditions for I_{DD} are as follows:
   All ports programmed as inputs
   V_{IL} = 0.2 V (PA0-PA7, PB0-PB7, PC0-PC3)
   $V_{IH} = V_{DD}-0.2$ V for $\overline{RESET}$, $\overline{IRQ}$, TIMER
   OSC1 input is a square wave from 0.2 V to $V_{DD}-0.2$ V
   OSC2 output load = 20 pF (WAIT I_{DD} is affected linearly by the OSC2 capacitance)

TABLE 1 – CONTROL TIMING CHARACTERISTICS (V_{DD} = 5 Vdc ± 10%, V_{SS} = 0, $T_A = T_L$ to T_H, f_{osc} = 4 MHz, t_{cyc} = 1 µs)

Characteristics	Symbol	Min	Max	Unit
Crystal Oscillator Startup Time (See Figure 5)	t_{OXOV}	–	100	ms
Stop Recovery Startup Time – Crystal Oscillator (See Figure 6)	t_{ILCH}	–	100	ms
Timer Pulse Width (See Figure 4)	t_{TH}, t_{TL}	0.5	–	t_{cyc}
Reset Pulse Width (See Figure 5)	t_{RL}	1.5	–	t_{cyc}
Timer Period (See Figure 4)	t_{TLTL}	1	–	t_{cyc}
Interrupt Pulse Width (See Figure 15)	t_{ILIH}	1	–	t_{cyc}
Interrupt Pulse Period (See Figure 15)	t_{ILIL}	*	–	t_{cyc}
OSC1 Pulse Width (See Figure 7)	t_{OH}, t_{OL}	100	–	ns
Cycle Time	t_{cyc}	1000	–	ns
Frequency of Operation Crystal External Clock	f_{osc}	– dc	4 4	MHz

*The minimum period, t_{ILIL}, should not be less than the number of t_{cyc} cycles it takes to execute the interrupt service routines plus 20 t_{cyc} cycles.

TERMINAL ASSIGNMENT



28-Lead Plastic Chip-Carrier Package
(Q Suffix)

CDP6805F2, CDP6805F2C

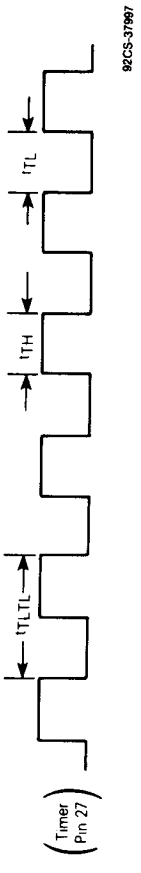


Fig. 4 - Timer relationships.

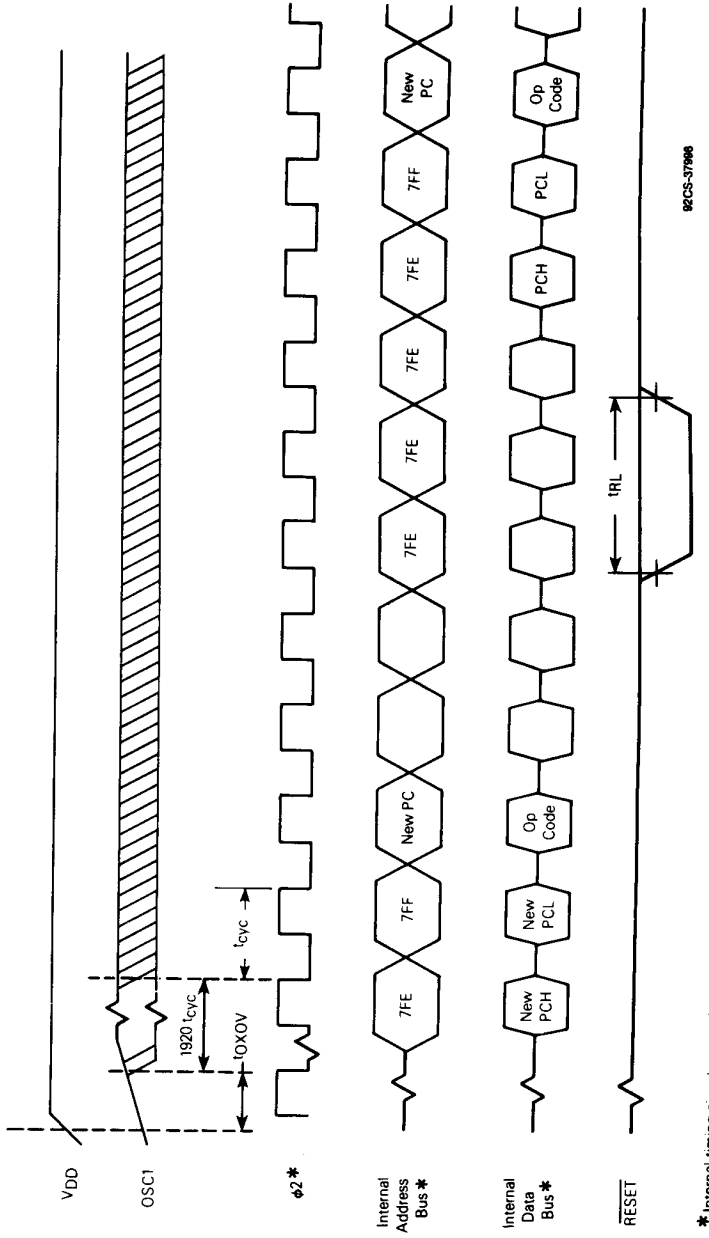
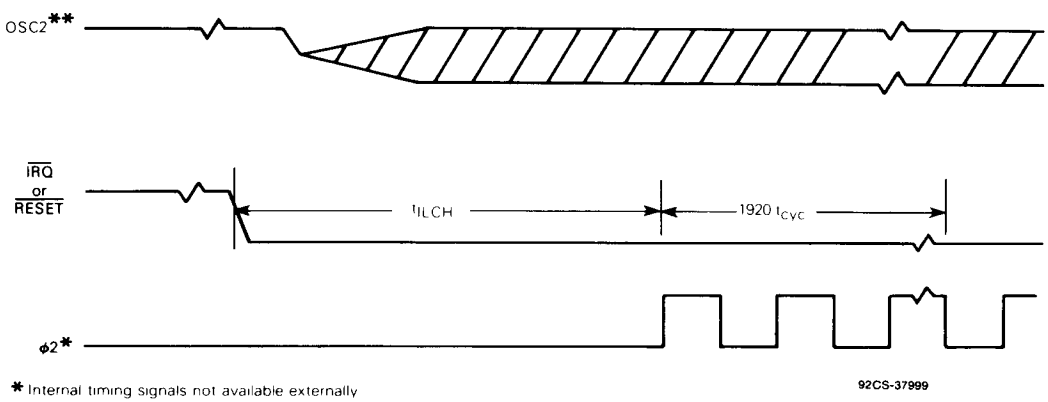


Fig. 5 - Power-on RESET and $\overline{RESET}$.



* Internal timing signals not available externally

** Represents the internal gating of the OSC1 input pin

Fig. 6 - Stop recovery.

FUNCTIONAL PIN DESCRIPTION

V_{DD} and V_{SS}

Power is supplied to the MCU using these two pins. V_{DD} is power and V_{SS} is ground.

IRQ (MASKABLE INTERRUPT REQUEST)

IRQ is photomask option selectable with the choice of interrupt sensitivity being both level and negative edge or negative edge only. The MCU completes the current instruction before it responds to the request. If IRQ is low and the interrupt mask bit (I bit) in the condition code register is clear, the MCU begins an interrupt sequence at the end of the current instruction.

If the photomask option is selected to include level sensitivity, then the IRQ input requires an external resistor to V_{DD} for "wire-OR" operation. See the Interrupt section for more detail.

RESET

The RESET input is not required for start-up but can be used to reset the MCU's internal state and provide an orderly software start-up procedure. Refer to the Resets section for a detailed description.

TIMER

The TIMER input may be used as an external clock for the on-chip timer. Refer to the Timer section for a detailed description.

NUM (NON-USER MODE)

This pin is intended for use in self-check only. User applications should leave this pin connected to ground through a 10 kilohm resistor.

OSC1, OSC2

The **CDP6805F2** can be configured to accept either a crystal input or an RC network. Additionally, the internal clocks can be derived from either a divide-by-two or divide-by-four of the external frequency (f_{OSC}). Both of these options are photomask selectable.

RC – If the RC oscillator option is selected, then a resistor is connected to the oscillator pins as shown in Figure 7(b). The relationship between R and f_{OSC} is shown in Figure 8.

CRYSTAL – The circuit shown in Figure 7(a) is recommended when using a crystal. The internal oscillator is designed to interface with an AT-cut parallel resonant quartz crystal resonator in the frequency range specified for f_{OSC} in the electical characteristics table. Using an external CMOS oscillator is suggested when crystals outside the specified ranges are to be used. The crystal and components should be mounted as close as possible to the input pins to minimize output distortion and start-up stabilization time. Crystal frequency limits are also affected by V_{DD}. Refer to Table 1, Control Timing Characteristics, for limits.

EXTERNAL CLOCK – An external clock should be applied to the OSC1 input with the OSC2 input not connected, as shown in Figure 7(c). An external clock may be used with either the RC or crystal oscillator mask option. t_{OXOV} or t_{ILCH} do not apply when using an external clock input.

PA0-PA7

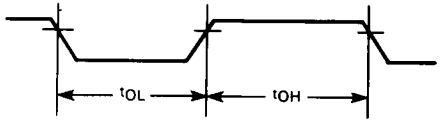
These eight I/O lines comprise Port A. The state of any pin is software programmable. Refer to the Input/Output Programming section for a detailed description.

CDP6805F2, CDP6805F2C

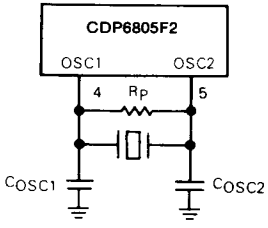
Crystal Parameters

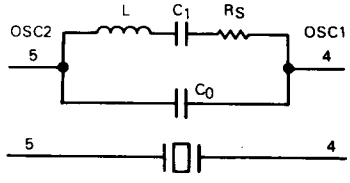
	1 MHz	4 MHz	Units
R_{SMAX}	400	75	Ω
C_0	5	7	pF
C_1	0.008	0.012	μF
C_{OSC1}	15-40	15-30	pF
C_{OSC2}	15-30	15-25	pF
R_P	10	10	MΩ
Q	30 k	40 k	–

Oscillator Waveform

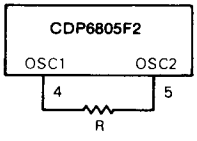


(a) Crystal Oscillator Connections and Equivalent Crystal Circuit





(b) RC Oscillator Connection



(c) External Clock Source Connections

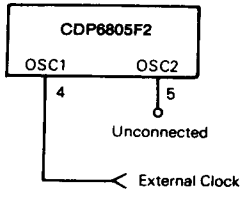
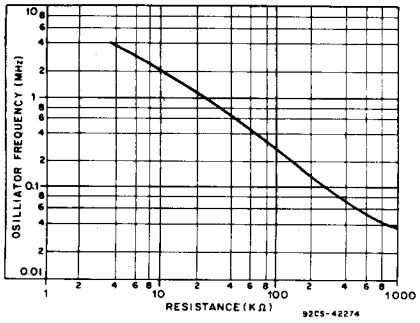


Fig. 7 – Oscillator connections.

92CS-38000



R (kΩ)

Fig. 8 - Typical frequency vs. resistance for RC oscillator option only.

PB0-PB7

These eight lines comprise Port B. The state of any pin is software programmable. Refer to the Input/Output Programming section for a detailed description.

PC0-PC3

These four lines comprise Port C, a fixed input port. When Port C is read, the four most-significant bits on the data bus are "1s". There is no data direction register associated with Port C.

INPUT/OUTPUT PROGRAMMING

Any Port A or B pin may be software programmed as an input or output by the state of the corresponding bit in the port data direction register (DDR). A pin is configured as an output if its corresponding DDR bit is set to a logic "1". A pin is configured as an input if its corresponding DDR bit is cleared to a logic "0". At reset, all DDRs are cleared, which configures all port pins as inputs. A port pin configured as an output will output the data in the corresponding bit of its port data latch. Refer to Figure 9 and Table 2.

3

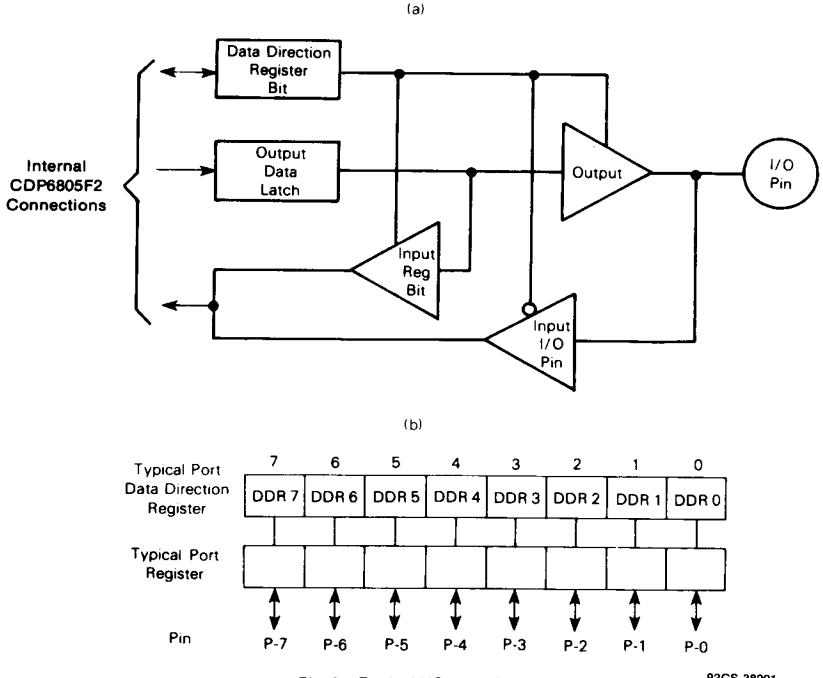


Fig. 9 - Typical I/O port circuitry.

TABLE 2 — I/O PIN FUNCTIONS

R/$\overline{W}$	DDR	I/O Pin Function
0	0	The I/O pin is in input mode. Data is written into the output data latch.
0	1	Data is written into the output data 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 an output mode. The output data latch is read.

CDP6805F2, CDP6805F2C

SELF-CHECK

The **CDP6805F2** self-check is performed using the circuit in Figure 10. Self-check is initiated by tying NUM and TIMER pins to a logic "1" then executing a reset. After reset, the following five tests are executed automatically:

I/O – Functionally Exercise Ports A, B, C

RAM – Walking Bit Test

ROM – Exclusive OR with ODD "1s" Parity Result

Timer – Functionally Exercise Timer

Interrupts – Functionally Exercise External and Timer Interrupts

Self-check results are shown in Table 3. The following subroutines are available to user programs and do not require any external hardware.

TABLE 3 – SELF-CHECK RESULTS

PB3	PB2	PB1	PB0	Remarks
1	0	1	1	Bad Timer
1	1	0	0	Bad RAM
1	1	0	1	Bad ROM
1	1	1	0	Bad Interrupt or Request Flag
All Cycling				Good Part
All Others				Bad Part

RAM SELF-CHECK SUBROUTINE

Returns with the Z bit clear if any error is detected; otherwise, the Z bit is set.

The RAM test must be called with the stack pointer at $7F and the accumulator zeroed. When run, the test checks every RAM cell except for $7F and $7E which are assumed to contain the return address.

A and X are modified. All RAM locations except the top 2 are modified. (Enter at location $78B.)

ROM CHECKSUM SUBROUTINE

Returns with Z bit cleared if any error was found; otherwise Z = 1, X = 0 on return, and A is zero if the test passed. RAM locations/**$41-$44** are overwritten. (Enter at location $7A4.)

TIMER TEST SUBROUTINE

Return with Z bit cleared if any error was found; otherwise Z = 1.

This routine runs a simple test on the timer. In order to work correctly as a user subroutine, the internal clock must be the clocking source and interrupts must be disabled. Also, on exit, the clock will be running and the interrupt mask will not be set, so the caller must protect himself from interrupts if necessary.

A and X register contents are lost; this routine counts how many times the clock counts in 128 cycles. The number of counts should be a power of two since the prescaler is a power of two. If not, the timer probably is not counting correctly. The routine also detects if the timer is running at all. (Enter at location $7BE.)

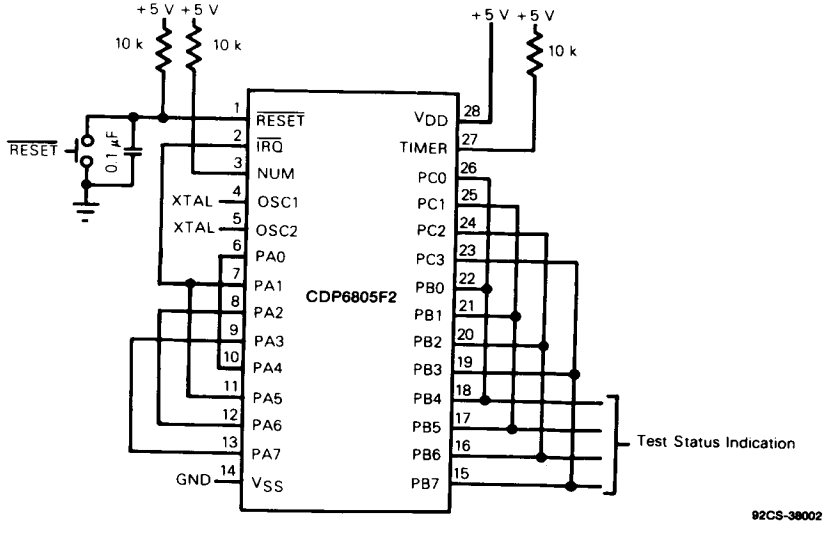


Fig. 10 - Self-check pinout configuration.

MEMORY

The **CDP6805F2** has a total address space of 2048 bytes of memory and I/O registers. The address space is shown in Figure 11.

The first 128 bytes of memory (first half of page zero) is comprised of the I/O port locations, timer locations, and 64 bytes of RAM. The next 1079 bytes comprise the user ROM. The 10 highest address bytes contain the reset and interrupt vectors.

The stack pointer is used to address data stored on the stack. Data is stored on the stack during interrupts and subroutine calls. At power-up, the stack pointer is set to $7F and it is decremented as data is pushed on the stack. When data is removed from the stack, the stack pointer is incremented. A maximum of 32 bytes of RAM are available for stack usage. Since most programs use only a small part of the allocated stack locations for interrupts and/or subroutine stacking purposes, the unused bytes are available for program data storage.

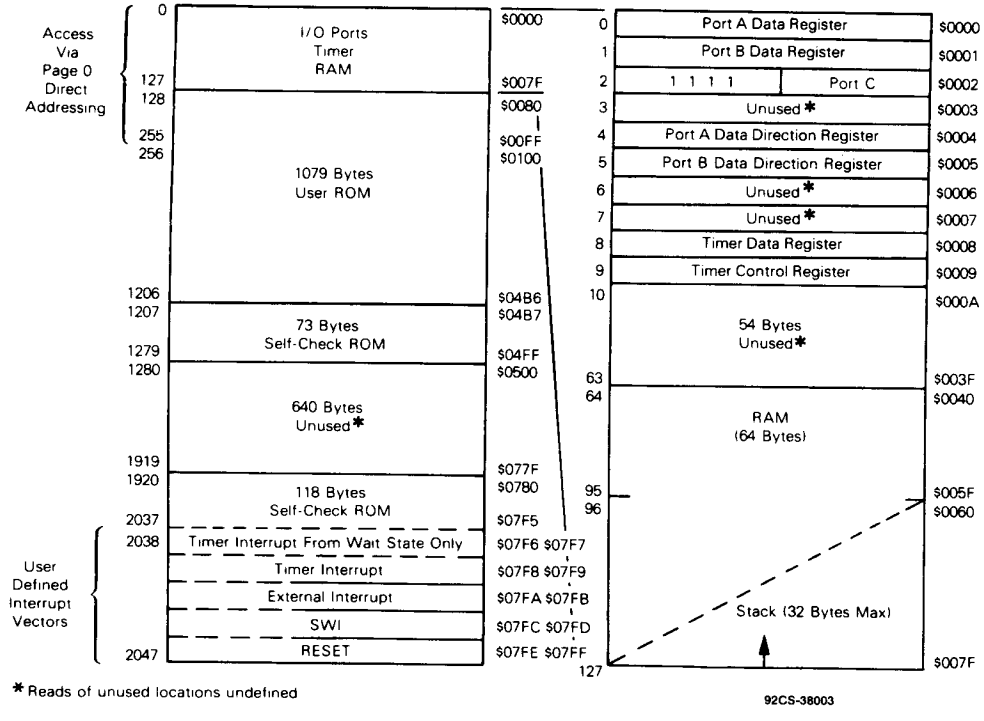


Fig. 11 - Address map.

CDP6805F2, CDP6805F2C

REGISTERS

The **CDP6805F2** contains five registers as shown in the programming model (Figure 12). The interrupt stacking order is shown in Figure 13.

ACCUMULATOR (A)

This accumulator is an 8-bit general purpose register used to hold operands and results of the arithmetic calculations and data manipulations.

INDEX REGISTER (X)

The X register is an 8-bit register which is used during the indexed modes of addressing. It provides the 8-bit operand which is used to create an effective address. The index register is also used for data manipulations with the read-modify-write type of instructions and as a temporary storage register when not performing addressing operations.

PROGRAM COUNTER (PC)

The program counter is an 11-bit register that contains the address of the next instruction to be executed by the processor.

STACK POINTER (SP)

The stack pointer is an 11-bit register containing the address of the next free location on the stack. When accessing memory, the six most-significant bits are appended to the five least-significant register bits to produce an address within the range of $7F to $60. The stack area of RAM is used to store the return address on subroutine calls and the machine state during interrupts. During external or power-on reset, and during a "reset stack pointer" instruction, the stack pointer is set to its upper limit ($7F). Nested interrupts and/or subroutines may use up to 32 (decimal) locations beyond which the stack pointer "wraps around" and points to its upper limit thereby losing the previously stored information. A subroutine call occupies two RAM bytes on the stack, while an interrupt uses five bytes.

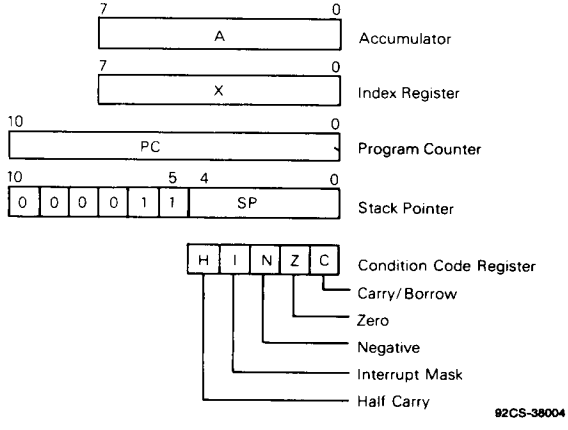
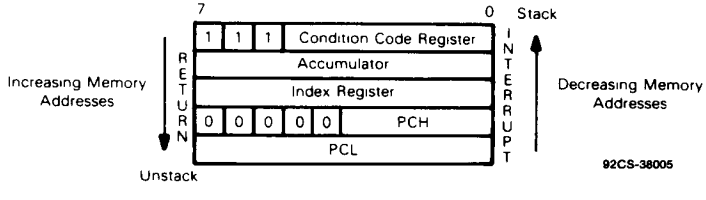


Fig. 12 - Programming model.



NOTE: Since the Stack Pointer decrements during pushes, the PCL is stacked first, followed by PCH, etc. Pulling from the stack is in the reverse order.

Fig. 13 - Stacking order.

CONDITION CODE REGISTER (CC)

The condition code register is a 5-bit register which indicates the results of the instruction just executed. These bits can be individually tested by a program and specific action taken as a result of their state. Each bit is explained in the following paragraphs.

HALF CARRY BIT (H) — The H bit is set to a "1" when a carry occurs between bits 3 and 4 of the ALU during an ADD or ADC instruction. The H bit is useful in binary coded decimal subroutines.

INTERRUPT MASK BIT (I) — When the I bit is set, both the external interrupt and the timer interrupt are disabled. Clearing this bit enables the above interrupts. If an interrupt occurs while the I bit is set, the interrupt is latched and is processed when the I bit is next cleared.

NEGATIVE (N) — Indicates that the result of the last arithmetic, logical, or data manipulation is negative (bit 7 in the result is a logical "1").

ZERO (Z) — Indicates that the result of the last arithmetic, logical, or data manipulation is zero.

CARRY/BORROW (C) — Indicates that a carry or borrow out of the arithmetic logic unit (ALU) occurred during the last arithmetic operation. This bit is also affected during bit test and branch instructions, shifts, and rotates.

RESETS

The **CDP6805F2** has two reset modes: an active low external reset pin ($\overline{\text{RESET}}$) and a power-on reset function; refer to Figure 5.

$\overline{\text{RESET}}$

The $\overline{\text{RESET}}$ input pin is used to reset the MCU to provide an orderly software start-up procedure. When using the external reset mode, the $\overline{\text{RESET}}$ pin must stay low for a minimum of one t_{RL}. The $\overline{\text{RESET}}$ pin is provided with a Schmitt Trigger input to improve its noise immunity.

POWER-ON RESET

The power-on reset occurs when a positive transition is detected on V_{DD}. The power-on reset is used strictly for power turn-on conditions and should not be used to detect any drops in the power supply voltage. There is no provision for a power-down reset. The power-on circuitry provides for a 1920 t_{cyc} delay from the time of the first oscillator operation. If the external $\overline{\text{RESET}}$ pin is low at the end of the 1920 time out, the processor remains in the reset condition.

Either of the two types of reset conditions causes the following to occur:

- Timer control register interrupt request bit (TCR7) is cleared to a "0".
- Timer control register interrupt mask bit (TCR6) is set to a "1".
- All data direction register bits are cleared to a "0". All ports are defined as inputs.
- Stack pointer is set to $7F.
- The internal address bus is forced to the reset vector ($7FE, $7FF).
- Condition code register interrupt mask bit (I) is set to a "1".
- STOP and WAIT latches are reset.
- External interrupt latch is reset.

All other functions, such as other registers (including output ports), the timer, etc., are not cleared by the reset conditions.

INTERRUPTS

Systems often require that normal processing be interrupted so that some external event may be serviced. The **CDP6805F2** may be interrupted by one of three different methods, either one of two maskable interrupts (external input or timer) or a non-maskable software interrupt (SWI).

Interrupts cause the processor registers to be saved on the stack and the interrupt mask set to prevent additional interrupts. The RTI instruction causes the register contents to be recovered from the stack and return to normal processing. The stacking order is shown in Figure 13.

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

When the current instruction is complete, the processor checks all pending hardware interrupts and if unmasked, proceeds with interrupt processing; otherwise, the next instruction is fetched and executed. Note that masked interrupts are latched for later interrupt service.

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 as any other instruction. Refer to Figure 14 for the interrupt and instruction processing sequence.

CDP6805F2, CDP6805F2C

TIMER INTERRUPT

Each time the timer decrements to zero (transitions from $01 to $00), the timer interrupt request bit (TCR7) is set. The processor is interrupted only if the timer mask bit (TCR6) and interrupt mask bit (I bit) are both cleared. When the interrupt is recognized, the current state of the machine is pushed onto the stack and the interrupt mask bit in the condition code register is set. This mask prevents further interrupts until the present one is serviced. The processor now vectors to the timer interrupt service routine. The address for this service routine is specified by the contents of $7F8 and $7F9 unless the processor is in a WAIT mode, in which case the contents of $7F6 and $7F7 specify the timer service routine address. Software must be used to clear the timer interrupt request bit (TCR7). At the end of the timer interrupt service routine, the software normally executes an RTI instruction which restores the machine state and starts executing the interrupted program.

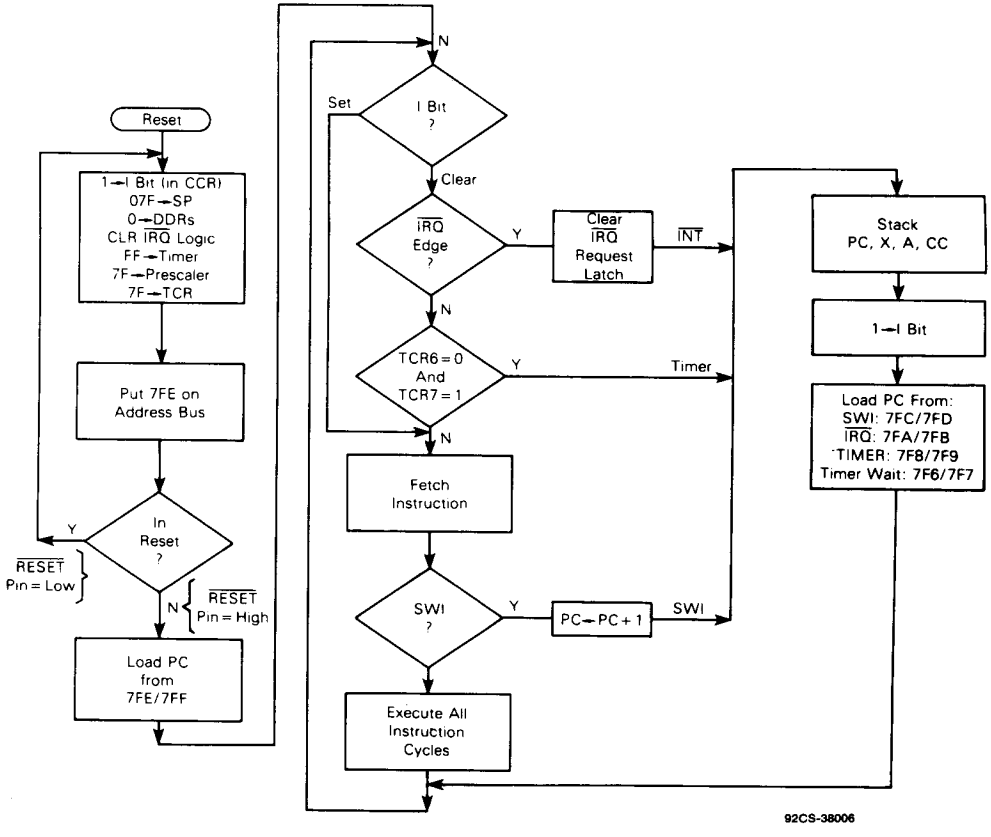


Fig. 14 - RESET and INTERRUPT processing flowchart.

EXTERNAL INTERRUPT

Either level- and edge-sensitive or edge-sensitive only inputs are available as mask options. If the interrupt mask bit of the condition code register is cleared and the external interrupt pin ($\overline{IRQ}$) is "low" or a negative edge has set the internal interrupt flip-flop, then the external interrupt occurs. The action of the external interrupt is identical to the timer except that the service routine address is specified by the contents of $7FA and $7FB. Figure 15 shows both a functional diagram and timing for the interrupt line. The timing diagram shows two different treatments of the interrupt line ($\overline{IRQ}$) to the processor. The first method is single pulses on the interrupt line spaced far enough apart to be serviced. The minimum time between pulses is a function of the length of the interrupt service routine. Once a pulse occurs, the next pulse should not occur until the MPU software has exited the routine (an RTI occurs). This time (t_{ILIL}) is obtained by adding 20 instruction cycles (t_{cyc}) to the total number of cycles it takes to complete the service routine including the RTI instruction; refer to Figure 15. The second configuration shows many interrupt lines "wire ORed" to form the interrupts at the processor. Thus, if after servicing an interrupt the $\overline{IRQ}$ remains low, then the next interrupt is recognized.

SOFTWARE INTERRUPT (SWI)

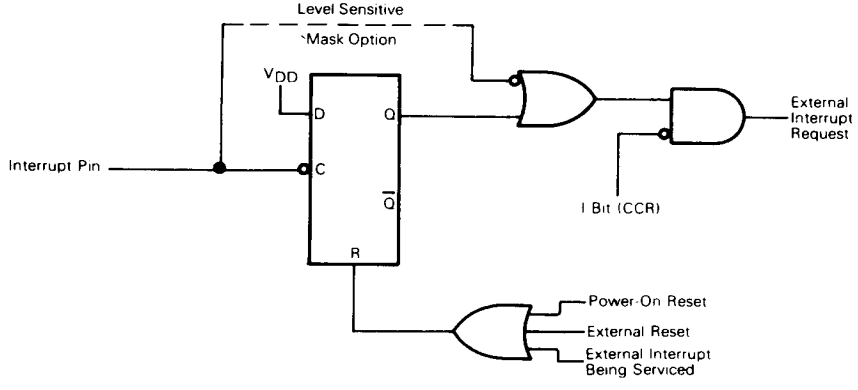
The software interrupt is an executable instruction. The action of the SWI instruction is similar to the hardware interrupts. The SWI is executed regardless of the state of the interrupt mask in the condition code register. The service routine address is specified by the contents of memory locations $7FC and $7FD.

The following three functions are not strictly interrupts, however, they are tied very closely to the interrupts. These functions are $\overline{RESET}$, STOP, and WAIT.

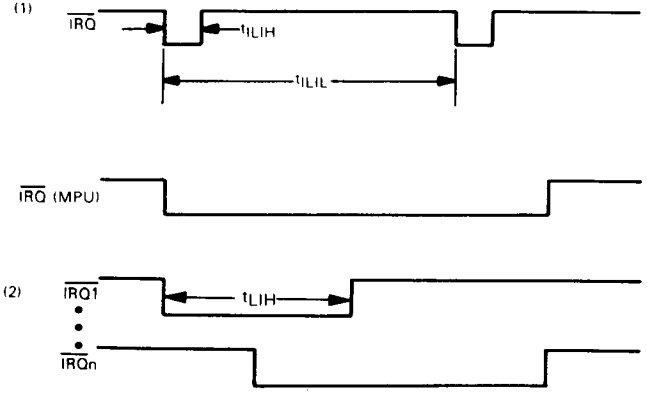
$\overline{RESET}$ – The $\overline{RESET}$ input pin and the internal power-on reset function each cause the program to vector to an initialization program. This vector is specified by the contents

3

(a) Interrupt Functional Diagram



(b) Interrupt Mode Diagram



Edge Condition

The minimum pulse width (t_{ILIH}) is one t_{cyc}. The period t_{ILIL} should not be less than the number of t_{cyc} cycles it takes to execute the interrupt service routine plus 20 t_{cyc} cycles.

Mask Optional Level Sensitive

If after servicing an interrupt the $\overline{IRQ}$ remains low, then the next interrupt is recognized.

92CS-38007

Fig. 15 - External interrupt.

CDP6805F2, CDP6805F2C

of memory locations $7FE and $7FF. The interrupt mask of the condition code register is also set. See preceding section on Reset for details.

STOP – The STOP instruction places the **CDP6805F2** in its lowest power consumption mode. In the STOP function, the internal oscillator is turned off causing all internal processing and the timer to be halted; refer to Figure 16.

During the STOP mode, timer control register (TCR) bits 6 and 7 are altered to remove any pending timer interrupt requests and to disable any further timing interrupts. External interrupts are enabled in the condition code register. All other registers and memory remain unaltered. All I/O lines remain unchanged. The processor can only be brought out of the STOP mode by an external $\overline{IRQ}$ or $\overline{RESET}$.

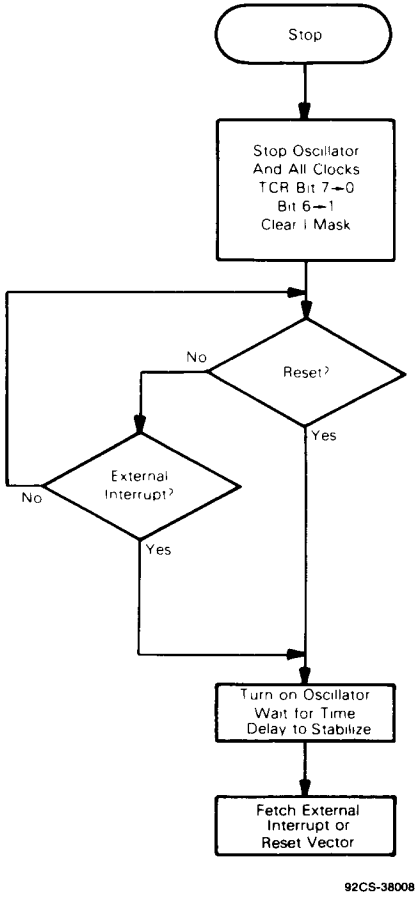


Fig. 16 - Stop function flowchart.

WAIT – The WAIT instruction places the **CDP6805F2** in a low-power consumption mode, but the WAIT mode consumes somewhat more power than the STOP mode. In the WAIT mode, the internal clock is disabled from all internal circuitry except the timer circuit; refer to Figure 17. Thus, all internal processing is halted, however, the timer continues to count normally.

During the WAIT mode, the I bit in the condition code register is cleared to enable interrupts. All other registers, memory, and I/O lines remain in their last state. The timer may be enabled by software prior to entering the WAIT mode to allow a periodic exit from the WAIT mode. If an external and a timer interrupt occur at the same time, the external interrupt is serviced first; then, if the timer interrupt request is not cleared in the external interrupt routine, the normal timer interrupt (not the timer WAIT interrupt) is serviced since the MCU is no longer in the WAIT mode.

TIMER

The MCU timer contains an 8-bit software programmable counter with a 7-bit software selectable prescaler. Figure 18 contains a block diagram of the timer. The counter may be preset under program control and decrements towards zero. When the counter decrements to zero, the timer interrupt request bit (i.e., bit 7 of the timer control register (TCR)) is set. Then, if the timer interrupt is not masked (i.e., bit 6 of the TCR and the I bit in the condition code register are both cleared) the processor receives an interrupt. After completion of the current instruction, the processor proceeds to store the appropriate registers on the stack and then fetches the timer vector address from locations $7F8 and $7F9 (or $7F6 and $7F7 if in the WAIT mode) in order to begin servicing.

The counter continues to count after it reaches zero allowing the software to determine the number of internal or external input clocks since the timer interrupt request bit was set. The counter may be read at any time by the processor without disturbing the count. The contents of the counter become stable, prior to the read portion of a cycle, and do not change during the read. The timer interrupt request bit remains set until cleared by the software. TCR7 may also be used as a scanned status bit in a non-interrupt mode of operation (TCR6 = 1).

The prescaler is a 7-bit divider which is used to extend the maximum length of the timer. Bit 0, bit 1, and bit 2 of the TCR are programmed to choose the appropriate prescaler output within the range of ÷1 to ÷128 which is used as the counter input. The processor cannot write into or read from the prescaler, however, its contents are cleared to all "0s" by the write operation into TCR when bit 3 of the written data equals one. This allows for truncation-free counting.

The timer input can be configured for three different operating modes plus a disable mode depending on the value written to the TCR4 and TCR5 control bits. Refer to the Timer Control Register section.

TIMER INPUT MODE 1

If TCR5 and TCR4 are both programmed to a "0", the input to the timer is from an internal clock and the TIMER input pin is disabled. The internal clock mode can be used for

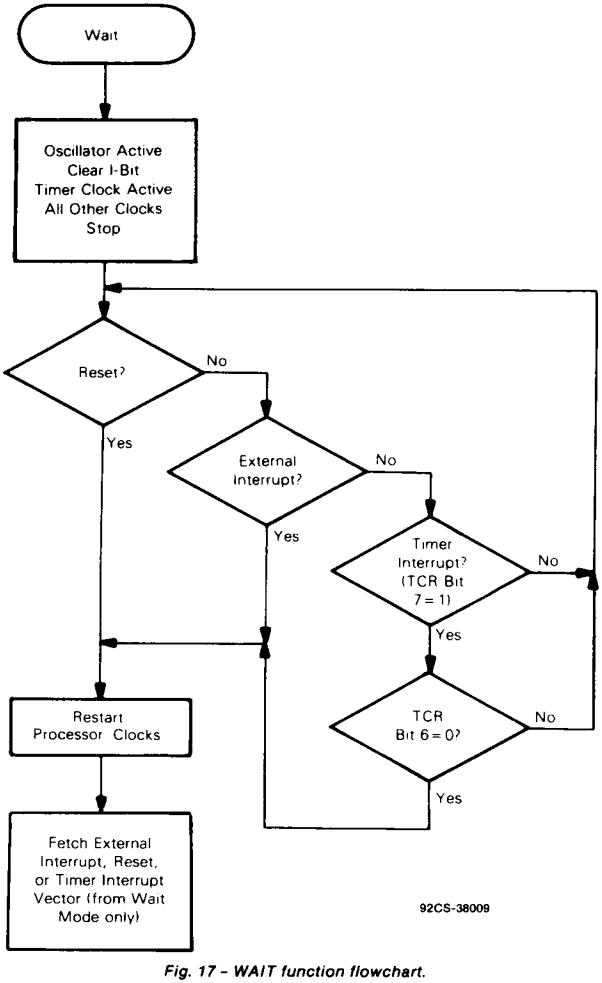


Fig. 17 - WAIT function flowchart.

periodic interrupt generation as well as a reference in frequency and event measurement. The internal clock is the instruction cycle clock. During a WAIT instruction, the internal clock to the timer continues to run at its normal rate.

TIMER INPUT MODE 2

With TCR5=0 and TCR4=1, the internal clock and the TIMER input pin are ANDed to form the timer input signal. This mode can be used to measure external pulse widths. The external timer input pulse simply turns on the internal clock for the duration of the pulse. The resolution of the count in this mode is ± one internal clock and therefore, accuracy improves with longer input pulse widths.

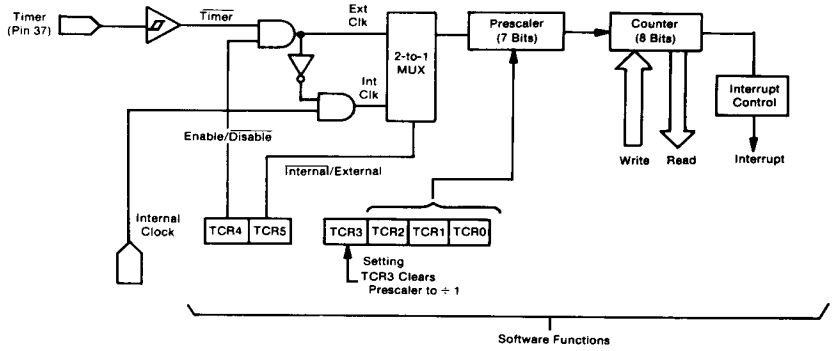
TIMER INPUT MODE 3

If TCR5=1 and TCR4=0, all inputs to the timer are disabled.

TIMER INPUT MODE 4

If TCR5=1 and TCR4=1, the internal clock input to the timer is disabled and the TIMER input pin becomes the input to the timer. The timer can, in this mode, be used to count external events as well as external frequencies for generating periodic interrupts. The counter is clocked on the falling edge of the external signal.

Figure 18 shows a block diagram of the timer subsystem. Power-on reset and the STOP instruction invalidate the contents of the counter.



NOTES:
1. Prescaler and 8-bit counter are clocked falling edge of the internal clock (AS) or external input.
2. Counter is written to during Data Strobe (DS) and counts down continuously.

92CM-38034R1

Fig. 18 - Programmable timer/counter block diagram.

TIMER CONTROL REGISTER (TCR)

7	6	5	4	3	2	1	0
TCR7	TCR6	TCR5	TCR4	TCR3	TCR2	TCR1	TCR0

All bits in this register except bit 3 are read/write bits.

TCR7 – Timer interrupt request bit: bit used to indicate the timer interrupt when it is logic "1".
1 – Set whenever the counter decrements to zero or under program control.
0 – Cleared on external $\overline{\text{RESET}}$, power-on reset, STOP instruction, or program control.

TCR6 – Timer interrupt mask bit: when this bit is a logic "1", it inhibits the timer interrupt to the processor.
1 – Set on external $\overline{\text{RESET}}$, power-on reset, STOP instruction, or program control.
0 – Cleared under program control.

TCR5 – External or internal bit: selects the input clock source to be either the external timer pin or the internal clock. (Unaffected by $\overline{\text{RESET}}$.)
1 – Select external clock source.
0 – Select internal clock source.

TCR4 – External enable bit: control bit used to enable the external TIMER pin. (Unaffected by $\overline{\text{RESET}}$.)
1 – Enable external TIMER pin.
0 – Disable external TIMER pin.

TCR5	TCR4	
0	0	Internal Clock to Timer
0	1	AND of Internal Clock and TIMER Pin to Timer
1	0	Inputs to Timer Disabled
1	1	TIMER Pin to Timer

TCR3 – Timer Prescaler Reset bit: writing a "1" to this bit resets the prescaler to zero. A read of this location always indicates "0". (Unaffected by $\overline{\text{RESET}}$.)

TCR2, TCR1, TCR0 – Prescaler select bits: decoded to select one of eight outputs on the prescaler. (Unaffected by $\overline{\text{RESET}}$.)

Prescaler

TCR2	TCR1	TCR0	Result
0	0	0	÷1
0	0	1	÷2
0	1	0	÷4
0	1	1	÷8
1	0	0	÷16
1	0	1	÷32
1	1	0	÷64
1	1	1	÷128

INSTRUCTION SET

The MCU has a set of 61 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. All the instructions within a given type are presented in individual tables.

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 operand for the jump unconditional (JMP) and jump to subroutine (JSR) instructions is the program counter. Refer to Table 4.

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. Refer to Table 5.

BRANCH INSTRUCTIONS

Most branch instructions test the state of the condition code register and, if certain criteria are met, a branch is executed. This adds an offset between −127 and +128 to the current program counter. Refer to Table 6.

BIT MANIPULATION INSTRUCTIONS

The MCU is capable of setting or clearing any bit which resides in the first 128 bytes of the memory space where all port registers, port DDRs, timer, timer control, and on-chip RAM reside. An additional feature allows the software to test and branch on the state of any bit within the first 256 locations. The bit set, bit clear, and bit test and branch functions are implemented with a single instruction. For the test and branch instructions, the value of the bit tested is also placed in the carry bit of the condition code register. Refer to Table 7.

CONTROL INSTRUCTIONS

These instructions are register reference instructions and are used to control processor operation during program execution. Refer to Table 8.

OPCODE MAP

Table 9 is an opcode map for the instructions used on the MCU.

ALPHABETICAL LISTING

The complete instruction set is given in alphabetical order in Table 10.

ADDRESSING MODES

The MCU uses ten different addressing modes to provide the programmer with an opportunity to optimize the code to 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 while the longest instructions (three bytes) permit tables throughout memory. Short and long absolute addressing is also included. Two-byte direct addressing instructions access all data bytes in most applications. Extended addressing permits jump instructions to reach all memory. Table 10 shows the addressing modes for each instruction with the effects each instruction has on the condition code register. An opcode map is shown in Table 9.

The term "Effective Address" (EA) is defined as the byte address to or from which the argument for an instruction is fetched or stored. The ten addressing modes of the processor are described below. Parentheses are used to indicate "contents of," an arrow indicates "is replaced by," and a colon indicates "concatenation of two bytes."

INHERENT

In inherent instructions, all the information necessary to execute the instruction is contained in the opcode. Operations specifying only the index registers or accumulator and no other arguments are included in this mode.

IMMEDIATE

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

$$EA = PC + 1;\ PC \leftarrow PC + 2$$

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. This includes all on-chip RAM and I/O registers and 128 bytes of on-chip ROM. Direct addressing is efficient in both memory and time.

$$EA = (PC + 1);\ PC + PC + 2$$

$$\text{Address Bus High} \leftarrow 0;\ \text{Address Bus Low} \leftarrow (PC + 1)$$

EXTENDED

In the extended addressing mode, the effective address of the argument is contained in the two bytes following the opcode. Instructions with extended addressing modes are capable of referencing arguments anywhere in memory with a single three-byte instruction.

$$EA = (PC + 1):(PC + 2);\ PC \leftarrow PC + 3$$

$$\text{Address Bus High} \leftarrow (PC + 1);\ \text{Address Bus Low} \leftarrow (PC + 2)$$

INDEXED, NO-OFFSET

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

$$EA = X;\ PC \leftarrow PC + 1$$

$$\text{Address Bus High} \leftarrow 0;\ \text{Address Bus Low} \leftarrow X$$

INDEXED, 8-BIT OFFSET

Here the EA is obtained by adding the contents of the byte following the opcode to that of the index register, therefore, the operand is located anywhere within the lowest 511 memory locations. For example, this mode of addressing is useful for selecting the mth element in an n element table. All instructions are two bytes. The content of the index register

(X) is not changed. The content of (PC + 1) is an unsigned 8-bit integer. One-byte offset indexing permits look-up tables to be easily accessed in either RAM or ROM.

EA = X + (PC + 1); PC ← PC + 2

Address Bus High ← K; Address Bus Low ← X + (PC + 1)

where K = The carry from the addition of X + (PC + 1)

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 addressing 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 (e.g., jump tables in ROM). The content of the index register is not changed.

EA = X + [(PC + 1):(PC + 2)]; PC ← PC + 3

Address Bus High ← (PC + 1) + K;

Address Bus Low ← X + (PC + 2)

where K = The carry from the addition of X + (PC + 2)

RELATIVE

Relative addressing is only used in branch instructions. In relative addressing, the contents of the 8-bit signed byte following the opcode (the offset) is added to the PC if and only if the branch condition is true. Otherwise, control proceeds to the next instruction. The span of relative addressing is limited to the range of −126 to +129 bytes from the branch instruction opcode location.

EA = PC + 2 + (PC + 1); PC ← EA if branch taken;

otherwise, PC ← PC + 2

BIT SET/CLEAR

Direct addressing and bit addressing are combined in instructions which set and clear individual memory and I/O bits. In the bit set and clear instructions, the byte is specified as a direct address in the location following the opcode. The first 128 addressable locations are thus accessed. The bit to be modified within that byte is specified with three bits of the opcode. The bit set and clear instructions occupy two bytes: one for the opcode (including the bit number) and the second for addressing the byte which contains the bit of interest.

EA = (PC + 1); PC ← PC + 2

Address Bus High ← 0; Address Bus Low ← (PC + 1)

BIT TEST AND BRANCH

Bit test and branch is a combination of direct addressing, bit addressing, and relative addressing. The bit address and condition (set or clear) to be tested is part of the opcode. The address of the byte to be tested is in the single byte immediately following the opcode byte (EA1). The signed relative 8-bit offset is in the third byte (EA2) and 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 bit in the first 256 locations of memory.

EA1 = (PC + 1)

Address Bus High ← 0; Address Bus Low ← (PC + 1)

EA2 = PC + 3 + (PC + 2); PC ← EA2 if branch taken;

otherwise, PC ← PC + 3

TABLE 4 — REGISTER/MEMORY INSTRUCTIONS

		Addressing Modes																	
		Immediate			Direct			Extended			Indexed (No Offset)			Indexed (8-Bit Offset)			Indexed (16-Bit Offset)		
Function	Mnemonic	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Load A from Memory	LDA	A6	2	2	B6	2	3	C6	3	4	F6	1	3	E6	2	4	D6	3	5
Load X from Memory	LDX	AE	2	2	BE	2	3	CE	3	4	FE	1	3	EE	2	4	DE	3	5
Store A in Memory	STA	–	–	–	B7	2	4	C7	3	5	F7	1	4	E7	2	5	D7	3	6
Store X in Memory	STX	–	–	–	BF	2	4	CF	3	5	FF	1	4	EF	2	5	DF	3	6
Add Memory to A	ADD	AB	2	2	BB	2	3	CB	3	4	FB	1	3	EB	2	4	DB	3	5
Add Memory and Carry to A	ADC	A9	2	2	B9	2	3	C9	3	4	F9	1	3	E9	2	4	D9	3	5
Subtract Memory	SUB	A0	2	2	B0	2	3	C0	3	4	F0	1	3	E0	2	4	D0	3	5
Subtract Memory from A with Borrow	SBC	A2	2	2	B2	2	3	C2	3	4	F2	1	3	E2	2	4	D2	3	5
AND Memory to A	AND	A4	2	2	B4	2	3	C4	3	4	F4	1	3	E4	2	4	D4	3	5
OR Memory with A	ORA	AA	2	2	BA	2	3	CA	3	4	FA	1	3	EA	2	4	DA	3	5
Exclusive OR Memory with A	EOR	A8	2	2	B8	2	3	C8	3	4	F8	1	3	E8	2	4	D8	3	5
Arithmetic Compare A with Memory	CMP	A1	2	2	B1	2	3	C1	3	4	F1	1	3	E1	2	4	D1	3	5
Arithmetic Compare X with Memory	CPX	A3	2	2	B3	2	3	C3	3	4	F3	1	3	E3	2	4	D3	3	5
Bit Test Memory with A (Logical Compare)	BIT	A5	2	2	B5	2	3	C5	3	4	F5	1	3	E5	2	4	D5	3	5
Jump Unconditional	JMP	–	–	–	BC	2	2	CC	3	3	FC	1	2	EC	2	3	DC	3	4
Jump to Subroutine	JSR	–	–	–	BD	2	5	CD	3	6	FD	1	5	ED	2	6	DD	3	7

TABLE 5 — READ-MODIFY-WRITE INSTRUCTIONS

		Addressing Modes														
		Inherent (A)			Inherent (X)			Direct			Indexed (No Offset)			Indexed (8-Bit Offset)		
Function	Mnemonic	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Increment	INC	4C	1	3	5C	1	3	3C	2	5	7C	1	5	6C	2	6
Decrement	DEC	4A	1	3	5A	1	3	3A	2	5	7A	1	5	6A	2	6
Clear	CLR	4F	1	3	5F	1	3	3F	2	5	7F	1	5	6F	2	6
Complement	COM	43	1	3	53	1	3	33	2	5	73	1	5	63	2	6
Negate (2's Complement)	NEG	40	1	3	50	1	3	30	2	5	70	1	5	60	2	6
Rotate Left Thru Carry	ROL	49	1	3	59	1	3	39	2	5	79	1	5	69	2	6
Rotate Right Thru Carry	ROR	46	1	3	56	1	3	36	2	5	76	1	5	66	2	6
Logical Shift Left	LSL	48	1	3	58	1	3	38	2	5	78	1	5	68	2	6
Logical Shift Right	LSR	44	1	3	54	1	3	34	2	5	74	1	5	64	2	6
Arithmetic Shift Right	ASR	47	1	3	57	1	3	37	2	5	77	1	5	67	2	6
Test for Negative or Zero	TST	4D	1	3	5D	1	3	3D	2	4	7D	1	4	6D	2	5

TABLE 6 — BRANCH INSTRUCTIONS

Function	Mnemonic	Relative Addressing Mode		
		Op Code	# Bytes	# Cycles
Branch Always	BRA	20	2	3
Branch Never	BRN	21	2	3
Branch IFF Higher	BHI	22	2	3
Branch IFF Lower or Same	BLS	23	2	3
Branch IFF Carry Clear	BCC	24	2	3
(Branch IFF Higher or Same)	(BHS)	24	2	3
Branch IFF Carry Set	BCS	25	2	3
(Branch IFF Lower)	(BLO)	25	2	3
Branch IFF Not Equal	BNE	26	2	3
Branch IFF Equal	BEQ	27	2	3
Branch IFF Half Carry Clear	BHCC	28	2	3
Branch IFF Half Carry Set	BHCS	29	2	3
Branch IFF Plus	BPL	2A	2	3
Branch IFF Minus	BMI	2B	2	3
Branch IFF Interrupt Mask Bit is Clear	BMC	2C	2	3
Branch IFF Interrupt Mask Bit is Set	BMS	2D	2	3
Branch IFF Interrupt Line is Low	BIL	2E	2	3
Branch IFF Interrupt Line is High	BIH	2F	2	3
Branch to Subroutine	BSR	AD	2	6

TABLE 7 — BIT MANIPULATION INSTRUCTIONS

Function	Mnemonic	Addressing Modes					
		Bit Set/Clear			Bit Test and Branch		
		Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Branch IFF Bit n is Set	BRSET n (n=0...7)	–	–	–	2•n	3	5
Branch IFF Bit n is Clear	BRCLR n (n=0...7)	–	–	–	01+2•n	3	5
Set Bit n	BSET n (n=0...7)	10+2•n	2	5	–	–	–
Clear Bit n	BCLR n (n=0...7)	11+2•n	2	5	–	–	–

TABLE 8 — CONTROL INSTRUCTIONS

Function	Mnemonic	Inherent		
		Op Code	# Bytes	# Cycles
Transfer A to X	TAX	97	1	2
Transfer X to A	TXA	9F	1	2
Set Carry Bit	SEC	99	1	2
Clear Carry Bit	CLC	98	1	2
Set Interrupt Mask Bit	SEI	9B	1	2
Clear Interrupt Mask Bit	CLI	9A	1	2
Software Interrupt	SWI	83	1	10
Return from Subroutine	RTS	81	1	6
Return from Interrupt	RTI	80	1	9
Reset Stack Pointer	RSP	9C	1	2
No-Operation	NOP	9D	1	2
Stop	STOP	8E	1	2
Wait	WAIT	8F	1	2

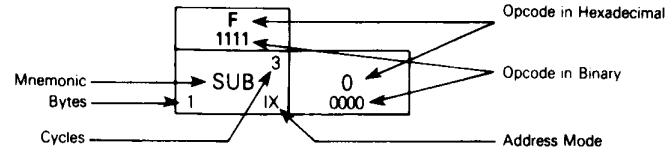
TABLE 9– INSTRUCTION SET OPCODE MAP

	Bit Manipulation		Branch	Read-Modify-Write					Control		Register/Memory						
	BTB 0 0000	BSC 1 0001	REL 2 0010	DIR 3 0011	INH 4 0100	INH 5 0101	IX1 6 0110	IX 7 0111	INH 8 1000	INH 9 1001	IMM A 1010	DIR B 1011	EXT C 1100	IX2 D 1101	IX1 E 1110	IX F 1111	
0 0000	BRSET0 5 3 BTB	BSET0 5 2 BSC	BRA 3 2 REL	NEG 5 2 DIR	NEG 3 1 INH	NEG 3 1 INH	NEG 6 2 IX1	NEG 5 1 IX	RTI 9 1 INH *		SUB 2 2 IMM	SUB 3 2 DIR	SUB 4 3 EXT	SUB 5 3 IX2	SUB 4 2 IX1	SUB 3 1 IX	0 0000
1 0001	BRCLR0 5 3 BTB	BCLR0 5 2 BSC	BRN 3 2 REL						RTS 6 1 INH		CMP 2 2 IMM	CMP 3 2 DIR	CMP 4 3 EXT	CMP 5 3 IX2	CMP 4 2 IX1	CMP 3 1 IX	1 0001
2 0010	BRSET1 5 3 BTB	BSET1 5 2 BSC	BHI 3 2 REL								SBC 2 2 IMM	SBC 3 2 DIR	SBC 4 3 EXT	SBC 5 3 IX2	SBC 4 2 IX1	SBC 3 1 IX	2 0010
3 0011	BRCLR1 5 3 BTB	BCLR1 5 2 BSC	BLS 3 2 REL	COM 5 2 DIR	COMA 3 1 INH	COMX 3 1 INH	COM 6 2 IX1	COM 5 1 IX	SWI 10 1 INH		CPX 2 2 IMM	CPX 3 2 DIR	CPX 4 3 EXT	CPX 5 3 IX2	CPX 4 2 IX1	CPX 3 1 IX	3 0011
4 0100	BRSET2 5 3 BTB	BSET2 5 2 BSC	BCC 3 2 REL	LSR 5 2 DIR	LSRA 3 1 INH	LSRX 3 1 INH	LSR 6 2 IX1	LSR 5 1 IX			AND 2 2 IMM	AND 3 2 DIR	AND 4 3 EXT	AND 5 3 IX2	AND 4 2 IX1	AND 3 1 IX	4 0100
5 0101	BRCLR2 5 3 BTB	BCLR2 5 2 BSC	BCS 3 2 REL								BIT 2 2 IMM	BIT 3 2 DIR	BIT 4 3 EXT	BIT 5 3 IX2	BIT 4 2 IX1	BIT 3 1 IX	5 0101
6 0110	BRSET3 5 3 BTB	BSET3 5 2 BSC	BNE 3 2 REL	ROR 5 2 DIR	RORA 3 1 INH	RORX 3 1 INH	ROR 6 2 IX1	ROR 5 1 IX			LDA 2 2 IMM	LDA 3 2 DIR	LDA 4 3 EXT	LDA 5 3 IX2	LDA 4 2 IX1	LDA 3 1 IX	6 0110
7 0111	BRCLR3 5 3 BTB	BCLR3 5 2 BSC	BEQ 3 2 REL	ASR 5 2 DIR	ASRA 3 1 INH	ASRX 3 1 INH	ASR 6 2 IX1	ASR 5 1 IX		TAX 2 1 INH		STA 4 2 DIR	STA 5 3 EXT	STA 6 3 IX2	STA 5 2 IX1	STA 4 1 IX	7 0111
8 1000	BRSET4 5 3 BTB	BSET4 5 2 BSC	BHCC 3 2 REL	LSL 5 2 DIR	LSLA 3 1 INH	LSLX 3 1 INH	LSL 6 2 IX1	LSL 5 1 IX		CLC 2 1 INH	EOR 2 2 IMM	EOR 3 2 DIR	EOR 4 3 EXT	EOR 5 3 IX2	EOR 4 2 IX1	EOR 3 1 IX	8 1000
9 1001	BRCLR4 5 3 BTB	BCLR4 5 2 BSC	BHCS 3 2 REL	ROL 5 2 DIR	ROLA 3 1 INH	ROLX 3 1 INH	ROL 6 2 IX1	ROL 5 1 IX		SEC 2 1 INH	ADC 2 2 IMM	ADC 3 2 DIR	ADC 4 3 EXT	ADC 5 3 IX2	ADC 4 2 IX1	ADC 3 1 IX	9 1001
A 1010	BRSET5 5 3 BTB	BSET5 5 2 BSC	BPL 3 2 REL	DEC 5 2 DIR	DECA 3 1 INH	DECX 3 1 INH	DEC 6 2 IX1	DEC 5 1 IX		CLI 2 1 INH	ORA 2 2 IMM	ORA 3 2 DIR	ORA 4 3 EXT	ORA 5 3 IX2	ORA 4 2 IX1	ORA 3 1 IX	A 1010
B 1011	BRCLR5 5 3 BTB	BCLR5 5 2 BSC	BMI 3 2 REL							SEI 2 1 INH	ADD 2 2 IMM	ADD 3 2 DIR	ADD 4 3 EXT	ADD 5 3 IX2	ADD 4 2 IX1	ADD 3 1 IX	B 1011
C 1100	BRSET6 5 3 BTB	BSET6 5 2 BSC	BMC 3 2 REL	INC 5 2 DIR	INCA 3 1 INH	INCX 3 1 INH	INC 6 2 IX1	INC 5 1 IX		RSP 2 1 INH		JMP 2 2 DIR	JMP 3 3 EXT	JMP 4 3 IX2	JMP 3 2 IX1	JMP 2 1 IX	C 1100
D 1101	BRCLR6 5 3 BTB	BCLR6 5 2 BSC	BMS 3 2 REL	TST 4 2 DIR	TSTA 3 1 INH	TSTX 3 1 INH	TST 5 2 IX1	TST 4 1 IX		NOP 2 1 INH	BSR 6 2 REL	JSR 5 2 DIR	JSR 6 3 EXT	JSR 7 3 IX2	JSR 6 2 IX1	JSR 5 1 IX	D 1101
E 1110	BRSET7 5 3 BTB	BSET7 5 2 BSC	BIL 3 2 REL						STOP 2 1 INH		LDX 2 2 IMM	LDX 3 2 DIR	LDX 4 3 EXT	LDX 5 3 IX2	LDX 4 2 IX1	LDX 3 1 IX	E 1110
F 1111	BRCLR7 5 3 BTB	BCLR7 5 2 BSC	BIH 3 2 REL	CLR 5 2 DIR	CLRA 3 1 INH	CLRX 3 1 INH	CLR 6 2 IX1	CLR 5 1 IX	WAIT 2 1 INH	TXA 2 1 INH		STX 4 2 DIR	STX 5 3 EXT	STX 6 3 IX2	STX 5 2 IX1	STX 4 1 IX	F 1111

Abbreviations for Address Modes

INH	Inherent
IMM	Immediate
DIR	Direct
EXT	Extended
REL	Relative
BSC	Bit Set/Clear
BTB	Bit Test and Branch
IX	Indexed (No Offset)
IX1	Indexed, 1 Byte (8-Bit) Offset
IX2	Indexed, 2 Byte (16-Bit) Offset

LEGEND



92CS-38011

CDP6805F2, CDP6805F2C

TABLE 10 — INSTRUCTION SET

Mnemonic	Addressing Modes										Condition Codes				
	Inherent	Immediate	Direct	Extended	Relative	Indexed (No Offset)	Indexed (8 Bits)	Indexed (16 Bits)	Bit Set/ Clear	Bit Test & Branch	H	I	N	Z	C
ADC		X	X	X		X	X	X			↕	•	↕	↕	↕
ADD		X	X	X		X	X	X			↕	•	↕	↕	↕
AND		X	X	X		X	X	X			•	•	↕	↕	•
ASL	X		X			X	X				•	•	↕	↕	↕
ASR	X		X			X	X				•	•	↕	↕	↕
BCC					X						•	•	•	•	•
BCLR									X		•	•	•	•	•
BCS					X						•	•	•	•	•
BEQ					X						•	•	•	•	•
BHCC					X						•	•	•	•	•
BHCS					X						•	•	•	•	•
BHI					X						•	•	•	•	•
BHS					X						•	•	•	•	•
BIH					X						•	•	•	•	•
BIL					X						•	•	•	•	•
BIT		X	X	X		X	X	X			•	•	↕	↕	•
BLO					X						•	•	•	•	•
BLS					X						•	•	•	•	•
BMC					X						•	•	•	•	•
BMI					X						•	•	•	•	•
BMS					X						•	•	•	•	•
BNE					X						•	•	•	•	•
BPL					X						•	•	•	•	•
BRA					X						•	•	•	•	•
BRN					X						•	•	•	•	•
BRCLR										X	•	•	•	↕	↕
BRSET										X	•	•	•	↕	↕
BSET									X		•	•	•	•	•
BSR					X						•	•	•	•	•
CLC	X										•	•	•	•	0
CLI	X										•	0	•	•	•
CLR	X		X			X	X				•	•	0	1	•
CMP		X	X	X		X	X	X			•	•	↕	↕	↕
COM	X		X			X	X				•	•	↕	↕	1
CPX		X	X	X		X	X	X			•	•	↕	↕	↕
DEC	X		X			X	X				•	•	↕	↕	•
EOR		X	X	X		X	X	X			•	•	↕	↕	•
INC	X		X			X	X				•	•	↕	↕	•
JMP			X	X		X	X	X			•	•	•	•	•
JSR			X	X		X	X	X			•	•	•	•	•
LDA		X	X	X		X	X	X			•	•	↕	↕	•
LDX		X	X	X		X	X	X			•	•	↕	↕	•
LSL	X		X			X	X				•	•	↕	↕	↕
LSR	X		X			X	X				•	•	0	↕	↕
NEG	X		X			X	X				•	•	↕	↕	↕
NOP	X										•	•	•	•	•
ORA		X	X	X		X	X	X			•	•	↕	↕	•
ROL	X		X			X	X				•	•	↕	↕	↕
ROR	X		X			X	X				•	•	↕	↕	↕
RSP	X										•	•	•	•	•
RTI	X										?	?	?	?	?
RTS	X										•	•	•	•	•
SBC		X	X	X		X	X	X			•	•	↕	↕	↕
SEC	X										•	•	•	•	1
SEI	X										•	1	•	•	•
STA			X	X		X	X	X			•	•	↕	↕	•
STOP	X										•	0	•	•	•
STX			X	X		X	X	X			•	•	↕	↕	•
SUB		X	X	X		X	X	X			•	•	↕	↕	↕
SWI	X										•	1	•	•	•
TAX	X										•	•	•	•	•
TST	X		X			X	X				•	•	↕	↕	•
TXA	X										•	•	•	•	•
WAIT	X										•	0	•	•	•

Condition Code Symbols

- H Half Carry (From Bit 3)
- I Interrupt Mask
- N Negative (Sign Bit)
- Z Zero
- C Carry/Borrow
- ↕ Test and Set if True, Cleared Otherwise
- • Not Affected
- ? Load CC Register From Stack
- 0 Cleared
- 1 Set