National Education Society (R.)
Jawaharlal Nehru National College of Engineering, Shivamogga
(Approved by AICTE, New Delhi, Certified by UGC 2f & 12B, Accredited by NAAC –'A',
Recognized by Govt. of Karnataka and Affiliated to VTU, Belagavi)

Microcontroller
BEE403

Module 1

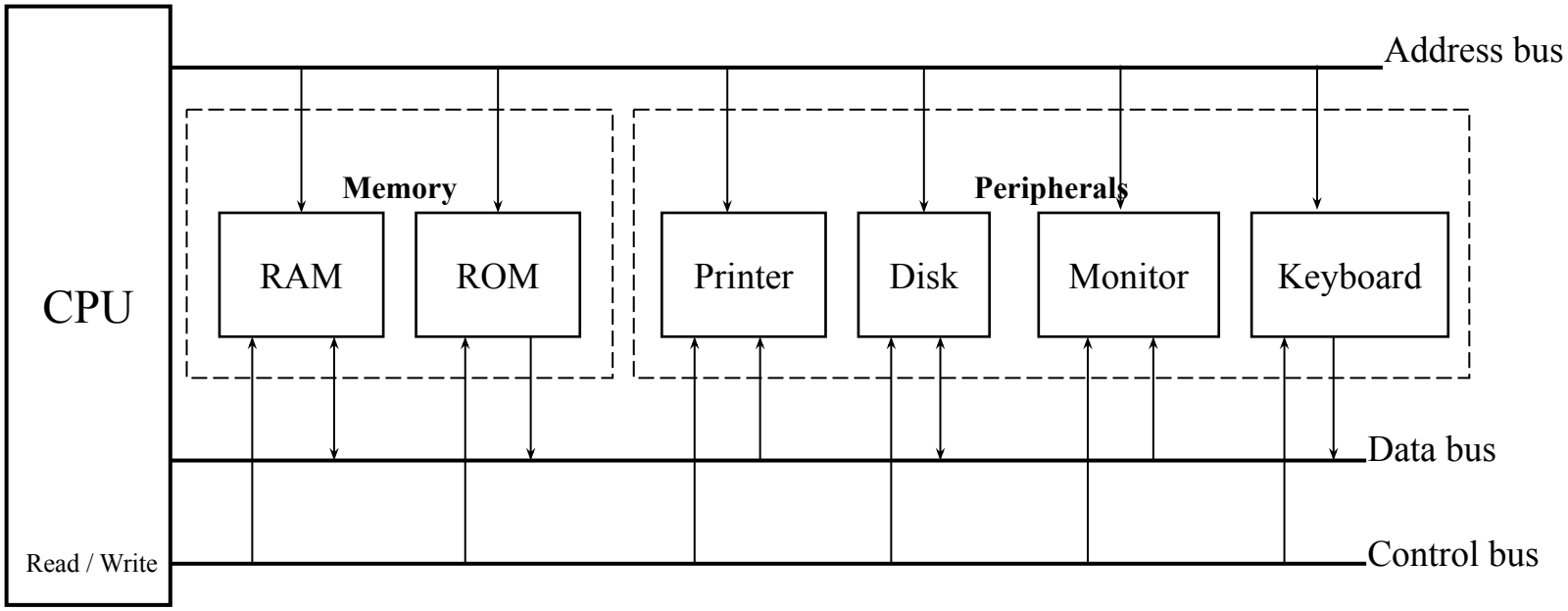
MODULE – 1: 8051 MICROCONTROLLER BASICS

Syllabus:

- Inside the Computer
- Microcontrollers and Embedded Processors
- **Block Diagram of 8051**
- **PSW and Flag Bits**
- **8051 Register Banks and Stack**
- **Internal Memory Organization of 8051**
- **Types of Special Function Registers and their uses in 8051**
- Pins Of 8051
- IO Port Usage in 8051
- Memory Address Decoding
- 8031/51 Interfacing With External ROM and RAM
- 8051 Addressing Modes.

Session 1

Inside the Computer



- CPU (Central Processing Unit)
- Memory
- I/O (input/ output)
- Bus – Address bus, Data bus, Control bus

Data bus:

- Data lines are used to carry information in and out of a CPU
- The more data lines available, the better the CPU and its grouping is called data bus.
- More data buses mean a more expensive CPU and computer.
- The average size of data buses in CPUs varies between 8-bit and 64-bit.
- Data buses are bidirectional, since the CPU must use them either to receive or to send data.
- The processing power of a CPU is related to the size of its buses, an 8-bit bus can send out 1 byte a time, but a 16-bit bus can send out 2 bytes at a time, which is twice as fast.

Address bus:

- The address bus is used to identify the devices and memory connected to the CPU
- The more address lines available, larger the number of devices that can be addressed. i.e., The number of locations is always equal to 2^x where x is the number of address lines, regardless of the size of the data bus.
- The address bus is a unidirectional bus, which means that the CPU uses the address bus only to send out addresses.

Control Bus:

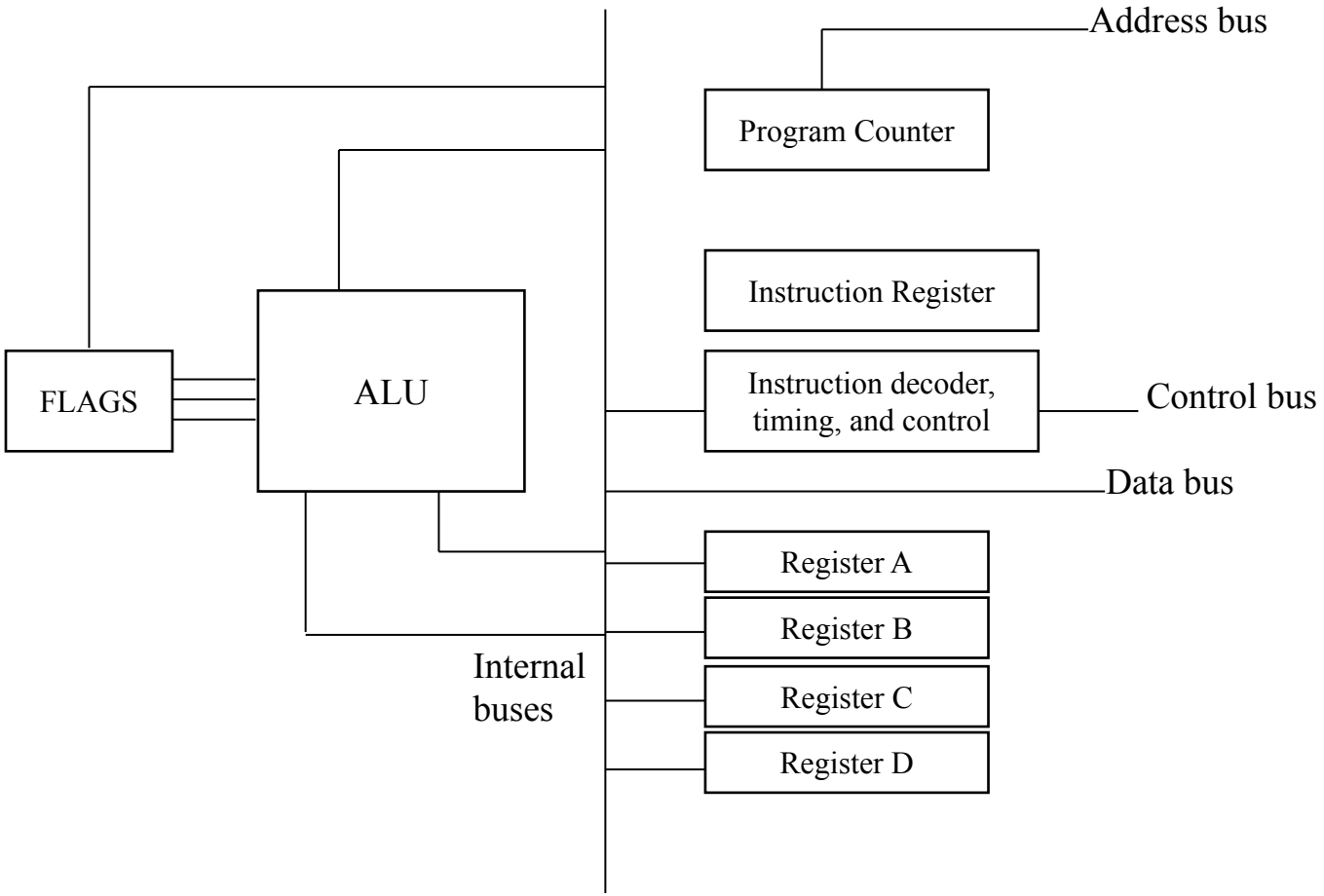
The control buses are used to provide read or write signals to the device to indicate if the CPU is asking for Information or sending it information.

CPU and its relation to RAM and ROM

- For the CPU to process information, the data must be stored in RAM or ROM.
- The function of ROM in computers is to provide information that is fixed and permanent.
- In contrast, RAM is used to store information that is not permanent and can change with time.
- RAM and ROM are sometimes referred to as primary memory and disks are called secondary memory.

Session 2

Inside CPU


Address bus
Program Counter
Instruction Register
FLAGS
ALU
Instruction decoder, timing, and control
Control bus
Data bus
Register A
Register B
Internal buses
Register C
Register D

❖ Number of registers:

- ❑ The CPU uses registers to store information temporarily.
- ❑ Registers inside the CPU can be 8-bit, 16-bit, 32-bit, or even 64-bit registers, depending on the CPU.
- ❑ In general, the more and bigger the registers, the better the CPU.
- ❑ The disadvantage of more and bigger registers is the increased cost of such a CPU

❖ ALU (Arithmetic Logic Unit):

The ALU section of the CPU is responsible for performing arithmetic functions such as add, subtract, multiply, and divide, and logic functions such as AND, OR, and NOT.

❖ Program Counter (PC):

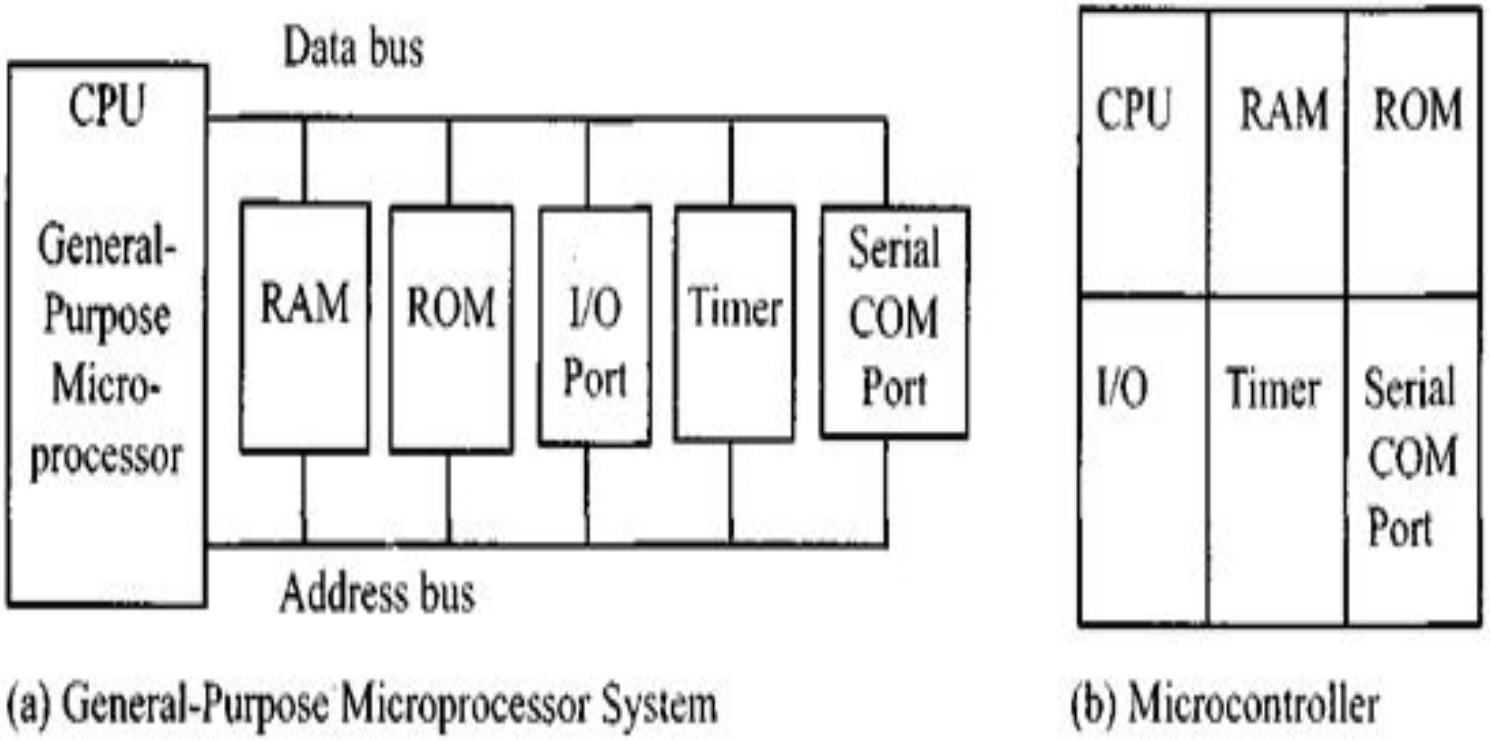
- ❑ The function of PC is to point to the address of the next instruction to be executed.
- ❑ As each instruction is executed, the program counter is incremented to point to the address of the next instruction to be executed.
- ❑ The contents of the PC are placed on the address bus to find and fetch the desired instruction.
- ❑ The program counter is also called as IP (Instruction pointer).

❖ Instruction Decoder:

- ❑ The function of the instruction decoder is to interpret the instruction fetched into the CPU.
- ❑ A CPU capable of understanding more instructions requires more transistors to design.

Microcontrollers and Embedded Processors

Microcontroller versus general- purpose microprocessor:



(a) General-Purpose Microprocessor System

(b) Microcontroller

Microcontrollers and Embedded Processors

Microcontroller versus general- purpose microprocessor:

Sl. No.	General purpose processors (GPP)	Microcontrollers
1	Do not contain RAM, ROM, and I/O ports on the chip itself.	Has a CPU (a microprocessor) in addition to a fixed amount of RAM, ROM, I/O ports, and a timer all on a single chip.
2	A system designer using a general-purpose microprocessor must add RAM, ROM, I/O ports, and timers externally to make them functional.	Designer need not add any external memory, I/O, or timer to it. ADC and many other peripherals are integrated inside.
3	Addition of external RAM, ROM, and I/O ports makes these systems bulkier and much more expensive.	Since memory & peripherals are integrated on chip, systems are smaller and cheaper.
4	GPP have the advantage of versatility such that the designer can decide on the amount of RAM, ROM, and I/O ports needed to fit the task at hand.	The fixed amount of on-chip ROM, RAM, and number of I/O ports in microcontrollers makes them ideal for many applications in which cost and space are critical.
5	Intel's x86 family (8086, 80286, 80386, 80486, and the Pentium) or Motorola's 680×0 family (68000, 68010, 68020, 68030, 68040, etc.)	8051 family, PIC 16F8X, Hitachi H8, 68HC11xx, etc.

Microcontrollers and Embedded Processors

Choosing a microcontroller:

1) **Meeting the computing needs of the task at hand efficiently and cost effectively**
   - ***Data handling size:*** an 8-bit, 16-bit, or 32-bit microcontroller can best handle the computing needs of the task most effectively
   - ***Speed:*** highest speed that the microcontroller supports?
   - ***Packaging:*** Does it come in a 40-pin DIP (dual inline package) or a QFP (quad flat package), or some other packaging format? This is important in terms of space, assembling, and prototyping the end product.
   - ***Power consumption:*** This is especially critical for battery-powered products.
   - The amount of RAM and ROM on chip.
   - The number of I/O pins and the timer on the chip.
   - How easy it is to upgrade to higher-performance or lower power-consumption versions.
   - Cost per unit. This is important in terms of the final cost of the product in which a microcontroller is used. For example, there are microcontrollers that cost 50 cents per unit when purchased 100,000 units at a time.

Microcontrollers and Embedded Processors

Choosing a microcontroller:

2) Availability of software development tools such as compilers, assemblers, and debuggers.

- Key considerations include the availability of an assembler, debugger, a code-efficient C language compiler, emulator, technical support, and both in-house and outside expertise.
- In many cases, third-party vendor (that is, a supplier other than the chip manufacturer) support for the chip is as good as, if not better than, support from the chip manufacturer.

Microcontrollers and Embedded Processors

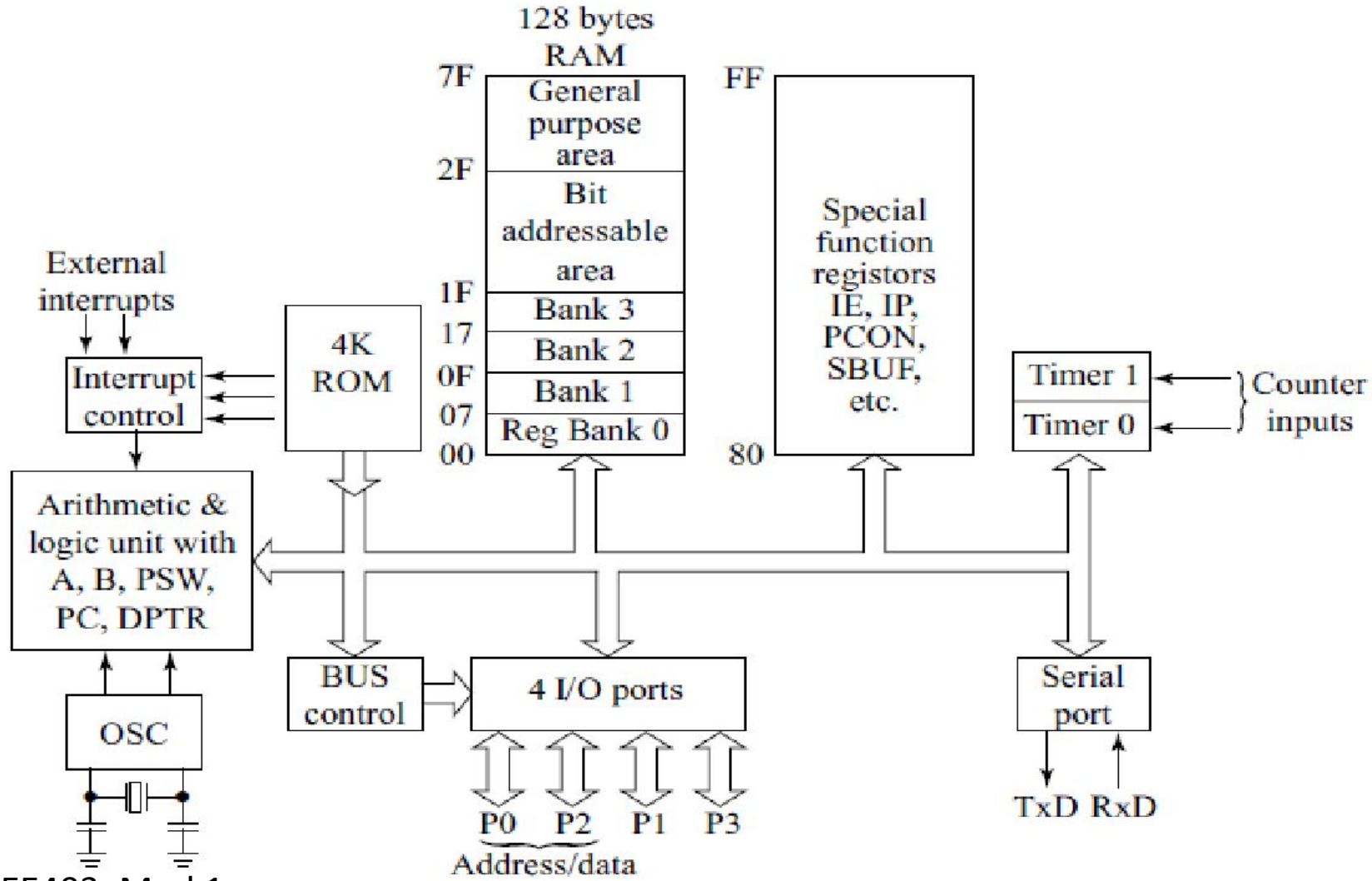
Choosing a microcontroller:

3) Wide availability and reliable sources of the microcontroller

- Ready availability in needed quantities both now and in the future.
- The 8051 family has the largest number of diversified (multiple source) suppliers. By supplier is meant a producer besides the originator of the microcontroller.
- In the case of the 8051, which was originated by Intel, several companies also currently produce (or have produced in the past) the 8051. These companies include: Intel, Atmel, Philips/Signetics, AMD, Infineon (formerly Siemens), Matra, and Dallas Semiconductor.

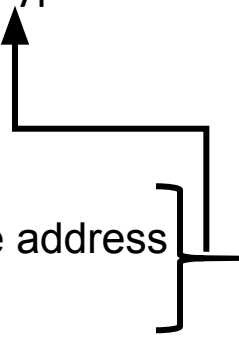
Session 3

Block diagram of 8051



Features of 8051

- **Harvard Vs Vonneumann** – memory access
- **CISC Vs RISC** – No. & type of Instructions



- An 8-bit ALU with A & B Registers, 8-bit PSW
- 16-bit address and 8-bit data bus.
- 16-bit Program Counter (PC) & Data Pointer (DPTR).
- 8-bit Stack Pointer (SP), initial default value is 07h.
- Harvard memory architecture. The program memory and data memory have separate address spaces from 0000h and separate control signals.
- CISC (Complex Instruction Set Computer) architecture.
- Special bit manipulation instructions.
- Clock and oscillator circuit.
- *Internal ROM of 4 KB, 8751–EPROM; 8951–EEPROM; 8031–0 bytes. Extendable up to 64 KB.*
- *Internal RAM of 128 bytes, Extendable up to 64 KB.*
  - *Four register banks, each containing 8 registers (32 bytes)*
  - *16 bytes bit addressable memory*
  - *80 bytes of general purpose data memory*
- *Two 16-bit timers/counters: T0 & T1*
- *Two external interrupts INT0 & INT1 and three internal interrupts T0, T1 & SI.*
- *32 I/O pins arranged as 04 8-bit ports: P0 - P3.*
- *Full duplex UART Serial interface.*
- *SFR: TCON, TMOD, SCON, PCON, SBUF, IP & IE*

Arithmetic & Logic Unit (ALU):

- The 8-bit ALU can perform arithmetic operations like addition, subtraction, multiplication & division and logical operations like OR, AND, XOR, etc.

A & B CPU registers:

- 'A' & 'B' registers holds the results of many isntructions, particularly math & logical operations.
- 'A' register called as **accumulator** is the most versatile of the two CPU registers & is used for many operations, including addition, subtraction, integer multiplication and division, and boolean bit manipulations.
- 'A' register is also used for all data transfers between 8051 and the external memory.
- 'B' register is used with the 'A' register for multiplication and division opeartions and has no other function other than as a location where data may be stored.

Session 4

PSW and Flag bits:

CY	AC	F0	RS1	RS0	OV	--	P

CY	PSW.7	Carry flag; Sets or resets based on carry/borrow generated during arithmetic or logical operations.
AC	PSW.6	Auxiliary carry flag; sets or resets based on carry/borrow generated b/w lower & higher nibble of a byte.
F0	PSW.5	Available to the user for general purpose.
RS1	PSW.4	Register Bank selector bit1.
RS0	PSW.3	Register Bank selector bit 0.

RS1	RS0	Register Bank	Address
0	0	0	00H -07H
0	1	1	08H-0FH
1	0	2	10H – 17H
1	1	3	18H-1FH

OV	PSW.2	Overflow flag.
--	PSW.1	User-definable bit.
P	PSW.0	Parity flag. Set/cleared by hardware each instruction cycle to indicate an odd/even number of 1 bits in the accumulator. 1 – Odd parity, 0 – Even parity

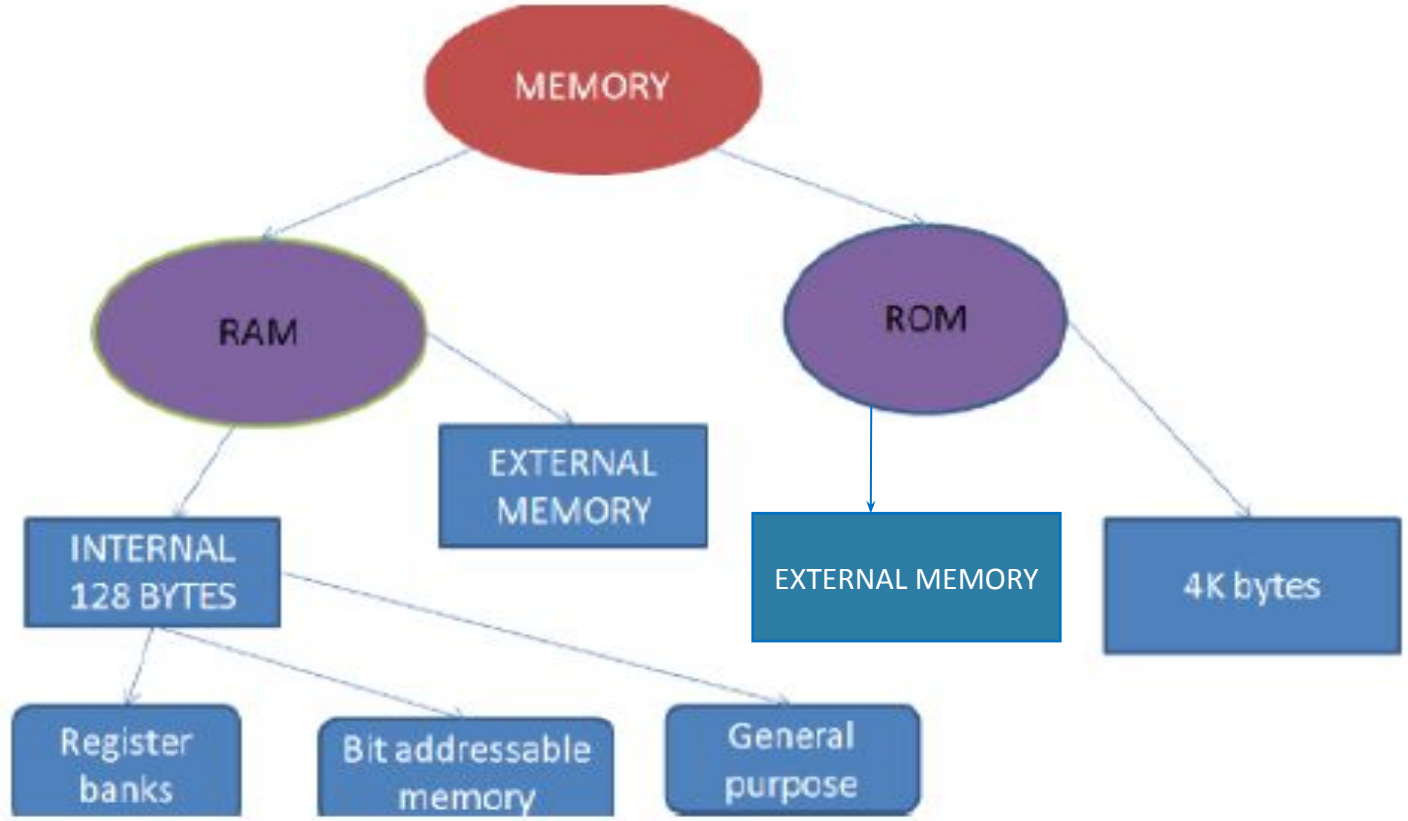
Program Counter (PC):

- PC is a 16-bit register which addresses the instruction bytes that are to be fetched from locations in program memory.
- The PC is automatically incremented after every instruction byte is fetched and may also be altered by certain instructions.
- The PC is the only register that doesn't have an internal address.

Data Pointer (DPTR):

- DPTR is a 16-bit register made up of two 8-bit registers, named DPH & DPL.
- used to furnish memory addresses for **internal & external code access & data access**.
- can be specified by its 16-bit name DPTR or by each individual byte name DPH & DPL.
- DPTR doesn't have a single internal address; DPH & DPL are each assigned an address.

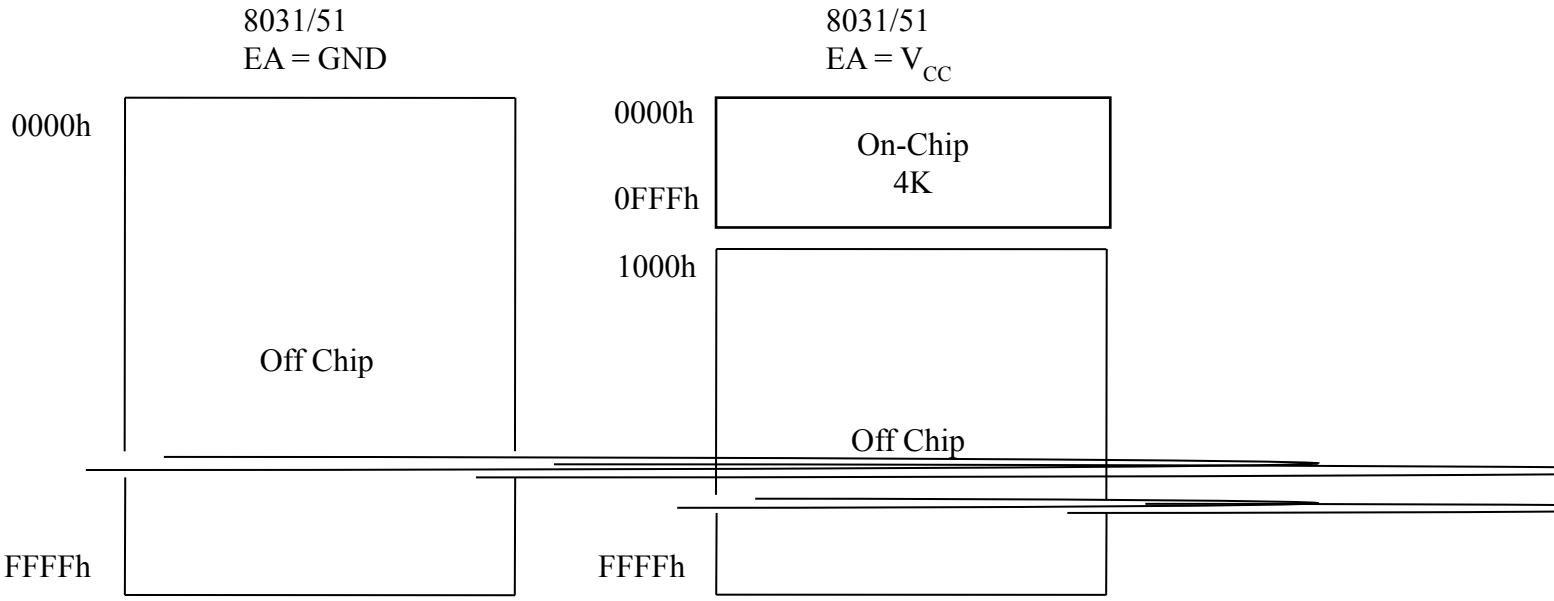
Internal Memory Organization of 8051



- Internal ROM of 4K bytes for code or program.
- Internal RAM of 128 bytes for data and Special Fucntion Registers (SFR) area of 128 bytes of RAM.

Internal ROM:

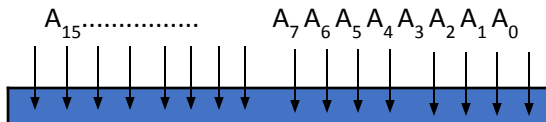
- The 8051 has 4K bytes of internal ROM, having the address space 0000h to 0FFFh.
- Program code address higher than 0FFFh, which exceed the internal ROM capacity, will cause the 8051 to automatically fetch code bytes from external program memory.
- Code bytes can also be fetched exclusively from an external memory addresses 0000h to FFFFh by connecting the external access pin (EA – pin 31) to ground.



Internal RAM Organization

Hex byte address	Hex bit address								Notes
7F 30	Directly and indirectly addressable general purpose RAM								Used as a STACK area
2F	7F	7E	7D	7C	7B	7A	79	78	
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	Bit addressable section
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	Register bank 3 (R0 - R7)								Bank is selected using RS0 and RS1 in the PSW register. See SFRs.
17 10	Register bank 2 (R0 - R7)								
0F 08	Register bank 1 (R0 - R7)								
07 00	Register bank 0 (R0 - R7)								

Address lines



x x x x x x x x 1 1 1 1 1 1 1 1	xxFFh								
xxxx xxxx 1111 1110	xxFEh								
xxxx xxxx 1111 1101	xxFDh								
xxxx xxxx 1111 1100	xxFCh								
	⋮	⋮	⋮			⋮			
xxxx xxxx 0000 0111	xx07h								
xxxx xxxx 0000 0110	xx06h								
xxxx xxxx 0000 0101	xx05h								
xxxx xxxx 0000 0100	xx04h								
xxxx xxxx 0000 0011	xx03h								
xxxx xxxx 0000 0010	xx02h								
xxxx xxxx 0000 0001	xx01h								
xxxx xxxx 0000 0000	xx00h								
Binary Addr.	Hexa.	07h	06h	05h	04h	03h	02h	01h	00h

Data lines

8051 Register banks and stack

Register banks in the 8051

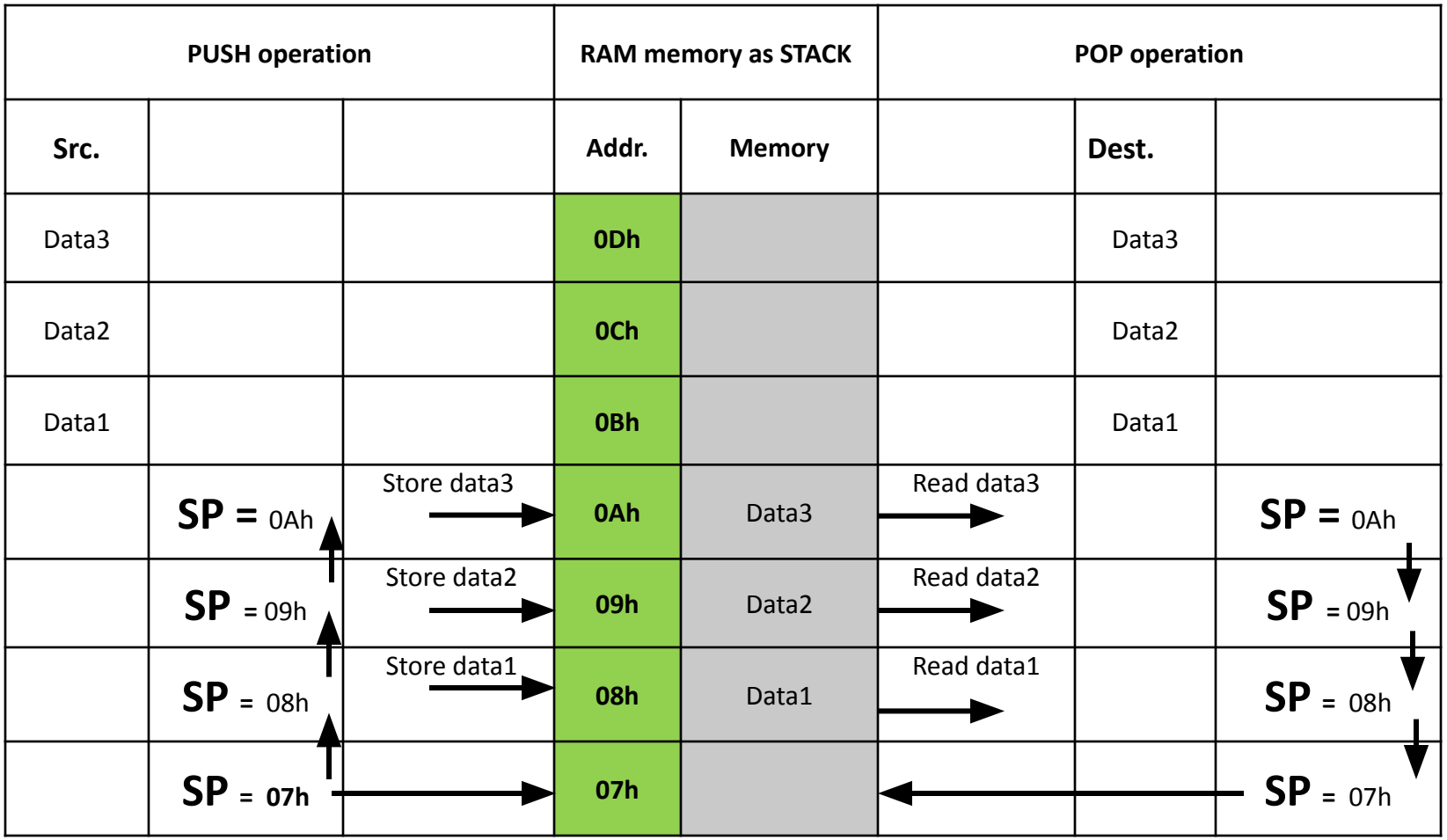
- The first 32 bytes of RAM starting from 00H to 1FH are divided into 4 banks of registers (Bank 0 to Bank 3) in which each bank has 8 registers, R0 - R7.
- The default register bank on reset will be Bank 0.
- Switching of register banks from one to other can be done through bits PSW.3 & PSW.4 of Program Status Word.

Stack in the 8051

- The stack is a section of RAM used by the CPU to store information temporarily.
- This information could be data or address.
- The CPU needs this storage area since there are only a limited number of registers.

How stacks are accessed in the 8051:

- Stack in 8051 is accessed by using 8-bit pointer register called Stack Pointer (SP) and the operations PUSH & POP.
- When the 8051 is powered up, the SP register contains value 07.
- The *Storing* of a CPU register in the stack is called a PUSH operation.
- The *Pulling* the contents off the stack back into a CPU register is called a POP.


PUSH operation
RAM memory as STACK
POP operation
Src.
Addr.
Memory
Dest.
Data3
0Dh
Data3
Data2
0Ch
Data2
Data1
0Bh
Data1
Store data3
Read data3
SP = 0Ah
0Ah
Data3
SP = 0Ah
Store data2
Read data2
SP = 09h
09h
Data2
SP = 09h
Store data1
Read data1
SP = 08h
08h
Data1
SP = 08h
SP = 07h
07h
SP = 07h

8051 Register banks and stack

Stack in the 8051

Pushing onto the stack:

- In the 8051 the stack pointer (SP) points to the last used location of the stack.
- For every PUSH operation, the SP is incremented by 1 and then the contents of the register are saved on the stack.
- To PUSH the registers onto the stack we must use their RAM addresses.

Popping from the stack:

- Popping the contents of the stack back into a given register is the opposite process of pushing.
- With every pop, the top byte of the stack is copied to the register specified by the instruction and the stack pointer is decremented once.

The upper limit of the stack:

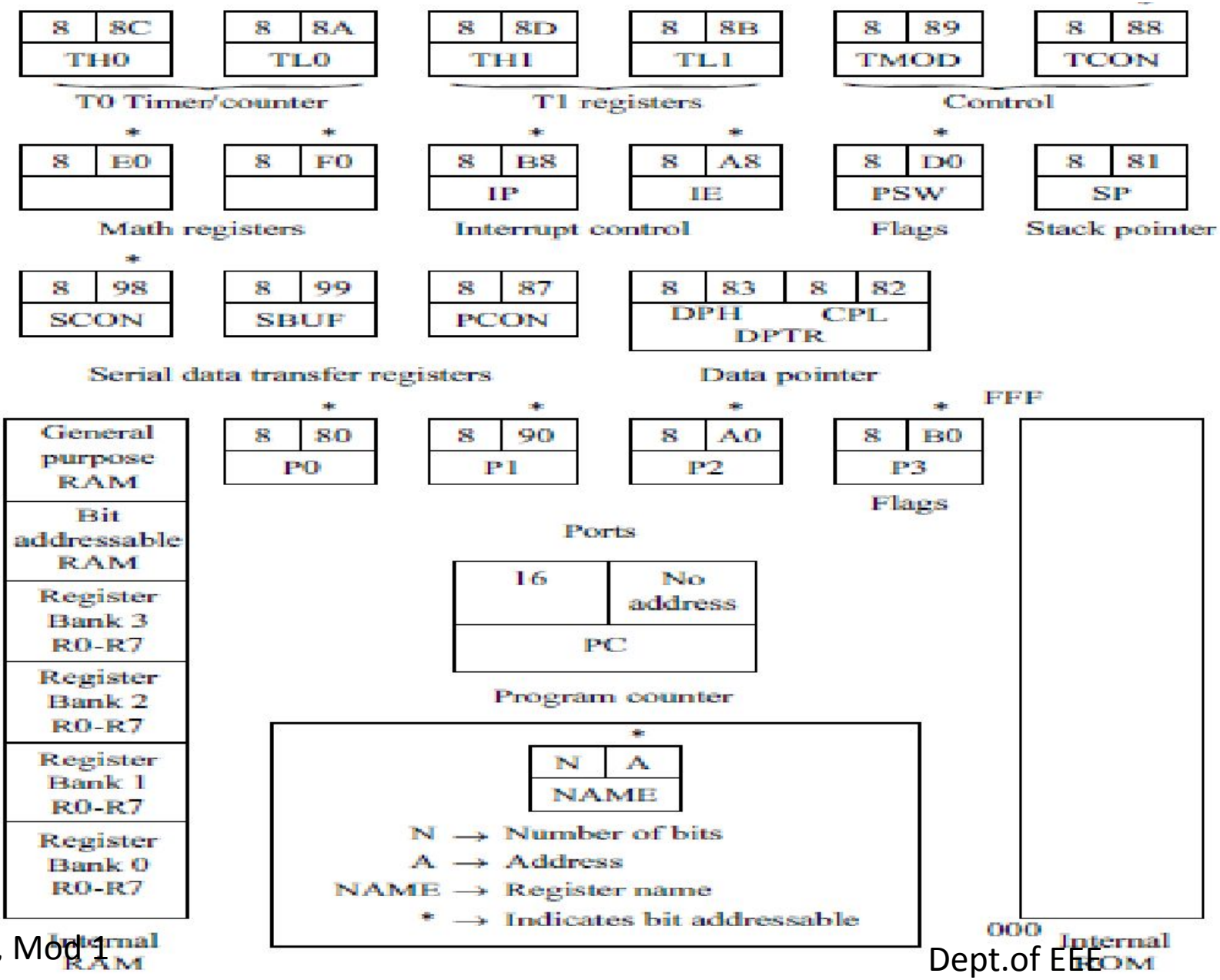
- Locations 08h to 1Fh and 30h to 7Fh in the 8051 RAM can be used for the stack.
- Locations 20h-2Fh of RAM are reserved for bit-addressable memory and must not be used by the stack.
- Stack location can be changed to a desired location by changing the address in stack pointer (SP).

Session 5

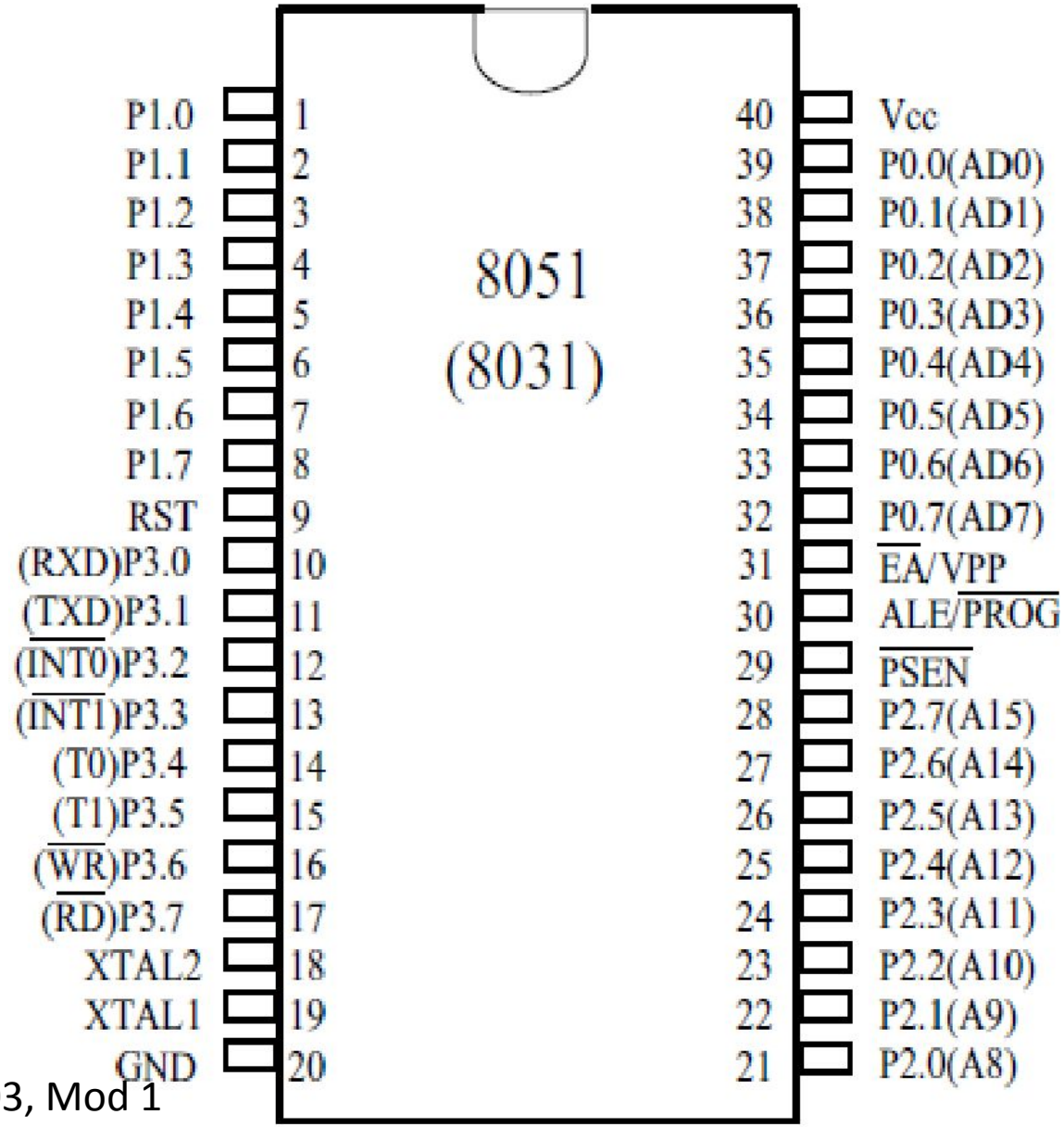
Special Function Registers (SFR) and their addresses

Symbol	Name		Address
ACC*	Accumulator		0E0H
B*	B register		0F0H
PSW*	Program status word		0D0H
SP	Stack pointer		81H
DPTR	Data pointer 2 bytes		
	DPL	Low byte	82H
	DPH	High byte	83H
P0*	Port 0		80H
P1*	Port 1		90H
P2*	Port 2		0A0H
P3*	Port 3		0B0H
IP*	Interrupt priority control		0B8H
IE*	Interrupt enable control		0A8H
TMOD	Timer/ counter mode control		89H
TCON*	Timer /counter control		88H
TH0	Timer / counter 0 high byte		8CH
TL0	Timer/ counter 0 low byte		8AH
TH1	Timer/ counter 1 high byte		8DH
TL1	Timer/ counter 1 low byte		8BH
SCON*	Serial control		98H
SBUF	Serial data buffer		99H
PCON	Power control		87H

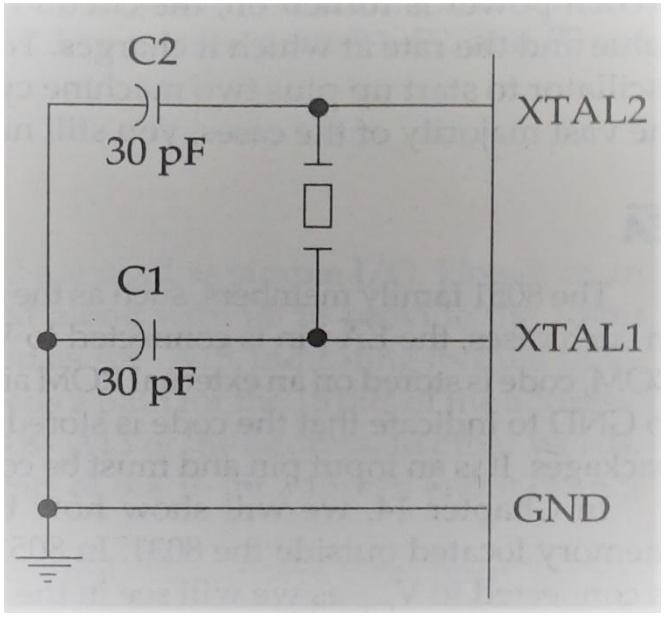
Programming model of 8051

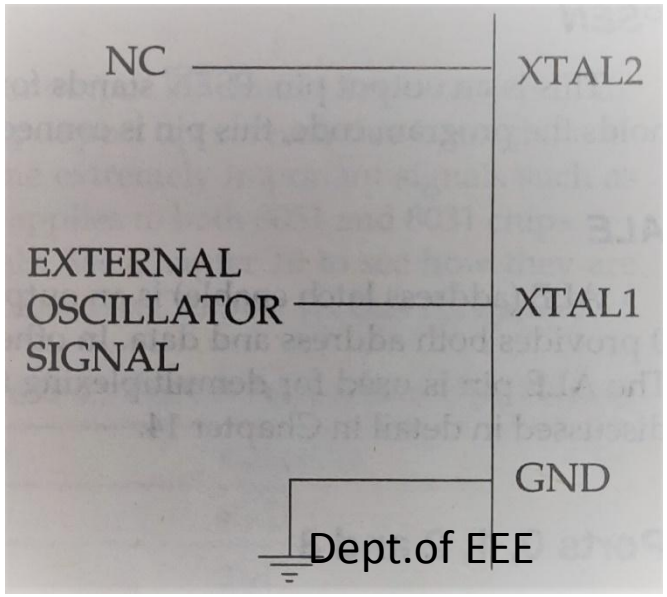


Pin Diagram of 8051 microcontroller

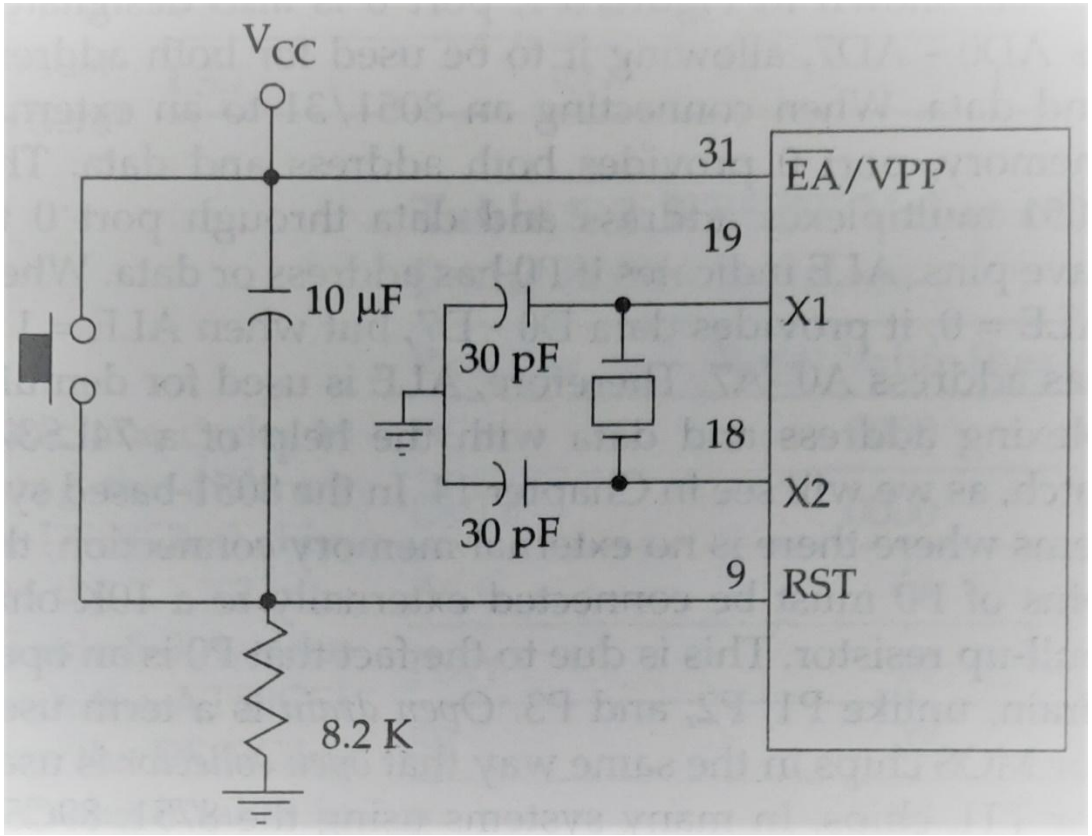


Clk i/p





Reset Circuit



Pin Details of 8051 microcontroller

8051 pins & its functions				
Pin nos.	**Pin names**	**Function**		
Pin 1 to 8	Port 1 (P1.0 to P1.7)	It has 8 bidirectional I/O lines used only for I/O purpose.		
Pin 9	RST	Reset I/P, +5V applied across this pin causes the microcontroller to restart.		
Pin 10 to pin 17	Port 3 (P3.0 to P3.7)	Port 3 is multifunctional. It is used as I/O and other function for individual port pin.		
		Pin 10 (P3.0)	RxD	Acts as serial data receiver
		Pin 11 (P3.1)	TxD	Used as serial data transmitting
		Pin 12 (P3.2)	/INT0	External interrupt 0
		Pin 13 (P3.3)	/INT1	External interrupt 1
		Pin 14 (P3.4)	T0	Clock input for Timer 0
		Pin 15 (P3.5)	T1	Clock input for Timer 1
		Pin 16 (P3.6)	/WR	Write signal for external RAM
		Pin 17 (P3.7)	/RD	Read signal for external RAM
Pin 18 & 19	XTAL2 & XTAL1	Clock input pins to connect internal oscillator circuit		
Pin 20	GND (Vss)			

Pin Details of 8051 microcontroller

8051 pins & its functions		
Pin nos.	**Pin names**	**Function**
Pin 21 to 28	Port 2 (P2.0 to P2.7) [A8 to A15]	Port2 is multifuncitonal. It can be used as I/O as well as higher order address lines (A8 – A15) for external memory if connected to 8051
Pin 29	PSEN	Program store enable pin is used in conjunction with EA pin to read external ROM if connected
Pin 30	ALE/PROG	Address Latch Enable signal is used to demultiplex address & data during external memory access. Other function is EPROM programming pulse during program code buring into ROM
Pin 31	EA/Vpp – External Access/EPROM programming voltage.	External access pin is used to access exernal memory if connected, if not connected it is connected to Vpp. EA is made high for external memory access
Pin 32 to 39	Port 0 / Address/data (P0.0 to P0.7 /AD0 to AD7)	Multifunctional pins can be used as I/O or address/data multiplexed lines if exernal memory is connected. Multiplexed lower address lines & data lines can be demultiplexed using ALE signal
Pin 40	Vcc	Power supply of +5V

Session 6

IO port usage in 8051

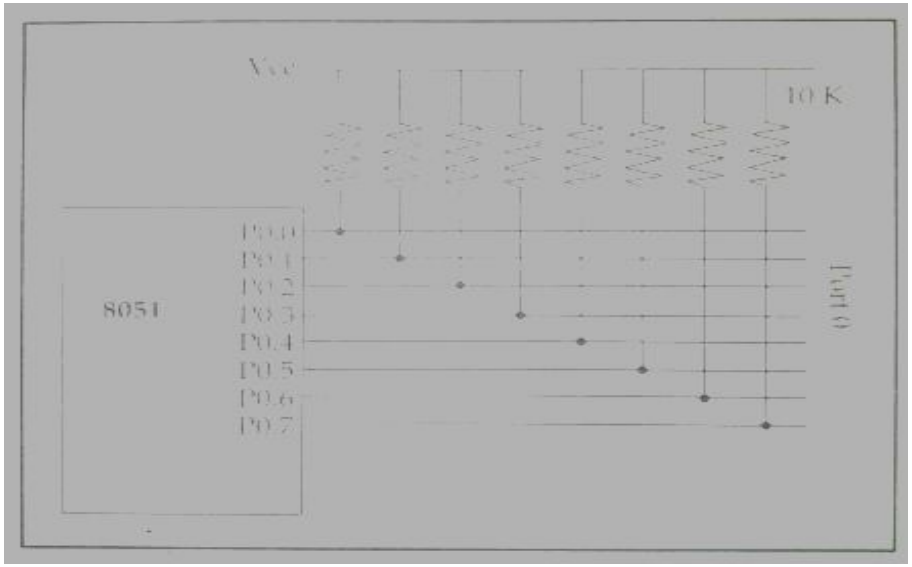
I/O port pins and their functions:

- The 04 ports P0, PI, P2, and P3 each use 8 pins, making them 8-bit ports.
- To use any of these ports as an input port, it must be programmed.
- All the ports upon RESET are configured as outputs, ready to be used as output ports.
- To configure as an input, a 1 must be sent to the port.

IO port usage in 8051

Port 0:

- Port 0 occupies a total of 8 pins (pins 32 - 39). It can be used for input or output.
- each pin must be connected externally to a 10K-ohm pull-up resistor to use port 0 as both input & output. This is due to the fact that P0 is an open drain, unlike P1, P2, and P3.
- **Ports 0 as input:**
  With resistors connected to port 0, in order to make it an input, the port must be programmed by writing 1 to all the bits.
- **Dual role of port 0:**
  Port 0 is also designated as AD0 - AD7, allowing it to be used for both address and data.



IO port usage in 8051

Port 1

- Port 1 occupies a total of 8 pins (pins 1 to 8). It can be used as input or output.
- In contrast to port 0, this port does not need any pull-up resistors since it already has pull-up resistors internally.
- Upon reset, port 1 is configured as an output port.
- If port 1 has been configured as an output port, to make it an input port again, it must be programmed as such by writing 1 to all its bits.

Port 2

- Port 2 occupies a total of 8 pins (pins 21 to 28). It can be used as input or output.
- Just like P1, port 2 does not need any pull-up resistors since it already has pull-up resistors internally.
- On reset, port 2 is configured as an output port.
- To make port 2 an input, it must programmed as such by writing 1 to all its bits.
- Port 2 is also designated as A8 - A15, indicating its dual function. When the 8051/31 is connected to external memory, P2 is used for the upper 8 bits of the 16-bit address, and it cannot be used for I/O.

IO port usage in 8051

Port 3

- Port 3 occupies a total of 8 pins, pins 10 through 17. It can be used as input or output.
- P3 does not need any pull- up resistors, just as P1 and P2 did not.
- Port 3 has the additional function of providing some extremely important signals such as interrupts.

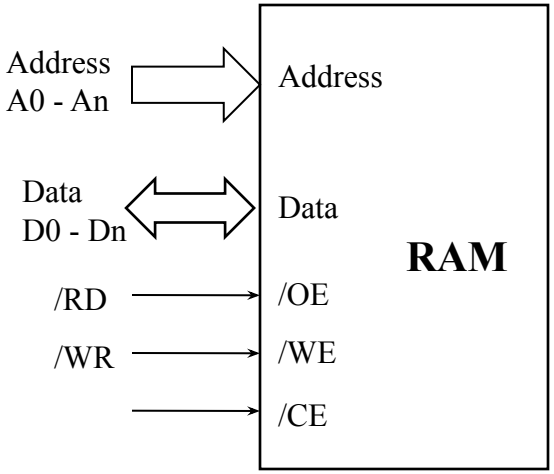
Port 3 Alternate Function

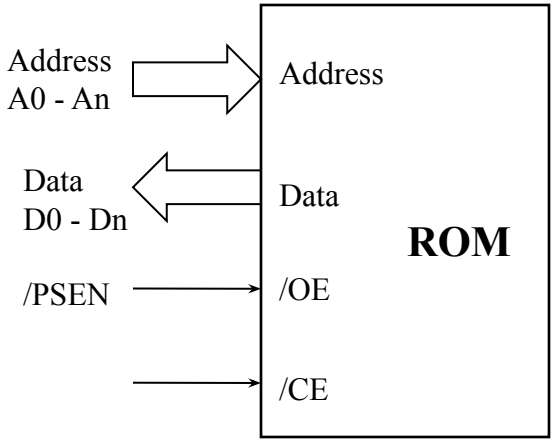
P3 Bit	Function	Pin
P3.0	RxD	10
P3.1	TxD	11
P3.2	/INT0	12
P3.3	/INT1	13
P3.4	T0	14
P3.5	T1	15
P3.6	/WR	16
P3.7	/RD	17

Session 7

Memory address decoding

- In addition to the internal memory of 4 Kbytes of ROM and 256 bytes of RAM, upto 64 Kbytes of ROM & 64 Kbytes of RAM can be interfaced to 8051 using Port 0 (AD0 – AD7), Port 2 (A8 – A15), P3.6 (/WR), P3.7 (/RD), /PSEN, ALE & EA pins.
- Memory chips have one or more pins called CS (Chip Select) or chip enable (CE), which must be activated for the memory's contents to be accessed.





- The data bus of the CPU is connected directly to the data pins of the memory chip.
- Control signals RD (read) & WR (write) from the CPU are connected to the OE (output enable) & WE (write enable) pins of the memory chip respectively.
- The address bus of CPU is connected to address lines of the memory chip and also used to decode the memory chip selection through CS pin of memory chip.

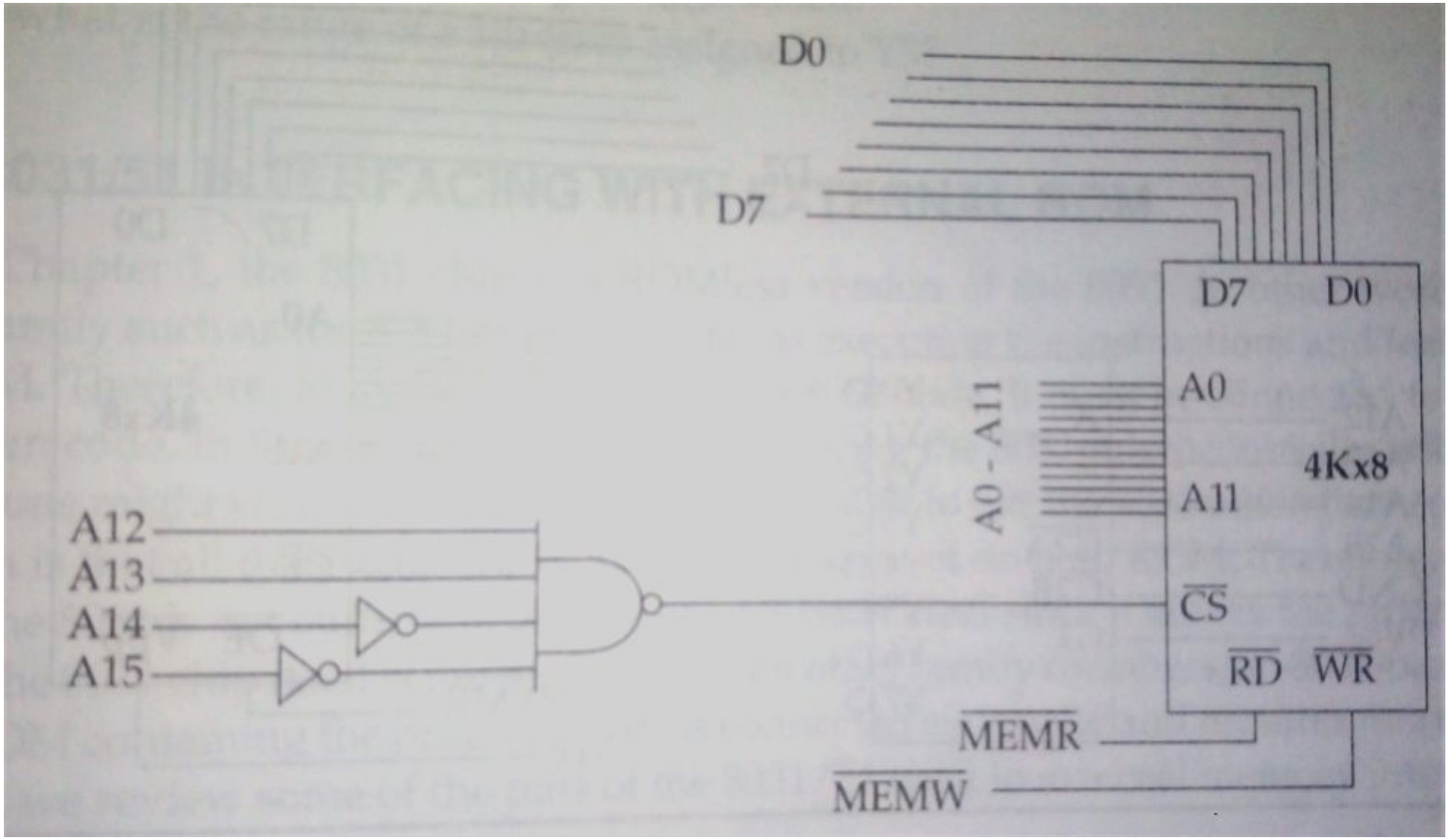
Memory address decoding

There are three ways to generate a memory block selector:

- Using simple logic gates
- Using the 74LS138
- Using programmable logics

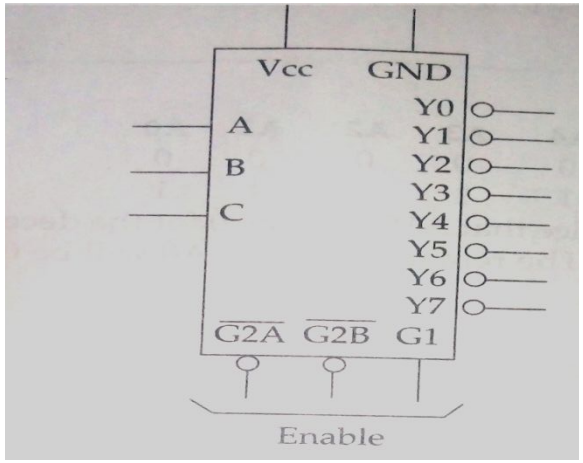
Memory address decoding

Simple logic gate address decoder



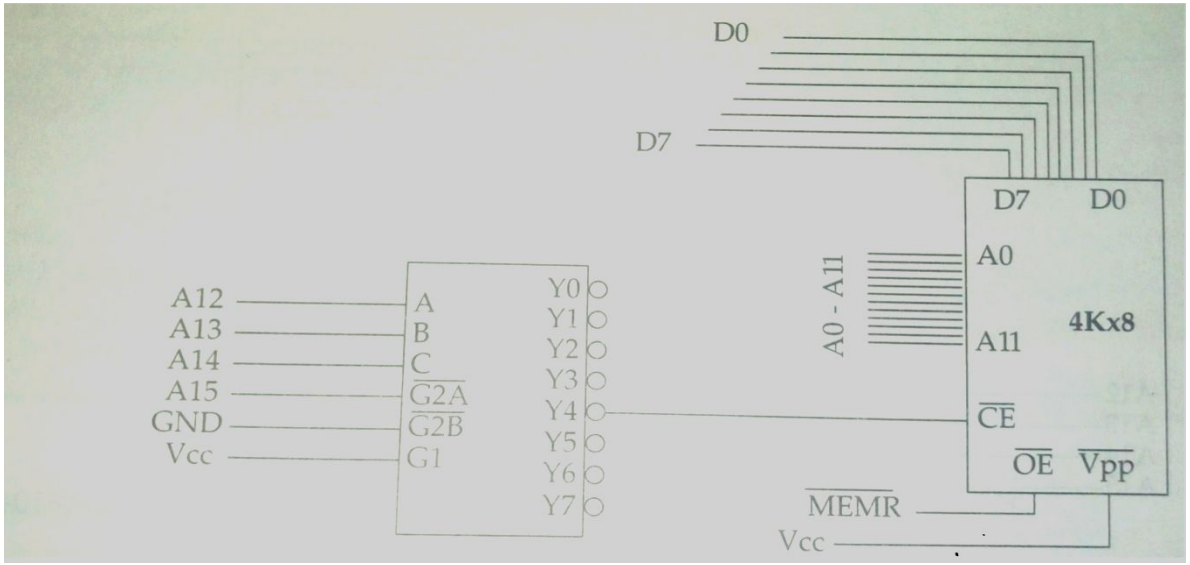
Memory address decoding

Using the 74LS138 3-8 decoder



Function Table

Inputs					Outputs							
Enable		Select										
G1	G2	C	B	A	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
X	H	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	H	H	L	H	H	H	H	H	H
H	L	L	H	L	H	H	L	H	H	H	H	H
H	L	L	H	H	H	H	H	L	H	H	H	H
H	L	H	L	L	H	H	H	H	L	H	H	H
H	L	H	L	H	H	H	H	H	H	L	H	H
H	L	H	H	L	H	H	H	H	H	H	L	H
H	L	H	H	H	H	H	H	H	H	H	H	L



Memory address decoding

Using the 74LS138 3-8 decoder

- This is one of the most widely used address decoders.
- The 3 inputs A, B, & C generate 8 active-low outputs Y0-Y7.
- Each Y output is connected to CS of a memory chip, allowing control of 8 memory blocks by a single 74LS138.
- there are three additional inputs, G2A, G2B, and G1. G2A & G2B are both active low, and G1 is active high. If any one of the inputs G1, G2A, or G2B is not connected to an address signal, they must be activated permanently either by Vcc or ground, depending on the activation level.

Memory address decoding

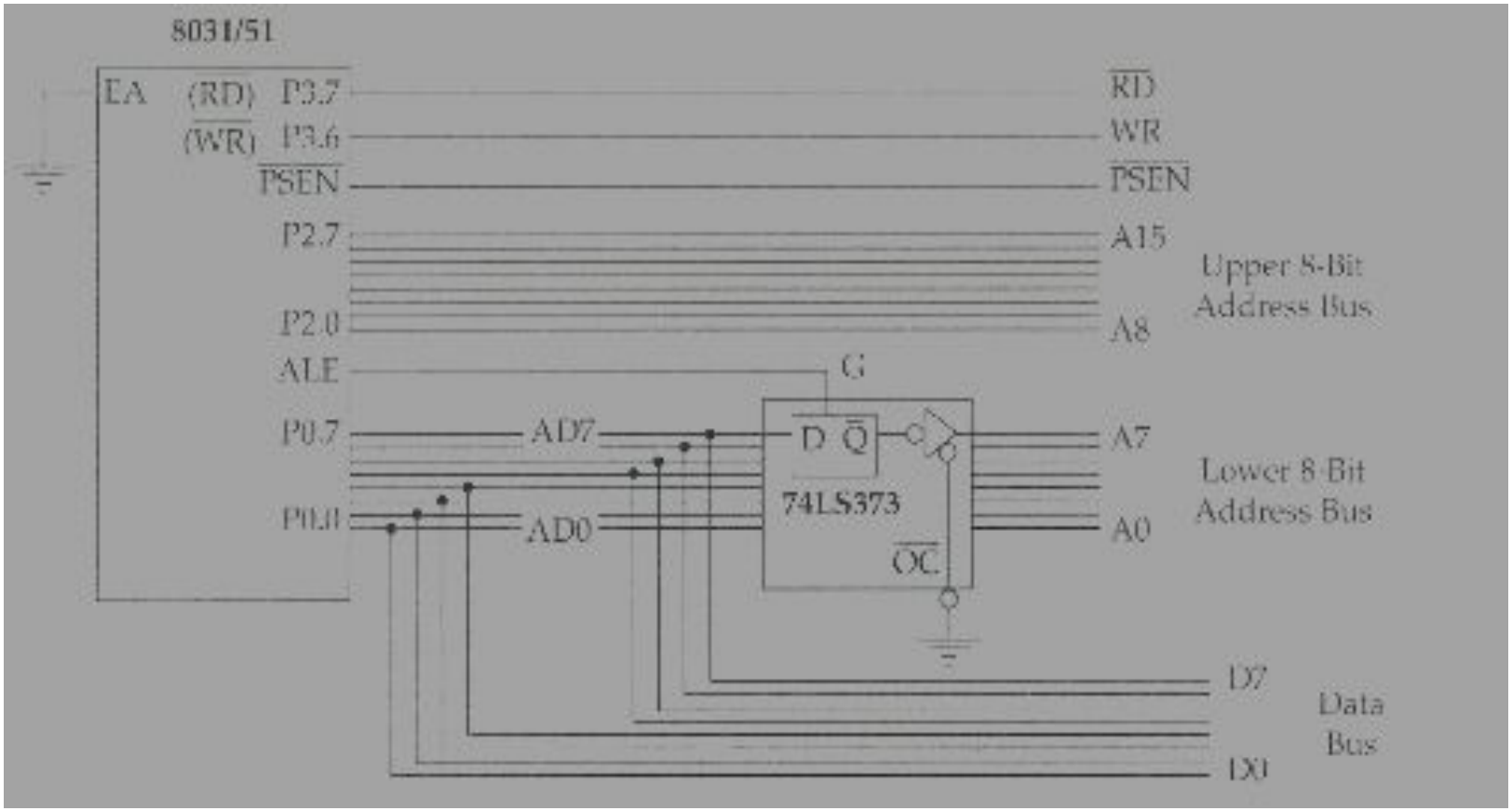
Using programmable logic as an address decoder

- Programmable logic chips such as PAL (Programmable Array Logic) & GAL (Generic Array Logic) chips are widely used.
- One disadvantage of these chips is that they require PAL/GAL software and a burner (programmer), whereas 74LS138 needs neither of these.
- The advantage of these chips is that they can be programmed for any combination of address ranges, and so are much more versatile.
- PALs & GALs have 10 or more inputs in contrast to 6 in 74LS138.

Session 8

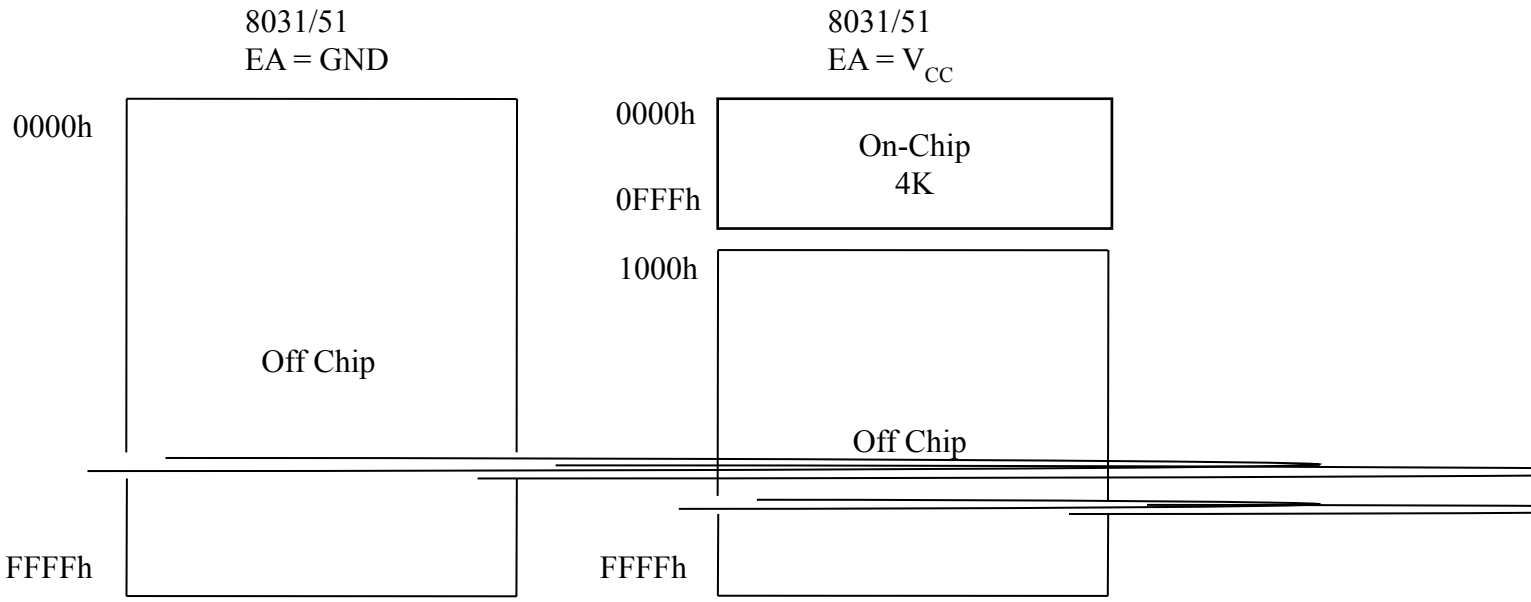
8031/51 Interfacing with External ROM and RAM:

- EA pin
- P0 and P2 role in providing addresses
- PSEN



8031/51 Interfacing with External ROM and RAM:

On- chip and off-chip code ROM



8031/51 Interfacing with External ROM and RAM:

External ROM for data:

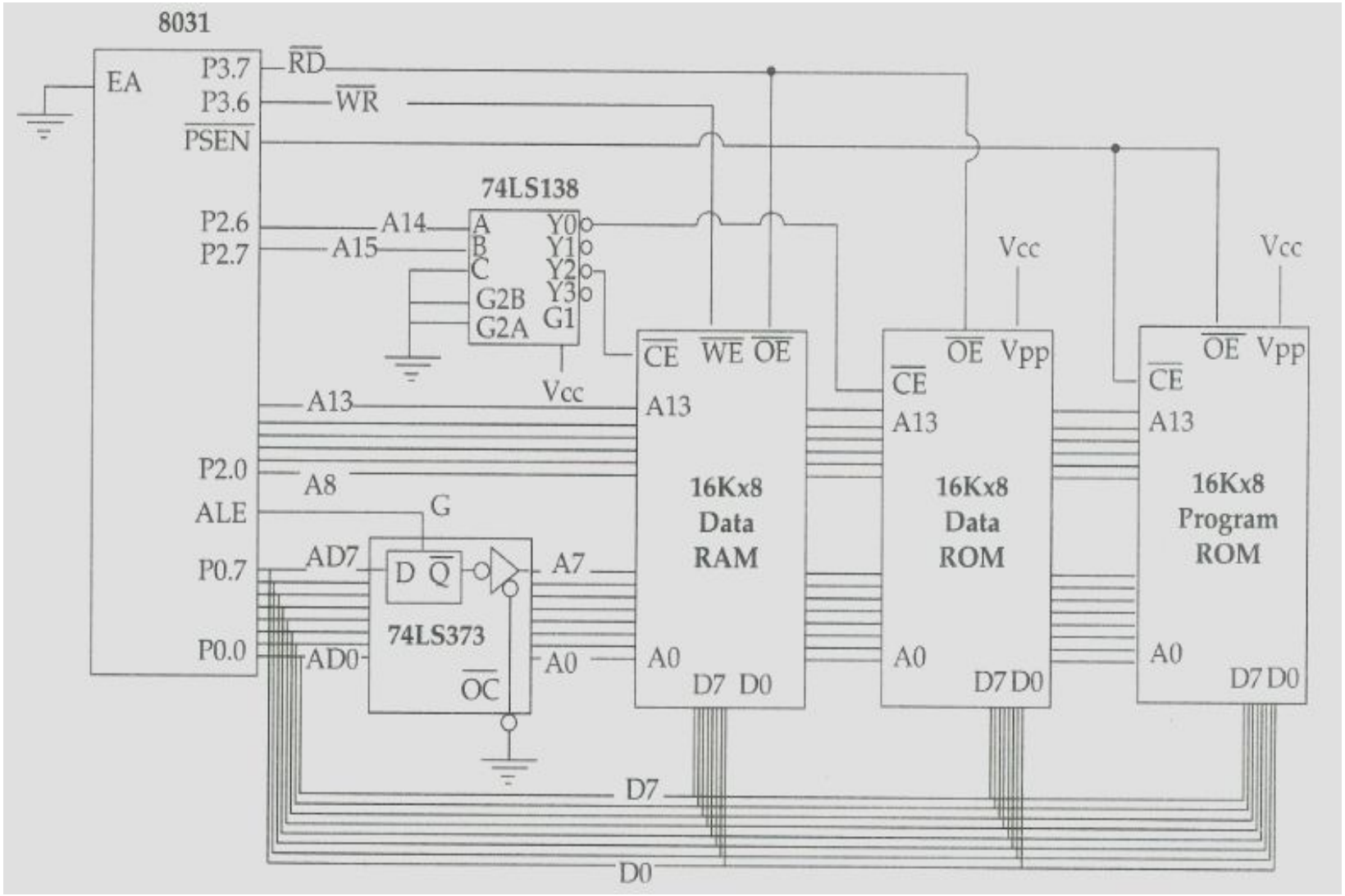
- To connect the 8031/51 to external ROM containing data, we use RD (pin P3.7).
- PSEN and RD signals play an important role in accessing ROM for code & data respectively. For the ROM containing the program code, PSEN is used to fetch the code. For the ROM containing data, the RD signal is used to fetch the data.
- MOVC instruction is used to access the data from ROM along with PC or DPTR.

8051 Data memory space :

- In addition to its code space, the 8051 family also has 64K bytes of data memory space.
- The data memory space is accessed using the DPTR register and an instruction called MOVX, where X stands for external (meaning that the memory space must be implemented externally).

Session 9

8031/51 Interfacing with External ROM and RAM:



8031/51 Interfacing with External ROM and RAM:

Memory	A 15 (B)	A 14 (A)	A 13	A 12	A 11	A 10	A 9	A 8	A 7	A 6	A 5	A 4	A 3	A 2	A 1	A 0	Address space
16K Data ROM	**0**	**0**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000h
	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~
	0	**0**	1	1	1	1	1	1	1	1	1	1	1	1	1	1	3FFFh
16K Data RAM	**1**	**0**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8000h
	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~
	1	**0**	1	1	1	1	1	1	1	1	1	1	1	1	1	1	BFFFh
16K Program ROM	**x**	**x**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000h
	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~
	x	**x**	1	1	1	1	1	1	1	1	1	1	1	1	1	1	3FFFh

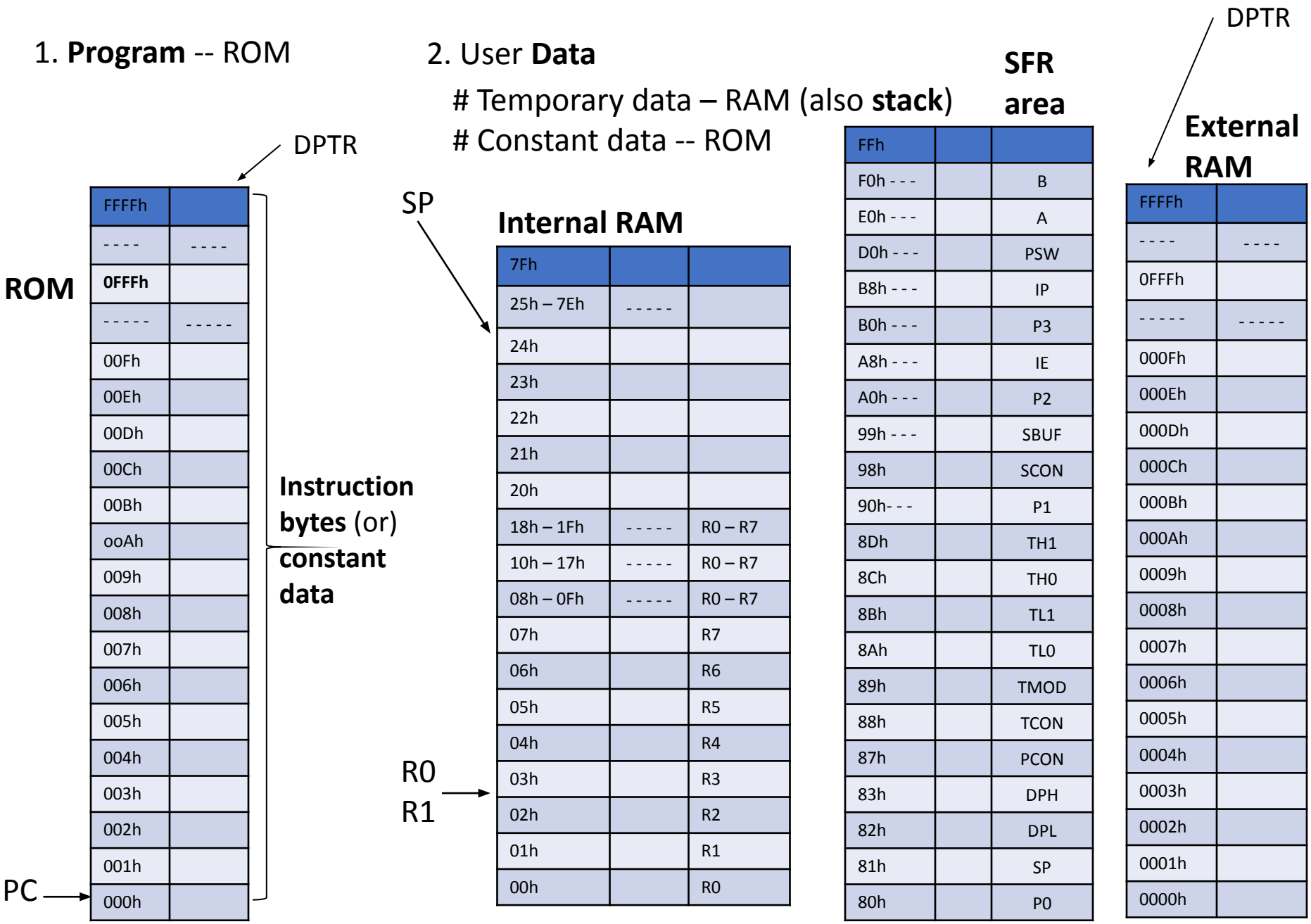
8051 Addressing modes

1. **Program** -- ROM

2. User **Data**

Temporary data – RAM (also **stack**)

Constant data -- ROM



Session 10

8051 Addressing modes

The method of specifying the operands in the instructions is called addressing mode.

Or

The method of supplying the data for the instructions operations is called addressing mode.

1. Data Memory Addressing modes

Sl. No.	Addressing mode	Description	Example
1	Immediate addressing mode	In immediate addressing mode, an 8-bit/16-bit immediate data or constant is specified in the instruction itself. The immediate data is represented by using symbol # along with data. Destination operand can't be immediate	i) MOV R0, #25h ii) ADD A, #0EFh
2	Register addressing mode	The name of the register in which the data is available is specified in the instruction.	i) MOV A, R0 ii) SUB A, R1
3	Direct addressing mode	The address of the data is directly specified in the instruction. The direct address can be the address of an internal data RAM location (00h to 7Fh) or the address of SFR (80h to FFh)	i) MOV R2, 50h ii) AND A, 75h

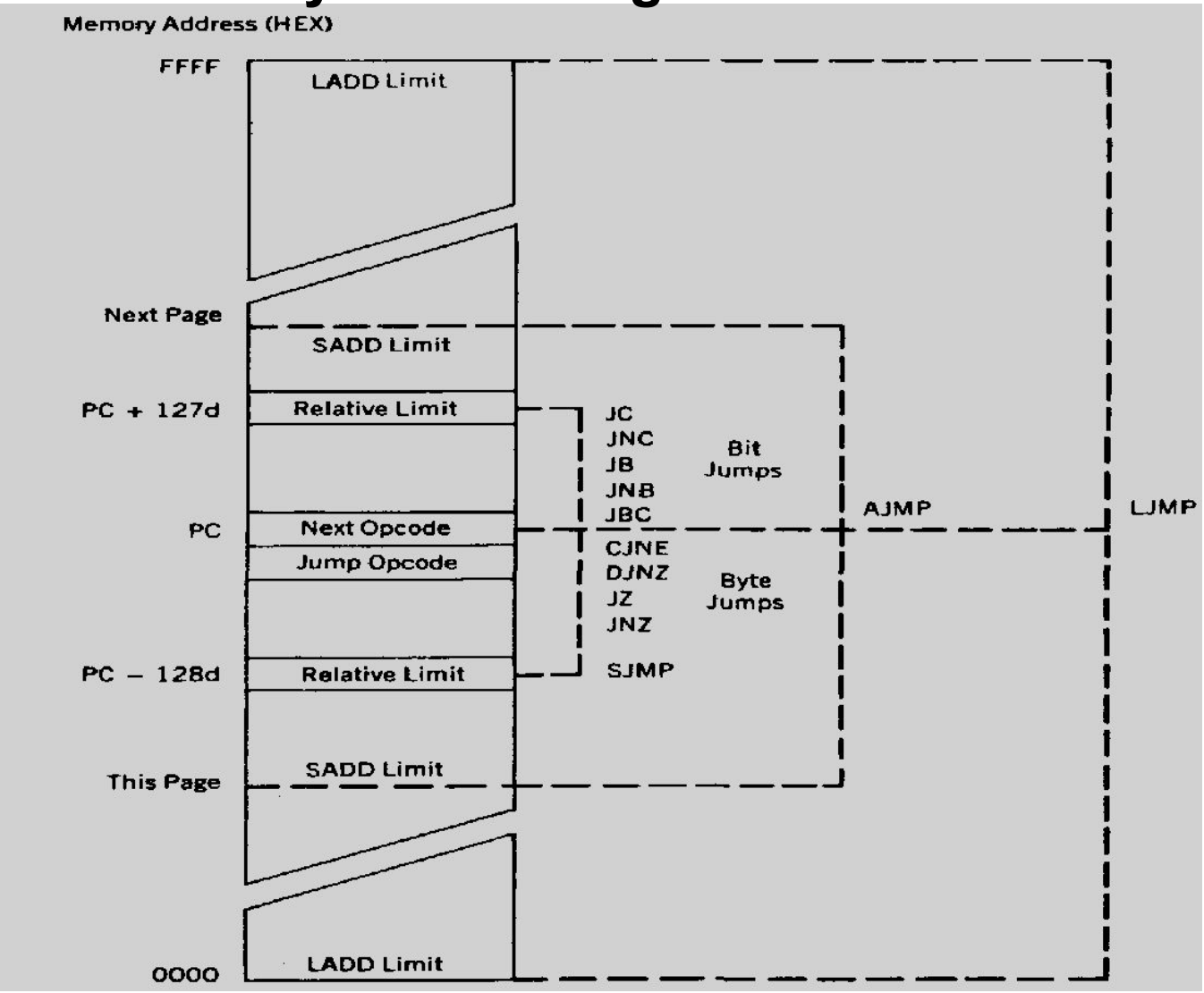
8051 Addressing modes

1. Data Memory Addressing modes

Sl. No.	Addressing mode	Description	Example
4	Register Indirect addressing mode	The address of the data (memory location) is specified through the registers in the instruction. Only registers R1 & R0 are used for this purpose. The external RAM can be addressed indirectly through DPTR. The register which holds the address is called as data pointer & in the instruction it is represented using @ symbol with the register name.	i) MOV A, @R0 ii) OR A, @R1 iii) MOVX A, @DPTR
5	Bit inherent addressing mode	The address of the bit operand is implied in the opcodes of the instruction itself. This mode is used to operate on flags.	i) CLR C
6	Bit direct addressing mode	The address of the bit location is directly mentioned in the instruction. This mode is used to access & operate on bit addressable area of internal RAM (20h to 2Fh) and bit addressable SFRs.	i) SETB 05h ii) MOV C, bit

8051 Addressing modes

2. Program Memory Addressing modes



8051 Addressing modes

2. Program Memory Addressing modes

Sl. No.	Addressing mode	Description	Example
1	Relative addressing mode	In relative addressing mode, the instruction specifies the address relative to the Program Counter (PC). The instruction will carry an offset whose range is -128d to + 127d. The offset is added to the PC to generate the 16-bit physical address.	i) JC offset ii) SJMP L1
2	Indexed addressing mode	Indexed addressing mode is used to access data elements of look up table entries located in the program ROM space of the 8051. In this mode, instruction MOVC is used and as operands the combination of PC & A or DPTR & A are used. PC/DPTR holds the base address & A register holds the offset address. The sum of base address & offset address gives the absolute 16-bit address.	i) MOVC A, @A+DPTR ii) MOVC A, @A+PC

8051 Addressing modes

2. Program Memory Addressing modes

<table>
<tr><th>Sl. No.</th><th>Addressing mode</th><th colspan="3">Description</th><th>Example</th></tr>
<tr><td rowspan="8">3</td><td rowspan="8">Short absolute addressing mode</td><td colspan="3">program memory is divided into a series of pages of 2K bytes each, giving a total of 32d (20H) pages. The upper 5 bits of the Program Counter (PC) hold the page number and the lower 11 bits hold the address within each page.</td><td rowspan="8">AJMP L2</td></tr>
<tr><td>Page no.</td><td>Hex address</td><td>Binary address</td></tr>
<tr><td>01</td><td>0000 – 07FFh</td><td>0000 0000 0000 0000 to
0000 0111 1111 1111</td></tr>
<tr><td>02</td><td>0800 – 0FFFh</td><td>0000 1000 0000 0000 to
0000 1111 1111 1111</td></tr>
<tr><td>05</td><td>2000 – 2FFFh</td><td>0010 0000 0000 0000 to
0010 0111 1111 1111</td></tr>
<tr><td>--</td><td>--</td><td>--</td></tr>
<tr><td>--</td><td>--</td><td>--</td></tr>
<tr><td>32</td><td>F800 - FFFFh</td><td>1111 1000 0000 0000 to
1111 1111 1111 1111</td></tr>
</table>

8051 Addressing modes

2. Program Memory Addressing modes

Sl. No.	Addressing mode	Description	Example
4	Long absolute addressing mode	The entire program space from 0000h to FFFFh can be accessed.	LJMP L1

3. Stack Memory Addressing modes

Sl. No.	Addressing mode	Description	Example
1	**Stack memory addressing mode**	Designated stack memory is accessed through a default SP register & PUSH & POP operations.	PUSH & POP

The World does not care what you intend, how committed you are, how you feel or what you think, and certainly it has no interest in what you want and don't want. ***It is important that you get clear for yourself that the world only moves for you when you act.***

- Werner Erhard