CHAPTER ONE

MICROPROGRAMMING (CPE575)

1.0 Computer Overview

Definitions

A computer is defined by:

1. the ability to be programmed to operate on data without human intervention.
2. the ability to store and retrieve data.

Generally, it includes the peripheral devices for communicating with the outside world as well as programs that process data. The equipment is hardware while the program is software. Figure 1.1 is the block diagram of a typical computer.

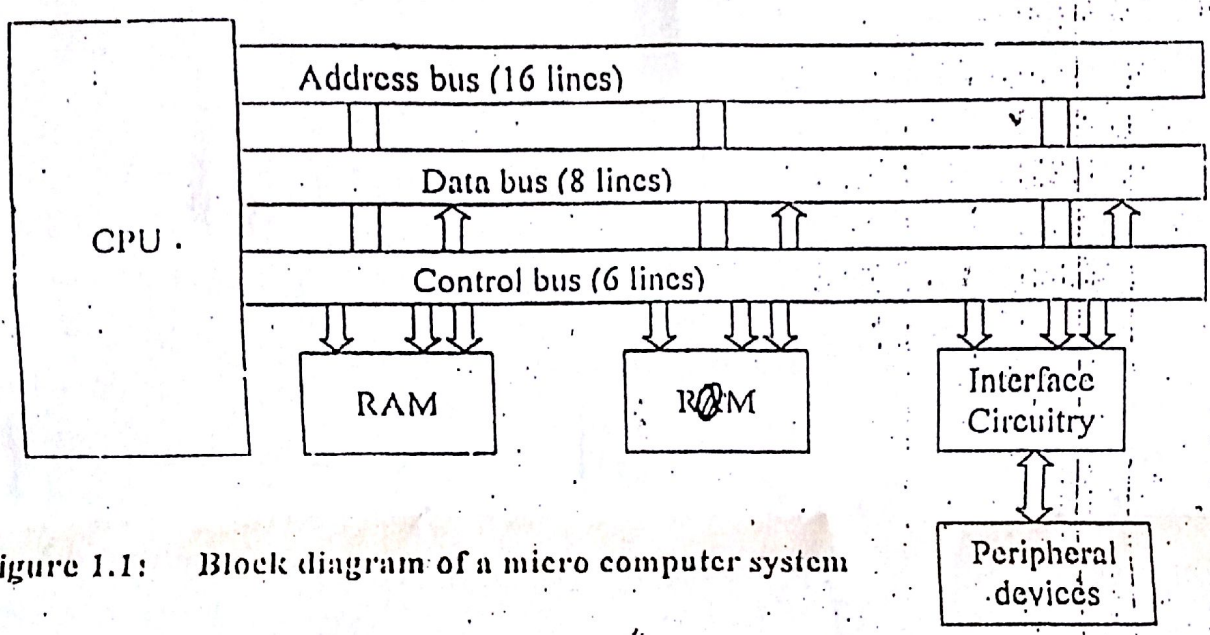


Figure 1.1: Block diagram of a micro computer system

The block diagram of Figure 1.1 is representative of all sizes of computers. It contains a Central Processing Unit (CPU) connected to random access memory (RAM) and read only memory (ROM) via the address bus, data bus and control bus. Interface circuits connect the system buses to peripheral devices.

1.1 THE CENTRAL PROCESSING UNIT

The CPU administers all activity in the system and performs all operations on data. The CPU consists of logic circuits that continuously perform two operations.

1. Fetching instructions and
2. Executing instructions

The CPU has the ability to understand and execute instructions based on a set of binary codes, each representing a simple operation. These instructions are usually:

1. Arithmetic (add, subtract, multiply, divide)

Generated by CamScanner

2. Logical (AND, OR, NOT, XOR, etc)
3. Data movement
4. Branch operations

All these instructions are represented by a set of binary codes called instruction set.

Figure 1.2 presents a typical view of a simplified internal structure of a CPU.

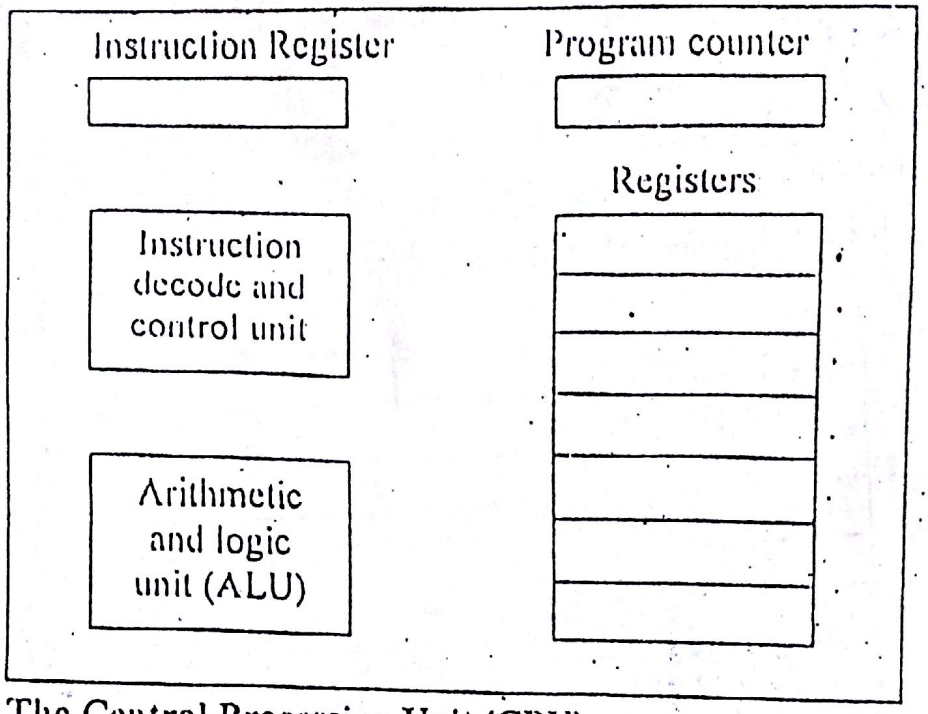


Figure 2: The Central Processing Unit (CPU)

The CPU consists of

1. Registers – for temporary storage of information
2. ALU – for performing operations on the information
3. Instruction decode and control unit – that determines the operation to perform and sets in motion the necessary actions to perform it.
4. The instruction register – that holds the binary code for each instruction as it is executed and
5. The program counter (PC) – that holds the memory address of the next instruction to be executed.

1.1.1 Fetching Instructions

Fetching instructions involves the following steps:

1. the contents of the program counter are placed on the address bus.
2. a READ control signal is activated.
3. data (the instruction opcode) are read from RAM (or memory) and placed on the data bus.

Generated by CamScanner

information must be loaded into main memory before the CPU accesses it. Mass storage devices are either 'online' or 'archival'. Online storage, usually on magnetic disk, is available to the CPU without human intervention upon request of a program while archival storage holds data that are rarely needed and require manual loading onto the system. Archival storage used to be held on magnetic tapes or disks but now optical disks, such as CD-ROM technology are used.

2. Human Interface Devices

These devices require human intervention for their use. The most common of these devices are video display unit/terminal (VDT) and printer. While printers are strictly output devices that generate hardcopy output, VDTs are two devices: they contain a keyboard for input and a CRT (cathode ray tube) for output. Other human interface devices include the joystick, light pen, mouse, microphone and loudspeaker.

3. Control/monitor Devices

These devices help computers to perform a variety of control-oriented tasks unceasingly without fatigue far beyond human capabilities. Applications such as temperature control of a building, home security, elevator control, home appliance control and even welding parts of an automobile are all made possible by using these devices.

Control devices are outputs or actuators, that can affect the world around them when supplied with a voltage or current (e.g. motors and relays). Monitoring devices are inputs or sensors, which are stimulated by heat, light, pressure, motion, etc and convert this energy to a voltage or current read by the computer (e.g. photo transistor. thermistors and switches). The interface circuitry converts the voltage or current to binary data or vice versa, and through software an orderly relationship between inputs and outputs is established.

These three groups of input/output devices are the most frequently used with microprocessors and microcontrollers.

1.4 Programs: Big and Small

The early days of computing witnessed the materials, manufacturing and maintenance costs of computer hardware far surpassing the software costs. Today, however, with mass produced large-scale integrated (LSI) chips, hardware costs are less dominant. It is the labor-intensive job writing, documenting, maintaining, updating and distributing software that constitutes the bulk of the expense in automating a process using computers.

There are different types of software. Figure 1.4 illustrates three levels of software between the user and the hardware of a computer system.

Generated by CamScanner

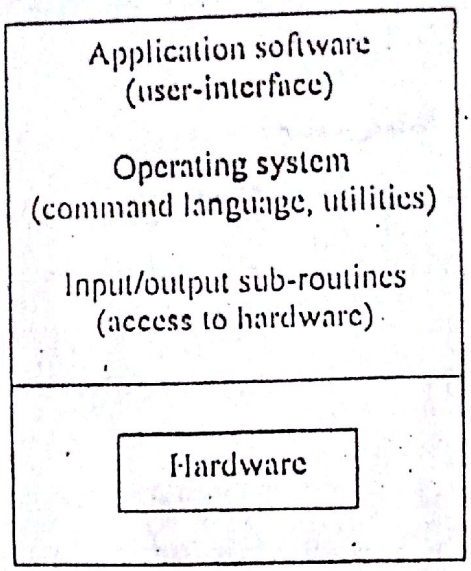


Figure 1.4: Levels of software

The three levels of software between the user and the hardware of a computer system are:

1. The application software
2. The operating system
3. The input/output subroutines

At the lowest level, the input/output subroutines directly manipulate the hardware of the system, reading characters from the keyboard, writing characters to the CRT, reading blocks of information from the disk, and so on. Since these subroutines are so intimately linked to the hardware, they are written by the hardware designers and are usually stored in ROM. They are the BIOS – basic input/output system on the desktop system, for example.

To provide close access to the system hardware for programmers, explicit entry and exit conditions are defined for the input/output subroutines. One only needs to initialize value CPU registers and call the subroutine. The action is carried out with results returned in CPU registers or left in system RAM. Apart from the input/output subroutines, the ROM contains start-up program that executes when the system is powered up or is reset manually by the operator. ROM is essential here since this program must exist upon power-up.

'House keeping' chores such as checking for options, initializing memory, performing diagnostic checks, etc, are all performed by the start-up program.

A bookshop loader routine reads the first tract (a small program) from the disk into RAM and passes control to it. This program then loads the RAM-resident portion of the operating system (a large program) from the disk and passes control to it; thus completing the start-up of the system.

Generated by CamScanner

The operating system is a large collection of programs that come with the computer system and provides the mechanism to access, manage, and effectively utilize the computer's resources. These abilities exist through the operating system's command language and utility programs which in turn facilitate the development of applications software. If the applications software is well designed, the user interacts with the computer with little or no knowledge of the operating system. Providing an effective, meaningful, and safe user interface is a prime objective in the design of applications software.

1.5 Micros, Minis and Mainframes

Using their size and power as a starting point, computers are classified as microcomputers, minicomputers or mainframe computers.

The key property of micro computers is the size and packaging of the CPU. It is contained within a single integrated circuit, the Microprocessor. Mini computers and mainframe computers, apart from being more complex in every architectural detail, have CPUs consisting of multiple ICs, ranging from several ICs (minicomputers) to several circuit boards of ICs (mainframes). This increased capacity is necessary to achieve the high speeds and computational power of larger computers.

Another feature separating micros from mini and mainframes is that microcomputers are single-user, single-task system, i.e., they interact with one user and they execute one program at a time. Minis and mainframes are multiuser, multitasking systems, i.e., they can accommodate many users and programs simultaneously. Simultaneous executing of programs is actually an illusion resulting from "time slicing" CPU resources. Multiprocessing systems, however, use multiple CPUs to execute tasks simultaneously.

1.5.1 Microprocessors Vs Microcontrollers

The distinguishing features between microprocessors and microcontrollers can be addressed from three perspectives.

1. **Hardware architecture**

Figure 1.5 shows a more detailed diagram of internal view of the microcomputer system

Generated by CamScanner

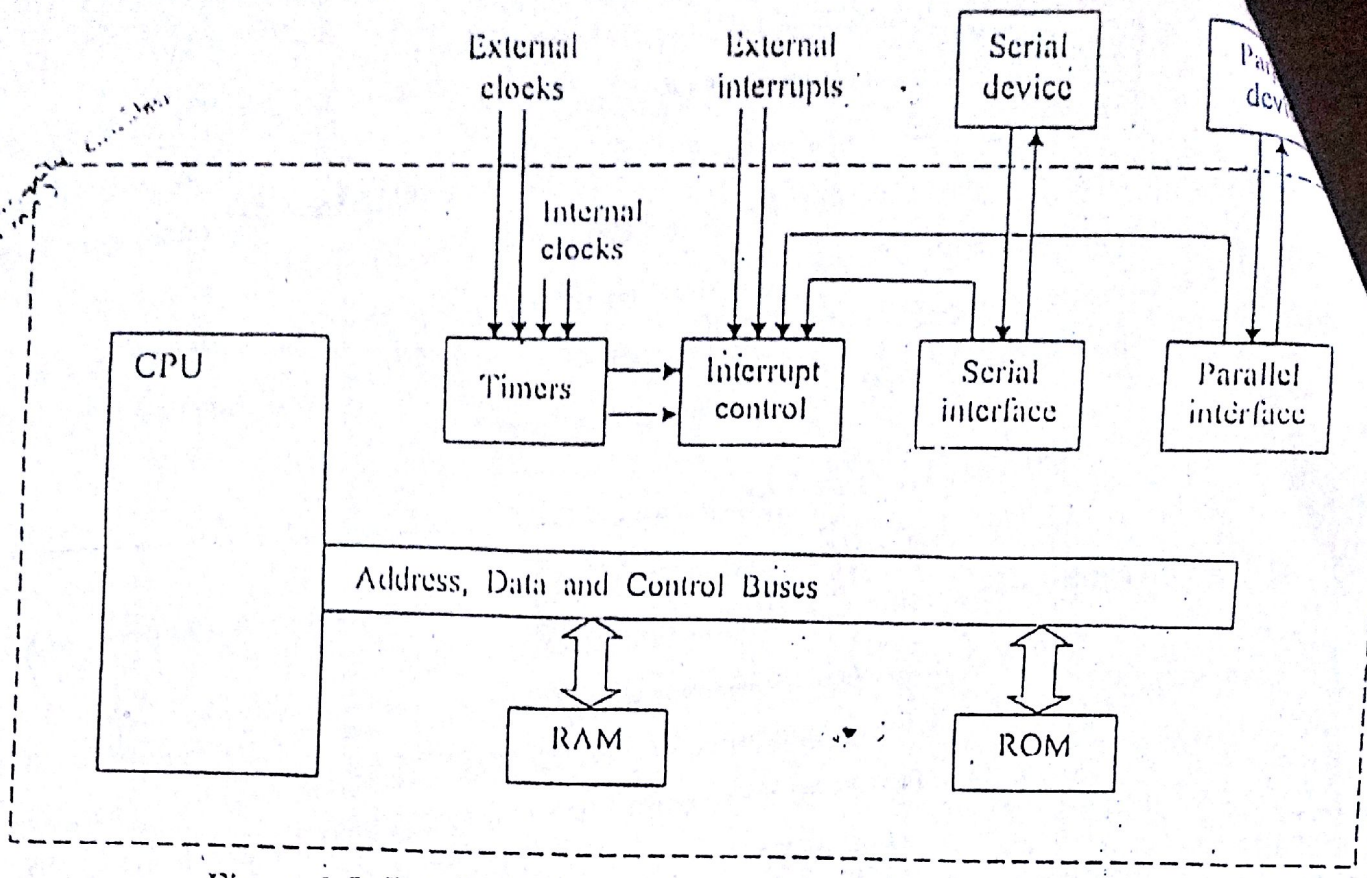


Figure 1.5: Detailed block diagram of a microcomputer system

The microprocessor is a single-chip while a microcontroller contains in a single IC a CPU and much of the remaining circuitry of a complete microcomputer system. Microcontrollers also include RAM, ROM, serial interface, a parallel interface, timer, interrupt scheduling circuitry, all within the same IC.

An important feature of microcontrollers is the built-in interrupt systems. As control-oriented devices, they are often called upon to respond to external stimuli (interrupts) in real time. They must perform fast context switching, suspending pre process while executing another in response to an 'event'. The opening of a microwave oven's door is an example of an event that might cause an interrupt in a microcontroller-based product.

Whereas a microprocessor requires external components to implement an interrupt scheme, a microcontroller's on-chip circuitry includes all the interrupt handling circuitry necessary.

2. Applications

Microprocessors are mostly used as the CPU in microcomputer systems. This function is what they are designed for and this is where their strength lies. Microcontrollers however, are found in small, minimum component design performing control-oriented activities. A microcontroller can aid in reducing overall component count in a circuit that would normally require dozens or hundreds of digital ICs. A microcontroller only requires a small number of support components and a control program in ROM. Microcontrollers are suited to control of I/O device, in design

Generated by CamScanner

requiring a minimum component count whereas microprocessors are suited to "processing" information in computer systems.

3. Instruction set features

Microprocessor instruction sets are processing intensive implying that they have powerful addressing modes with instruction catering for operations on large volumes of data. Their instructions operate on nibbles, bytes, words or even double words. Addressing modes provide access to large arrays of data, using address pointers and offsets. Auto-increment and auto-decrement modes simplify stepping through arrays on bytes, word or double-word boundaries. The bit goes on.

Microcontrollers, on the other hand, have instruction sets catering for the control of inputs and outputs. The interface to many inputs and outputs uses a single bit. For example, a motor may be turned on and off by a solenoid energized by a 1-bit output port. Microcontrollers have instructions to set and clear individual bits and perform other bit-oriented operations such as logically ANDing, ORing, or EXORing bits', jumping if a bit is set or clear and so on. This feature is rarely present in microprocessors, which, are usually designed to operate on bytes or larger units of data.

1.5.2 I/O PORT STRUCTURE

The internal circuitry for the port pins is as shown in Figure 1.6.

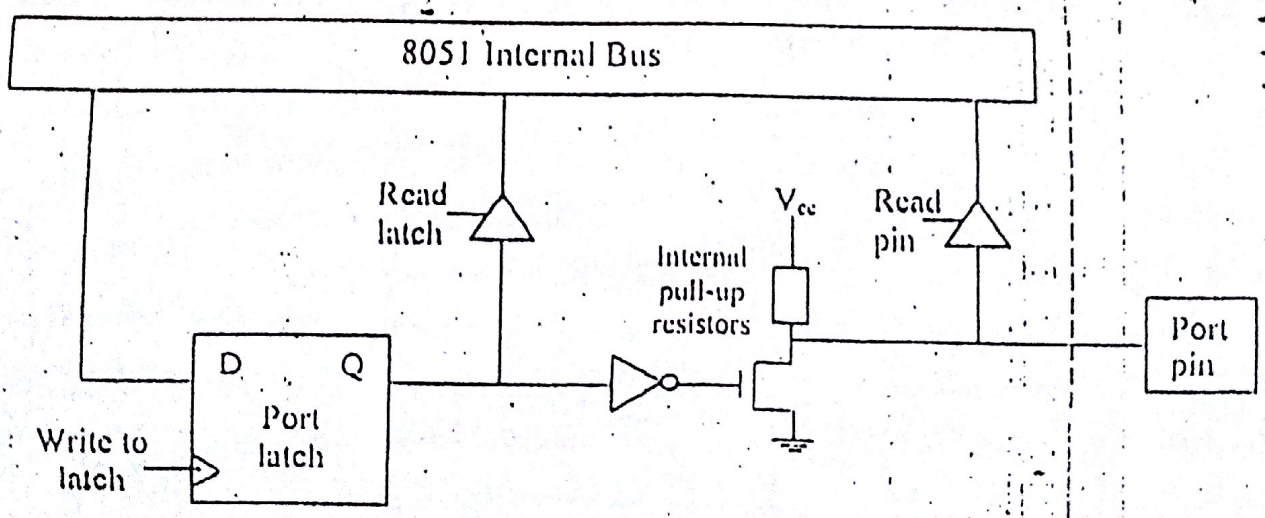


Figure 1.6: Circuitry for I/O ports

Writing to a port pin loads data into a port latch that drives a field-effect transistor connected to the port pin. The drive capability is four low-power schottky TTL loads for ports 1, 2 and 3, and eight loads for port 0. The pull up resistor is absent on port 0 when functioning as

the external address/data bus. An external pull-up resistor may be needed depending on characteristics of the device driven by the port pin.

There is both a 'read latch' and 'read pin' capability. Instructions that require modify operation (e.g. CPL P1.5) read the latch to avoid misinterpreting the voltage level event. The pin is heavily loaded (e.g. when driving the base of a transistor). Instructions input a port bit (e.g. MOV C, P1.5) read the pin. The port latch must contain a 1, in this otherwise the FET driver is ON and pulls the output low. A system reset sets all port latches, port pins may be used as inputs without explicitly setting the port latches. If however, a port latch is cleared (e.g. CLR P1.5) then it cannot function subsequently as an input unless is set first (e.g. SETB P1.5).

When the alternate functions of port 0, 2 and 3 (not shown in the diagram for I/O circuitry, Figure 1.6) is in effect, the output drivers are switched to an internal address (port 2), address/data (port 0), or control (port 3) signal as appropriate.

1.5.3 TIMING AND THE MACHINE CYCLE

The 8051's on-chip oscillator is driven by an external quartz crystal through pins 18 and 19. This crystal has a typical frequency 12MHz. These oscillator clock cycles form the basis of the 8051's timing and synchronization.

With the oscillator as reference, the 8051 requires two clock cycles to perform a single discrete operation, which is either fetching an instruction, decoding, or executing it. This duration of two clock cycles is also called a state. Therefore, in order to fully process an instruction, the 8051 would generally require 6 such states, or 12 clock cycles since it would have to first fetch and decode the instruction before it goes to execute it. This duration of six states is also known as one 'machine cycle'. More complex instructions would take more than one machine cycle to be carried out. This number ranges from one to four machine cycles. The Figure 1.7 shows the relationship between clock cycles (P), states (S) and a machine cycle.

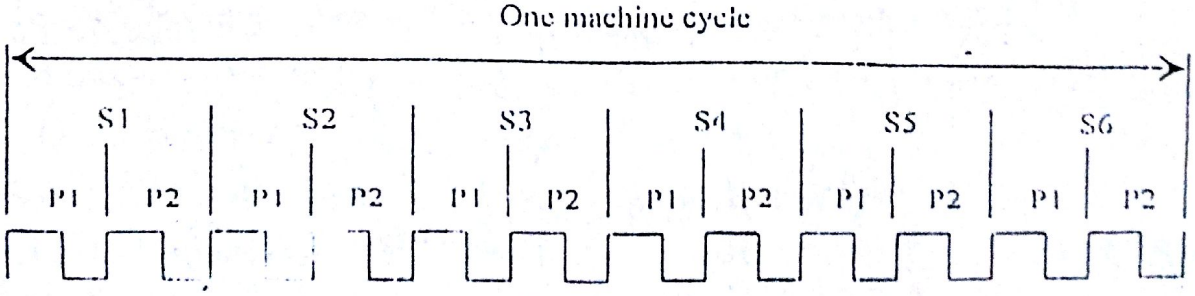


Figure ... ationship between oscillator clock cycles, states and machine cycle

* 8051 requires two clock cycles (P1 & P2) to perform a single discrete operation
* Duration of two clock cycles is called a state. (P2 = S1)
* It would require 6 states or 12 clock cycles to process an instruction
* The duration of six (6) states is called one (1) machine cycle

Generated by CamScanner

Typically, the 8051's on-chip oscillator, f_{osc} is driven by a 12MHz crystal, so the period of one clock pulse,

$$T_{clock} = \frac{1}{f_{osc}} = \frac{1}{12MHz} = 83.33ns$$

One machine cycle consists of 12 such clock pulses, hence its duration is 83.33ns x 12 = 1ms

1.5.4 MEMORY ORGANIZATION

Most microprocessors implement a shared memory space for data and programs. Programs are usually stored on disk and loaded into RAM for execution. Thus, both data and programs reside in the system RAM. Microcontrollers are rarely used as the CPU in computer systems. Instead they are employed as the central component in control-oriented designs. There is limited memory and there is no drive or disk operating system. The control program must reside in ROM.

The 8051 implements a separate memory space for programs (code) and data. The internal memory consists of on-chip ROM (8051/8052) and on-chip data RAM. The on-chip RAM contains a rich management of general purpose storage, bit-addressable storage, register banks and special function registers.

Two notable features are:

1. The registers and input/output ports are memory mapped and accessible like any other memory location.
2. The stack resides within the internal RAM, rather than in external RAM as typical of microprocessors.

Any location in the general purpose RAM can be accessed freely using the direct or indirect addressing modes. MCS-51 is a family of which 8051 was the first commercially provided member. Table 1.1 compares the family.

Table1.1: MCS-51 Family

Port Number	On-chip Code Memory	On-chip Data Memory	Timers
8051	4K ROM	128 bytes	2
8031	0K	128 bytes	2
8751	4K EPROM	128 bytes	2
8052	8K ROM	256 bytes	2
8032	0K	256 bytes	2
8752	8K EPROM	256 bytes	2

The term 8051 refers to the MCS-51 family of microcontrollers.

Generated by CamScanner

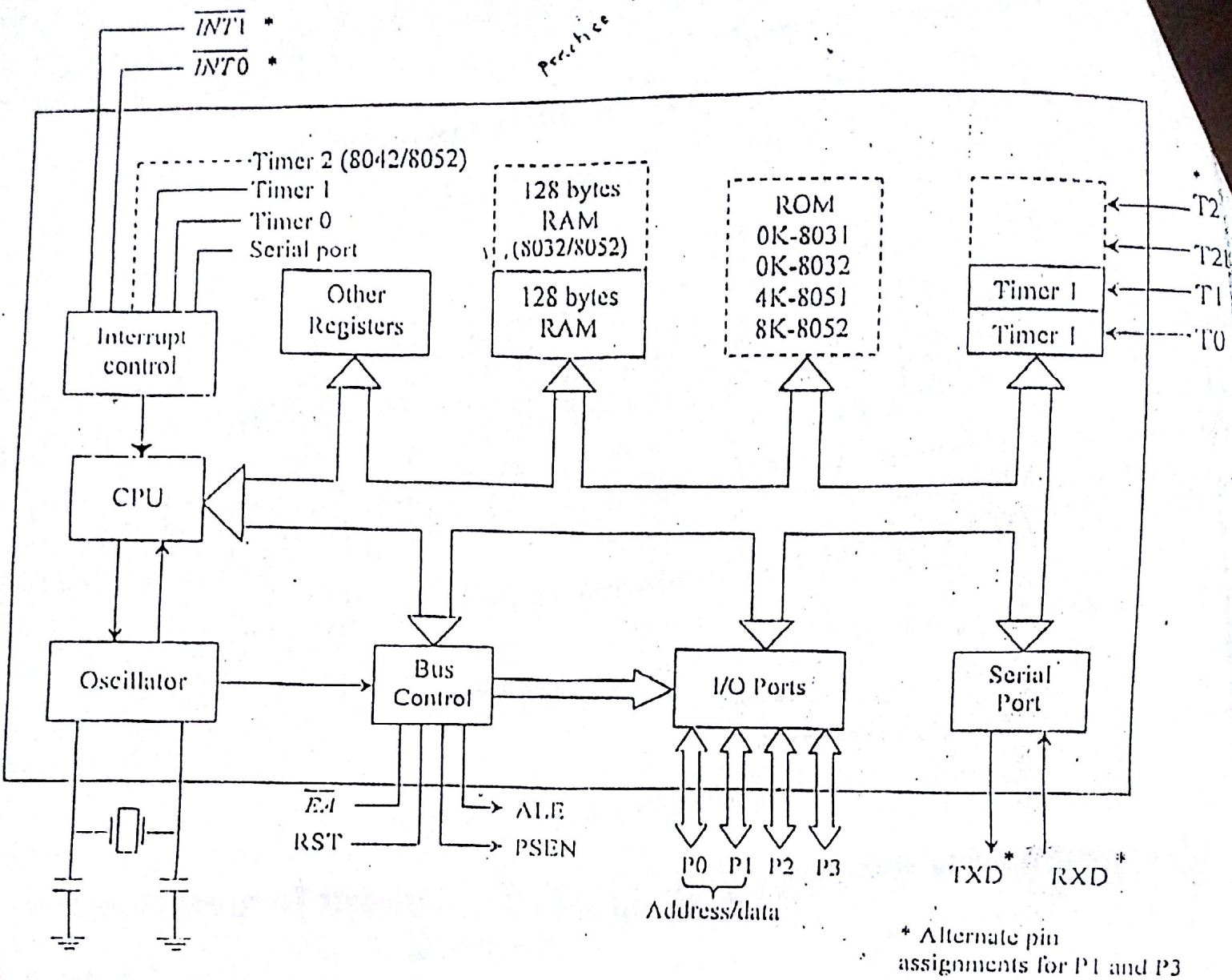


Figure 1. 8: 8051 MCS-51 Block Diagram

The figure 8 and Figure 9 show the 8051 hardware architecture from an external perspective, i.e., the pinouts.

The 8051 has 40 pins out of which 32 function as I/O port lines. However 24 of these are dual purpose (26 on the 8032/8052). Each can operate as I/O or as a control line or part of the address or data bus.

Designs requiring a minimum of external memory or other external components use these ports for general purpose I/O. The eight lines in each port can be treated as a unit in interfacing to parallel devices such as printer, digital-to-analog converters and so on. Each line can also operate independently in interfacing to single-bit devices such as switches, CEDS, transistors, solenoids, motors and loudspeakers.

Generated by CamScanner

(I) Port 0

Port 0 is a dual purpose port on pins 32 – 39 of the 8057 IC. In minimum component design, it is used as a general purpose I/O port. For larger designs with external memory, it becomes a multiplexed address and data bus.

(II) Port 1

Port 1 is a dedicated I/O port on pins 1- 8. the pin designated as P1.0, P1.1, P1.2, etc, are available for interfacing to external devices as required. No alternate functions are assigned for port 1 pins. They are used solely for interfacing to external devices.

(III) Port 2

Port 2 (pins) 21-28) is a dual-purpose port serving by general purpose I/O, or as the high byte of the address bus for design with external code memory or more than 256 bytes of external data memory.

(IV) Port 3

Port 3 is a dual-purpose port on pins 10-17. Apart of being used as general-purpose I/O, these pins are multifunctional with each having an alternate purpose related to special features of the 8051. The alternate purpose of the port 3 and port 1 pin is summarized in Table 1.2.

Table 1.2: Alternate pin functions for port pins

	Bit	Name	Bit address	Alternate function
	P3.0	RXD	B0H	Receive data for serial port
	P3.1	TXD	B1H	Transmit data for serial port
	P3.2	$\overline{INT0}$	B2H	External interrupt 0
	P3.3	$\overline{INT1}$	B3H	External interrupt 1
	P3.4	T0	B4H	Timer/counter 0 external input
	P3.5	T1	B5H	Timer/counter 1 external input
	P3.6	$\overline{WR}$	B6H	External data memory write strobe
	P3.7	$\overline{RD}$	B7H	External data memory read strobe
8032/8052 only	P1.0	T2	90H	Timer/counter 2 external input
	P1.1	T2EX	91H	Timer/counter 2 capture/reload

(V) $\overline{PSEN}$ (Program store enable)

The 8051 has four dedicated bus control signals. Program store enable ($\overline{PSEN}$) in an output signal on pin 29. it is a control signal that enables external program (code) memory. It usually connects to an EPROM Output Enable ($\overline{OE}$) pin to permit reading of program bytes. The $\overline{PSEN}$ signal pulses low during the fifth stage of an instruction which is stored in external

Generated by CamScanner

program memory. The binary code of a program (OPCode) are read from EPROM, t
the data bus, and are catched into the 8051s instruction register for decoding. When ex
program from internal ROM, $\overline{PSEN}$ remains in the inactive (high) state.

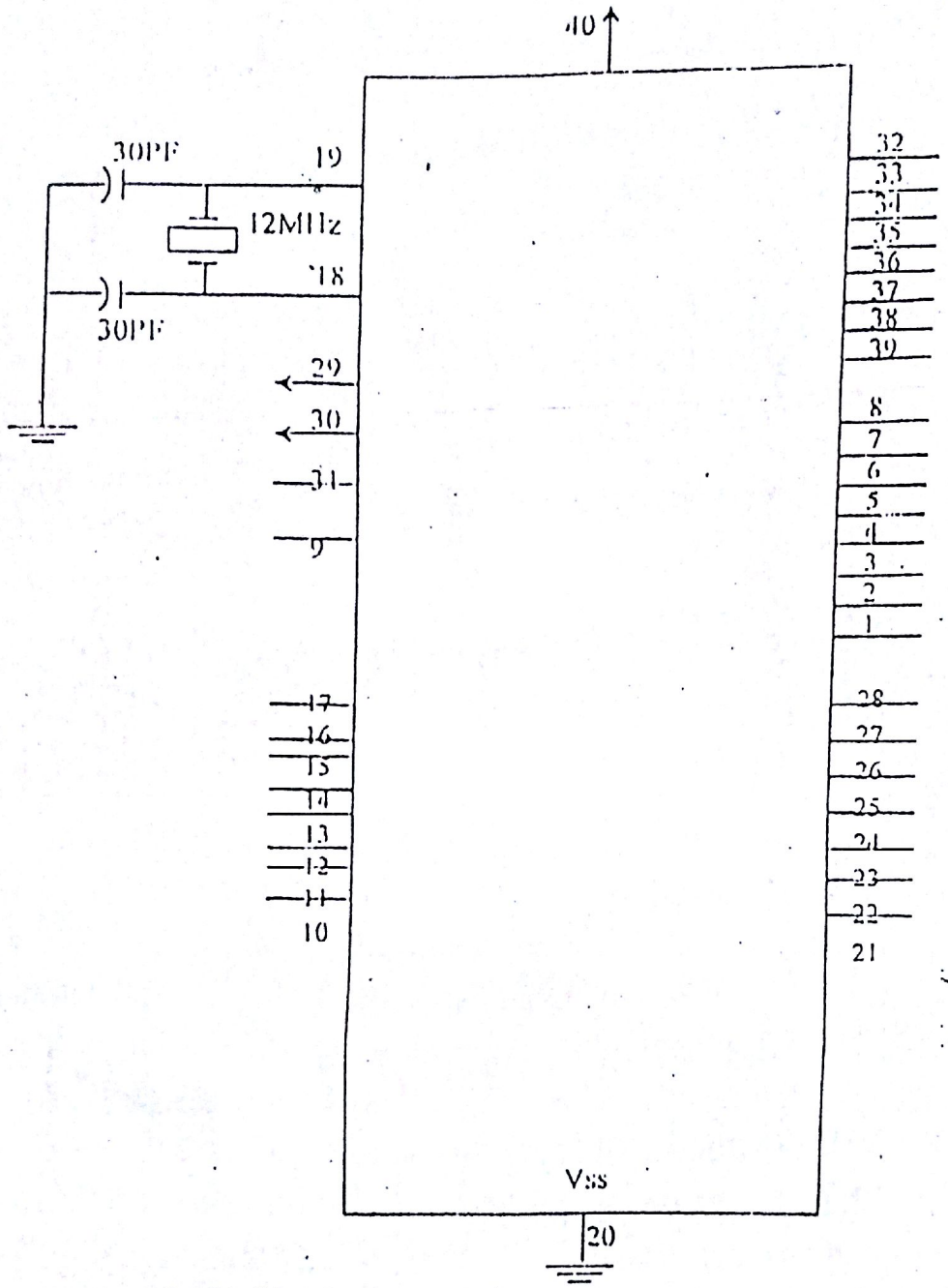


Figure 1.9: 8051 Pin Assignments

The ALE output signal on pin 30 is used by the 8051 for demultiplexing the address and data bus. When port O is used in alternate mode, on the data bus and low byte of the address bus, ALE in the signal that catches the address into an external register during the first half of a memory cycle. This done, the port lines are then available for data input or output during the second half of the memory cycle, when the data transfer takes place.

The ALE signal pulses at $^1/_6^{th}$ the on-chip oscillator frequency can be used on a general purpose clock for the rest of the system. If the 8051 is clocked from a 12MHz crystal, the ALE signal oscillates at 2MHz. the only exception is during the MOVx instruction, when one ALE

Generated by CamScanner

pulse is missed. This pin is also used for the programme input pulse for EPRom versions of 8051.

(VI) $\overline{EA}$ (External Access)

The $\overline{EA}$ input signal on pin 31 in generally tied high (+5V) or low (ground). If high, the 8051/8052 executes programs from internal ROM when executing in the lower 4K/8K of memory. If low, programs execute from external memory only (and $\overline{PSEN}$ pulses low accordingly).

$\overline{EA}$ must be tied low for 8031/8032 ICs, since there is no on-chip program memory. If $\overline{EA}$ is tied low on an 8051/8052, internal ROM is disabled and programs execute from external EPROM. The EPROM versions of the 8051 also use the $\overline{EA}$ line for the +21 volt supply (Vpp) for programming the internal EPRom.

(VII) RST (Reset)

The RST input on pin 9 is the master reset for the 8051. When the signal is brought high for at least two machine cycles, the 8051 internal registers are loaded with appropriate values for an orderly system start-up. For normal operation, RST is low.

1.5.5 ON-CHIP OSCILLATOR INPUTS

The 8051 features an on-chip oscillator that is typically driven by a crystal connected to pins 18 and 19. Stabilizing capacitors are also required as shown in the sketch for the 8051 pin outs.

The nominal crystal frequency is 12MHz for most ICs in the MCS-51 family although the 80C 51 BH-1 can operate with crystal frequencies or up to 16MHz. the on-chip oscillator need not be driven by a crystal. As shown in Figure1.10 a TTL clock source can be connected to XTAL1and XTAL2.

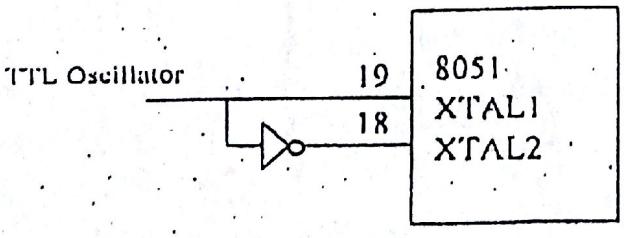


Figure 1.10: Driving the 8051 from a TTL Oscillator

1.5.6 POWER CONNECTIONS

The 8051 operates from a single +5 volt supply the Vcc connection is on pin 40, and Vss (ground).

Generated by CamScanner

1.6 8051 Memory

The 8051 has three very general types of memory. To effectively program the 8 necessary to have a basic understanding of these memory types. The memory types shown in Figure 1. 11 and include On-Chip Memory, External Code Memory, and Ex RAM.

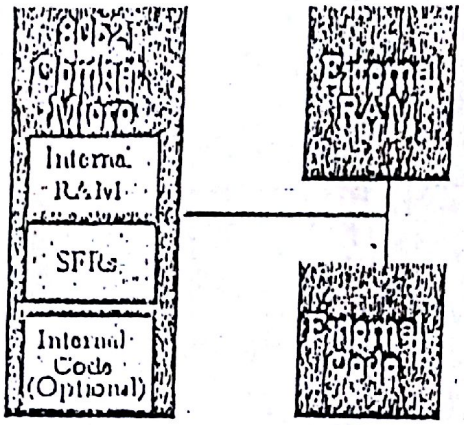


Figure 1.11: 8051 Memory Types

(I) On-Chip Memory

This refers to any memory (Code, RAM, or other) that physically exists on the microcontroller itself. On-chip memory can be of several types, but we'll get into that shortly.

(II) External Code Memory

This is code (or program) memory that resides off-chip. This is often in the form of an external EPROM.

(III) External RAM

This is RAM memory that resides off-chip. This is often in the form of standard static RAM or flash RAM.

1.6.1 Code Memory

Code memory is the memory that holds the actual 8051 program that is to be run. This memory is limited to 64K and comes in many shapes and sizes: Code memory may be found on-chip, either burned into the microcontroller as ROM or EPROM. Code may also be stored completely off-chip in an external ROM or, more commonly, an external EPROM. Flash RAM is also another popular method of storing a program. Various combinations of these memory types may also be used--that is to say, it is possible to have 4K of code memory on-chip and 64k of code memory off-chip in an EPROM.

Generated by CamScanner

When the program is stored on-chip the 64K maximum is often reduced to 4k, 8k, or 16k. This varies depending on the version of the chip that is being used. Each version offers specific capabilities and one of the distinguishing factors from chip to chip is how much ROM/EPROM space the chip has. However, code memory is most commonly implemented as off-chip EPROM. This is especially true in low-cost development systems and in systems developed by students.

1.6.2 External RAM

The 8051 also supports External RAM. Which is any random access memory which is found off-chip. Since the memory is off-chip it is not as flexible in terms of accessing, and is also slower. For example, to increment an Internal RAM location by 1 requires only 1 instruction and 1 instruction cycle. To increment a 1-byte value stored in External RAM requires 4 instructions and 7 instruction cycles, implying in this case ,that external memory is 7 times slower.

However while Internal RAM is limited to 128 bytes (256 bytes with an 8052), the 8051 supports External RAM up to 64K. (65536)

1.6.3 On-Chip Memory

On-chip memory is one of two types: Internal RAM and Special Function Register (SFR) memory.

Generated by CamScanner

CHAPTER TWO

8051 MEMORY AND PROGRAMMING MODEL

2.1 8051 INTERNAL RAM STRUCTURE

The layout of the 8051's internal memory is presented in the memory map of Figure 2.1 and Figure 2.2. The internal data memory space is divided between internal RAM (00H - 7FH) and special function registers (80H - FFH). The internal data memory has the range from 00H - FFH (256 bytes), out of which 00H - 7FH is for general data while 80H - FFH is mostly for specific purposes and not for general data. Hence, 00H - 7FH is considered to be internal RAM.

The internal RAM is further subdivided into register banks (00H - 1FH), bit addressable RAM (20H - 2FH), and general-purpose RAM (30H - 7FH).

General purpose RAM is from addresses 30H to 7FH. Locations 00H – 2FH can similarly be used but they also have other purposes.

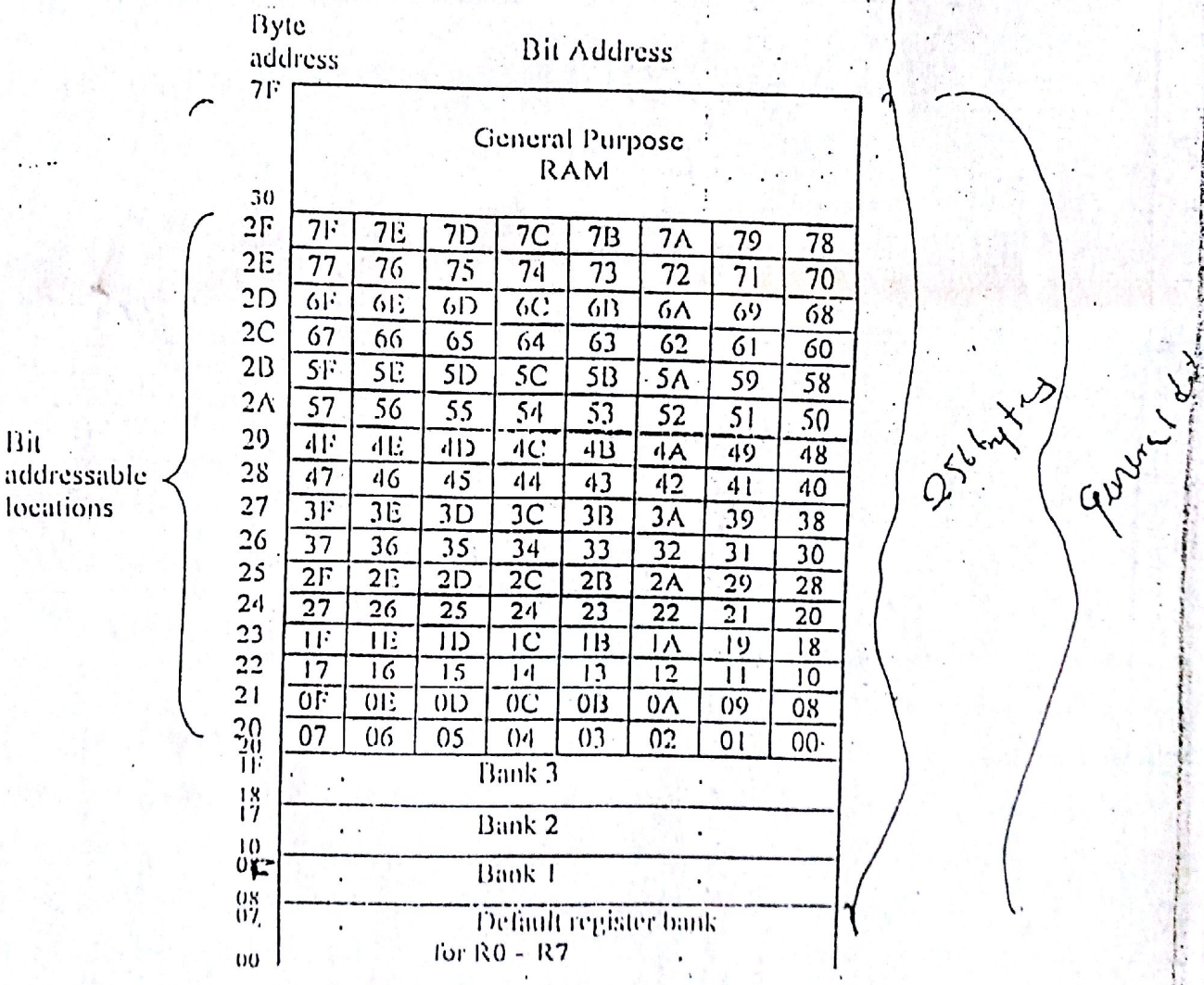


Figure 2.1: Internal 128 RAM structure of the 8051.

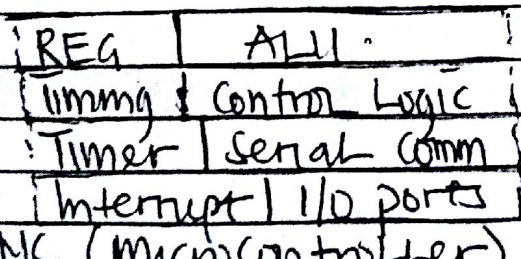
Generated by CamScanner

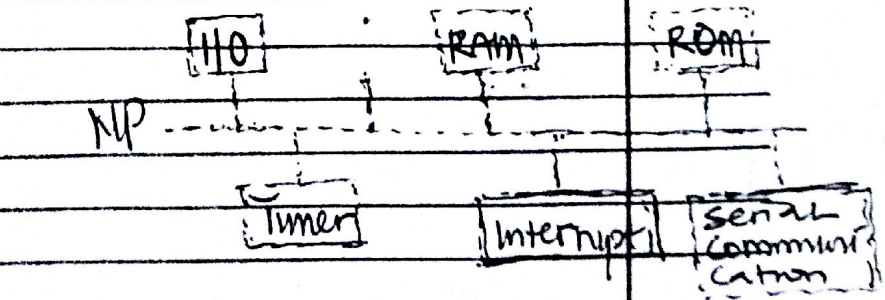
8051 FAMILY MICROCONTROLLER.

When the whole processing unit is fabricated on a single integrated socket or circuit it is called microprocessor. Inside the microprocessor we have (registers, ALU, decoders, timing and control Unit etc). While a microcontroller contains in a single integrated circuit a CPU and much of the remaining circuitry of a complete microcomputer system. Microcontrollers also include RAM, ROM, serial interface, parallel interface, timer, interrupt scheduling circuitry, all within the same integrated circuit IC

Difference Between Microcontroller and Microprocessor

	MICROCONTROLLER	MICROPROCESSOR
1)	ROM is inbuilt in microcontroller	ROM (code memory) outside of processor.
2)	RAM is inbuilt	RAM is External
3)	Timer is inbuilt	Timer is external
4)	I/O ports are inbuilt	ports (I/O) are external
5)	Serial Communication is inbuilt	Serial Communication hardware is external
6)	Interrupt Logic is inbuilt	Interrupt Logic is External.





MC (Microcontroller).

The microcontroller is smaller

This is bcoz all things are external.

In 1971 ÷ The first microprocessor with number [4004] was manufactured containing 4-Bits i.e 4-bit data can be executed at a time. Reg, ALU, decoder (internal)

Generated by CamScanner

RAM, ROM, timer etc where external because electronic companies where not too advanced to fabricate all components into one CPU.

In 1981 - First 8-bit microcontroller. The 8051 based microcontroller was manufactured. The batch number was MCS-51 (microcontroller system 51).
Then Intel allowed other companies to produce microcontroller but with the 8051 architecture and there must be compactibility with applications.
Additional frame work was allowed on the micro controller MC.
Company X could decide to work on improving the speed of the microcontroller (operating frequency) say 24MHz and Intel is 16MHz.
Company Y could decide to improve on CMOS technology for lower power (Less power) but Intel could use N-Mos technology.
Few companies that manufacture these microcontrollers are
INTEL, ATMEL, PHILIPS, SIEMENS, DALLAS SEMICONDUCTOR.

Some Families of Microcontroller
We have 8051, 8052, 8031 microcontrollers. Programs written on any of them runs on all but have slight variations

	RAM	ROM	TIMERS	SERIAL COMM	INTERRUPT
8051	128 Bytes	4KB	2 Inbuilt timers	1	6
8052	256 Bytes	8KB	3 Inbuilt timers	1	8
8031	128 Bytes	0KB (ROMLess)	2 Inbuilt timers	1	6

Timer is used to generate delays.

Generated by CamScanner

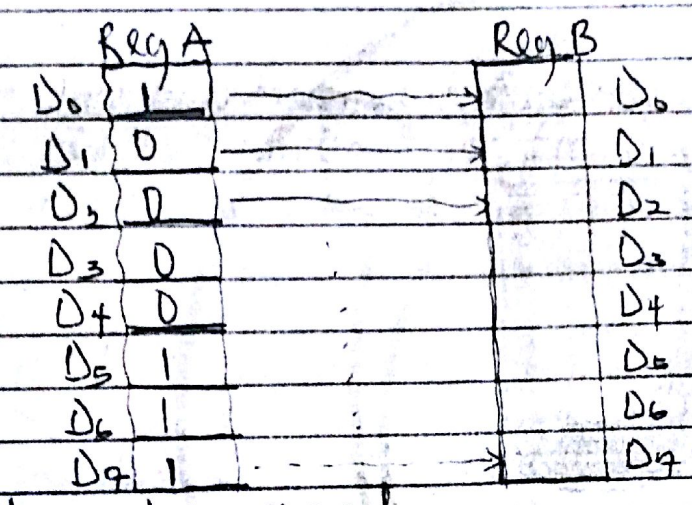
Write on both sides of the paper

Purpose of Serial Communication.

Suppose we want to transfer the value data from Reg A to Reg B. we have two approches parallel and serial communication.

In parallel transfer

In parallel transfer you use one clock cycles and 8 wires



Input data stored in Reg A (1,0,0,0,0,1,1,1)

In Serial Communication

In Case of serial communication one bit is transferred at a time with a single wire but with 8 clock cycles from Reg A to Reg B. The advantage is only one wire at a time and cost will be less but the disadvantages you require 8 clocks cycle so time taken is more.

Batch types.

AT 89C51

ATMEL

Flash memory (ROM)

CMOS (Less power)

AT 87C51

UV-EPROM

CMOS.

Generated by CamScanner

ROM

OTP — UV-EPROM — EEPROM — Flash Memory.

One time Programmable

OTP (one time programmable). Once it is programmed. it content cannot be erased.

UV-EPROM (Erasable programmable read only only memory) Once we program this memory it content can be erased. How we can erase this memory is by exposing it to Ultra violent rays.

EEPROM (Electrical erasable Programmable read only memory)
This can be erased through electrical pulse.

MACHINE CYCLE AND RESET

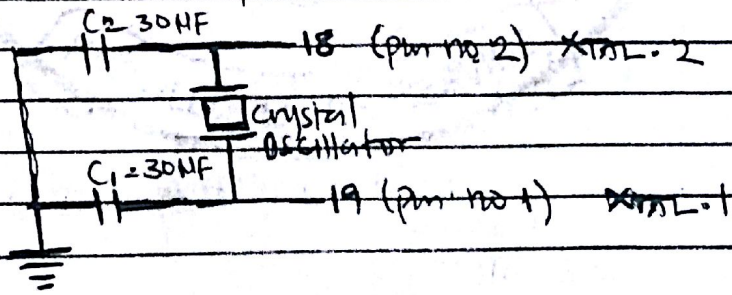
Suppose our microcontroller is executing some instruction. Then what the microcontroller will do is to split the execution of the instruction into small tasks (modules)
Firstly, it will load the memory address of instruction in code memory into program counter (PC).
Secondly, the content of the program counter will be loaded into address bus, the content of that particular address will be fetched by the microcontroller, that content which is the opcode will enter into the decoder. The decoder Unit will decode the instruction and then instruct the various components of the controller to complete the task. The task is the addition of two numbers, ~~Say numbers~~ Once the decoder decodes that we have to add two numbers, i.e One number is in the accumulator and the other in a register both of these numbers transferred from their respective place to arithmetic & Logical Unit (ALU).

Generated by CamScanner

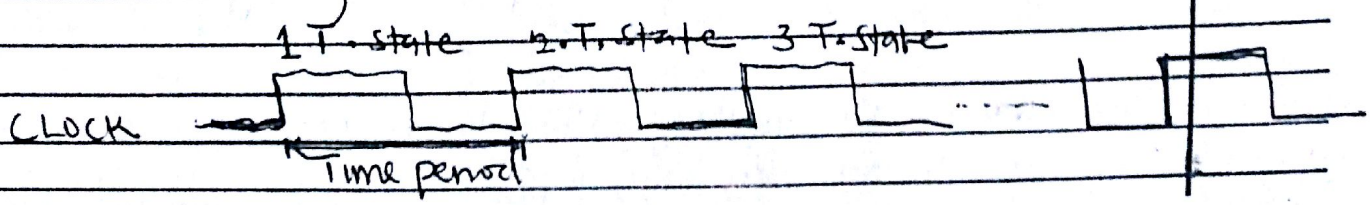
Then the ALU session of "Adder" is activated. Inside ALU you have Logical Unit, Subtractor, Addition, multiplication division. The addition is performed and the result is stored back into the accumulator.

To complete one machine instruction. The controller divides the instruction into modules and then it will execute the whole instruction. The time required to complete one machine instruction is called the "machine cycle". Since there are various electronic session inside a microcontroller so all of these session should operate in a clock cycle otherwise there will be mismatch, so we require a clock signal for single addition of electronics inside the microcontroller. The clock is provided by one external crystal oscillator that has to be connected.

Our 8051 microcontroller is a 40 pin integrated circuit IC two pins are used for crystal Osillator (pin 18 & 19). These pins are designated as XTAL.1 & XTAL 2 (Crystal oscillator pin number 1 and crystal oscillator pin number 2). The two pins will be connected to crystal oscillator. The maximum oscillator that we can connect to MC 8051 is 24mHz Max freq of XTAL is 24MHz



The capacitor is connected C1 & C2 to one end of pin 18 and the other to pin 19. The end of the capacitors are connected to ground. C1 and C2 are used for stabilizing the clock. The crystal Oscillator will use the internal hardware of the 8051 MC and it will generate the clock. The clock is generated like this.



Generated by CamScanner

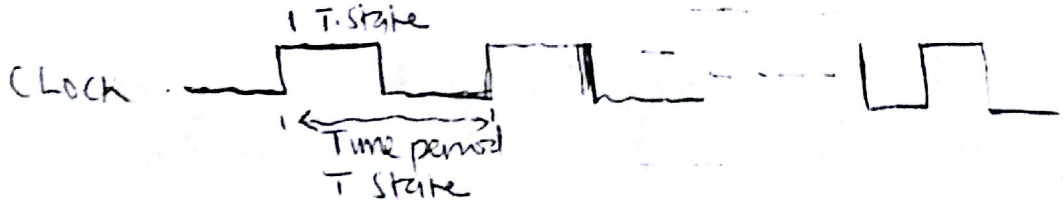
One time period is called T. State
In 8051 Microcontroller one machine cycle consists of 12 T. state

1 Machine cycle = 12 T. State

There are many instruction which can be be executed in 1 machine cycle, there are few instructions which require 2 machine cycle, then few instructions which require 3 machine cycle. To know which instruction will have 1mc, 2mc, 3mc you will have to go to the data set (Instruction set) to confirm. There you will find the name of the instruction, no of bytes required m/c required for the execution
Now depending on the crystal frequency we will get the time period
Suppose we have a frequency of

e.g F_{XTAL} = 12MHz

The clock period will be generated like this



Time period for 1 T State

$$T = \frac{1}{F_{XTAL}} = \frac{1}{12} \times 10^{-6} \text{ sec}$$

Time required for 1 mc = $12 \times \frac{1}{12} \times 10^{-6}$ sec

$= 10^{-6}$ sec = 1μs (micro sec)

i.e if we have an instruction that can be executed in 1 m/c the microcontroller will execute in

Generated by CamScanner

Write on both sides of the paper

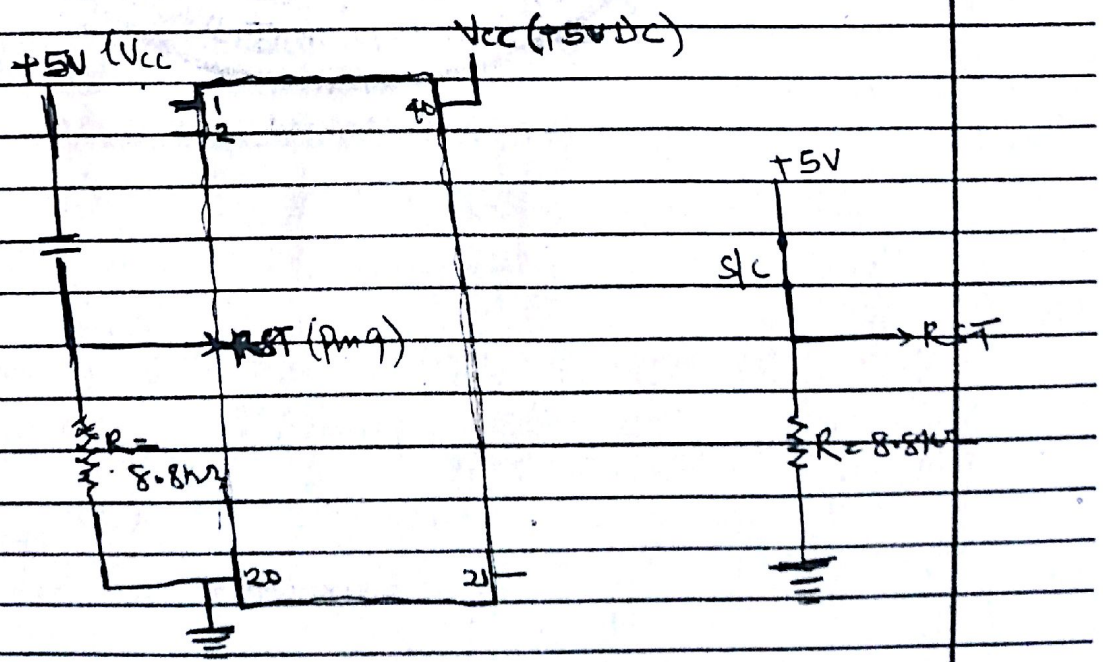
1 m/c = 1μs
2 m/c = 2μs
3 m/c = 3μs.

In 8051 microcontroller there is a pin called the reset pin. Whenever, a microcontroller is put ON the value of the program counter register will be 0000H. The microcontroller fetches the content of the address memory from the Memory location, the address memory here is 0000H.
ORG 0000H ensures that the first byte of program must start at location 0000H. Now suppose we start our program from 0000H and the system hangs or stops we can reset the Microcontrol to restart the execution of the program from 0000H.

RST = '0' (ground "GND) for normal operation of microcontroller

RST = '1' (Vcc) Then Microcontroller resets. We reset our m/c by connecting the reset pin to Vcc

To implement RST = '0' & RST = '1'



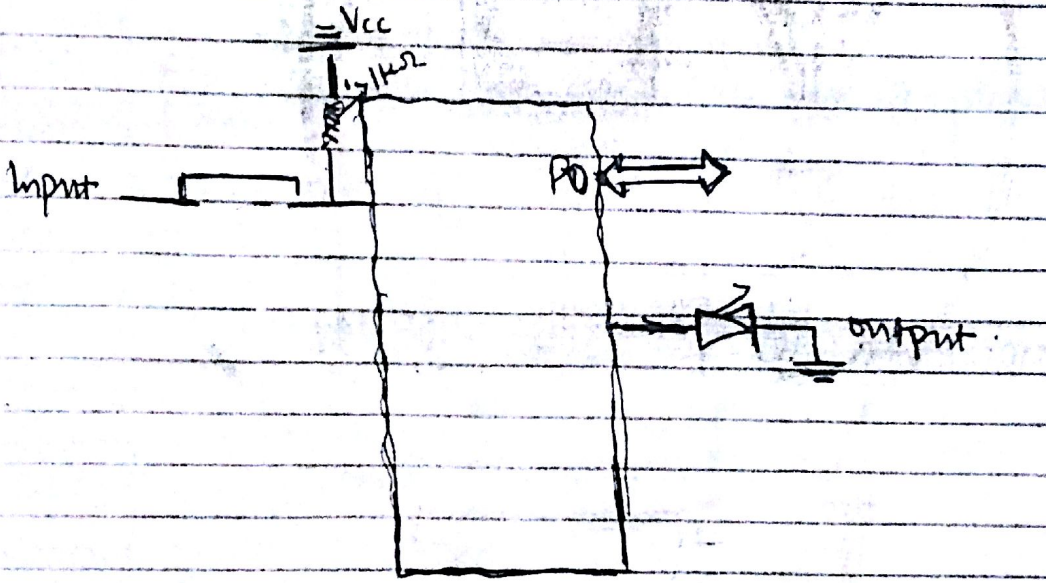
Ground pin is in pin no 20 while Vcc is pin no 40. pin 9 is

Generated by CamScanner

used to reset the MC. This pin will be connected to capacitor and resistor.

1) Turn ON MC
2) C is Uncharged
3) Uncharged C acts as short circuit for DC
4) RST pin is connected to Vcc through short circuit part because resistance is LOW
5) If RST pin is connected to Vcc for minimum of 2m/c then MC resets
6) Fully charged C acts as open circuit for DC
7) Now RST pin connected to ground through R.

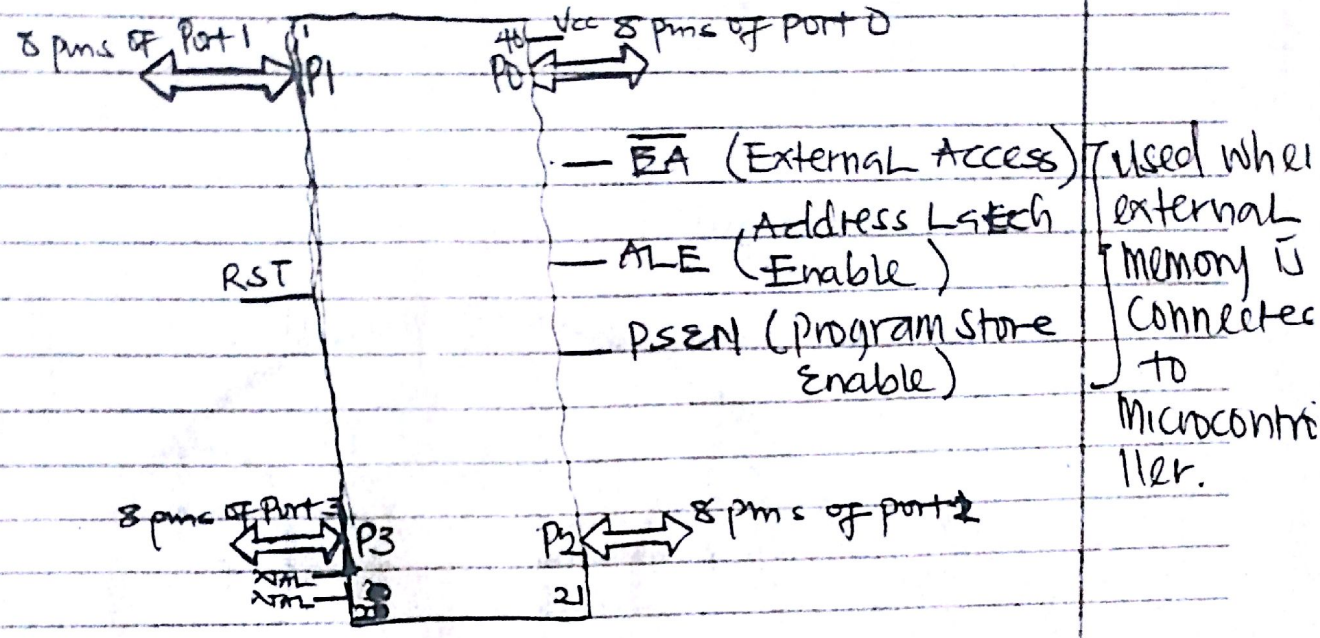
PIN DIAGRAM OF 8051 MICROCONTROLLER.



In 8051 microcontroller there are four (4) ports. Each port is of 8 pins. These four ports are named as port 0 (P0), port. 1 (P1), port. 2 (P2), port. 3 (P3). Each is 8 bit ports meaning there are 8 pins ports. These ports acts as input output ports so as a programmer you can use any of the ports as I/O ports.

A switch can be connected to any of the ports as an input device and LED connected as an output device. The pins to which the switch or LED is connected has to be programmed either as an input/output pins. i.e. if you have to configure any of the pins in the ports it must act as either input or output pins.

Generated by CamScanner



There will be eight pins in each of these ports making 32 pins. Vcc, GND, Reset, XTAL.1, XTAL.2 making 37 in total and the remaining 3 pins are PSEN (program Store Enable), ALE (Address Latch Enable), EA (External access). now we have 40 pins. The PSEN, ALE, EA are used when we are to use External Code Memory.

For the Microcontroller (MC) ports

1) Port.1 Exclusive for I/O. The eight pins of port1 is used for connecting Input/output devices. It is not for any other purpose.

2) P.0, P.2, & P.3 are in dual function. They can be used for other functions or conditions.

i) They can be used for I/O purpose or other functions

Now P.0 and P.2 are used for external memory. If they are used for external memory connection we can't use them for Input/Output connections

Generated by CamScanner

for Port 3 (P3.0, P3.1, P3.2, P3.3, P3.4, P3.5, P3.6, P3.7)

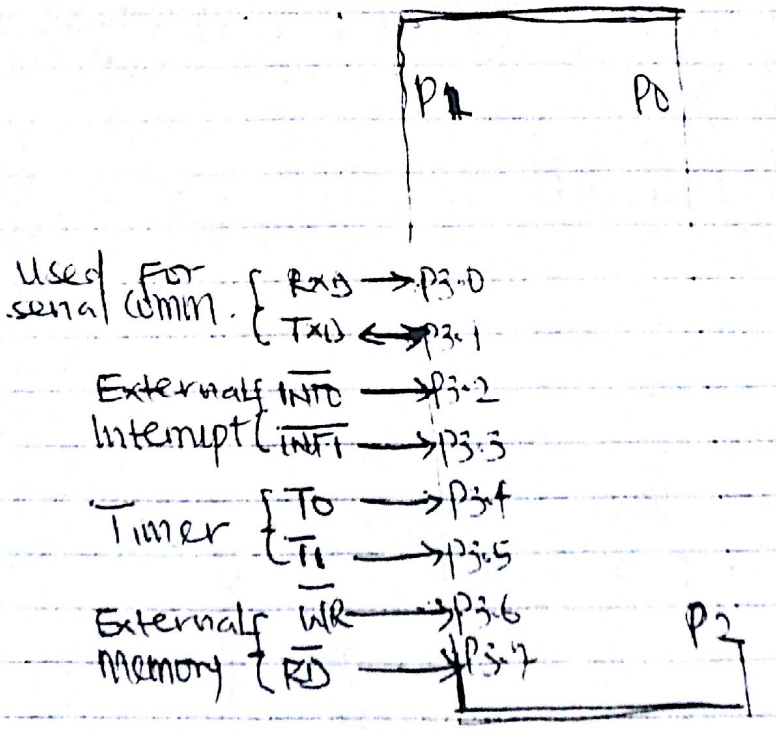
P3.0 (port 3 pin 0)

If you are using P3 that means you are using all the 8 pins but if you are using just a single pin say P3.1 then only that port is used

P3.0	P2.0	P1.0	P0.0
P3.1	P2.1	P1.1	P0.1

P3.7	P2.7	P1.7	P0.7
↓	↓	↓	↓
8	8	8	8 = 32 + others = 40 pins

Function of the 8 bits of port 3.



Generated by CamScanner