***Section II***

***Preliminary***

* **Introduction to Microcontrollers**
* **The 89C51 Microcontroller**
* **Features of 89C51**
* **Pin Diagram of 89C51**
* **Architecture of 89C51**

**Introduction to Microcontrollers**

**Definition**

An embedded microcontroller is a chip which has a computer processor with all its support functions (clock & reset), memory (both program and data), and I/O (including bus interface) built into the device. These built in functions minimize the need for external circuits and devices to be designed in the final application.

**Types of Microcontroller**

Creating applications for microcontrollers is completely different than any other development job in computing and electronics. In most other applications one probably have a number of subsystem and interfaces already available for his/her use. This is not the case with a microcontroller where one is responsible for –

* Power distribution
* System clocking
* Interface design and wiring
* System programming
* Application programming
* Device programming

Before selecting a particular device for an application, it’ s important to understand what the different options and features are and what they can mean with regard to developing application.

* **Embedded Microcontroller**

When all the hardware required to run the application is provided on the chip, it is refer to as an Embedded Microcontroller. All that is typically required to operate the device is power, reset, and a clock. Digital I/O pins are provided to allow interfacing with external devices.

* **External Memory Microcontroller**

Sometimes, the program memory is insufficient for an application or, during debug; a separate ROM (or even RAM) would make the work easier. Some microcontrollers including the 8051 allow the connection of external memory.

An external memory microcontroller seems to primarily differ from a microprocessor in the areas of built-in-peripheral features. These features could include memory device selection (avoiding the need for external address decoders or DRAM address multiplexers), timers, interrupt controllers, DMA, and I/O devices like serial ports.

**Microcontroller Memory Types**

There are number of different types of control store (Program Memory) that are available in different versions and different manufacturers’ 8051s. There is a fairly simple convention that is used to identify what type of control store a device has.

**The following is the list of conventions used for 8X51-**

**“X” value Control Store Type**

0 None

3 Mask ROM

7 EPROM

9 EEPROM/FLASH

**The 89C51 microcontroller**

The AT89C51 is a low-power, high-performance CMOS 8-bit microcomputer with 4K bytes of Flash programmable and erasable read only memory (PEROM). The device is manufactured using Atmel’s high-density nonvolatile memory technology and is compatible with the industry-standard MCS-51 instruction set and pin out. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with Flash on a monolithic chip, the Atmel AT89C51 is a powerful microcomputer which provides a highly-flexible and cost-effective solution to many embedded control applications.

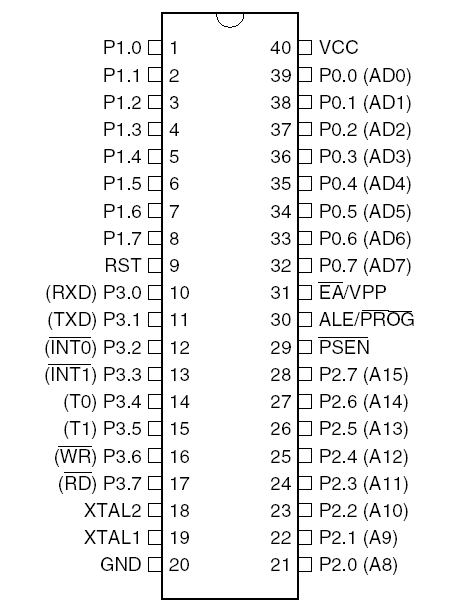
The AT89C51 is designed with static logic for operation down to zero frequency and supports two Software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port and interrupt system to continue functioning. The Power-down Mode saves the RAM contents but freezes the oscillator disabling all other chip functions until the next Hardware reset.

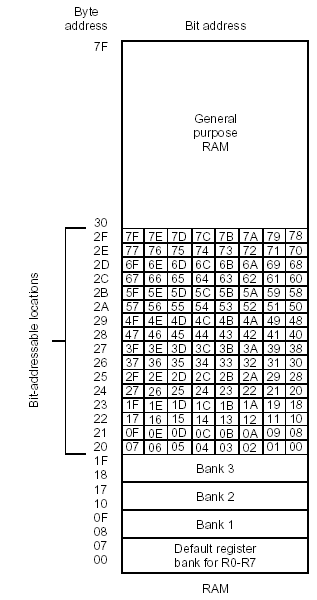
**Features of 89C51**

**Following is the features of 89C51 microcontroller as per the** **datasheet given by Atmel-**

* Compatible with MCS-51™ Products
* 4K Bytes of In-System Reprogrammable Flash Memory Endurance: 1,000 Write/Erase Cycles
* Fully Static Operation: 0 Hz to 24 MHz
* Three-level Program Memory Lock
* 128 x 8-bit Internal RAM
* 32 Programmable I/O Lines
* Two 16-bit Timer/Counters
* Six Interrupt Sources
* Programmable Serial Channel
* Low-power Idle and Power-down Modes

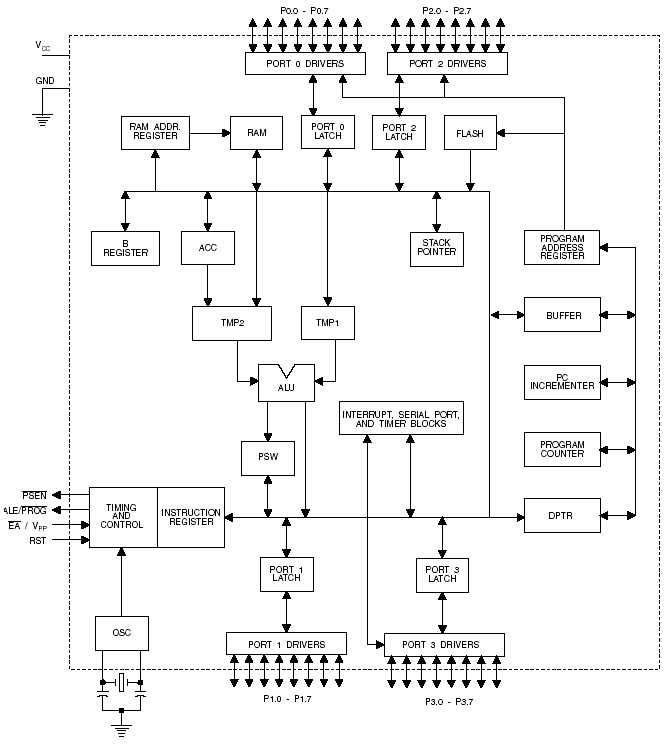
**Pin Diagram of 89C51:**



**Basic Reset Ckt.**

**Power On Reset Value.**

**Architecture of 89C51**

**Block Diagram** 

**Brief Description**

The AT89C51 is a low-power, high-performance CMOS 8-bit microcomputer with 4K bytes of Flash programmable and erasable read only memory (PEROM). The device is manufactured using Atmel’s high-density nonvolatile memory technology and is compatible with the industry-standard MCS-51 instruction set and pinout. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with Flash on a monolithic chip, the Atmel AT89C51 is a powerful microcomputer which provides a highly-flexible and cost-effective solution to many embedded control applications.

**Pin Description**

* **VCC**

Supply voltage.

* **GND**

Ground.

* **Port 0**

Port 0 is an 8-bit open-drain bi-directional I/O port. As an output port, each pin can sink eight TTL inputs. When 1s are written to port 0 pins, the pins can be used as high impedance inputs. Port 0 may also be configured to be the multiplexed low order address/data bus during accesses to external program and data memory. In this mode P0 has internal pull-ups. Port 0 also receives the code bytes during Flash programming, and outputs the code bytes during program verification. External pull-ups are required during program verification.

* **Port 1**

Port 1 is an 8-bit bi-directional I/O port with internal pull-ups. The Port 1 output buffers can sink/source four TTL inputs. When 1s are written to Port 1 pins they are pulled high by the internal pull-ups and can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (IIL) because of the internal pull-ups. Port 1 also receives the low-order address bytes during Flash programming and verification.

* **Port 2**

Port 2 is an 8-bit bi-directional I/O port with internal pull-ups. The Port 2 output buffers can sink/source four TTL inputs. When 1s are written to Port 2 pins they are pulled high by the internal pull-ups and can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (IIL) because of the internal pull-ups. Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that uses 16-bit addresses (MOVX @ DPTR). In this application, it uses strong internal pull-ups when emitting 1s. During accesses to external data memory that uses 8-bit addresses (MOVX @ RI), Port 2 emits the contents of the P2 Special Function Register.

Port 2 also receives the high-order address bits and some control signals during Flash programming and verification.

* **Port 3**

Port 3 is an 8-bit bi-directional I/O port with internal pull-ups. The Port 3 output buffers can sink/source four TTL inputs. When 1s are written to Port 3 pins they are pulled high by the internal pull-ups and can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source Current (IIL) because of the pull-ups. Port 3 also serves the functions of various special features of the AT89C51 as listed below:

**Port Pin Alternate Functions –**

P3.0 RXD (serial input port)

P3.1 TXD (serial output port)

P3.2 INT0 (external interrupt 0)

P3.3 INT1 (external interrupt 1)

P3.4 T0 (timer 0 external input)

P3.5 T1 (timer 1 external input)

P3.6 WR (external data memory write strobe)

P3.7 RD (external data memory read strobe)

Port 3 also receives some control signals for Flash programming and verification.

* **RST**

Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device.

* **ALE/PROG**

Address Latch Enable output pulse for latching the low byte of the address during accesses to external memory. This pin is also the program pulse input (PROG) during Flash programming. In normal operation ALE is emitted at a constant rate of 1/6 the oscillator frequency, and may be used for external timing or clocking purposes. Note, however, that one ALE pulse is skipped during each access to external Data Memory. If desired, ALE operation can be disabled by setting bit 0 of SFR location 8EH. With the bit set, ALE is active only during a MOVX or MOVC instruction. Otherwise, the pin is weakly pulled high. Setting the ALE-disable bit has no effect if the microcontroller is in external execution mode.

* **PSEN**

Program Store Enable is the read strobe to external program memory. When the AT89C51 is executing code from external program memory, PSEN is activated twice each machine cycle, except that two PSEN activations are skipped during each access to external data memory.

* **EA/VPP**

External Access Enable. EA must be strapped to GND in order to enable the device to fetch code from external program memory locations starting at 0000H up to FFFFH. Note, however, that if lock bit 1 is programmed, EA will be internally latched on reset. EA should be strapped to VCC for internal program executions. This pin also receives the 12-volt programming enable voltage (VPP) during Flash programming, for parts that require 12-volt VPP.

* **XTAL1**

Input to the inverting oscillator amplifier and input to the internal clock operating circuit.

* **XTAL2**

Output from the inverting oscillator amplifier.

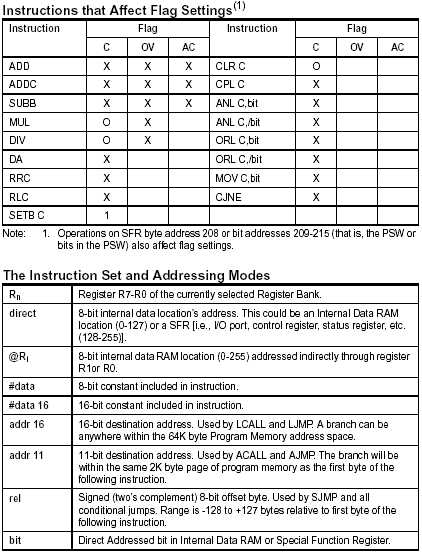
**Memory Organization**



**Program Memory**

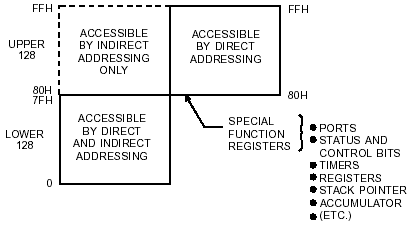
The TEMIC C51 Microcontroller Family has separate address spaces for program Memory and Data Memory. The program memory can be up to 64 K bytes long. The lower 4 K for the 80C51 (8 K for the 80C52, 16 K for the 83 C154 and 32 K for the 83C154D) may reside on chip.

**Data Memory**

The C51 Microcontroller Family can address up to 64 K bytes of Data Memory to the chip. The “MOVX” instruction is used to access the external data memory (refer to the C51 instruction set, in this chapter, for detailed description of instructions). The 80C51 has 128 bytes of on-chip-RAM (256 bytes in the 80C52, 83C154 and 83C154D) plus a number of Special Function Registers (SFR). The lower 128 bytes of RAM can be accessed either by direct addressing (MOV data addr). or by indirect addressing (MOV @Ri). 

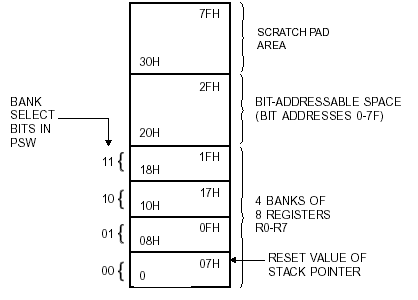
**Internal Data Memory Organization**

Internal data memory is shown in Figure. The memory space is divided into three blocks, which are generally referred to as the Lower 128, the Upper 128, and SFR space.



Internal data memory addresses are always 1 byte wide, which implies an address space of only 256 bytes. However, the addressing modes for internal RAM can in fact accommodate 384 bytes. Direct addresses higher than 7FH access one memory space, and indirect addresses higher than 7FH access a different memory space. Thus, Figure 7 shows the Upper 128 and SFR space occupying the same block of addresses, 80H through FFH, although they are physically separate entities.

**Scratch Pad RAM Organization**

Figure shows how the lower 128 bytes of RAM are mapped. The lowest 32 bytes are grouped into 4 banks of 8 registers. Program instructions call out these registers as R0 through R7. Two bits in the Program Status Word (PSW) select which register bank is in use. This architecture allows more efficient use of code space, since register instructions are shorter than instructions that use direct addressing.

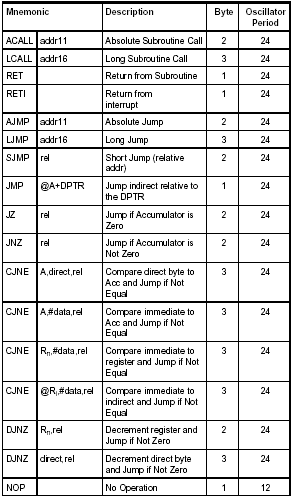
**The 128 Byte Memory**

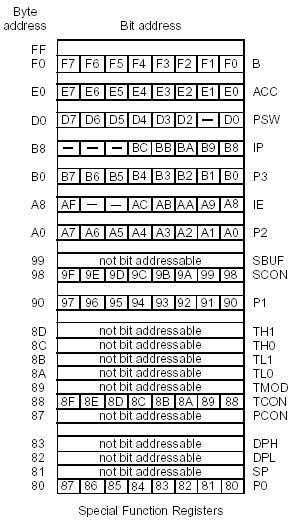
The next 16 bytes above the register banks form a block of bit-addressable memory space. The microcontroller instruction set includes a wide selection of single-bit instructions, and these instructions can directly address the 128 bits in this area. These bit addresses are 00H through 7FH. All of the bytes in the Lower 128 can be accessed by either direct or indirect addressing. The Upper 128 (Figure 9) can only be accessed by indirect addressing. The Upper 128 bytes of RAM are only in the devices with 256 bytes of RAM.

**The SFR**

Figure gives a brief look at the Special Function Register (SFR) space. SFRs include Port latches, timers, peripheral controls, etc. These registers can only be accessed by direct addressing. In general, all Atmel microcontrollers

have the same SFRs at the same addresses in SFR space as the AT89C51 and other compatible microcontrollers. However, upgrades to the AT89C51 have additional SFRs. Sixteen addresses in SFR space are both byte- and bitaddressable. The bit-addressable SFRs are those whose address ends in 000B. The bit addresses in this area are 80H through FFH.





**Appendix C**

**89C51 Instruction Set**

