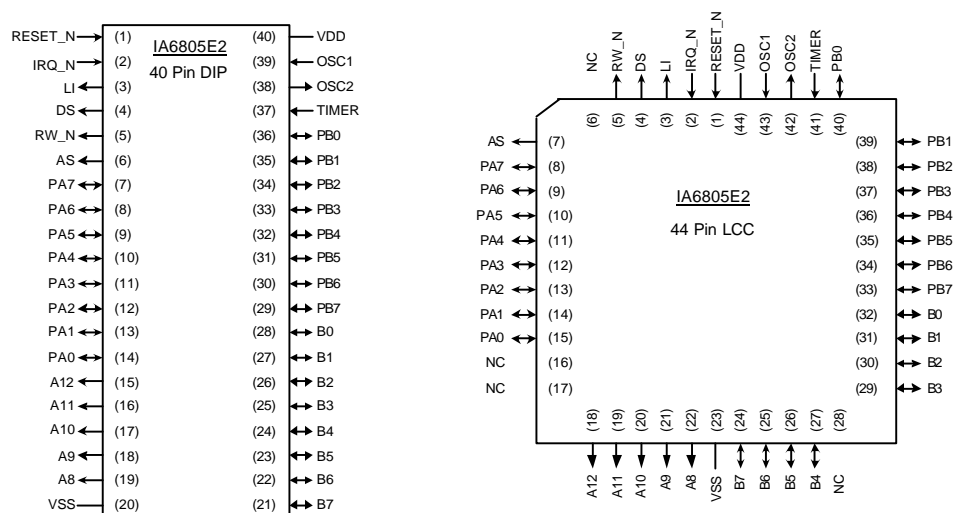


## FEATURES

- Form, Fit, and Function Compatible with the Harris<sup>®</sup> CDP6805E2CE and Motorola<sup>®</sup> MC146805E2
- Internal 8-bit Timer with 7-Bit Programmable Prescaler
- On-chip Clock
- Memory Mapped I/O
- Versatile Interrupt Handling
- True Bit Manipulation
- Bit Test and Branch Instruction
- Vectored Interrupts
- Power-saving STOP and WAIT Modes
- Fully Static Operation
- 112 Bytes of RAM

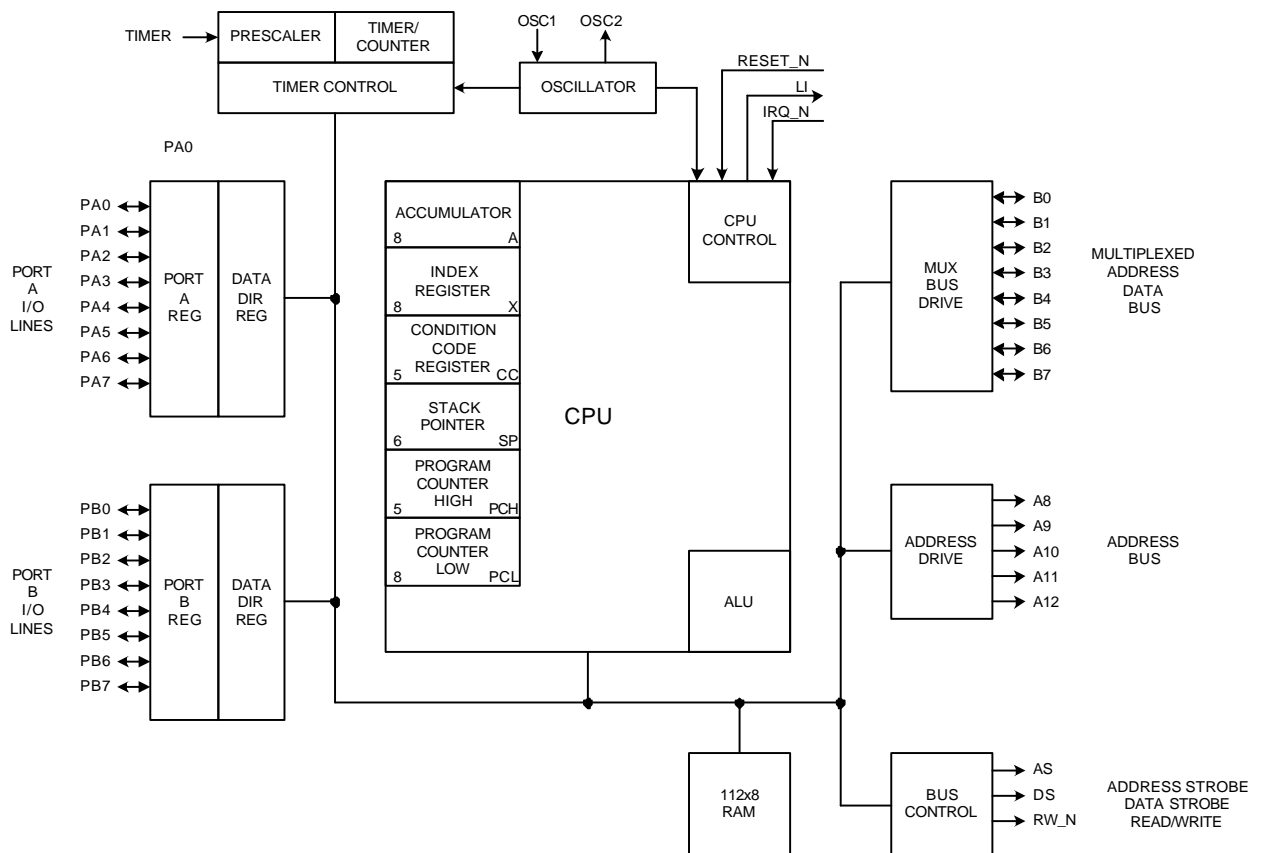
The IA6805E2 is a "plug-and-play" drop-in replacement for the original IC. **innovASIC** produces replacement ICs using its MILES<sup>™</sup>, or Managed IC Lifetime Extension System, cloning technology. This technology produces replacement ICs far more complex than "emulation" while ensuring they are compatible with the original IC. MILES<sup>™</sup> captures the design of a clone so it can be produced even as silicon technology advances. MILES<sup>™</sup> also verifies the clone against the original IC so that even the "undocumented features" are duplicated. This data sheet documents all necessary engineering information about the IA6805E2 including functional and I/O descriptions, electrical characteristics, and applicable timing.

## Package Pinout



**Description**

The IA6805E2 (CMOS) Microprocessor Unit (MPU) is a low cost, low power MPU. It features a CPU, on-chip RAM, parallel I/O compatibility with pins programmable as input or output. The following paragraphs will further describe this system block diagram and design in more detail.



**Figure 1. System Block Diagram**

**I/O Signal Description**

The table below describes the I/O characteristics for each signal on the IC. The signal names correspond to the signal names on the pinout diagrams provided.

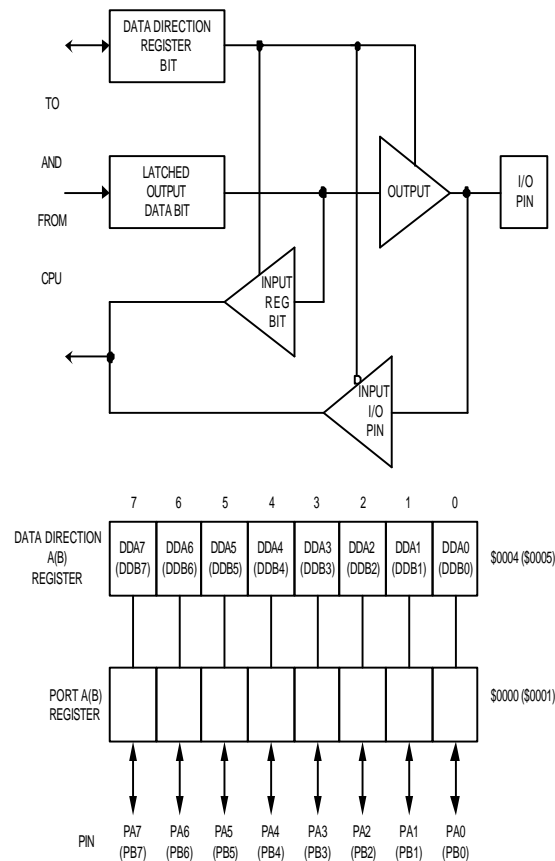
SIGNAL NAME	I/O	DESCRIPTION
V <sub>DD</sub> and V <sub>SS</sub> (Power and Ground)	N/A	Source: These two pins provide power to the chip. V <sub>DD</sub> provides +5 volts (±0.5) power and V <sub>SS</sub> is ground.
RESET <sub>n</sub> (Reset)	I	TTL: Input pin that can be used to reset the MPU's internal state by pulling the reset <sub>n</sub> pin low.
IRQ <sub>n</sub> (Interrupt Request)	I	TTL: Input pin that is level and edge sensitive. Can be used to request an interrupt sequence.
LI (Load Instruction)	O	TTL with slew rate control: Output pin used to indicate that a next opcode fetch is in progress. Used only for certain debugging and test systems. Not connected in normal operation. Overlaps Data Strobe (DS) signal. This output is capable of driving one standard TTL load and 50pF.
DS (Data Strobe)	O	TTL with slew rate control: Output pin used to transfer data to or from a peripheral or memory. DS occurs anytime the MPU does a data read or write and during data transfer to or from internal memory. DS is available at f <sub>OSC</sub> + 5 when the MPU is not in the WAIT or STOP mode. This output is capable of driving one standard TTL load and 130pF.
RW <sub>n</sub> (Read/Write)	O	TTL with slew rate control: Output pin used to indicate the direction of data transfer from internal memory, I/O registers, and external peripheral devices and memories. Indicates to a selected peripheral whether the MPU is to read (RW <sub>n</sub> high) or write (RW <sub>n</sub> low) data on the next data strobe. This output is capable of driving one standard TTL load and 130pF.
AS (Address Strobe)	O	TTL with slew rate control: 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. AS is available at f <sub>OSC</sub> + 5 when the MPU is not in the WAIT or STOP modes. This output is capable of driving one standard TTL load and 130pF.
PA0-PA7/PB0-PB7 (Input/Output Lines)	I/O	TTL with slew rate control: These 16 lines constitute Input/Output ports A and B. Each line is individually programmed to be either an input or output under software control of the Data Direction Register (DDR) as shown below in Table 1 and Figure 2. The port I/O is programmed by writing the corresponding bit in the DDR to a "1" for output and a "0" for input. In the output mode the bits are latched and appear on the corresponding output pins. All the DDR's are initialized to a "0" on reset. The output port registers are not initialized on reset. Each output is capable of driving one standard TTL load and 50pF.
A8-A12 (High Order Address Lines)	O	TTL with slew rate control: These five outputs constitute the higher order non-multiplexed address lines. Each output is capable of driving one standard TTL load and 130pF.
B0-B7 (Address/Data Bus)	I/O	TTL with slew rate control: These bi-directional 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 bi-directional, transferring data to and from memory and peripheral devices as indicated by the RW <sub>n</sub> pin. As outputs, these lines are capable of driving one standard TTL load and 130pF.
Timer	I	TTL: Input used to control the internal timer/counter circuitry.
OSC1, OSC2 (System Clock)	I/O	TTL Oscillator input/output: These pins provide control input for the on-chip clock oscillator circuits. Either a crystal or external clock is connected to these pins to provide a system clock. The crystal connection is shown in Figure 3. The OSC1 to bus transitions for system designs using oscillators slower than 5MHz is shown in Figure 4.
Crystal		The circuit shown in Figure 3 is recommended when using a crystal. An external CMOS oscillator is recommended when using crystals outside the specified ranges. To minimize output distortion and start-up stabilization time, the crystal and components should be mounted as close to the input pins as possible.
External Clock		When an external clock is used, it should be applied to the OSC1 input with the OSC2 input not connected, as shown in Figure 3.

**Table 1**

I/O Pin Functions

R/W-n	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.

I/O Port Circuitry and Register Configuration:

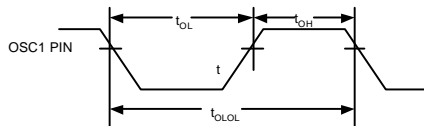
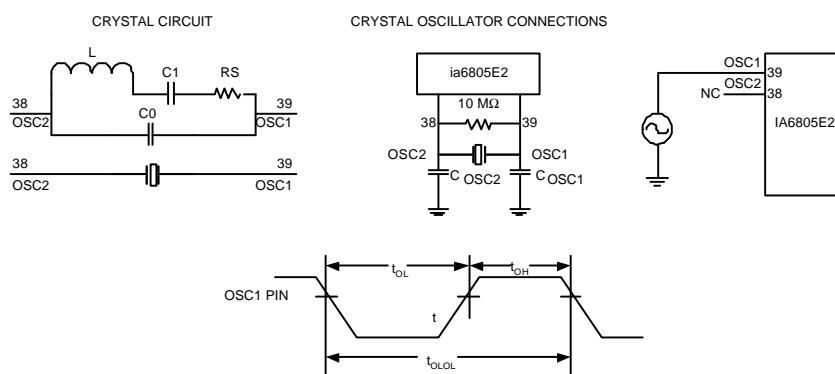


**Figure 2. PA0-PA7/PB0-PB7 (Input/Output Lines)**

Crystal Parameters Representative Frequencies:

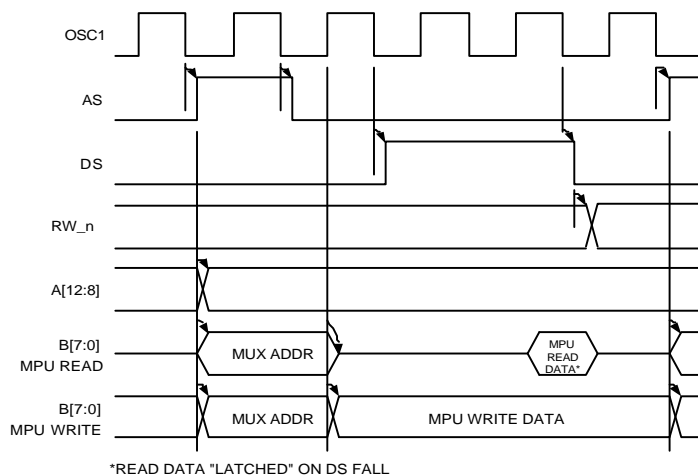
	5.0 MHz	4.0 MHz	1.0 MHz
$R_s$ max	50Ω	75Ω	400Ω
C0	8 pF	7 pF	5 pF
C1	0.02 pF	0.012 pF	0.008 pF
Q	50 k	40 k	30 k
$C_{OSC1}$	15-30 pF	15-30 pF	15-40 pF
$C_{OSC2}$	15-25 pF	15-25 pF	15-30 pF

Oscillator Connections:



**Figure 3. OSC1, OSC2 (System Clock)**

OSC1 to Bus Transitions Timing Waveforms:



\*READ DATA "LATCHED" ON DS FALL

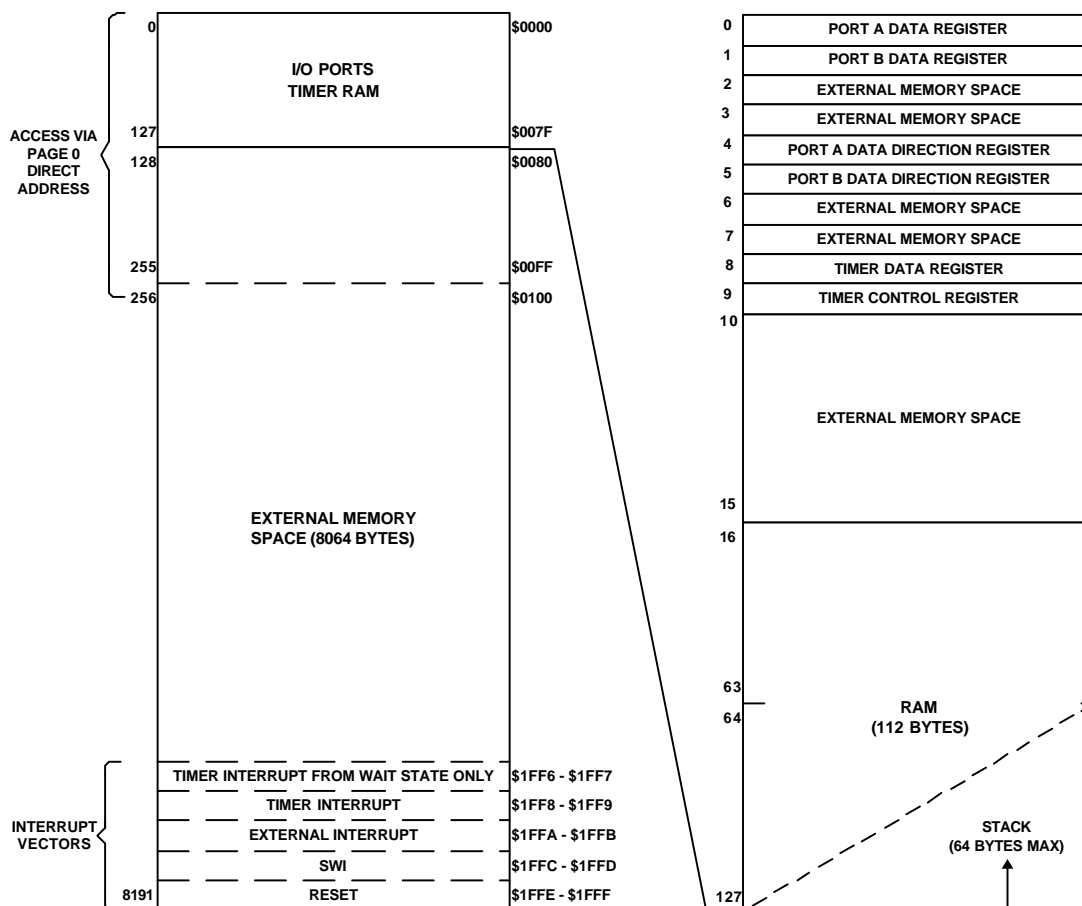
**Figure 4. OSC1, OSC2 (System Clock)**

Functional Description

Memory:

The MPU is capable of addressing 8192 bytes of memory and I/O registers. The locations are divided into internal memory space and external memory space as shown in Figure 5.

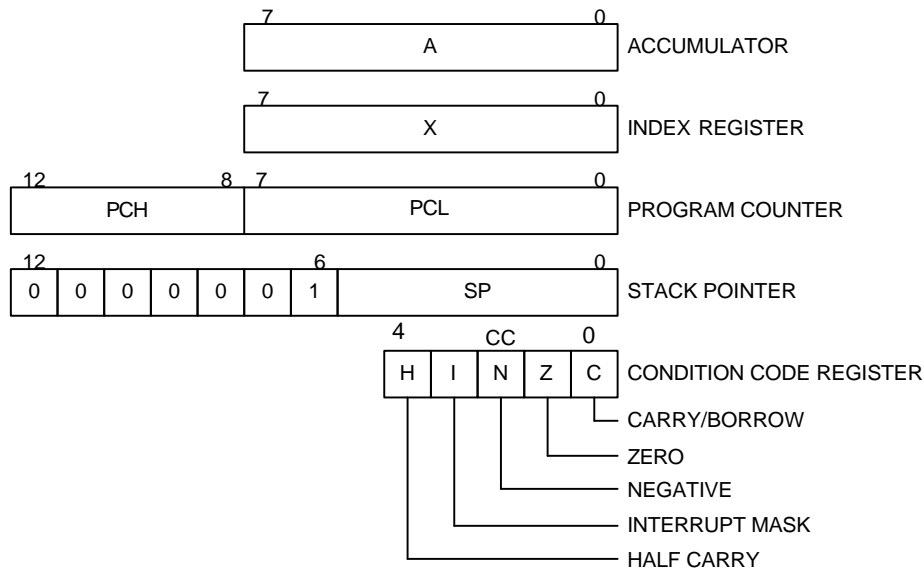
The first 128 bytes of memory contain internal port I/O locations, timer locations, and 112 bytes of RAM. The MPU can read from or write to any of these locations. During program reads from on chip locations, the MPU accepts data only from the addressed on chip location. Any read data appearing on the input bus is ignored. The shared stack area is used during interrupts or subroutine calls. A maximum of 64 bytes of RAM is available for stack usage. The stack pointer is set to \$7f at power up. The unused bytes of the stack can be used for data storage or temporary work locations, but care must be taken to prevent it from being overwritten due to stacking from an interrupt or subroutine call.



**Figure 5. Memory Map**

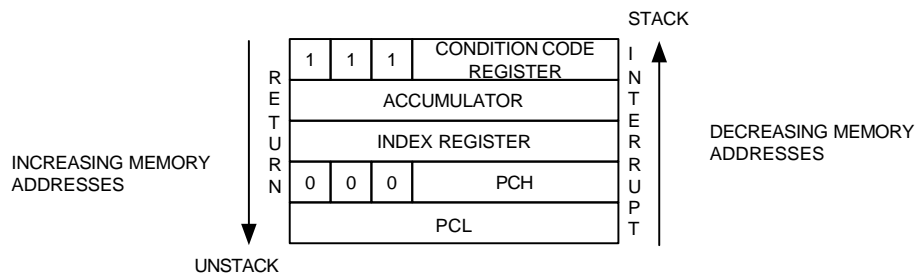
Registers:

The following paragraphs describe the registers contained in the MPU. Figure 6 shows the programming model and Figure 7 shows the interrupt stacking order.



**Figure 6. 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.



**Figure 7. Interrupt Stacking Order**

**A(Accumulator):**

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

**X(Index Register):**

The index register is an 8-bit register used during the indexed addressing mode. It contains an 8-bit value used to create an effective address. The index register may also be used as a temporary storage area when not performing addressing operations.

**PC(Program Counter):**

The program counter is a 13-bit register that holds the address of the next instruction to be performed by the MPU.

**SP(Stack Pointer):**

The stack pointer is a 13-bit register that holds the address of the next free location on the stack. During an MPU reset or the reset stack pointer (RSP) instruction, the stack pointer is set to location \$007f. The seven most significant bits of the stack pointer are permanently set to 0000001. They are appended to the six least significant register bits to produce an address range down to location \$0040. The stack pointer gets decremented as data is pushed onto the stack and incremented as data is removed from the stack. The stack area of RAM is used to store the return address on subroutine calls and the machine state during interrupts. The maximum number of locations for the stack pointer is 64 bytes. If the stack goes beyond this limit the stack pointer wraps around and points to its upper limit thereby losing the previously stored information. Subroutine calls use 2 bytes of RAM on the stack and interrupts use 5 bytes.

**CC(Condition code Register):**

The condition code register is a 5-bit register that indicates the results of the instruction just executed. The bit is set if it is high. A program can individually test these bits and specific actions can be taken as a result of their states. Following is an explanation of each bit.

**C(Carry Bit):**

The carry bit indicates that a carry or borrow out of the Arithmetic Logical Unit (ALU) occurred during the last arithmetic instruction. This bit is also modified during bit test, shift, rotate, and branch types of instructions.

**Z(Zero Bit):**

The zero bit indicates the result of the last arithmetic, logical, or data manipulation was zero.

**N(Negative Bit):**

The negative bit indicates the result to the last arithmetic, logical, or data manipulation was negative (bit 7 in the result is high).



#### I(Interrupt Mask Bit)

The interrupt mask bit indicates that both the external interrupt and the timer interrupt are disabled (masked). If an interrupt occurs while this bit is set, the interrupt is latched and is processed as soon as the interrupt bit is cleared.

#### H(Half Carry Bit)

The half carry bit indicates that a carry occurred between bits 3 and 4 of the ALU during an ADD or ADC operation.

#### Resets:

The MPU can be reset by initial power up or by the external reset pin (reset\_n).

#### POR(Power On Reset)

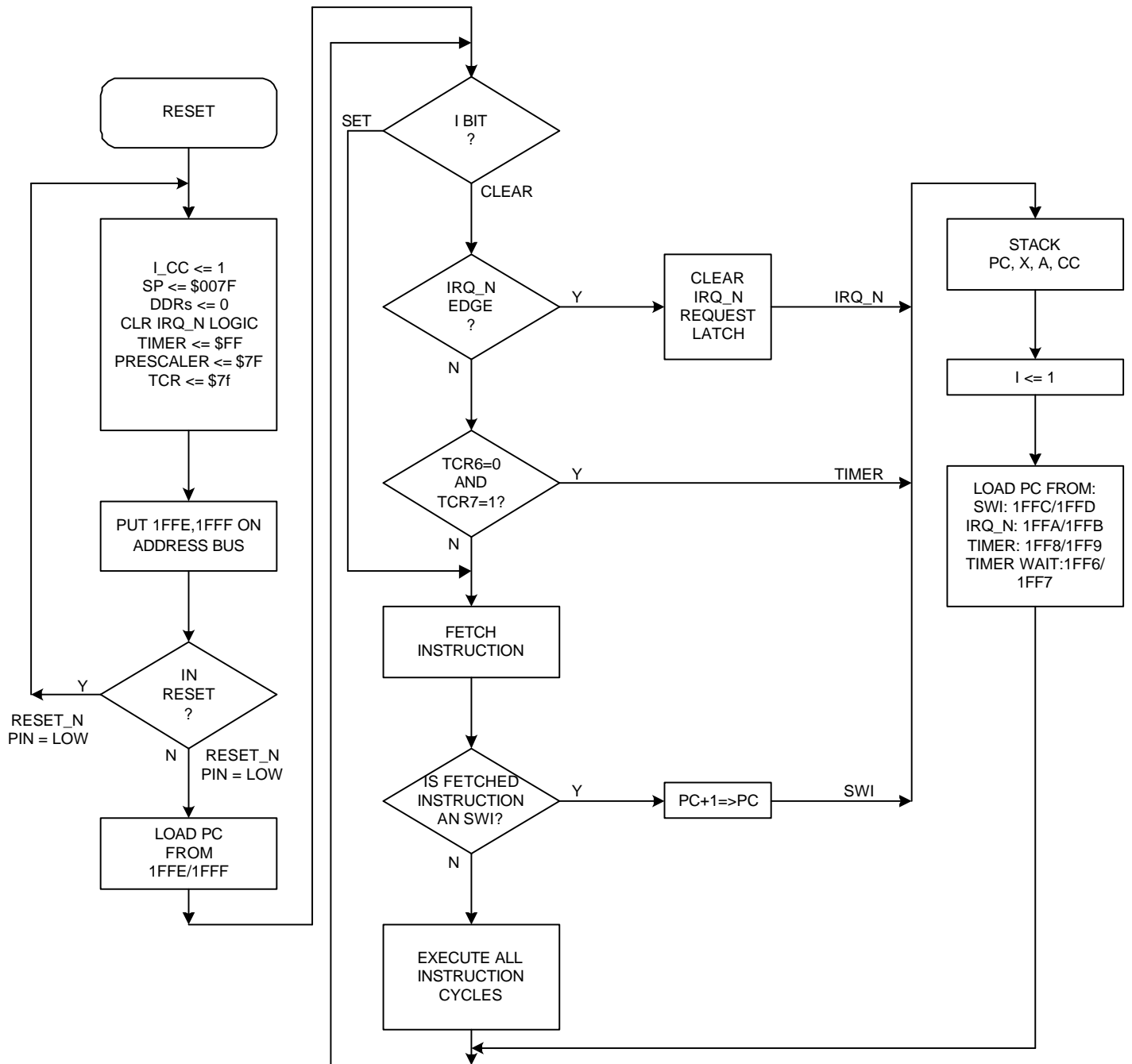
Power on reset occurs on initial power up. It is strictly for power initialization conditions and should not be used to detect drops in the power supply voltage. There is a 1920 t<sub>CYC</sub> time out delay from the time the oscillator is detected. If the reset\_n pin is still low at the end of the delay, the MPU will remain in the reset state until the external pin goes high.

#### Reset\_n

The reset\_n pin is used to reset the MPU. The reset pin must stay low for a minimum of t<sub>CYC</sub> to guarantee a reset. The reset\_n pin is provided with a Schmitt Trigger to improve noise immunity capability.

#### Interrupts:

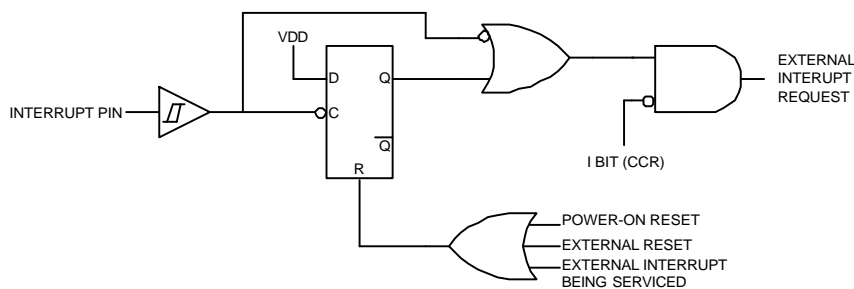
The MPU can be interrupted with the external interrupt pin (irq\_n), the internal timer interrupt request, or the software interrupt instruction. When any of these interrupts occur, normal processing is suspended at the end of the current instruction execution. The processor registers are saved on the stack (stacking order shown in Figure 7) and the interrupt mask (I) is set to prevent additional interrupts. Normal processing resumes after the RTI instruction causes the register contents to be recovered from the stack. When the current instruction is completed, the processor checks all pending hardware interrupts and if unmasked (I bit clear) proceeds with interrupt processing. Otherwise, the next instruction is fetched and executed. Masked interrupts are latched for later interrupt service. External interrupts hold higher priority than timer interrupts. At the end of an instruction execution, if both an external interrupt and timer interrupt are pending, the external interrupt is serviced first. The SWI gets executed with the same priority as any other instruction if the hardware interrupts are masked (I bit set). Figure 8 shows the Reset and Interrupt processing flowchart.



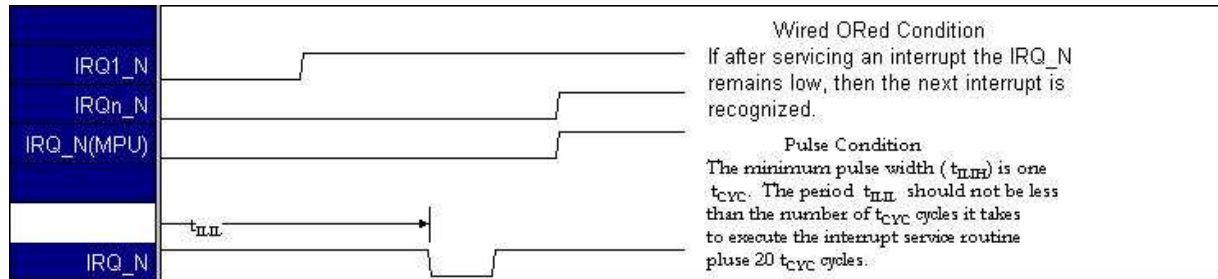
**Figure 8. Reset and Interrupt Processing Flowchart**

**External Interrupt:**

If the external interrupt pin irq\_n is “low” and the interrupt mask bit of the condition code register is cleared, the external interrupt occurs. When the interrupt is recognized, the current state of the machine is pushed onto the stack and the condition code register I-bit gets set masking further interrupts until the present one is serviced. The program counter is then loaded with the contents of the interrupt vector, which contains the location of the interrupt service routine. The contents of \$1FFA and \$1FFB specify the address for this service routine. A functional diagram of the external interrupt is shown in Figure 9 and a mode diagram of the external interrupt is shown in Figure 10. The timing diagram shows two different treatments of the interrupt line (irq\_n) to the processor. The first shows several interrupt lines “wire ORed” to form the interrupts at the processor. If the interrupt line (irq\_n) remains low after servicing an interrupt, the next interrupt is recognized. The second shows 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. After a pulse occurs, the next pulse should not occur until an RTI has occurred. The time between pulses ( $t_{ILIL}$ ) is obtained by adding 20 instruction cycles to the total number of cycles it takes to complete the service routine including the RTI instruction.



**Figure 9. Interrupt Functional Diagram**



**Figure 10. Interrupt Mode Diagram**

**Timer Interrupt:**

If the timer mask bit (TCR6) and the interrupt mask bit (I) of the condition code register are cleared, each time the timer decrements to zero (\$01 to \$00 transition) an interrupt request is generated. When the interrupt is recognized, the current state of the machine is pushed onto the stack and the condition code register Ibit gets set masking further interrupts until the present one is serviced. The program counter is then loaded with the contents of the timer interrupt vector, which contains the location of the timer interrupt service routine. The contents of \$1FF8 and \$1FF9 specify the address for this service routine. If the MPU is in the wait mode and a timer interrupt occurs, then the contents of \$1FF6 and \$1FF7 specify the service routine. When the timer interrupt service routine is complete, the software executes an RTI instruction to restore the machine state and starts executing the interrupt program.

**Software Interrupt:**

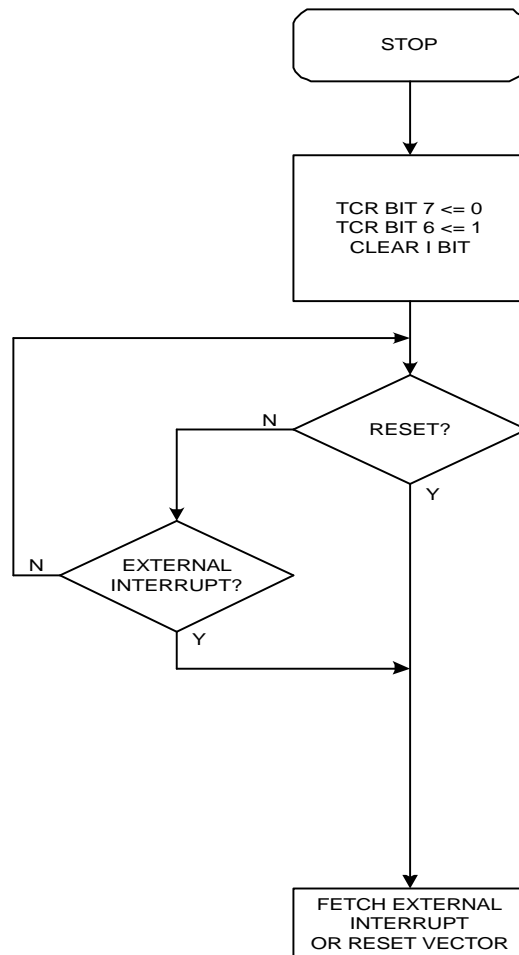
Software interrupt is an executable instruction regardless of the state of the interrupt mask bit (I) in the condition code register. SWI is similar to hardware interrupts. It executes after the other interrupts if the interrupt mask bit is zero. The contents of \$1FFC and \$1FFD specify the address for this service routine.

**Low Power Modes:**

The low power modes consist of the stop instruction and the wait instruction. The following paragraphs explain these modes of operation.

Stop Modes:

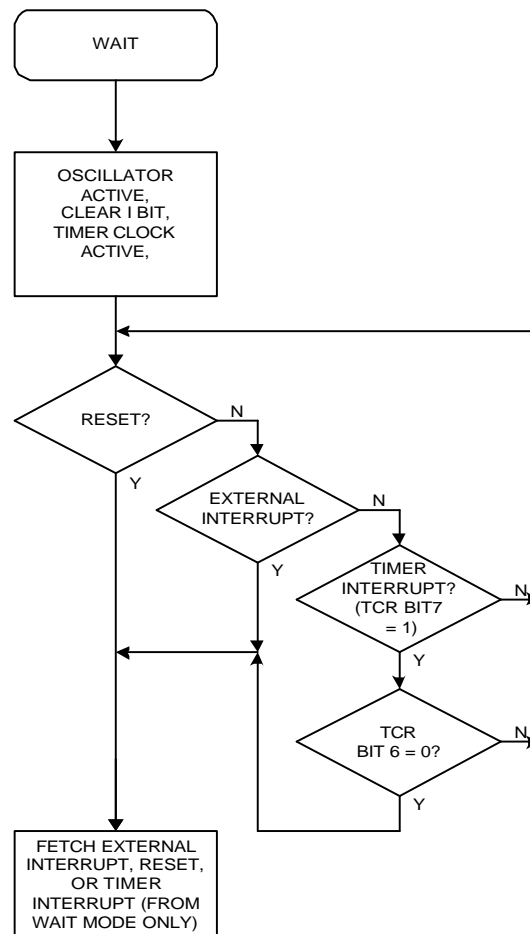
The stop instruction places the MPU in low power consumption mode. The stop instruction disables clocking of most internal registers. Timer control register bits 6 and 7 (TCR6 and TCR7) are altered to remove any pending timer interrupt requests and to disable any further timer interrupts. The DS and AS output lines go “low” and the RW\_n line goes “high”. The multiplexed address/data bus goes to the data input state. The high order address lines remain at the address of the next instruction. External interrupts are enabled by clearing the I bit in the condition code register. All other registers, memory, and I/O remain unaltered. Only an external interrupt or reset will bring the MPU out of the stop mode. Figure 11 shows a flowchart of the stop function.



**Figure 11. STOP Function Flowchart**

**Wait Mode:**

The wait instruction places the MPU in low power consumption mode. The wait instruction disables clocking of most internal registers. The DS and AS output lines go “low” and the RW\_n line goes “high”. The multiplexed address/data bus goes to the data input state. The high order address lines remain at the address of the next instruction. External interrupts are enabled by clearing the I bit in the condition code register. All other registers, memory, and I/O remain unaltered. Only an external interrupt, timer interrupt, or reset will bring the MPU out of the wait mode. 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 MPU is no longer in the wait mode. Figure 12 shows a flowchart of the wait function.

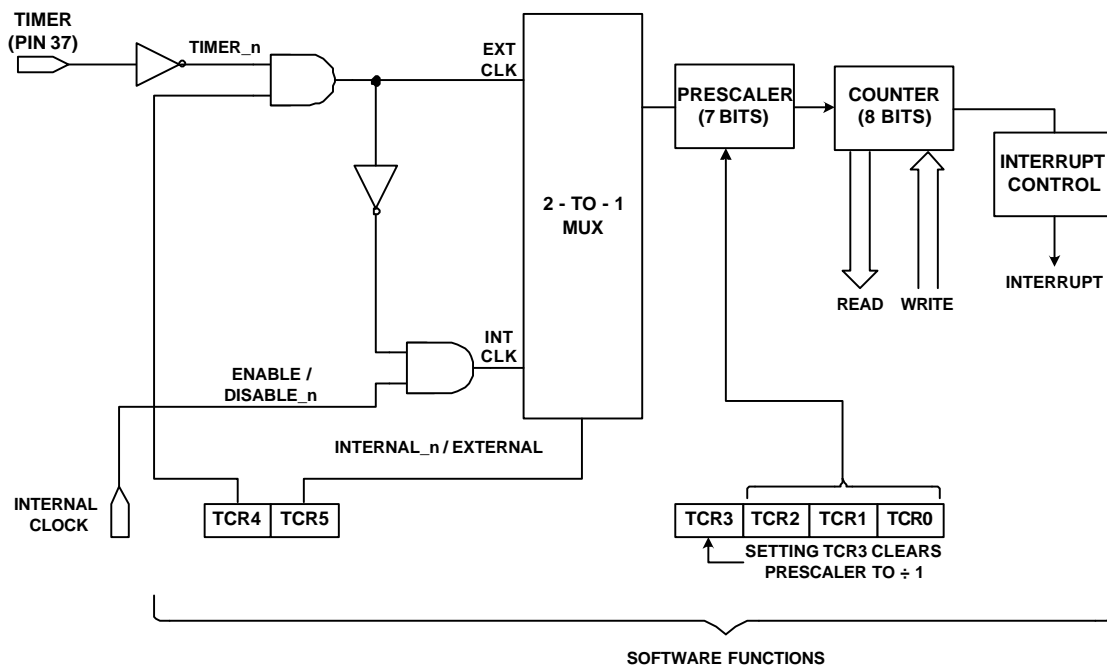


**Figure 12. WAIT Function Flowchart**

Timer:

The MPU contains a single 8-bit software programmable counter driven by a 7-bit software programmable prescaler. The counter may be loaded under program control and decrements to zero. When the counter decrements to zero, the timer interrupt request bit in the timer control register (TCR7) is set. Figure 13 shows a block diagram of the timer. If the timer mask bit (TCR6) and the interrupt mask bit (I) of the condition code register are cleared, an interrupt request is generated. After completion of the current instruction, the current state of the machine is pushed onto the stack. The timer interrupt vector address is then fetched from locations \$1FF8 and \$1FF9 and the interrupt routine is executed, unless the MPU was in the WAIT mode in which case the interrupt vector address in locations \$1FF6 and \$1FF7 is fetched. Power-On-Reset causes the counter to set to \$FF.

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



**Figure 13. Timer Block Diagram**

The counter continues to count past zero, falling from \$00 to \$FF, and continues. The processor may read the counter at any time without disturbing the count by reading the timer data register (TDR). This allows a program to determine the length of time since a timer interrupt has occurred. The timer interrupt request bit remains set until cleared by software. The interrupt is lost if this happens before the timer interrupt is serviced.

The prescaler is a 7-bit divider used to extend the maximum length of the timer. TCR bits 0-2 are programmed to choose the appropriate prescaler output, which is used as the count input. The prescaler is cleared by writing a "1" into TCR bit 3, which avoids truncation errors. The processor cannot write to or read from the prescaler.

#### Timer Input Mode 1:

When  $TCR4 = 0$  and  $TCR5 = 0$ , 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 as a reference for frequency and event measurement. The internal clock is the instruction cycle clock and is coincident with Address Strobe (AS) except during the wait instruction where it goes low. During the wait instruction the internal clock to the timer continues to run at its normal rate.

#### Timer Input Mode 2:

When  $TCR4 = 1$  and  $TCR5 = 0$ , the internal clock and timer input signal are ANDed to form the timer input. This mode can be used to measure external pulse widths. The external pulse turns on the internal clock for the duration of the pulse. The count accuracy in this mode is  $\pm 1$  clock. Accuracy improves with longer input pulse widths.

#### Timer Input Mode 3:

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

#### Timer Input Mode 4:

When  $TCR4 = 1$  and  $TCR5 = 1$ , the internal clock input to the timer is disabled and the timer input then comes from the external TIMER pin. The external clock can be used to count external events as well as to provide an external frequency for generating periodic interrupts.



TCR (Timer Control Register (\$0009)):

An 8-bit register that controls functions such as configuring operation mode, setting ratio of the prescaler, and generating timer interrupt request signals. All bits except bit 3 are read/write. Bits TCR5 - TCR0 are unaffected by reset\_n.

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

Reset:

0	1	0	0	0	0	0	0
---	---	---	---	---	---	---	---

**TCR7 – Timer Interrupt Request**

Used to indicate the timer interrupt when it is logic one.

- 1 – Set when the counter decrements to zero or under program control.
- 0 – Cleared on external reset, POR, STOP instruction, or program control.

**TCR6 – Timer Interrupt Mask**

Used to inhibit the timer interrupt.

- 1 – Interrupt inhibited. Set on external reset, POR, STOP instruction, or program control.
- 0 – Interrupt enabled.

**TCR5 – External or Internal**

Selects input clock source. Unaffected by reset.

- 1 – External clock selected.
- 0 – Internal clock selected (AS) ( $f_{osc}/5$ ).

**TCR4 – Timer External Enable**

Used to enable external timer pin or to enable the internal clock. Unaffected by reset.

- 1 – Enables external timer pin.
- 0 – Disables external timer pin.

TCR3 – Prescaler Clear

Write only bit. Writing a “1” to this bit resets the prescaler to zero. A read of this location always indicates a zero. Unaffected by reset.

TCR2, TCR1, TCR0 – Prescaler select bits

Decoded to select one of eight outputs of the prescaler. Unaffected by reset.

Prescaler

TRC2	TRC1	TRC0	RESET
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 Description**

The MPU has 61 basic instructions divided into 5 types. The 5 types are Register/memory, read-modify-write, branch, bit manipulation, and control.

**Register/Memory Instructions:**

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

Function	Mnemonic
Load A from memory	LDA
Load X from memory	LDX
Store A in memory	STA
Store X in memory	STX
Add memory to A	ADD
Add memory and carry to A	ADC
Subtract memory	SUB
Subtract memory from A with Borrow	SBC
AND memory to A	AND
OR memory with A	ORA
Exclusive OR memory with A	EOR
Arithmetic compare A with memory	CMP
Arithmetic compare X with memory	CPX
Bit test memory with A (logical compare)	BIT
Jump Unconditional	JMP
Jump to subroutine	JSR

**Read-Modify-Write Instructions:**

These instructions read a memory or register location, modify or test its contents and then write the modified value back to memory or the register.

Function	Mnemonic
Increment	INC
Decrement	DEC
Clear	CLR
Complement	COM
Negate (2's complement)	NEG
Rotate Left Thru Carry	ROL
Rotate Right Thru Carry	ROR
Logical shift left	LSL
Logical shift right	LSR
Arithmetic shift right	ASR
Test for negative or zero	TST

Bit Manipulation Instructions:

The MPU is capable of altering any bits residing in the first 256 bytes of memory. An additional feature allows the software to test and branch on the state of any bit within these locations. For test and branch instructions the value of the bit tested is placed in the carry bit of the condition code register.

Function	Mnemonic n = 0...7
Branch if bit n set	BRSET n
Branch if bit n clear	BRCLR n
Set bit n	BSET n
Clear bit n	BCLR n

Branch Instructions:

If a specific condition is met, the instruction branches. If not, no operation is performed.

Function	Mnemonic
Branch always	BRA
Branch never	BRN
Branch if higher	BHI
Branch if lower or same	BLS
Branch if carry clear	BCC
Branch if higher or same	BHS
Branch if carry set	BCS
Branch if lower	BLO
Branch if not equal	BNE
Branch if equal	BEQ
Branch if half carry clear	BHCC
Branch if half carry set	BHCS
Branch if plus	BPL
Branch if minus	BMI
Branch if interrupt mask bit clear	BMC
Branch if interrupt mask bit set	BMS
Branch if interrupt line low	BIL
Branch if interrupt line high	BIH
Branch to subroutine	BSR

Control Instructions:

Used to control processor operation during program execution. They are register reference instructions.

Function	Mnemonic
Transfer A to X	TAX
Transfer X to A	TXA
Set carry bit	SEC
Clear carry bit	CLC
Set interrupt mask bit	SEI
Clear interrupt mask bit	CLI
Software interrupt	SWI
Return from subroutine	RTS
Return from interrupt	RTI
Reset stack pointer	RSP
No-Operation	NOP
Stop	STOP
Wait	WAIT

# IA6805E2 Microprocessor Unit

# Data Sheet As of Production Version 00

## Opcode Map Summary:

The following table is an opcode map for the instructions used on the MPU. The legend following the table shows how to use the table.

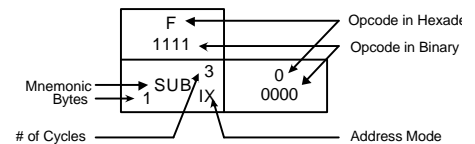
	Bit Manipulation		Branch	Read-Modify-Write				Control		Register/Memory						Hi Low		
	BTB	BSC	REL	DIR	INH	INH	IX1	IX	INH	INH	IMM	DIR	EXT	IX2	IX1		IX	
Hi Low	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111	Hi Low	
0 0000	BRSET0 BTB 3	BSET0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

Abbreviations for Address Modes:

INH Inherent  
A Accumulator  
X Index Register  
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 offset  
IX2 Indexed, 2 byte offset

Legend:



Copyright © 2002

innovASIC

The End of Obsolescence™

ENG21108140100

Page 22 of 31

www.innovasic.com

Customer Support:

1-888-824-4184

**AC/DC Parameters**

Absolute maximum ratings:

Supply Voltage ( $V_{DD}$ ).....-0.3V to 6V  
 Input Pin Voltage ( $V_{IN}$ ).....-0.3 to  $V_{DD}+0.3V$   
 Operating Temperature.....-40°C to 85°C  
 Storage temperature Range ( $T_{stg}$ ).....- 55°C to 150°C  
 ESD Protection (HBM).....5000V

Note: The specifications indicate levels where permanent damage to the device may occur. Functional operation is not guaranteed under these conditions. Operation at absolute maximum conditions for extended periods may adversely affect the long-term reliability of the device.

**DC Characteristics**

( $V_{DD}=4.5$  to  $5.5$  Vdc,  $V_{SS}=0$ ,  $T_A=T_L$  to  $T_H$ ), unless otherwise specified

DC CHARACTERISTICS

Symbol	Parameter	Min	Max	Unit
$V_{DD}$	Supply Voltage	4.5	5.5	V
$V_{OL}$	Output Voltage, $I_{LOAD} \leq 2mA$	-	0.4	V
$V_{OH}$		3.5	-	V
$I_{OL}$	Output Current	-	2	mA
$I_{OH}$		-	-2	mA
$V_{IH}$	High Level input Voltage	2	-	V
$V_{IL}$	Low Level input Voltage	-	0.8	V
$I_{IH}$	High Level input Current	-	1	$\mu A$
$I_{IL}$	Low Level input Current	-	-1	$\mu A$
$V_{t-}$	Schmitt Negative Threshold	1.1	-	V
$V_{t+}$	Schmitt Positive Threshold	-	1.87	V
	Frequency of Operation			
$f_{osc}$	Crystal	-	5	MHz
$f_{osc}$	External Clock	DC	5	MHz

**Control Timing**

$V_{SS}=0V$ ,  $T_A=T_L$  to  $T_H$

Parameters	Sym	$V_{DD} = 5.0V \pm 10\%$ $f_{OSC} = 5MHz$			Unit
		Min	Typ	Max	
I/O Port Timing – Input Setup Time (Figure 14)	$t_{pVASL}$	196	-	-	ns
Input Hold Time (Figure 14)	$t_{ASLPX}$	0	-	-	ns
Output Delay Time (Figure 14)	$t_{ASLPV}$	-	-	0	ns
Interrupt Setup Time (Figure 15)	$T_{ILASL}$	0.4	-	-	$\mu s$
Crystal Oscillator Startup Time (Figure 16)	$t_{OXOV}$	-	5	100	ms
Wait Recovery Startup Time (Figure 17)	$t_{IVASH}$	-	-	2	$\mu s$
Stop Recovery Startup Time (Figure 18)	$t_{ILASH}$	-	-	2	$\mu s$
Required Interrupt Release (Figure 15)	$t_{DSLH}$	-	-	1.0	$\mu s$
Timer Pulse Width (Figure 17)	$t_{TH}, t_{TL}$	0.5	-	-	$t_{CYC}$
Reset Pulse Width (Figure 16)	$t_{RL}$	1.05	-	-	$\mu s$
Timer Period (Figure 17)	$t_{TLTL}$	1.0	-	-	$t_{CYC}$
Interrupt Pulse Width Low (Figure 10)	$t_{LILH}$	1.0	-	-	$t_{CYC}$
Interrupt Pulse Period (Figure 10)	$t_{LILL}$	*	-	-	$t_{CYC}$
Oscillator Cycle Period (1/5 of $t_{CYC}$ ) (Figure 3)	$t_{OLOL}$	200	-	-	ns
OSC1 Pulse Width High (Figure 3)	$t_{OH}$	75	-	-	ns
OSC1 Pulse Width Low (Figure 3)	$t_{OL}$	75	-	-	ns

\*The minimum period of  $t_{LILL}$  should not be less than the number of  $t_{CYC}$  cycles it takes to execute the interrupt service routine plus 20  $t_{CYC}$  cycles.



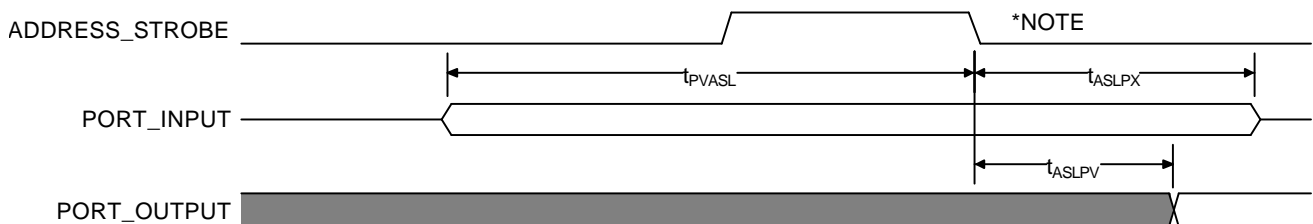
**Bus Timing**

$V_{SS}=0V$ ,  $T_A=T_L$  to  $T_H$  (Figure 19)

Num	Parameters	$V_{DD} = 5.0V \pm 10\%$ $f_{osc} = 5MHz$ 1 TTL, 100pF Load		Unit
		Min	Max	
1	Cycle Time	1000	DC	ns
2	Pulse Width, DS Low	587	-	ns
3	Pulse Width, DS High	403	-	ns
4	Clock Transition	-	4	ns
8	RW_n	9	-	ns
9	Non-Muxed Address Hold	97	-	ns
11	RW_n Delay From DS Fall	-	40	ns
16	Non-Muxed Address Delay From AS Rise	-	11	ns
17	MPU Read Data Setup	18	-	ns
18	Read Data Hold	0	-	ns
19	MPU Data Delay, Write	-	0	ns
21	Write Data Hold	204	-	ns
23	Muxed Address Delay From AS Rise	-	26	ns
24	Muxed Address Valid to AS Fall	185	-	ns
25	Muxed Address Hold	103	-	ns
26	Delay DS Fall to AS Rise	190	-	ns
27	Pulse Width, AS High	203	-	ns
28	Delay, AS Fall to DS Rise	185	-	ns

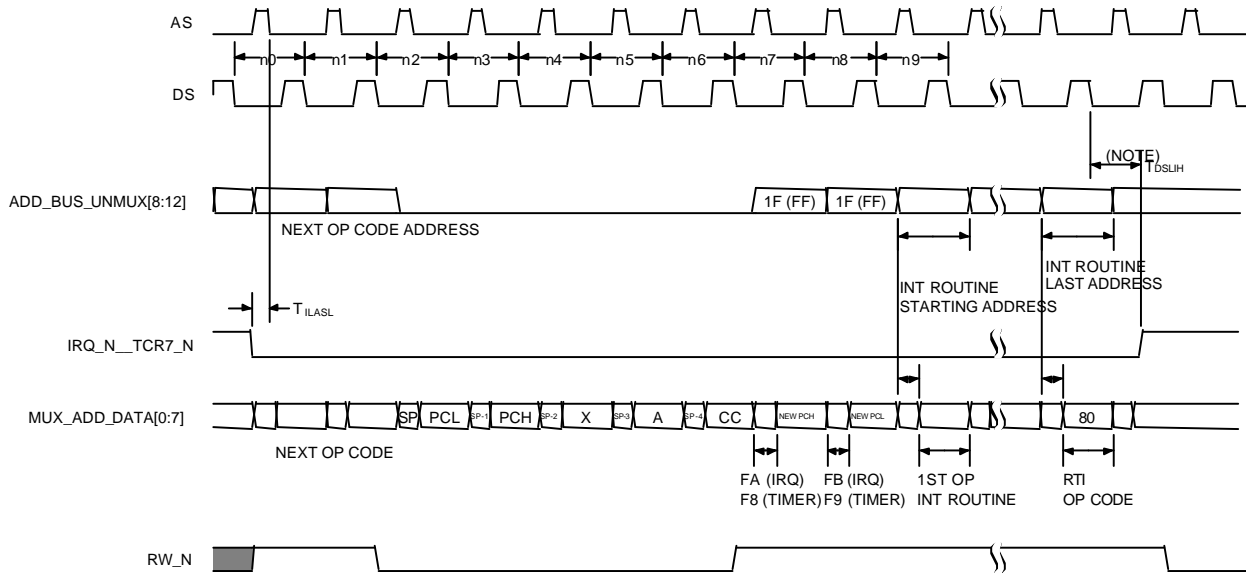
$$V_{LOW} = 0.8V, V_{HIGH} = V_{DD} - 2.0V, V_{DD} = 5.0V \pm 10\%$$

$$T_A = T_L \text{ to } T_H, C_L \text{ on Port} = 50pF, f_{osc} = 5MHz$$



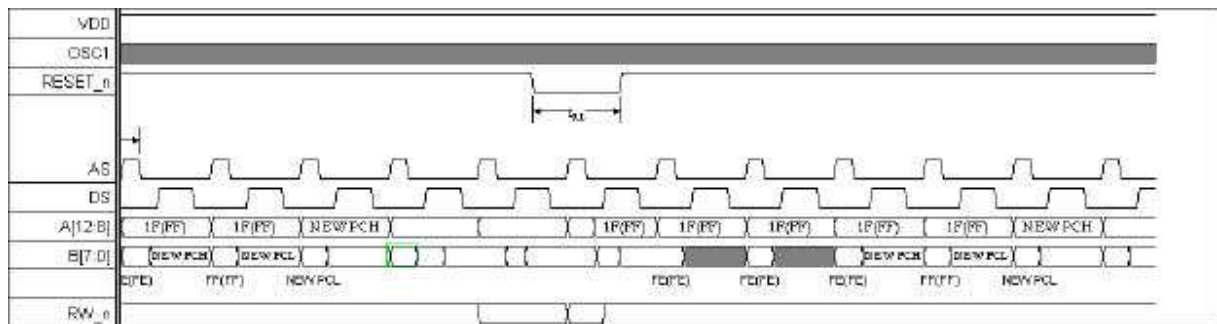
\*Note: The address strobe of the first cycle of the next instruction.

**Figure 14. I/O Port Timing**

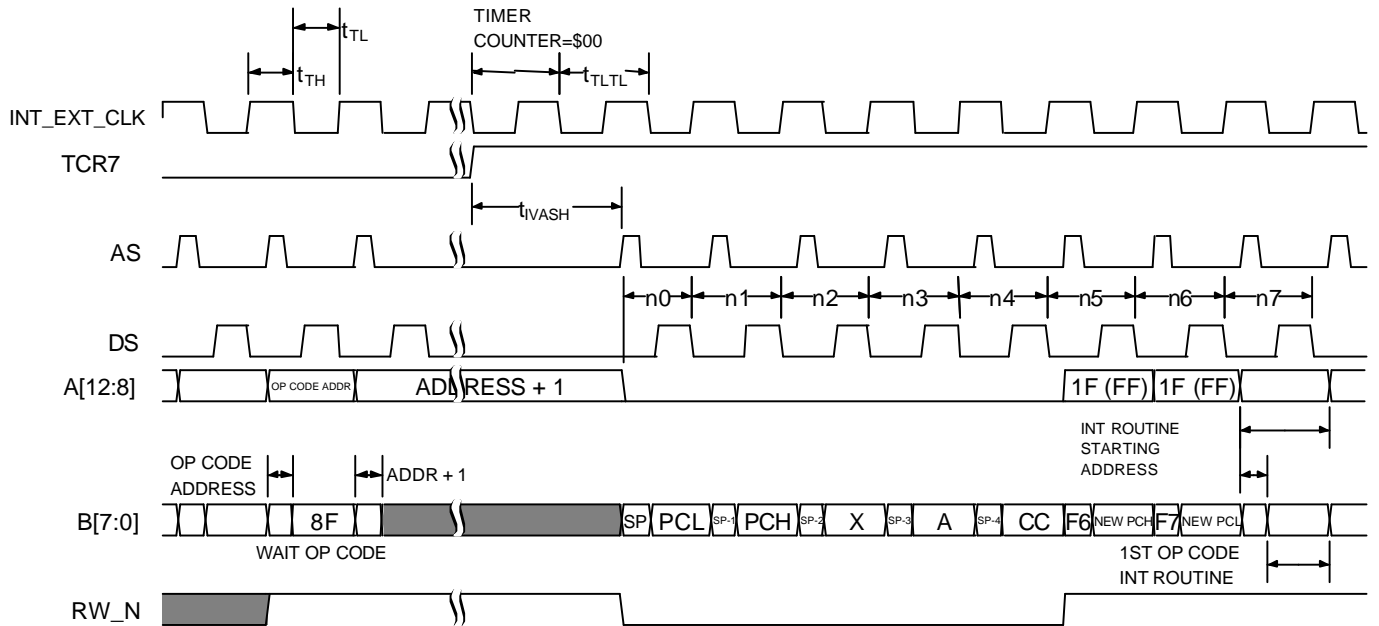


Note:  $t_{DSLH}$ - the interrupting device must release the  $IRQ_N$  line within this time to prevent subsequent recognition of the same interrupt.

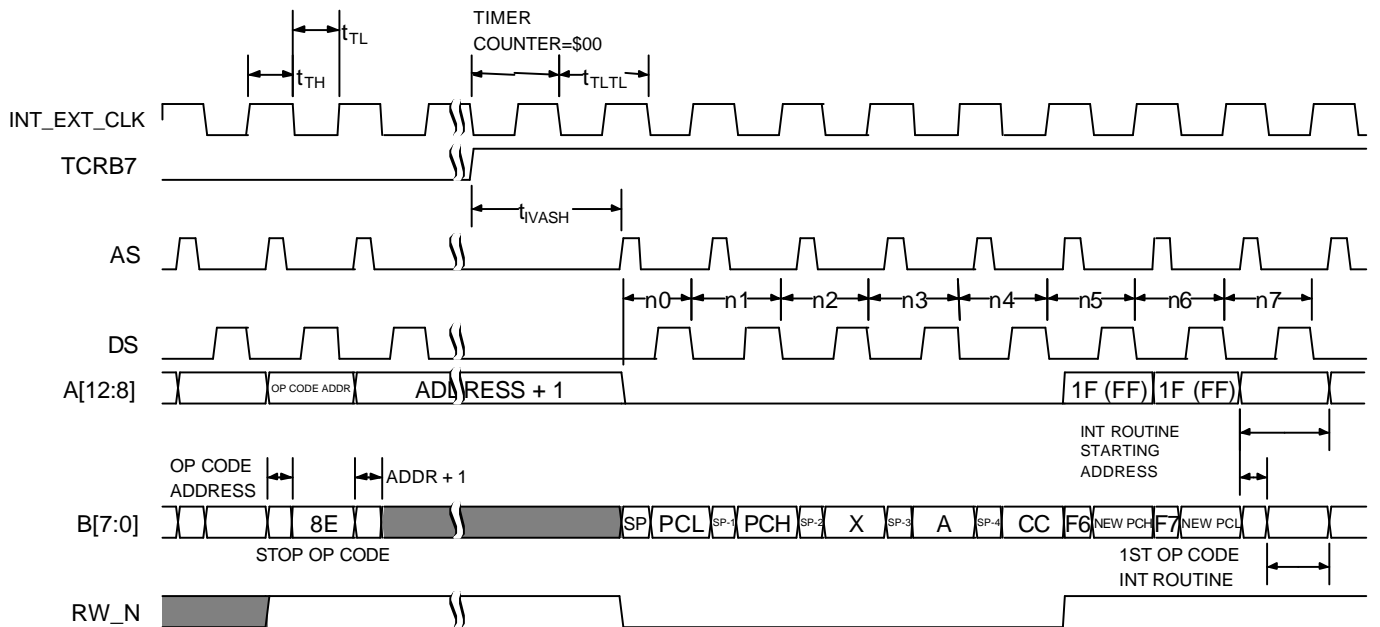
**Figure 15.  $IRQ_n$  and  $TCR7_N$  Interrupt Timing**



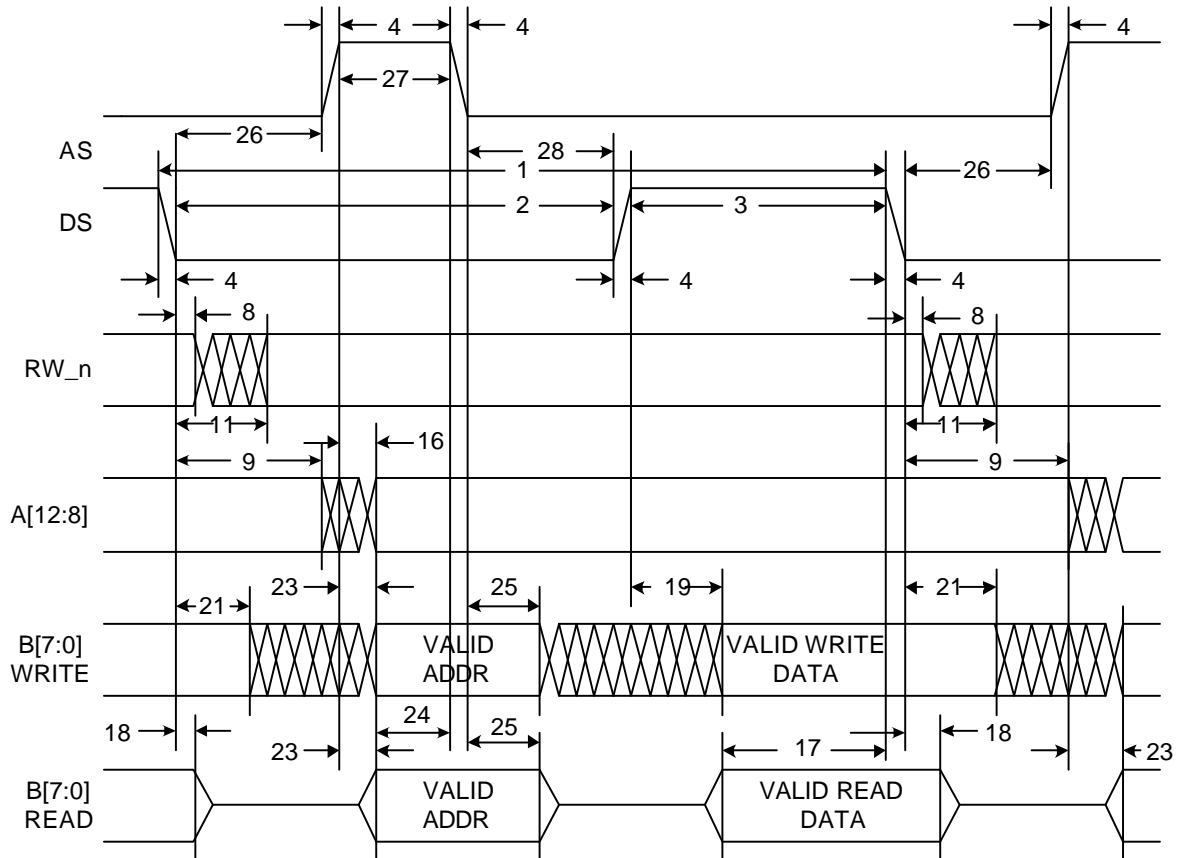
**Figure 16. Power-On-Reset and  $RESET_n$  Timing**



**Figure 17. Timer Interrupt After WAIT Instruction Timing**



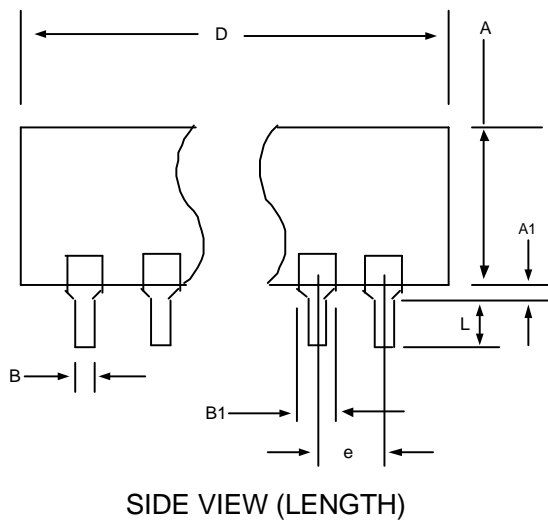
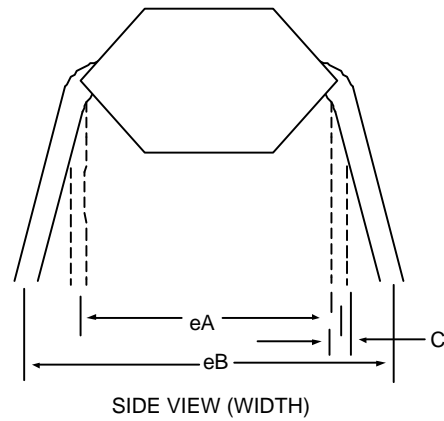
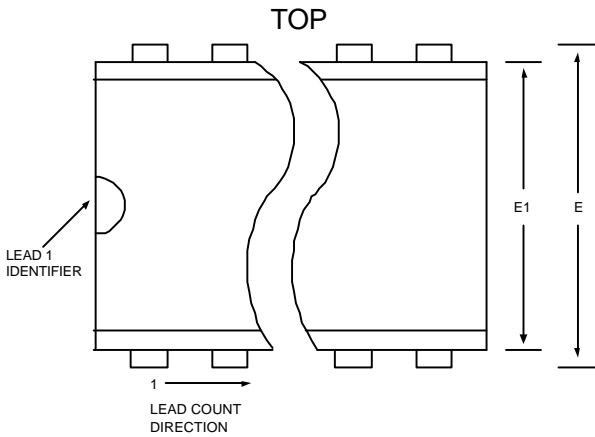
**Figure 18. Interrupt Recovery From STOP Instruction Timing**



**Figure 19. Bus Timing**

**Packaging Information**

PDIP Packaging



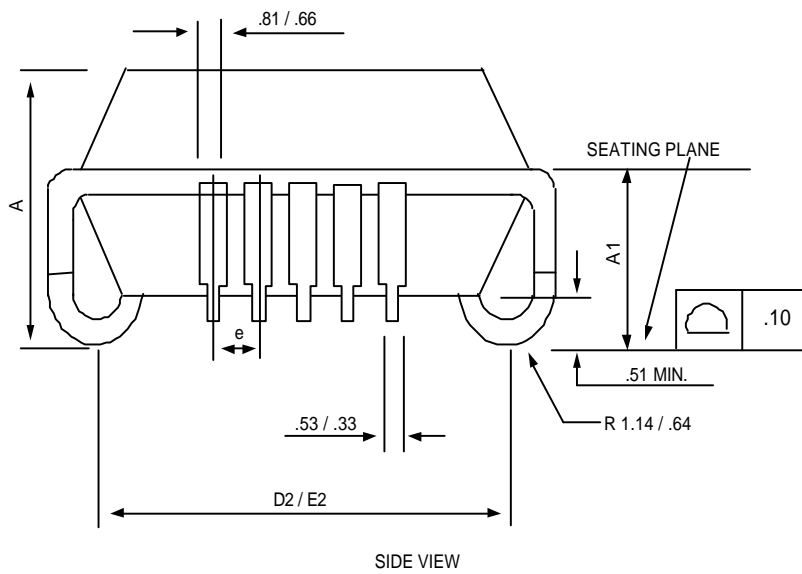
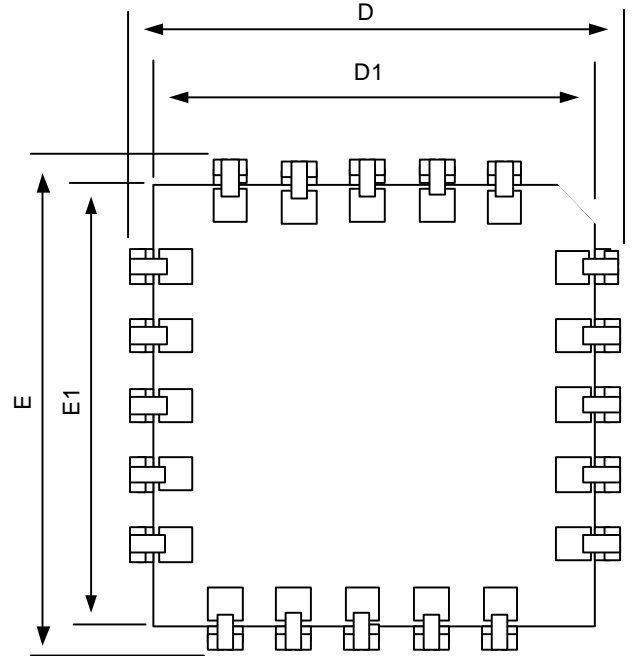
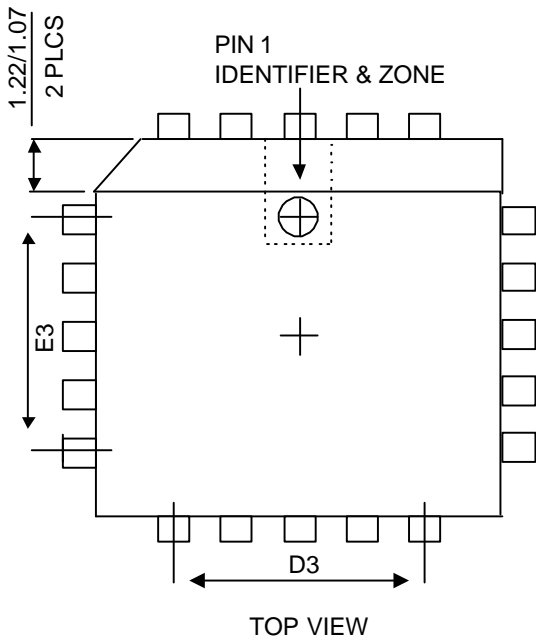
Lead Count

Symbol	40 (in Inches)	
	MIN	MAX
A	-	.200
A1	.015	-
B	.015	.020
B1	.040	.060
C	.008	.012
D	1.980	2.065
E	.580	.610
E1	.520	.560
e	.100 TYP	
eA	.580	-
eB	-	.686
L	.100 MIN	

# IA6805E2 Microprocessor Unit

**Data Sheet**  
As of Production Version 00

## PLCC Packaging



### LEAD COUNT

Symbol	44 (in Millimeters)	
	MIN	MAX
A	4.20	4.57
A1	2.29	3.04
D1	16.51	16.66
D2	14.99	16.00
D3	12.70 BSC	
E1	16.51	16.66
E2	14.99	16.00
E3	12.70 BSC	
e	1.27 BSC	
D	17.40	17.67
E	17.40	17.65

## Ordering Information

The IA6805E2 is available in two package styles listed in the table below. Other packages and temperature grades may be available for additional cost and lead time.

Package Type	Temperature Grade	Order Number
40 Lead Plastic DIP, 600 mil wide	Industrial	IA6805E2-PDW40I
44 Lead Plastic Leaded Chip Carrier	Industrial	IA6805E2-PLC44I

## Cross Reference to Original Manufacturers

innovASIC Part Number	Motorola® Part Number	Harris® Part Number
IA6805E2-PDW40I	<input type="checkbox"/> MC146805E2CP	<input type="checkbox"/> CDP6805E2CE
	<input type="checkbox"/> MC146805E2P	<input type="checkbox"/> CDP6805E2E
IA6805E2-PLC44I	<input type="checkbox"/> MC146805E2CFN	<input type="checkbox"/> CDP6805E2CQ
	<input type="checkbox"/> MC146805E2FN	<input type="checkbox"/> CDP6805E2Q

## Errata

Production Version 00

1. *Functional differences between IA6805E2 and Harris and Motorola Versions:* Stop mode on IA6805E2 will not halt oscillator. Recovery from stop will be quicker.
2. *Observations:*
  - A. Original data sheets for Motorola and Harris are inconsistent when describing timer input mode 2. Original parts and InnovASIC will AND together the timer input with the inverse of the internal clock (AS).
  - B. Original Harris part would unpredictably “pre-increment” timer counter when writing to timer registers. IA6805E2 will not.
  - C. Original Harris part displays incorrect address on external pins during intermediate cycles (not a functional problem) of multi-cycle instructions when accessing memory at page boundaries. IA6805E2 will not.
  - D. Execution of illegal op-codes on the IA6805E2 will force a system reset. On the original Harris and Motorola parts, execution of illegal op-codes would produce unpredictable results.