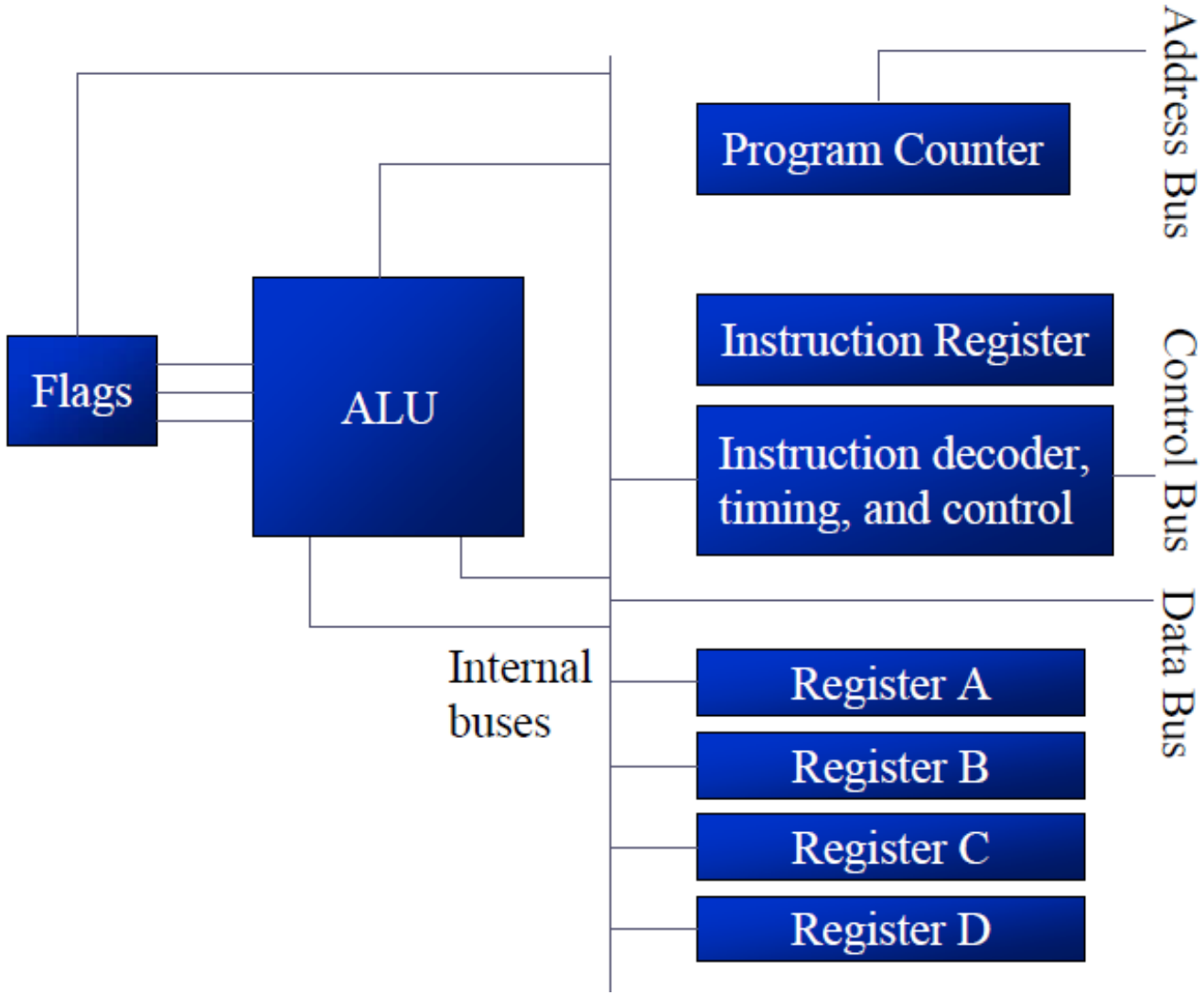
Microcontrollers and Applications – Unit 2

Prepared by,
Dr. Shilpa Hudnurkar
For
Sem V
Batch 2021-25
Department of Electronics & Telecommunication Engineering
Symbiosis Institute of Technology, Pune

Introduction to 8051 Microcontroller

- Introduction to 8051 family of microcontrollers, microcontroller peripherals, comparison of 8051 microcontroller with other controllers. Port structure, memory organization, stack. Introduction to addressing modes and assembly language instruction set.

Inside CPUs


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

Internal working of CPUs

- Suppose the action to be performed is:

Action	Code	Data
Move value 20H into register A	B0H	20H
Add value 40H to register A	04H	40H
Add value 10H to register A	04H	10H

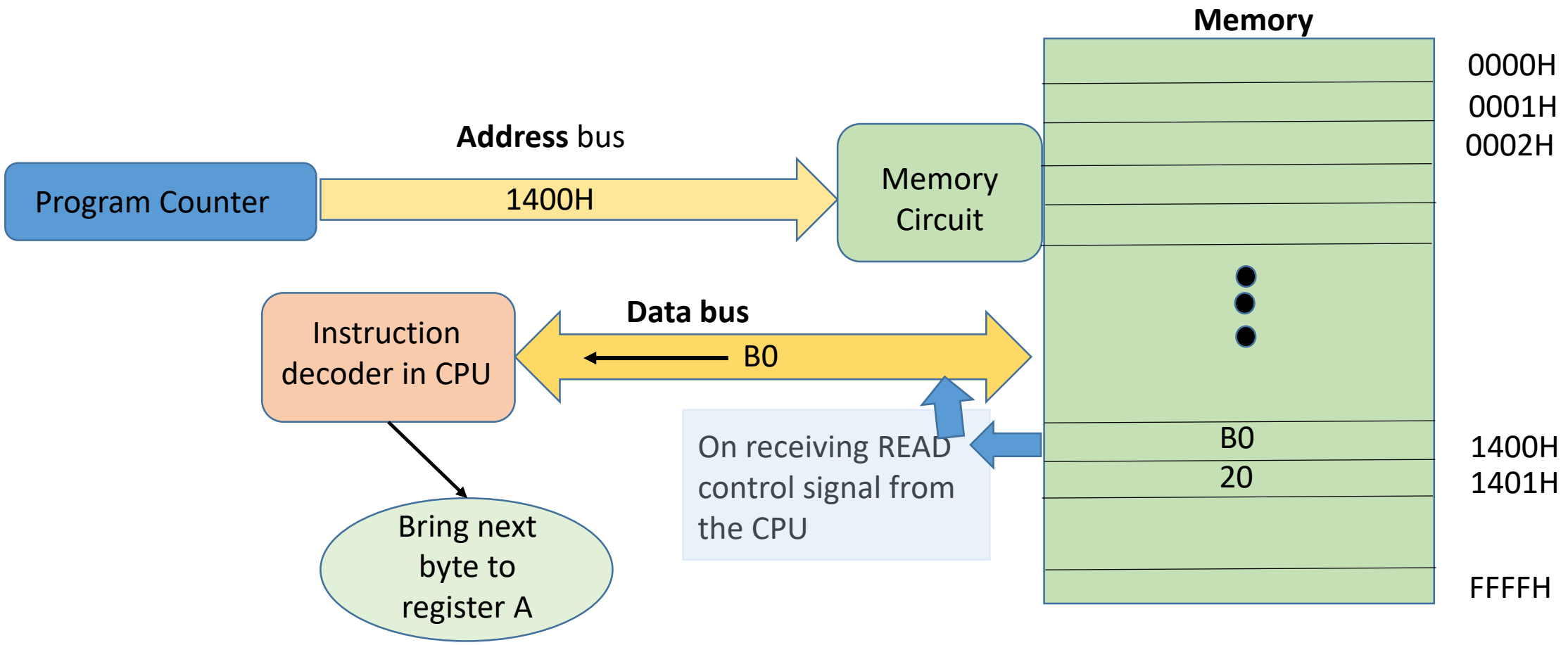
- If the program to perform the actions is stored in memory locations starting at 1400H, the contents for each memory address location would be

Memory address	Cont1406ents of memory address
1400	B0 – Code for moving a value to register A
1401	20 – Value to be moved
1402	04 – Code for adding a value to register A
1403	40 – Value to be added
1404	04– Code for adding a value to register A
1405	10– Value to be added
1406	F4 (halt)

Internal working of CPUs

- Given that address bus is 16 bit and data bus is 8 bit, 2^16 locations can be used to store data, meaning CPU can address 0000H to FFFFH memory locations.
- The actions performed by CPU would be
  - Program counter (PC) must be set to a value 1400H indicating the first instruction code to be executed. Once PC is loaded with the value, CPU is ready to execute.
  - CPU puts 1400H on the address bus and sends it out. The memory circuitry finds the location while the CPU activates the READ signal, indicating to memory that it wants the byte at location 1400H. The B0 (content of 1400H memory location is put on data bus and brought into the CPU.
  - The instruction decoder in the CPU decodes the instruction. It finds the definition for the instruction and understands that it must bring the byte stored at the next memory location to the register A. It commands controller circuitry to do so. It closes all other registers than A so that the data goes to the register A directly. The PC points to the next instruction after completing the first instruction.
  - This sequence of operations is repeated until it encounters halt instruction. The halt instruction tells the CPU to stop incrementing PC and asking about next instruction. In the absence of the Halt instruction, CPU would continue updating the PC and fetching instructions.

Internal working of CPUs



Example

- Find the ROM memory address of each of the following 8051 chips.
  1. AT89C51 with 4KB
  2. DC89C420 with 16KB
  3. DS5000-32 with 32KB

Answers

Example - Answers

1. AT89C51 with 4KB : 4 x 2^10 (1KB = 2^10= 1024)

   = 2 ^ 12 = 4096.

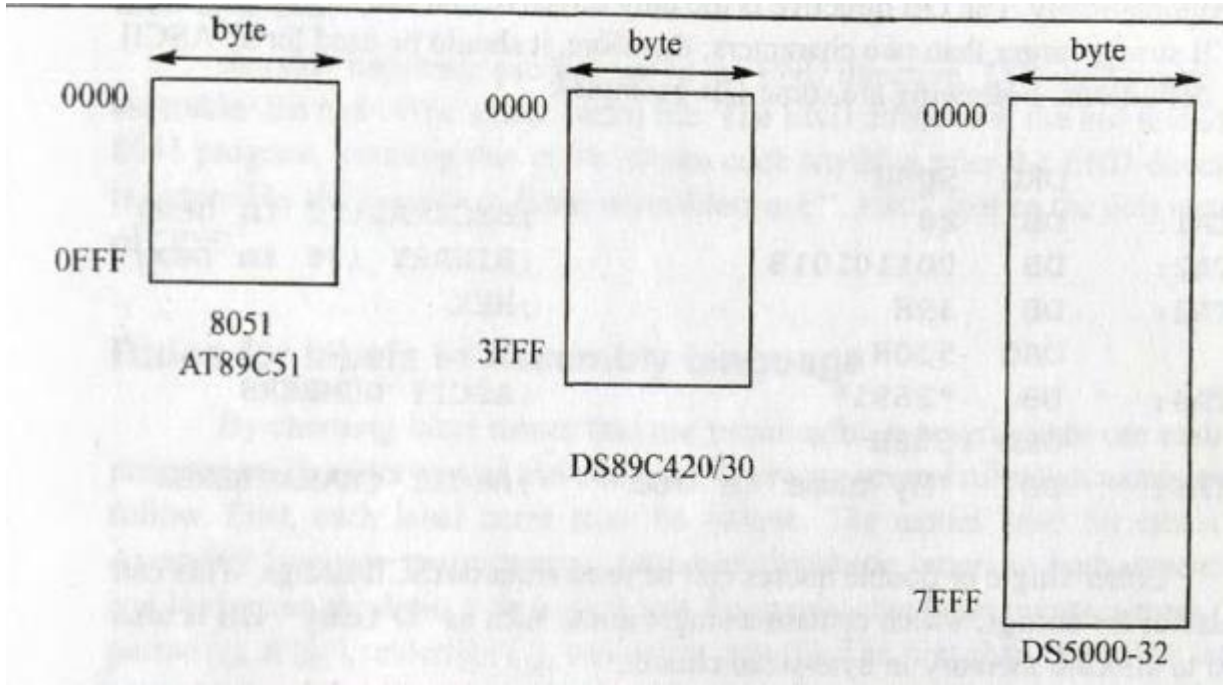
   =0000 H to 0FFFH

2. DC89C420 with 16KB: 16 x 2^10 = 2 ^ 14

   = 0000H to 3FFFH

3. DS5000-32 with 32KB : 32 x 2^10 = 2^15

   = 0000H to 7FFFH



Back

8051 family

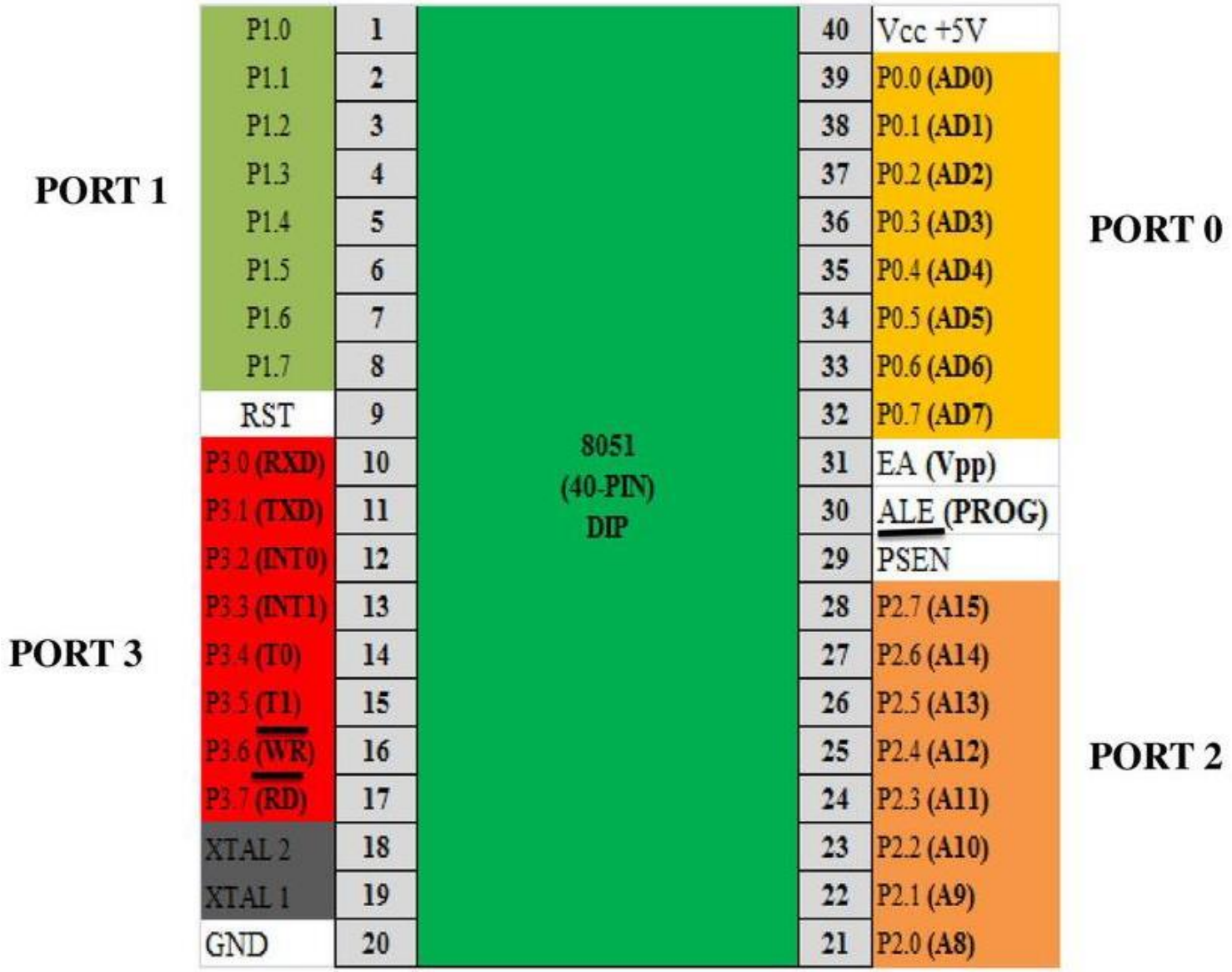
- Various 8051 microcontrollers
  - Available in different memory types: UV-EPROM, NV-RAM, Flash
  - UV-EPROM- 8751, Flash- 8951- by Atmel (AT89C51), Dallas- DC89C4x0

Feature	8051	8052	8031
ROM (on-chip program space in bytes)	4K	8K	0K
RAM (bytes)	128	256	128
Timers	2	3	2
I/O pins	32	32	32
Serial port	1	1	1
Interrupt sources	6	8	6

*** Comparison of 8051 family members**

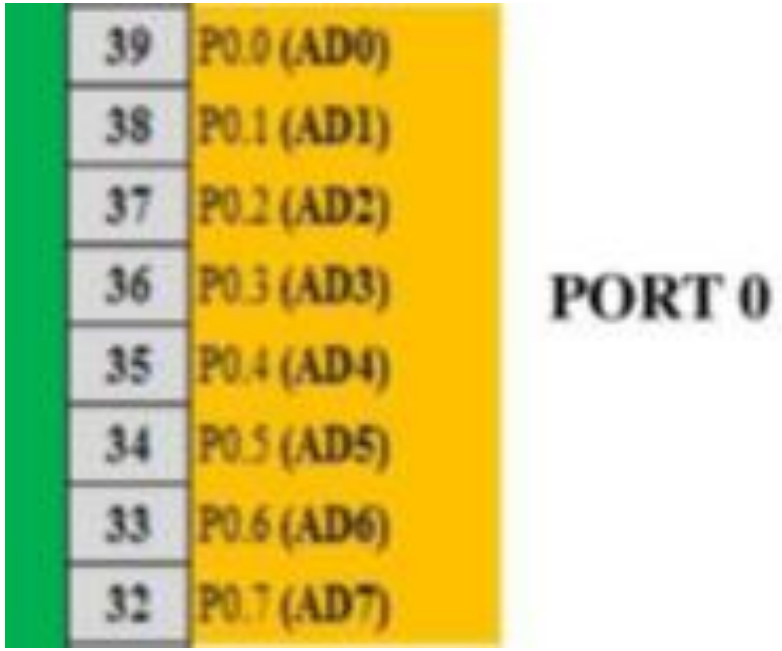
*From Introduction to computing by Chung-Ping Young

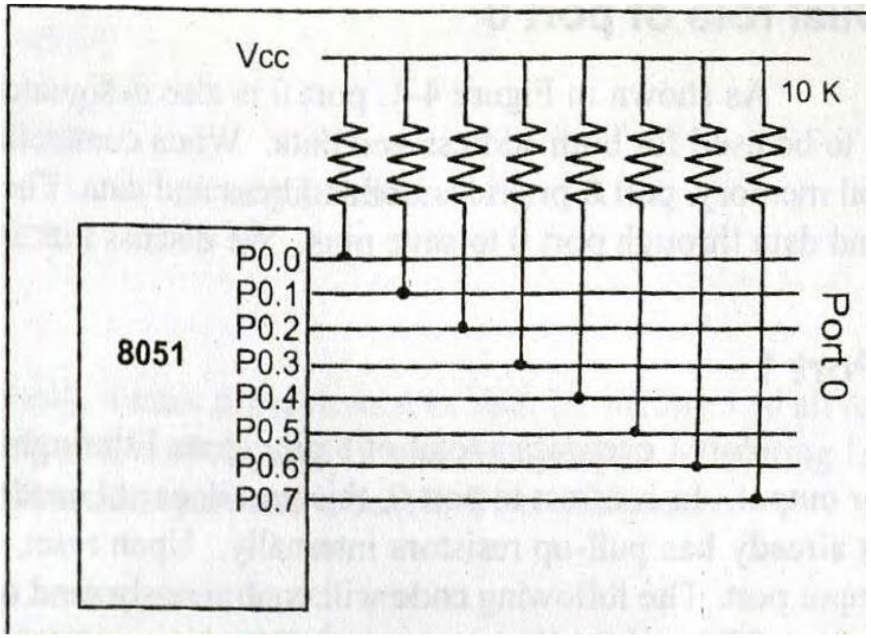
Pin Diagram of 8051



Port 0

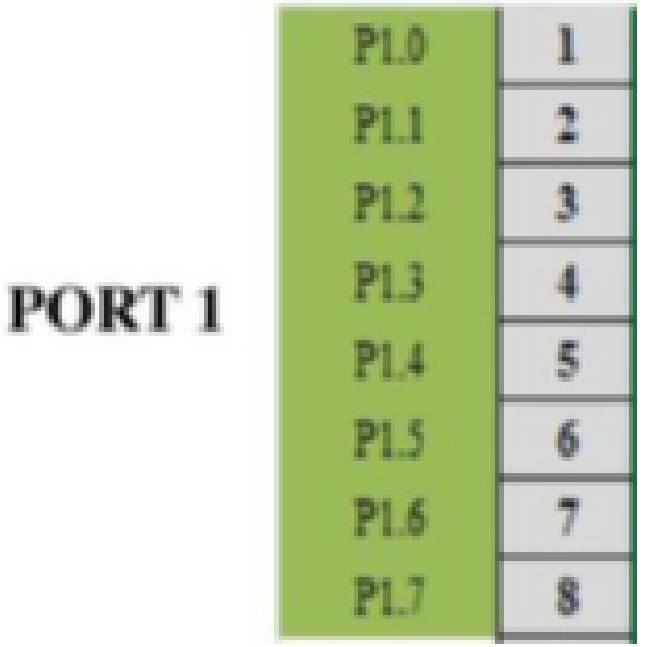
- Occupies 8 pins from pin 32 to 39
- Names: P0.0, P0.1, …, P0.7
- **Dual role** - Can be used for both Address and Data transfer
- These are used for General Purpose Input Output processing
- Do not have built-in pull-up resistors, and hence pull-up resistors are connected externally while using as GPIO
- For using Port 0 as **input**, the port must be programmed by **writing 1** to all the bits





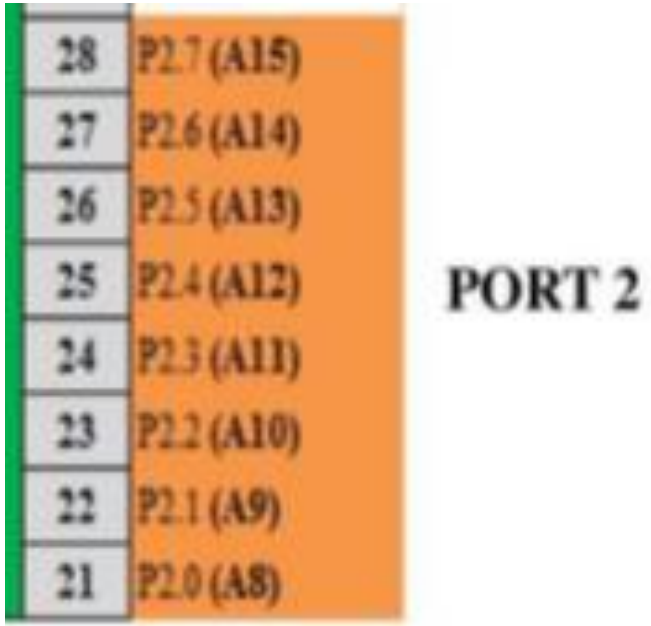
Port 1

- Occupies 8 pins from pin 1 to 8
- Names: P1.0, P1.1, …, P1.7
- These are used for General Purpose Input Output processing
- Does not need pull-up resistors.
- For using Port 1 as **input**, the port must be programmed by **writing 1** to all the bits



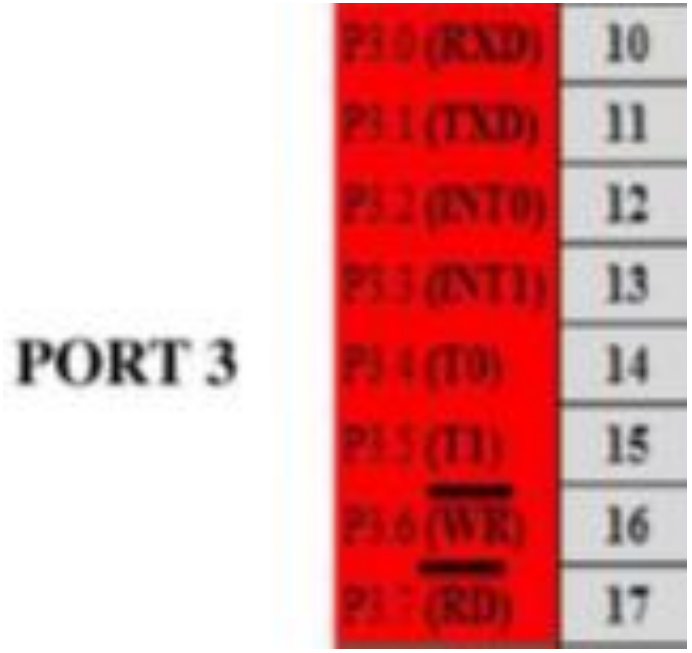
Port 2

- Occupies 8 pins from pin 21 to 28
- Names: P2.0, P2.1, …, P2.7
- Dual role: Provides upper 8 bits A8-A15 of the address while accessing 64KB memory.
- These are used for General Purpose Input Output processing
- Does not need pull-up resistors.
- For using Port 2 as **input**, the port must be programmed by **writing 1** to all the bits



Port 3

- Occupies 8 pins from pin 10 to 17
- Names: P3.0, P3.1, …, P3.7
- These are used for General Purpose Input Output processing
- Does not need pull-up resistors.
- For using Port 3 as **input**, the port must be programmed by **writing 1** to all the bits
- Alternate functions of all pins



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

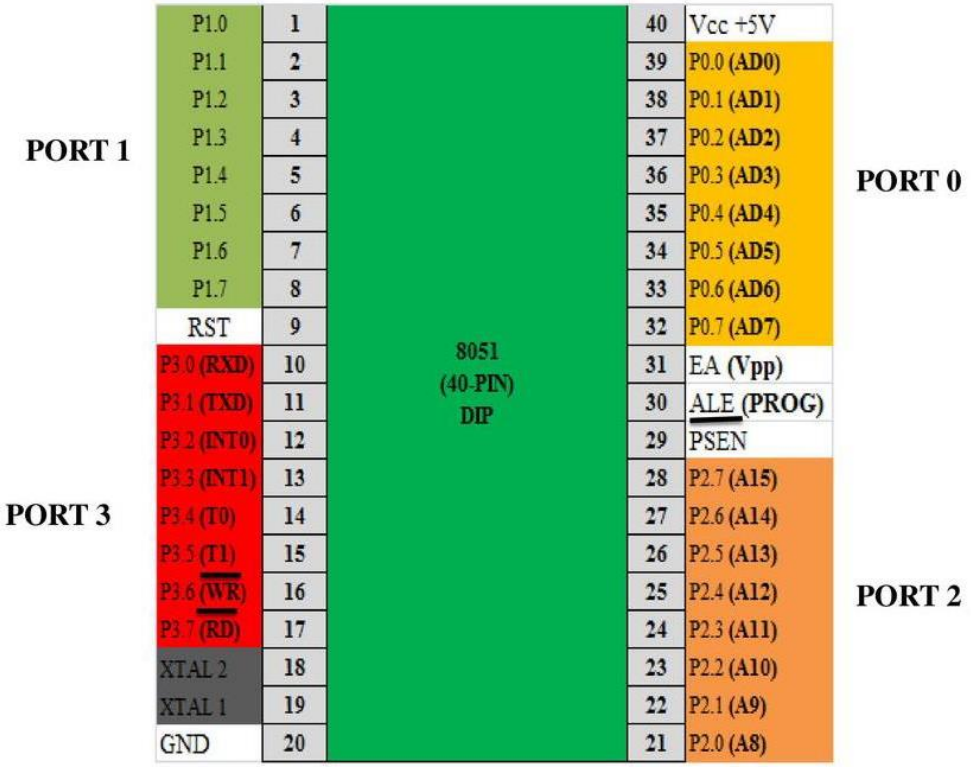
Port 3

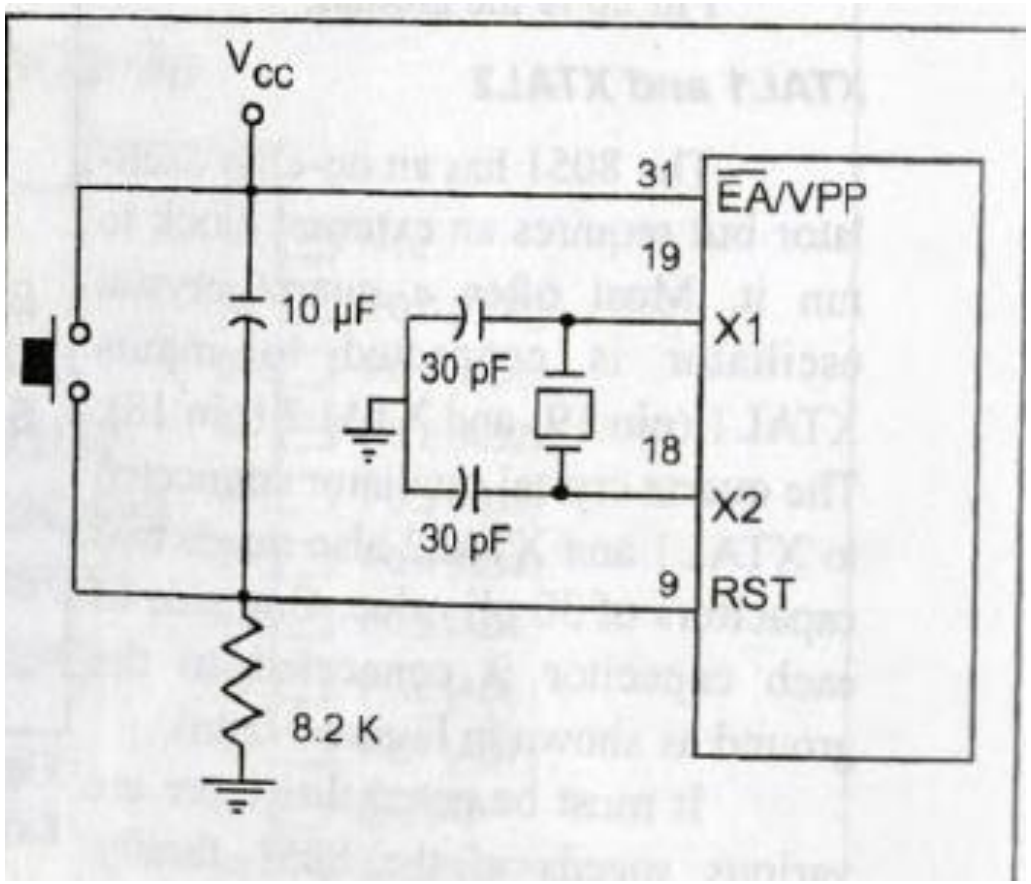
- RXD: Receive Data in serial communication
- TXD: Transmit Data in serial communication
- INT0 & INT1: External interrupt pins
- T0 & T1: Time 0 and Timer 1 input pins
- WR: Write to External Memory
- RD: Read from External Memory

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

Remaining 8 Pins

- **RST : Pin 9**
  - RESET Pin – Active high
  - Set the microcontroller to its primary values
  - Starts the program execution from initial instruction
- **XTAL 1 & XTAL 2: Pin 19 & 18**
  - These are employed for interfacing an outer crystal to give system clock.
- **PSEN: Pin 29**
  - Program Store Enable- Active low
  - Used when program is stored in External ROM
  - Connected to External ROM
- **EA: Pin 30**
  - External Access- Active low
  - Used when program is stored in External ROM
- **ALE: Pin 31**
  - Address Latch Enable
  - ALE = **LOW:** PORT 0 will be used for Address transfer
  - ALE = **HIGH:** PORT 0 will be used for Data transfer

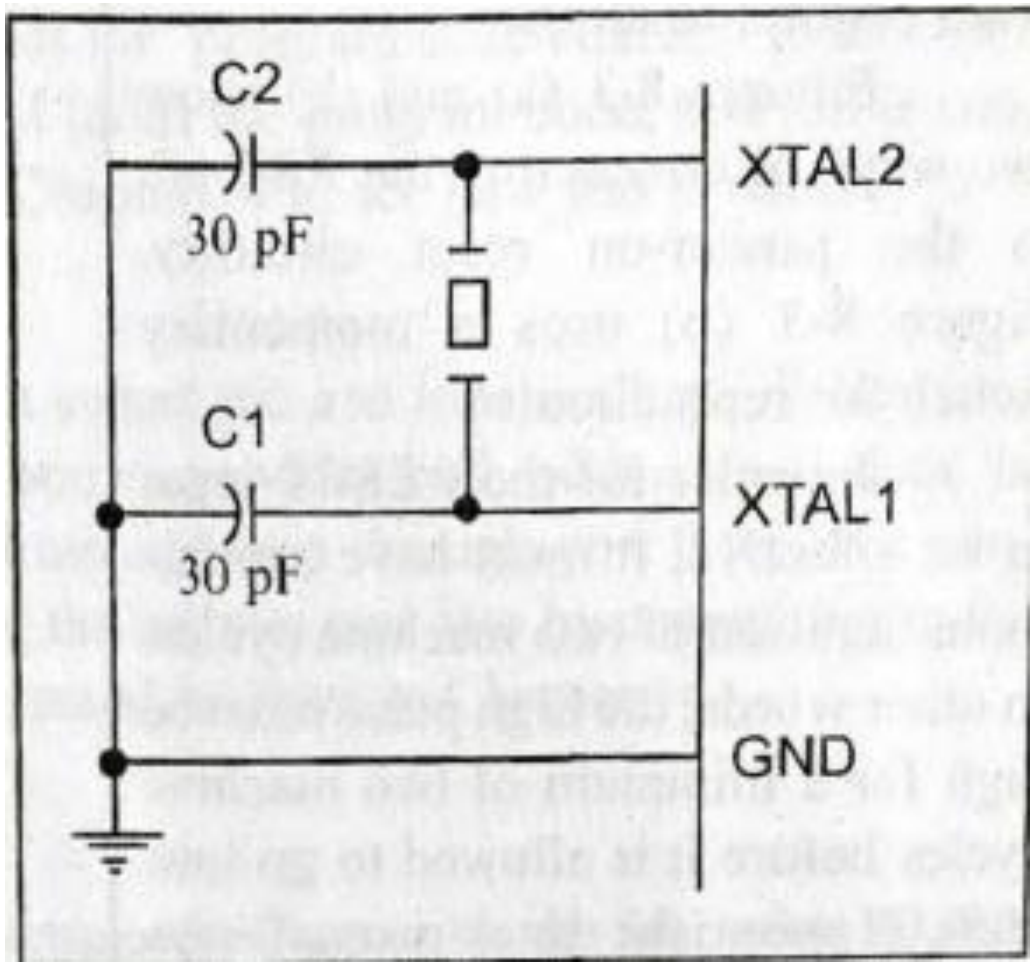




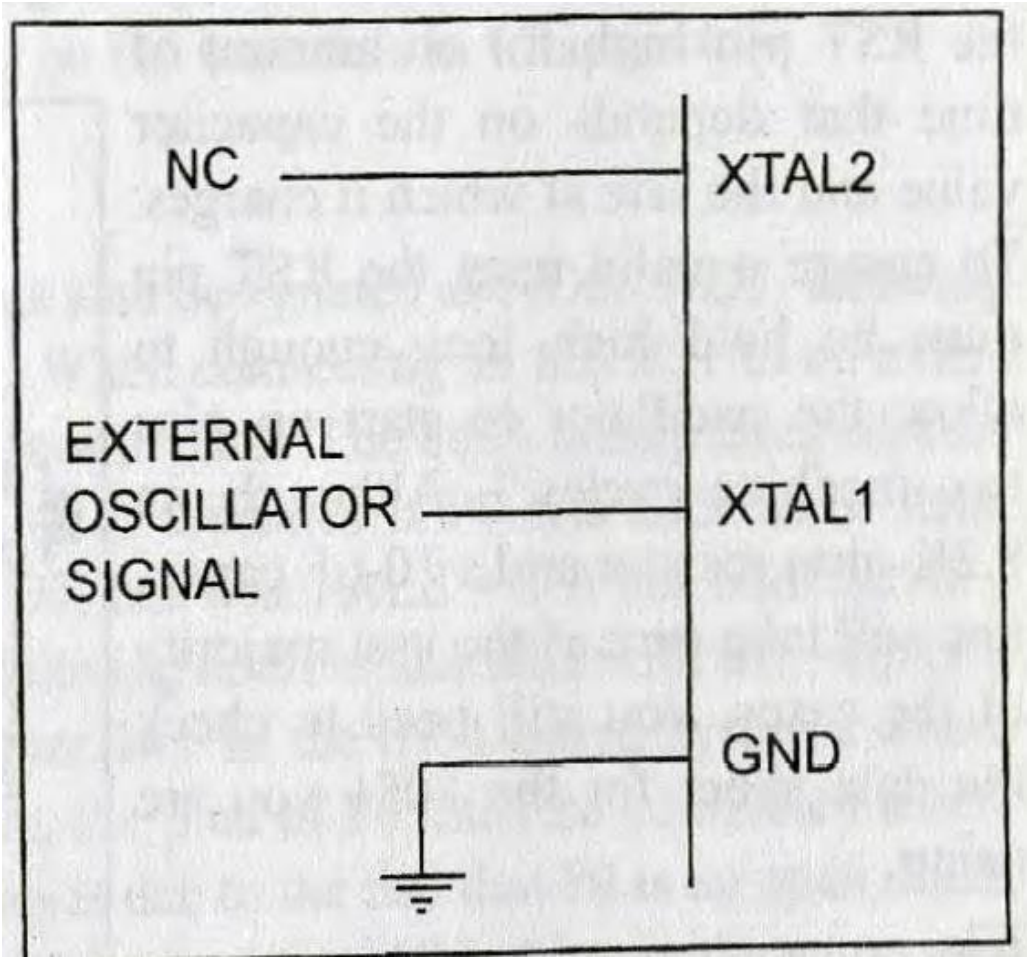
Power on reset with momentary switch

Register	Reset Value (hex)
PC	0000
DPTR	0000
ACC	00
PSW	00
SP	07
B	00
P0-P3	FF

Reset value of some registers



XTAL connection to 8051



XTAL connection to an external clock source

Interrupts

- Interrupt is something that produces some kind of interruption.
- Interrupts are very useful in case of emergency.
- They are used to put the main program on hold when a specific function is required to perform over the main function.
- The system switches to the main program after the interrupt is called and executed properly.
- Six interrupts in 8051
  - INT0 & INT1
  - T0 & T1
  - Serial port
  - RESET

Interrupts

- Interrupts are useful when we need to read or write some data from or to an externally connected device.
- Without interrupts, the normal procedure adopted is polling the device to get the status.
- We can write our program in two ways to poll the device.
- In the first method our program polls the device continuously till the device is ready to send data to the controller or ready to accept data from the controller.
- Here there is a chance that the program hang up and the total system to crash some times when the external device fails or stops functioning.
- Here we are sacrificing the processor time for a single task.

Interrupts

- If the external device supports interrupt, we can connect the interrupt pin of the device to the interrupt line of the controller.
- We can enable the corresponding interrupt in firmware.
- One can write the code to handle the interrupt request service in a separate function and put the other tasks in the main program code.
- Here the main program is executed normally and when the external device asserts an interrupt, the main program is interrupted and the processor switches the program execution to the interrupt request service.
- On finishing the execution of the interrupt request service, the program flow is automatically diverted back to the main stream and the main program resumes its execution exactly from the point where it got interrupted.

Timers and Counters

- Two 16 bit timers & counters
- Timers:
  - used to count the internal signals
  - used as delay generator
- Counters:
  - used to count the external signals
  - used as external event counter

Serial Port

- Standard 8051 supports a full duplex communication
- Follows a Universal Asynchronous Receiver Transmitter (UART) protocol
- The serial communication module contains a
  - Transmit control unit
  - Receive control unit

Power Saving Modes

- For applications like battery operated systems where power consumption is critical, the CMOS version of 8051 provides power reduced modes of operation namely IDLE and POWER DOWN modes.
- Idle Mode / sleep mode
  - the clock provided to CPU gets deactivated
  - Peripherals clock will remain active
  - Oscillator provides clock to peripherals
  - Register and on chip RAM contents remain unchanged
- Power Down Mode
  - the clock provided to CPU gets deactivated
  - Peripherals clock remains inactive
  - Absolute minimum power consumption
  - Contents of RAM and SFR are saved and remain unchanged