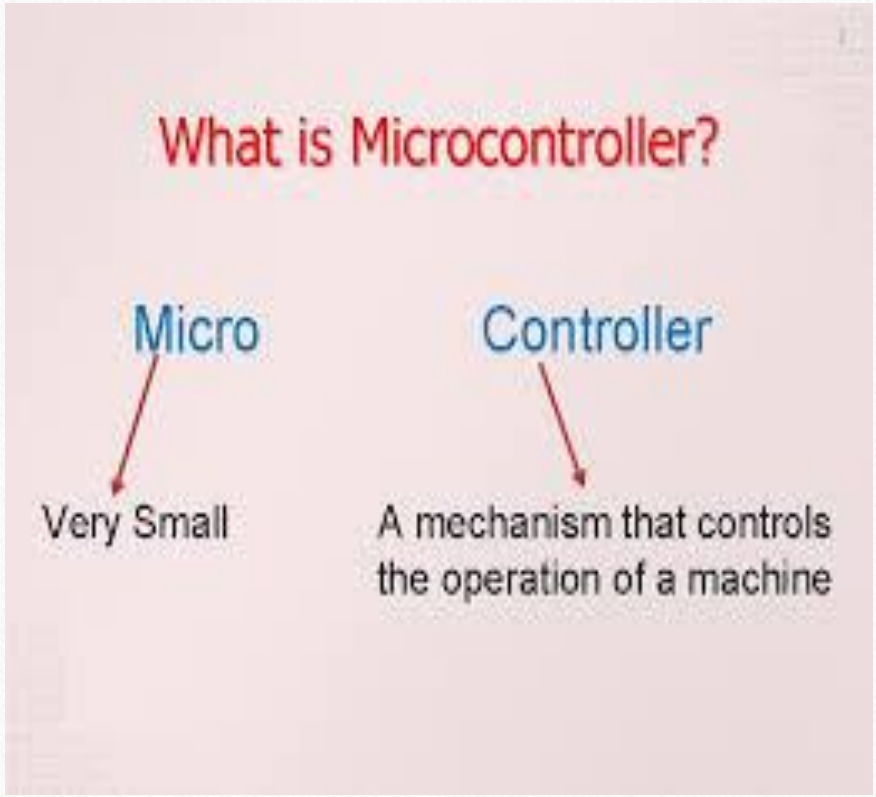
Unit1
Introduction to Microcontroller

Microcontroller basic concept

- A microcontroller is an integrated circuit (IC) device used for controlling other portions of an electronic system, usually via a microprocessor unit (MPU), memory, and some peripherals.
- The most common way to refer to this category of integrated circuits is “microcontroller" but the abbreviation “MCU” is used interchangeably as it stands for “microcontroller unit”.
- You may also occasionally see “µC” (where the Greek letter mu replaces “micro”).



- A microcontroller is a compact integrated circuit designed to govern a specific operation in an embedded system.
- A typical microcontroller includes a processor, memory and input/output (I/O) peripherals on a single chip.
- Microcontrollers are used in automatically controlled products and devices, such as automobile engine control systems, implantable medical devices, remote controls, office machines, appliances, power tools, toys and other embedded systems.

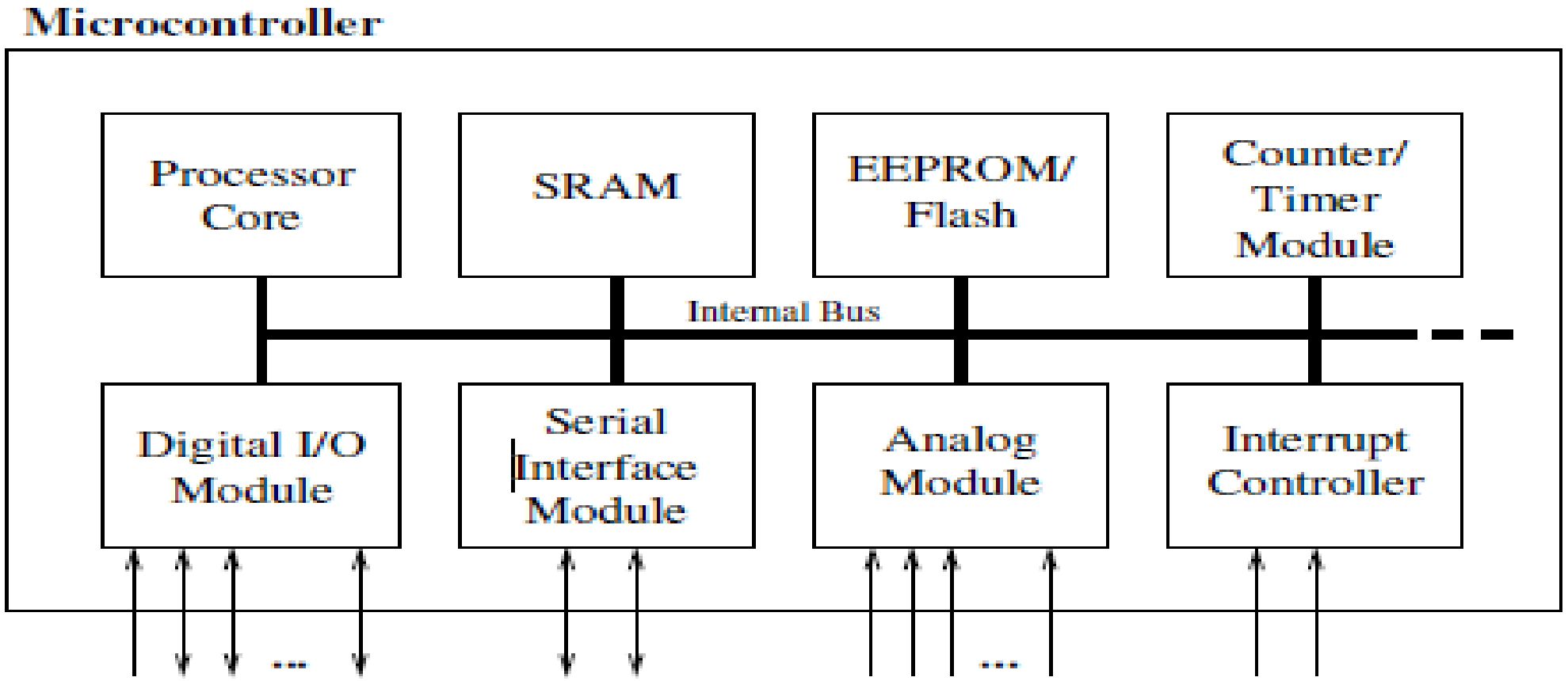


Figure 1.4: Basic layout of a microcontroller.

Processor Core: The CPU of the controller. It contains the arithmetic logic unit, the control unit, and the registers (stack pointer, program counter, accumulator register, register file, . . .).

Memory: The memory is sometimes split into program memory and data memory. In larger controllers, a DMA controller handles data transfers between peripheral components and the memory.

Interrupt Controller: Interrupts are useful for interrupting the normal program flow.

Timer/Counter: Most controllers have at least one and more likely 2-3 Timer/Counters, which can be used to timestamp events, measure intervals, or count events.

Many controllers also contain PWM (pulse width modulation) outputs, which can be used to drive motors or for safe breaking (antilock brake system, ABS). Furthermore the PWM output can, in conjunction with an external filter, be used to realize a cheap digital/analog converter.

Digital I/O: Parallel digital I/O ports are one of the main features of microcontrollers. The number of I/O pins varies from 3-4 to over 90, depending on the controller family and the controller type.

Analog I/O: Apart from a few small controllers, most microcontrollers have integrated analog/digital converters, which differ in the number of channels (2-16) and their resolution (8-12 bits). The analog module also generally features an analog comparator. In some cases, the microcontroller includes digital/analog converters.

Interfaces: Controllers generally have at least one serial interface which can be used to download the program and for communication with the development PC in general. Since serial interfaces can also be used to communicate with external peripheral devices, most controllers offer several and varied interfaces like SCI(**Serial Communications Interface**) and SPI(Serial Peripheral Interface).

Many microcontrollers also contain integrated bus controllers for the most common (field)busses. IIC(alternatively known as ***I2C*** *or* ***IIC****, is a synchronous, multi-controller/ multi-target (controller/ target), packet switched, single-ended, serial communication bus invented in 1982 by Philips Semiconductors.) and CAN(***Controller Area Network****) controllers lead the field here. Larger microcontrollers may also contain PCI(***Peripheral Component Interconnect****), USB, or Ethernet interfaces.*

Watchdog Timer: Since safety-critical systems form a major application area of microcontrollers, it is important to guard against errors in the program and/or the hardware. The watchdog timer is used to reset the controller in case of software "crashes".

Debugging Unit: Some controllers are equipped with additional hardware to allow remote debugging of the chip from the PC. So there is no need to download special debugging software, which has the distinct advantage that erroneous application code cannot overwrite the debugger.

Advantages of Microcontrollers

- A Microcontroller is a true device that fits the computer-on-a-chip idea.
- No need for any external interfacing of basic components like Memory, I/O Ports, etc.
- Microcontrollers doesn't require complex operating systems as all the instructions must be written and stored in the memory. (RTOS *{Real Time Operating System}* is an exception).
- All the Input/Output Ports are programmable.
- Integration of all the essential components reduces the cost, design time and area of the product (or application).

Disadvantages of Microcontrollers

- Microcontrollers are not known for their computation power.
- The amount of memory limits the instructions that a microcontroller can execute.
- No Operating System and hence, all the instruction must be written.

Applications of Microcontrollers

There are huge number of applications of Microcontrollers. In fact, the entire embedded systems industry is dependent on Microcontrollers. The following are few applications of Microcontrollers.

- Front Panel Controls in devices like Oven, washing Machine etc.
- Smoke and Fire Alarms
- Home Automation Systems
- Automatic Headlamp ON in Cars
- Speed Sensed
- Door Locking System and many more…….

Microprocessor	Microcontroller
Microprocessor assimilates the function of a central processing unit (CPU) on to a single integrated circuit (IC).	A microcontroller can be considered as a small computer that has a processor and some other components in order to make it a computer.
Microprocessors are mainly used in designing general-purpose systems from small to large and complex systems like supercomputers.	Microcontrollers are used in automatically controlled devices.
Microprocessors are basic components of personal computers.	Microcontrollers are generally used in embedded systems
A microprocessor-based system can perform numerous tasks.	A microcontroller based system can perform single or very few tasks.

The main task of the microprocessor is to perform the instruction cycle repeatedly. This includes fetch, decode and execute.	In addition to performing the tasks of fetch, decode and execute, a microcontroller also controls its environment based on the output of the instruction cycle.
System cost is high.	System cost is low.
Applications: 1. The microprocessor is used in personal computers (PCs). 2. The microprocessor is used in LASER printers for good speed and making automatic photocopies.	It is used for automatically controlled products and devices, such as automobile engine control systems, implantable medical devices, remote controls, appliances, power tools, office machines, toys, and other embedded systems.

Embedded System

- An embedded system is a microprocessor-based computer hardware system with software that is designed to perform a dedicated function, either as an independent system or as a part of a large system.
- At the core is an integrated circuit designed to carry out computation for real-time operations.
- Complexities range from a single microcontroller to a suite of processors with connected peripherals and networks; from no user interface to complex graphical user interfaces.

- The complexity of an embedded system varies significantly depending on the task for which it is designed.
- Embedded system applications range from digital watches and microwaves to hybrid vehicles and avionics.
- As much as 98 percent of all microprocessors manufactured are used in embedded systems.

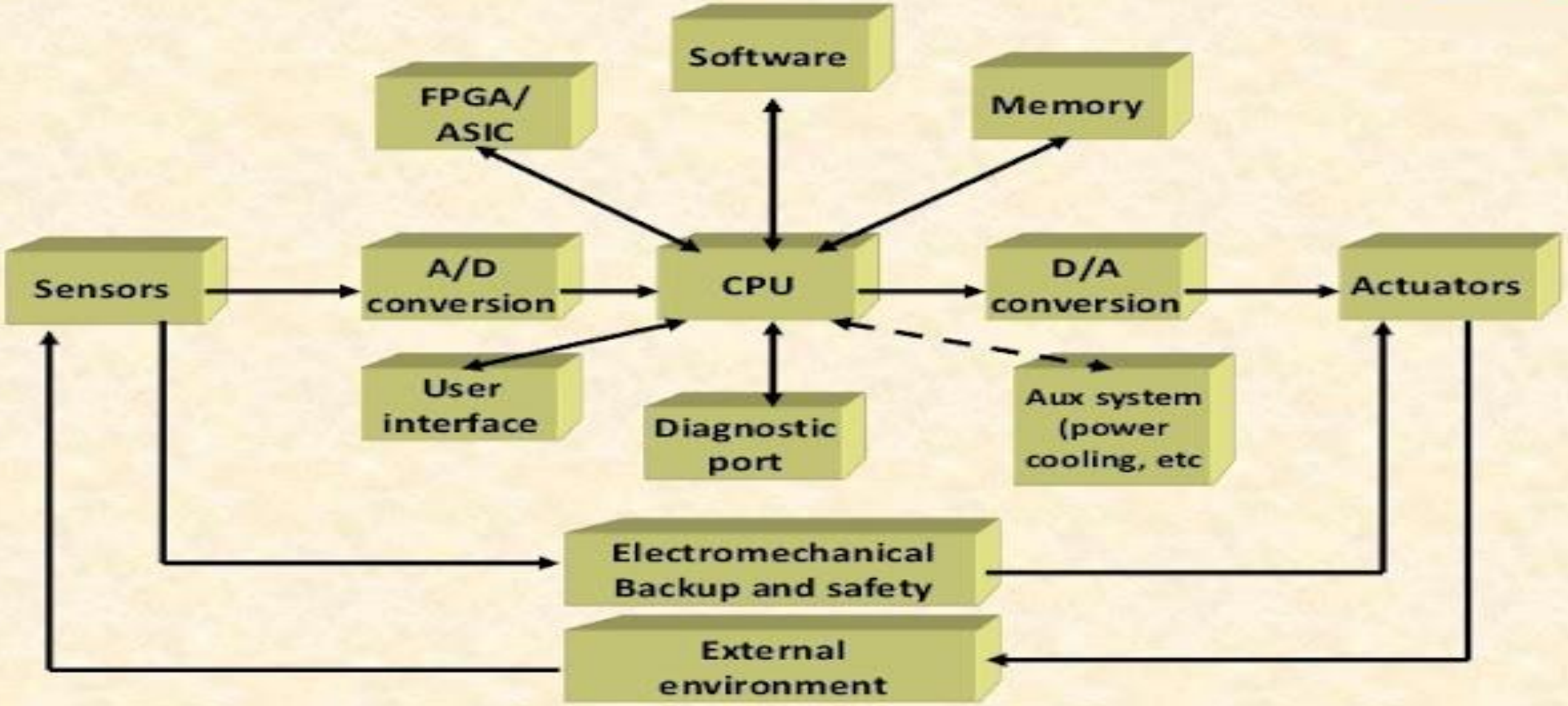
How an Embedded System Works

- Embedded systems are managed by microcontrollers or digital signal processors (DSP), application-specific integrated circuits (ASIC), field-programmable gate arrays (FPGA), GPU technology, and gate arrays.
- These processing systems are integrated with components dedicated to handling electric and/or mechanical interfacing.
- Embedded systems programming instructions, referred to as firmware, are stored in read-only memory or flash memory chips, running with limited computer hardware resources.
- Embedded systems connect with the outside world through peripherals, linking input and output devices.

The basic structure of an embedded system includes the following components:

- **Sensor:** The sensor measures and converts the physical quantity to an electrical signal, which can then be read by an embedded systems engineer or any electronic instrument. A sensor stores the measured quantity to the memory.
- **A-D Converter:** An analog-to-digital converter converts the analog signal sent by the sensor into a digital signal.
- **Processor & ASICs(application-specific integrated circuit)**: Processors assess the data to measure the output and store it to the memory.
- **D-A Converter:** A digital-to-analog converter changes the digital data fed by the processor to analog data
- **Actuator:** An actuator compares the output given by the D-A Converter to the actual output stored and stores the approved output.

Embedded system



Common embedded systems can be broken into four types based on performance as well as functional requirements:

- Real-Time
- Stand-alone
- Networked
- Mobile

Real-Time

Real-time embedded systems are designed and installed to carry out specific tasks within a pre-defined time limit. They are further divided into two different types:

- Soft Real-Time Embedded Systems: For these systems, the completion of the task is of paramount importance, while the deadline is not a priority.
- Hard Real-Time Embedded Systems: These systems prioritize deadlines, so they shouldn't be missed in any case.

Some of the real-time embedded systems examples are:

- Sound System of a computer (Soft real-time system)
- Aircraft control system (Hard real-time system)

Stand-alone

These are self-sufficient systems that do not rely on a host system like a processor or a computer to perform tasks. Here are some standalone embedded technology examples:

- Microwave ovens
- Washing machines
- Video game consoles

Networked

These systems are connected to a wired or wireless network to perform assigned tasks and provide output to the connected devices. They are comprised of components like controllers and sensors. Here are some network embedded software examples:

- ATMs
- Home security systems
- Card swipe machines

Mobile

These systems are smaller in size and easy to use. Though they come with limited memory, people still prefer them due to their portability and handiness. Here are a few mobile embedded control systems examples:

- Digital cameras
- Mobile phones
- Smart watch
- Fitness tracker

Characteristics of Embedded Computer Systems

The main characteristics of typical embedded systems include

1. **Small Form Factor (SFF):** These are PCB(*printed circuit board*) designs packed with robust processing power in smaller rugged enclosures, which maximizes space efficiency.
2. **Power Efficient Components:** These are processors with lower thermal design power that minimize cooling and eradicate the need for fans as well as moving components.
3. **Single-Functioned:** These systems are designed to perform a specific operation during their lifetime.
4. **Lower Cost:** Since they don't feature expansion slots for peripherals, embedded systems are generally lower cost than full-featured computers and have fewer component complexities.

Here are some of the real-life examples of embedded system applications.

- **GPS systems**

Nowadays, GPS systems are generally used in: Cars, Mobile devices,a Palmtop

- **Fitness trackers**

Fitness trackers are generally used for: Monitoring personal activity

- **Medical monitoring:**

Medical devices have been widely used for diagnosing and treating patients efficiently, and some of their examples are: Pacemaker, Ultrasound scanners, Sports training, Medical devices.

•**Automotive systems:**

Some more examples of automotive embedded systems include: Car navigation system, Anti-lock braking system, Vehicle entertainment system

•**Transit and fare collection:**

Automated fare collection systems are generally found at: Metro stations, Bus stations, Railway stations

•**ATMs**:

An ATM is mostly used to: Withdraw cash, Check account balance and transactions details, Deposit money into another account

•**Factory robots:**

Factory robots have a range of applications: Assembly line, Quality monitoring, Welding, Painting , Palletizing

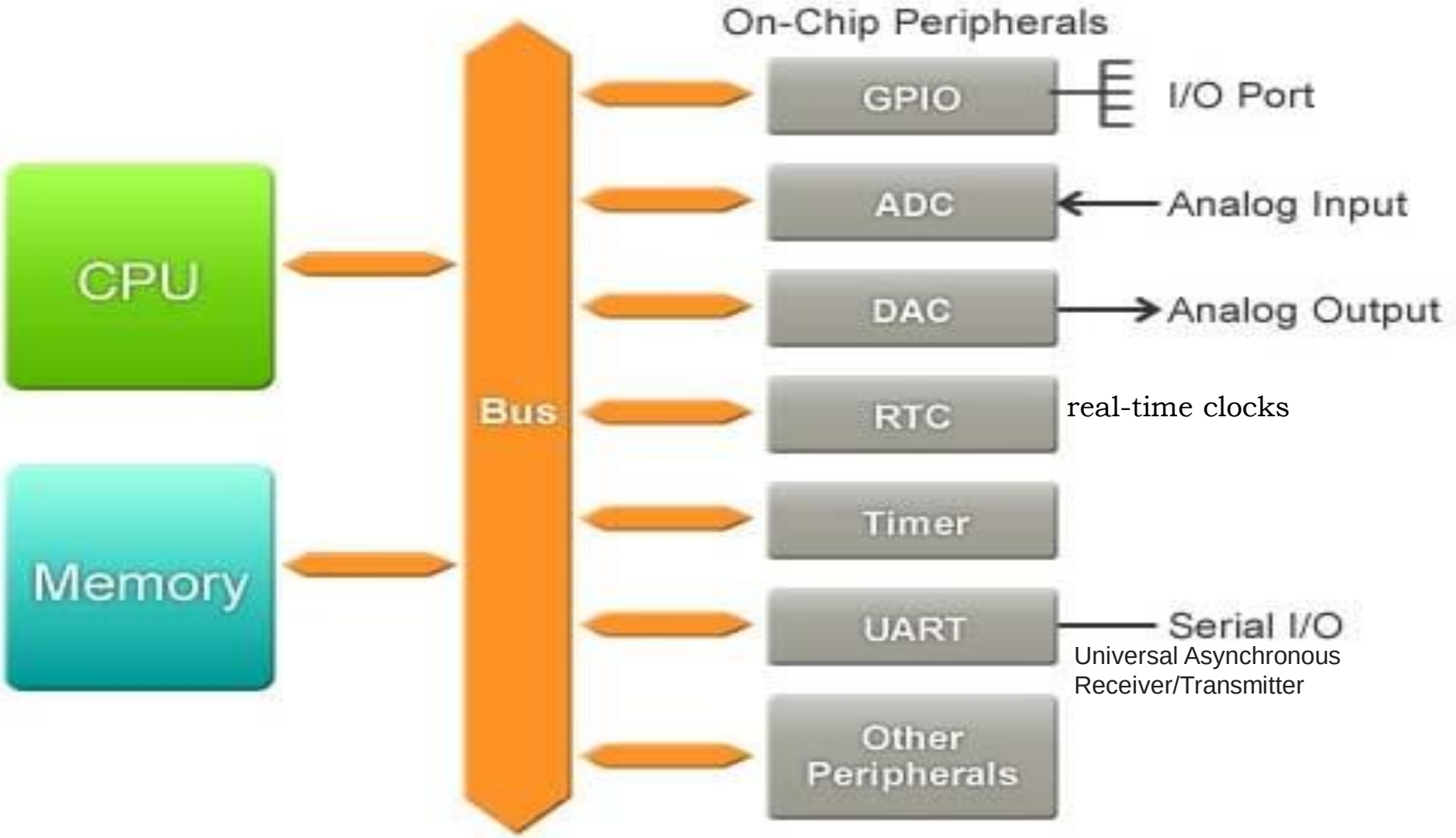
•**Electric vehicle charging stations:**

Some of the common uses of electric vehicle charging stations include: Charging vehicles, Swapping batteries, Parking vehicles

Embedded Microcontroller:

- Embedded MCUs(microcontroller unit) are purpose-built for use in various specific applications and rely on embedded peripherals, various communications protocols, conventional operating systems (OS) or specialized real-time operating systems (RTOS), middleware, application software, and development environments.
- MCUs are found in applications that require repetitive operations such as microwave ovens and traffic lights and in a wide variety of well-defined control applications involving sensors and other inputs.

- An MCU contains one or more CPUs (processor cores) along with memory, input/output (I/O) peripherals, real-time clocks (RTC), timers, and other functions **(Figure 1)**.
- For example, the analog/digital converters (ADCs) convert analog inputs (such as from sensors) into digital values that the CPU can use.
- In contrast, the digital/analog converters (DACs) perform the reverse function and take the processor's digital outputs and convert them to analog values to support various system functions.
- A real-time clock (RTC) and one or more universal asynchronous receiver/transmitters (UARTs) convert parallel signals into serial ones and serial ones into parallel ones.
- Additional peripherals such as I2C interfaces, pulse-width modulators (PWMs), and so on are included to support specific application needs.

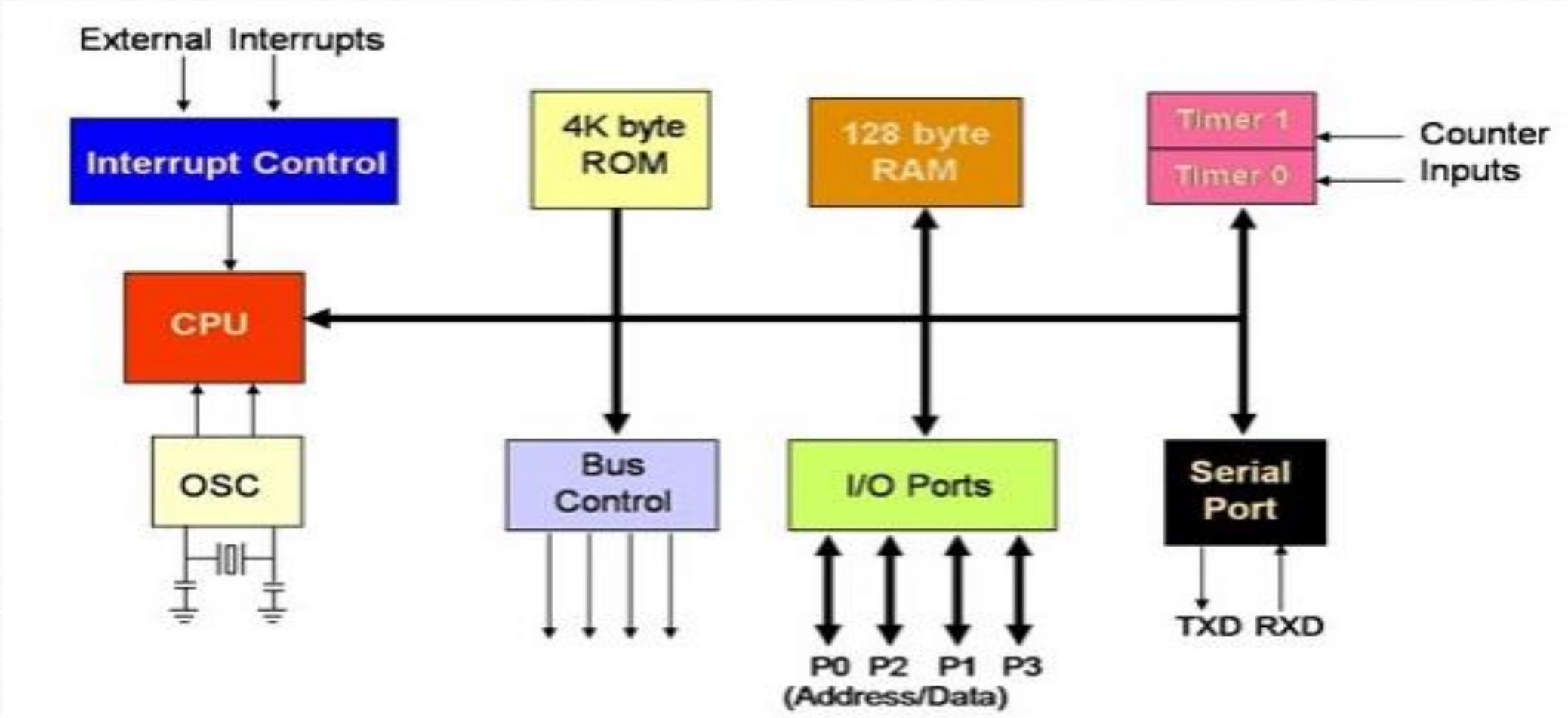

On-Chip Peripherals
CPU
Memory
Bus
GPIO
I/O Port
ADC
Analog Input
DAC
Analog Output
RTC
real-time clocks
Timer
UART
Serial I/O
Universal Asynchronous
Receiver/Transmitter
Other
Peripherals

- MCUs are used because they are cost-effective and compact. Integrating the processor, memory, and various peripherals on a single chip and testing them as a unit may increase the cost of a chip, but it produces a reduced solution size, generally decreases the overall cost of the system, and improves both quality and reliability compared with a collection of individual functional elements.

8051 microcontroller

- 8051 microcontroller is designed by Intel in 1981.
- It is an 8-bit microcontroller.
- It is built with 40 pins DIP (dual inline package), 4kb of ROM storage and 128 bytes of RAM storage, two 16-bit timers.
- It consists of are four parallel 8-bit ports, which are programmable as well as addressable as per the requirement.
- An on-chip crystal oscillator is integrated in the microcontroller having crystal frequency of 12 MHz.

8051 Microcontroller Architecture



CPU (Central Processing Unit):

- CPU act as a mind of any processing machine. It synchronizes and manages all processes that are carried out in microcontroller.
- User has no power to control the functioning of CPU.
- It interprets the program stored in ROM and carries out from storage and then performs it projected duty.
- CPU manage the different types of registers available in 8051 microcontroller.

Interrupts: Interrupts is a sub-routine call that given by the microcontroller when some other program with high priority is request for acquiring the system buses the ‘n’ interrupts occur in current running program.

Types of interrupt in 8051 Microcontroller:

Let's see the five sources of interrupts in 8051 Microcontroller:

- Timer 0 overflow interrupt - TF0
- Timer 1 overflow interrupt - TF1
- External hardware interrupt - INT0
- External hardware interrupt - INT1
- Serial communication interrupt - RI/TI

Memory:

- For operation Micro-controller required a program. This program guides the microcontroller to perform the specific tasks. This program installed in microcontroller required some on chip memory for the storage of the program.
- Microcontroller also required memory for storage of data and operands for the short duration.
- In microcontroller 8051 there is code or program memory of 4 KB that is it has 4 KB ROM and it also comprise of data memory (RAM) of 128 bytes.

Bus : Bus is a group of wires which uses as a communication canal or acts as means of data transfer. The different bus configuration includes 8, 16 or more cables. Therefore, a bus can bear 8 bits, 16 bits all together.

Types of buses in 8051 Microcontroller:

Let's see the two types of bus used in 8051 microcontroller:

- **Address Bus**: 8051 microcontrollers is consisting of 16 bit address bus.
- **Data bus**: 8051 microcontroller is consisting of 8 bits data bus. It is generally be used for transferring the data from one peripherals position to other peripherals.

Input/Output Ports

- Normally microcontroller is used in embedded systems to control the operation of machines in the microcontroller.
- Therefore, to connect it to other machines, devices or peripherals we require I/O interfacing ports in the microcontroller interface.
- For this purpose microcontroller 8051 has 4 input, output ports to connect it to the other peripherals

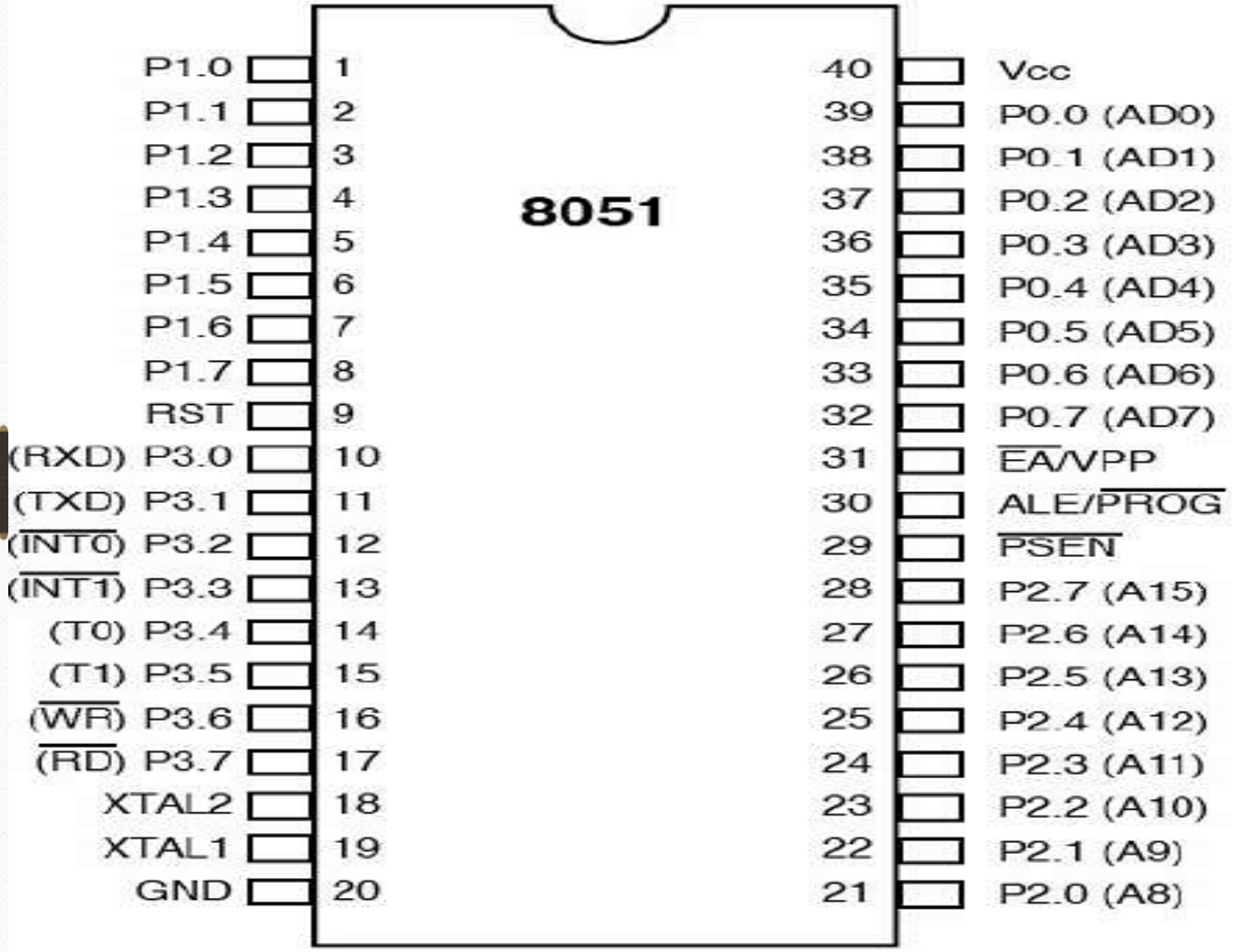
Oscillator:

- As the microcontroller is digital circuit therefore it needs timer for their operation.
- To perform timer operation inside microcontroller it required externally connected or on-chip oscillator.
- Microcontroller is used inside an embedded system for managing the function of devices.
- Therefore, 8051 uses the two 16 bit counters and timers.
- For the operation of this timers and counters the oscillator is used inside microcontroller.

Characteristics of 8051 Microcontroller

1. An 8-bit processor.
2. Data memory or RAM of 128 bytes.
3. Program memory or ROM of 4 KB.
4. 2 timers of 16 bit each.
5. 8-bit data bus.
6. 16-bit address bus.
7. Offers bit addressable format.
8. Special function registers and serial port.
9. 32 input/output lines.

8051 Microcontroller Pin Diagram



Pin Description of 8051 Microcontroller

- **Pins 1 to 8** – These pins are known as **Port 1.** This port doesn't serve any other functions. It is internally pulled up, bi-directional I/O port.
- **Pin 9** – It is a RESET pin, which is used to reset the microcontroller to its initial values.
- **Pins 10 to 17** – These pins are known as **Port 3**. This port serves some functions like interrupts, timer input, control signals, serial communication signals RxD and TxD, etc.

- **P3.0 (RXD) :**

10th pin is RXD (serial data receive pin) which is for serial input. Through this input signal microcontroller receives data for serial communication.

- **P3.1 (TXD) :**

11th pin is TXD (serial data transmit pin) which is serial output pin. Through this output signal microcontroller transmits data for serial communication.

- **P3.2 and P3.3 (INT0′, INT1′) :**

12th and 13th pins are for External Hardware Interrupt 0 and Interrupt 1 respectively. When this interrupt is activated(i.e. when it is low), 8051 gets interrupted in whatever it is doing and jumps to the vector value of the interrupt (0003H for INT0 and 0013H for INT1) and starts performing Interrupt Service Routine (ISR) from that vector location.

- **P3.4 and P3.5 (T0 and T1) :**

14th and 15th pin are for Timer 0 and Timer 1 external input. They can be connected with 16 bit timer/counter.

- **P3.6 (WR') :**

16th pin is for external memory write i.e. writing data to the external memory.

- **P3.7 (RD') :**

17th pin is for external memory read i.e. reading data from external memory.

- **Pins 18 & 19** – These pins are used for interfacing an external crystal to get the system clock.
- **Pin 20** – This pin is used for Grounding.
- **Pins 21 to 28** – These pins are known as Port 2. It serves as I/O port. Higher order address bus signals are also multiplexed using this port.
- **Pin 29** – This is PSEN pin which stands for **Program Store Enable**. It is used to read a signal from the external program memory.

- **Pin 30** – This is ALE pin which stands for **Address Latch Enable**. It is used to demultiplex the address-data signal of port.
- **Pin 31** – This is EA pin which stands for **External Access input.** It is used to enable/disable the external memory interfacing.
- **Pins 32 to 39** – These pins are known as Port 0. It serves as I/O port. Lower order address and data bus signals are multiplexed using this port.
- **Pin 40** – This pin is used to provide power supply to the circuit.

Registers in 8051 Microcontroller

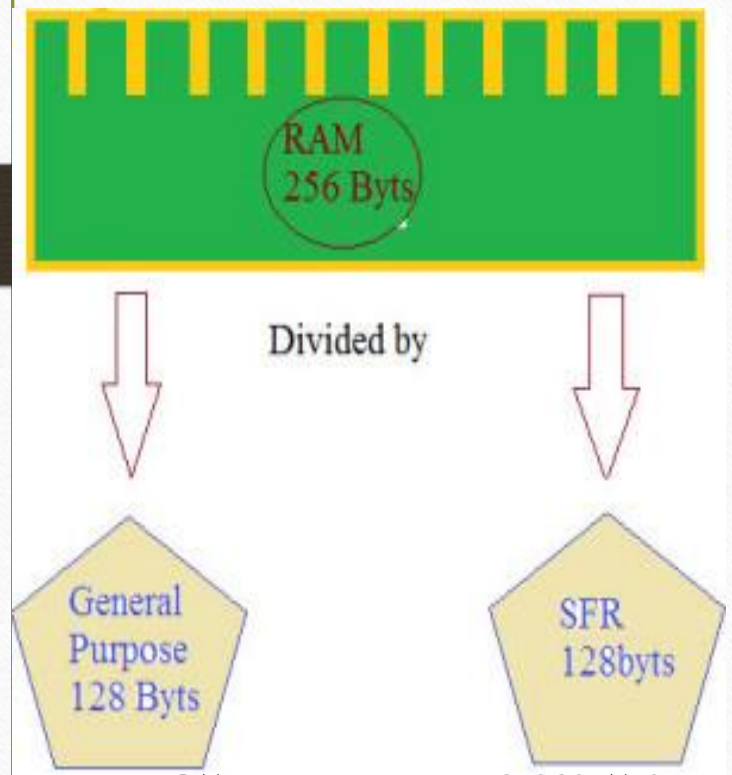
- The register is the main part of the microcontrollers and processors that provide a fast way to collect and store data.
- If we want to manipulate data with a controller or processor by performing addition, subtraction, and so on, we cannot do that directly in the memory, but it needs registers to process and store the data.
- Microcontrollers contain several types of registers that can be classified according to their content or instructions that operate in them.

A register is a small place in a CPU that can store small amounts of the data used for performing various operations such as addition and multiplication and loads the resulting data on the main memory.

Types of Registers

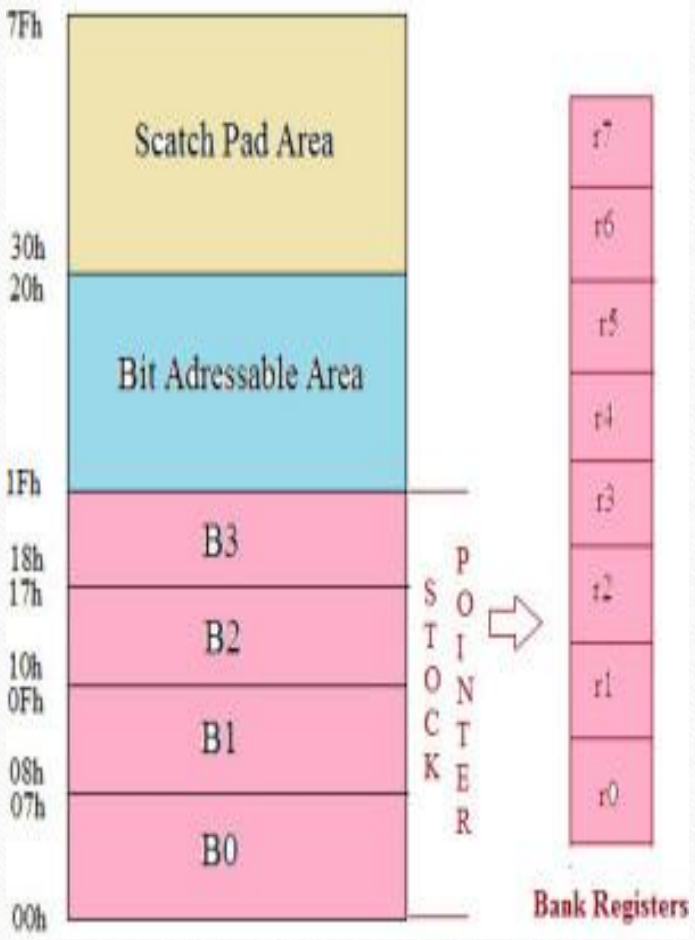
The 8051 microcontroller contains mainly two types of registers:

- General-purpose registers (Byte addressable registers)
- Special function registers (Bit addressable registers)



- The 8051 microcontroller consists of 256 bytes of RAM, which is divided into two ways, such as 128 bytes for general purpose and 128 bytes for special function registers (SFR) memory.
- The memory which is used for general purpose is called as RAM, and the memory used for SFR contains all the peripheral related registers like Accumulator, 'B' register, Timers or Counters, and interrupt related registers.

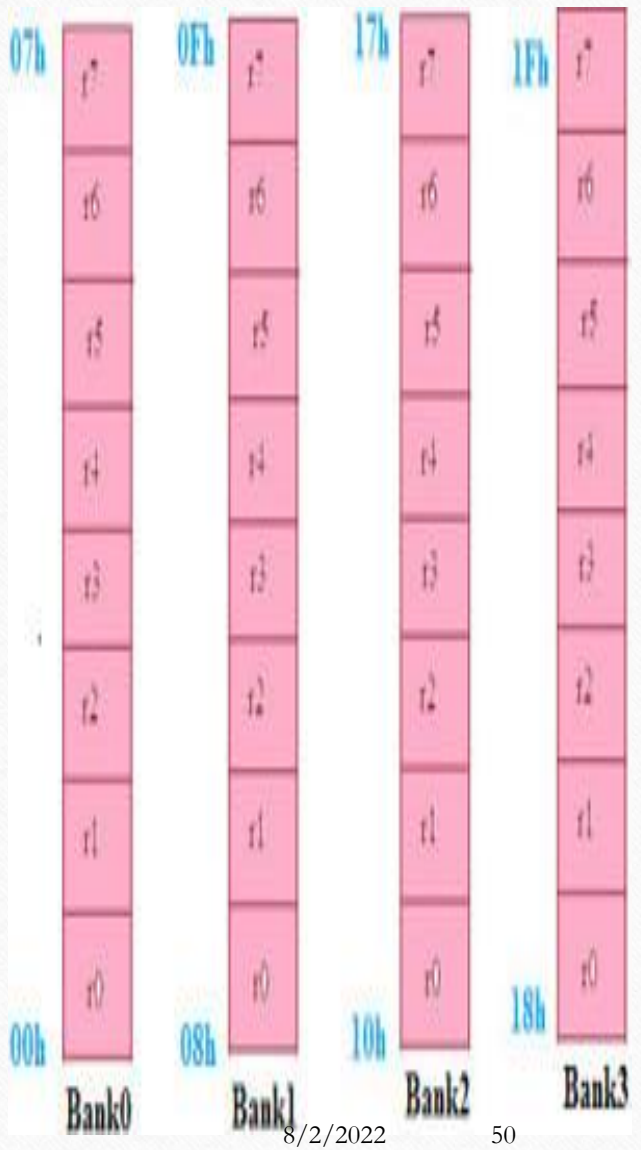
General Purpose Registers



- The general-purpose memory is called as the RAM of the 8051 microcontrollers, which is divided into 3 areas such as banks, bit-addressable area, and scratch-pad area.
- scratchpad refers to a special high-speed memory used to hold small items of data for rapid retrieval.
- The banks contain different general-purpose registers such as R0-R7, and all such registers are byte-addressable registers that store or remove only 1-byte of data.

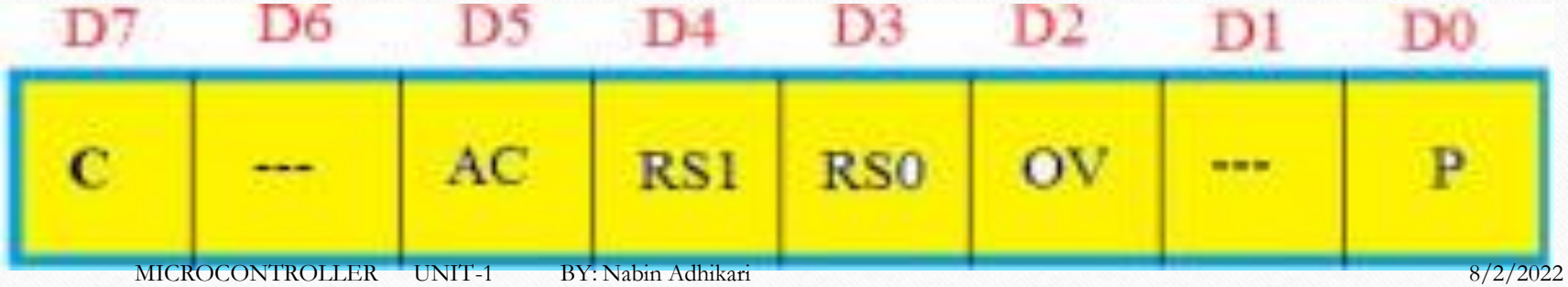
Banks and Registers

- The B0, B1, B2, and B3 stand for banks, and each bank contains eight general-purpose registers ranging from ‘R0’ to ‘R7’.
- All these registers are byte-addressable.
- Data transfer between general-purpose registers to general-purpose registers is not possible.
- These banks are selected by the Program Status Word (PSW) register.

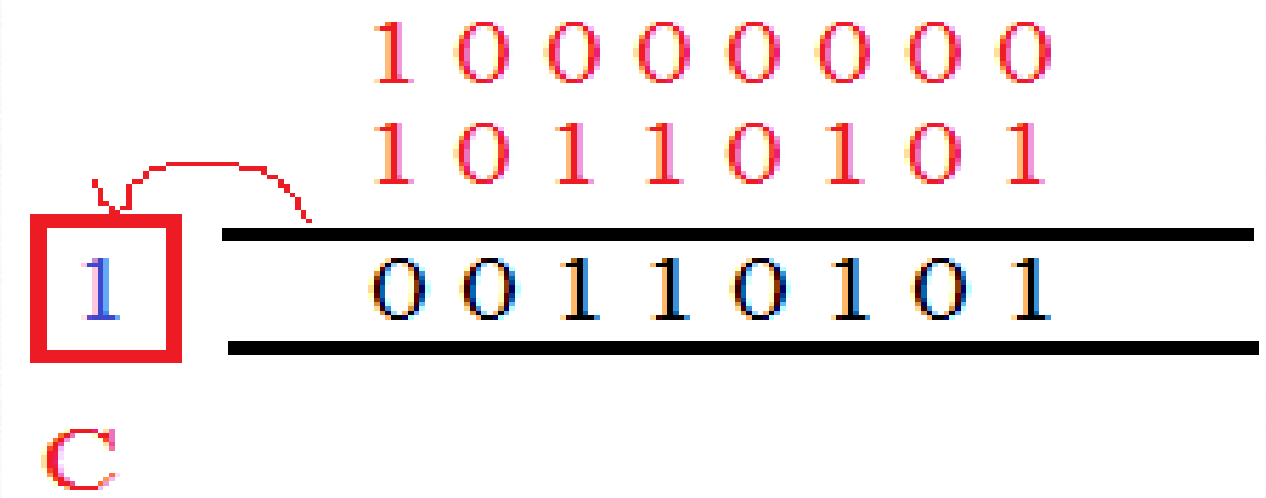


PSW (Program Status Word) Register

- The PSW register is a bit and byte-addressable register. This register reflects the status of the operation that is carried out in the controller.
- The PSW register determines bank selection by an RS1 and RS0, as shown below.
- The physical address of the PSW starts from D0h and the individual bits are accessed with D0h to D7h.



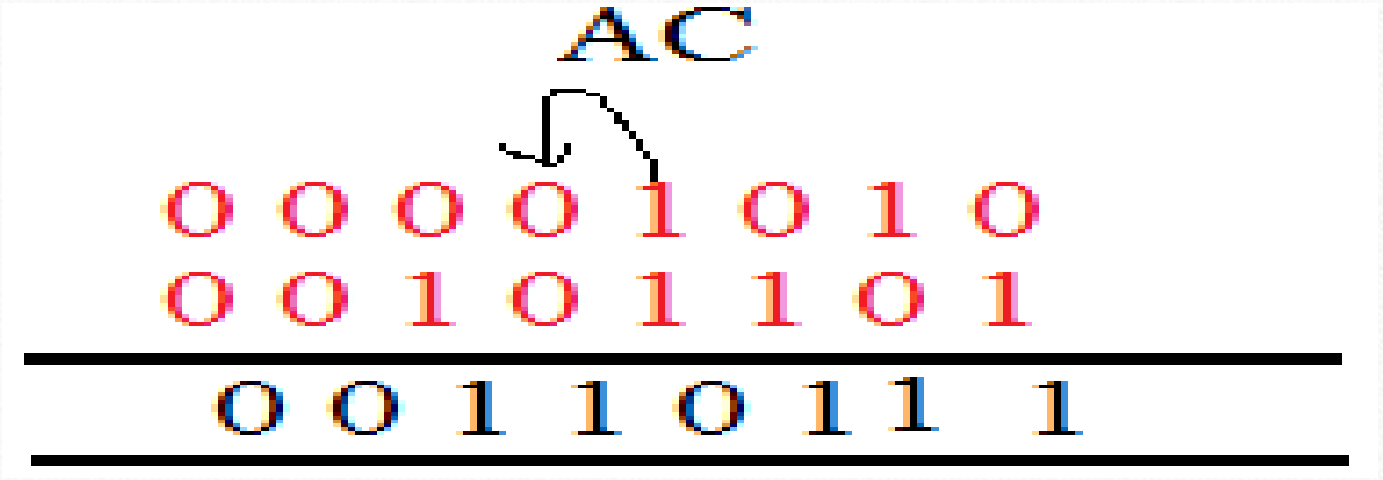
Carry Flag (C): The Address of the Carry flag is D7. This carry flag is affected when the bit is generated from the 7th position.
When C=0 carry resets
C=1 carry sets



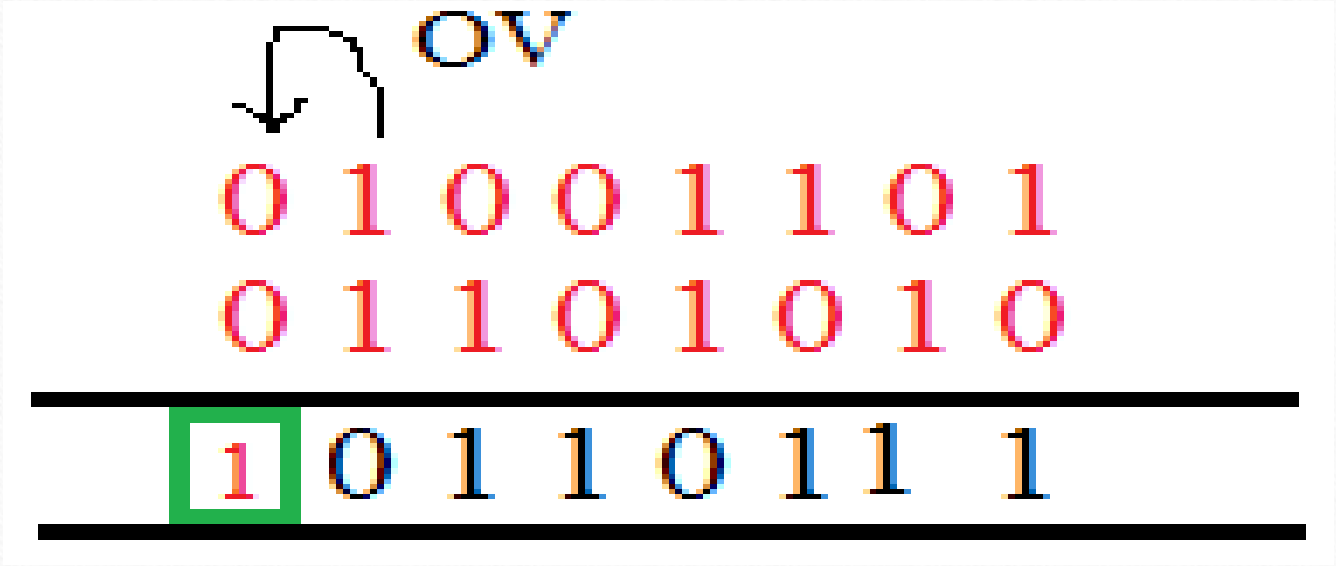
Auxiliary Flag(AC): The address of the auxiliary carry is D5. This auxiliary carry is affected when a bit is generated from the 3rd position to the 4th position.

AC=0 auxiliary is reset

AC=1 auxiliary is set



Overflow Flag (OV): The address of the overflow flag is D2. When a bit is generated from the 6th position to the 7th position, then the overflow flag is affected.
OV=0 overflow flag resets
OV=1 overflow flag sets



Parity Flag (P): The address of the parity flag is D0. While performing arithmetic operations, if the result is 1, then the parity flag is set – otherwise, reset.

RS1 and RS0

The RS1 and RS0, the bits in the PSW(program status word) register, are used to select different memory locations (bank0 to bank3) in the RAM.

RS1	RS0	Value	
0	0	00h	Bank0
0	1	08h	Bank1
1	0	10h	Bank2
1	1	18h	Bank3

Special Function Registers (SFR)

80	P0	SP	DPL	DPH				PCON
88	TCON	TMOD	TL0	TL1	TH0	TH1		
90	P1							
98	SCON	SBUF						
A0	P2							
A8	IE							
B0	P3							
B8	IP							
C0								
C8								
D0	PSW							
D8								
E0	ACC							
E8								
F0	B							
F8								

87	8F	97
9F	A7	AF
B7	B9	C7
CF	D7	DF
E7	EF	F7
FF		

Blue background are I/O port SFRs
Yellow background are control SFRs
Green background are other SFRs

8051 microcontrollers. These like P0, P1, P2, P3, timers or The SFR memory address starts bit-address registers and byte-

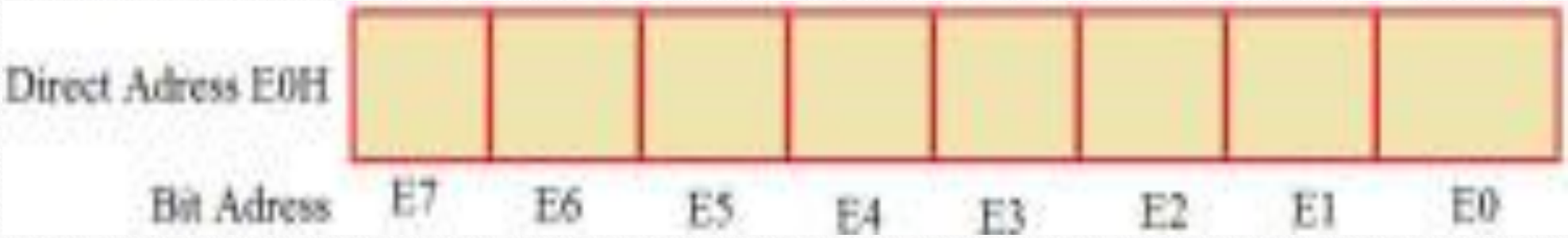
DPL-Data pointer low
DPH-Data pointer high
TMOD-Timer model
TCON-Timer Control
THO-Timer zero high byte
TH1-Timer one high byte
TLO-Timer zero low byte
TL1-Timer one low byte
SCON-Serial port control
SBUF-serial port data buffer
PSW-program status word
PCON-power control
IP-interrupt priority
IE-interrupt enable

Name of the Register	*Function*	*Internal RAM Address (HEX)*
ACC	Accumulator	E0H
B	B Register (for Arithmetic)	F0H
DPH	Addressing External Memory	83H
DPL	Addressing External Memory	82H
IE	Interrupt Enable Control	A8H
IP	Interrupt Priority	B8H
P0	PORT 0 Latch	80H
P1	PORT 1 Latch	90H
P2	PORT 2 Latch	A0H
P3	PORT 3 Latch	B0H
PCON	Power Control	87H
PSW	Program Status Word	D0H
SCON	Serial Port Control	98H
SBUF	Serial Port Data Buffer	99H
SP	Stack Pointer	81H
TMOD	Timer / Counter Mode Control	89H
TCON	Timer / Counter Control	88H
TL0	Timer 0 LOW Byte	8AH
TH0	Timer 0 HIGH Byte	8CH
TL1	Timer 1 LOW Byte	8BH
TH1	Timer 1 HIGH Byte	8DH

The accumulator, B register, Po, P1, P2, P3, IE registers are bit-addressable register remaining all are byte-addressable registers.

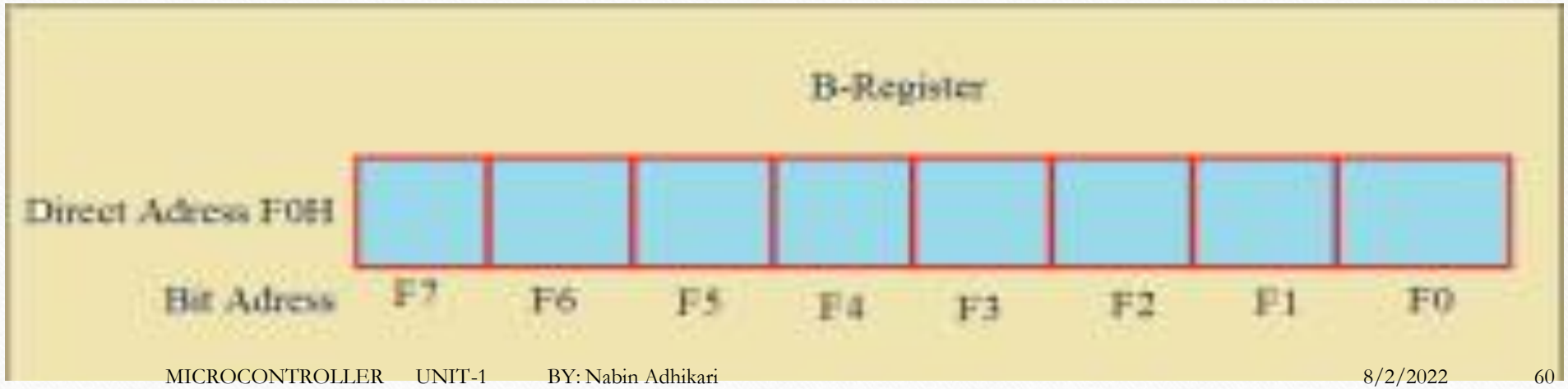
Accumulator

- The accumulator which is also known as ACC or A is a bit as well as a byte-addressable register by an address of the accumulator.
- If you want to use a bit-addressable register, you can use a single bit (E0) of the register and you can use an 8-bit of the accumulator as a byte-addressable register.
- The accumulator holds the results of most Arithmetic and logical operations.



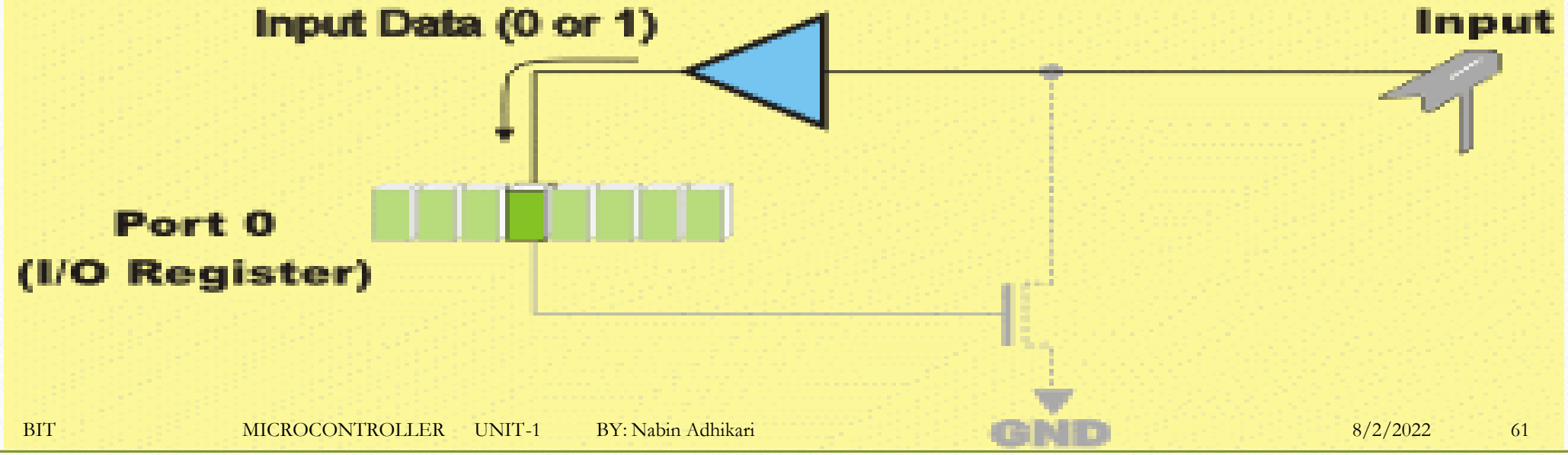
B-Register

- The B-register is a bit and byte-addressable register.
- You can access 1-bit or all 8-bits by a physical address F0h.
- Suppose to access a bit 1, we have to use f1. The B register is only used for multiplication and division operations.



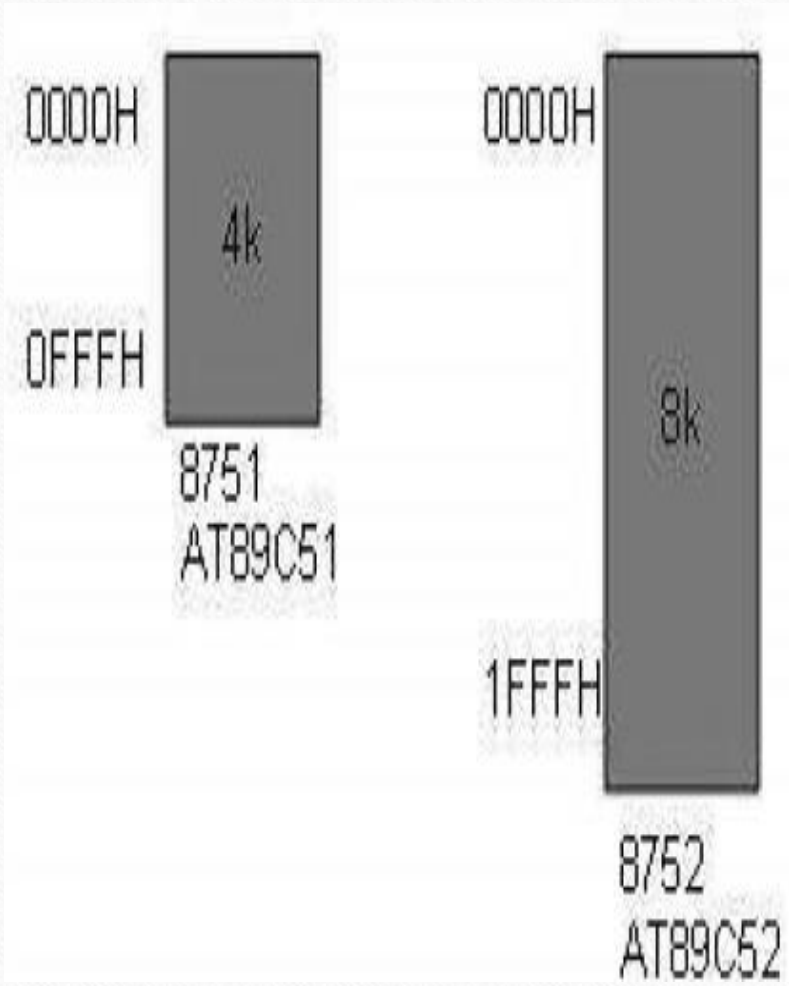
Port Registers

- The 8051 microcontroller consists of 4-input and output ports (P0, P1, P2, and P3) or 32-I/O pins.
- Each pin is designed with a transistor and P registers.
- The pin configuration is very important for a microcontroller that depends on the logic states of the registers.
- The pin configuration as the input given by 1 or output 0 depends on the logic states.



ROM Space in 8051

- Some family members of 8051 have only 4K bytes of on-chip ROM (e.g. 8751, AT8951); some have 8K ROM like AT89C52, and there are some family members with 32K bytes and 64K bytes of on-chip ROM such as Dallas Semiconductor.
- The point to remember is that no member of the 8051 family can access more than 64K bytes of opcode since the program counter in 8051 is a 16-bit register (0000 to FFFF address).
- The first location of the program ROM inside the 8051 has the address of 0000H, whereas the last location can be different depending on the size of the ROM on the chip. Among the 8051 family members, AT8951 has ROM having a memory address of 0000 (first location) to 0FFFH (last location).

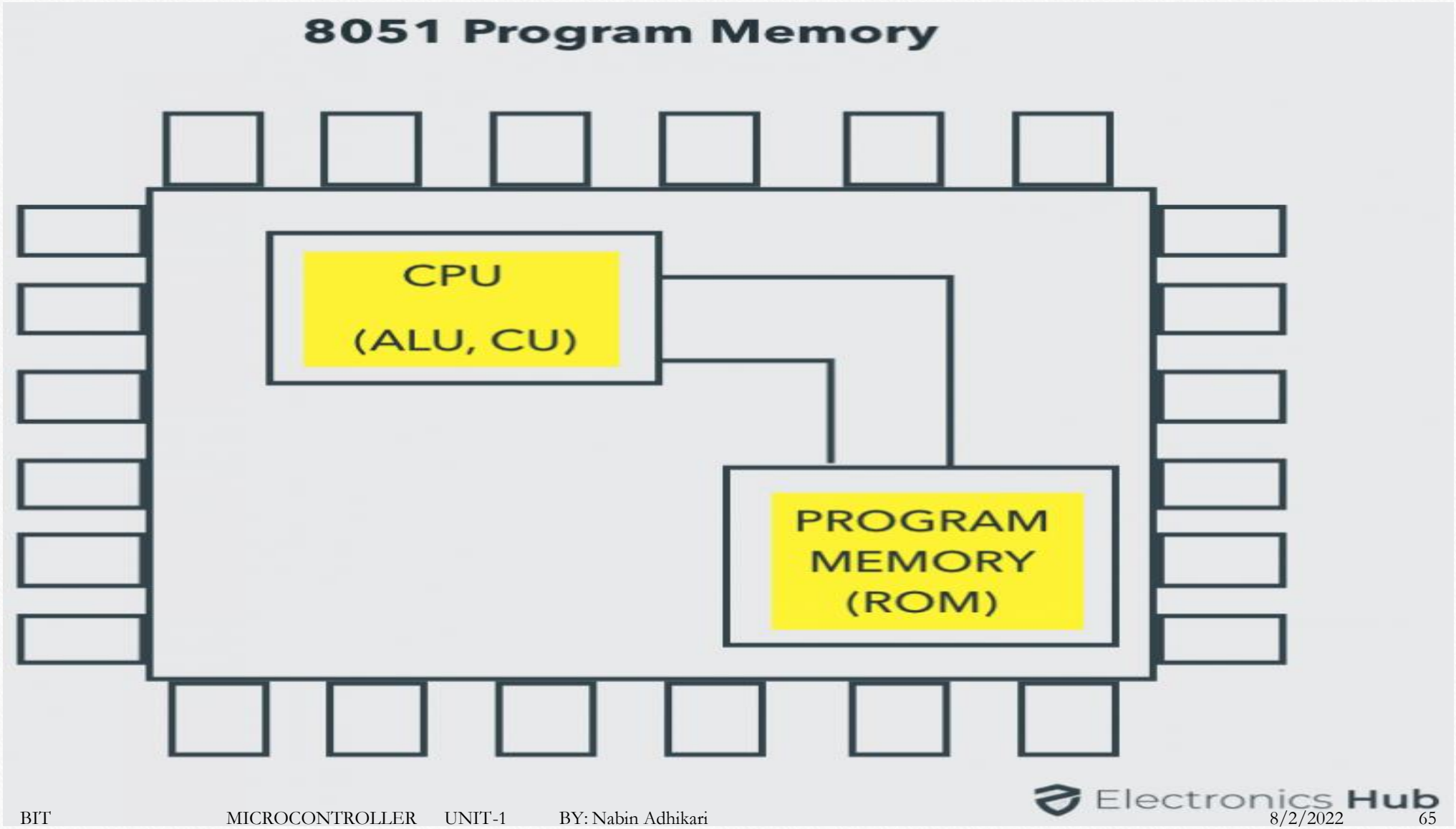


8051 Microcontroller Memory Organization

- The 8051 Microcontroller Memory is separated in Program Memory (ROM) and Data Memory (RAM).
- The Program Memory of the 8051 Microcontroller is used for storing the program to be executed i.e., instructions.
- The Data Memory on the other hand, is used for storing temporary variable data and intermediate results.
- 8051 Microcontroller has both Internal ROM and Internal RAM. If the internal memory is inadequate, you can add external memory using suitable circuits.

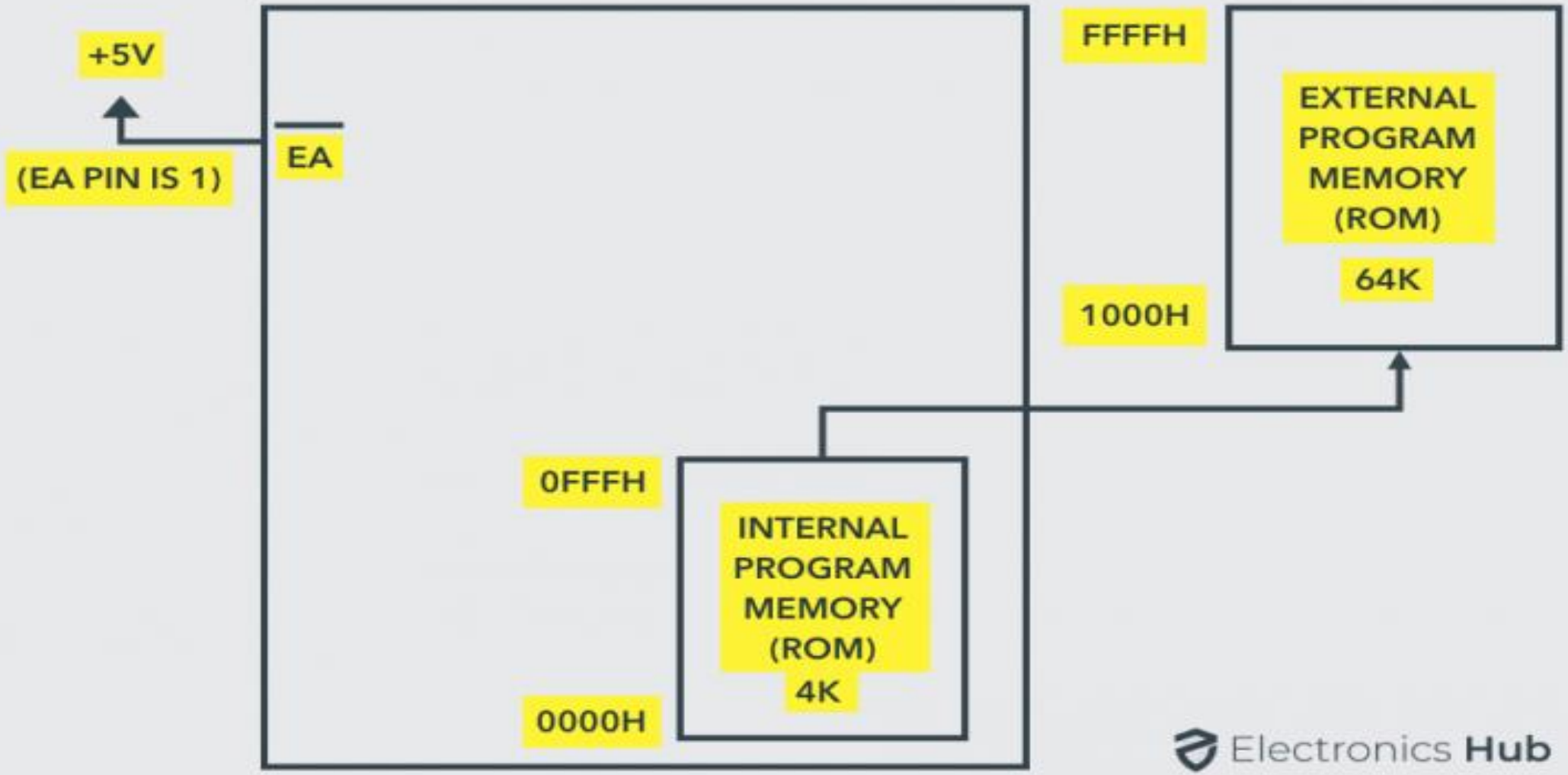
Program Memory (ROM) of 8051 Microcontroller

- In 8051 Microcontroller, the code or instructions to be executed are stored in the Program Memory, which is also called as the ROM of the Microcontroller.
- The original 8051 Microcontroller by Intel has 4KB of internal ROM.
- Some variants of 8051 like the 8031 and 8032 series doesn't have any internal ROM (Program Memory) and must be interfaced with external Program Memory with instructions loaded in it.
- Almost all modern 8051 Microcontrollers, like 8052 Series, have 8KB of Internal Program Memory (ROM) in the form of Flash Memory (ROM) and provide the option of reprogramming the memory.


8051 Program Memory
CPU
(ALU, CU)
PROGRAM
MEMORY
(ROM)
Electronics Hub

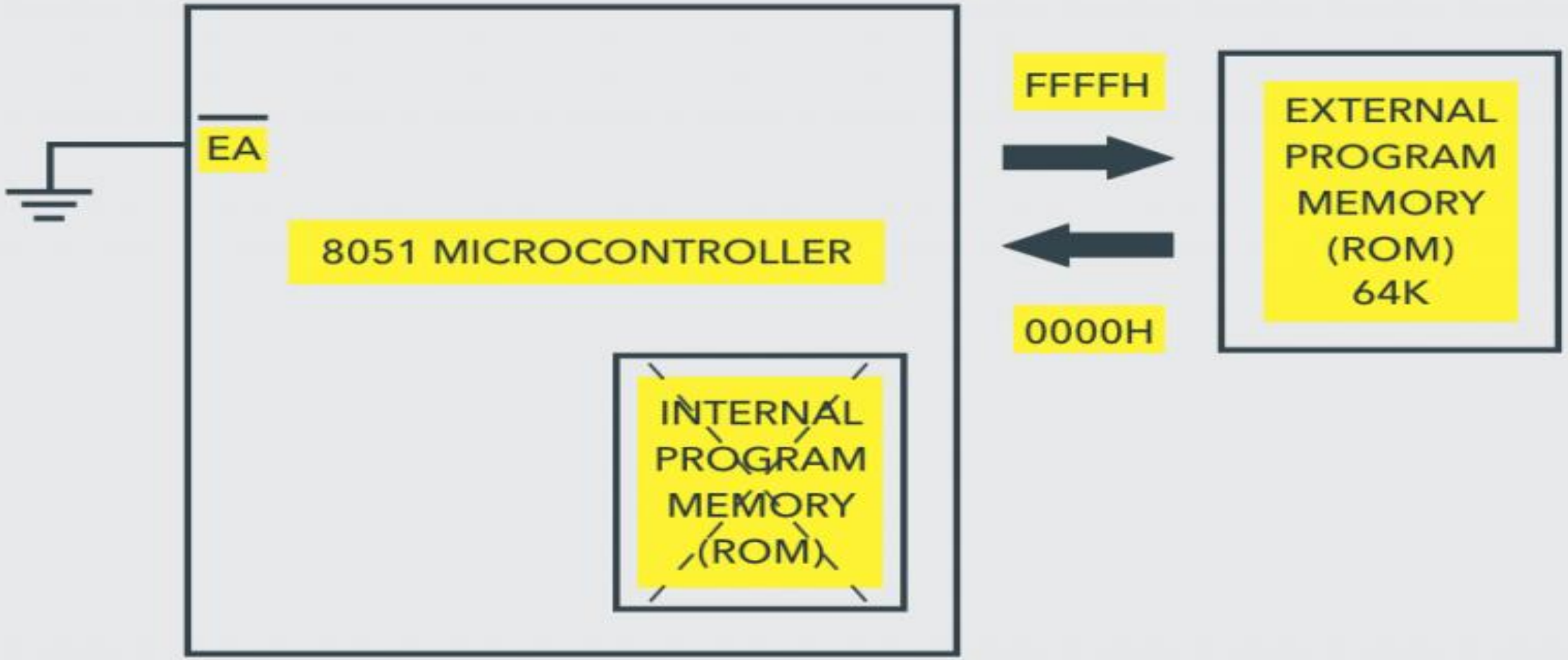
- In case of 4KB of Internal ROM, the address space is 0000H to 0FFFH.
- If the address space i.e., the program addresses exceed this value, then the CPU will automatically fetch the code from the external Program Memory.
- For this, the External Access Pin (EA Pin) must be pulled HIGH i.e., when the EA Pin is high, the CPU first fetches instructions from the Internal Program Memory in the address range of 0000H to 0FFFFH and if the memory addresses exceed the limit, then the instructions are fetched from the external ROM in the address range of 1000H to FFFFH.

Using Both Internal And External Program Memory With 8051



- There is another way to fetch the instructions: ignore the Internal ROM and fetch all the instructions only from the External Program Memory (External ROM).
- For this scenario, the EA Pin must be connected to GND. In this case, the memory addresses of the external ROM will be from 0000H to FFFFH.

Using Only External Program Memory With 8051

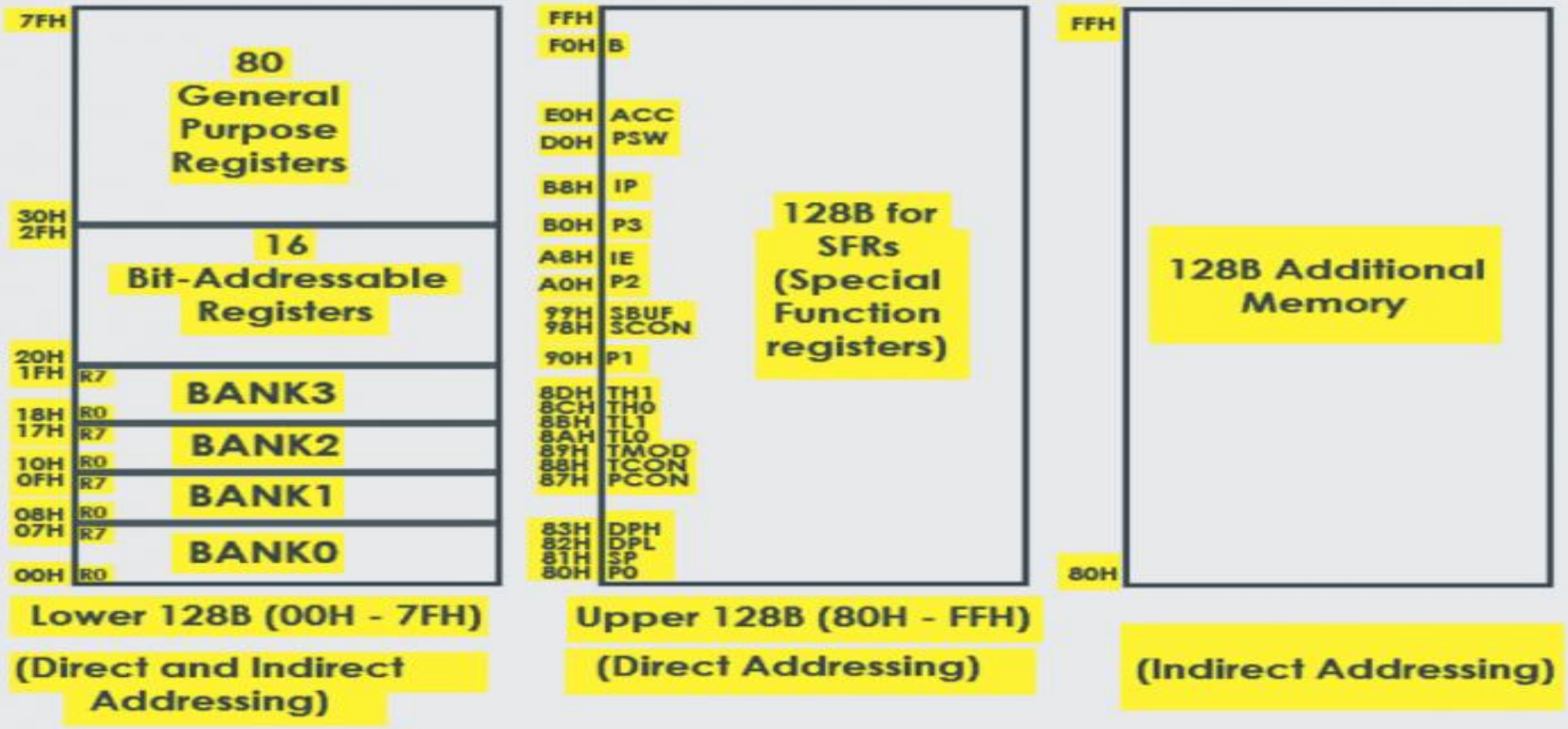


Electronics Hub

Data Memory (RAM) of 8051 Microcontroller

- The Data Memory or RAM of the 8051 Microcontroller stores temporary data and intermediate results that are generated and used during the normal operation of the microcontroller.
- Original Intel's 8051 Microcontroller had 128B of internal RAM.
- But almost all modern variants of 8051 Microcontroller have 256B of RAM. In this 256B, the first 128B i.e., memory addresses from 00H to 7FH is divided in to Working Registers (organized as Register Banks), Bit – Addressable Area and General Purpose RAM (also known as Scratchpad area).
- In the first 128B of RAM (from 00H to 7FH), the first 32B i.e., memory from addresses 00H to 1FH consists of 32 Working Registers that are organized as four banks with 8 Registers in each Bank.

Data Memory (RAM) Of 8051 Microcontroller



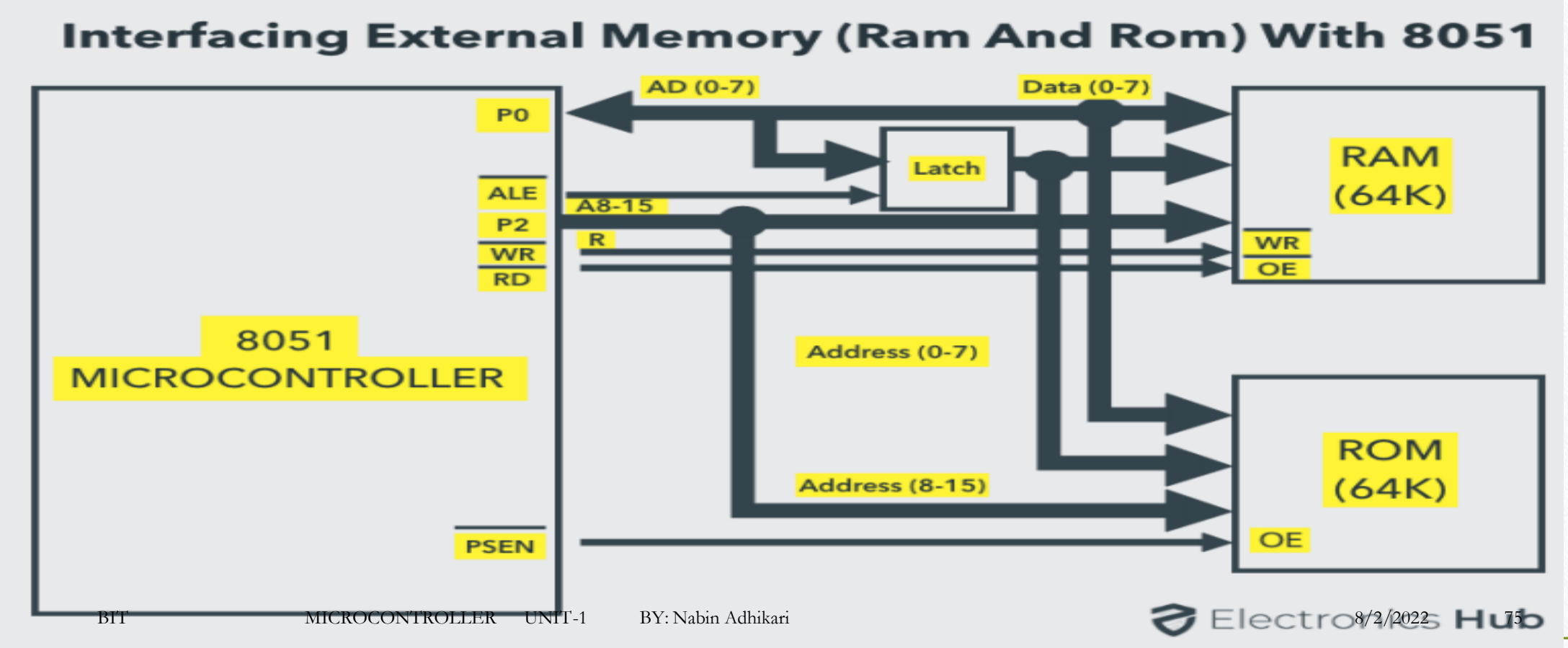
- The 4 banks are named as Bank0, Bank1, Bank2 and Bank3. Each Bank consists of 8 registers named as R0 – R7. Each Register can be addressed in two ways: either by name or by address.
- To address the register by name, first the corresponding Bank must be selected. In order to select the bank, we have to use the RS0 and RS1 bits of the Program Status Word (PSW) Register (RS0 and RS1 are 3rd and 4th bits in the PSW Register).
- When addressing the Register using its address i.e., 12H for example, the corresponding Bank may or may not be selected. (12H corresponds to R2 in Bank2).
- The next 16B of the RAM i.e., from 20H to 2FH are Bit – Addressable memory locations.
- There are totally 128 bits that can be addressed individually using 00H to 7FH or the entire byte can be addressed as 20H to 2FH.

- For example 32H is the bit 2 of the internal RAM location 26H.
- The final 80B of the internal RAM i.e., addresses from 30H to 7FH, is the general purpose RAM area which are byte addressable.
- These lower 128B of RAM can be addressed directly or indirectly.
- The upper 128B of the RAM i.e., memory addresses from 80H to FFH is allocated for Special Function Registers (SFRs). SFRs control specific functions of the 8051 Microcontroller.
- Some of the SFRs are I/O Port Registers (P0, P1, P2 and P3), PSW (Program Status Word), A (Accumulator), IE (Interrupt Enable), PCON (Power Control), etc.

Interfacing External Memory with 8051 Microcontroller

- It is always good to have an option to expand the capabilities of a Microcontroller, whether it is in terms of Memory or IO or anything else.
- Such expansion will be useful to avoid design throttling. We have seen that a typical 8051 Microcontroller has 4KB of ROM and 128B of RAM (most modern 8051 Microcontroller variants have 8K ROM and 256B of RAM).
- The designer of an 8051 Microcontroller based system is not limited to the internal RAM and ROM present in the 8051 Microcontroller.
- There is a provision of connecting both external RAM and ROM i.e., Data Memory and Program.
- The reason for interfacing external Program Memory or ROM is that complex programs written in high – level languages often tend to be larger and occupy more memory.
- Another important reason is that chips like 8031 or 8032, which doesn't have any internal ROM, have to be interfaced with external ROM.

- A maximum of 64KB of Program Memory (ROM) and Data Memory (RAM) each can be interface with the 8051 Microcontroller.
- The following image shows the block diagram of interfacing 64KB of External RAM and 64KB of External ROM with the 8051 Microcontroller.



- An important point to remember when interfacing external memory with 8051 Microcontroller is that Port 0 (P0) cannot be used as an IO Port as it will be used for multiplexed address and data bus (A0 – A7 and D0 – D7). Not always, but Port 2 may be used as higher byte of the address bus.
- In this tutorial, we have seen the 8051 Microcontroller Memory Organization, Program Memory, Data Memory, Internal ROM and RAM and how to interface external Memory (ROM and RAM) with 8051 Microcontroller.