PRELIMINARY

8031/8051/8751
SINGLE-COMPONENT 8-BIT MICROCOMPUTER

- 8031 - Control Oriented CPU With RAM and I/O
- 8051 - An 8031 With Factory Mask-Programmable ROM
- 8751 - An 8031 With User Programmable/Erasable EPROM

- 4K x 8 ROM/EPROM
- 128 x 8 RAM
- Four 8-Bit Ports, 32 I/O Lines
- Two 16-Bit Timer/Event Counters
- High-Performance Full-Duplex Serial Channel
- External Memory Expandable to 128K
- Compatible with MCS-80®/MCS-85® Peripherals
- Boolean Processor
- MCS-48® Architecture Enhanced with:
  - Non-Paged Jumps
  - Direct Addressing
  - Four 8-Register Banks
  - Stack Depth Up to 128-Bytes
  - Multiply, Divide, Subtract, Compare
- Most Instructions Execute in 1μs
- 4μs Multiply and Divide

The Intel® 8031/8051/8751 is a stand-alone, high-performance single-chip computer fabricated with Intel's highly-reliable +5 Volt, depletion-load, N-Channel, silicon-gate HMOS technology and packaged in a 40-pin DIP. It provides the hardware features, architectural enhancements and new instructions that are necessary to make it a powerful and cost effective controller for applications requiring up to 64K bytes of program memory and/or up to 64K bytes of data storage.

The 8051/8751 contains a non-volatile 4K x 8 read-only program memory; a volatile 128 x 8 read/write data memory; 32 I/O lines; two 16-bit timer/counters; a five-source, two-priority-level, nested interrupt structure; a serial I/O port for either multi-processor communications, I/O expansion, or full duplex UART; and on-chip oscillator and clock circuits. The 8031 is identical, except that it lacks the program memory. For systems that require extra capability, the 8051 can be expanded using standard TTL compatible memories and the byte oriented MCS-80 and MCS-85 peripherals.

The 8051 microcomputer, like its 8048 predecessor, is efficient both as a controller and as an arithmetic processor. The 8051 has extensive facilities for binary and BCD arithmetic and excels in bit-handling capabilities. Efficient use of program memory results from an instruction set consisting of 44% one-byte, 41% two-byte, and 15% three-byte instructions. With a 12 MHz crystal, 58% of the instructions execute in 1μs, 40% in 2μs and multiply and divide require only 4μs. Among the many instructions added to the standard 8048 instruction set are multiply, divide, subtract and compare.

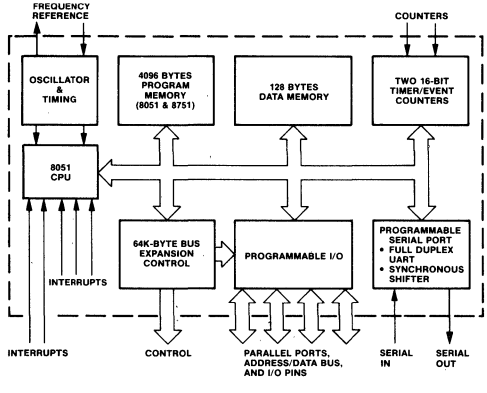


Figure 1.
Block Diagram

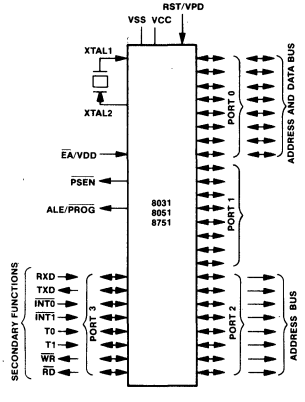


Figure 2.
Logic Symbol

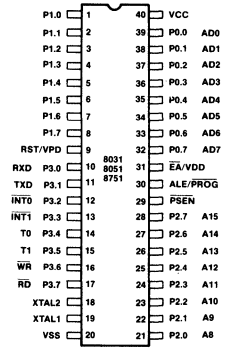


Figure 3. Pin
Configuration

AFN-01462B-01

INTRODUCTION

This data sheet provides an introduction to the 8051 family. A detailed description of the hardware required to expand the 8051 with more program memory, data memory, I/O, specialized peripherals and into multiprocessor configurations is described in the 8051 Family User's Manual.

The 8051 Family

The 8051 is a stand-alone high-performance single-chip computer intended for use in sophisticated real-time applications such as instrumentation, industrial control and intelligent computer peripherals. It provides the hardware features, architectural enhancements and new instructions that make it a powerful and cost effective controller for applications requiring up to 64K-bytes of program memory and/or up to 64K-bytes of data storage. A Block Diagram is shown in Figure 1.

The 8031 is a control-oriented CPU without on-chip program memory. It can address 64K-bytes of external Program Memory in addition to 64K-bytes of External Data Memory. For systems requiring extra capability, each member of the 8051 family can be expanded using standard memories and the byte oriented MCS-80 and MCS-85 peripherals. The 8051 is an 8031 with the lower 4K-bytes of Program Memory filled with on-chip mask programmable ROM while the 8751 has 4K-bytes of UV-light-erasable/electrically-programmable ROM.

The three pin-compatible versions of this component reduce development problems to a minimum and provide maximum flexibility. The 8751 is well suited for development, prototyping, low-volume production and applications requiring field updates; the 8051 for low-cost, high volume production; and the 8031 for applications desiring the flexibility of external Program Memory which can be easily modified and updated in the field.

MACRO-VIEW OF THE 8051 ARCHITECTURE

On a single die the 8051 microcomputer combines CPU; non-volatile 4K x 8 read-only program memory; volatile 128 x 8 read/write data memory; 32 I/O lines; two 16-bit timer/event counters; a five-source, two-priority-level, nested interrupt structure; serial I/O port for either multi-processor communications, I/O expansion, or full duplex UART; and on-chip oscillator and clock circuits. This section will provide an overview of the 8051 by providing a high-level description of its major elements: the CPU architecture and the on-chip functions peripheral to the CPU. The generic term "8051" is used to refer collectively to the 8031, 8051, and 8751.

8051 CPU Architecture

The 8051 CPU manipulates operands in four memory spaces. These are the 64K-byte Program Memory, 64K-byte External Data Memory, 384-byte Internal Data Memory and 16-bit Program Counter spaces. The Internal Data Memory address space is further divided into the 256-byte Internal Data RAM and 128-byte Special Function Register (SFR) address spaces shown in Figure 4. Four Register Banks (each with eight registers), 128 addressable bits, and the stack reside in the Internal Data RAM. The stack depth is limited only by the available Internal Data RAM and its location is determined by the 8-bit stack pointer. All registers except the Program Counter and the four 8-Register Banks reside in the Special Function Register address space. These memory mapped registers include arithmetic registers, pointers, I/O ports, interrupt system registers, timers and serial port. 128 bit locations in the SFR address space are addressable as bits. The 8051 contains 128 bytes of Internal Data RAM and 20 SFRs.

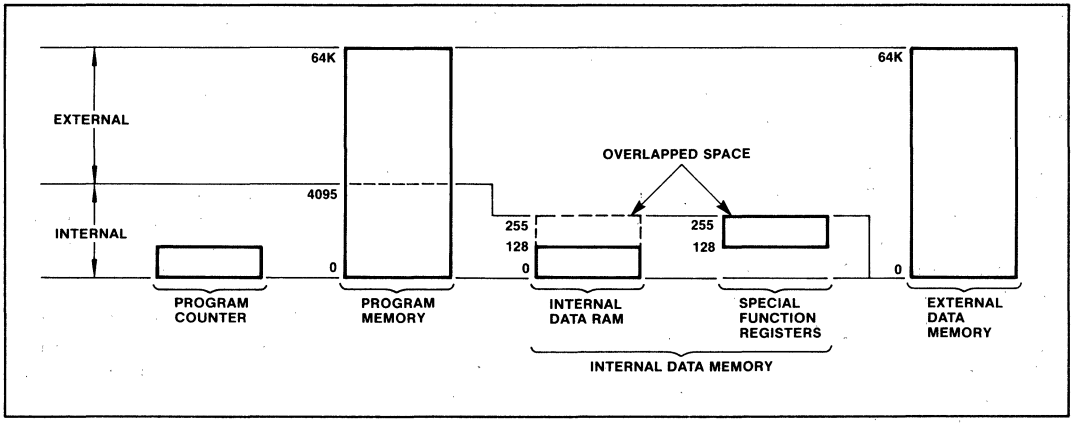


Figure 4. 8051 Family Memory Organization

The 8051 provides a non-paged Program Memory address space to accommodate relocatable code. Conditional branches are performed relative to the Program Counter. The register-indirect jump permits branching relative to a 16-bit base register with an offset provided by an 8-bit index register. Sixteen-bit jumps and calls permit branching to any location in the contiguous 64K Program Memory address space.

The 8051 has five methods for addressing source operands: Register, Direct, Register-Indirect, Immediate and Base-Register- plus Index-Register-Indirect Addressing. The first three methods can be used for addressing destination operands. Most instructions have a "destination, source" field that specifies the data type, addressing methods and operands involved. For operations other than moves, the destination operand is also a source operand.

Registers in the four 8-Register Banks can be accessed through Register, Direct, or Register-Indirect Addressing; the 128 bytes of Internal Data RAM through Direct or Register-Indirect Addressing; and the Special Function Registers through Direct Addressing. External Data Memory is accessed through Register-Indirect Addressing. Look-Up-Tables resident in Program Memory can be accessed through Base-Register- plus Index-Register- Indirect Addressing.

The 8051 is classified as an 8-bit machine since the internal ROM, RAM, Special Function Registers, Arithmetic/Logic Unit and external data bus are each 8-bits wide. The 8051 performs operations on bit, nibble, byte and double-byte data types.

The 8051 has extensive facilities for byte transfer, logic, and integer arithmetic operations. It excels at bit handling since data transfer, logic and conditional branch operations can be performed directly on Boolean variables.

The 8051's instruction set is an enhancement of the instruction set familiar to MCS-48 users. It is enhanced to allow expansion of on-chip CPU peripherals and to optimize byte efficiency and execution speed. Op codes were reassigned to add new high-power operations and to permit new addressing modes which make the old operations more orthogonal. Efficient use of program memory results from an instruction set consisting of 49 single-byte, 45 two-byte and 17 three-byte instructions. When using a 12 MHz oscillator, 64 instructions execute in 1μs and 45 instructions execute in 2μs. The remaining instructions (multiply and divide) execute in only 4μs. The number of bytes in each instruction and the number of oscillator periods required for execution are listed in the appended 8051 Instruction Set Summary.

On-Chip Peripheral Functions

Thus far only the CPU and memory spaces of the 8051 have been described. In addition to the CPU and memories, an interrupt system, extensive I/O facilities, and several peripheral functions are integrated on-chip to relieve the CPU of repetitious, complicated or time-critical tasks and to permit stringent real-time control of external system interfaces. The extensive I/O facilities include the I/O pins, parallel I/O ports, bidirectional address/data bus and the serial port for I/O expansion. The CPU peripheral functions integrated on-chip are the two 16-bit counters and the serial port. All of these work together to greatly boost system performance.

INTERRUPT SYSTEM

External events and the real-time-driven on-chip peripherals require service by the CPU asynchronous to the execution of any particular section of code. To tie the asynchronous activities of these functions to normal program execution, a sophisticated multiple-source, two-priority-level, nested interrupt system is provided. Interrupt response latency ranges from 3μs to 7μs when using a 12 MHz crystal.

The 8051 acknowledges interrupt requests from five sources: Two from external sources via the $\overline{\text{INT0}}$ and $\overline{\text{INT1}}$ pins, one from each of the two internal counters and one from the serial I/O port. Each interrupt vectors to a separate location in Program Memory for its service program. Each of the five sources can be assigned to either of two priority levels and can be independently enabled and disabled. Additionally all enabled sources can be globally disabled or enabled. Each external interrupt is programmable as either level- or transition-activated and is active-low to allow the "wire or-ing" of several interrupt sources to the input pin. The interrupt system is shown diagrammatically in Figure 5.

I/O FACILITIES

The 8051 has instructions that treat its 32 I/O lines as 32 individually addressable bits and as four parallel 8-bit ports addressable as Ports 0, 1, 2 and 3. Ports 0, 2 and 3 can also assume other functions. Port 0 provides the multiplexed low-order address and data bus used for expanding the 8051 with standard memories and peripherals. Port 2 provides the high-order address bus when expanding the 8051 with external Program Memory or more than 256 bytes of External Data Memory. The pins of Port 3 can be configured individually to provide external interrupt request inputs, counter inputs, the serial port's receiver input and transmitter output, and to generate the control signals used for reading and writing External Data Memory. The generation or use of an alternate function on a Port 3 pin is done automatically by the 8051 as long as the pin

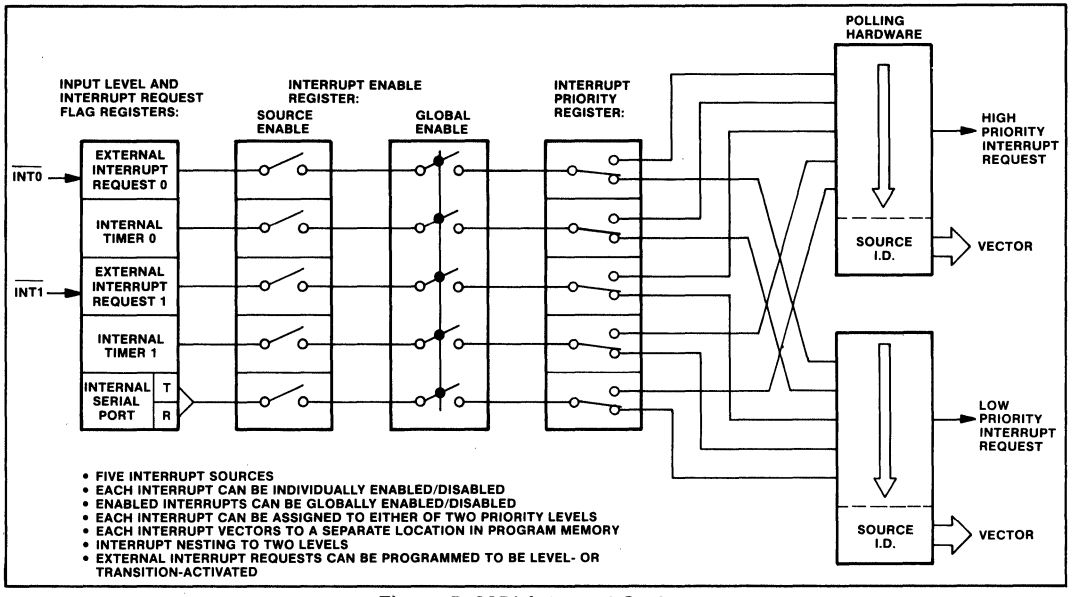


Figure 5. 8051 Interrupt System

is configured as an input. The configuration of the ports is shown on the 8051 Family Logic Symbol of Figure 2.

Open Drain I/O Pins

Each pin of Port 0 can be configured as an open drain output or as a high impedance input. Resetting the microcomputer programs each pin as an input by writing a one (1) to the pin. If a zero (0) is later written to the pin it becomes configured as an output and will continuously sink current. Re-writing the pin to a one (1) will place its output driver in a high-impedance state and configure the pin as an input. Each I/O pin of Port 0 can sink two TTL loads.

Quasi-Bidirectional I/O Pins

Ports 1, 2 and 3 are quasi-bidirectional buffers. Resetting the microcomputer programs each pin as an input by writing a one (1) to the pin. If a zero (0) is later written to the pin it becomes configured as an output and will continuously sink current. Any pin that is configured as an output will be reconfigured as an input when a one (1) is written to the pin. Simultaneous to this reconfiguration the output driver of the quasi-bidirectional port will source current for two oscillator periods. Since current is sourced only when a bit previously written to a zero (0) is updated to a one (1), a pin programmed as an input will not source current into the TTL gate that is driving it if the pin is later written with another one (1). Since the quasi-bidirectional output driver sources current for only two oscillator periods, an internal pullup resistor of approximately 20K- to 40K-ohms is provided to hold the external driver's loading at a TTL high level. Ports 1, 2 and 3 can sink/source one TTL load.

Microprocessor Bus

A microprocessor bus is provided to permit the 8051 to solve a wide range of problems and to allow the upward growth of user products. This multiplexed address and data bus provides an interface compatible with standard memories, MCS-80 peripherals and the MCS-85 memories that include on-chip programmable I/O ports and timing functions. These are summarized in the 8051 Microcomputer Expansion Components chart of Figure 6.

When accessing external memory the high-order address is emitted on Port 2 and the low-order address on Port 0. The ALE signal is provided for strobing the address into an external latch. The program store enable ($\overline{PSEN}$) signal is provided for enabling an external memory device to Port 0 during a read from the Program Memory address space. When the MOVX instruction is executed Port 3 automatically generates the read ($\overline{RD}$) signal for enabling an External Data Memory device to Port 0 or generates the write ($\overline{WR}$) signal for strobing the external memory device with the data emitted by Port 0. Port 0 emits the address and data to the external memory through a push/pull driver that can sink/source two TTL loads. At the end of the read/write bus cycle Port 0 is automatically reprogrammed to its high

	Category	I.D.	Description	Comments	Program Or Data Memory	Crystal Frequency MHz (Max)
	I/O Expander		8 Line I/O Expander (Shift Register)	Low Cost I/O Expander		12
Compatible MCS-80/85 Components	Standard EPROMs	2758	1K x 8 450 ns Light Erasable	User programmable and erasable.	P	9
		2716-1	2K x 8 350 ns Light Erasable		P	11
		2732	4K x 8 450 ns Light Erasable		P	9
		2732A	4K x 8 250 ns Light Erasable		P	12
	Standard RAMs	2114A	1K x 4 100 ns RAM	Data memory can be easily expanded using standard NMOS RAMs.	D	12
		2148	1K x 4 70 ns RAM		D	12
		2142-2	1K x 4 200 ns RAM		D	12
	Multiplexed Address/ Data RAMs	8185A	1K x 8 300 ns RAM		D	12
	Standard I/O	8212	8-Bit I/O Port	Serves as Address Latch or I/O port.	D	12
		8282	8-Bit I/O Port		D	12
		8283	8-Bit I/O Port		D	12
		8255A	Programmable Peripheral Interface	Three 8-bit programmable I/O ports.	D	12
		8251A	Programmable Communications Interface	Serial Communications Receiver/Transmitter.	D	12
	Standard Peripherals	8205	1 of 8 Binary Decoder	MCS-80 and MCS-85 peripheral devices are compatible with the 8051 allowing easy addition of specialized interfaces. Future MCS-80/85 devices will also be compatible.	D	12
		8286	Bi-directional Bus Driver		D	12
		8287	Bi-directional Bus Driver (Inverting)		D	12
		8253A	Programmable Interval Timer		D	12
		8279	Programmable Keyboard/Display Interface (128 Keys)		D	12
		8291	GPIB Talker/Listener		D	12
		8292	GPIB Controller		D	11.7
	Universal Peripheral Interfaces	8041A	ROM Program Memory	User programmable to perform custom I/O and control functions.	D/P	12/11.7
		8741A	EPROM Program Memory		D/P	12/11.7
	Memories with on-chip I/O and Peripheral Functions.	8155-2	256 x 8 330 ns RAM		D	12
		8355-2	2K x 8 330 ns ROM		P	11.6
		8755-2	2K x 8 330 ns EPROM		P	11.6

Figure 6. 8051 Microcomputer Expansion Components

impedance state and Port 2 is returned to the state it had prior to the bus cycle. The 8051 generates the address, data and control signals needed by memory and I/O devices in a manner that minimizes the requirements placed on external program and data memories. At 12 MHz, the Program Memory cycle time is 500ns and the access times required from stable address and $\overline{PSEN}$ are approximately 320ns and 150ns respectively. The External Data Memory cycle time is 1μs and the access times required from stable address and from read ($\overline{RD}$) or write ($\overline{WR}$) command are approximately 600ns and 250ns respectively.

TIMER/EVENT COUNTERS

The 8051 contains two 16-bit counters for measuring time intervals, measuring pulse widths, counting events and generating precise, periodic interrupt requests. Each can be programmed independently to operate similar to an 8048 8-bit timer with divide by 32 prescaler or as an 8-bit counter with divide by 32 prescaler (Mode 0), as a 16-bit time-interval or event counter (Mode 1), or as an 8-bit time-interval or event counter with automatic reload upon overflow (Mode 2).

Additionally, counter 0 can be programmed to a mode that divides it into one 8-bit time-interval or

event counter and one 8-bit time-interval counter (Mode 3). When counter 0 is in Mode 3, counter 1 can be programmed to any of the three aforementioned modes, although it cannot set an interrupt request flag or generate an interrupt. This mode is useful because counter 1's overflow can be used to pulse the serial port's transmission-rate generator. Along with their multiple operating modes and 16-bit precision, the counters can also handle very high input frequencies. These range from 0.1 MHz to 1.0 MHz (for 1.2 MHz to 12 MHz crystal) when programmed for an input that is a division by 12 of the oscillator frequency and from 0 Hz to an upper limit of 50 KHz to 0.5 MHz (for 1.2 MHz to 12 MHz crystal) when programmed for external inputs. Both internal and external inputs can be gated to the counter by a second external source for directly measuring pulse widths.

The counters are started and stopped under software control. Each counter sets its interrupt request flag when it overflows from all ones to all zeros (or auto-reload value). The operating modes and input sources are summarized in Figures 7 and 8. The effects of the configuration flags and the status flags are shown in Figures 9 and 10.

SERIAL COMMUNICATIONS

The 8051 has a serial I/O port that is useful for serially linking peripheral devices as well as multiple 8051s through standard asynchronous protocols with full-duplex operation. The serial port also has a synchronous mode for expansion of I/O lines using CMOS and TTL shift registers. This hardware serial communications interface saves ROM code and permits a much higher transmission rate than could be achieved through software. In response to a serial port interrupt request the CPU has only to read/write the serial port's buffer to service the serial link. A block diagram of the serial port is shown in Figures 11 and 12. Methods for linking UART (universal asynchronous receiver/transmitter) devices are

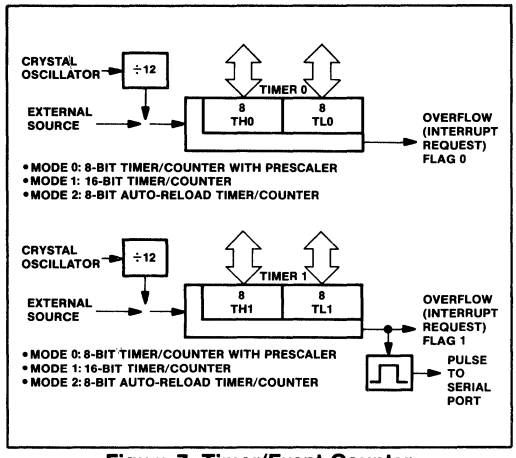


Figure 7. Timer/Event Counter Modes 0, 1 and 2

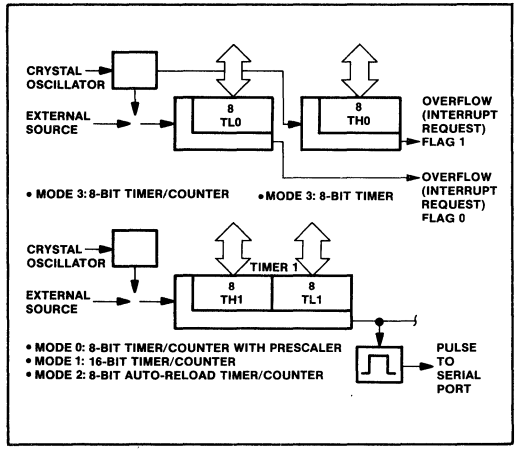


Figure 8. Timer/Event Counter 0 in Mode 3

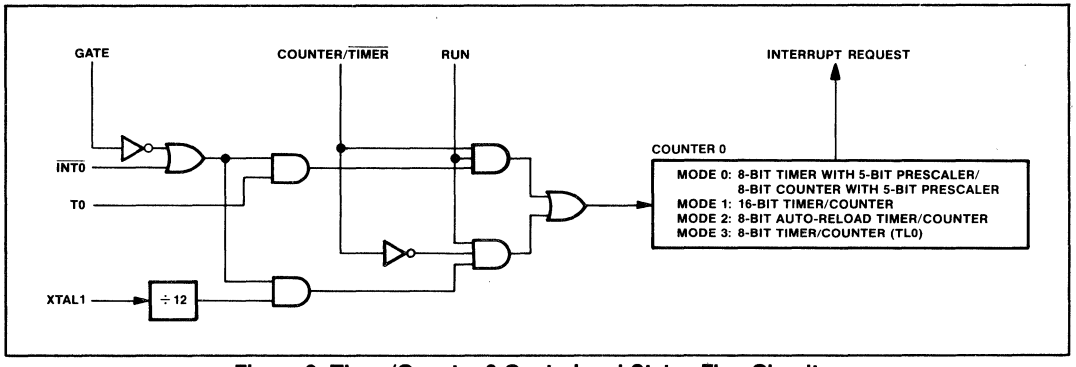


Figure 9. Timer/Counter 0 Control and Status Flag Circuitry

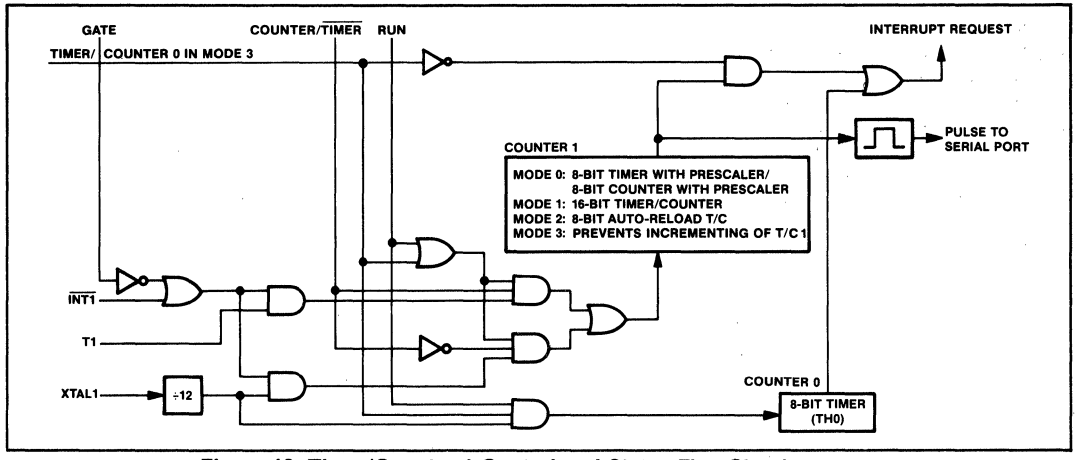


Figure 10. Timer/Counter 1 Control and Status Flag Circuitry

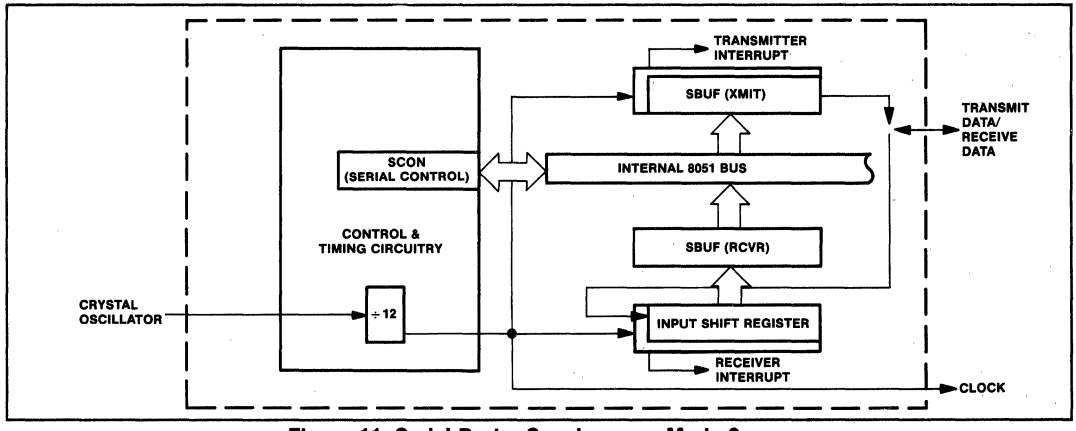


Figure 11. Serial Port—Synchronous Mode 0

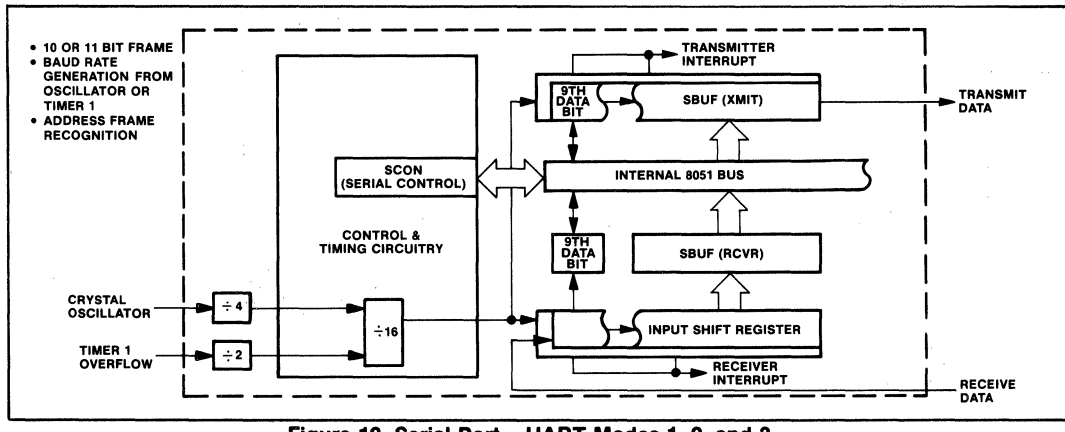


Figure 12. Serial Port—UART Modes 1, 2, and 3

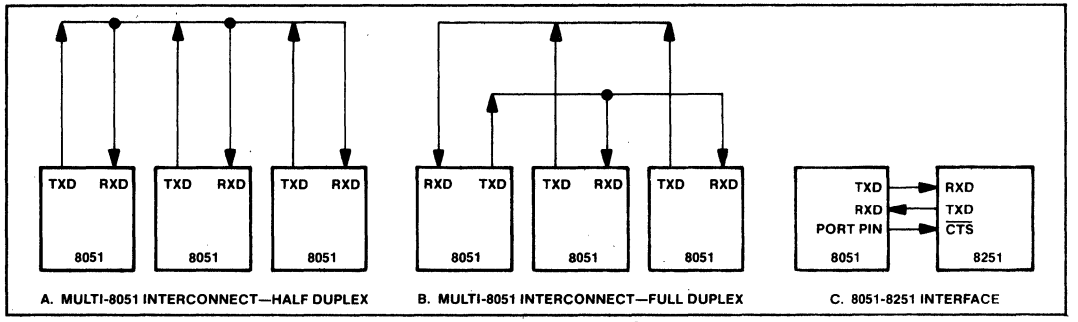


Figure 13. UART Interfacing Schemes

shown in Figure 13 and a method for I/O expansion is shown in Figure 14.

The full-duplex serial I/O port provides asynchronous modes to facilitate communications with standard UART devices, such as printers and CRT terminals, or communications with other 8051s in multi-processor systems. The receiver is double buffered to eliminate the overrun that would occur if the CPU failed to respond to the receiver's interrupt before the beginning of the next frame. Double buffering of the transmitter is not needed since the 8051 can generally maintain the serial link at its maximum rate without it. A minor degradation in transmission rate can occur in rare events such as when the servicing of the transmitter has to wait for a lengthy interrupt service program to complete. In asynchronous modes, false start-bit rejection is provided on received frames. For noise rejection a best two-out-of-three vote is taken on three samples near the center of each received bit.

When interfacing with standard UART devices the serial channel can be programmed to a mode (Mode 1) that transmits/receives a ten-bit frame or programmed to a mode (Mode 2 or 3) that transmits/receives an eleven-bit frame as shown in Figure 15. The frame consists of a start bit, eight or nine data bits and a stop bit. In Modes 1 and 3, the transmission-rate timing circuitry receives a pulse from counter 1 each time the counter overflows. The input to counter 1 can be an external source or a division by 12 of the oscillator frequency. The auto-reload mode of the counter provides communication rates of 122 to 31,250 bits per second (including start and stop bits) for a 12 MHz crystal. In Mode 2 the communication rate is a division by 64 of the oscillator frequency yielding a transmission rate of 187,500 bits per second (including start and stop bits) for a 12 MHz crystal.

Distributed processing offers a faster, more powerful system than can be provided by a single CPU processor. This results from a hierarchy of interconnected processors, each with its own memories and

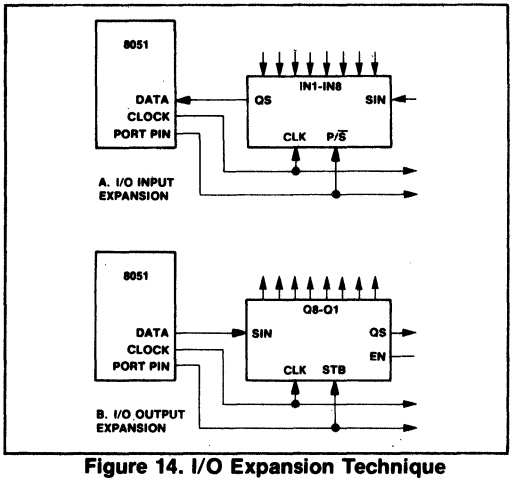


Figure 14. I/O Expansion Technique

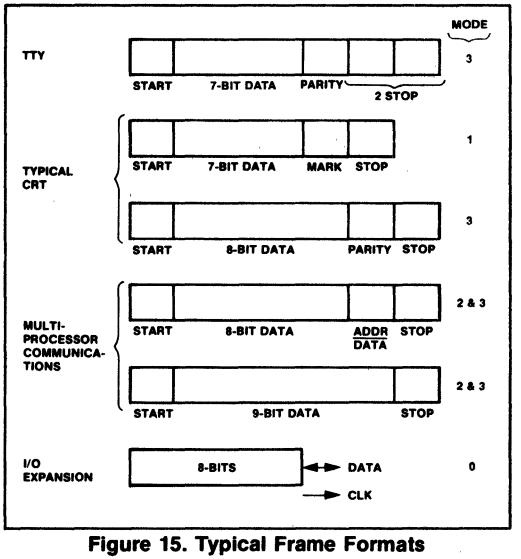


Figure 15. Typical Frame Formats

1. Slaves—Configure serial port to interrupt CPU if the received ninth data bit is a one (1).
2. Master—Transmit frame containing address in first 8 data bits and set ninth data bit (i.e. ninth data bit designates address frame).
3. Slaves—Serial port interrupts CPU when address frame is received. Interrupt service program compares received address to its address. The slave which has been addressed reconfigures its serial port to interrupt the CPU on all subsequent transmissions.
4. Master—Transmit control frames and data frames (these will be accepted only by the previously addressed slave).

Figure 16. Protocol for Multi-Processor Communications

I/O. In multiprocessing, a host 8051 microcomputer controls a multiplicity of 8051s configured to operate simultaneously on separate portions of the program, each controlling a portion of the overall process. The interconnected 8051s reduce the load on the host processor and result in a low-cost system of data transmission. This form of distributed processing is especially effective in systems where controls in a complex process are required at physically separated locations.

In Modes 2 and 3 the automatic wake-up of slave processors through interrupt driven address-frame recognition is provided to facilitate interprocessor communications. The protocol for interprocessor communications is shown in Figure 16.

In synchronous mode (Mode 0) the high speed serial port provides an efficient, low-cost method of expanding I/O lines using standard TTL and CMOS shift registers. The serial channel provides a clock output for synchronizing the shifting of bits to/from an external register. The data rate is a division by 12 of the oscillator frequency and is 1M bits per second at 12 MHz.

8051 Family Pin Description

V_{SS}
Circuit ground potential.

V_{CC}
+5V power supply during operation, programming and verification.

PORT 0

Port 0 is an 8-bit open drain bidirectional I/O port. It is also the multiplexed low-order address and data bus when using external memory. It is used for data input and output during programming and verification. Port 0 can sink/source two TTL loads.

PORT 1

Port 1 is an 8-bit quasi-bidirectional I/O port. It is used for the low-order address byte during programming and verification. Port 1 can sink/source one TTL load.

PORT 2

Port 2 is an 8-bit quasi-bidirectional I/O port. It also emits the high-order address byte when accessing external memory. It is used for the high-order address and the control signals during programming and verification. Port 2 can sink/source one TTL load.

PORT 3

Port 3 is an 8-bit quasi-bidirectional I/O port. It also contains the interrupt, timer, serial port and $\overline{RD}$ and $\overline{WR}$ pins that are used by various options. The output latch corresponding to a secondary function must be programmed to a one (1) for that function to operate. Port 3 can sink/source one TTL load. The secondary functions are assigned to the pins of Port 3, as follows:

- RXD/data (P3.0). Serial port's receiver data input (asynchronous) or data input/output (synchronous).
- TXD/clock (P3.1). Serial port's transmitter data output (asynchronous) or clock output (synchronous).
- $\overline{INT0}$ (P3.2). Interrupt 0 input or gate control input for counter 0.
- $\overline{INT1}$ (P3.3). Interrupt 1 input or gate control input for counter 1.
- T0 (P3.4). Input to counter 0.
- T1 (P3.5). Input to counter 1.
- $\overline{WR}$ (P3.6). The write control signal latches the data byte from Port 0 into the External Data Memory.
- $\overline{RD}$ (P3.7). The read control signal enables External Data Memory to Port 0.

RST/V_{PD}

A low to high transition on this pin (at approximately 3V) resets the 8051. If V_{PD} is held within its spec (approximately +5V), while V_{CC} drops below spec, V_{PD} will provide standby power to the RAM. When V_{PD} is low, the RAM's current is drawn from V_{CC}. A small internal resistor permits power-on reset using only a capacitor connected to V_{CC}.

ALE/$\overline{PROG}$

Provides Address Latch Enable output used for latching the address into external memory during normal operation. Receives the program pulse

input during EPROM programming.

$\overline{\text{PSEN}}$

The Program Store Enable output is a control signal that enables the external Program Memory to the bus during normal fetch operations.

$\overline{\text{EA}}$/VDD

When held at a TTL high level, the 8051 executes instructions from the internal ROM/EPROM when the PC is less than 4096. When held at a TTL low level, the 8051 fetches all instuctions from external Program Memory. The pin also receives the 21V EPROM programming supply voltage.

XTAL1

Input to the oscillator's high gain amplifier. A crystal or external source can be used.

XTAL2

Output from the oscillator's amplifier. Required when a crystal is used.

8051 FAMILY DEVELOPMENT SYSTEM AND SOFTWARE SUPPORT

The 8051 is supported by a total range of Intel development tools. This broad range of support shortens the product development cycle and thus brings the product to market sooner.

- ASM51 Absolute macro assembler for the 8051.
- CONV51 8048 assembly language source code to 8051 assembly source code conversion program.
- EM-51 8051/8751 emulator board that uses a modified 8051 and an EPROM.
- ICE-51™ Real-time in-circuit emulator.
- SDK-51 System design kit for developing user Prototype around the 8051.
- UPP-551 8751 personality card for UPP-103 Universal PROM Programmer.
- 8051 Workshop.

8051 Software Development Package (ASM51 and CONV51)

The 8051 software development package provides development system support for the powerful 8051 family of single-chip microcomputers. The package contains a symbolic macro assembler and a 8048 to 8051 source code converter. This diskette-based software package runs under ISIS-II on any Intellec® Microcomputer Development System with 64K bytes of memory.

8051 Macro Assembler (ASM51)

The 8051 macro assembler translates symbolic 8051 assembly language instructions into machine executable object code. These assembly language mnemonics are easier to program and are more readable than binary or hexidecimal machine instructions. Also, by allowing the programmer to give symbolic names to memory locations rather than absolute addresses, software design and debug are performed more quickly and reliably.

ASM51 provides symbolic access for the many useful addressing methods in the 8051 architecture which reference bit, nibble and byte locations.

The assembler supports macro definitions and calls. This provides a convenient means of programming a frequently used code sequence only once. The assembler also provides conditional assembly capabilities. Cross referencing is provided in the symbol table listing, which shows the user the lines in which each symbol was defined and referenced.

If an 8051 program contains errors, the assembler provides a comprehensive set of error diagnostics, which are included in the assembly listing.

The object code generated may be used to program the 8751 EPROM version of the chip or sent to Intel for fabricating the 8051 ROM version. The assembler output can also be debugged using the ICE-51 in-circuit emulator.

8048 to 8051 Assembly Language Converter Utility Program (CONV51)

The 8048 to 8051 assembly language converter is a utility to help users of the MCS-48 family of microcomputers upgrade their designs to the high performance 8051 architecture. By converting 8048 source code to 8051 source code, the investment in software developed for the 8048 is maintained when the system is upgraded.

8051 Emulation Board (EM-51)

The EM-51 8051 emulation board is a small (2.85" x 5.25") board which emulates an 8031/8051/8751 microcomputer using standard PROMs or EPROMs in place of the 8051's on-chip program memory. The board includes a modified 8051 microcomputer, supporting circuits, and two sockets for program memory. The user may select two 2716 EPROMs, a 2732 EPROM, or two 3636 bipolar PROMs depending on crystal frequency and power requirements.

8051 In-Circuit Emulator (ICE-51™)

The 8051 In-Circuit Emulator resides in the Intellec development system. The development system interfaces with the user's 8051 system through an in-cable buffer box with the cable terminating in an 8051 pin-compatible plug. Together these replace the 8051 device in the system. With the emulator plug in place, the designer can exercise the system in real-time while collecting up to 255 instruction

cycles of real-time data. In addition, he can single step the system program.

Static RAM memory is available in the ICE-51 buffer box to emulate the 8051's internal and external program memories and external data memory. The designer can display and alter the contents of the replacement memory in the ICE-51 buffer box, internal 8051 registers, internal data RAM, and Special Function Registers. Symbolic reference capability allows the designer to use meaningful symbols provided by ASM51 rather than absolute values when examining and modifying these memory, register, flag, and I/O locations in his system.

Personality Card for Universal PROM Programmer (UPP-551)

The UPP-551 is a personality card for the UPP-103 Universal PROM Programmer. The Universal PROM Programmer is an Intellec system peripheral capable of programming and verifying the 8751 when the UPP-551 is inserted. Programming and verfication operations are initiated from the Intellec development system console and are controlled by the Universal PROM Mapper (UPM) program.

8051 Workshop

The workshop provides the design engineer or system designer hands-on experience with the 8051 microcomputers. The course includes explanation of the Intel 8051 architecture, system timing and input/output design. Lab sessions will allow the attendee to gain detailed familiarity with the 8051 family and support tools.

INSITE™ Library

The INSITE Library contains 8051 utilities and applications programs.

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias ..... 0°C to 70°C
Storage Temperature ..... -65°C to +150°C
Voltage on Any Pin With
Respect to Ground (V_{SS}) ..... -0.5V to +7V
Power Dissipation ..... 2 Watts

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

D.C. CHARACTERISTICS (T_A = 0°C to 70°C; V_{CC} = 5V ± 5%; V_{SS} = 0V)

Symbol	Parameter	Min	Typ	Max	Units	Test Conditions
V_{IL}	Input Low Voltage (All except XTAL1)	-0.5		0.8	V	
V_{IL1}	Input Low Voltage (XTAL1)	-0.5		TBD	V	
V_{IH}	Input High Voltage (All Except XTAL1, RST/V_{PD})	2.0		V_{CC}+0.5	V	
V_{IH1}	Input High Voltage (XTAL1)	TBD		V_{CC}+0.5	V	
V_{IH2}	Input High Voltage (RST)	3.0		V_{CC} + 0.5	V	
V_{IH3}	Input High Voltage (V_{PD})	4.5		5.5	V	Power Down Only (V_{CC} = 0)
V_{OL}	Output Low Voltage (All Outputs Except Port 0)			0.45	V	I_{OL}=2 mA
V_{OL1}	Output Low Voltage (Port 0)			0.45	V	I_{OL}=4 mA
V_{OH}	Output High Voltage (All Outputs Except Port 0, ALE and $\overline{PSEN}$)	2.4			V	I_{OH}=-100 μA
V_{OH1}	Output High Voltage (ALE and $\overline{PSEN}$. Port 0 in External Bus Mode)	2.4			V	I_{OH}=-400 μA
I_{LO}	Pullup Resistor Current (P1, P2, P3)			500	μA	.45V≤V_{IN}≤V_{CC}
I_{LO1}	Output Leakage Current (P0)			±10	μA	.45V≤V_{IN}≤V_{CC}
I_{CC}	Power Supply Current (All Outputs Disconnected)			150	mA	T_A=25°C
I_{PD}	Power Down Supply Current			20	mA	T_A=25°C, V_{PD}=5V, V_{CC}=0V
C_{IO}	Capacitance Of I/O Buffer			10	pF	fc=1MHz

AFN-01462B-11

A.C. CHARACTERISTICS (T_A = 0°C to 70°C; V_{CC} = 5V ± 5%; V_{SS} = 0V; C_L for Port 0, ALE and $\overline{PSEN}$ Outputs = 150 pF; C_L for All Other Outputs = 80 pF)

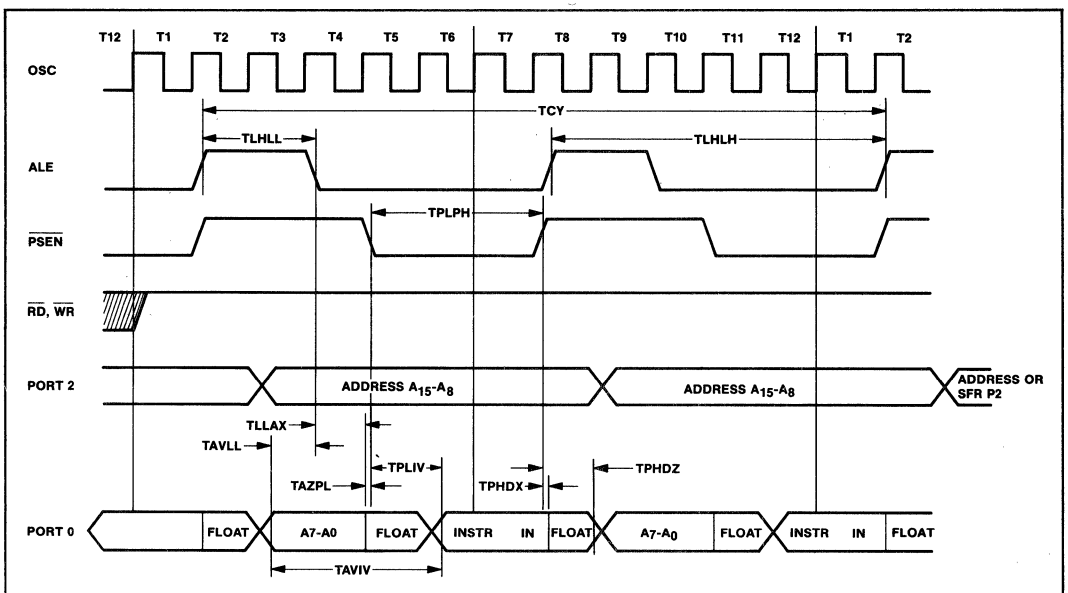
Program Memory Characteristics

Symbol	Parameter	12 MHz Clock			Variable Clock 1/TCLCL=1.2 MHz to 12 MHz		
		Min	Max	Units	Min	Max	Units
TCLCL	Oscillator Period	83		ns			ns
TCY	Min Instruction Cycle Time	1.0		μs	12TCLCL	12TCLCL	ns
TLHLL	ALE Pulse Width	140		ns	2TCLCL-30		ns
TAVLL	Address Set Up To ALE	60		ns	TCLCL-25		ns
TLLAX	Address Hold After ALE	50		ns	TCLCL-35		ns
TPLPH	$\overline{PSEN}$ Width	230		ns	3TCLCL-20		ns
TLHLH	$\overline{PSEN}$, ALE Cycle Time	500		ns	6TCLCL		ns
TPLIV	$\overline{PSEN}$ To Valid Instruction In		150	ns		3TCLCL-100	ns
TPHDX	Input Data Hold After $\overline{PSEN}$	0		ns	0		ns
TPHDZ	Input Data Float After $\overline{PSEN}$		75	ns		TCLCL-10	ns
TAVIV	Address To Valid Instr In		320	ns		5TCLCL-100	ns
TAZPL	Address Float To $\overline{PSEN}$	0		ns	0		ns

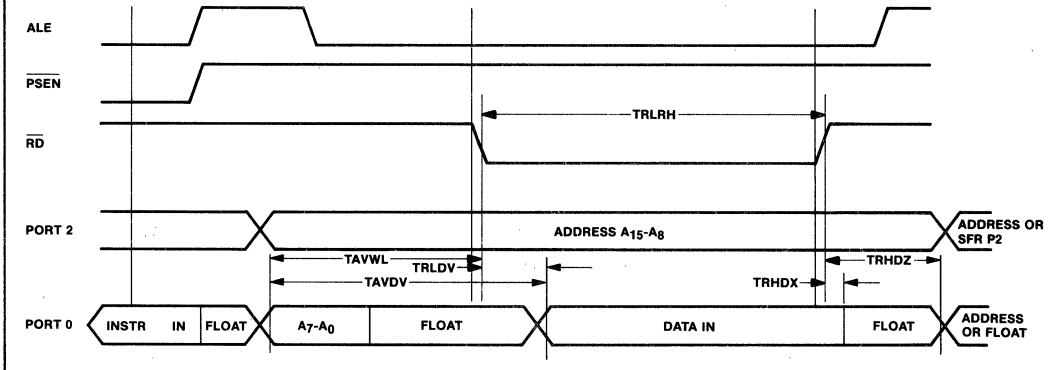
External Data Memory Characteristics

Symbol	Parameter	12 MHz Clock			Variable Clock		
		Min	Max	Units	Min	Max	Units
TRLRH	$\overline{RD}$ Pulse Width	400		ns	6TCLCL-100		ns
TWLWH	$\overline{WR}$ Pulse Width	400		ns	6TCLCL-100		ns
TRLDV	$\overline{RD}$ To Valid Data In		250	ns		5TCLCL-170	ns
TRHDX	Data Hold After $\overline{RD}$	0		ns	0		ns
TRHDZ	Data Float After $\overline{RD}$		100	ns		2TCLCL-70	ns
TAVDV	Address To Valid Data In		600	ns		9TCLCL-150	ns
TAVWL	Address To $\overline{WR}$ or $\overline{RD}$	200		ns	4TCLCL-130		ns
TDVWX	Data Valid To $\overline{WR}$ Transition			ns			ns
TQVWH	Data Setup Before $\overline{WR}$	400		ns	7TCLCL-180		ns
TWHQX	Data Held After $\overline{WR}$	80		ns	2TCLCL-90		ns

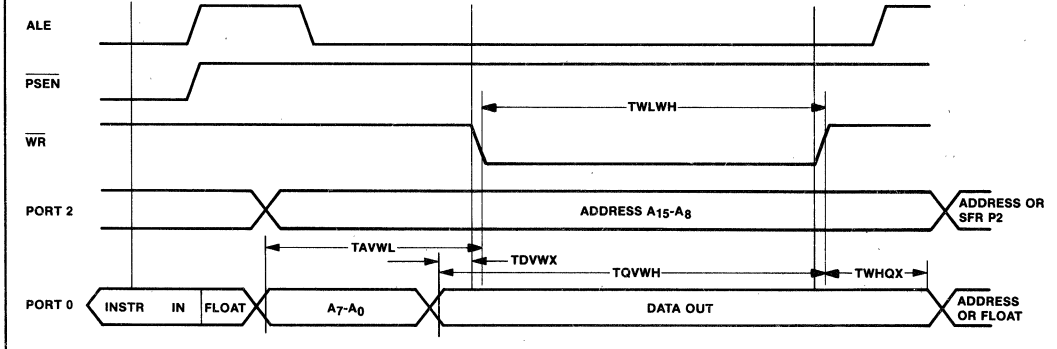
NOTE:
There are 2 to 8 ALE cycles per instruction. Clocks and state timing are shown on the timing diagram for reference purposes only. They are not accessible outside the package. TCY is the minimum instruction cycle time which consists of 12 oscillator clocks or two ALE cycles. Address setup and hold time from ALE are the same for data and program memory.



Program Memory Read Cycle



Data Memory Read Cycle



Data Memory Write Cycle

Table 1. 8051 Instruction Set Summary

Notes on instruction set and addressing modes:

- **Rn** — Register R7-R0 of the currently selected Register Bank.
- **data** — 8-bit internal data location's address. This could be an Internal Data Ram location (0-127) or a SFR (i.e. I/O port, control register, status register, etc. (128-255).
- **@Ri** — 8-bit Internal Data RAM location (0-255) addressed indirectly through register R1 or R0.
- **#data** — 8-bit constant included in instruction.
- **#data16** — 16-bit constant included in instruction.
- **addr16** — 16-bit destination address. Used by LCALL & LJMP. A branch can be anywhere within the 64K-byte Program Memory address space.
- **addr11** — 11-bit destination address. Used by ACALL & AJMP. The branch will be within the same 2K-byte page of program memory as the first byte of the following instruction.
- **rel** — Signed (two's complement) 8-bit offset byte. Used by SJMP and all conditional jumps. Range is -128 to +127 bytes relative to first byte of the following instruction.
- **bit** — Direct Addressed bit in Internal Data RAM or Special Function Register.
- **.** — New operation not provided by 8048/8049.

Interrupt Response Time: To finish execution of current instruction, respond to the interrupt request, push the PC and to vector to the first instruction of the interrupt service program requires 38 to 81 oscillator periods (3 to 7 μs @12MHz).

INSTRUCTIONS THAT AFFECT FLAG SETTINGS[1]

INSTRUCTION	FLAG C	OV	AC	INSTRUCTION	FLAG C	OV	AC
ADD	X	X	X	CLR C	O		
ADDC	X	X	X	CPL C	X		
SUBB	X	X	X	ANL C, bit	X		
MUL	O	X		ANL C,/bit	X		
DIV	O	X		ORL C, bit	X		
DA	X			ORL C,/bit	X		
RRC	X			MOV C, bit	X		
RLC	X			CJNE	X		
SETB C	1						

[1]Note that operations on SFR byte address 208 or bit addresses 209-215 (i.e. the PSW or bits in the PSW) will also affect flag settings.

Data Transfer

Mnemonic	Description	Bytes	Oscillator Periods
MOV A,Rn	Move register to A	1	12
*MOV A,data	Move direct byte to A	2	12
MOV A,@Ri	Move indirect RAM to A	1	12
MOV A,#data	Move immediate data to A	2	12
MOV Rn,A	Move A to register	1	12
*MOV Rn,data	Move direct byte to register	2	24
MOV Rn,#data	Move immediate data to register	2	12
*MOV data,A	Move A to direct byte	2	12
*MOV data,Rn	Move register to direct byte	2	24
*MOV data,data	Move direct byte to direct byte	3	24
*MOV data,@Ri	Move indirect RAM to direct byte	2	24
*MOV data,#data	Move immediate data to direct byte	3	24
MOV @Ri,A	Move A to indirect RAM	1	12
*MOV @Ri,data	Move direct byte to indirect RAM	2	24
MOV @Ri,#data	Move immediate data to indirect RAM	2	12
*MOV DPTR, #data16	Move 16-bit constant to Data Pointer	3	24
*MOV C,bit	Move direct bit to carry	2	12
*MOV bit,C	Move carry to direct bit	2	24
*MOVC A,@A+ DPTR	Move Program Memory byte addressed by A+DPTR to A	1	24
*MOVC A,@A+PC	Move Program Memory byte addressed by A+PC to A	1	24
MOVX A,@Ri	Move External Data (8-bit address) to A	1	24
*MOVX A,@DPTR	Move External Data (16-bit address) to A	1	24
MOVX @Ri,A	Move A to External Data (8-bit address)	1	24
*MOVX @DPTR,A	Move A to External Data (16-bit address)	1	24
*PUSH data	Move direct byte to stack and inc. SP	2	24
*POP data	Move direct byte from stack and dec. SP	2	24
XCH A,Rn	Exchange register with A	1	12
*XCH A,data	Exchange direct byte with A	2	12
XCH A,@Ri	Exchange indirect RAM with A	1	12
XCHD A,@Ri	Exchange indirect RAM's least sig nibble with A's LSN	1	12

Logic

Mnemonic	Description	Bytes	Oscillator Periods
ANL A,Rn	AND register to A	1	12
*ANL A,data	AND direct byte to A	2	12
ANL A,@Ri	AND indirect RAM to A	1	12
ANL A,#data	AND immediate data to A	2	12
*ANL data,A	AND A to direct byte	2	12
*ANL data,#data	AND immediate data to direct byte	3	24
*ANL C,bit	AND direct bit to carry	2	24
*ANL C,/bit	AND complement of direct bit to carry	2	24
ORL A,Rn	OR register to A	1	12
*ORL A,data	OR direct byte to A	2	12
ORL A,@Ri	OR indirect RAM to A	1	12
ORL A,#data	OR immediate data to A	2	12
*ORL data,A	OR A to direct byte	2	12
*ORL data,#data	OR immediate data to direct byte	3	24
*ORL C,bit	OR direct bit to carry	2	24
*ORL C,/bit	OR complement of direct bit to carry	2	24
XRL A,Rn	Exclusive-OR register to A	1	12
*XRL A,data	Exclusive-OR direct byte to A	2	12
XRL A,@Ri	Exclusive-OR indirect RAM to A	1	12
XRL A,#data	Exclusive-OR immediate data to A	2	12
*XRL data,A	Exclusive-OR A to direct byte	2	12
*XRL data,#data	Exclusive-OR immediate data to direct byte	3	24
*SETB C	Set carry	1	12
*SETB bit	Set direct bit	2	12
CLR A	Clear A	1	12
CLR C	Clear carry	1	12
*CLR bit	Clear direct bit	2	12
CPL A	Complement A	1	12
CPL C	Complement carry	1	12
*CPL bit	Complement direct bit	2	12
RL A	Rotate A Left	1	12
RLC A	Rotate A Left through carry	1	12
RR A	Rotate A Right	1	12
RRC A	Rotate A Right through carry	1	12
SWAP A	Rotate A left four (exchange nibbles within A)	1	12

Arithmetic

Mnemonic	Description	Bytes	Oscillator Periods
ADD A,Rn	Add register to A	1	12
*ADD A,data	Add direct byte to A	2	12
ADD A,@Ri	Add indirect RAM to A	1	12
ADD A,#data	Add immediate data to A	2	12
ADDC A,Rn	Add register and carry flag to A	1	12
*ADDC A,data	Add direct byte and carry flag to A	2	12
ADDC A,@Ri	Add indirect RAM and carry flag to A	1	12
ADDC A,#data	Add immediate data and carry flag to A	2	12
*SUBB A,Rn	Subtract register and carry flag from A	1	12
*SUBB A,data	Subtract direct byte and carry flag from A	2	12
*SUBB A,@Ri	Subtract indirect RAM and carry flag from A	1	12
*SUBB A,#data	Subtract immediate data and carry flag from A	2	12
INC A	Increment A	1	12
INC Rn	Increment register	1	12
*INC data	Increment direct byte	2	12
INC @Ri	Increment indirect RAM	1	12
DEC A	Decrement A	1	12
DEC Rn	Decrement register	1	12
*DEC data	Decrement direct byte	2	12
*DEC @Ri	Decrement indirect RAM	1	12
*INC DPTR	Increment Data Pointer	1	24
*MUL AB	Multiply A times B	1	48
*DIV AB	Divide A by B	1	48
DA A	Decimal add Adjust of A	1	12

Other

Mnemonic	Description	Bytes	Oscillator Periods
NOP	No Operation	1	12

Control Transfer (Branch)

Mnemonic	Description	Bytes	Oscillator Periods
AJMP addr 11	Absolute Jump	2	24
*LJMP addr16	Long Jump	3	24
*SJMP rel	Short Jump	2	24
*JMP @A+DPTR	Jump indirect relative to the DPTR	1	24
JZ rel	Jump if A is zero	2	24
JNZ rel	Jump if A is not zero	2	24
JC rel	Jump if carry is set	2	24
JNC rel	Jump if carry is not set	2	24
*JB bit,rel	Jump relative if direct bit is set	3	24
*JNB bit,rel	Jump relative if direct bit is not set	3	24
*JBC bit,rel	Jump relative if direct bit is set, then clear bit	3	24
*CJNE A,data,rel	Compare direct byte to A & Jump if not Eq. See Note a.	3	24
*CJNE A,#data,rel	Compare immed. to A & Jump if not Eq. See Note a.	3	24
*CJNE Rn,#data, rel	Compare immed. to reg & Jump if not Eq. See Note a.	3	24
*CNJE @Ri, #data,rel	Compare immed. to indirect RAM & Jump if not Eq. See Note a.	3	24
DJNZ Rn,rel	Decrement register & Jump if not zero	2	24
*DJNZ data,rel	Decrement direct byte & Jump if not zero	3	24

Note a) Set C if the first operand is less than the second operand; else clear

Control Transfer (Subroutine)

Mnemonic	Description	Bytes	Oscillator Periods
ACALL addr11	Absolute Subroutine Call	2	24
LCALL addr16	Long Subroutine Call	3	24
RET	Return from Subroutine Call	1	24
RETI	Return from Interrupt Call	1	24