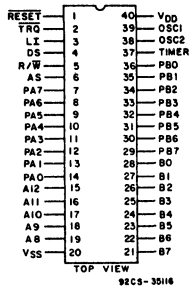
CDP6805E2, CDP6805E2C



TERMINAL ASSIGNMENT

CMOS 8-Bit Microprocessor

Hardware Features:

- *Typical full speed operating power of 35 mW @ 5 V*
- *Typical WAIT mode power of 5 mW*
- *Typical STOP mode power of 25 μW*
- *112 bytes of on-chip RAM*
- *16 bidirectional I/O lines*
- *Internal 8-bit timer with software programmable 7-bit prescaler*
- *External timer input*
- *Full external and timer interrupts*
- *Multiplexed address/data bus*
- *Master reset and power-on reset*
- *Capable of addressing up to 8K bytes of external memory*
- *Single 3- to 6-volt supply*
- *On-chip oscillator*
- *40-pin dual-in-line package*
- *44-lead plastic chip-carrier package*

The CDP6805E2 Microprocessor Unit (MPU) belongs to the CDP6805 Family of CMOS Microcomputers. This 8-bit fully static and expandable microprocessor contains a CPU, on-chip RAM, I/O, and Timer. It is a low-power, low-cost processor designed for mid-range applications in the consumer, automotive, industrial, and communications markets where very low power consumption constitutes an important factor. The following are the major features of the CDP6805E2 MPU.

Software Features:

- *Efficient use of program space*
- *Versatile interrupt handling*
- *True bit manipulation*
- *Addressing modes with indexed addressing for tables*
- *Efficient instruction set*
- *Memory mapped I/O*
- *Two power saving standby modes*

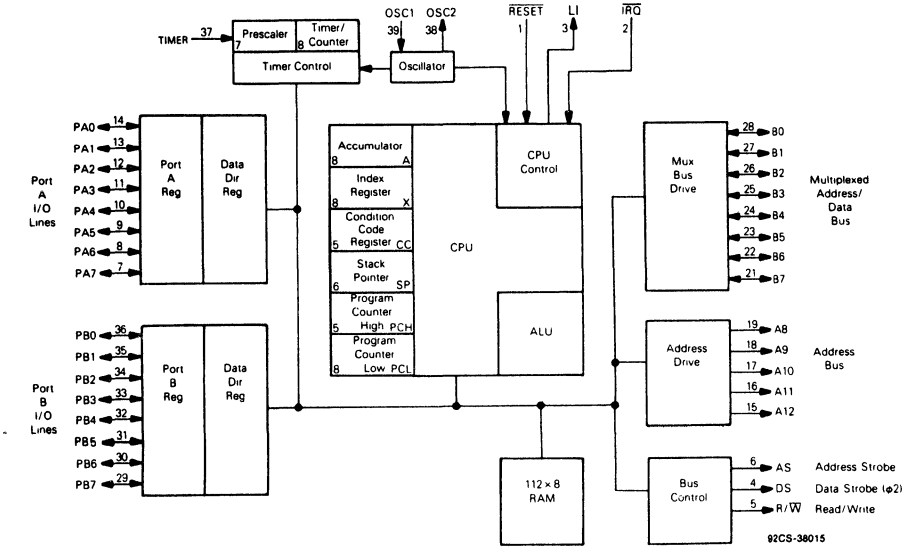


Fig. 1 - Block diagram.

CDP6805E2, CDP6805E2C

MAXIMUM RATINGS (voltages referenced to V_{SS})

Ratings	Symbol	Value	Unit
Supply Voltage	V_{DD}	−0.3 to +8.0	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 CDP6805E2 CDP6805E2C	T_A	T_L to T_H 0 to 70 −40 to 85	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C

DC ELECTRICAL CHARACTERISTICS 3.0 V (V_{DD}=3 Vdc, V_{SS}=0, T_A=T_L to T_H, unless otherwise noted)

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
Total Supply Current (C_L=50 pF − no DC loads) t_{cyc}=5 μs				
Run (V_{IL}=0.2 V, V_{IH}=V_{DD}−0.2 V)	I_{DD}	−	1.3	mA
Wait (Test Conditions − See Note Below)	I_{DD}	−	200	μA
Stop (Test Conditions − See Note Below)	I_{DD}	−	100	μA
Output High Voltage				
(I_{LOAD}=0.25 mA) A8-A12,B0-B7	V_{OH}	2.7	−	V
(I_{LOAD}=0.1 mA) PA0-PA7, PB0-PB7	V_{OH}	2.7	−	V
(I_{LOAD}=0.25 mA) DS, AS, R/$\overline{W}$	V_{OH}	2.7	−	V
Output Low Voltage				
(I_{LOAD}=0.25 mA) A8-A12, B0-B7	V_{OL}	−	0.3	V
(I_{LOAD}=0.25 mA) PA0-PA7, PB0-PB7	V_{OL}	−	0.3	V
(I_{LOAD}=0.25 mA) DS, AS, R/$\overline{W}$	V_{OL}	−	0.3	V
Input High Voltage				
PA0-PA7, PB0-PB7, B0-B7	V_{IH}	2.1	−	V
TIMER, $\overline{IRQ}$, $\overline{RESET}$	V_{IH}	2.5	−	V
OSC1	V_{IH}	2.1	−	V
Input Low Voltage (All inputs)	V_{IL}	−	0.5	V
Frequency of Operation				
Crystal	f_{OSC}	0.032	1.0	MHz
External Clock	f_{OSC}	DC	1.0	MHz
Input Current $\overline{RESET}$, $\overline{IRQ}$, Timer, OSC1	I_{in}	−	±1	μA
Three-State Output Leakage PA0-OA7, PB0-PB7, B0-B7	I_{TSL}	−	±10	μA
Capacitance $\overline{RESET}$, $\overline{IRQ}$, Timer	C_{in}	−	8.0	pF
Capacitance DS, AS, R/$\overline{W}$, A8-A12, PA0-PA7, PB0-PB7, B0-B7	C_{out}	−	12.0	pF

NOTE Test conditions for Quiescent Current Values are:
- Port A and B programmed as inputs.
- V_{IL}=0.2 V for PA0-PA7, PB0-PB7, and B0-B7
- V_{IH}=V_{DD} − 0.2 V for $\overline{RESET}$, $\overline{IRQ}$, and Timer.
- OSC1 input is a squarewave from V_{SS}+0.2 V to V_{DD} − 0.2 V.
- OSC2 output load (including tester) is 35 pF maximum.
- Wait mode I_{DD} is affected linearly by this capacitance

CDP6805E2, CDP6805E2C

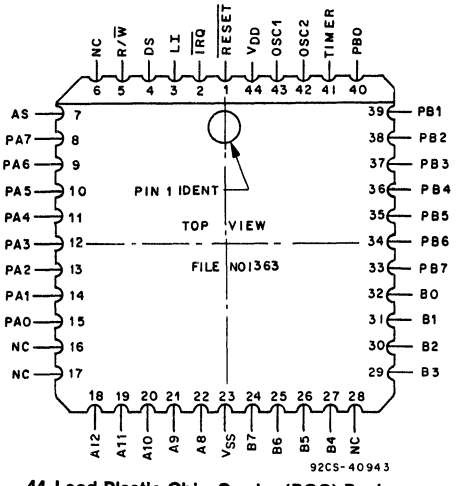
DC ELECTRICAL CHARACTERISTICS 5.0 V (V_{DD}=5 Vdc ± 10%, V_{SS}=0, T_A=T_L to T_H, unless otherwise noted)

Characteristics	Symbol	Min	Max	Unit
Output Voltage $I_{LOAD} \leq 10.0\ \mu A$	V_{OL}	–	0.1	V
	V_{OH}	V_{DD} – 0.1	–	V
Total Supply Current (C_L = 130 pF – On Bus, C_L = 50 pF – On Ports, No DC Loads, t_{cyc} = 1.0 µs				
Run (V_{IL} = 0.2 V, V_{IH} = V_{DD} – 0.2 V)	I_{DD}	–	10	mA
Wait (Test Conditions – See Note Below)	I_{DD}	–	1.5	mA
Stop (Test Conditions – See Note Below)	I_{DD}	–	200	µA
Output High Voltage				
(I_{LOAD} = 1.6 mA) A8-A12, B0-B7	V_{OH}	4.1	–	V
(I_{LOAD} = 0.36 mA) PA0-PA7, PB0-PB7	V_{OH}	4.1	–	V
(I_{LOAD} = 1.6 mA) DS, AS, R/$\overline{W}$	V_{OH}	4.1	–	V
Output Low Voltage				
(I_{LOAD} = 1.6 mA) A8-A12, B0-B7	V_{OL}	–	0.4	V
(I_{LOAD} = 1.6 mA) PA0-PA7, PB0-PB7	V_{OL}	–	0.4	V
(I_{LOAD} = 1.6 mA) DS, AS, R/$\overline{W}$	V_{OL}	–	0.4	V
Input High Voltage				
PA0-PA7, PB0-PB7, B0-B7	V_{IH}	V_{DD} – 2.0	–	V
TIMER, $\overline{IRQ}$, $\overline{RESET}$	V_{IH}	V_{DD} – 0.8	–	V
OSC1	V_{IH}	V_{DD} – 1.5	–	V
Input Low Voltage (All Inputs)	V_{IL}	–	0.8	V
Frequency of Operation				
Crystal	f_{OSC}	0.032	5.0	MHz
External Clock	f_{OSC}	DC	5.0	MHz
Input Current				
$\overline{RESET}$, $\overline{IRQ}$, Timer, OSC1	I_{in}	–	±1	µA
Three-State Output Leakage				
PA0-PA7, PB0-PB7, B0-B7	I_{TSI}	–	±10	µA
Capacitance				
$\overline{RESET}$, $\overline{IRQ}$, Timer	C_{in}	–	8.0	pF
Capacitance				
DS, AS, R/$\overline{W}$, A8-A12, PA0-PA7, PB0-PB7, B0-B7	C_{out}	–	12.0	pF

NOTE Test conditions for Quiescent Current Values are
Port A and B programmed as inputs
V_{IL} = 0.2 V for PA0-PA7, PB0-PB7, and B0-B7
V_{IH} = V_{DD} – 0.2 V for $\overline{RESET}$, $\overline{IRQ}$, and Timer

OSC1 input is a squarewave from V_{SS} + 0.2 V to V_{DD} – 0.2 V
OSC2 output load (including tester) is 35 pF maximum
Wait mode (I_{DD}) is affected linearly by this capacitance

TERMINAL ASSIGNMENT



44-Lead Plastic Chip-Carrier (PCC) Package (Q suffix)

CDP6805E2, CDP6805E2C

TABLE 1 — CONTROL TIMING (V_{SS}=0, T_A=T_L to T_H)

		V_{DD}=3 V f_{OSC}=1 MHz			V_{DD}=5 V ± 10% f_{OSC}=5 MHz			
Characteristics	**Symbol**	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Unit**
I/O Port Timing — Input Setup Time (Figure 3)	t_{PVASL}	500	–	–	250	–	–	ns
Input Hold Time (Figure 3)	t_{ASLPX}	100	–	–	100	–	–	ns
Output Delay Time (Figure 3)	t_{ASLPV}	–	–	0	–	–	0	ns
Interrupt Setup Time (Figure 6)	t_{ILASL}	2	–	–	0.4	–	–	µs
Crystal Oscillator Startup Time (Figure 5)	t_{OXOV}	–	30	300	–	15	100	ms
Wait Recovery Startup Time (Figure 7)	t_{IVASH}	–	–	10	–	–	2	µs
Stop Recovery Startup Time (Crystal Oscillator) (Figure 8)	t_{ILASH}	–	30	300	–	15	100	ms
Required Interrupt Release (Figure 6)	t_{DSLIH}	–	–	5	–	–	1 0	µs
Timer Pulse Width (Figure 7)	t_{TH}, t_{TL}	0.5	–	–	0.5	–	–	t_{cyc}
Reset Pulse Width (Figure 5)	t_{RL}	5.2	–	–	1 05	–	–	µs
Timer Period (Figure 7)	t_{TLTL}	1.0	–	–	1 0	–	–	t_{cyc}
Interrupt Pulse Width Low (Figure 16)	t_{ILIH}	1 0	–	–	1 0	–	–	t_{cyc}
Interrupt Pulse Period (Figure 16)	t_{ILIL}	*	–	–	*	–	–	t_{cyc}
Oscillator Cycle Period (1/5 of t_{cyc})	t_{OLOL}	1000	–	–	200	–	–	ms
OSC1 Pulse Width High	t_{OH}	350	–	–	75	–	–	ns
OSC1 Pulse Width Low	t_{OL}	350	–	–	75	–	–	ns

* The minimum 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.

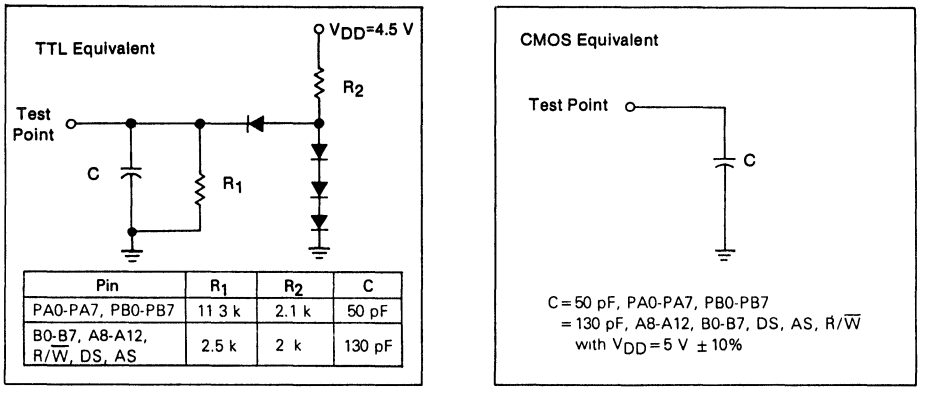
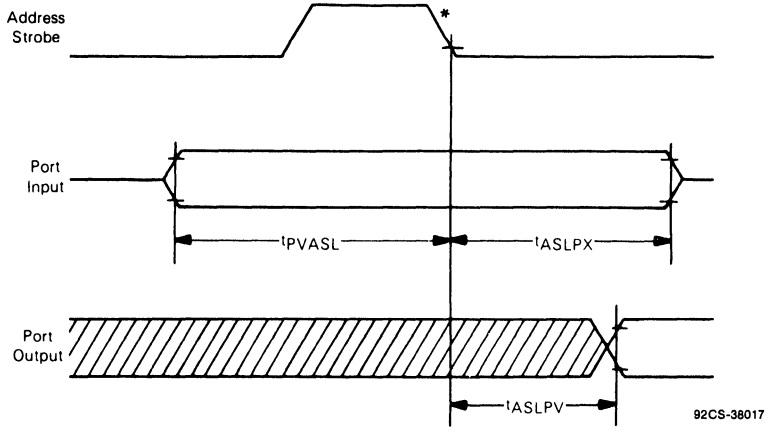


Fig. 2 - Equivalent test-load circuits.

(V_{LOW}=0.8 V, V_{HIGH}=V_{DD}−2 V, V_{DD}=5 ±10%
Temp=0° to 70°C, C_L on Port=50 pF, f_{OSC}=5 MHz)

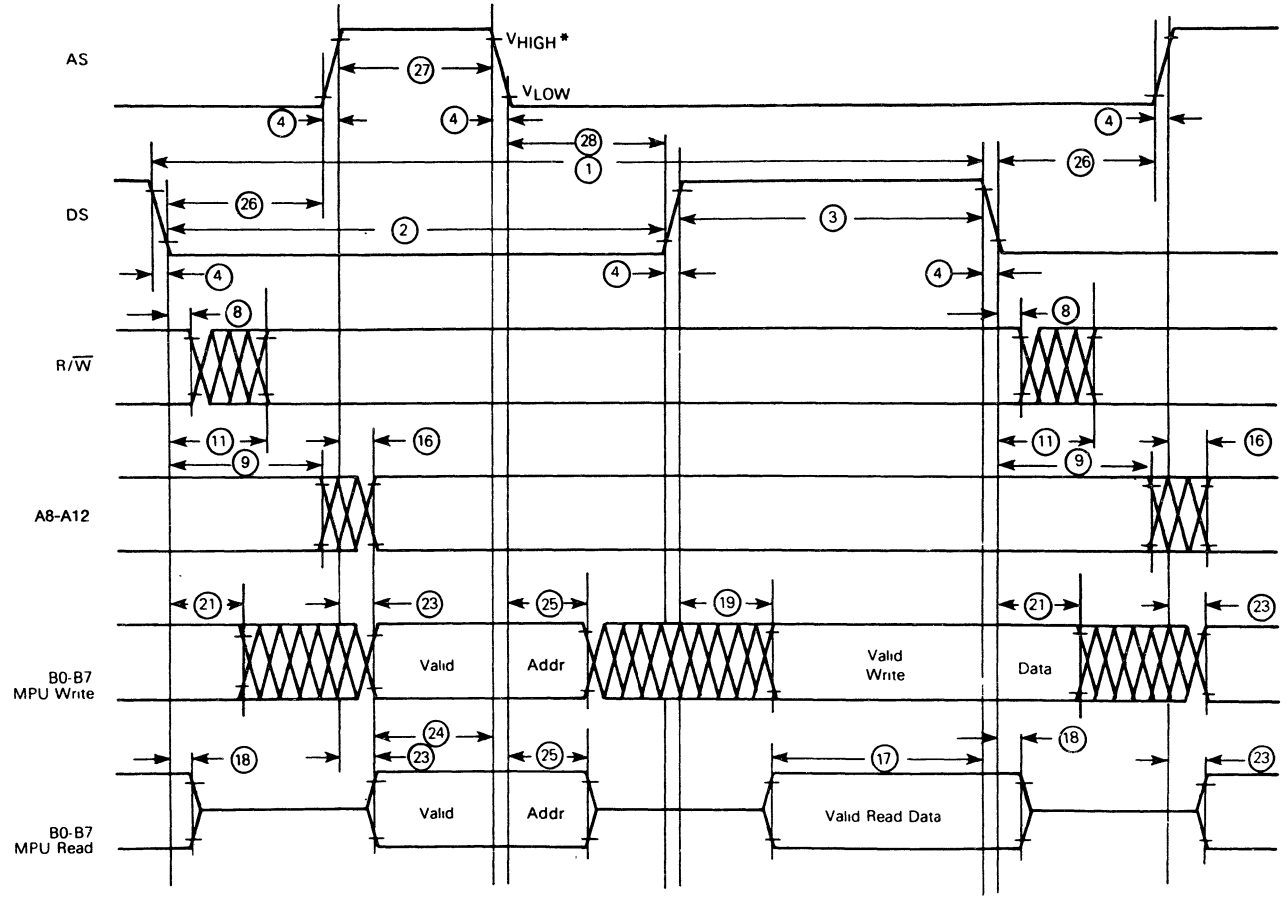


*The address strobe of the first cycle of the next instruction as shown in Table 11

Fig. 3 - I/O port timing waveforms.

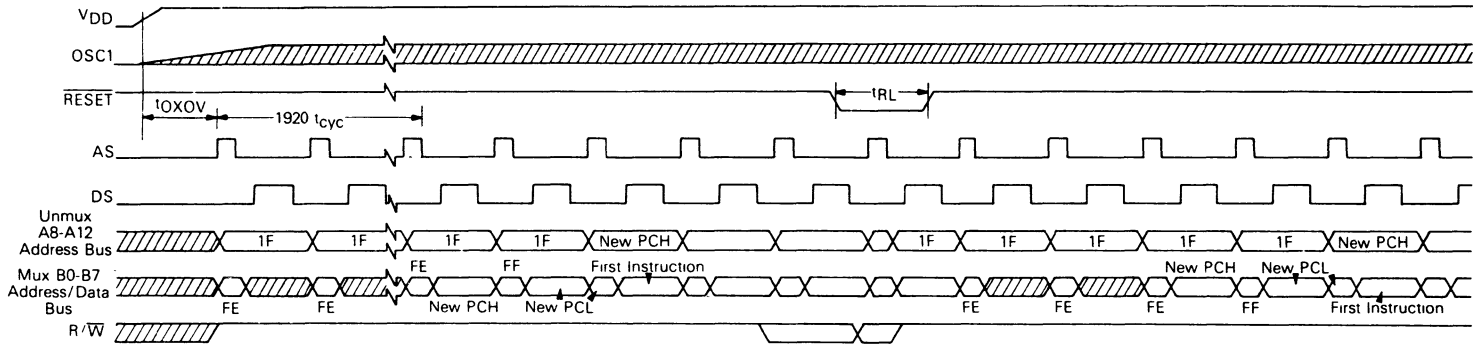
TABLE 2 — BUS TIMING (T_A=T_L to T_H, V_{SS}=0 V) See Figure 4

Num	Characteristics	Symbol	f_{OSC}=1 MHz, V_{DD}=3 V 50 pF Load		f_{OSC}=5 MHz V_{DD}=5 V ±10%, 1 TTL and 130 pF Load		Unit
			Min	Max	Min	Max	
1	Cycle Time	t_{cyc}	5000	DC	1000	DC	ns
2	Pulse Width, DS Low	PW_{EL}	2800	–	560	–	ns
3	Pulse Width, DS High or $\overline{RD}$, $\overline{WR}$, Low	PW_{EH}	1800	–	375	–	ns
4	Clock Transition	t_r, t_f	–	100	–	30	ns
8	R/$\overline{W}$ Hold	t_{RWH}	10	–	10	–	ns
9	Non-Muxed Address Hold	t_{AH}	800	–	100	–	ns
11	R/$\overline{W}$ Delay from DS Fall	t_{AD}	–	500	–	300	ns
16	Non-Muxed Address Delay from AS Rise	t_{ADH}	0	200	0	100	ns
17	MPU Read Data Setup	t_{DSR}	200	–	115	–	ns
18	Read Data Hold	t_{DHR}	0	1000	0	160	ns
19	MPU Data Delay, Write	t_{DDW}	–	0	–	120	ns
21	Write Data Hold	t_{DHW}	800	–	55	–	ns
23	Muxed Address Delay from AS Rise	t_{BHD}	0	250	0	120	ns
24	Muxed Address Valid to AS Fall	t_{ASL}	600	–	55	–	ns
25	Muxed Address Hold	t_{AHL}	250	750	60	180	ns
26	Delay DS Fall to AS Rise	t_{ASD}	800	–	160	–	ns
27	Pulse Width, AS High	PW_{ASH}	850	–	175	–	ns
28	Delay, AS Fall to DS Rise	t_{ASED}	800	–	160	–	ns

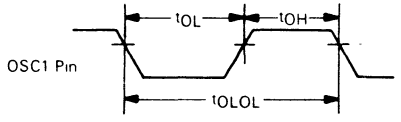


* V_{HIGH} = -2 V, V_{LOW} = 0.5 V for V_{DD} = 3 V
V_{HIGH} = V_{DD} -2 V, V_{LOW} = 0.8 V for V_{DD} = 5 V ± 10 %

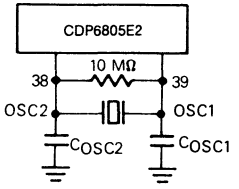
Fig. 4 - Bus timing waveforms.



Oscillator Waveform



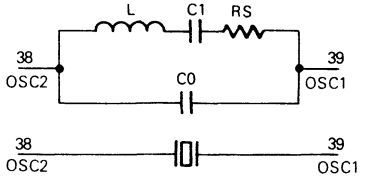
Crystal Oscillator Connections



Crystal Parameters Representative Frequencies

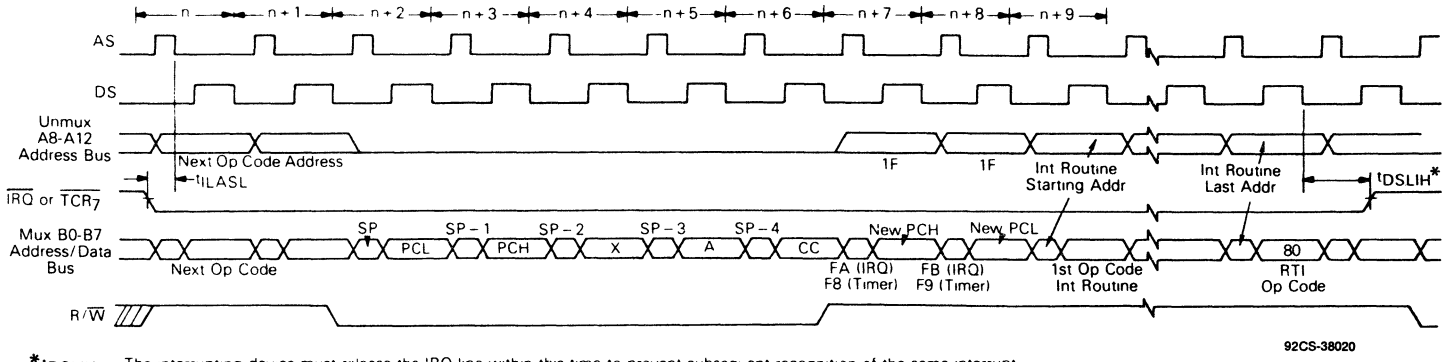
	5 MHz	4 MHz	1 MHz
RS max	50Ω	75Ω	400Ω
C0	8 pF	7 pF	5 pF
C1	0 02 pF	0 012 pF	0 008 pF
Q	50 k	40 k	30 k
C_{OSC1}	15-30 pF	15-30 pF	15-40 pF
C_{OSC2}	15-25 pF	15-25 pF	15-30 pF

Crystal Circuit



92CS-38019

Fig. 5 - Power-on reset and $\overline{\text{reset}}$ timing waveforms.



*t_{DSLIH} – The interrupting device must release the IRQ line within this time to prevent subsequent recognition of the same interrupt

Fig. 6 - $\overline{IRQ}$ and $\overline{TCR_7}$ interrupt timing waveforms.

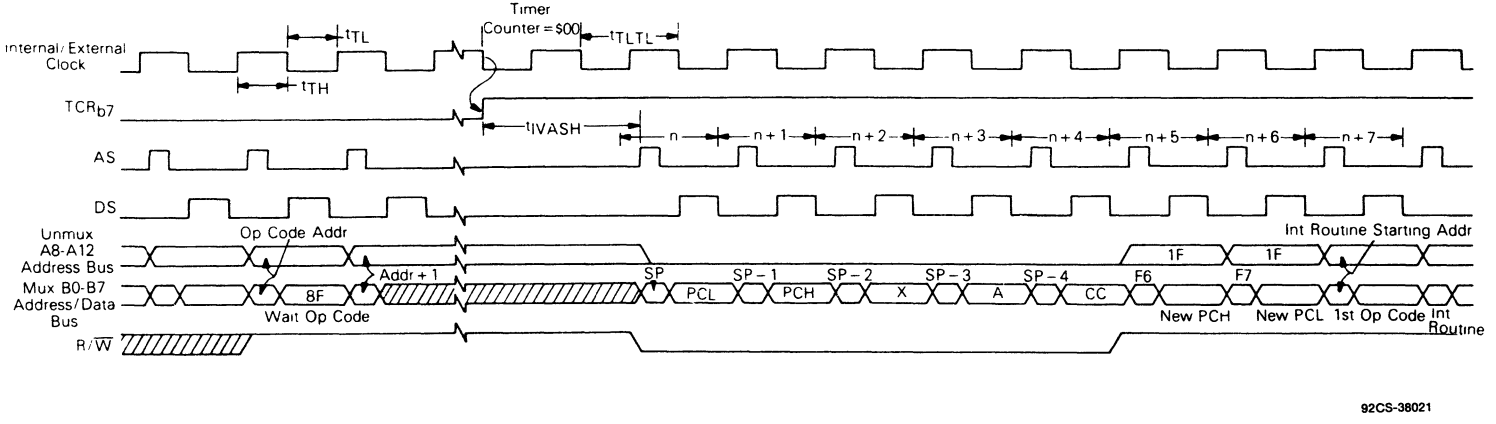
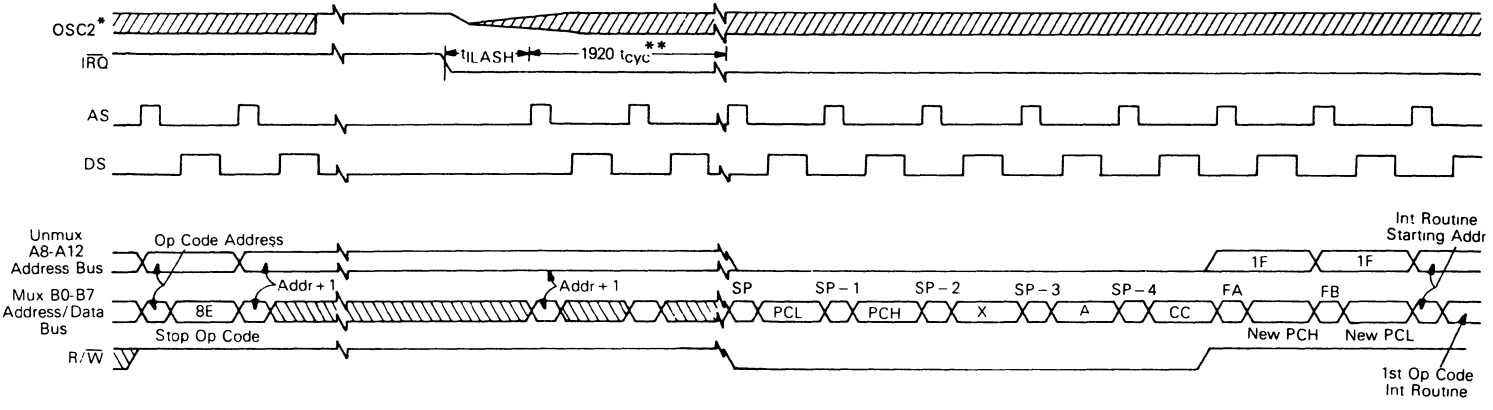


Fig. 7 - Timer interrupt after WAIT instruction timing waveforms.



* Represents the internal gating of the OSC1 input pin.
** t_{cyc} is one instruction cycle (for $f_{OSC} = 5$ MHz, $t_{cyc} = 1\ \mu s$)

Fig. 8 - Interrupt recovery from STOP instruction timing waveforms.

CDP6805E2, CDP6805E2C

FUNCTIONAL PIN DESCRIPTION

V_{DD} and V_{SS} – V_{DD} and V_{SS} provide power to the chip V_{DD} provides power and V_{SS} is ground

$\overline{IRQ}$ (Maskable Interrupt Request) – $\overline{IRQ}$ is a level-sensitive and edge sensitive input which can be used to request an interrupt sequence The MPU completes the current instruction before it responds to the request IF $\overline{IRQ}$ is low and the interrupt mask bit (I-bit) in the Condition Code Register is clear, the MPU begins an interrupt sequence at the end of the current instruction The interrupt circuit recognizes both a "Wire ORed" level as well as pulses on the $\overline{IRQ}$ line (see Interrupt Section for more details) $\overline{IRQ}$ requires an external resistor to V_{DD} for "Wire OR" operation

$\overline{RESET}$ – The $\overline{RESET}$ input is not required for start-up but can be used to reset the MPU's internal state and provide an orderly software start-up procedure Refer to the RESET section for a detailed description

TIMER – The TIMER input is used for clocking the on-chip timer. Refer to TIMER section for a detailed description

AS (Address Strobe) – Address Strobe (AS) is an output strobe used to indicate the presence of an address on the 8-bit multiplexed bus. The AS line is used to demultiplex the eight least significant address bits from the data bus. A latch controlled by Address Strobe should capture addresses on the negative edge. This output is capable of driving one standard TTL load and 130 pF and is available at $f_{OSC} \div 5$ when the MPU is not in the WAIT or STOP states.

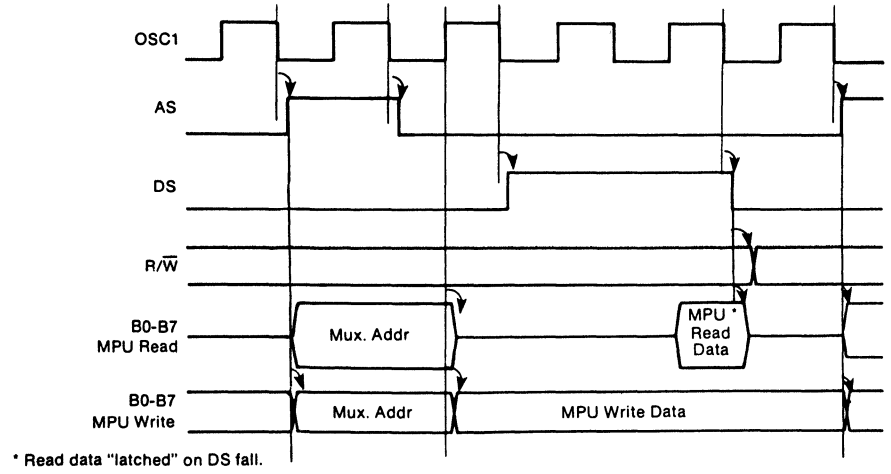
DS (Data Strobe) – This output is used to transfer data to or from a peripheral or memory. DS occurs anytime the MPU does a data read or write. DS also occurs when the MPU does a data transfer to or from the MPU's internal memory Refer to Table 2 and Figure 4 for timing characteristics This output is capable of driving one standard TTL load and 130 pF. DS is a continuous signal at $f_{OSC} \div 5$ when the MPU is not in WAIT or STOP state. Some bus cycles are redundant reads of op code bytes

R/$\overline{W}$ (Read/Write) – The R/$\overline{W}$ output is used to indicate the direction of data transfer for both internal memory and I/O registers, and external peripheral devices and memories. This output is used to indicate to a selected peripheral whether the MPU is going to read or write data on the next Data Strobe (R/$\overline{W}$ low = processor write; R/$\overline{W}$ high = processor read). The R/$\overline{W}$ output is capable of driving one standard TTL load and 130 pF. The normal standby state is Read (high)

A8-A15 (High Order Address Lines) — The A8-A15 output lines constitute the higher order non-multiplexed addresses Each output line is capable of driving one standard TTL load and 130 pF

B0-B7 (Address/Data Bus) – The B0-B7 bidirectional lines constitute the lower order addresses and data. These lines are multiplexed, with address present at Address Strobe time and data present at Data Strobe time When in the data mode, these lines are bidirectional, transferring data to and from memory and peripheral devices as indicated by the R/$\overline{W}$ pin As outputs in either the data or address modes, these lines are capable of driving one standard TTL load and 130 pF

OSC1, OSC2 — The CDP6805E3 provides for two types of oscillator inputs — crystal circuit or external clock. The two oscillator pins are used to interface to a crystal circuit, as shown in Figure 5. If an external clock is used, it must be connected to OSC1 The input at these pins is divided by five to form the cycle rate seen on the AS and DS pins The frequency range is specified by f_{OSC} The OSC1 to bus transitions relationships are provided in Figure 9 for system designs using oscillators slower than 5 MHz



92CS-38023R1

Fig. 9 - OSC1 to bus transitions timing waveforms.

Crystal – The circuit shown in Figure 5 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 electrical characteristics table An external CMOS oscillator is recommended 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.

External Clock – An external clock should be applied to the OSC1 input with the OSC2 input not connected, as shown in Figure 10

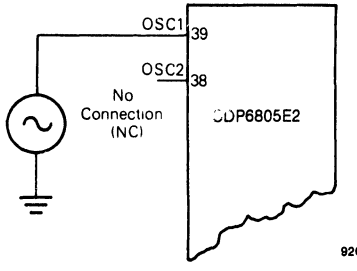
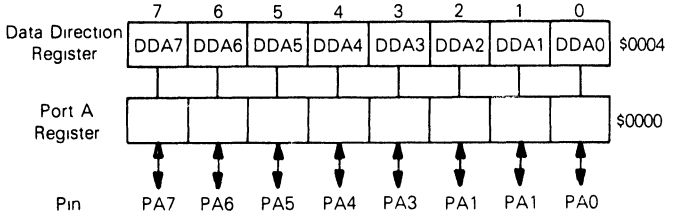


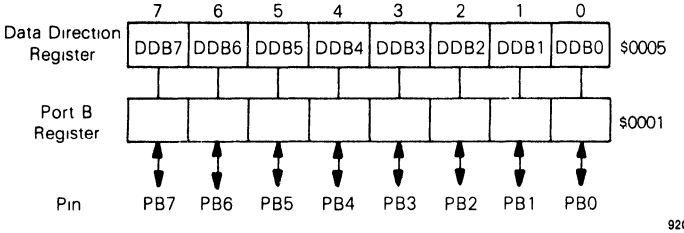
Fig. 10 - External clock connection.

LI (Load Instruction) – This output is used to indicate that a fetch of the next opcode is in progress. LI remains low during an External or Timer interrupt The LI output is only used for certain debugging and test systems For normal operations this pin is not connected. The LI output is capable of driving one standard TTL load and 50 pF This signal overlaps Data Strobe

PA0-PA7 – These eight pins constitute Input/Output Port A. Each line is individually programmed to be either an input or output under software control via its Data Direction Register as shown below. An I/O pin is programmed as an output when the corresponding DDR bit is set to a "1," and as an input when it is set to a "0" In the output mode the bits are latched and appear on the corresponding output pins. An MPU read of the port bits programmed as outputs reflect the last value written to that location. When programmed as an input, the input data bit(s) are not latched An MPU read of the port bits programmed as inputs reflects the current status of the corresponding input pins The Read/Write port timing is shown in Figure 3 See typical I/O Port Circuitry in Figure 11. During a Power-On Reset or external $\overline{\text{RESET}}$ all lines are configured as inputs (zero in Data Direction Register) The output port register is not initialized by reset The TTL compatible three-state output buffers are capable of driving one standard TTL load and 50 pF The DDR is a read/write register

PB0-PB7 – These eight pins interface to Input/Output Port B Refer to PA0-PA7 description for details of operation





CDP6805E2, CDP6805E2C

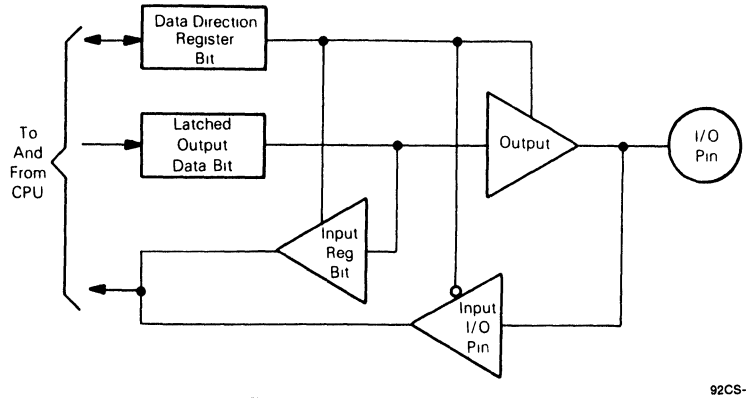


Fig. 11 - Typical I/O port circuitry.

TABLE 3 — I/O PIN FUNCTIONS

R/$\overline{W}$	DDR	I/O Pin Functions
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.

MEMORY ADDRESSING

The CDP6805E2 is capable of addressing 8192 bytes of memory and I/O registers. The address space is divided into internal memory space and external memory space, as shown in Figure 12.

The internal memory space is located within the first 128 bytes of memory (first half of page zero) and is comprised of the I/O port locations, timer locations, and 112 bytes of RAM. The MPU can read from or write to any of these locations. A program write to on-chip locations is repeated on the external bus to permit off-chip memory to duplicate the content of on-chip memory. Program reads to on-chip locations also appear on the external bus, but the MPU accepts data only from the addressed on-chip location. Any read data appearing on the input bus is ignored.

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 onto the stack. When data is removed from the stack, the stack pointer is incremented. A maximum of 64 bytes of RAM is available for stack usage. Since most programs use only a small part of the allotted stack locations for interrupts and/or subroutine stacking purposes, the unused bytes are usable for program data storage.

All memory locations above location $007F are part of the external memory map. In addition, ten locations in the I/O portion of the lower 128 bytes of memory space, as shown in Figure 12, are part of the external memory map. All of the external memory space is user definable except the highest 10 locations. Locations $1FF6 to $1FFF of the external address space are reserved for interrupt and reset vectors (see Figure 12).

REGISTERS

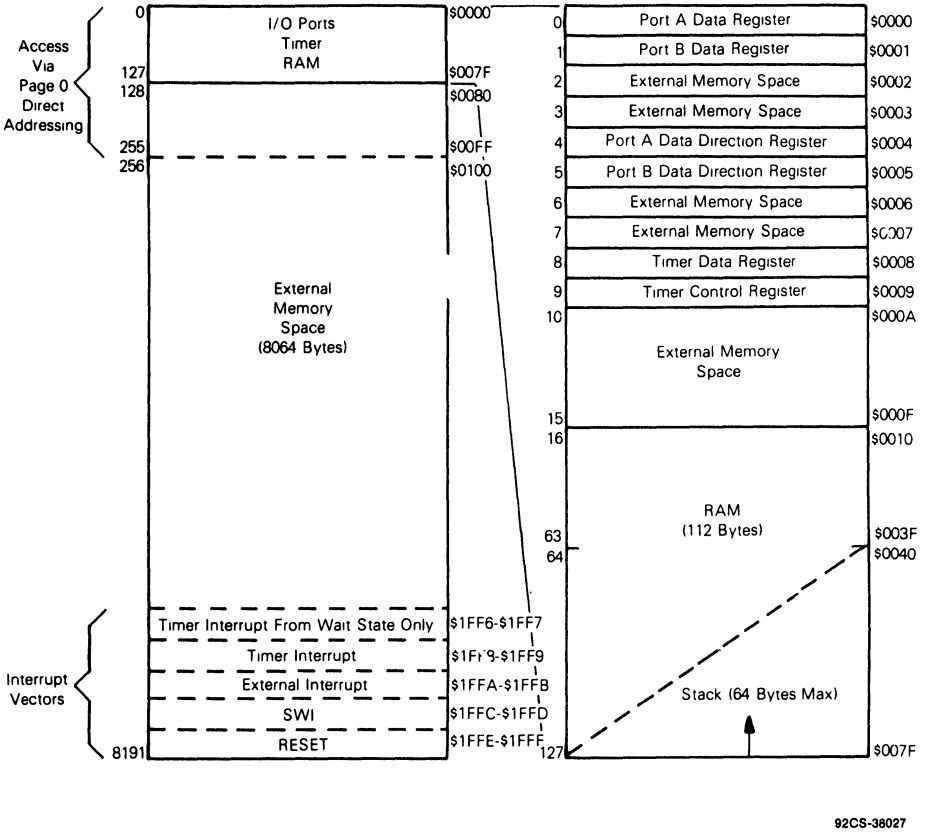
The CDP6805E2 contains five registers as shown in the programming model in Figure 13. The interrupt stacking order is shown in Figure 14.

ACCUMULATOR (A) — This Accumulator is an 8-bit general purpose register used for 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 an 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 a 13-bit register that contains the address of the next instruction to be executed by the processor.

CDP6805E2, CDP6805E2C



92CS-38027

Fig. 12 - Address map.

CDP6805E2, CDP6805E2C

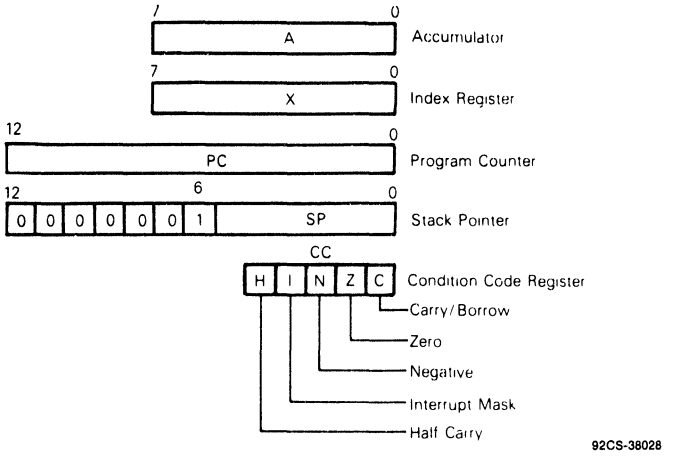


Fig. 13 - Programming model.

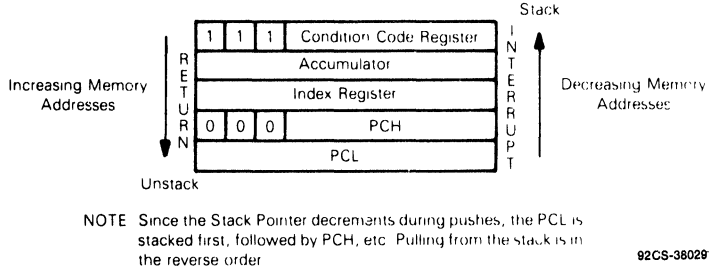


Fig. 14 - Stacking order.

STACK POINTER (SP) – The stack pointer is a 13-bit register containing the address of the next free location on the stack When accessing memory, the seven most-significant bits are permanently set to 0000001. They are appended to the six least-significant register bits to produce an address within the range of $007F to $0040. 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 ($007F). Nested interrupts and/or subroutines may use up to 64 (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

CONDITION CODE REGISTER (CC) – The condition code register is a 5-bit register in which each bit is used to indicate 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 of the five bits is explained below

Half Carry Bit (H) – The H-bit is set to a one 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 addition 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 will be processed when the I-bit is next cleared

Negative Bit (N) – When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was negative (bit 7 in the result is a logical one)

Zero Bit (Z) – When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was zero

Carry Bit (C) – The C-bit is set when a carry or a borrow out of the ALU occurs during an arithmetic instruction. The C-bit is also modified during bit test, shift, rotate, and branch types of instruction

RESETS

The CDP6805E2 has two reset modes: an active low external reset pin ($\overline{\text{RESET}}$) and a Power-On Reset function, refer to Figure 5

$\overline{\text{RESET}}$ (Pin #1) – The $\overline{\text{RESET}}$ input pin is used to reset the MPU and 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_{CYC}. The $\overline{\text{RESET}}$ pin is provided with a Schmitt Trigger to improve its noise immunity capability.

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 reset pin is low at the end of the 1920 t_{CYC} 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 (bit 7) is cleared to a "0".
- Timer control register interrupt mask bit (bit 6) is set to a "1".
- All data direction register bits are cleared to a "0" (inputs).
- Stack pointer is set to $007F.
- The address bus is forced to the reset vector ($1FFE, $1FFF).
- 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

The CDP6805E2 is capable of operation with three different interrupts, two hardware (timer interrupt and external interrupt) and one software (SWI). When any of these interrupts occur, normal processing is suspended at the end of the current instruction execution. All of the program registers (the machine state) are pushed onto the stack, refer to Figure 14 for stacking order The appropriate vector pointing to the starting address of the interrupt service routine is then fetched, refer to Figure 15 for the interrupt sequence.

The priority of the various interrupts from highest to lowest is as follows:

RESET→*→External Interrupt→Timer Interrupt

TIMER INTERRUPT – If the timer mask bit (TCR6) is cleared, then each time the timer decrements to zero (transitions from $01 to $00) an interrupt request is generated The actual processor interrupt is generated only if the interrupt mask bit of the condition code register is also cleared When the interrupt is recognized, the current state of the machine is pushed onto the stack and the I-bit in the condition code register is set This masks 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 $1FF8 and $1FF9. The contents of $1FF6 and $1FF7 specify the service routine. Also, 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.

EXTERNAL INTERRUPT – If the interrupt mask bit of the condition code register is cleared and the external interrupt pin $\overline{\text{IRQ}}$ is "low," then the external interrupt occurs The action of the external interrupt is identical to the timer interrupt with the exception that the service routine address is specified by the contents of $1FFA and $1FFB The interrupt logic recognizes both a "wire ORed" level and pulses on the external interrupt line Figure 16 shows both a functional diagram and timing for the interrupt line The timing diagram shows two different treatments of the interrupt line ($\overline{\text{IRQ}}$) to the processor The first configuration shows many interrupt lines "wire ORed" to form the interrupts at the processor Thus, if after servicing an interrupt the $\overline{\text{IRQ}}$ remains low, then the next interrupt is recognized The second method is single pulses on the interrupt line spaced far enough apart to be service. 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 (one cycle $t_{CYC}=5/f_{OSC}$) to the total number of cycles it takes to complete the service routine including the RTI instruction; refer to Figure 6.

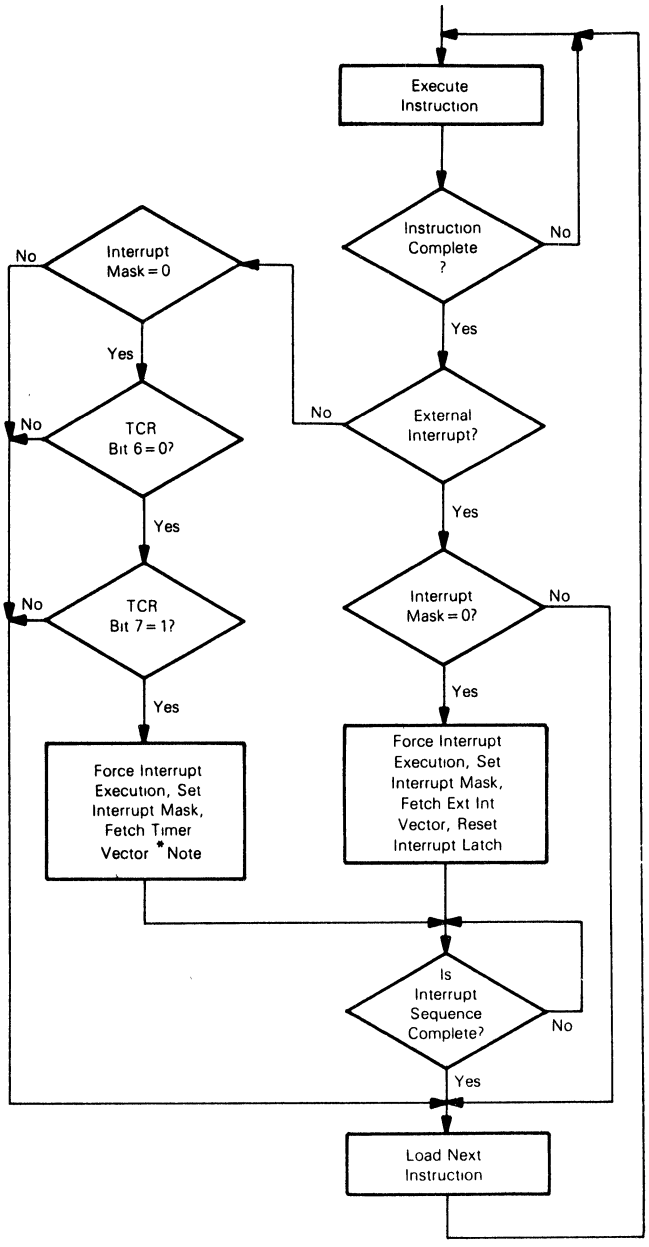
SOFTWARE INTERRUPT (SWI) – The software interrupt is an executable instruction The action of the SWI instruction is similar to the hardware interrupts The SWI is executed regardless of the state of the interrupt mask in the condition code register. The service routine address is specified by the contents of memory locations $1FFC and $1FFD See Figure 15 for Interrupt and Instruction Processing Flowchart

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

$\overline{\text{RESET}}$ – The $\overline{\text{RESET}}$ 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 of memory locations $1FFE and $1FFF The interrupt mask of the condition code register is also set Refer to RESET section for details

*Any current instruction including SWI.

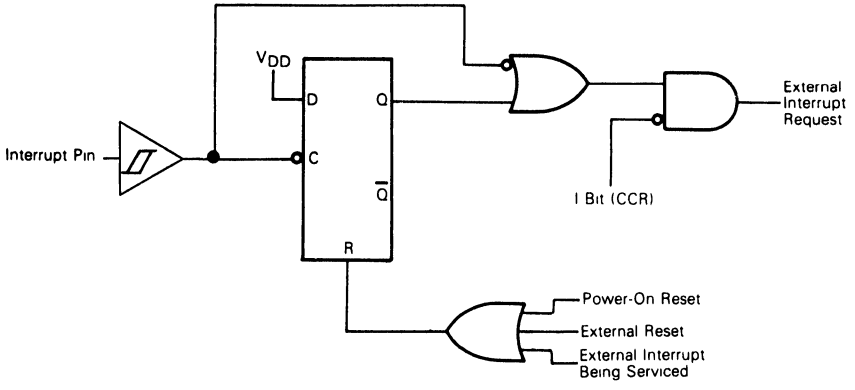
CDP6805E2, CDP6805E2C



*NOTE: The clear of TCR bit 7 must be accomplished with software.

Fig. 15 - Interrupt and instruction processing flowchart.

(a) Interrupt Functional Diagram



(b) Interrupt Mode Diagram

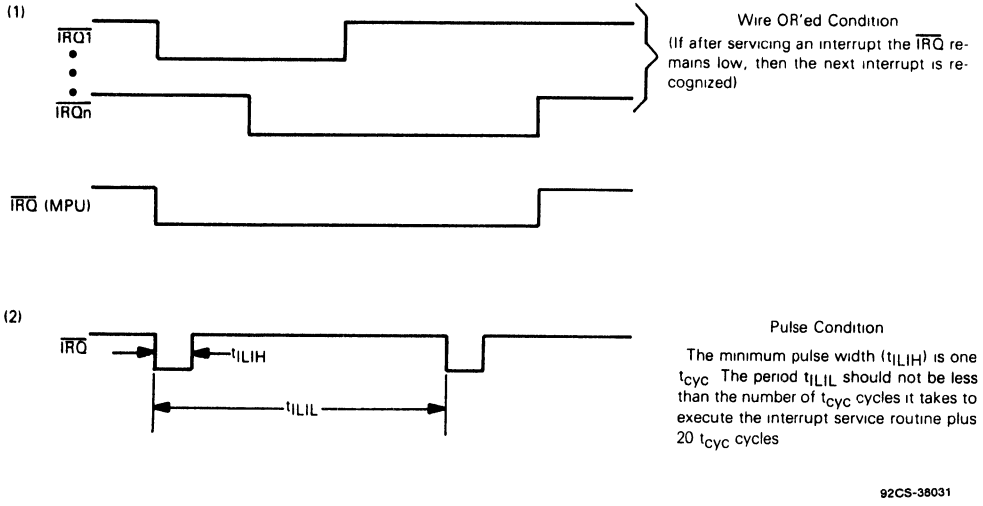


Fig. 16 - External interrupt.

CDP6805E2, CDP6805E2C

STOP – The STOP instruction places the CDP6805E2 in a low 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 17. The DS and AS lines go to a low state and the R/$\overline{W}$ line goes to a high state. The multiplexed address/data bus goes to the data input state. The high order address lines remain at the address of the next instruction. The MPU remains in the STOP mode until an external interrupt or reset occurs; refer to Figure 8 and 17

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 timer interrupts. External interrupts are enabled in the condition code register. All other registers and memory remain unaltered. All I/O lines remain unchanged.

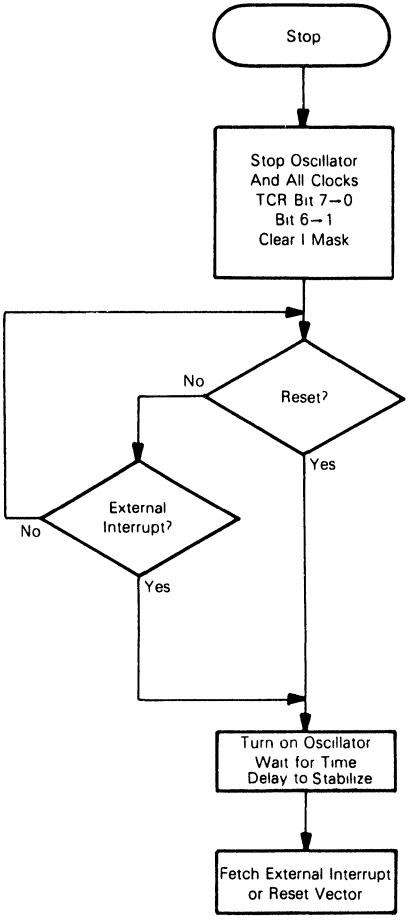


Fig. 17 - Stop function flowchart.

WAIT – The WAIT instruction places the CDP6805E2 in a low power consumption mode, but the WAIT mode consumes somewhat more power than the STOP mode; refer to Table 1 In the WAIT function, the internal clock is disabled from all internal circuitry except the Timer circuit, refer to Figure 18 Thus, all internal processing is halted except the Timer which is allowed to count in a normal sequence The R/$\overline{W}$ line goes to a high state, the multiplexed address/data bus goes to the data input state, and the DS and AS lines go to the low state The high order address lines remain at the address of the next instruction The MPU remains in this state until an external interrupt, timer interrupt, or a reset occurs, refer to Figures 7 and 18

92CS-38032

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 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 MPU timer contains a single 8-bit software programmable counter with 7-bit software selectable prescaler. 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 $1FF8 and $1FF9 in order to begin servicing the interrupt, unless it was in locations $1FF6 and $1FF7 the WAIT mode.

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 becomes stable prior to the read portion of a cycle and does not change during the read. The timer interrupt request bit remains set until cleared by the software If this happens before the timer interrupt is serviced, the interrupt is lost 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 which is used as the counter input The processor cannot write into or read from the prescaler, however, its contents are cleared to all "0's" by the write operation into TCR when bit 3 of the written data equals 1, which 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, TCR5 control bits Refer to the TIMER CONTROL REGISTER section

Timer Input Mode 1 – If TCR4 and TCR5 are both programmed to a "0", the input to the Timer is from an internal clock and the Timer input is disabled. The internal clock mode can be used for periodic interrupt generation, as well

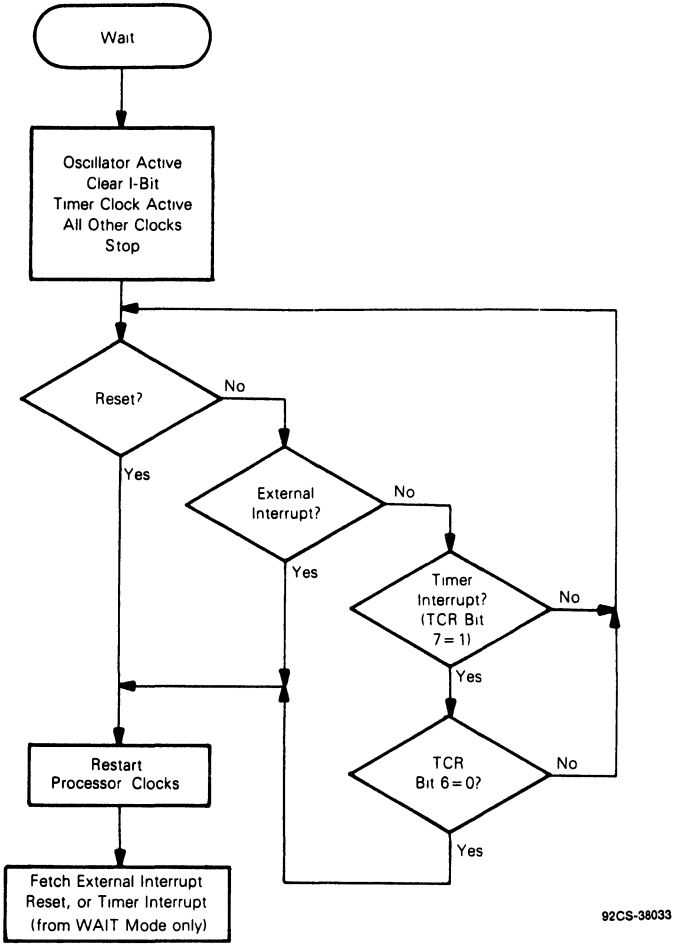


Fig. 18 - Wait function flowchart.

as a reference in frequency and event measurement. The internal clock is the instruction cycle clock and is coincident with Address Strobe (AS) except during a WAIT instruction. During a WAIT instruction the AS pin goes to a low state but the internal clock to the Timer continues to run at its normal rate.

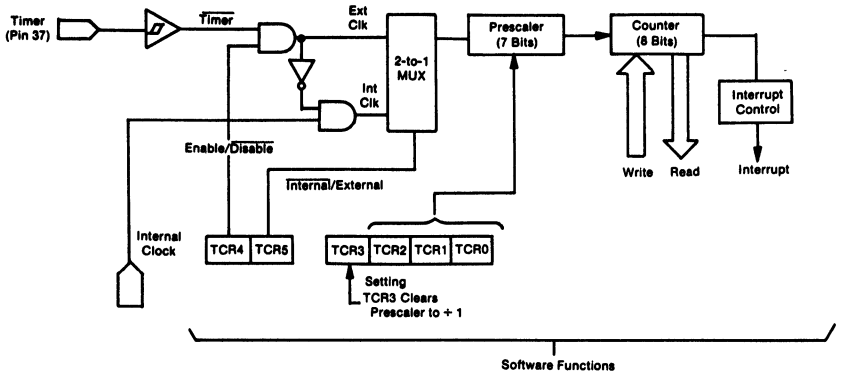
Timer Input Mode 2 – With TCR4 = 1 and TCR5 = 0, the internal clock and the TIMER input pin are ANDed together to form the Timer input signal. This mode can be used to measure external pulse widths. The external pulse simply turns on the internal clock for the duration of the pulse. The resolution of the count in this mode is ±1 clock and therefore accuracy improves with longer input pulse widths.

Timer Input Mode 3 – If TCR4 = 0 and TCR5 = 1, then all inputs to the Timer are disabled.

Timer Input Mode 4 – If TCR4 = 1 and TCR5 = 1, the internal clock input to the Timer is disabled and the TIMER input pin becomes the input to the Timer. The external Timer pin can, in this mode, be used to count external events as well as external frequencies for generating periodic interrupts.

Figure 19 shows a block diagram of the Timer subsystem Power-on Reset and the STOP instruction cause the counter to be set to $FO

CDP6805E2, CDP6805E2C



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. 19 - Timer 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 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 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 (AS).

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 (AS) to Timer
0	1	AND of internal clock (AS) and TIMER pin to Timer
1	0	Inputs to Timer disabled
1	1	TIMER pin to Timer

Refer to Figure 19 for Logic Representation.

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

TCR2, TCR1, TCR0 – Prescaler address bits: decoded to select one of eight taps 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 MPU 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 jump unconditional (JMP) and jump to subroutine (JSR) instructions have no register operand. 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 – This set of instructions branches if a particular condition is met, otherwise no operation is performed. Branch instructions are two byte instructions. Refer to Table 6.

BIT MANIPULATION INSTRUCTIONS – The MPU is capable of setting or clearing any bit which resides in the first 256 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 these 256 locations. The bit set, bit clear and bit test and branch functions are all 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 for instruction cycle timing.

CONTROL INSTRUCTIONS – These instructions are register reference instructions and are used to control processor operation during program execution. Refer to Table 8 for instruction cycle timing.

ALPHABETICAL LISTING – The complete instruction set is given in alphabetical order in Table 9.

OPCODE MAP SUMMARY – Table 10 is an opcode map for the instructions used on the MCU.

ADDRESSING MODES

The MPU uses ten different addressing modes to give the programmer 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 9 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 10.

The term "Effective Address" or EA is used in describing the various addressing modes, which is defined as the 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 register 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 up to 128 bytes of off-chip ROM. Direct addressing is efficient in both memory and speed.

$$EA = (PC + 1);\ PC \leftarrow PC + 2$$
$$\text{Address Bus High} \leftarrow 0;\ \text{Address Bus Low} \leftarrow (PC + 1)$$

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.

$$EA = X,\ PC \leftarrow PC + 1$$
$$\text{Address Bus High} \leftarrow 0;\ \text{Address Bus Low} \leftarrow X$$

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

CDP6805E2, CDP6805E2C

TABLE 9 - 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			Λ	•	Λ	Λ	Λ
ADD		X	X	X		X	X	X			Λ	•	Λ	Λ	Λ
AND		X	X	X		X	X	X			•	•	Λ	Λ	•
ASL	X		X			X	X				•	•	Λ	Λ	Λ
ASR	X		X			X	X				•	•	Λ	Λ	Λ
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			•	•	Λ	Λ	•
BLO					X						•	•	•	•	•
BLS					X						•	•	•	•	•
BMC					X						•	•	•	•	•
BMI					X						•	•	•	•	•
BMS					X						•	•	•	•	•
BNE					X						•	•	•	•	•
BPL					X						•	•	•	•	•
BRA					X						•	•	•	•	•
BRN					X						•	•	•	•	•
BRCLR										X	•	•	•	•	Λ
BRSET										X	•	•	•	•	Λ
BSET									X		•	•	•	•	•
BSR					X						•	•	•	•	•
CLC	X										•	•	•	•	0
CLI	X										•	0	•	•	•
CLR	X		X			X	X				•	•	0	1	•
CMP		X	X	X		X	X	X			•	•	Λ	Λ	Λ
COM	X		X			X	X				•	•	Λ	Λ	1
CPX		X	X	X		X	X	X			•	•	Λ	Λ	Λ
DEC	X		X			X	X				•	•	Λ	Λ	•
EOR		X	X	X		X	X	X			•	•	Λ	Λ	•
INC	X		X			X	X				•	•	Λ	Λ	•
JMP			X	X		X	X	X			•	•	•	•	•
JSR			X	X		X	X	X			•	•	•	•	•
LDA		X	X	X		X	X	X			•	•	Λ	Λ	•
LDX		X	X	X		X	X	X			•	•	Λ	Λ	•
LSL	X		X			X	X				•	•	Λ	Λ	Λ
LSR	X		X			X	X				•	•	0	Λ	Λ
NEG	X		X			X	X				•	•	Λ	Λ	Λ
NOP	X										•	•	•	•	•
ORA		X	X	X		X	X	X			•	•	Λ	Λ	•
ROL	X		X			X	X				•	•	Λ	Λ	Λ
ROR	X		X			X	X				•	•	Λ	Λ	Λ
RSP	X										•	•	•	•	•
RTI	X										?	?	?	?	?
RTS	X										•	•	•	•	•
SBC		X	X	X		X	X	X			•	•	Λ	Λ	Λ
SEC	X										•	•	•	•	1
SEI	X										•	1	•	•	•
STA			X	X		X	X	X			•	•	Λ	Λ	•
STOP	X										•	0	•	•	•
STX			X	X		X	X	X			•	•	Λ	Λ	•
SUB		X	X	X		X	X	X			•	•	Λ	Λ	Λ
SWI	X										•	1	•	•	•
TAX	X										•	•	•	•	•
TST	X		X			X	X				•	•	Λ	Λ	•
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
- Λ Test and Set if True Cleared Otherwise
- • Not Affected
- ? Load CC Register From Stack
- 0 Cleared
- 1 Set

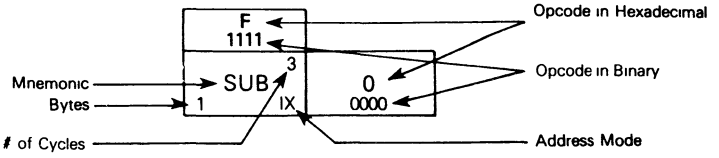
TABLE 10 — CDP6805E2 INSTRUCTION SET OPCODE MAP

	Bit Manipulation		Branch	Read/Modify/Write					Control		Register/Memory						
	BTB	BSC	REL	DIR	INH(A)	INH(X)	IX1	IX	INH	INH	IMM	DIR	EXT	IX2	IX1	IX	
Hi / Low	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111	Hi / Low
0 0000	BRSET0 5 3 BTB	BSET0 5 2 BSC	BRA 3 2 REL	NEG 5 2 DIR	NEGA 3 1 INH	NEGX 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
*	CMOS Versions Only

LEGEND



CDP6805E2, CDP6805E2C

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 The operand is therefore located anywhere within the lowest 511 memory locations For example, this mode of addressing is useful for selecting the m-th element in an n element table All instructions are two bytes The contents of the index register (X) is not changed The contents 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 256 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 to address 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 clear 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

SYSTEM CONFIGURATION

Figures 20 through 25 show in general terms how the CDP6805E2 bus structure may be utilized. Specified interface details vary with the various peripheral and memory devices employed.

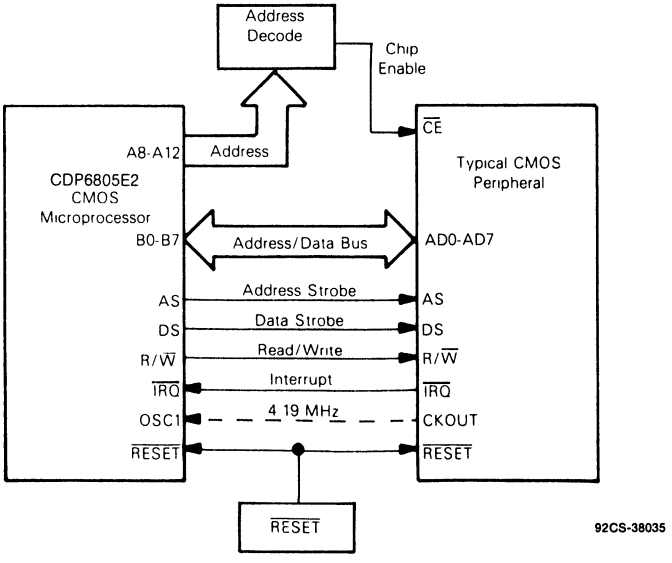


Fig. 20 - Connection to CMOS peripherals.

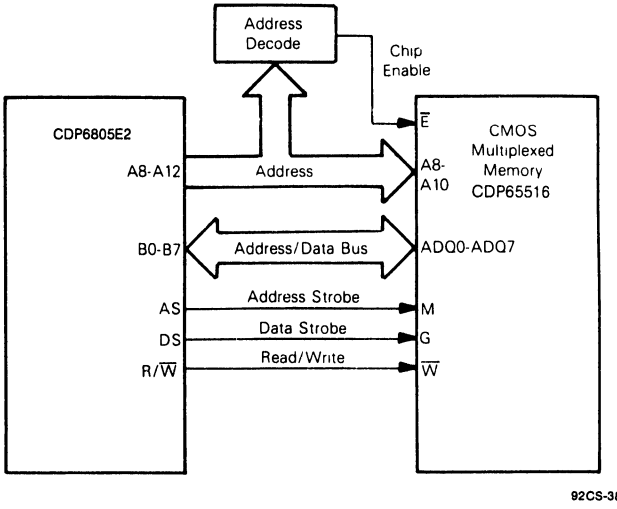
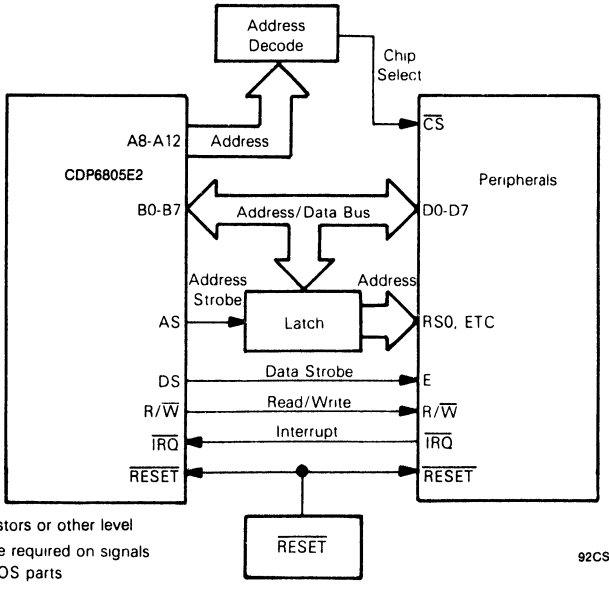


Fig. 21 - Connection to CMOS multiplexed memories.



NOTE In some cases, pullup resistors or other level shifting techniques may be required on signals going from NMOS to CMOS parts

Fig. 22 - Connection to peripherals.

CDP6805E2, CDP6805E2C

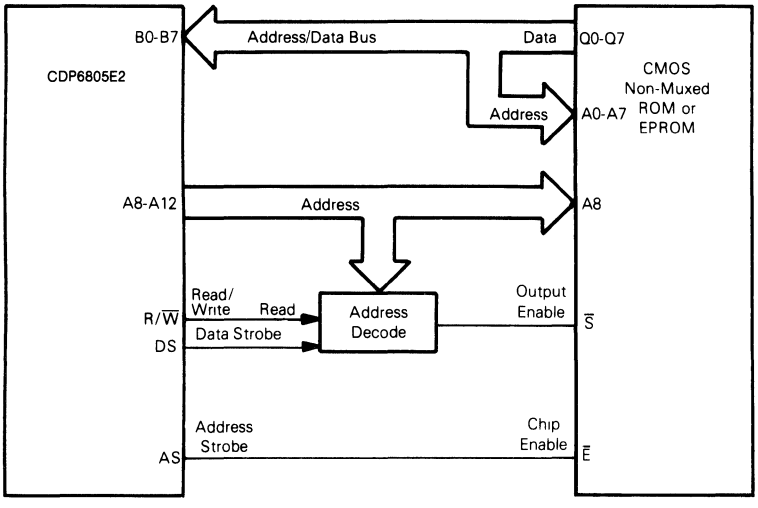


Fig. 23 - Connection to latch non-multiplexed CMOS ROM or EPROM.

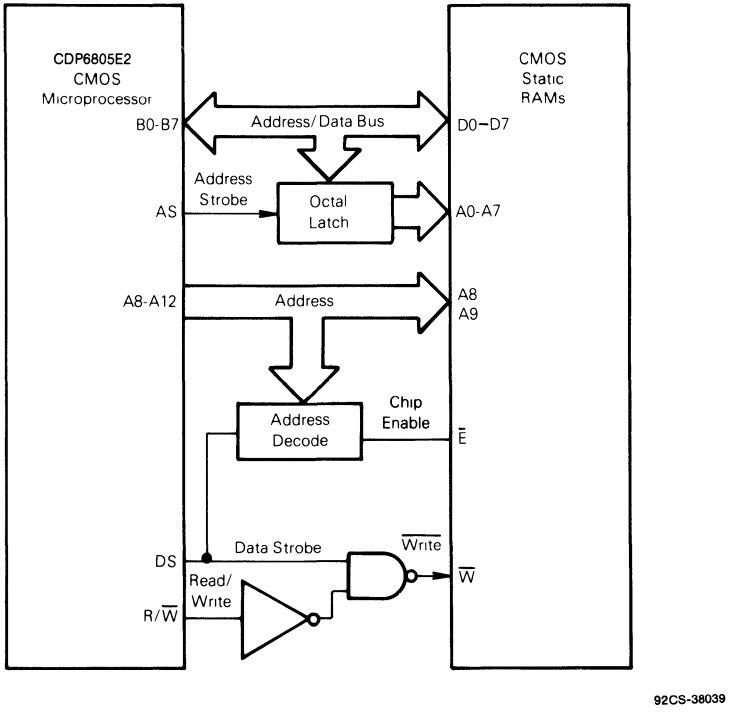


Fig. 24 - Connection to static CMOS RAMs.

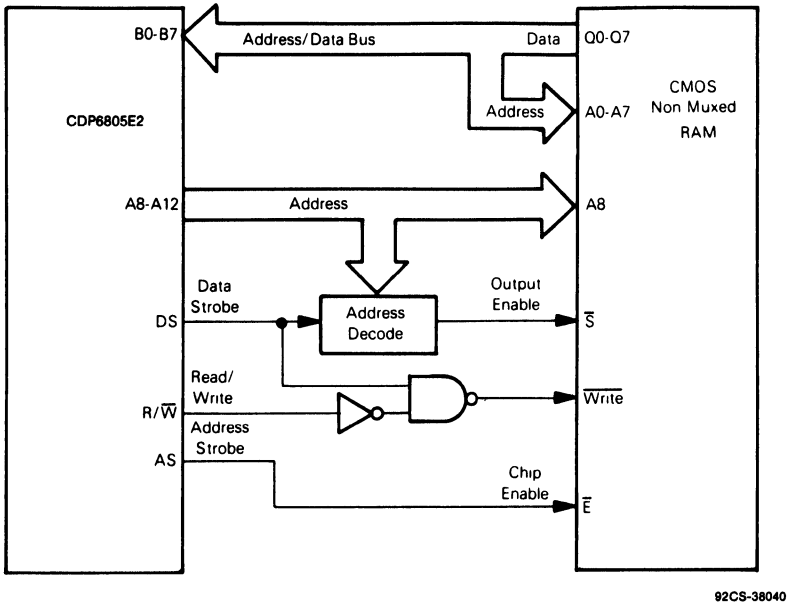


Fig. 25 - Connection to latched non-multiplexed CMOS RAM.

CDP6805E2, CDP6805E2C

Table 11 provides a detailed description of the information present on the Bus, the Read/Write (R/$\overline{W}$) pin and the Load Instruction (LI) pin during each cycle for each instruction.

This information is useful in comparing actual with expected results during debug of both software and hardware as the control program is executed. The information is categorized in groups according to addressing mode and number of cycles per instruction.

TABLE 11 — SUMMARY OF CYCLE BY CYCLE OPERATION

Address Mode / Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Pin	LI Pin	Data Bus
Inherent						
LSR LSL ASR NEG CLR ROL COM ROR DEC INC TST	3	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Op Code Next Instruction
		3	Op Code Address + 1	1	0	Op Code Next Instruction
TAX CLC SEC STOP CLI SEI RSP WAIT NOP TXA	2	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Op Code Next Instruction
RTS	6	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Op Code Next Instruction
		3	Stack Pointer	1	0	Irrelevant Data
		4	Stack Pointer + 1	1	0	Irrelevant Data
		5	Stack Pointer + 2	1	0	Irrelevant Data
		6	New Op Code Address	1	0	New Op Code
SWI	10	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Op Code Next Instruction
		3	Stack Pointer	0	0	Return Address (LO Byte)
		4	Stack Pointer − 1	0	0	Return Address (HI Byte)
		5	Stack Pointer − 2	0	0	Contents of Index Register
		6	Stack Pointer − 3	0	0	Contents of Accumulator
		7	Stack Pointer − 4	0	0	Contents of CC Register
		8	Vector Address 1FFC (Hex)	1	0	Address of Int Routine (HI Byte)
		9	Vector Address 1FFD (Hex)	1	0	Address of Int. Routine (LO Byte)
		10	Interrupt Routine Starting Address	1	0	Interrupt Routine First Opcode
RTI	9	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Op Code Next Instruction
		3	Stack Pointer	1	0	Irrelevant Data
		4	Stack Pointer + 1	1	0	Irrelevant Data
		5	Stack Pointer + 2	1	0	Irrelevant Data
		6	Stack Pointer + 3	1	0	Irrelevant Data
		7	Stack Pointer + 4	1	0	Irrelevant Data
		8	Stack Pointer + 5	1	0	Irrelevant Data
		9	New Op Code Address	1	0	New Op Code
Immediate						
ADC EOR CPX ADD LDA LDX AND ORA BIT SBC CMB SUB	2	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Operand Data
Bit Set/Clear						
BSET n BCLR n	5	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Address of Operand
		3	Address of Operand	1	0	Operand Data
		4	Address of Operand	1	0	Operand Data
		5	Address of Operand	0	0	Manipulated Data
Bit Test and Branch						
BRSET n BRCLR n	5	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Address of Operand
		3	Address of Operand	1	0	Operand Data
		4	Op Code Address + 2	1	0	Branch Offset
		5	Op Code Address + 2	1	0	Branch Offset
Relative						
BCC BHI BNE BEQ BCS BPL BHCC BLS BIL BMC BRN BHCS BIH BMI BMS BRA	3	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Branch Offset
		3	Op Code Address + 1	1	0	Branch Offset
BSR	6	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Branch Offset
		3	Op Code Address + 1	1	0	Branch Offset
		4	Subroutine Starting Address	1	0	First Subroutine Op Code
		5	Stack Pointer	0	0	Return Address (LO Byte)
		6	Stack Pointer − 1	0	0	Return Address (HI Byte)

TABLE 11 – SUMMARY OF CYCLE BY CYCLE OPERATION (CONTINUED)

Address Mode / Instructions	Cycles	Cycles #	Address Bus	R/$\overline{W}$ Pin	LI Pin	Data Bus
Direct						
JMP	2	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Jump Address
ADC EOR CPX ADD LDA LDX AND ORA BIT SBC CMP SUB	3	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Address of Operand
		3	Address of Operand	1	0	Operand Data
TST	4	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Address of Operand
		3	Address of Operand	1	0	Operand Data
		4	Op Code Address + 2	1	0	Op Code Next Instruction
STA STX	4	1	Op Code Address	1	1	Op Code
		2	Op Code Adrress + 1	1	0	Address of Operand
		3	Op Code Address + 1	1	0	Address of Operand
		4	Address of Operand	0	0	Operand Data
LSL LSR DEC ASR NEG INC CLR ROL COM ROR	5	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Address of Operand
		3	Operand Address	1	0	Current Operand Data
		4	Operand Address	1	0	Current Operand Data
		5	Operand Address	0	0	New Operand Data
JSR	5	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Subroutine Address (LO Byte)
		3	Subroutine Starting Address	1	0	1st Subroutine Op Code
		4	Stack Pointer	0	0	Return Address (LO Byte)
		5	Stack Pointer – 1	0	0	Return Address (HI Byte)
Extended						
JMP	3	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Jump Address (HI Byte)
		3	Op Code Address + 2	1	0	Jump Address (LO Byte)
ADC BIT ORA ADD CMP LDX AND EOR SBC CPX LDA SUB	4	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Address Operand (HI Byte)
		3	Op Code Address + 2	1	0	Address Operand (LO Byte)
		4	Address of Operand	1	0	Operand Data
STA STX	5	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Address of Operand (HI Byte)
		3	Op Code Address + 2	1	0	Address of Operand (LO Byte)
		4	Op Code Address + 2	1	0	Address of Operand (LO Byte)
		5	Address of Operand	0	0	Operand Data
JSR	6	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Address of Subroutine (HI Byte)
		3	Op Code Address + 2	1	0	Address of Subroutine (LO Byte)
		4	Subroutine Starting Address	1	0	1st Subroutine Op Code
		5	Stack Pointer	0	0	Return Address (LO Byte)
		6	Stack Pointer – 1	0	0	Return Address (HI Byte)
Indexed, No-Offset						
JMP	2	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Op Code Next Instruction
ADC EOR CPX ADD LDA LDX AND ORA BIT SBC CMP SUB	3	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Op Code Next Instruction
		3	Index Register	1	0	Operand Data
TST	4	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Op Code Next Instruction
		3	Index Register	1	0	Operand Data
		4	Op Code Address + 1	1	0	Op Code Next Instruction
STA STX	4	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Op Code Next Instruction
		3	Op Code Address + 1	1	0	Op Code Next Instruction
		4	Index Register	0	0	Operand Data
LSL LSR DEC ASR NEG INC CLR ROL COM ROR	5	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Op Code Next Instruction
		3	Index Register	1	0	Current Operand Data
		4	Index Register	1	0	Current Operand Data
		5	Index Register	0	0	New Operand Data
JSR	5	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Op Code Next Instruction
		3	Index Register	1	0	1st Subroutine Op Code
		4	Stack Pointer	0	0	Return Address (LO Byte)
		5	Stack Pointer – 1	0	0	Return Address (HI Byte)

CDP6805E2, CDP6805E2C

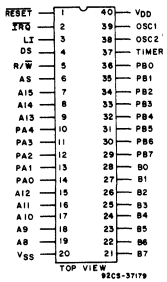
TABLE 11 — SUMMARY OF CYCLE BY CYCLE OPERATION (CONTINUED)

Address Mode / Instructions	Cycles	Cycles #	Address Bus	R/$\overline{W}$ Pin	LI Pin	Data Bus
Indexed 8-Bit Offset						
JMP	3	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Offset
		3	Op Code Address + 1	1	0	Offset
ADC EOR CPX ADD LDA LDX AND ORA CMP SUB BIT SBC	4	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Offset
		3	Op Code Address + 1	1	0	Offset
		4	Index Register + Offset	1	0	Operand Data
STA STX	5	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Offset
		3	Op Code Address + 1	1	0	Offset
		4	Op Code Address + 1	1	0	Offset
		5	Index Register + Offset	0	0	Operand Data
TST	5	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Offset
		3	Op Code Address + 1	1	0	Offset
		4	Index Register + Offset	1	0	Operand Data
		5	Op Code Address + 2	1	0	Op Code Next Instruction
LSL LSR ASR NEG CLR ROL COM ROR DEC INC	6	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Offset
		3	Op Code Address + 1	1	0	Offset
		4	Index Register + Offset	1	0	Current Operand Data
		5	Index Register + Offset	1	0	Current Operand Data
		6	Index Register + Offset	0	0	New Operand Data
JSR	6	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Offset
		3	Op Code Address + 1	1	0	Offset
		4	Index Register + Offset	1	0	1st Subroutine Op Code
		5	Stack Pointer	0	0	Return Address LO Byte
		6	Stack Pointer − 1	0	0	Return Address HI Byte
Indexed, 16-Bit Offset						
JMP	4	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Offset (HI Byte)
		3	Op Code Address + 2	1	0	Offset (LO Byte)
		4	Op Code Address + 2	1	0	Offset (LO Byte)
ADC CMP SUB ADD EOR SBC AND ORA CPX LDA BIT LDX	5	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Offset (HI Byte)
		3	Op Code Address + 2	1	0	Offset (LO Byte)
		4	Op Code Address + 2	1	0	Offset (LO Byte)
		5	Index Register + Offset	1	0	Operand Data
STA STX	6	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Offset (HI Byte)
		3	Op Code Address + 2	1	0	Offset (LO Byte)
		4	Op Code Address + 2	1	0	Offset (LO Byte)
		5	Op Code Address + 2	1	0	Offset (LO Byte)
		6	Index Register + Offset	0	0	Operand Data
JSR	7	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Offset (HI Byte)
		3	Op Code Address + 2	1	0	Offset (LO Byte)
		4	Op Code Address + 2	1	0	Offset (LO Byte)
		5	Index Register + Offset	1	0	1st Subroutine Op Code
		6	Stack Pointer	0	0	Return Address (LO Byte)
		7	Stack Pointer − 1	0	0	Return Address (HO Byte)

TABLE 11 — SUMMARY OF CYCLE BY CYCLE OPERATION (CONTINUED)

Instructions	Cycles	Cycles #	Address Bus	$\overline{RESET}$ Pin	R/$\overline{W}$ Pin	LI Pin	Data Bus
Other Functions							
Hardware $\overline{RESET}$	5		$1FFE	0	1	0	Irrelevant Data
			$1FFE	0	1	0	Irrelevant Data
		1	$1FFE	1	1	0	Irrelevant Data
		2	$1FFE	1	1	0	Irrelevant Data
		3	$1FFE	1	1	0	Vector High
		4	$1FFF	1	1	0	Vector Low
		5	Reset Vector	1	1	0	Op Code
Power on Reset	1922	1	$1FFE	1	1	0	Irrelevant Data
		•	•	•	•	•	•
		•	•	•	•	•	•
		•	•	•	•	•	•
		1919	$1FFE	1	1	0	Irrelevant Data
		1920	$1FFE	1	1	0	Vector High
		1921	$1FFF	1	1	0	Vector Low
		1922	Reset Vector	1	1	0	Op Code
Instruction	**Cycles**	**Cycles #**	**Address Bus**	$\overline{IRQ}$ Pin	R/$\overline{W}$ Pin	LI Pin	**Data Bus**
$\overline{IRQ}$ Interrupt (Timer Vector $1FF8, $1FF9)	10		Last Cycle of Previous Instruction	0	X	0	X
		1	Next Op Code Address	0	1	0	Irrelevant Data
		2	Next Op Code Address	X	1	0	Irrelevant Data
		3	SP	X	0	0	Return Address (LO Byte)
		4	SP – 1	X	0	0	Return Address (HI Byte)
		5	SP – 2	X	0	0	Contents Index Reg
		6	SP – 3	X	0	0	Contents Accumulator
		7	SP – 4	X	0	0	Contents CC Register
		8	$1FFA	X	1	0	Vector High
		9	$1FFB	X	1	0	Vector Low
		10	$\overline{IRQ}$ Vector	X	1	0	Int Routine First

CDP6805E3, CDP6805E3C



TERMINAL ASSIGNMENT

CMOS 8-Bit Microprocessor

Hardware Features:

- *64K address space version of CMOS 6805E2*
- *Typical full speed operating power of 35 mW @ 5 V*
- *Typical WAIT mode power of 5 mW*
- *Typical STOP mode power of 25 µW*
- *112 bytes of on-chip RAM*
- *13 bidirectional I/O lines*
- *Internal 8-bit timer with software programmable 7-bit prescaler*
- *External timer input*
- *Full external and timer interrupts*
- *Multiplexed address/data bus*
- *Master reset and power-on reset*
- *Capable of addressing up to 64K bytes of external memory*
- *Single 3- to 6-volt supply*
- *On-chip oscillator*

Software Features:

- *Similar to the CDP6805E2, F2, and G2*
- *Efficient use of program space*
- *Versatile interrupt handling*
- *Memory mapped I/O*
- *True bit manipulation*
- *Addressing modes with indexed addressing for tables*
- *Efficient instruction set*
- *Two power savings standby modes*

The CDP6805E3 Microprocessor Unit (MPU) belongs to the CDP6805 Family of CMOS Microcomputers. This 8-bit fully static and expandable microprocessor contains a CPU, on-chip RAM, I/O, and Timer. It is a low-power, low-cost processor designed for mid-range applications in the consumer, automotive, industrial, and communications markets where very low power consumption constitutes an important factor. The major features of the CDP6805E3 are listed under "Hardware Features" and "Software Features".

The CDP6805E3 is supplied in a 40-lead hermetic dual-in-line ceramic package (D suffix), a 40-lead dual-in-line plastic package (E suffix), and a 44-lead plastic chip-carrier package (Q suffix).

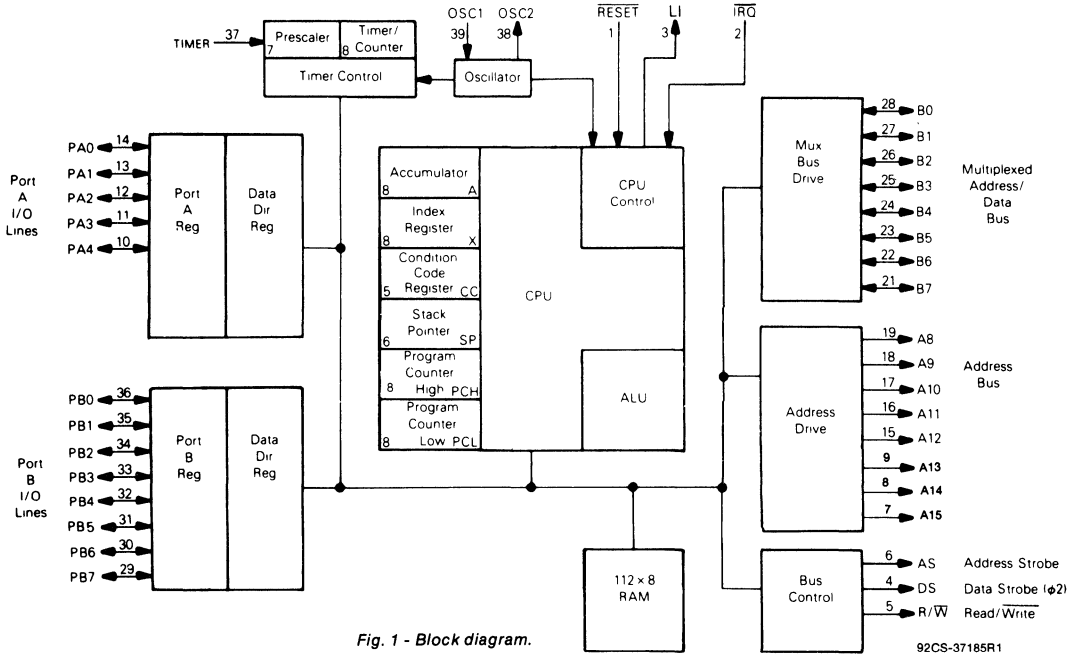


Fig. 1 - Block diagram.

File Number **1503**

CDP6805E3, CDP6805E3C

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 CDP6805E3 CDP6805E3C	T_A	T_L to T_H 0 to 70 −40 to +85	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C

DC ELECTRICAL CHARACTERISTICS 3 V (V_{DD}=3 Vdc, V_{SS}=0, T_A=T_L to T_H, unless otherwise noted)

Characteristics	Symbol	Min	Max	Unit
Output Voltage $I_{LOAD} \leq 10\ \mu A$	V_{OL} V_{OH}	– V_{DD} − 0.1	0.1 –	V
Total Supply Current (C_L = 50 pF – no DC loads) t_{cyc} = 5 µs				
Run (V_{IL} = 0.2 V, V_{IH} = V_{DD} − 0.2 V)	I_{DD}	–	1.3	mA
Wait (Test Conditions – See Note Below)	I_{DD}	–	200	µA
Stop (Test Conditions – See Note Below)	I_{DD}	–	100	µA
Output High Voltage				
(I_{LOAD} = 0.25 mA) A8-A15, B0-B7	V_{OH}	2.7	–	V
(I_{LOAD} = 0.1 mA) PA0-PA4, PB0-PB7	V_{OH}	2.7	–	V
(I_{LOAD} = 0.25 mA) DS, AS, R/$\overline{W}$	V_{OH}	2.7	–	V
Output Low Voltage				
(I_{LOAD} = 0.25 mA) A8-A15, B0-B7	V_{OL}	–	0.3	V
(I_{LOAD} = 0.25 mA) PA0-PA4, PB0-PB7	V_{OL}	–	0.3	V
(I_{LOAD} = 0.25 mA) DS, AS, R/$\overline{W}$	V_{OL}	–	0.3	V
Input High Voltage				
PA0-PA4, PB0-PB7, B0-B7	V_{IH}	2.1	–	V
TIMER, $\overline{IRQ}$, $\overline{RESET}$	V_{IH}	2.5	–	V
OSC1	V_{IH}	2.1	–	V
Input Low Voltage (All inputs)	V_{IL}	–	0.5	V
Frequency of Operation				
Crystal	f_{OSC}	0.032	1	MHz
External Clock	f_{OSC}	DC	1	MHz
Input Current $\overline{RESET}$, $\overline{IRQ}$, Timer, OSC1	I_{in}	–	±1	µA
Three-State Output Leakage PA0-PA4, PB0-PB7, B0-B7	I_{TSL}	–	±10	µA
Capacitance $\overline{RESET}$, $\overline{IRQ}$, Timer	C_{in}	–	8	pF
Capacitance DS, AS, R/$\overline{W}$, A8-A15, PA0-PA4, PB0-PB7, B0-B7	C_{out}	–	12	pF

NOTE Test conditions for Quiescent Current Values are
Port A and B programmed as inputs
V_{IL} = 0.2 V for PA0-PA4, PB0-PB7, and B0-B7
V_{IH} = V_{DD} − 0.2 V for $\overline{RESET}$, $\overline{IRQ}$, and Timer
OSC1 input is a squarewave from V_{SS} + 0.2 V to V_{DD} − 0.2 V
OSC2 output load (including tester) is 35 pF maximum
Wait mode I_{DD} is affected linearly by this capacitance

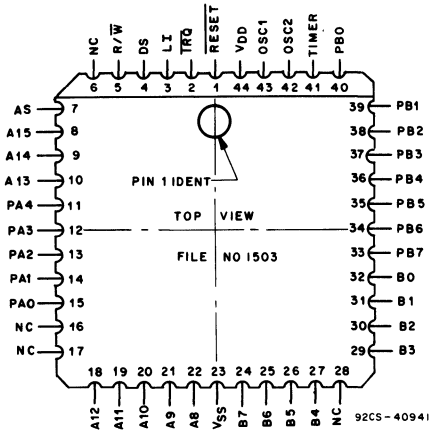
CDP6805E3, CDP6805E3C

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

Characteristics	Symbol	Min	Max	Unit
Output Voltage $I_{LOAD} \leq 10\ \mu A$	V_{OL} V_{OH}	– $V_{DD}-0.1$	0.1 –	V V
Total Supply Current (C_L = 130 pF – On Bus, C_L = 50 pF – On Ports, No DC Loads, t_{cyc} = 1 µs				
Run (V_{IL} = 0.2 V, V_{IH} = V_{DD} – 0.2 V)	I_{DD}	–	10	mA
Wait (Test Conditions – See Note Below)	I_{DD}	–	1.5	mA
Stop (Test Conditions – See Note Below)	I_{DD}	–	200	µA
Output High Voltage				
(I_{LOAD} = 1.6 mA) A8-A15, B0-B7	V_{OH}	4.1	–	V
(I_{LOAD} = 0.36 mA) PA0-PA4, PB0-PB7	V_{OH}	4.1	–	V
(I_{LOAD} = 1.6 mA) DS, AS, R/$\overline{W}$	V_{OH}	4.1	–	V
Output Low Voltage				
(I_{LOAD} = 1.6 mA) A8-A15, B0-B7	V_{OL}	–	0.4	V
(I_{LOAD} = 1.6 mA) PA0-PA4, PB0-PB7	V_{OL}	–	0.4	V
(I_{LOAD} = 1.6 mA) DS, AS, R/$\overline{W}$	V_{OL}	–	0.4	V
Input High Voltage				
PA0-PA4, PB0-PB7, B0-B7	V_{IH}	$V_{DD}-2$	–	V
TIMER, $\overline{IRQ}$, $\overline{RESET}$	V_{IH}	$V_{DD}-0.8$	–	V
OSC1	V_{IH}	$V_{DD}-1.5$	–	V
Input Low Voltage (All Inputs)	V_{IL}	–	0.8	V
Frequency of Operation				
Crystal	f_{OSC}	0.032	5	MHz
External Clock	f_{OSC}	DC	5	MHz
Input Current				
$\overline{RESET}$, $\overline{IRQ}$, Timer, OSC1	I_{in}	–	±1	µA
Three-State Output Leakage				
PA0-PA4, PB0-PB7, B0-B7	I_{TSI}	–	±10	µA
Capacitance				
$\overline{RESET}$, $\overline{IRQ}$, Timer	C_{in}	–	8	pF
Capacitance				
DS, AS, R/$\overline{W}$, A8-A15, PA0-PA4, PB0-PB7, B0-B7	C_{out}	–	12	pF

NOTE Test conditions for Quiescent Current Values are
- Port A and B programmed as inputs
- V_{IL} = 0.2 V for PA0-PA4, PB0-PB7, and B0-B7
- V_{IH} = V_{DD} – 0.2 V for $\overline{RESET}$, $\overline{IRQ}$, and Timer
- OSC1 input is a squarewave from V_{SS} + 0.2 V to V_{DD} – 0.2 V
- OSC2 output load (including tester) is 35 pF maximum
- Wait mode (I_{DD}) is affected linearly by this capacitance

TERMINAL ASSIGNMENT

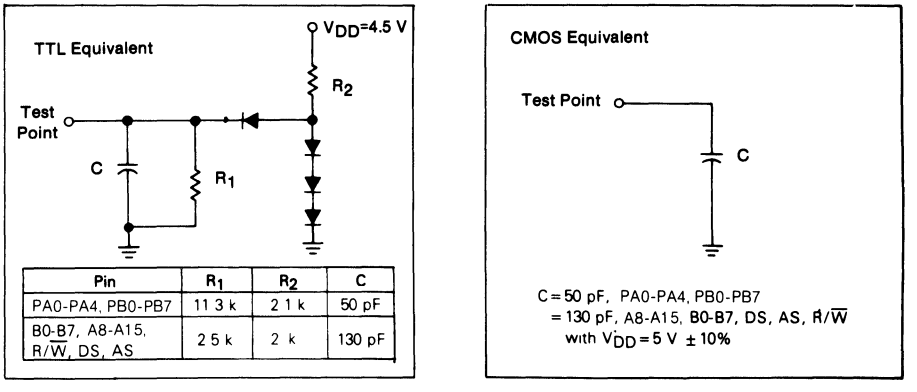


Plastic Chip-Carrier Package

TABLE 1 – CONTROL TIMING (V_{SS}=0, T_A=T_L to T_H)

		V_{DD} = 3 V f_{OSC} = 1 MHz			V_{DD} = 5 V ± 10% f_{OSC} = 5 MHz			
Characteristics	**Symbol**	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Unit**
I/O Port Timing – Input Setup Time (Figure 3)	t_{PVASL}	500	–	–	250	–	–	ns
Input Hold Time (Figure 3)	t_{ASLPX}	100	–	–	100	–	–	ns
Output Delay Time (Figure 3)	t_{ASLPV}	–	–	0	–	–	0	ns
Interrupt Setup Time (Figure 6)	t_{ILASL}	2	–	–	0 4	–	–	µs
Crystal Oscillator Startup Time (Figure 5)	t_{OXOV}	–	30	300	–	15	100	ms
Wait Recovery Startup Time (Figure 7)	t_{IVASH}	–	–	10	–	–	2	µs
Stop Recovery Startup Time (Crystal Oscillator) (Figure 8)	t_{ILASH}	–	30	300	–	15	100	ms
Required Interrupt Release (Figure 6)	t_{DSLIH}	–	–	5	–	–	1	µs
Timer Pulse Width (Figure 7)	t_{TH}, t_{TL}	0 5	–	–	0.5	–	–	t_{cyc}
Reset Pulse Width (Figure 5)	t_{RL}	5 2	–	–	1 05	–	–	µs
Timer Period (Figure 7)	t_{TLTL}	1	–	–	1	–	–	t_{cyc}
Interrupt Pulse Width Low (Figure 16)	t_{ILIH}	1	–	–	1	–	–	t_{cyc}
Interrupt Pulse Period (Figure 16)	t_{ILIL}	*	–	–	*	–	–	t_{cyc}
Oscillator Cycle Period (1/5 of t_{cyc})	t_{OLOL}	1000	–	–	200	–	–	ms
OSC1 Pulse Width High	t_{OH}	350	–	–	75	–	–	ns
OSC1 Pulse Width Low	t_{OL}	350	–	–	75	–	–	ns

*The minimum 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



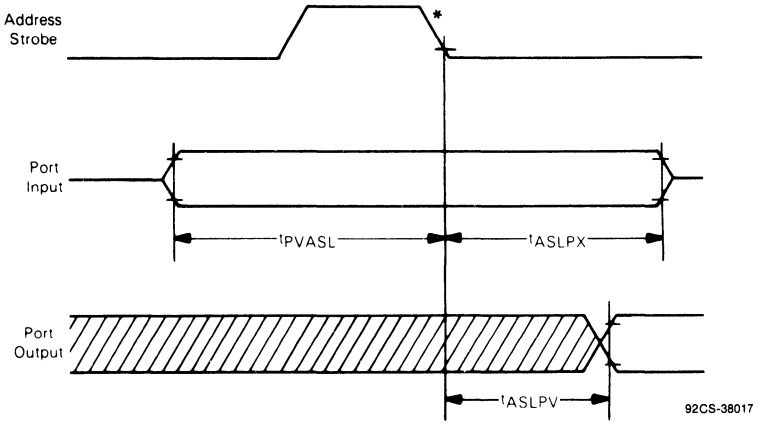
Pin	R_1	R_2	C
PA0-PA4, PB0-PB7	11 3 k	2 1 k	50 pF
B0-B7, A8-A15, R/$\overline{W}$, DS, AS	2 5 k	2 k	130 pF

92CS-38351

Fig. 2 – Equivalent test-load circuits.

CDP6805E3, CDP6805E3C

(V_{LOW} = 0.8 V, V_{HIGH} = V_{DD} − 2 V, V_{DD} = 5 ± 10%
Temp = 0° to 70°C, C_L on Port = 50 pF, f_{OSC} = 5 MHz)

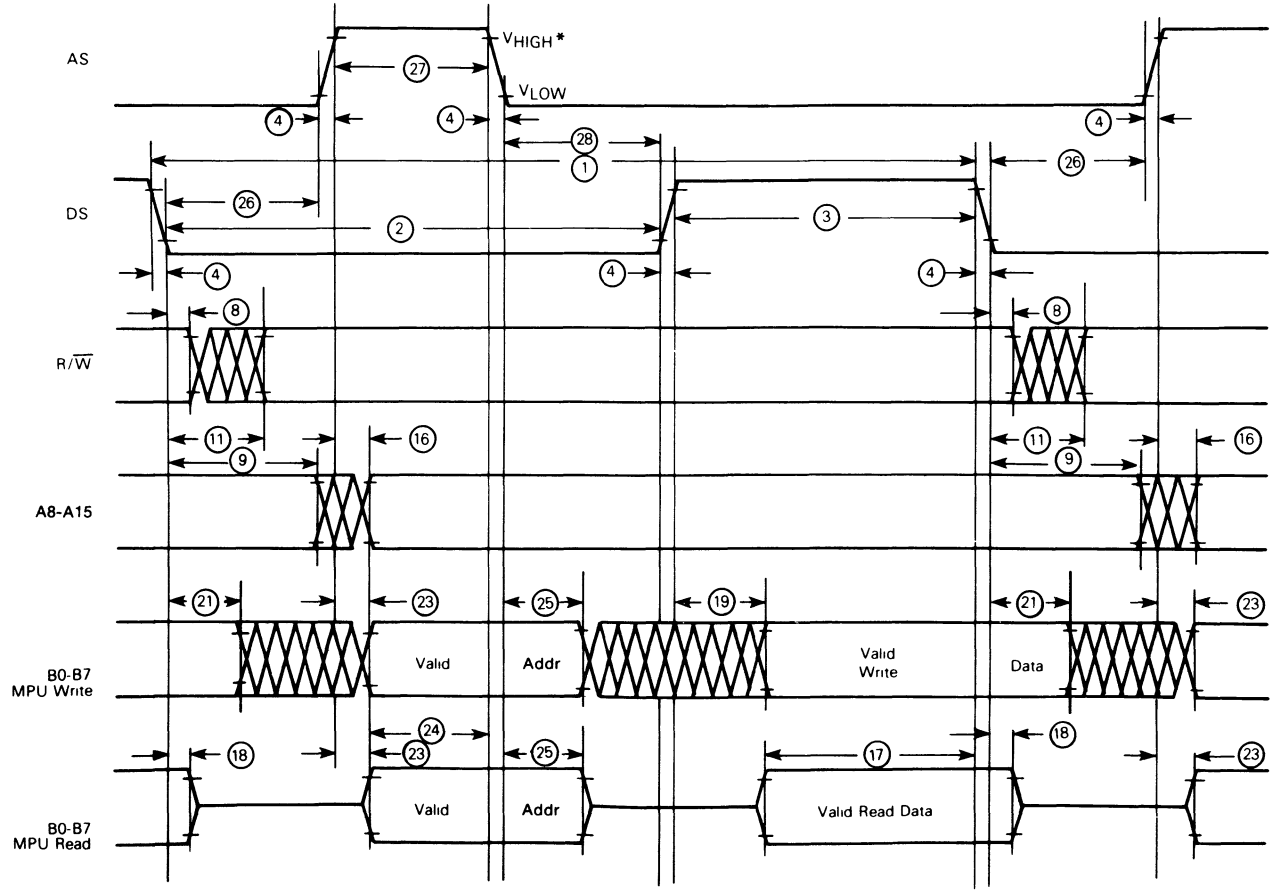


*The address strobe of the first cycle of the next instruction as shown in Table 11

Fig. 3 - I/O port timing waveforms.

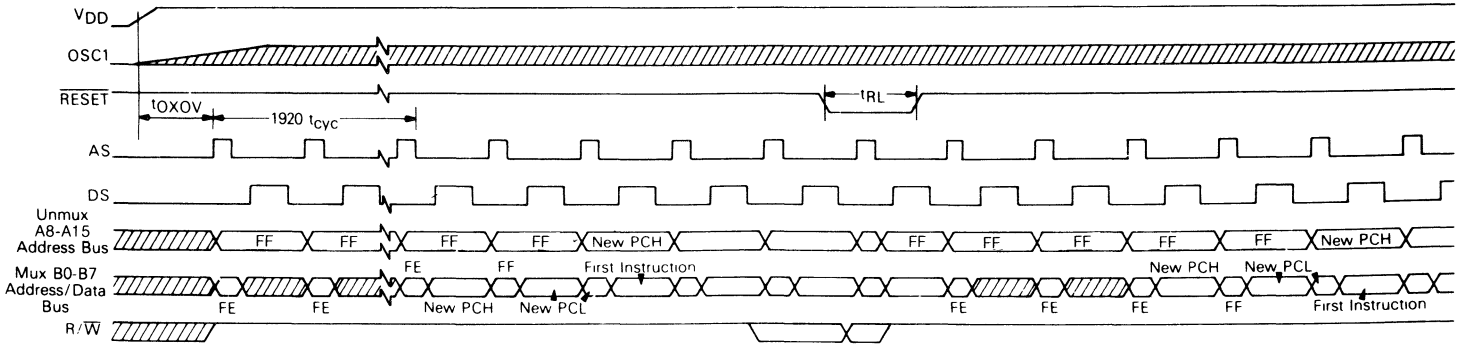
TABLE 2 — BUS TIMING (T_A=T_L to T_H, V_{SS}=0 V) See Figure 4

Num	Characteristics	Symbol	f_{OSC} = 1 MHz, V_{DD} = 3 V 50 pF Load		f_{OSC} = 5 MHz V_{DD} = 5 V ± 10%, 1 TTL and 130 pF Load		Unit
			Min	Max	Min	Max	
1	Cycle Time	t_{cyc}	5000	DC	1000	DC	ns
2	Pulse Width, DS Low	PW_{EL}	2800	–	560	–	ns
3	Pulse Width, DS High or $\overline{RD}$, $\overline{WR}$, Low	PW_{EH}	1800	–	375	–	ns
4	Clock Transition	t_r, t_f	–	100	–	30	ns
8	R/$\overline{W}$ Hold	t_{RWH}	10	–	10	–	ns
9	Non-Muxed Address Hold	t_{AH}	800	–	100	–	ns
11	R/$\overline{W}$ Delay from DS Fall	t_{AD}	–	500	–	300	ns
16	Non-Muxed Address Delay from AS Rise	t_{ADH}	0	200	0	100	ns
17	MPU Read Data Setup	t_{DSR}	200	–	115	–	ns
18	Read Data Hold	t_{DHR}	0	1000	0	160	ns
19	MPU Data Delay, Write	t_{DDW}	–	0	–	120	ns
21	Write Data Hold	t_{DHW}	800	–	55	–	ns
23	Muxed Address Delay from AS Rise	t_{BHD}	0	250	0	120	ns
24	Muxed Address Valid to AS Fall	t_{ASL}	600	–	55	–	ns
25	Muxed Address Hold	t_{AHL}	250	750	60	180	ns
26	Delay DS Fall to AS Rise	t_{ASD}	800	–	160	–	ns
27	Pulse Width, AS High	PW_{ASH}	850	–	175	–	ns
28	Delay, AS Fall to DS Rise	t_{ASED}	800	–	160	–	ns

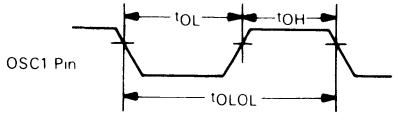


* V_{HIGH} = -2 V, V_{LOW} = 0.5 V for V_{DD} = 3 V
V_{HIGH} = V_{DD} -2 V, V_{LOW} = 0.8 V for V_{DD} = 5 V ± 10 %

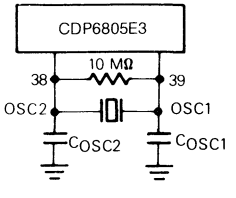
Fig. 4 - Bus timing waveforms.



Oscillator Waveform



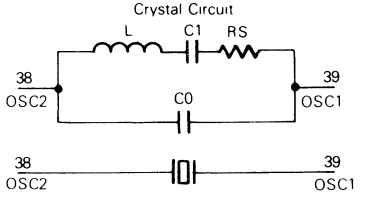
Crystal Oscillator Connections



Crystal Parameters Representative Frequencies

	5 MHz	4 MHz	1 MHz
RS max	50Ω	75Ω	400Ω
C0	8 pF	7 pF	5 pF
C1	0 02 pF	0 012 pF	0 008 pF
Q	50 k	40 k	30 k
C_{OSC1}	15-30 pF	15-30 pF	15-40 pF
C_{OSC2}	15-25 pF	15-25 pF	15-30 pF

Crystal Circuit



92CS-38353

Fig. 5 – Power-on reset and reset timing waveforms.

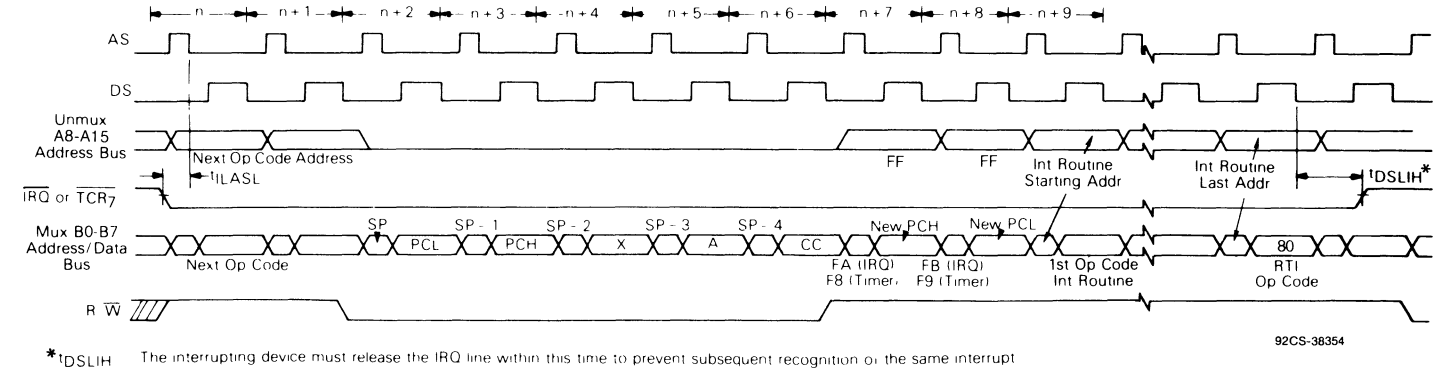


Fig. 6 - $\overline{IRQ}$ and $\overline{TCR_7}$ interrupt timing waveforms.

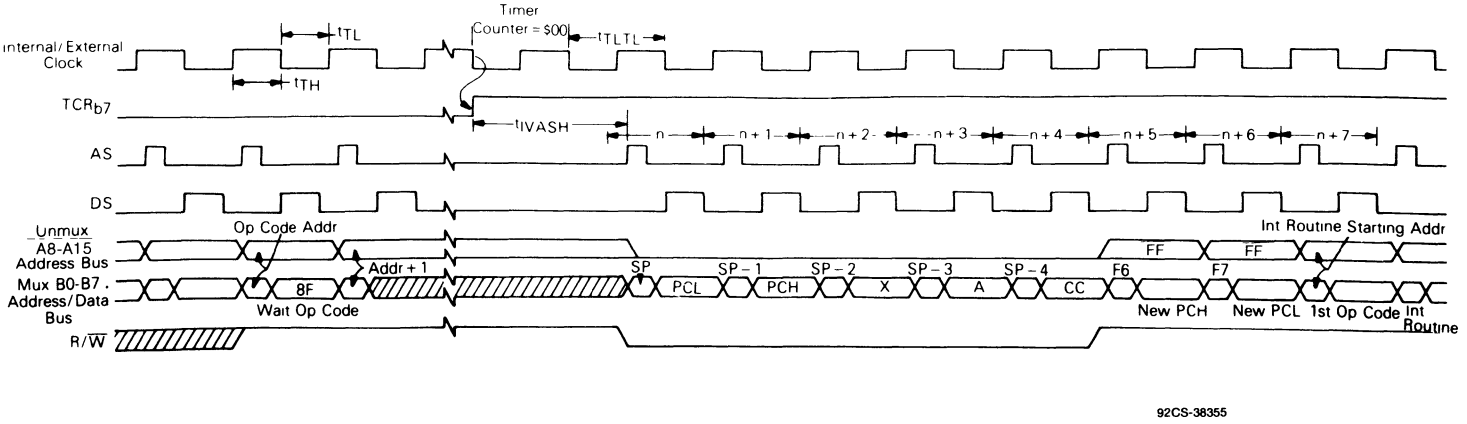
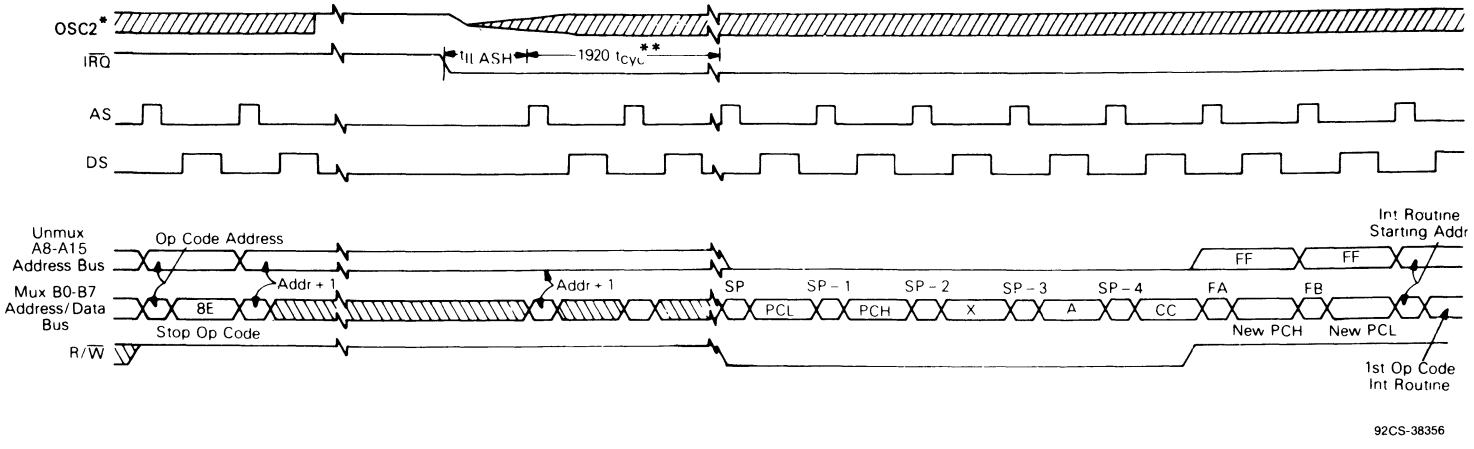


Fig. 7 - Timer interrupt after WAIT instruction timing waveforms.



*Represents the internal gating of the OSC1 input pin

**t_{cyc} is one instruction cycle (for $f_{OSC} = 5$ MHz, $t_{cyc} = 1\ \mu s$)

Fig. 8 - Interrupt recovery from STOP instruction timing waveforms.

FUNCTIONAL PIN DESCRIPTION

V_{DD} and V_{SS} – V_{DD} and V_{SS} provide power to the chip V_{DD} provides power and V_{SS} is ground

$\overline{IRQ}$ (Maskable Interrupt Request) – $\overline{IRQ}$ is a level-sensitive and edge sensitive input which can be used to request an interrupt sequence The MPU completes the current instruction before it responds to the request IF $\overline{IRQ}$ is low and the interrupt mask bit (I-bit) in the Condition Code Register is clear, the MPU begins an interrupt sequence at the end of the current instruction The interrupt circuit recognizes both a "Wire ORed" level as well as pulses on the $\overline{IRQ}$ line (see Interrupt Section for more details) $\overline{IRQ}$ requires an external resistor to V_{DD} for "Wire OR" operation

$\overline{RESET}$ – The $\overline{RESET}$ input is not required for start-up but can be used to reset the MPU's internal state and provide an orderly software start-up procedure Refer to the RESET section for a detailed description

TIMER – The TIMER input is used for clocking the on-chip timer Refer to TIMER section for a detailed description

AS (Address Strobe) – Address Strobe (AS) is an output strobe used to indicate the presence of an address on the 8-bit multiplexed bus The AS line is used to demultiplex the eight least significant address bits from the data bus A latch controlled by Address Strobe should capture addresses on the negative edge This output is capable of driving one standard TTL load and 130 pF and is available at $f_{OSC} - 5$ when the MPU is not in the WAIT or STOP states

DS (Data Strobe) – This output is used to transfer data to or from a peripheral or memory DS occurs anytime the MPU does a data read or write DS also occurs when the MPU does a data transfer to or from the MPU's internal memory Refer to Table 2 and Figure 4 for timing characteristics This output is capable of driving one standard TTL load and 130 pF DS is a continuous signal at $f_{OSC} \div 5$ when the MPU is not in WAIT or STOP state. Some bus cycles are redundant reads of op code bytes

R/$\overline{W}$ (Read/Write) – The R/$\overline{W}$ output is used to indicate the direction of data transfer for both internal memory and I/O registers, and external peripheral devices and memories. This output is used to indicate to a selected peripheral whether the MPU is going to read or write data on the next Data Strobe (R/$\overline{W}$ low = processor write; R/$\overline{W}$ high = processor read) The R/$\overline{W}$ output is capable of driving one standard TTL load and 130 pF The normal standby state is Read (high)

A8-A15 (High Order Address Lines) — The A8-A15 output lines constitute the higher order non-multiplexed addresses Each output line is capable of driving one standard TTL load and 130 pF

B0-B7 (Address/Data Bus) – The B0-B7 bidirectional lines constitute the lower order addresses and data. These lines are multiplexed, with address present at Address Strobe time and data present at Data Strobe time. When in the data mode, these lines are bidirectional, transferring data to and from memory and peripheral devices as indicated by the R/$\overline{W}$ pin As outputs in either the data or address modes, these lines are capable of driving one standard TTL load and 130 pF

OSC1, OSC2 — The CDP6805E3 provides for two types of oscillator inputs — crystal circuit or external clock. The two oscillator pins are used to interface to a crystal circuit, as shown in Figure 5 If an external clock is used, it must be connected to OSC1 The input at these pins is divided by five to form the cycle rate seen on the AS and DS pins The frequency range is specified by f_{OSC} The OSC1 to bus transitions relationships are provided in Figure 9 for system designs using oscillators slower than 5 MHz

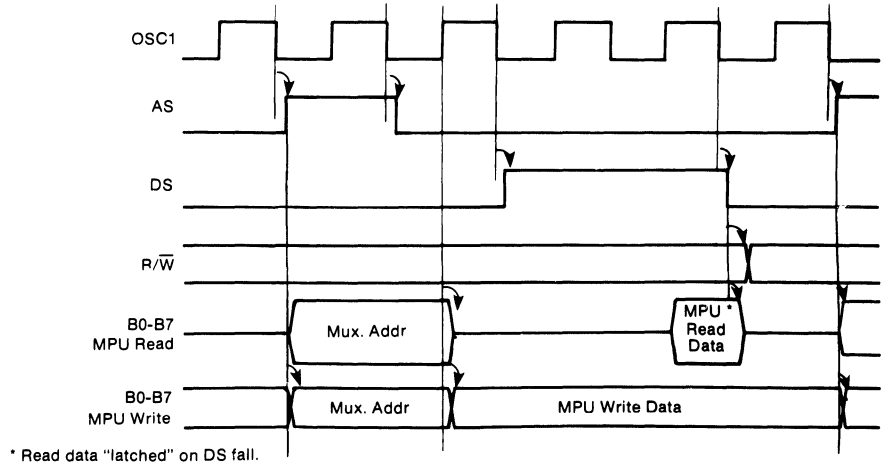


Fig. 9 - OSC1 to bus transitions timing waveforms.

CDP6805E3,. CDP6805E3C

Crystal – The circuit shown in Figure 5 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 electrical characteristics table. An external CMOS oscillator is recommended 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

External Clock – An external clock should be applied to the OSC1 input with the OSC2 input not connected, as shown in Figure 10

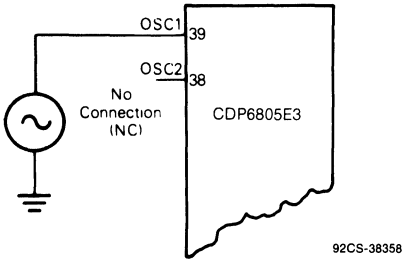
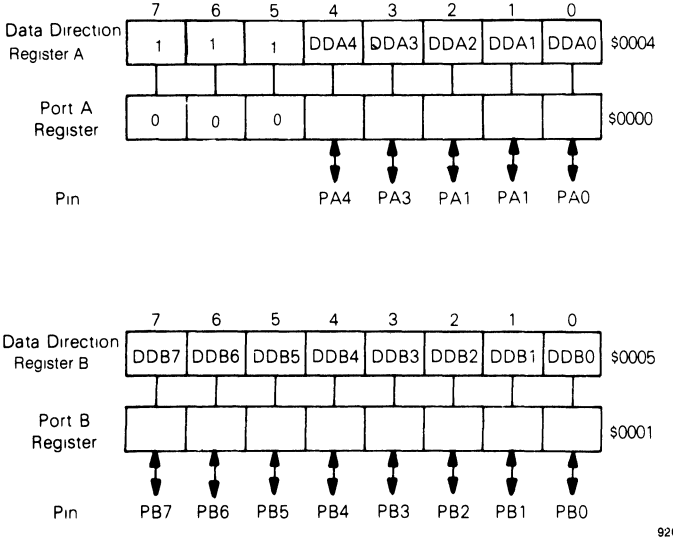


Fig. 10 - External clock connection.

LI (Load Instruction) – This output is used to indicate that a fetch of the next opcode is in progress LI remains low during an External or Timer interrupt The LI output is only used for certain debugging and test systems For normal operations this pin is not connected The LI output is capable of driving one standard TTL load and 50 pF This signal overlaps Data Strobe

PA0-PA4 — These five pins constitute Input/Output Port A. Each line is individually programmed to be either an input or output under software control via its Data Direction Register as shown below An I/O pin is programmed as an output when the corresponding DDR bit is set to a "1", and as an input when it is set to a "0" In the output mode the bits are latched and appear on the corresponding output pins An MPU read of the port bits programmed as outputs reflect the last value written to that location When programmed as an input, the input data bit(s) are not latched An MPU read of the port bits programmed as inputs reflects the current status of the corresponding input pins The Read/Write port timing is shown in Figure 3 See typical I/O Port Circuitry in Figure 11 During a Power-On Reset or external $\overline{\text{RESET}}$ all lines are configured as inputs (zero in Data Direction Register) The latched output data is not initialized by reset The TTL compatible three-state output buffers are capable of driving one standard TTL load and 50 pF. The DDR is a read/write register Bits 7-5 of the DDR "A" will be always read as "1" and bits 7-5 of the Port "A" Register will be read as "0"

PB0-PB7 – These eight pins interface to Input/Output Port B Refer to PA0-PA4 description for details of operation



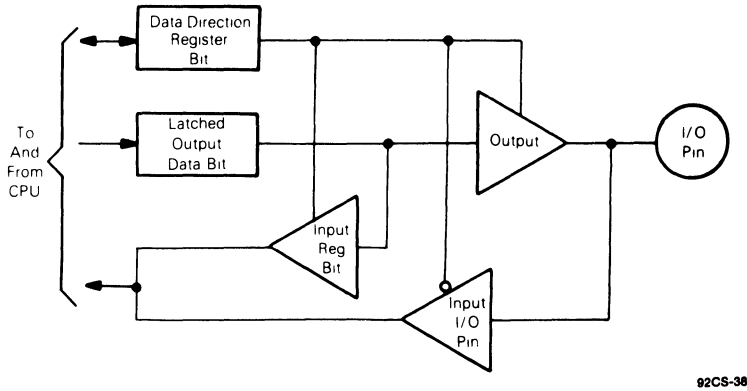


Fig. 11 - Typical I/O port circuitry.

TABLE 3 – I/O PIN FUNCTIONS

R/$\overline{W}$	DDR	I/O Pin Functions
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.

MEMORY ADDRESSING

The CDP6805E3 is capable of addressing 65536 bytes of memory and I/O registers. The address space is divided into internal memory space and external memory space, as shown in Figure 12.

The internal memory space is located within the first 128 bytes of memory (first half of page zero) and is comprised of the I/O port locations, timer locations, and 112 bytes of RAM. The MPU can read from or write to any of these locations. A program write to on-chip locations is repeated on the external bus to permit off-chip memory to duplicate the content of on-chip memory. Program reads to on-chip locations also appear on the external bus, but the MPU accepts data only from the addressed on-chip location. Any read data appearing on the input bus is ignored.

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 onto the stack. When data is removed from the stack, the stack pointer is incremented. A maximum of 64 bytes of RAM is available for stack usage. Since most programs use only a small part of the allotted stack locations for interrupts and/or subroutine stacking purposes, the unused bytes are usable for program data storage.

All memory locations above location $007F are part of the external memory map. In addition, ten locations in the I/O portion of the lower 128 bytes of memory space, as shown in Figure 12, are part of the external memory map. All of the external memory space is user definable except the highest 10 locations. Locations $FFF6 to $FFFF of the external address space are reserved for interrupt and reset vectors (see Figure 12).

REGISTERS

The CDP6805E3 contains five registers as shown in the programming model in Figure 13. The interrupt stacking order is shown in Figure 14.

ACCUMULATOR (A) – This Accumulator is an 8-bit general purpose register used for 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 an 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 a 16-bit register that contains the address of the next instruction to be executed by the processor.

CDP6805E3, CDP6805E3C

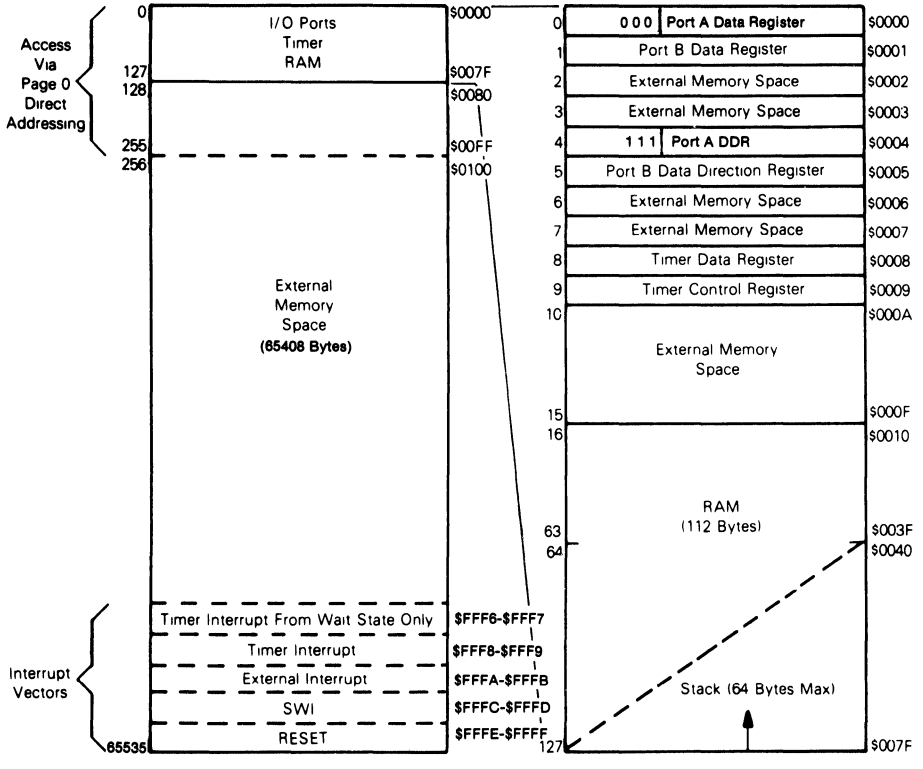


Fig. 12 - Address map.

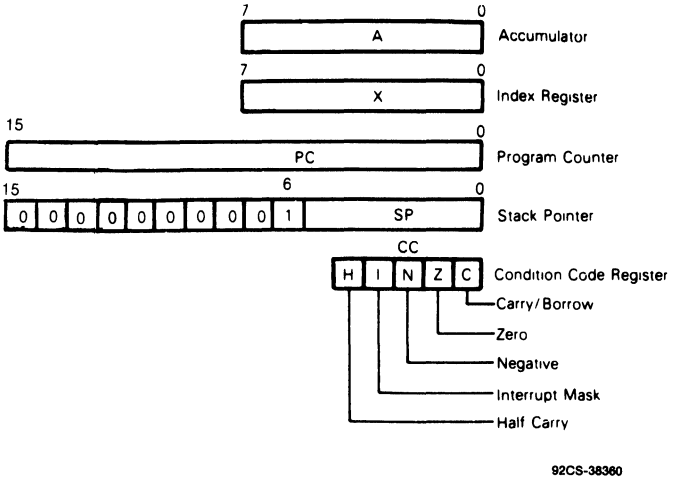


Fig. 13 - Programming model.

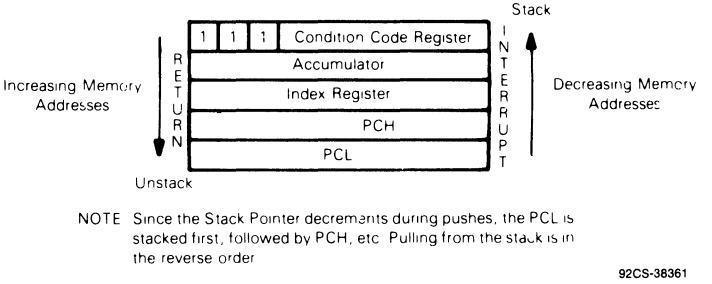


Fig. 14 - Stacking order.

STACK POINTER (SP) — The stack pointer is a 16-bit register containing the address of the next free location on the stack. When accessing memory, the ten most-significant bits are permanently set to 000 000 0001. They are appended to the six least-significant register bits to produce an address within the range of $007F to $0040. 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 ($007F). Nested interrupts and/or subroutines may be use up to 64 (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.

CONDITION CODE REGISTER (CC) — The condition code register is a 5-bit register in which each bit is used to indicate 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 of the five bits is explained below.

Half Carry Bit (H) — The H-bit is set to a one 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 addition 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 will be processed when the I-bit is next cleared.

Negative Bit (N) — When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was negative (bit 7 in the result is a logical one).

Zero Bit (Z) — When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was zero.

CDP6805E3, CDP6805E3C

Carry Bit (C) – The C-bit is set when a carry or a borrow out of the ALU occurs during an arithmetic instruction The C-bit is also modified during bit test, shift, rotate, and branch types of instruction.

RESETS

The CDP6805E3 has two reset modes: an active low external reset pin ($\overline{RESET}$) and a Power-On Reset function, refer to Figure 5

$\overline{RESET}$ (Pin #1) – The $\overline{RESET}$ input pin is used to reset the MPU and provide an orderly software start-up procedure. When using the external reset mode, the $\overline{RESET}$ pin must stay low for a minimum of one t_{CYC} The $\overline{RESET}$ pin is provided with a Schmitt Trigger to improve its noise immunity capability.

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 reset pin is low at the end of the 1920 t_{CYC} 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 (bit 7) is cleared to a "0"
- Timer control register interrupt mask bit (bit 6) is set to a "1".
- All data direction register bits are cleared to a "0" (inputs) - except Port "A" DDR Bits 7-5.
- Stack pointer is set to $007F
- The address bus is forced to the reset vector ($FFFE, $FFFF).
- 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

The CDP6805E3 is capable of operation with three different interrupts, two hardware (timer interrupt and external interrupt) and one software (SWI) When any of these interrupts occur, normal processing is suspended at the end of the current instruction execution All of the program registers (the machine state) are pushed onto the stack, refer to Figure 14 for stacking order The appropriate vector pointing to the starting address of the interrupt service routine is then fetched; refer to Figure 15 for the interrupt sequence

The priority of the various interrupts from highest to lowest is as follows

RESET→*→External Interrupt→Timer Interrupt

TIMER INTERRUPT – If the timer mask bit (TCR6) is cleared, then each time the timer decrements to zero (transitions from $01 to $00) an interrupt request is generated The actual processor interrupt is generated only if the interrupt mask bit of the condition code register is also cleared. When the interrupt is recognized, the current state of the machine is pushed onto the stack and the I-bit in the condition code register is set. This masks 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 $FFF8 and $FFF9. The contents of $FFF6 and $FFF7 specify the service routine if the processor is in the WAIT mode. Also, 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.

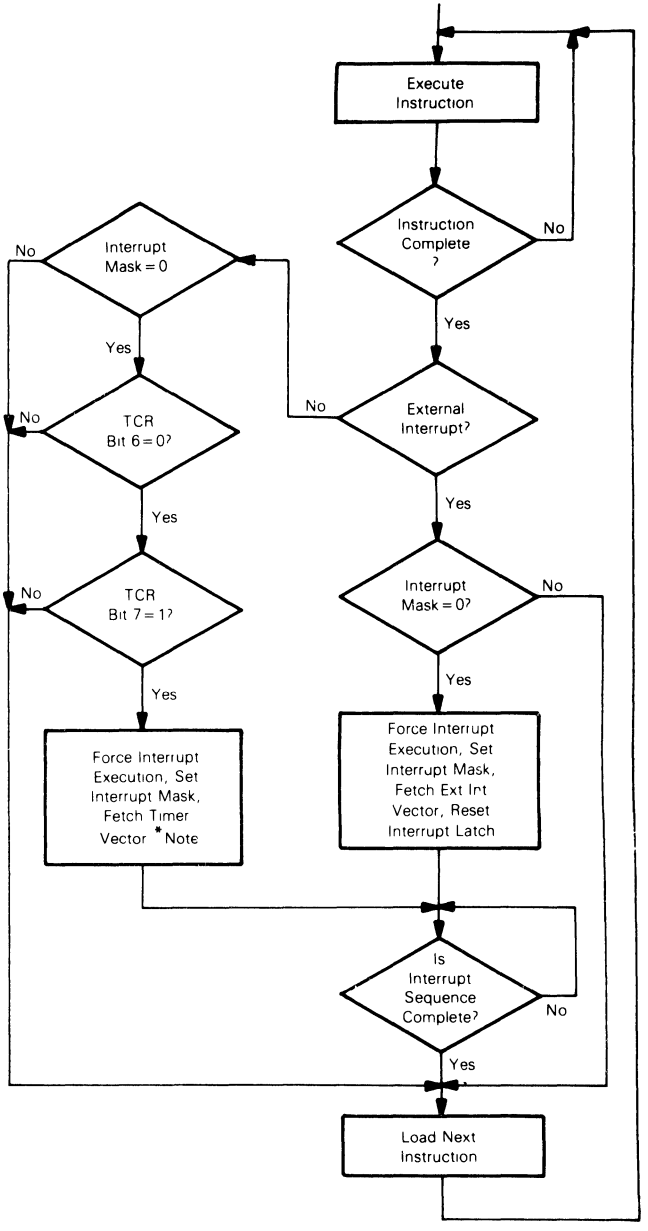
EXTERNAL INTERRUPT – If the interrupt mask bit of the condition code register is cleared and the external interrupt pin $\overline{IRQ}$ is "low," then the external interrupt occurs The action of the external interrupt is identical to the timer interrupt with the exception that the service routine address is specified by the contents of $FFFA and $FFFB. The interrupt logic recognizes both a "wire ORed" level and pulses on the external interrupt line Figure 16 shows both a functional diagram and timing for the interrupt line The timing diagram shows two different treatments of the interrupt line ($\overline{IRQ}$) to the processor The first 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 The second method is single pulses on the interrupt line spaced far enough apart to be service. 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 (one cycle $t_{CYC}=5/f_{OSC}$) to the total number of cycles it takes to complete the service routine including the RTI instruction; refer to Figure 6.

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 $FFFC and $FFFD See Figure 15 for Interrupt and Instruction Processing Flowchart.

The following three functions are not strictly interrupts, however, they are tied very closely to the interrupts These functions are $\overline{RESET}$, STOP, 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 of memory locations $FFFE and $FFFF The interrupt mask of the condition code register is also set. Refer to RESET selection for details

*Any current instruction including SWI



*NOTE The clear of TCR bit 7 must be accomplished with software

92CS-38030

Fig. 15 - Interrupt and instruction processing flowchart.

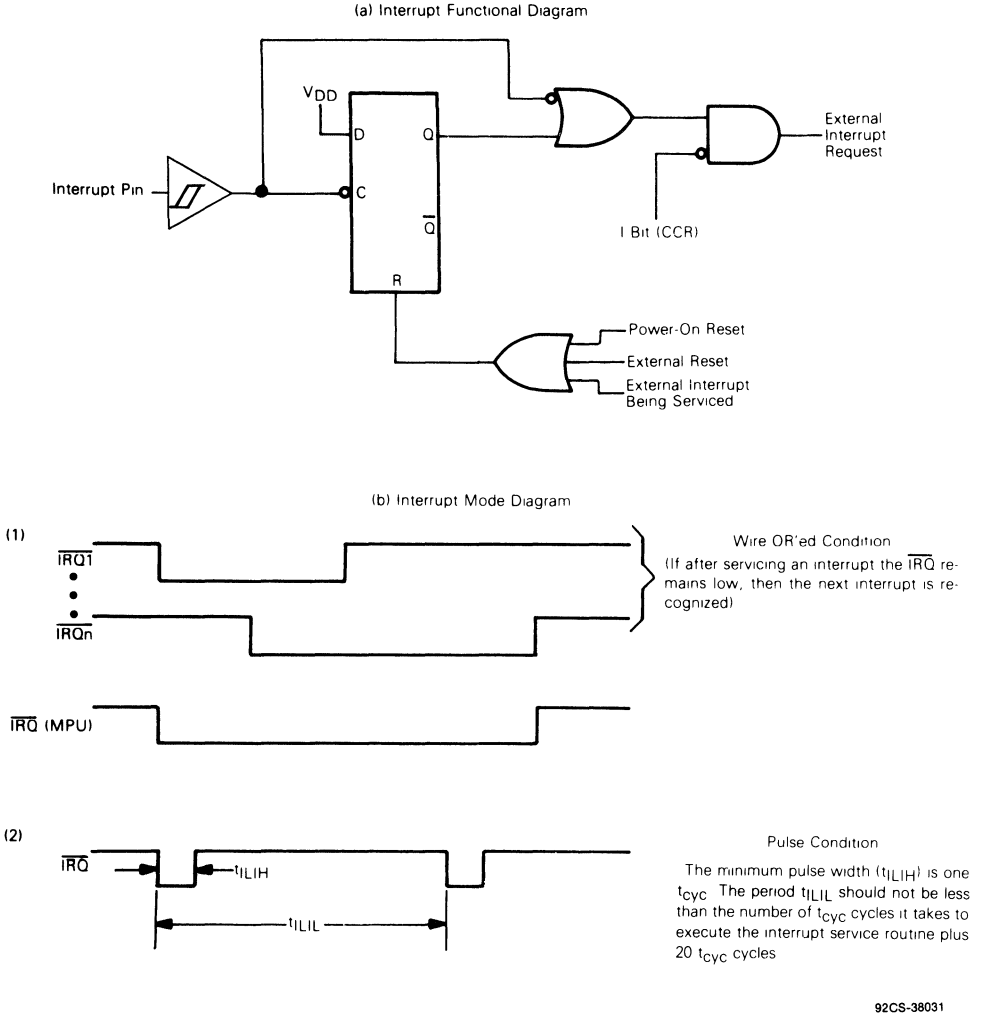


Fig. 16 - External interrupt.

STOP — The STOP instruction places the CDP6805E3 in a low 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 17. The DS and AS lines go to a low state and the R/$\overline{W}$ line goes to a high state The multiplexed address/data bus goes to the data input state The high order address lines remain at the address of the next instruction The MPU remains in the STOP mode until an external interrupt or reset occurs; refer to Figure 8 and 17.

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 timer interrupts External interrupts are enabled in the condition code register All other registers and memory remain unaltered All I/O lines remain unchanged

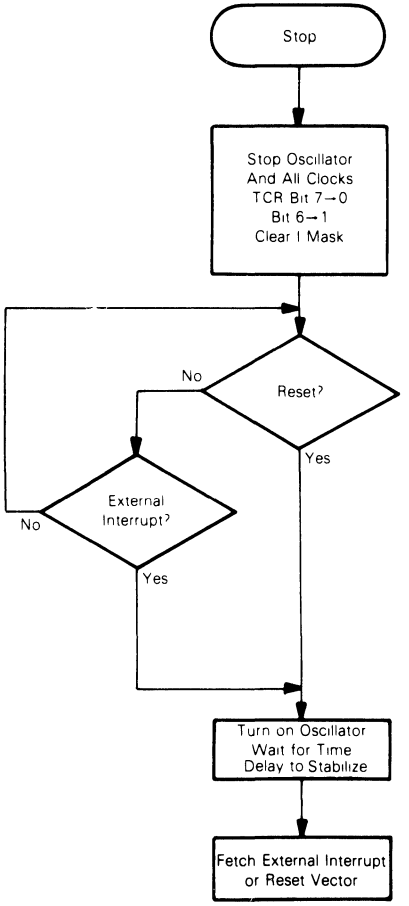


Fig. 17 - Stop function flowchart.

WAIT – The WAIT instruction places the CDP6805E3 in a low power consumption mode, but the WAIT mode consumes somewhat more power than the STOP mode; refer to Table 1 In the WAIT function, the internal clock is disabled from all internal circuitry except the Timer circuit; refer to Figure 18 Thus, all internal processing is halted except the Timer which is allowed to count in a normal sequence. The R/$\overline{W}$ line goes to a high state, the multiplexed address/data bus goes to the data input state, and the DS and AS lines go to the low state The high order address lines remain at the address of the next instruction. The MPU remains in this state until an external interrupt, timer interrupt, or a reset occurs, refer to Figures 7 and 18

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 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

3

TIMER

The MPU timer contains a single 8-bit software programmable counter with 7-bit software selectable prescaler. 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 $FFF8 and $FFF9 in order to begin servicing the interrupt, unless it was in locations $FFF6 and $FFF7 the WAIT mode

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 becomes stable prior to the read portion of a cycle and does not change during the read The timer interrupt request bit remains set until cleared by the software If this happens before the timer interrupt is serviced, the interrupt is lost. 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 which is used as the counter input The processor cannot write into or read from the prescaler, however, its contents are cleared to all "0's" by the write operation into TCR when bit 3 of the written data equals 1, which 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, TCR5 control bits. Refer to the TIMER CONTROL REGISTER section.

Timer Input Mode 1 – If TCR4 and TCR5 are both programmed to a "0", the input to the Timer is from an internal clock and the Timer input is disabled. The internal clock mode can be used for periodic interrupt generation, as well

92CS-38032

CDP6805E3, CDP6805E3C

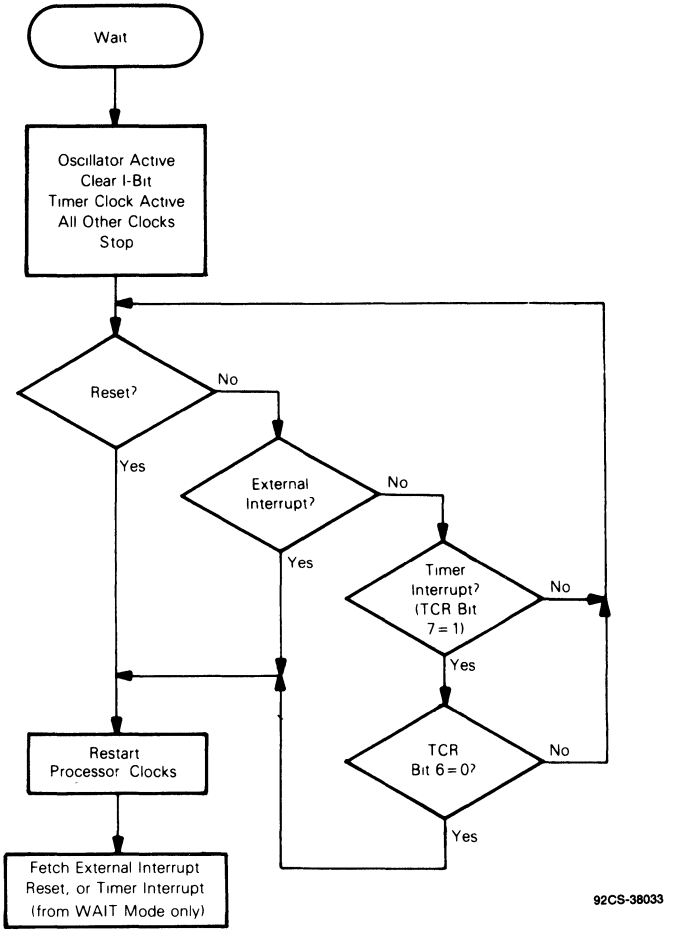


Fig. 18 - Wait function flowchart.

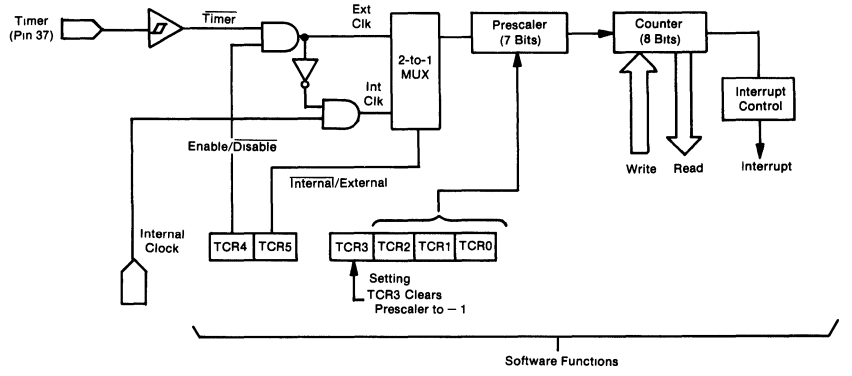
as a reference in frequency and event measurement The internal clock is the instruction cycle clock and is coincident with Address Strobe (AS) except during a WAIT instruction. During a WAIT instruction the AS pin goes to a low state but the internal clock to the Timer continues to run at its normal rate.

Timer Input Mode 2 – With TCR4 = 1 and TCR5 = 0, the internal clock and the TIMER input pin are ANDed together to form the Timer input signal. This mode can be used to measure external pulse widths The external pulse simply turns on the internal clock for the duration of the pulse. The resolution of the count in this mode is ±1 clock and therefore accuracy improves with longer input pulse widths.

Timer Input Mode 3 – If TCR4 = 0 and TCR5 = 1, then all inputs to the Timer are disabled

Timer Input Mode 4 – If TCR4 = 1 and TCR5 = 1, the internal clock input to the Timer is disabled and the TIMER input pin becomes the input to the Timer. The external Timer pin can, in this mode, be used to count external events as well as external frequencies for generating periodic interrupts.

Figure 19 shows a block diagram of the Timer subsystem Power-on Reset and the STOP instruction cause the counter to be set to $FO



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. 19 - Timer 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 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 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 (AS)

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 (AS) to Timer
0	1	AND of internal clock (AS) and TIMER pin to Timer
1	0	Inputs to Timer disabled
1	1	TIMER pin to Timer

Refer to Figure 19 for Logic Representation

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

TCR2, TCR1, TCR0 – Prescaler address bits decoded to select one of eight taps 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

CDP6805E3, CDP6805E3C

INSTRUCTION SET

The MPU 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 jump unconditional (JMP) and jump to subroutine (JSR) instructions have no register operand. 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 – This set of instructions branches if a particular condition is met, otherwise no operation is performed. Branch instructions are two byte instructions Refer to Table 6

BIT MANIPULATION INSTRUCTIONS – The MPU is capable of setting or clearing any bit which resides in the first 256 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 these 256 locations The bit set, bit clear and bit test and branch functions are all 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 for instruction cycle timing

CONTROL INSTRUCTIONS – These instructions are register reference instructions and are used to control processor operation during program execution Refer to Table 8 for instruction cycle timing

ALPHABETICAL LISTING – The complete instruction set is given in alphabetical order in Table 9

OPCODE MAP SUMMARY – Table 10 is an opcode map for the instructions used on the MCU

ADDRESSING MODES

The MPU uses ten different addressing modes to give the programmer 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 9 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 10.

The term "Effective Address" or EA is used in describing the various addressing modes, which is defined as the 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 register 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 up to 128 bytes of off-chip ROM Direct addressing is efficient in both memory and speed

$$EA = (PC + 1),\ PC \leftarrow PC + 2$$
$$\text{Address Bus High} \leftarrow 0,\ \text{Address Bus Low} \leftarrow (PC + 1)$$

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

$$EA = X,\ PC \leftarrow PC + 1$$
$$\text{Address Bus High} \leftarrow 0,\ \text{Address Bus Low} \leftarrow X$$

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

CDP6805E3, CDP6805E3C

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

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			Λ	●	Λ	Λ	Λ
ADD		X	X	X		X	X	X			Λ	●	Λ	Λ	Λ
AND		X	X	X		X	X	X			●	●	Λ	Λ	●
ASL	X		X			X	X				●	●	Λ	Λ	Λ
ASR	X		X			X	X				●	●	Λ	Λ	Λ
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			●	●	Λ	Λ	●
BLO					X						●	●	●	●	●
BLS					X						●	●	●	●	●
BMC					X						●	●	●	●	●
BMI					X						●	●	●	●	●
BMS					X						●	●	●	●	●
BNE					X						●	●	●	●	●
BPL					X						●	●	●	●	●
BRA					X						●	●	●	●	●
BRN					X						●	●	●	●	●
BRCLR										X	●	●	●	●	Λ
BRSET										X	●	●	●	●	Λ
BSET									X		●	●	●	●	●
BSR					X						●	●	●	●	●
CLC	X										●	●	●	●	0
CLI	X										●	0	●	●	●
CLR	X		X			X	X				●	●	0	1	●
CMP		X	X	X		X	X	X			●	●	Λ	Λ	Λ
COM	X		X			X	X				●	●	Λ	Λ	1
CPX		X	X	X		X	X	X			●	●	Λ	Λ	Λ
DEC	X		X			X	X				●	●	Λ	Λ	●
EOR		X	X	X		X	X	X			●	●	Λ	Λ	●
INC	X		X			X	X				●	●	Λ	Λ	●
JMP			X	X		X	X	X			●	●	●	●	●
JSR			X	X		X	X	X			●	●	●	●	●
LDA		X	X	X		X	X	X			●	●	Λ	Λ	●
LDX		X	X	X		X	X	X			●	●	Λ	Λ	●
LSL	X		X			X	X				●	●	Λ	Λ	Λ
LSR	X		X			X	X				●	●	0	Λ	Λ
NEG	X		X			X	X				●	●	Λ	Λ	Λ
NOP	X										●	●	●	●	●
ORA		X	X	X		X	X	X			●	●	Λ	Λ	●
ROL	X		X			X	X				●	●	Λ	Λ	Λ
ROR	X		X			X	X				●	●	Λ	Λ	Λ
RSP	X										●	●	●	●	●
RTI	X										?	?	?	?	?
RTS	X										●	●	●	●	●
SBC		X	X	X		X	X	X			●	●	Λ	Λ	Λ
SEC	X										●	●	●	●	1
SEI	X										●	1	●	●	●
STA			X	X		X	X	X			●	●	Λ	Λ	●
STOP	X										●	0	●	●	●
STX			X	X		X	X	X			●	●	Λ	Λ	●
SUB		X	X	X		X	X	X			●	●	Λ	Λ	Λ
SWI	X										●	1	●	●	●
TAX	X										●	●	●	●	●
TST	X		X			X	X				●	●	Λ	Λ	●
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
- Λ Test and Set if True Cleared Otherwise
- ● Not Affected
- ? Load CC Register From Stack
- 0 Cleared
- 1 Set

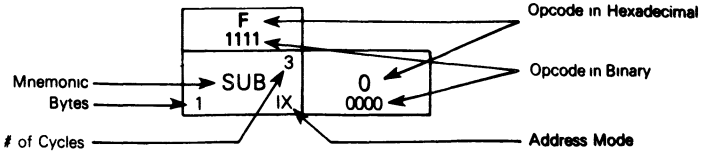
TABLE 10 — CDP6805E3 INSTRUCTION SET OPCODE MAP

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

LEGEND



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 The operand is therefore located anywhere within the lowest 511 memory locations. For example, this mode of addressing is useful for selecting the m-th element in an n element table All instructions are two bytes The contents of the index register (X) is not changed The contents 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 256 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 to address 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 clear 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

SYSTEM CONFIGURATION

Figures 20 through 25 show in general terms how the CDP6805E3 bus structure may be utilized. Specified interface details vary with the various peripheral and memory devices employed

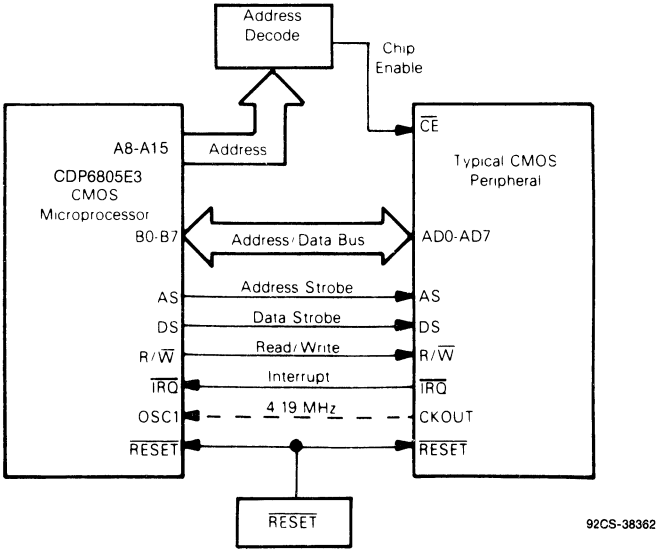


Fig. 20 - Connection to CMOS peripherals.

CDP6805E3, CDP6805E3C

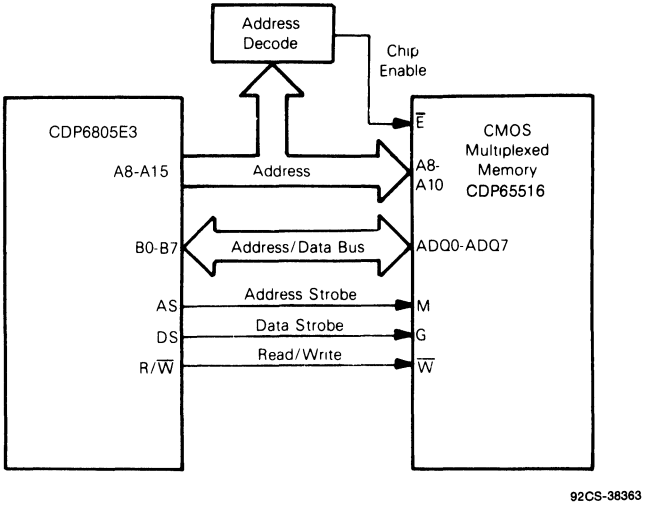
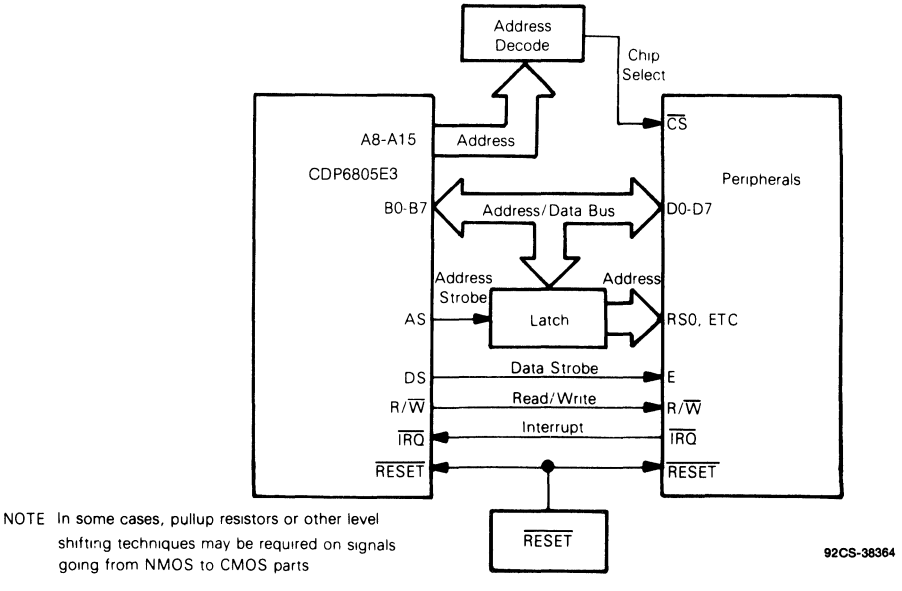
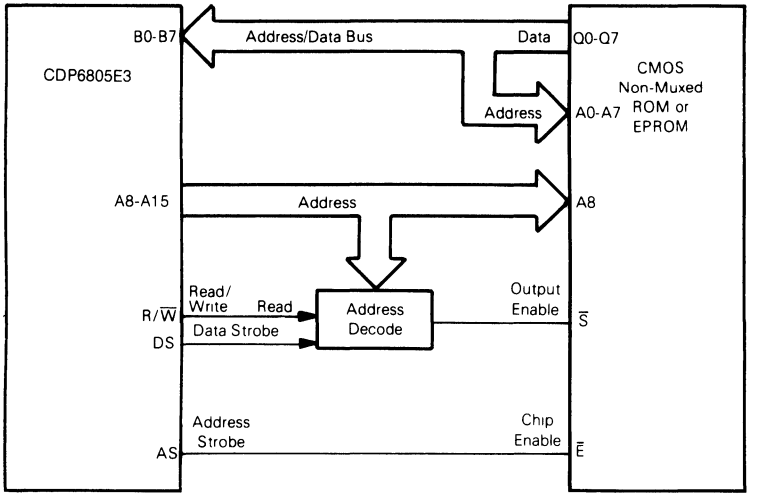


Fig. 21 - Connection to CMOS multiplexed memories.



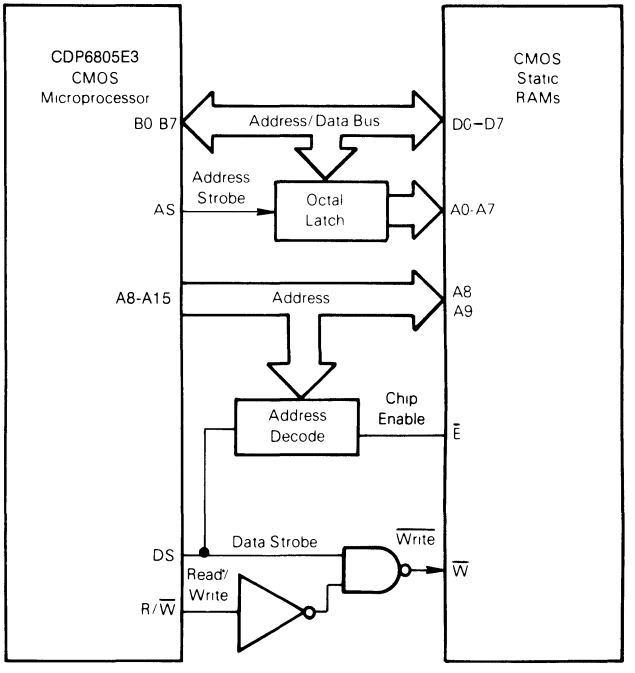
NOTE In some cases, pullup resistors or other level shifting techniques may be required on signals going from NMOS to CMOS parts

Fig. 22 - Connection to peripherals.



92CS-38365

Fig. 23 - Connection to latch non-multiplexed CMOS ROM or EPROM.



92CS-38367

Fig. 24 - Connection to static CMOS RAMs.

CDP6805E3, CDP6805E3C

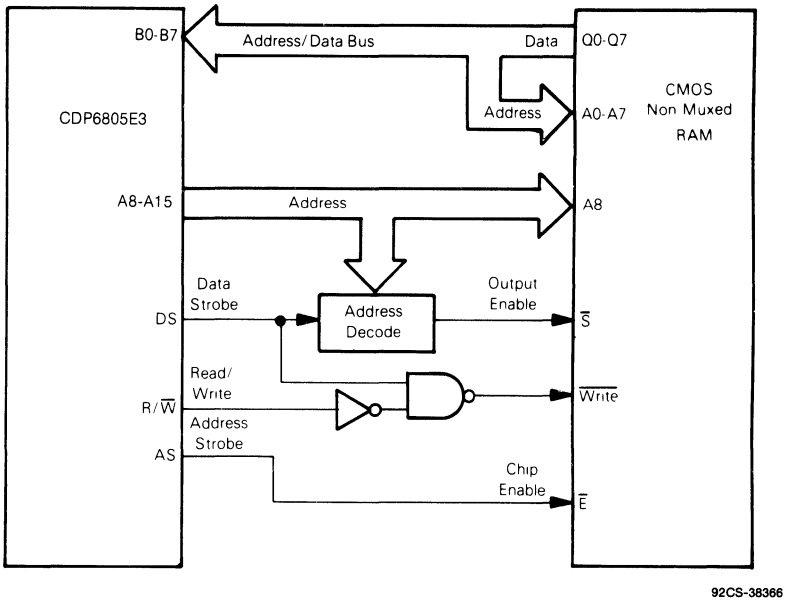


Fig. 25 - Connection to latched non-multiplexed CMOS RAM.

Table 11 provides a detailed description of the information present on the Bus, the Read/Write (R/$\overline{W}$) pin and the Load Instruction (LI) pin during each cycle for each instruction.

This information is useful in comparing actual with expected results during debug of both software and hardware as the control program is executed. The information is categorized in groups according to addressing mode and number of cycles per instruction.

TABLE 11 — SUMMARY OF CYCLE BY CYCLE OPERATION

Address Mode / Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Pin	LI Pin	Data Bus
Inherent						
LSR LSL ASR NEG CLR ROL COM ROR DEC INC TST	3	1 2 3	Op Code Address Op Code Address + 1 Op Code Address + 1	1 1 1	1 0 0	Op Code Op Code Next Instruction Op Code Next Instruction
TAX CLC SEC STOP CLI SEI RSP WAIT NOP TXA	2	1 2	Op Code Address Op Code Address + 1	1 1	1 0	Op Code Op Code Next Instruction
RTS	6	1 2 3 4 5 6	Op Code Address Op Code Address + 1 Stack Pointer Stack Pointer + 1 Stack Pointer + 2 New Op Code Address	1 1 1 1 1 1	1 0 0 0 0 0	Op Code Op Code Next Instruction Irrelevant Data Irrelevant Data Irrelevant Data New Op Code
SWI	10	1 2 3 4 5 6 7 8 9 10	Op Code Address Op Code Address + 1 Stack Pointer Stack Pointer − 1 Stack Pointer − 2 Stack Pointer − 3 Stack Pointer − 4 Vector Address FFFC (Hex) Vector Address FFFD (Hex) Interrupt Routine Starting Address	1 1 0 0 0 0 0 1 1 1	1 0 0 0 0 0 0 0 0 0	Op Code Op Code Next Instruction Return Address (LO Byte) Return Address (HI Byte) Contents of Index Register Contents of Accumulator Contents of CC Register Address of Int Routine (HI Byte) Address of Int. Routine (LO Byte) Interrupt Routine First Opcode
RTI	9	1 2 3 4 5 6 7 8 9	Op Code Address Op Code Address + 1 Stack Pointer Stack Pointer + 1 Stack Pointer + 2 Stack Pointer + 3 Stack Pointer + 4 Stack Pointer + 5 New Op Code Address	1 1 1 1 1 1 1 1 1	1 0 0 0 0 0 0 0 0	Op Code Op Code Next Instruction Irrelevant Data Irrelevant Data Irrelevant Data Irrelevant Data Irrelevant Data Irrelevant Data New Op Code
Immediate						
ADC EOR CPX ADD LDA LDX AND ORA BIT SBC CMB SUB	2	1 2	Op Code Address Op Code Address + 1	1 1	1 0	Op Code Operand Data
Bit Set/Clear						
BSET n BCLR n	5	1 2 3 4 5	Op Code Address Op Code Address + 1 Address of Operand Address of Operand Address of Operand	1 1 1 1 0	1 0 0 0 0	Op Code Address of Operand Operand Data Operand Data Manipulated Data
Bit Test and Branch						
BRSET n BRCLR n	5	1 2 3 4 5	Op Code Address Op Code Address + 1 Address of Operand Op Code Address + 2 Op Code Address + 2	1 1 1 1 1	1 0 0 0 0	Op Code Address of Operand Operand Data Branch Offset Branch Offset
Relative						
BCC BHI BNE BEQ BCS BPL BHCC BLS BIL BMC BRN BHCS BIH BMI BMS BRA	3	1 2 3	Op Code Address Op Code Address + 1 Op Code Address + 1	1 1 1	1 0 0	Op Code Branch Offset Branch Offset
BSR	6	1 2 3 4 5 6	Op Code Address Op Code Address + 1 Op Code Address + 1 Subroutine Starting Address Stack Pointer Stack Pointer − 1	1 1 1 1 0 0	1 0 0 0 0 0	Op Code Branch Offset Branch Offset First Subroutine Op Code Return Address (LO Byte) Return Address (HI Byte)

CDP6805E3, CDP6805E3C

TABLE 11 — SUMMARY OF CYCLE BY CYCLE OPERATION (CONTINUED)

Address Mode / Instructions	Cycles	Cycles #	Address Bus	R/$\overline{W}$ Pin	LI Pin	Data Bus
Direct						
JMP	2	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Jump Address
ADC EOR CPX ADD LDA LDX AND ORA BIT SBC CMP SUB	3	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Address of Operand
		3	Address of Operand	1	0	Operand Data
TST	4	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Address of Operand
		3	Address of Operand	1	0	Operand Data
		4	Op Code Address + 2	1	0	Op Code Next Instruction
STA STX	4	1	Op Code Address	1	1	Op Code
		2	Op Code Adrress + 1	1	0	Address of Operand
		3	Op Code Address + 1	1	0	Address of Operand
		4	Address of Operand	0	0	Operand Data
LSL LSR DEC ASR NEG INC CLR ROL COM ROR	5	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Address of Operand
		3	Operand Address	1	0	Current Operand Data
		4	Operand Address	1	0	Current Operand Data
		5	Operand Address	0	0	New Operand Data
JSR	5	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Subroutine Address (LO Byte)
		3	Subroutine Starting Address	1	0	1st Subroutine Op Code
		4	Stack Pointer	0	0	Return Address (LO Byte)
		5	Stack Pointer − 1	0	0	Return Address (HI Byte)
Extended						
JMP	3	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Jump Address (HI Byte)
		3	Op Code Address + 2	1	0	Jump Address (LO Byte)
ADC BIT ORA ADD CMP LDX AND EOR SBC CPX LDA SUB	4	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Address Operand (HI Byte)
		3	Op Code Address + 2	1	0	Address Operand (LO Byte)
		4	Address of Operand	1	0	Operand Data
STA STX	5	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Address of Operand (HI Byte)
		3	Op Code Address + 2	1	0	Address of Operand (LO Byte)
		4	Op Code Address + 2	1	0	Address of Operand (LO Byte)
		5	Address of Operand	0	0	Operand Data
JSR	6	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Address of Subroutine (HI Byte)
		3	Op Code Address + 2	1	0	Address of Subroutine (LO Byte)
		4	Subroutine Starting Address	1	0	1st Subroutine Op Code
		5	Stack Pointer	0	0	Return Address (LO Byte)
		6	Stack Pointer − 1	0	0	Return Address (HI Byte)
Indexed, No-Offset						
JMP	2	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Op Code Next Instruction
ADC EOR CPX ADD LDA LDX AND ORA BIT SBC CMP SUB	3	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Op Code Next Instruction
		3	Index Register	1	0	Operand Data
TST	4	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Op Code Next Instruction
		3	Index Register	1	0	Operand Data
		4	Op Code Address + 1	1	0	Op Code Next Instruction
STA STX	4	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Op Code Next Instruction
		3	Op Code Address + 1	1	0	Op Code Next Instruction
		4	Index Register	0	0	Operand Data
LSL LSR DEC ASR NEG INC CLR ROL COM ROR	5	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Op Code Next Instruction
		3	Index Register	1	0	Current Operand Data
		4	Index Register	1	0	Current Operand Data
		5	Index Register	0	0	New Operand Data
JSR	5	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Op Code Next Instruction
		3	Index Register	1	0	1st Subroutine Op Code
		4	Stack Pointer	0	0	Return Address (LO Byte)
		5	Stack Pointer − 1	0	0	Return Address (HI Byte)

TABLE 11 – SUMMARY OF CYCLE BY CYCLE OPERATION (CONTINUED)

Address Mode / Instructions	Cycles	Cycles #	Address Bus	R/$\overline{W}$ Pin	LI Pin	Data Bus
Indexed 8-Bit Offset						
JMP	3	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Offset
		3	Op Code Address + 1	1	0	Offset
ADC EOR CPX ADD LDA LDX AND ORA CMP SUB BIT SBC	4	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Offset
		3	Op Code Address + 1	1	0	Offset
		4	Index Register + Offset	1	0	Operand Data
STA STX	5	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Offset
		3	Op Code Address + 1	1	0	Offset
		4	Op Code Address + 1	1	0	Offset
		5	Index Register + Offset	0	0	Operand Data
TST	5	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Offset
		3	Op Code Address + 1	1	0	Offset
		4	Index Register + Offset	1	0	Operand Data
		5	Op Code Address + 2	1	0	Op Code Next Instruction
LSL LSR ASR NEG CLR ROL COM ROR DEC INC	6	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Offset
		3	Op Code Address + 1	1	0	Offset
		4	Index Register + Offset	1	0	Current Operand Data
		5	Index Register + Offset	1	0	Current Operand Data
		6	Index Register + Offset	0	0	New Operand Data
JSR	6	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Offset
		3	Op Code Address + 1	1	0	Offset
		4	Index Register + Offset	1	0	1st Subroutine Op Code
		5	Stack Pointer	0	0	Return Address LO Byte
		6	Stack Pointer – 1	0	0	Return Address HI Byte
Indexed, 16-Bit Offset						
JMP	4	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Offset (HI Byte)
		3	Op Code Address + 2	1	0	Offset (LO Byte)
		4	Op Code Address + 2	1	0	Offset (LO Byte)
ADC CMP SUB ADD EOR SBC AND ORA CPX LDA BIT LDX	5	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Offset (HI Byte)
		3	Op Code Address + 2	1	0	Offset (LO Byte)
		4	Op Code Address + 2	1	0	Offset (LO Byte)
		5	Index Register + Offset	1	0	Operand Data
STA STX	6	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Offset (HI Byte)
		3	Op Code Address + 2	1	0	Offset (LO Byte)
		4	Op Code Address + 2	1	0	Offset (LO Byte)
		5	Op Code Address + 2	1	0	Offset (LO Byte)
		6	Index Register + Offset	0	0	Operand Data
JSR	7	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Offset (HI Byte)
		3	Op Code Address + 2	1	0	Offset (LO Byte)
		4	Op Code Address + 2	1	0	Offset (LO Byte)
		5	Index Register + Offset	1	0	1st Subroutine Op Code
		6	Stack Pointer	0	0	Return Address (LO Byte)
		7	Stack Pointer – 1	0	0	Return Address (HO Byte)

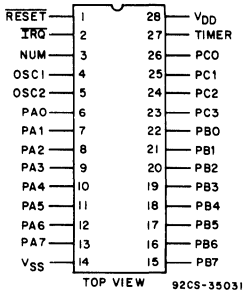
TABLE 11 — SUMMARY OF CYCLE BY CYCLE OPERATION (CONTINUED)

Instructions	Cycles	Cycles #	Address Bus	$\overline{\text{RESET}}$ Pin	R/$\overline{\text{W}}$ Pin	LI Pin	Data Bus
Other Functions							
Hardware $\overline{\text{RESET}}$	5		$FFFE	0	1	0	Irrelevant Data
			$FFFE	0	1	0	Irrelevant Data
		1	$FFFE	1	1	0	Irrelevant Data
		2	$FFFE	1	1	0	Irrelevant Data
		3	$FFFE	1	1	0	Vector High
		4	$FFFF	1	1	0	Vector Low
		5	Reset Vector	1	1	0	Op Code
Power on Reset	1922	1	$FFFF	1	1	0	Irrelevant Data
		•	•	•	•	•	•
		•	•	•	•	•	•
		•	•	•	•	•	•
		1919	$FFFE	1	1	0	Irrelevant Data
		1920	$FFFE	1	1	0	Vector High
		1921	$FFFF	1	1	0	Vector Low
		1922	Reset Vector	1	1	0	Op Code
Instruction	**Cycles**	**Cycles #**	**Address Bus**	$\overline{\text{IRQ}}$ Pin	R/$\overline{\text{W}}$ Pin	LI Pin	**Data Bus**
$\overline{\text{IRQ}}$ Interrupt	10		Last Cycle of Previous Instruction	0	X	0	X
		1	Next Op Code Address	0	1	0	Irrelevant Data
		2	Next Op Code Address	X	1	0	Irrelevant Data
		3	SP	X	0	0	Return Address (LO Byte)
		4	SP – 1	X	0	0	Return Address (HI Byte)
		5	SP – 2	X	0	0	Contents Index Reg
		6	SP – 3	X	0	0	Contents Accumulator
		7	SP – 4	X	0	0	Contents CC Register
		8	See Note Below	X	1	0	Vector High
		9	See Note Below	X	1	0	Vector Low
		10	$\overline{\text{IRQ}}$ Vector	X	1	0	Int Routine First

NOTE: Interrupt Cycles.

	Ext. Int Address	Timer Int Address	Timer Int From Wait Address
Cycle #8	$FFFA	$FFF8	$FFF6
Cycle #9	$FFFB	$FFF9	$FFF7

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*

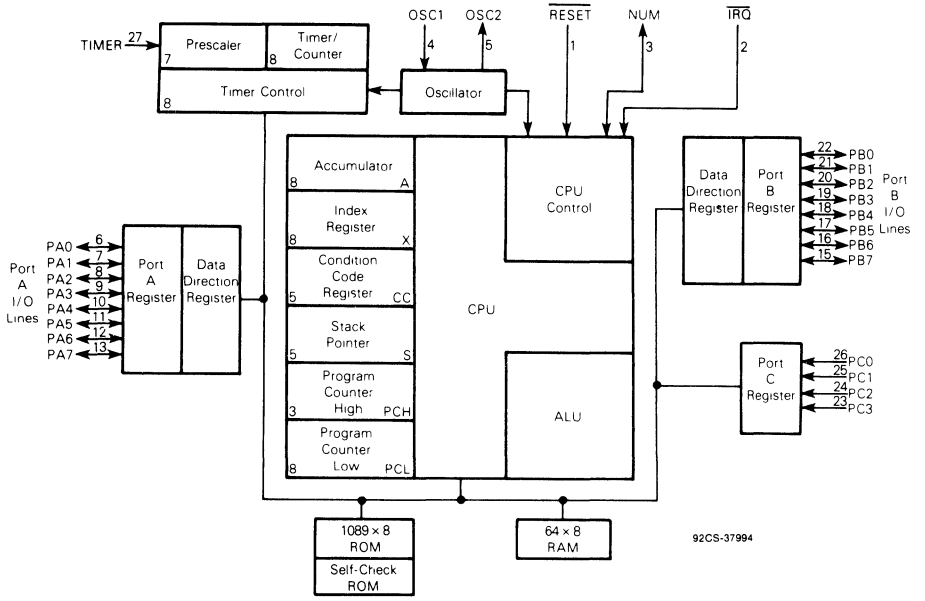


Fig. 1 - CDP6805F2 CMOS microcomputer block diagram

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

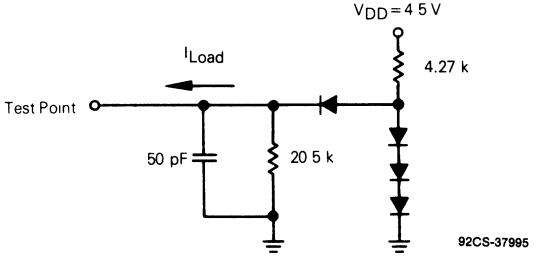


Fig. 2 - Equivalent test load.

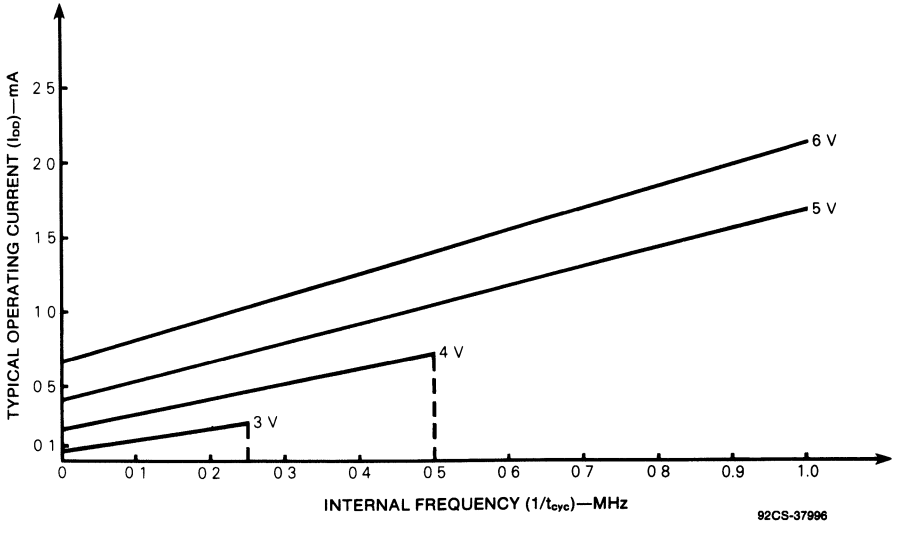


Fig. 3 - Typical operating current vs. internal frequency.

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

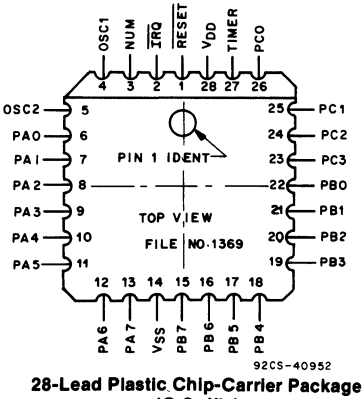
2 Test Conditions for I_{DD} are as follows
- All ports programmed as inputs
- V_{IL} = 0.2 V (PA0-PA7, PB0-PB7, PC0-PC3)
- $V_{IH} = V_{DD}-0.2$ V for $\overline{RESET}$, $\overline{IRQ}$, TIMER
- OSC1 input is a square wave from 0.2 V to $V_{DD}-0.2$ V
- OSC2 output load = 20 pF (WAIT I_{DD} is affected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

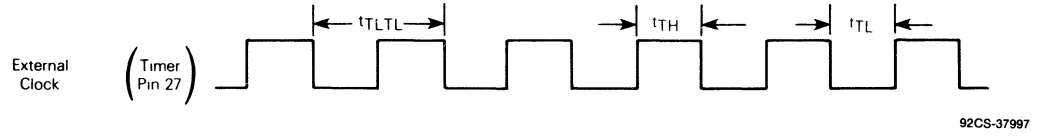


Fig. 4 - Timer relationships.

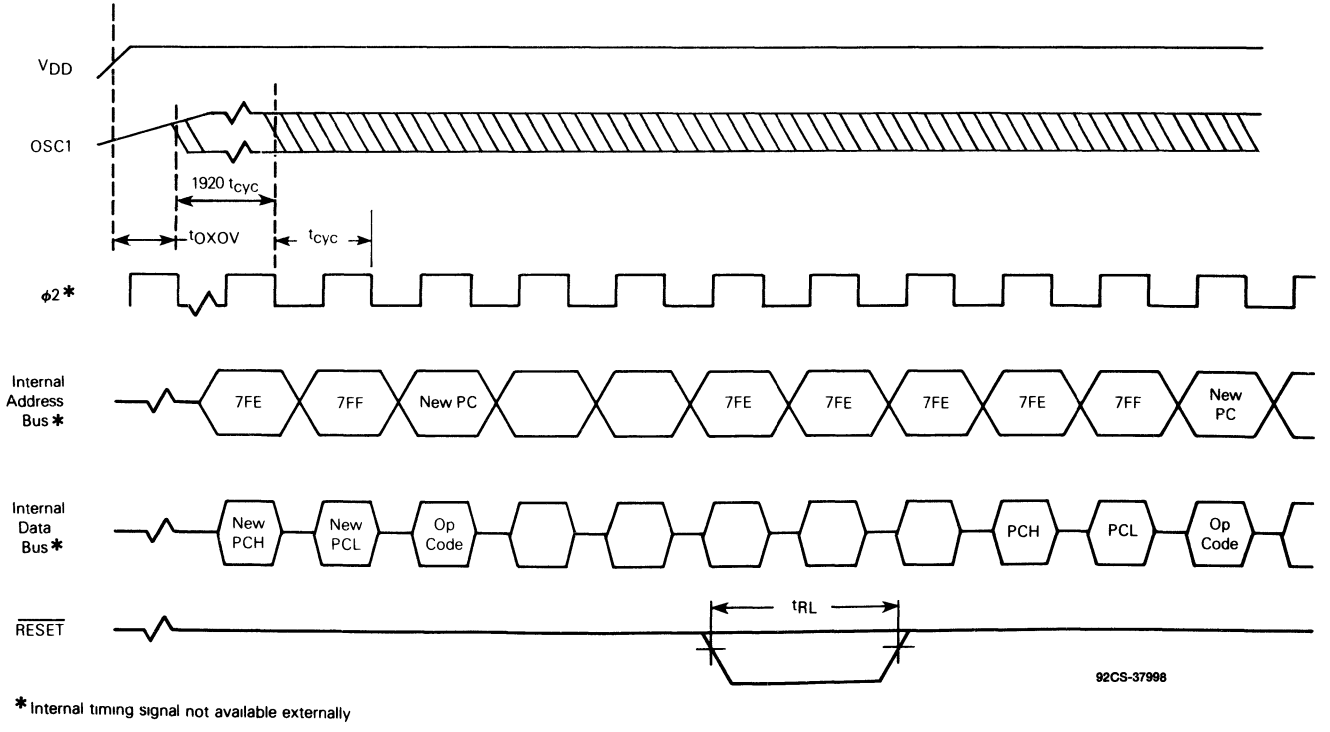
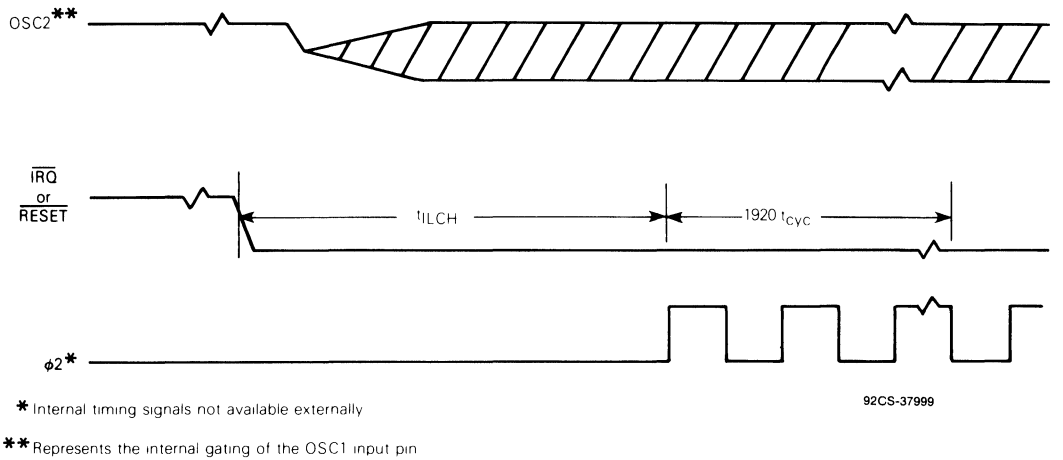


Fig. 5 - Power-on RESET and RESET.



* 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

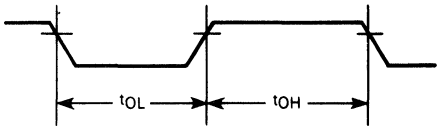
These eight I/O lines comprise Port A The state of any pin is software programmable Refer to the Input/Output Programming section for a detailed description

CDP6805F2, CDP6805F2C

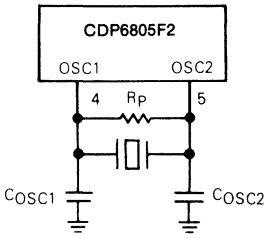
Crystal Parameters

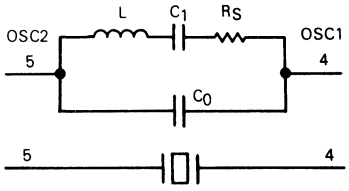
	1 MHz	4 MHz	Units
R_{SMAX}	400	75	Ω
C_0	5	7	pF
C_1	0 008	0 012	μF
C_{OSC1}	15-40	15-30	pF
C_{OSC2}	15-30	15-25	pF
R_P	10	10	MΩ
Q	30 k	40 k	–

Oscillator Waveform

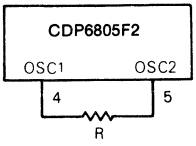


(a) Crystal Oscillator Connections and Equivalent Crystal Circuit





(b) RC Oscillator Connection



(c) External Clock Source Connections

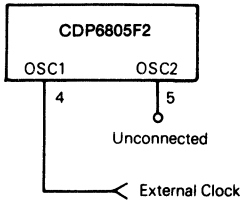
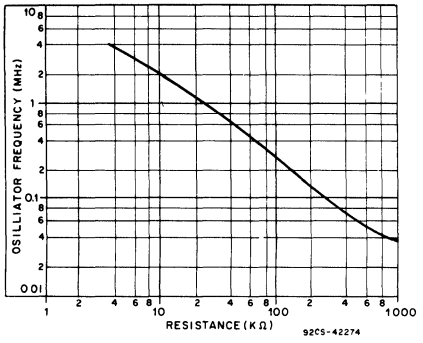


Fig. 7 – Oscillator connections.

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

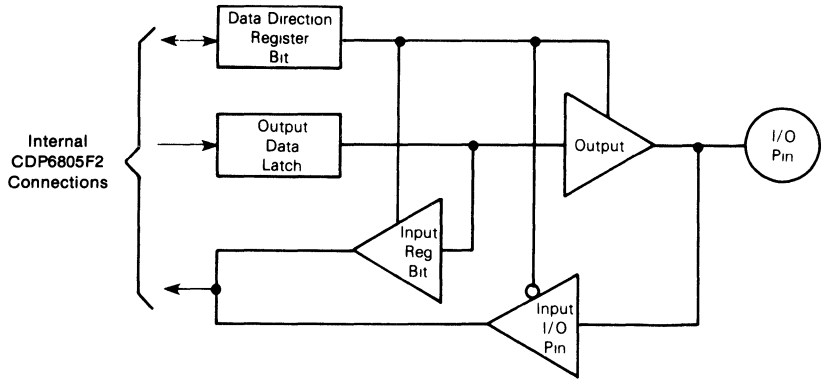
These four lines comprise Port C, a fixed input port When Port C is read, the four most-significant bits on the data bus are "1s" There is no data direction register associated with Port C

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

(a)



(b)

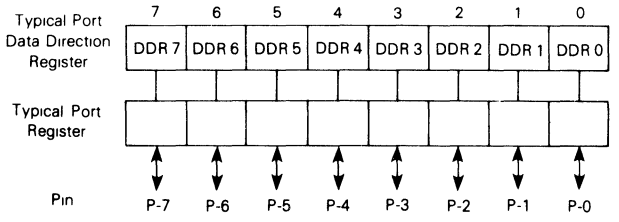


Fig. 9 - Typical I/O port circuitry.

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.)

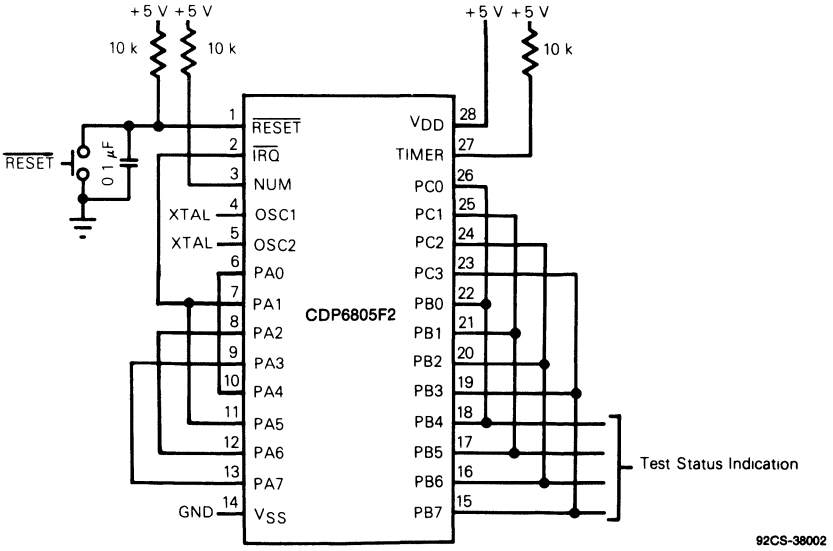


Fig. 10 - Self-check pinout configuration.

CDP6805F2, CDP6805F2C

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

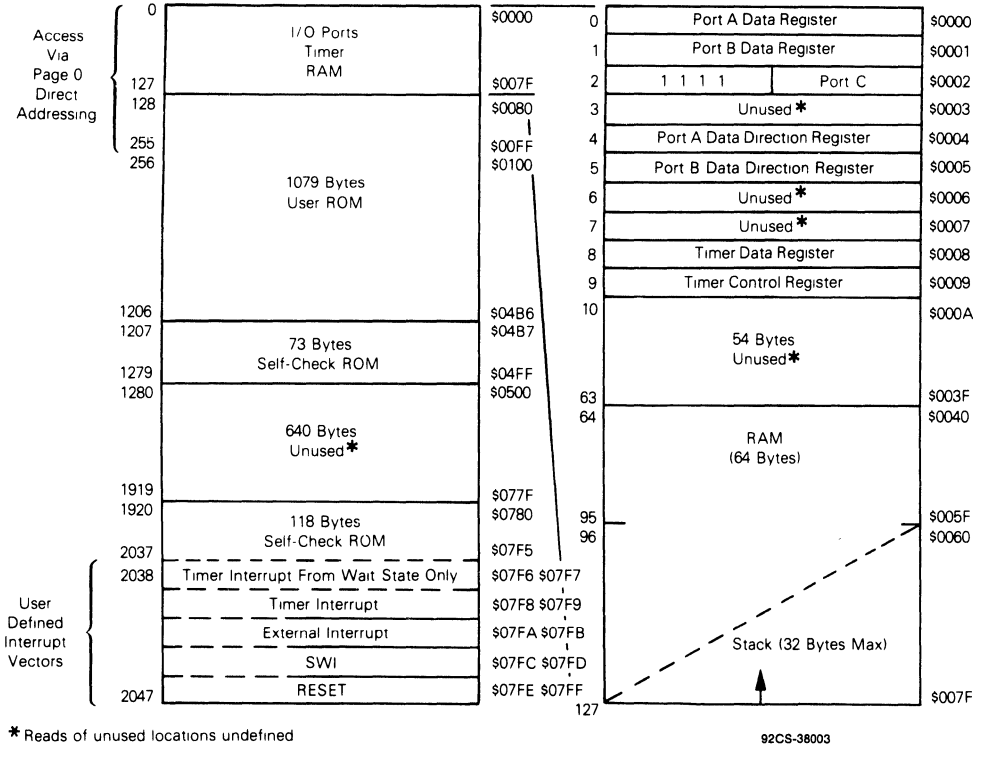


Fig. 11 - Address map.

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.

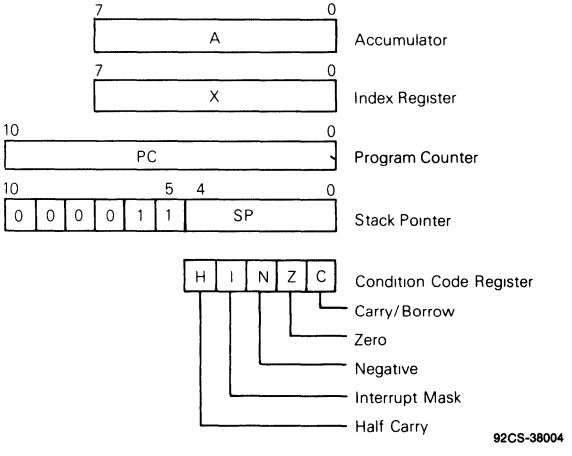
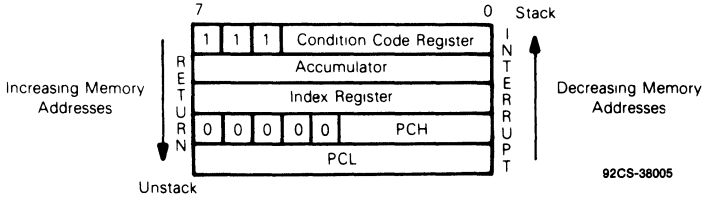


Fig. 12 - Programming model.



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.

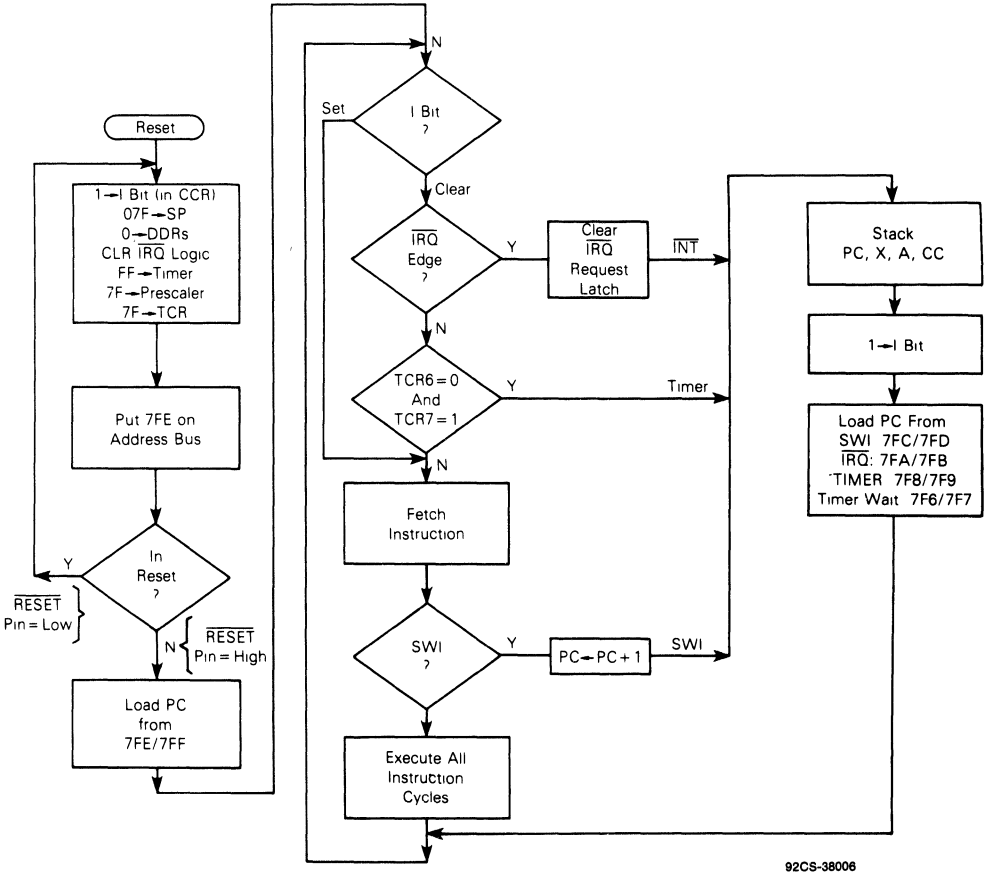


Fig. 14 - RESET and INTERRUPT processing flowchart.

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)

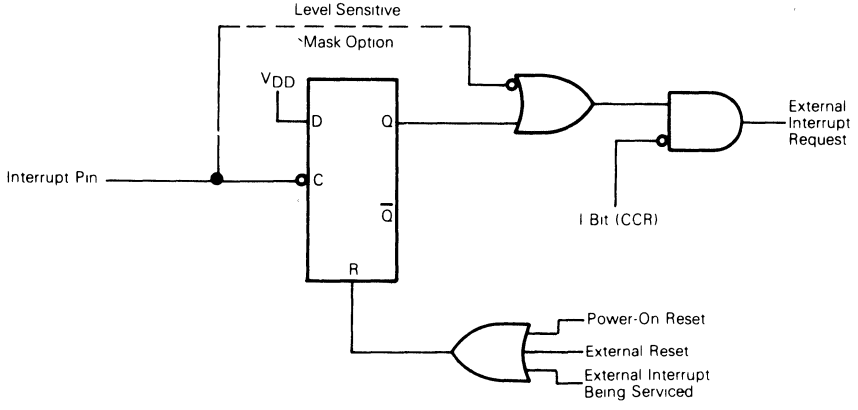
The software interrupt is an executable instruction The action of the SWI instruction is similar to the hardware interrupts. The SWI is executed regardless of the state of the interrupt mask in the condition code register The service routine address is specified by the contents of memory locations $7FC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

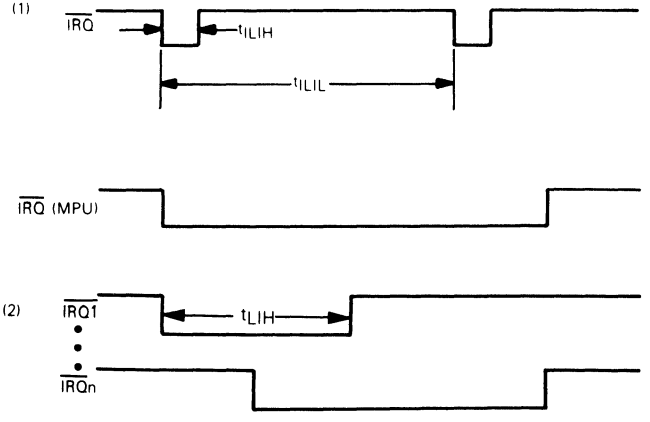
$\overline{RESET}$ – The $\overline{RESET}$ input pin and the internal power-on reset function each cause the program to vector to an initialization program This vector is specified by the contents

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.

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 $\overline{RESET}$.

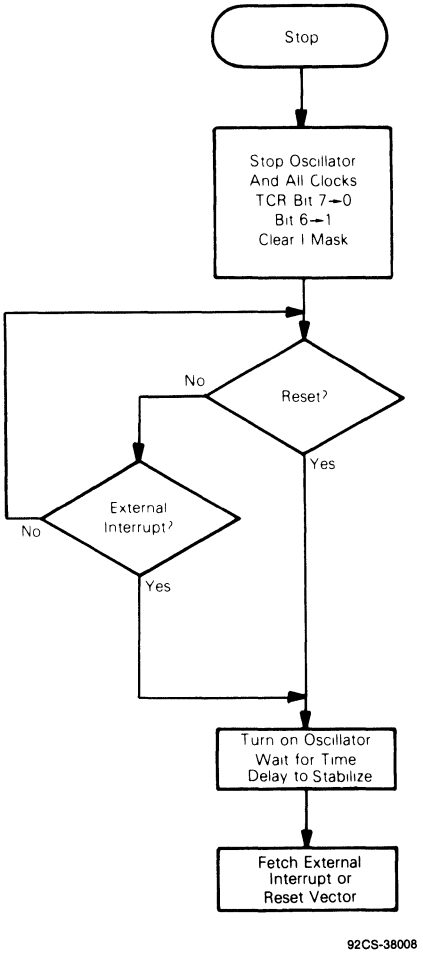


Fig. 16 - Stop function flowchart.

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

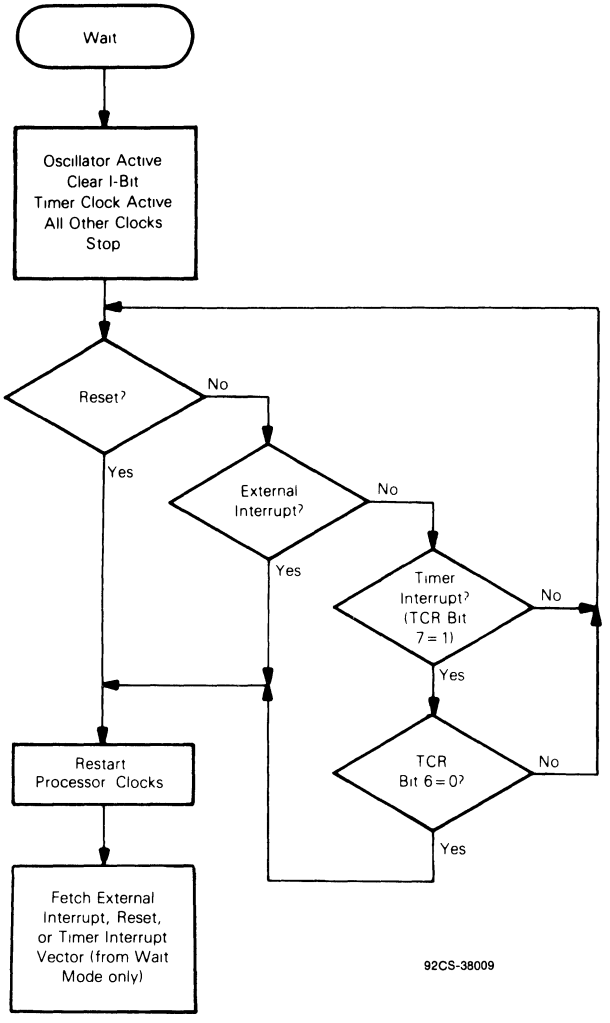


Fig. 17 - WAIT function flowchart.

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

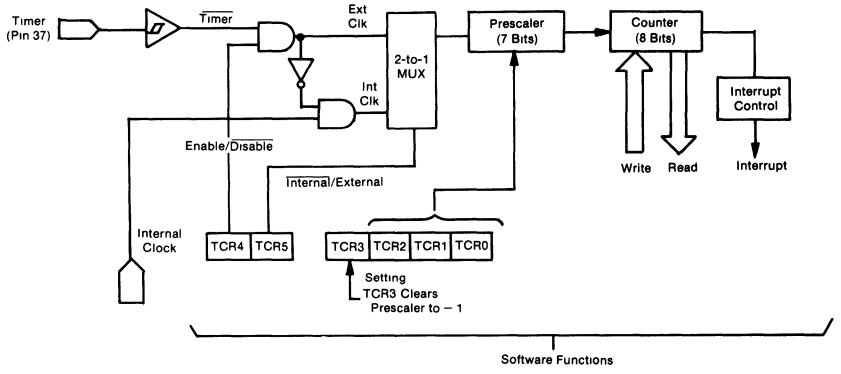
If TCR5=1 and TCR4=0, all inputs to the timer are disabled

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

CDP6805F2, CDP6805F2C



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

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

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

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 instruc ions 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.

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

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.

$$EA = X + [(PC + 1)\ (PC + 2)];\ PC \leftarrow PC + 3$$

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

$$EA = PC + 2 + (PC + 1);\ PC \leftarrow EA \text{ if branch taken;}$$

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.

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

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)

$$EA2 = PC + 3 + (PC + 2);\ PC \leftarrow EA2 \text{ if branch taken;}$$

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

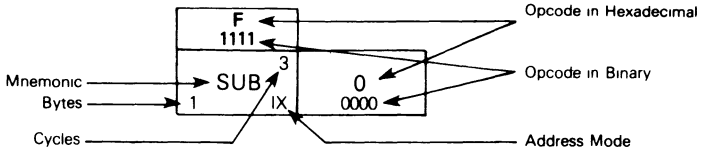
TABLE 9— INSTRUCTION SET OPCODE MAP

	Bit Manipulation		Branch	Read-Modify-Write					Control		Register/Memory						
	BTB	BSC	REL	DIR	INH	INH	IX1	IX	INH	INH	IMM	DIR	EXT	IX2	IX1	IX	
Hi / Low	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111	Hi / Low
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			Λ	●	Λ	Λ	Λ
ADD		X	X	X		X	X	X			Λ	●	Λ	Λ	Λ
AND		X	X	X		X	X	X			●	●	Λ	Λ	●
ASL	X		X			X	X				●	●	Λ	Λ	Λ
ASR	X		X			X	X				●	●	Λ	Λ	Λ
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			●	●	Λ	Λ	●
BLO					X						●	●	●	●	●
BLS					X						●	●	●	●	●
BMC					X						●	●	●	●	●
BMI					X						●	●	●	●	●
BMS					X						●	●	●	●	●
BNE					X						●	●	●	●	●
BPL					X						●	●	●	●	●
BRA					X						●	●	●	●	●
BRN					X						●	●	●	●	●
BRCLR										X	●	●	●	●	Λ
BRSET										X	●	●	●	●	Λ
BSET									X		●	●	●	●	●
BSR					X						●	●	●	●	●
CLC	X										●	●	●	●	0
CLI	X										●	0	●	●	●
CLR	X		X			X	X				●	●	0	1	●
CMP		X	X	X		X	X	X			●	●	Λ	Λ	Λ
COM	X		X			X	X				●	●	Λ	Λ	1
CPX		X	X	X		X	X	X			●	●	Λ	Λ	Λ
DEC	X		X			X	X				●	●	Λ	Λ	●
EOR		X	X	X		X	X	X			●	●	Λ	Λ	●
INC	X		X			X	X				●	●	Λ	Λ	●
JMP			X	X		X	X	X			●	●	●	●	●
JSR			X	X		X	X	X			●	●	●	●	●
LDA		X	X	X		X	X	X			●	●	Λ	Λ	●
LDX		X	X	X		X	X	X			●	●	Λ	Λ	●
LSL	X		X			X	X				●	●	Λ	Λ	Λ
LSR	X		X			X	X				●	●	0	Λ	Λ
NEG	X		X			X	X				●	●	Λ	Λ	Λ
NOP	X										●	●	●	●	●
ORA		X	X	X		X	X	X			●	●	Λ	Λ	●
ROL	X		X			X	X				●	●	Λ	Λ	Λ
ROR	X		X			X	X				●	●	Λ	Λ	Λ
RSP	X										●	●	●	●	●
RTI	X										?	?	?	?	?
RTS	X										●	●	●	●	●
SBC		X	X	X		X	X	X			●	●	Λ	Λ	Λ
SEC	X										●	●	●	●	1
SEI	X										●	1	●	●	●
STA			X	X		X	X	X			●	●	Λ	Λ	●
STOP	X										●	0	●	●	●
STX			X	X		X	X	X			●	●	Λ	Λ	●
SUB		X	X	X		X	X	X			●	●	Λ	Λ	Λ
SWI	X										●	1	●	●	●
TAX	X										●	●	●	●	●
TST	X		X			X	X				●	●	Λ	Λ	●
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
- Λ Test and Set if True Cleared Otherwise
- ● Not Affected
- ? Load CC Register From Stack
- 0 Cleared
- 1 Set