MICROCONTROLLERS AND APPLICATIONS

BY

Dr. **P. VENKATESWARAN** M.E. Tel. E. (J.U.), Ph.D. (JU)
MISTE MIETE F.I.E.(I) SMIEEE (USA) [redacted]
[redacted]
[redacted] OF [redacted]
[redacted] - [redacted] Dt.

ACKNOWLEDGEMENTS

[redacted] DEPT. OF ETCE
JADAVPUR UNIVERSITY CALCUTTA - 700 032.

Mr. VIVEK G. KHEKALE
CENTRE FOR ELECTRONICS DESIGN TECHNOLOGY
CEDT - AURANGABAD MAHARASTRA

TOPICS

- **INTRODUCTION**
- **8051**
- **FEATURES**
- **PIN-OUT DETAILS**
- **MEMORY ORGANIZATION**
- **ADDRESSING MODES**
- **INSTRUCTION SET**
- **SAMPLE PROGRAMS**
- **APPLICATIONS**

1. Introduction

"The microprocessors made an impact on industries of vast diversification in nature such as machine tools [1], bio-medical instrumentation and chemical processes [l], etc., thus proved its utility as a general purpose system design (GPSD) tool [2], with *suitable interfacing devices*. The necessity to evolve a higher performance architecture, to be used as a specific purpose system design (SPSD) tool [2] for *dedicated* applications leads to the development of *microcontrollers.*

The microprocessor architecture has three basic segments : arithmatic / logic unit (ALU), register unit and control unit [1]. Other hardware requirements such as, for clock generator, RAM, ROM / EPROM, parallel and serial I / O ports are to be supported in the form of external chips and that is the difference between microprocessors and microcontrollers [2].

Microcontrollers, also called as Single-chip microcomputers are designed on a single-chip such that besides microprocessing unit, it includes RAM, ROM / EPROM, I / O ports, timer / counter and other circuitry required for microcomputer working [3]. This is shown in Fig. 1.

Microcontrollers are used primarily to perform dedicated functions [1]. They are also used as independent controllers in machines or as slave processors in the distributed processing environment [1].

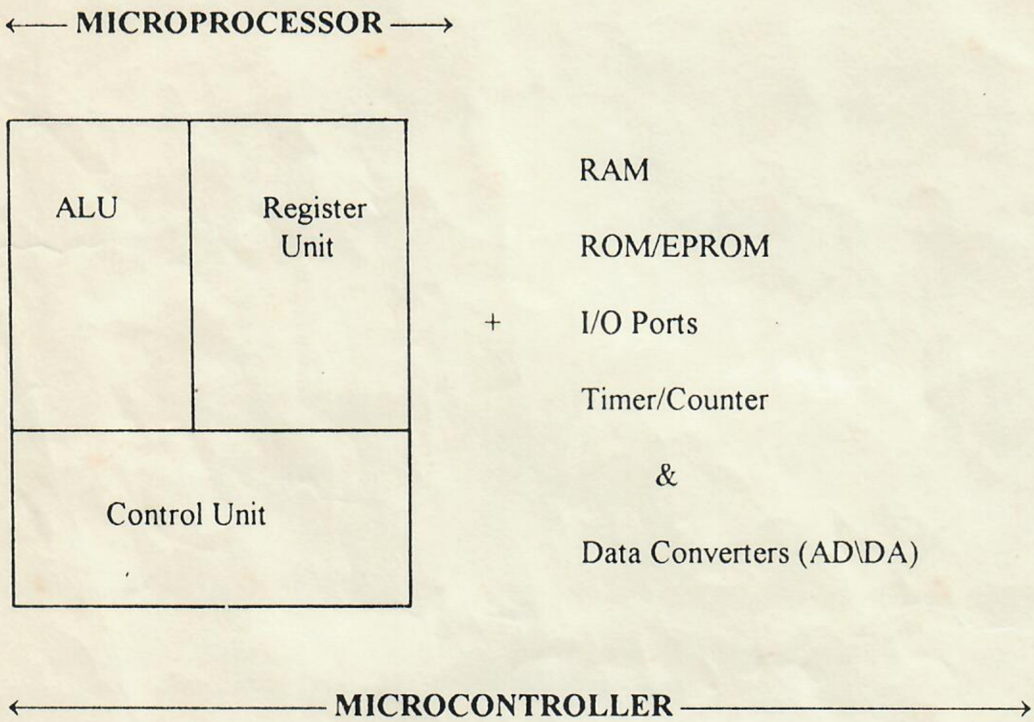


Fig. 1 : Block diagram of Microcontroller

LDR and Stepper Motor

Sensor and *actuator* are the two major cmponents of a positioning system. The requirements of the control task determines the nature of these two components and subsequently the cost, also. Light dependant resistor (LDR), which exhibits light

II. EVOLUTION OF MICROCONTROLLERS

In 1971, Intel corporation developed the first 4 bit microprocesor architecture. Soon, Motorola, Zilog and other manufactures brought out their respective microprocessors to the market. The 8085, an 8-bit micrprocesor from Intel, made a big impact in the market.

One design trend is to go for 16 and higher bit microprocessors to improve the data handling capabilities and speed of execution. 8086 and 8088 are the 16-bit micro processors from Intel Corporation. The 8088 is a proceessor with 8-bit data bus and 16-bit capabilities. Another design trend is to go for higher performance architecture for dedicated applications, meant to be used as specific purpose system design tool.

For example microcomputer design based on microprocessor involves the interfacing of several hardwares such as memory, parallel I/O ports, serial I/O ports, timer, etc. Thus any microprocessor based application design necessitates the handling of multiple chips, resulting in increased PCB size and lesser reliability in the functioning of the circuit. This concept was totally changed with the introduction of single-chip microcomputer/microcontroller by Intel Corporation in 1976. Because of recent advances in semiconductor design and fabrication technology, it is possible to integrate the entire system on a single chip.

For example, Intel's MCS-48 microcontroller family has RAM, ROM/EPROM, Parallel I/O ports, timer/counter besides an 8-bit CPU. Thus with microcontrollers, the designer can make the system design more compact at less cost and yet powerful. In 1980, Intel came up with MCS-51 series of microcontrollers featuring serial interface facility, also. This second generation microcontrollers are more popular for sophisticated real-time instrumentation and industrial control applications. This was followed by MCS-96 family of microcontrollers, fastest among Intel's series, incorporating A/D converter also, with a 16 bit CPU.

Intel products are upward compatible with successive architectures. Peripherals and interfacing chips are easily available in the market for Intel products. Moreover, Intel provides well documentation support and hence Intel products are more popular.

III. MCS - 51 Family and its features :

The 8051 is the core of MCS-51 family and its **features** are (14.1):

8-bit CPU meant for control applications
Incorporated with Boolean (single-bit) processor
128 bytes Scratch-pad memory (on –chip data RAM)
4K bytes on-chip program memory (PM)
64K expandable program and data memory (PM and DM) space
32 bidierctional, individually accessible I/O lines, and can be accessed as a group of 8 lines.
Two 16-bit Timer/Counter
Serial interface facility
6/5 vecter interrupts with two levels of priority
On-chip clock oscillator

IV. Pin description :

Fig 2 shows the MCS-51 Pin diagram. Various signals available at the 40 pins can be classified under the following categories:

1. Port 0
2. Port 1
3. Port 2
4. Port 3
5. Bus-control signals
6. On-chip oscillator signals
7. Power supply signals

3. MCS - 51 Hardware Description

3.1 Introduction

It has been pointed out in section 2.2 that, MCS-51 is a second generation microcontroller from Intel Corporation, highly popular for real-time instrumentation and control applicaions. The logic design of MCS-51 family, which influences the data manipulation and communication [1], is presented in this chapter. This chapter begins with the introduction to 'MCS-51 Family, and its Features'. The MCS-51. 'Pin Diagram' is presented next. MCS-51 devices features both Program and Data Memory. The reason for their logical separation, their memory map, access to external PM and DM, on-chip memory configuration including the Special Function Register (SFR) area, are discussed under 'Memory Organization'. This is followed by the MCS-51 'Architectural Block Diagram'. The simplified internal circuitry of an 'I/O Port Pin' is also presented, at the end.

3.2 MCS-51 Family and its Features

The MCS-51 family of devices have different versions, depending on the internal hardware availability, as listed in Table 3.1

Table 3.1 : MCS-51 Family of Microcontrollers [14-C]

device name	ROM less version	EPROM version	ROM bytes	RAM bytes	16-bit Timers	circuit type
8051	8031	(8751)	4K	128	2	HMOS
8051AH[1]	8031AH	8751H	4K	128	2	HOMS
8052AH[2]	8032AH	8752BH	8K	256	3	HOMS
80C51BH	80C31BH	87C51	4K	128	2	CHMOS[3]

[1] 8051AH is identical to 8051, but fabrication is with HMOS 11 technology.

[2] 8052AH is the enhanced version of 8051AH

[3] Both HOMS and CHMOS devices (with 'C' designation in the middle of the device name) are fully compatible, but CMOS draw less current. In CHMOS devices, two reduced power modes are added [14.a] (Details of these modes are given under 'Power Control Register' of Special Function Register - SFR, in Section 3.4).

The 8051 is the *core* of MCS-51 family and its **features** are [14.a] :

8-bit CPU meant for control applications

Incorporated with Boolean (single-bit) processor

128 bytes Scratch-pad memory (on-chip data RAM)

4k bytes on-chip program memory (PM)

64K expandable program and data memory (PM and DM) space

32 bi-dierctional, individually accessible I/O lines, and can be accessed as a group of 8

lines.

Two 16-bit Timer/counter

Serial interface facility

6/5-vecter interrupts with two levels of priority

On -chip claock oscillator

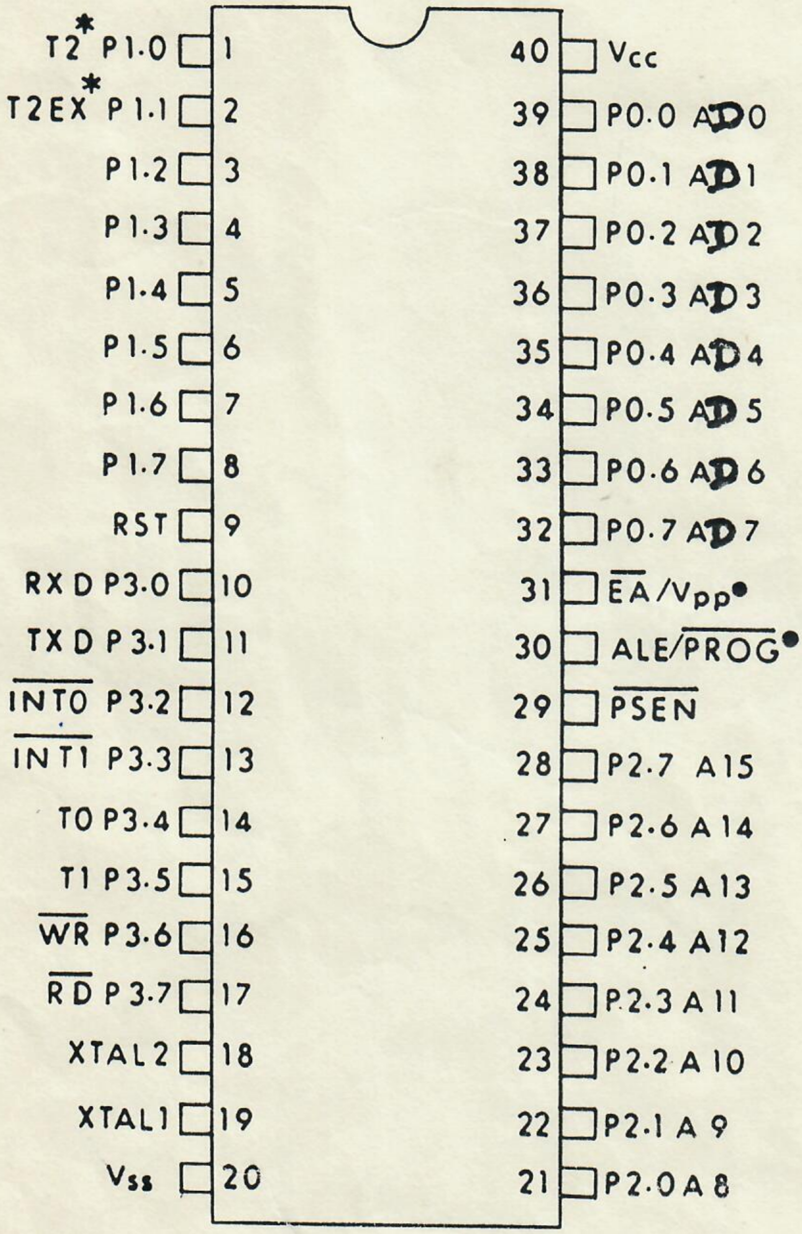
3.3 Pin Description

Fig 3.1 shows the MCS-51 Pin diagram [14.c]. Various signals available at the 40 pins can be classified under the following categories :

1. Port 0
2. Port 1
3. Port 2
4. Port 3
5. Bus-control signals
6. On-chip oscillator signals
7. Power supply signals

Memory requirements of the system designed determines the nature of use of these ports. For designs with no or minimum external memory, they are used as general purpose I/O ports. For system design with external memory, the ports provide the address and / data lines, and control signals. Thus, *basically* port lines serve dual purpose. In 8051, three out of four ports (24 lines) are dual purpose (26 lines on the 8052/8032).

The 8-lines in each port can be treated as a *single-unit* while interfacing to parallel devices such as seven - segment LED's, data converters, memory, etc. Each line can *also* be operated independently while interfacing to single-bit devices such as LED's, CRT terminals, etc.



*8032/8052 only

•EPROM only

Fig. 3.1 MCS-51 Pin (DIP) Diagram [14.C]

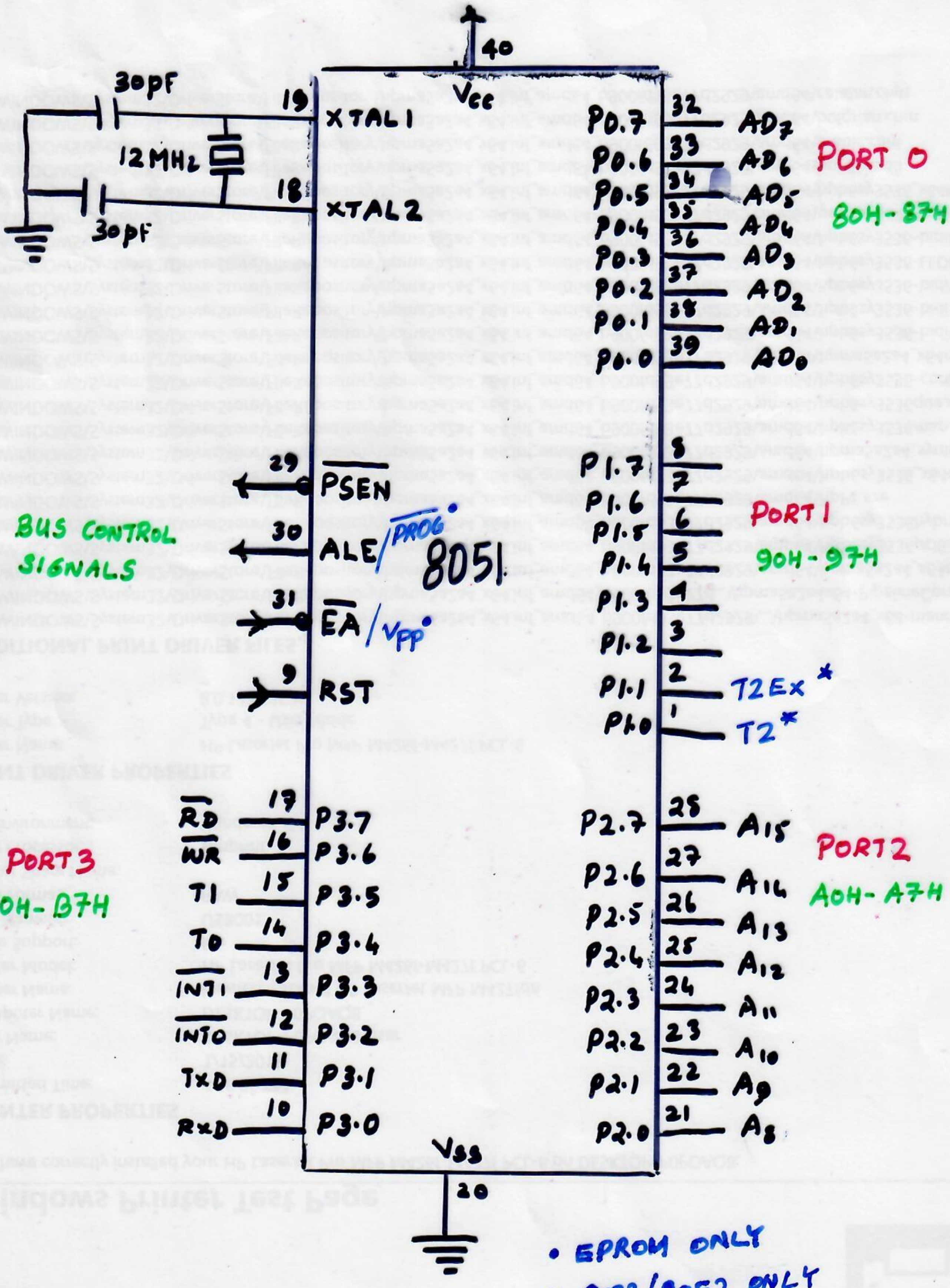


FIG. 1 : 8051 PIN-OUTS

1. PORT O

Port O serves dual purpose. Depends on the memory requirement of the system **designed,** it is used either as a general purpose I/O port or as a low order *multiplexed* **address/data** bus. During programming of the (on-chip) EPROM parts, Port O receives the code bytes. During progrm verification of the ROM/EPROM parts, Port O emits the code bytes [14.c].

2. PORT 1

Port1 is a dedicated, general purpose I/O port. But in 8052/8032, bits 0 and 1 of Port 1 has the alternate function as the external inputs to the third timer. This is shown in Table 3.2. During programming of the EPROM parts, and program verification of the ROM and EPROM parts, Port1 receives the low-order address bytes [14.c].

3. PORT 2

Port 2 is also a dual purpose port. It serves as a general purpose I/O port or as high-order address bus for designs with external program (code) memory or more than 256 bytes of external data memory. During programming of the EPROM parts and program verification of the ROM and EPROM parts, Port 2 receives the high-order address bits [14.c].

4. PORT 3

Port 3 is another dual purpose port. Apart from their use as a general purpose I/O, each line in Port3 have specific alternate function, as listed in Table 3.2.

Table 3.2 : **Port Pin Alternative Function** [14.c]

Port Pin	Alternate Function	Name
P 3.0	serial port (I/P)	R x D
P 3.1	serial port (O/P)	T x D
P 3.2	external interrupt O	$\overline{\text{INTO}}$
P 3.3	external interrupt 1	$\overline{\text{INTI}}$
P 3.4	Timer/Counter O external input	TO
P 3.5	Timer/Counter external input	TI
P 3.6	external data memory write strobe	$\overline{\text{WR}}$
P 3.7	external data memory read strobe	$\overline{\text{RD}}$
P 1.0*	Timer / Counter 2 external input	T2
P 1.1*	Timer / Counter 2 capture reload	T2EX

* 8052 / 8032 only

5. BUS CONTROL SIGNALS

There are four bus control signals.

i) $\overline{\text{PSEN}}$ - Program Store ENable, is the *read* control signal to external program (code) memory and is activated twice in each machine cycle. During external data memory access, two $\overline{\text{PSEN}}$ avtivations are skipped in each machine cycle [14.c]. For reading of program bytes, $\overline{\text{PSEN}}$ is connected to EPROM's OE (output Enable) pin. During execution from on-chip ROM, $\overline{\text{PSEN}}$ remains inactive (high).

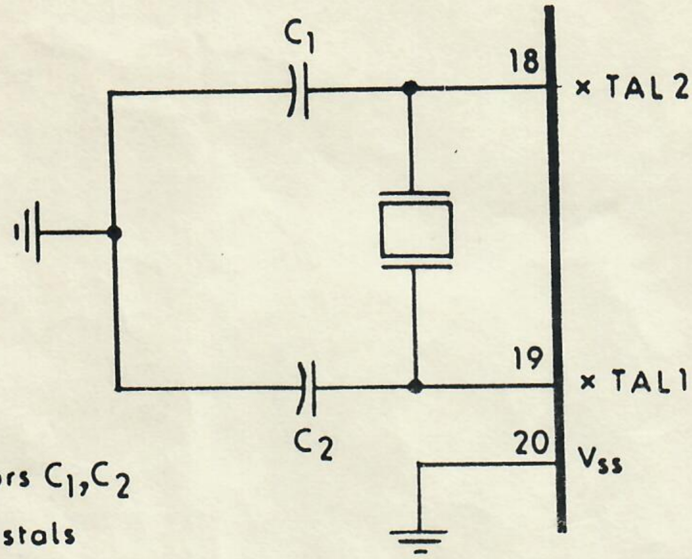
ii) ALE/ $\overline{PROG}$ - Address Latch Enable, is used to demultiplex the low order address / data bus, when Port O is used in its alternate mode. During programming of the EPROM parts, this is the program pulse input ($\overline{PROG}$) line [14.c] ALE switches at the rate of 1/6th of the oscillator frequency, and can be used for clocking / timing function. During external data memory access, ALE pulse is skipped [14.c].

(iii) $\overline{EA}$/ V_{PP} - External Access enable is an input signal. It must be tide to ground (V_{SS}) during execution from external program memory, and to V_{CC} during on-chip (internal) program execution. In the EPROM device, if the Security Bit is programmed, then no program (code) byte can be fetched from the external program memory [14.c]. During programming of the EPROM parts, it receives the programming supply voltage (V_{PP}) [14.c].

(iv) RST - When Reset input is made high for two machine cycles, device gets resetted for start up.

6. ON-CHIP OSCILLATOR SIGNALS

XTAL - 1 is the is the input to the inverting oscillator amplifier and XTAL - 2 is the output from the inverting oscillator, and can be configured *for use* as on-chip osillator. This is shown in Fig 3.2 [14.c]. The device can be driven by the external clock source also. For this, XTAL1 is grounded and XTAL2 is driven, as shown in Fig 3.3 [14.c]. As the input to internal clock circuit is through a divide-by-two flip-flip, only the minimum and maximum high and low times of the external clock signal must be observed as per the data sheet and duty cycle requirement is not a factor [14.c]. For *most* devices in the MCS-51 family, the crystal frequency is 12 MHZ.



Stabilizing Capacitors C_1, C_2
30 pF ± 10 pF for Crystals
40 pF ± 10 pF for Ceramic Resonators

Fig. 3.2 MCS-51 On-chip Oscillator Configuration [14.C]

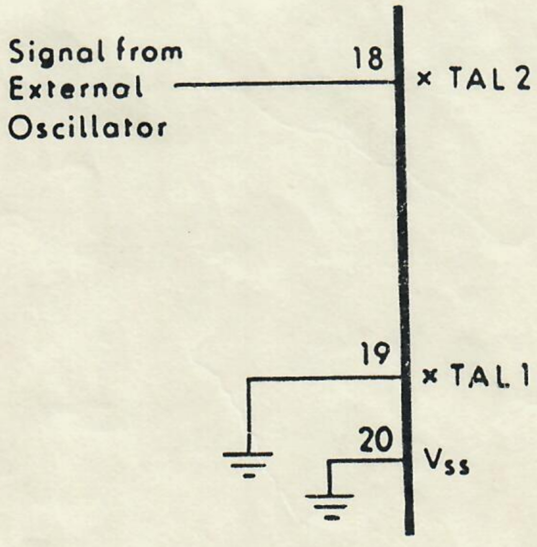


Fig. 3.3 MCS-51 External Clock Drive [14.C]

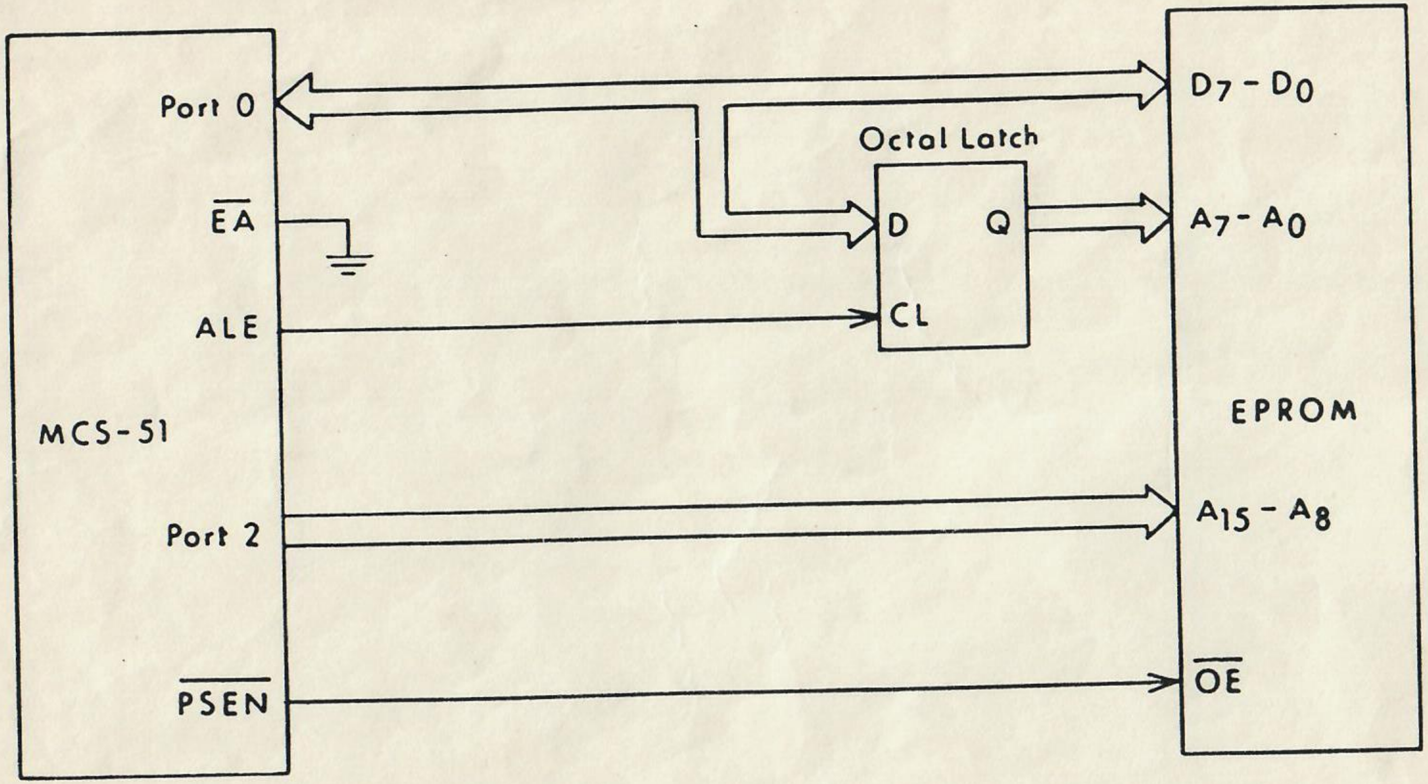


Fig. 3.5 External Program Memory access [14.a]

DATA MEMORY

Fig 3.7 shows the memory map of 8051 data memory. The internal (on-chip RAM) data memory is divided into *two* distinct blocks. They are : lower 128 bytes of RAM (256 bytes in 8052) in the address range OOH-7FH, Special Function Register (SFR) area, in the address range 80H-FFH [14.b].

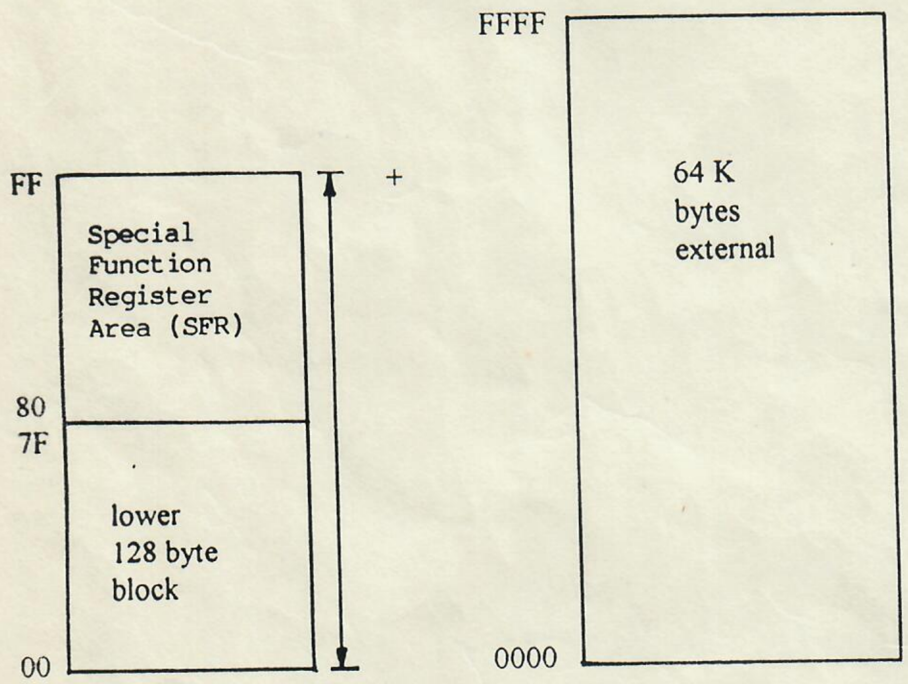


Fig 3.7 : Memory Map : Data Memory of 8051 [14.b]

As shown in Fig. 3.7, the 8051 can address upto 64K bytes of data memory external to the chip. The 'MOVX' intruction is used to access the external data memory [14.b]. For external memory (PM or DM) with 16-bit address, a 16-bit data pointer (DPTR) register is used to hdd the external memory address.

External Data Memory access

The configuration for external data memory access of upto 2k bytes is shown in Fig 3.8 [14.a]. Port O is usedas multiplexed address/data bus. In Port 2, 3-lines are used to *page* the RAM, as the external data address is 11-bits wide. The necessary $\overline{RD}$ and $\overline{WR}$ control signals are generated by the CPU, during the external RAM access [14.a].

Timing for External Data Memory access

The timing for a read operation of external data memory is shown in Fig 3.9. The execution corresponds to the instruction MOVX A, @ DPTR - move the content of external memory location, pointed at the by data pointer, DPTR, to accumulator. As shown in Fig. 3.9, both ALE and $\overline{PSEN}$ pulses are missed, instead the RAM is enabled by the $\overline{RD}$ control signal [14. a,c]

For MOVX @ DPTR, A - write operation, the timing is essentially the same except that $\overline{WR}$ signal is activated (low) and data is available on Port O. The $\overline{RD}$ line remains de-activated (high).

On-chip Data Memory - The lower 128 bytes

The memory configuration of the lower 128 byte block of internal RAM is shown in Fig. 3.10. The first 32 bytes, in the address range OOH-IFH constitutes *general purpose registor banks*, bank O through 3, with 8 registers in each bank, identified by the software as RO through R7. The current register bank in use is selected by the register bank select bits in the program satus word (PSW). By default, register bank 0

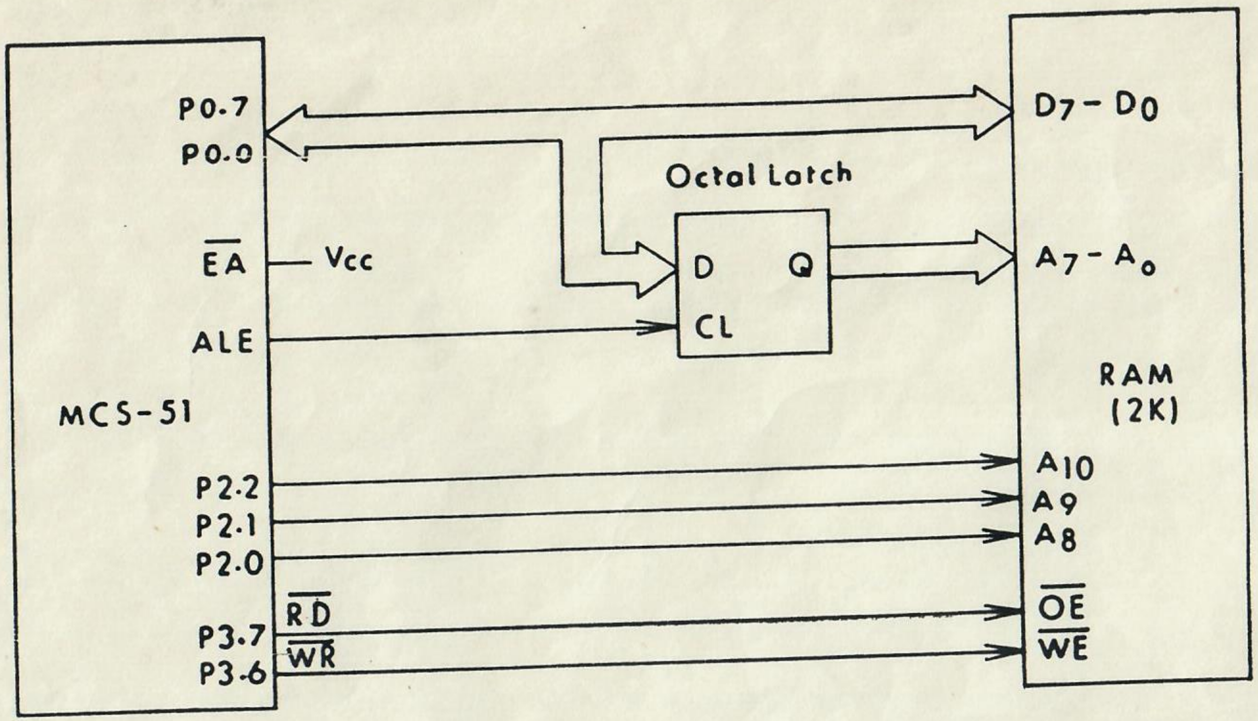


Fig. 3.8 External Data Memory access [14.a]

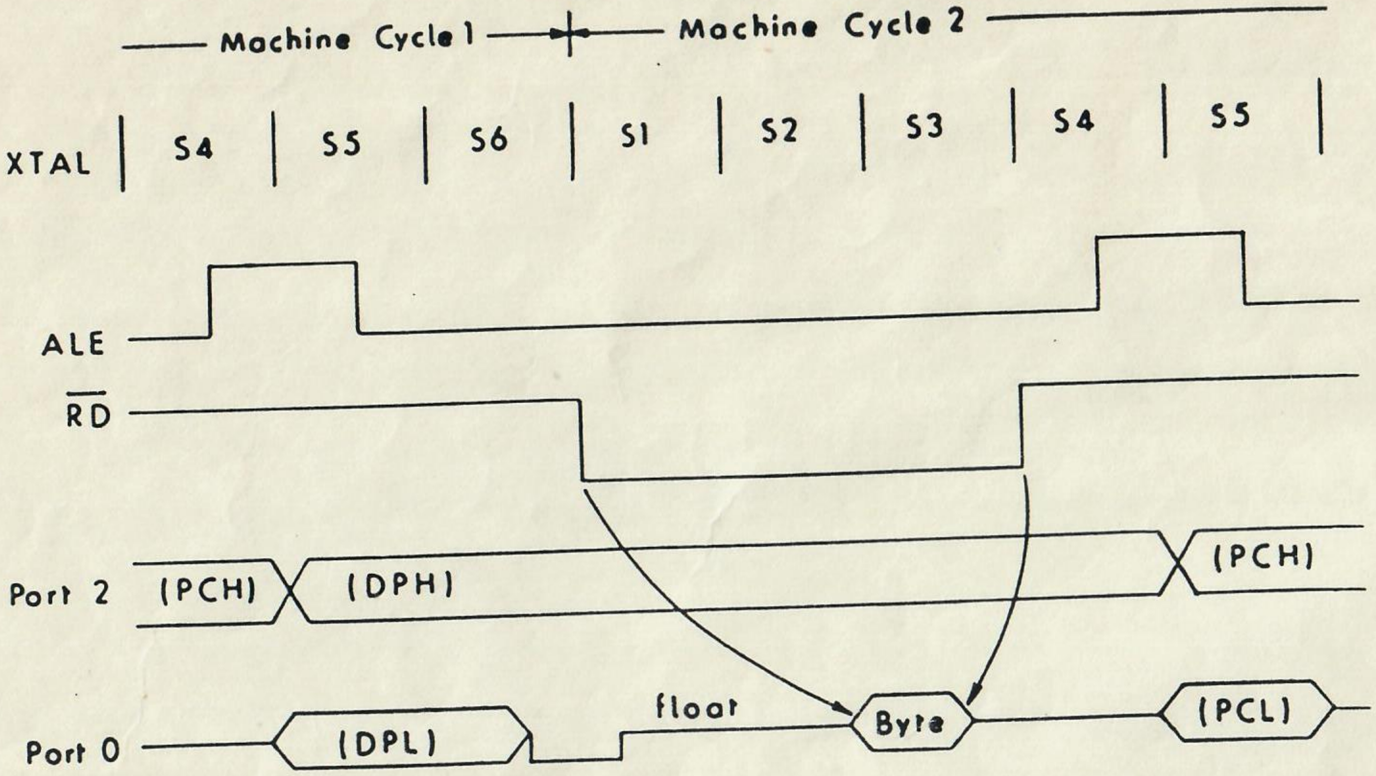


Fig. 3.9 Timing for External Data Memory Read [14.C]

External Data Memory access

The configuration for external data memory access of upto 2k bytes is shown in Fig 3.8 [14.a]. Port O is usedas multiplexed address/data bus. In Port 2, 3-lines are used to *page* the RAM, as the external data address is 11-bits wide. The necessary $\overline{RD}$ and $\overline{WR}$ control signals are generated by the CPU, during the external RAM access [14.a].

Timing for External Data Memory access

The timing for a read operation of external data memory is shown in Fig 3.9. The execution corresponds to the instruction MOVX A, @ DPTR - move the content of external memory location, pointed at the by data pointer, DPTR, to accumulator. As shown in Fig. 3.9, both ALE and $\overline{PSEN}$ pulses are missed, instead the RAM is enabled by the $\overline{RD}$ control signal [14. a,c]

For MOVX @ DPTR, A - write operation, the timing is essentially the same except that $\overline{WR}$ signal is activated (low) and data is available on Port O. The $\overline{RD}$ line remains de-activated (high).

On-chip Data Memory - The lower 128 bytes

The memory configuration of the lower 128 byte block of internal RAM is shown in Fig. 3.10. The first 32 bytes, in the address range OOH-IFH constitutes *general purpose registor banks*, bank O through 3, with 8 registers in each bank, identified by the software as RO through R7. The current register bank in use is selected by the register bank select bits in the program satus word (PSW). By default, register bank 0

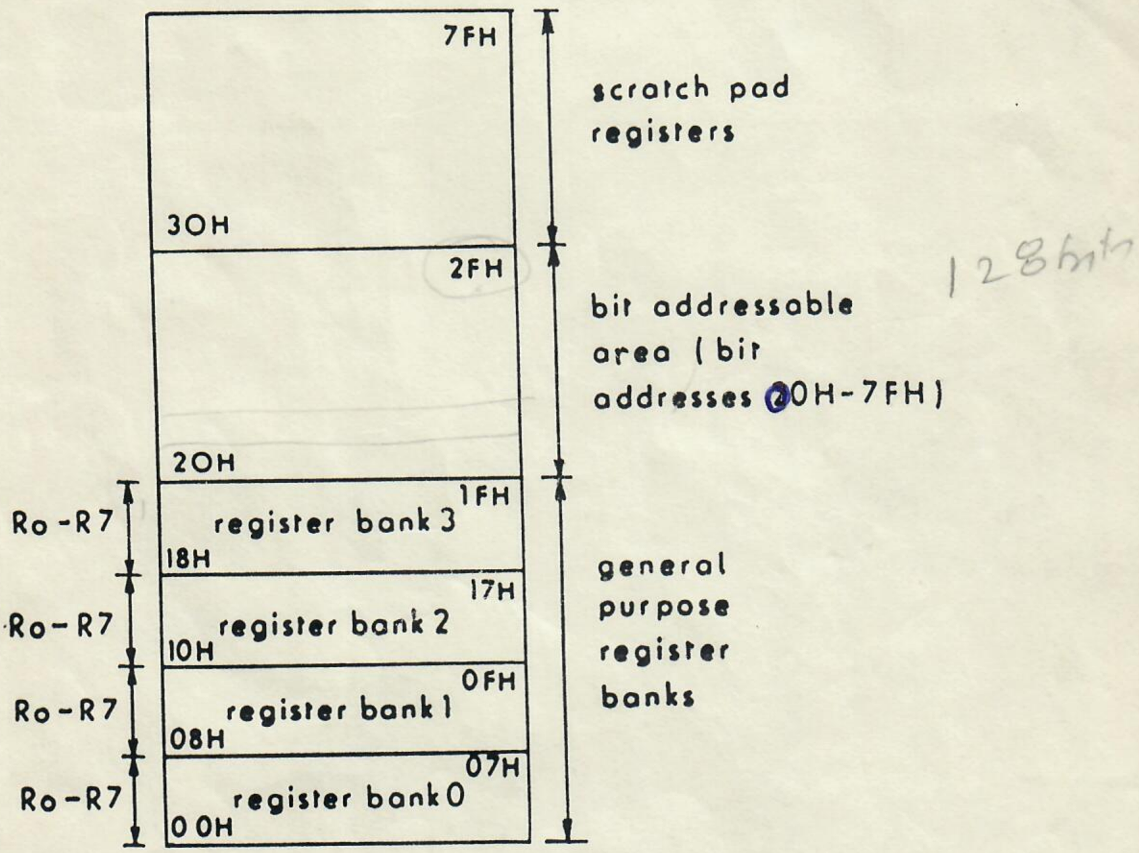


Fig. 3.10 The Lower 128 Byte Block of Internal (on-chip RAM) Data Memory [14.a]

is selected always. The functional description of PSW register bits of MCS-51 devices is given in Fig. 3.11. [14.a]. The registers in the given bank can be identified with their respective address also instead of RO-R7. For example,

MOV A, O1H (2-byte instruction - Move the content of internal RAM location with the address O1H to accumulator - direct addressing) and

MOV A, R1 (1-byte instruction - register direct addressing).

perform the *same* operations. As registers RO-R7 oriented instrucions are shorter, they are fast and allows efficient memory management [14.a]. In microcontrollers, *stack* resides in the internal data memory instead of external RAM, which is typical of microprocessor. By default, the stack pointer register (SP) is initialized to 07H and the stack starts from 08H onwards. If the Programmer wants to use the register banks, as 08H corresponds to RO register of bank 1, then stack pointer register can be reinitialized with the *scratch pad location*.

CY	AC	FO	RS1	RSO	OV	-	P
7	6	5	4	3	2	1	0

Fig. 3.11 : PSW register (program status word) of MCS - 51 [14.a]

Bit 0 - Parity flag : Maintain even parity with accumulator A. For (A) = 1000 0011, (P) = 1

Bit 1 - Reserved

Bit 2 - Overflow flag : set by arithmatic overflow with signed number (2's complement) scheme.

Bit 4 & Bit 3 - Register Bank Select bits.

0	0	-	bank 0
0	1	-	bank 1
1	0	-	bank 2
1	1	-	bank 3

Bit 5 - General purpose flag : for user application

Bit 6 - Auxillary Carry flag : set to 1 during BCD addition,

i) if there is a carry out of bit 3 to bit 4 or when lower nibble is > 09H.

ii) if there is a carryout of bit 7 or when upper nibble is > 09H.

Bit 7 - Carry (CY) flag : dual purpose

1. set to 1 if thereis a carry/borrow out of bit 7 during arith matic operations.

2. Bollean Processor - single - bit accumulator.

The next sixteen bytes in the lower 128 byte block of on-chip RAM contain 128 (16 x 8) *bit-addressable locations*. When these locations are accessed as bytes, the memory map will be 20H-2FH. When they are accessed as bits, the memory map will be 00H-7FH. The remaining bytes in the address range 30H-7FH constitute the *scratch pad area*.

The bytes in the lower 128 block of on-chip RAM can be accessed by direct or indirect addressing. The upper 128 byte block of on-chip RAM, in the address range 80H-FFH, can be accessed only by indirect addressing. This block is *not* implemented in 8051. In devices with 256 bytes of RAM such a 8052/8032, the upper 128 byte block is available and can be used as *stack space* [14.a].

Special Function Registers

A memory map of the on-chip memory space called Special Function Register (SFR) area [14.c], is shown in Fig. 3.12. SFRs, include the Port latches, timer/counters and other peripheral controls [14.a]. SFR memory map is also 80H-FFH, but are accessed by *direct addressing*. The registers with address that ends as OH/8H/OOOB in the SFR area, are both bit and byte addressable. As shown in Fig.3.12, most of the addresses in the SFR space are not implemented.

RAM

Byte address	Bit address								
7F – 30	General purpose RAM								
2F	7F	7E	7D	7C	7B	7A	79	78	Bit-addressable locations
2E	77	76	75	74	73	72	71	70	
2D	6F	6E	6D	6C	6B	6A	69	68	
2C	67	66	65	64	63	62	61	60	
2B	5F	5E	5D	5C	5B	5A	59	58	
2A	57	56	55	54	53	52	51	50	
29	4F	4E	4D	4C	4B	4A	49	48	
28	47	46	45	44	43	42	41	40	
27	3F	3E	3D	3C	3B	3A	39	38	
26	37	36	35	34	33	32	31	30	
25	2F	2E	2D	2C	2B	2A	29	28	
24	27	26	25	24	23	22	21	20	
23	1F	1E	1D	1C	1B	1A	19	18	
22	17	16	15	14	13	12	11	10	
21	0F	0E	0D	0C	0B	0A	09	08	
20	07	06	05	04	03	02	01	00	
1F – 18	Bank 3								
17 – 10	Bank 2								
0F – 08	Bank 1								
07 – 00	Default register bank for R0 – R7								

SPECIAL FUNCTION REGISTERS

Byte address	Bit address								
FF									
F0	F7	F6	F5	F4	F3	F2	F1	F0	B
E0	E7	E6	E5	E4	E3	E2	E1	E0	ACC
D0	D7	D6	D5	D4	D3	D2	–	D0	PSW
B8	–	–	–	BC	BB	BA	B9	B8	IP
B0	B7	B6	B5	B4	B3	B2	B1	B0	P3
A8	AF	–	–	AC	AB	AA	A9	A8	IE
A0	A7	A6	A5	A4	A3	A2	A1	A0	P2
99	not bit addressable								SBUF
98	9F	9E	9D	9C	9B	9A	99	98	SCON
90	97	96	95	94	93	92	91	90	P1
8D	not bit addressable								TH1
8C	not bit addressable								TH0
8B	not bit addressable								TL1
8A	not bit addressable								TL0
89	not bit addressable								TMOD
88	8F	8E	8D	8C	8B	8A	89	88	TCON
87	not bit addressable								PCON
83	not bit addressable								DPH
82	not bit addressable								DPL
81	not bit addressable								SP
80	87	86	85	84	83	82	81	80	P0

Fig. - 2.6

8 Bytes

F8									FF
FO	B								F7
E8									EF
EO	ACC								E7
D8									DF
DO	PSW								D7
C8	T2CON*		RCAP2L*	RCAP2H*	TL2*	TH2*			CF
CO									C7
B8	IP								BF
BO	P3								B7
A8	IE								AF
AO	P2								A7
98	SCON	SBUF							9F
90	P1								97
88	TCON	TMOD	TLO	TL1	THO	THI			8F
80	PO	SP	DPL	DPH				PCON	87

* resident in 8052/8032 only

Fig. 3.12 : Special Function Registers (SFR) Memory Map [14.b]

A brief description of the function of various registers in the SFR space is follows.

ACC

Acc is the accumulator register.

B REGISTER

This register is used in conjunction with accumulator during multiplication and division operation. During multiplication operation, the operands in Acc and B register are multiplied and the low and high byte of the product is available with Acc and B, respectively. During division operation, the content of Acc is divided by content of B register and the quotient and remainder are available with Acc and B, respectively.

PSW

The program status word register contain the status bits/flags, as given in Fig. 3.11.

SP

Stack pointer register is 8-bits wide and holds the current location of top of the stack. As the stack resides in the internal data memory and the data storage starts from O8H onwards, by default, the SP is loaded with 07H.

DPTR

Data pointer is a 16-bit register, used to hold the 16-bit address of the external program or data memory. External memory access is through DPTR only. It is configured as DPH (high-byte) and DPL (low-byte).

PORT REGISTERS

P0, P1, P2 and P3 are the four port registers, and application point of view, they provide potential interfacing facilities.

TIMER/COUNTER REGISTERS

There are two 16-bit timer/counter registers, Timer 0 and Timer 1 in 8051 (3 in 8052, Timer 2 also). They are configured as TOH & TOL and TIH & TIL, respectively. The required 'Timer' or 'Counter' function is selected by the control bits in the TMOD register, in conjunction with TCON register [14.c].

SERIAL PORT REGISTERS

MCS-51 serial port can transmit and receive the data simultaneously, thus it is full duplex [14.c]. On writing the data to SBUFF - serial buffer register, data is loaded for transmission and on reading, the received data is accessed. Through the serial port control register - SCON, various modes of serial port operation can be programmed [14.c].

INTERRUPT REGISTERS

The 8051 is provided with 5 interrupt sources, while 8052 has 6 [14.c]. The interrupt sources can be enabled or disabled independently through IE - interrupt enable register. They can also be programmed to either one of two priority level through IP - interrupt priority register [14.c].

POWER CONTROL REGISTER

For effective power saving, the CHMOS devices can be made to operate in two reduced power modes [14.c], which are invoked through PCON register by the *software*. In idle (IDL) mode, the oscillator is ON, serial port, interrupt and timer blocks are clocked continuously, but the clock signal to the CPU is gated OFF [14.c]. When compared with the active mode, this mode draws 15% less current[2]. In power down (PD) mode, the oscillator is frozen [14.c], but on-chip RAM holds the data continuously [2]. The current drawn in this mode is 15 μA only [2].

3.5 Architectural Block Diagram

The architectural block diagram of MCS-51 devices is shown in Fig 3.13 [14.c].

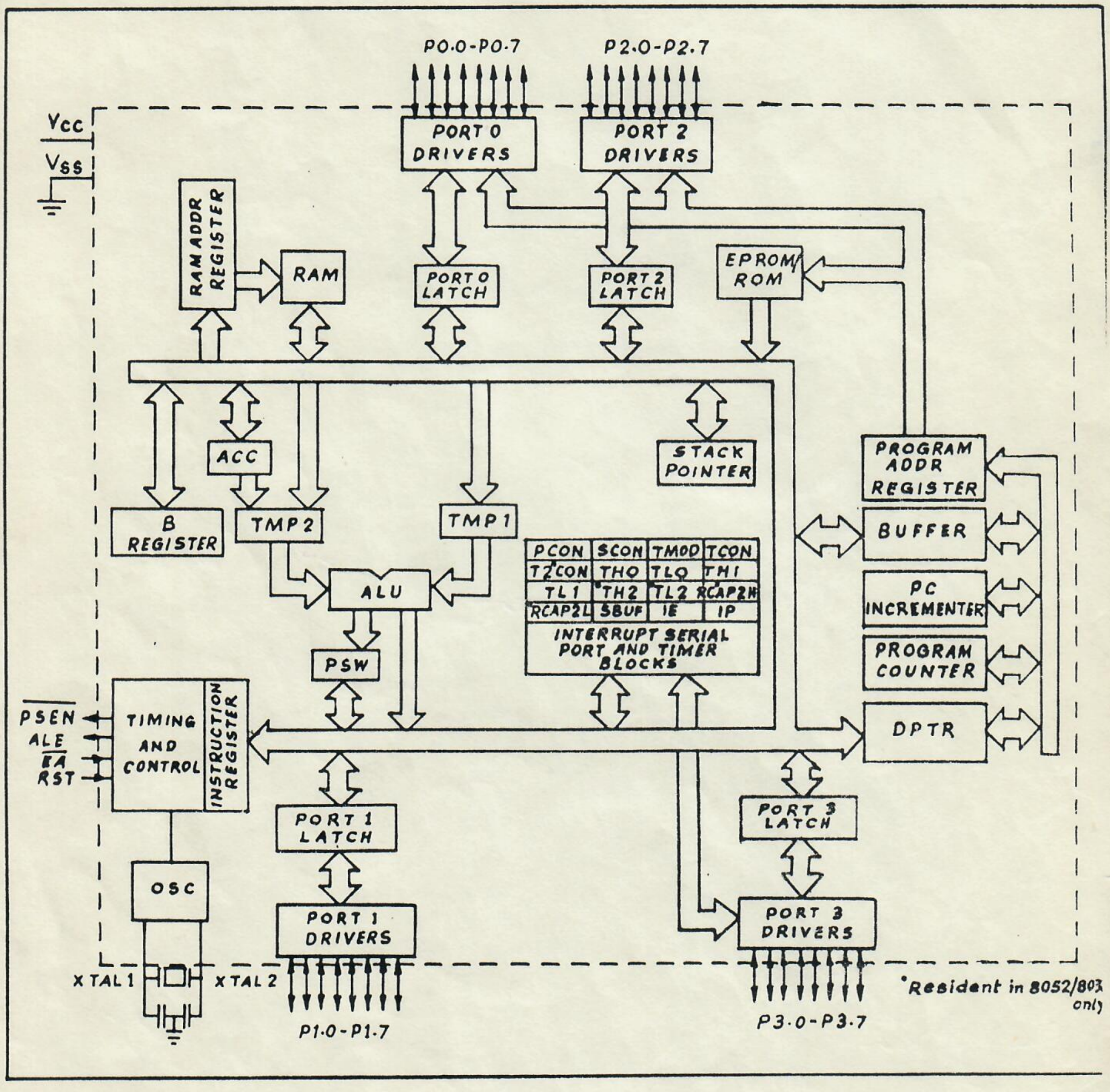


Fig. 3.13 Architectural Block Diagram of MCS-51 [14.C]

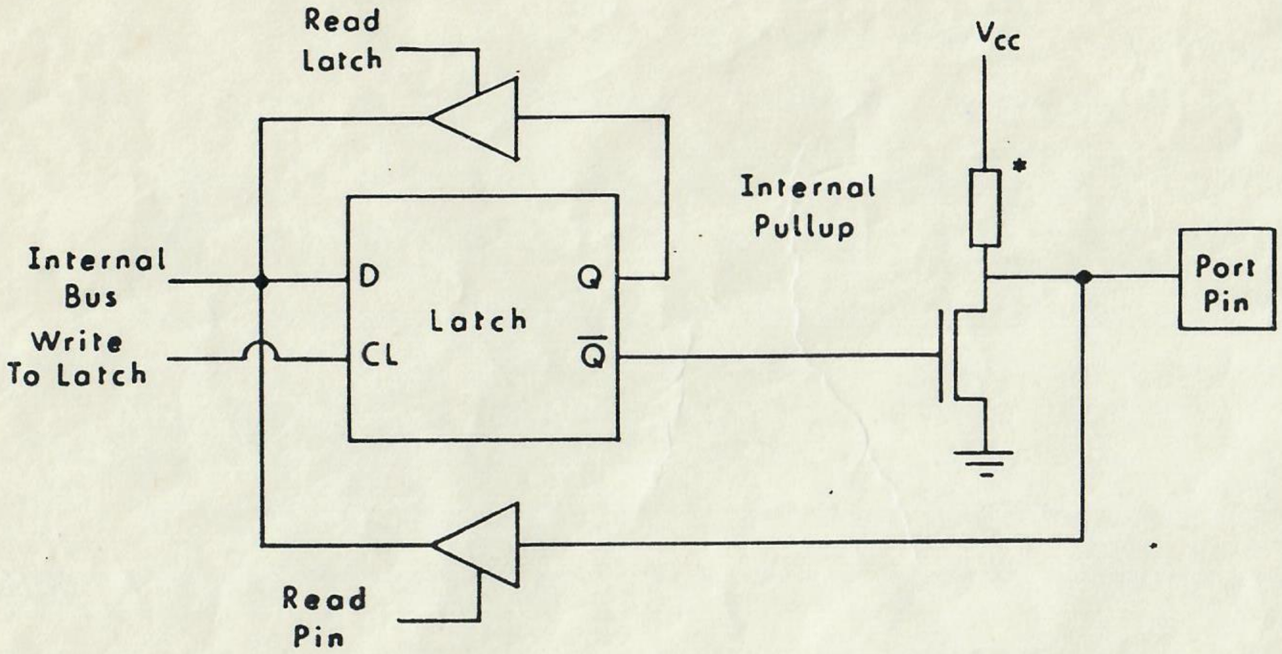
3.6 I/O Port Structure

The simplified internal circuitry of a port pin is shown in Fig 3.14. All the four ports, Po through P3 in 8051 are bidirectional [14.c]. Each port pin consists of a latch, an input buffer and an output driver.

The latch (single port pin) is a Type-D flip-flip. In respose to a 'Write to latch' signal from the CPU, the latch will clock-in the stauts of the internal bus. For a 'read latch' signal from the CPU, flip-flip's Q-output will be placed on the internal bus. In response to the 'read pin' signal from the CPU, the status of port pin itself is returned to the internal bus [14.c].

During 'write' operation to a port pin, the data is loaded from the internal bus into the latch, which drives the FET connected to that port pin. (Drive capability : 8LS TTL load for Port O, 4 LS TTL load for Ports 1,2 and 3)[14.c]. Ports 1,2 and 3 are provided with internal pullups and Port O has open drain output [14.c]. Depending on the input charcteristics of the device driven, an external pull-up resister may be required.

For use as input port, the port latch must have a 1, in order to turn OFF the output driver, otherwise the output will be always low. Thus, in case of Ports 1,2 and 3, the pin is pulled high by the internall pullup. However, it can be pulled low by an external source [14.c].



* Open drain O/p for Port 0 – Provided with Pullup FET

Fig. 3.14 Internal Circuitry of a Port Pin (simplified) [14.C]

GLOSSARY DEFINITION FOR OPEN-DRAIN

Glossary Term: Open-drain

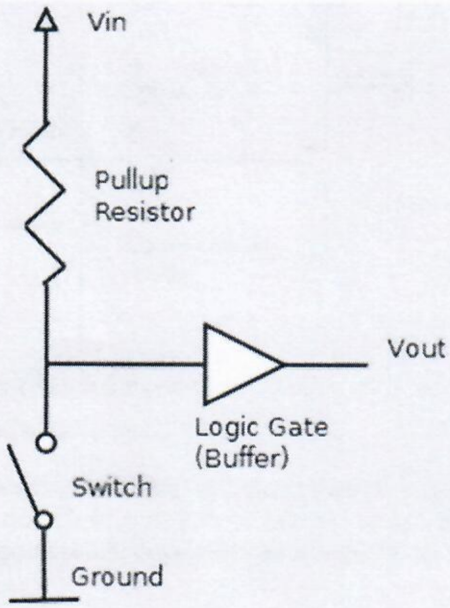
Definition

An open-drain or open-collector output pin is driven by a single transistor, which pulls the pin to only one voltage (generally, to ground). When the output device is off, the pin is left floating (open, or hi-z). A common example is an n-channel transistor which pulls the signal to ground when the transistor is on or leaves it open when the transistor is off.

Open-drain refers to such a circuit implemented in FET technologies because the transistor's drain terminal is connected to the output; open-collector means a bipolar transistor's collector is performing the function.

When the transistor is off, the signal can be driven by another device or it can be pulled up or down by a resistor. The resistor prevents an undefined, floating state. (See the related term, hi-z.)

Pull-up resistor



When the switch is open the voltage of the gate input is pulled up to the level of Vin. When the switch is closed, the input voltage at the gate goes to ground.

In electronic logic circuits, a **pull-up resistor** is a resistor used to ensure a known state for a signal. It is typically used in combination with components such as switches and transistors, which physically interrupt the connection of subsequent components to ground. The pull-up resistor then ensures a well-defined voltage (i.e. V_{CC}) across the latter during interruption.

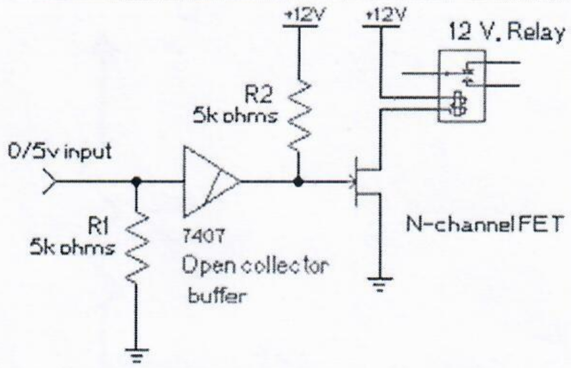
An open switch is not equivalent to a component with infinite impedance, since in the former case, the stationary voltage in any loop in which it is involved can no longer be determined by Kirchhoff's laws. Consequently, the voltages across those *critical* components (such as the logic gate in the example on the right) which are only in loops involving the open switch are undefined, too.

A pull-up resistor effectively establishes an additional loop over the critical components, ensuring that the voltage is well-defined even when the switch is open.

For a pull-up resistor to only serve this one purpose and not interfere with the circuit otherwise, it is assumed that the critical components have infinite or sufficiently high impedance, which is guaranteed for example for logic gates made from FETs. In this case, when the switch is open, the voltage across a pull-up resistor with **sufficiently low** impedance vanishes to the effect that it looks like a wire to VCC. On the other hand, when the switch is closed, the pull-up resistor must have **sufficiently high** impedance in comparison to the closed switch to not affect the connection to ground. Together, these two conditions can be used to derive an appropriate value for the impedance of the pull-up resistor but usually, only a lower bound is derived assuming that the critical components do indeed have infinite impedance.

A **pull-down resistor** works in the same way but is connected to ground. It holds the logic signal at a low logic level when no other active device is connected.

Application



A circuit showing a pull-up resistor (R2) and a pull-down resistor (R1), as well as an open collector(7407) to drive the line to the FET only when given a low 0 V input

A pull-up resistor may be used when interfacing logic gates to inputs. For example, an input signal may be pulled by a resistor, then a switch or jumper strap can be used to connect that input to ground. This can be used for configuration information, to select options or for troubleshooting of a device.

Pull-up resistors may be used at logic outputs where the logic device cannot source current such as open-collector TTL logic devices. Such outputs are used for driving external devices, for a wired-OR function in combinational logic, or for a simple way of driving a logic bus with multiple devices connected to it. For example, the circuit shown on the right uses 0 V logic level inputs to actuate a relay. If the input is left unconnected, pull-down resistor R1 ensures that the input is pulled down to a logic low. The 7407 TTL device, an open collector buffer, simply outputs whatever it receives as input, but as an open collector device, the output is left effectively unconnected when outputting a "1". Pull-up resistor R2 thus pulls the output all the way up to 12 V when the buffer outputs a "1", providing enough voltage to turn the power MOSFET all the way on and actuate the relay.

Pull-up resistors may be discrete devices mounted on the same circuit board as the logic devices. Many microcontrollers intended for embedded control applications have internal, programmable pull-up resistors for logic inputs so that minimal external components are needed.

Some disadvantages of pull-up resistors are the extra power consumed when current is drawn through the resistor and the reduced speed of a pull-up compared to an active current source. Certain logic families are susceptible to power supply transients introduced into logic inputs through pull-up resistors, which may force the use of a separate filtered power source for the pull-ups.

Pull-down resistors can be safely used with CMOS logic gates because the inputs are voltage-controlled. TTL logic inputs that are left un-connected inherently float high, and require a much lower valued pull-down resistor to force the input low. A standard TTL input at logic "1" is normally operated assuming a source current of 40 µA, and a voltage level above 2.4 V, allowing a pull-up resistor of no more than 50 kohms; whereas the TTL input at logic "0" will be expected to sink 1.6 mA at a voltage below 0.8 V, requiring a pull-down resistor less than 500 ohms.[1] Holding unused TTL inputs low consumes more current. For that reason, pull-up resistors are preferred in TTL circuits.

In bipolar logic families operating at 5 VDC, a typical pull-up resistor value will be 1000–5000 Ω, based on the requirement to provide the required logic level current over the full operating range of temperature and supply voltage. For CMOS and MOS logic, much higher values of resistor can be used, several thousand to a million ohms, since the required leakage current at a logic input is small.

Instead of internal pullup, Port 0 is provided with a pullup FET, which is used only when Porto is outputting 1's during external memory accesses. Otherwise, the FET (pullup) is OFF only. Thus Port O lines, when used as an output port is *open drain*. As both the FET's are OFF when the bit latch has a 1, the pin *floats* [14.c].

When configured as input port, Port O floats while Port 1,2 and 3 are pulled high and will source current, provided it is pulled low externally. On this account, Port O is considered as 'true'bidirectional, while ports 1,2 and 3 are called 'quasi-bidirectional' [14.c].

On a system reset, all the 8051's port latches have a 1's written to them. Subsequently when a O is written to a port latch means, in order to reconfigure the port again as an input port, a 1 has to be written to the latch [14.c].

When alternate functions are in effect, the output drivers are switched to an internal address line for Port 2, address / data line for Port O and appropriate control line for Port 3 by a multiplexer (MUX) arrangement, which is not shwon in Fig 3.14.

READ - MODIFY - WRITE OPERATION

For reading a port pin, some instructions read the *latch* and some other read the *pin*. 'Read-modify-write' instruction are those which read a *value*, probably change it and rewrite it to the latch. These instructions, read the *latch* instead of *pin*, when the destination operand happens to be a port / port bit. (e.g.) CPL PI.5. The reason is to avoid misinterpretation of voltage level at the pin. For example, if a Port bit is used to drive a tranistor. Then the base of the transistor is connected to the Port Pin and the transistor is ON, when a 1 is emitted to that Port pin. If the same port bit is read at the *pin*, instead of at *latch*, then the CPU will read the base voltage and interpret it as a 0. On the other hand, the correct value of 1 will be returned, when the *latch* is read [14.c]. Instructions that input a Port bit, such as MOV C, P1. O, read the *pin*.

3.7 Summary

The MCS-51 Family members and their Features were presented in this chapter. The hardware details of MCS-51 devices such as Pin Diagram, Memory Organization, Architectural Block Diagram and I/O Port Structure were also discussed in this chapter.