Microprocessors & Microcontrollers (ECE3004)

Module – 01 (Part – 02)

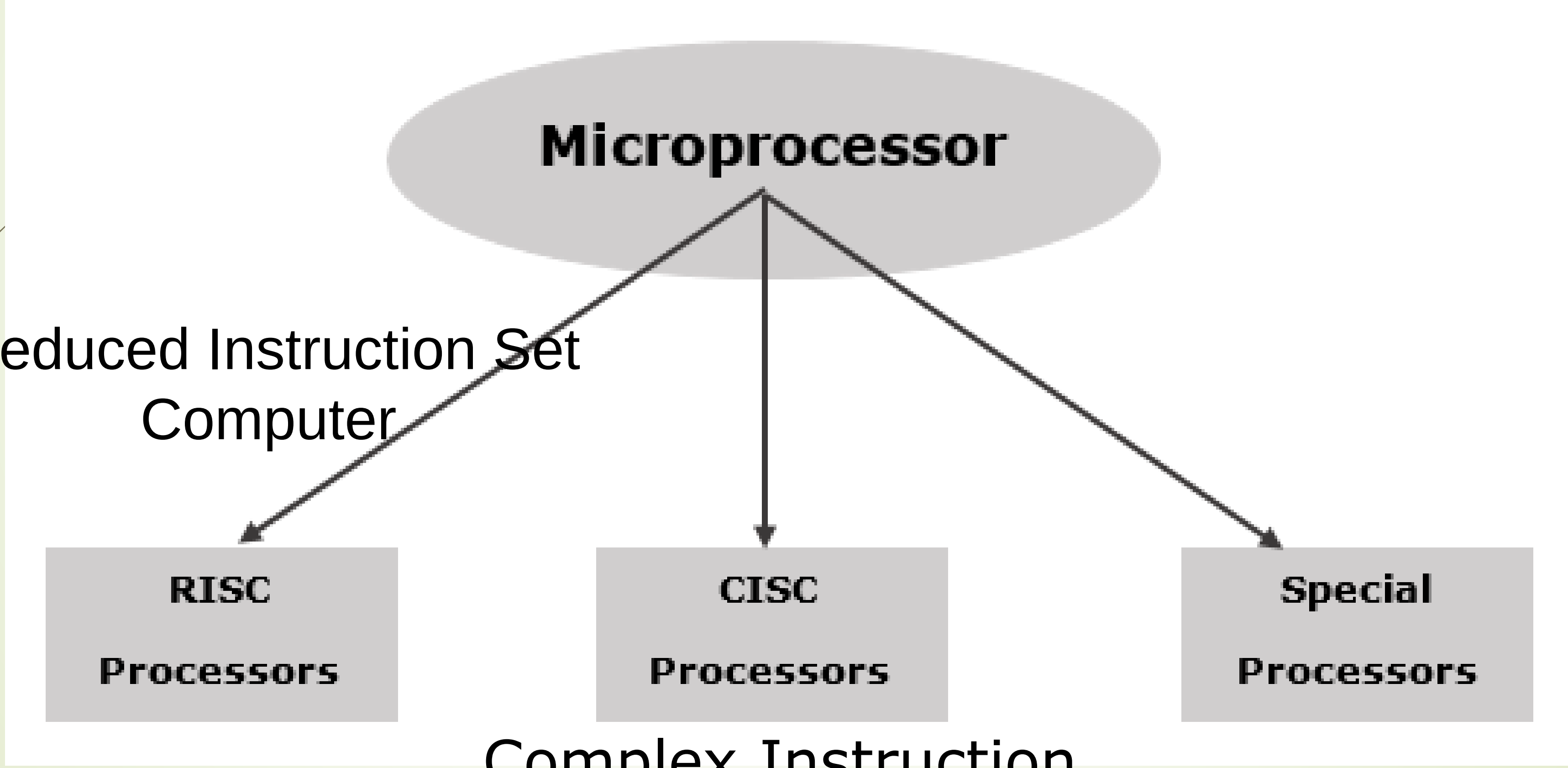
Evolution of Microprocessor, Internal microprocessor (8086 to Pentium), Introduction to Microprocessors and Microcontrollers Introduction to development tools, logic analyzer, in-circuit emulator

Dr. Susant Kumar Panigrahi
Assistant Professor
School of Electrical & Electronics Engineering

VIT BHOPAL

Types of Microprocessor (μP)

Based on Instruction Set



Reduced Instruction Set Computer

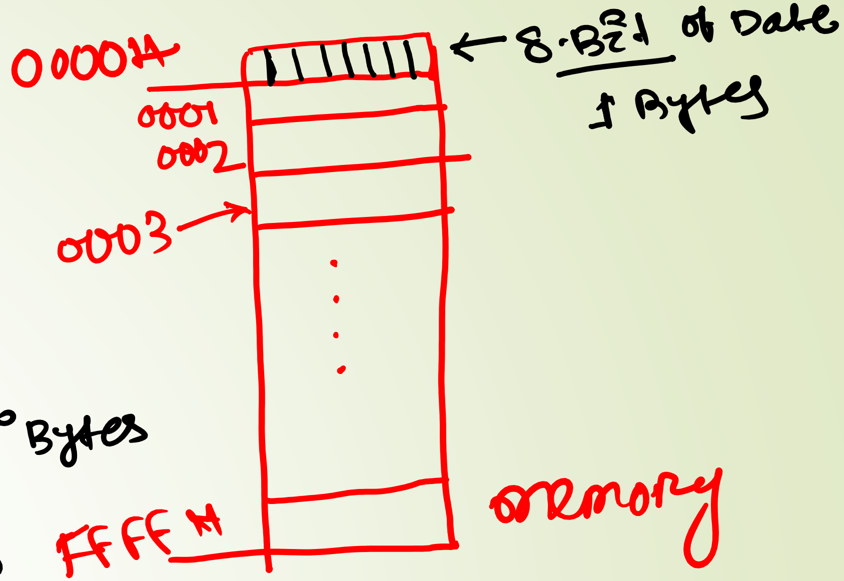
Complex Instruction Set Computers

(1) Length of Address Bus =

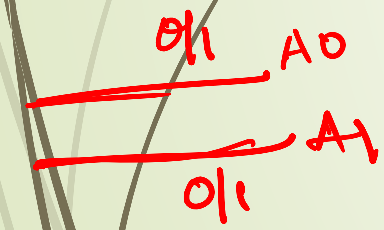
= 16 Bit ✓ → Size of Memory $= 2^{16}$ Bytes

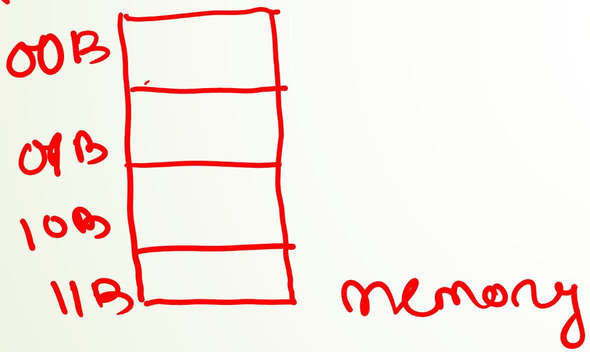
$= 2^{6} \cdot 2^{10}$

$= 64$ KB



(2) Length of Address Bus = 2 Bit





(3) Length of Address Bus = 20 Bits

Size of memory = ? $= 2^{20}$ Bytes

$= 1$ MB

(4) Given memory Size = 64 MB find Length of Address Bus?

"26 Bits"

L·A·B = 26

Length of Add Bus $= \log_2 (2^6 \times 2^{20})$ 64 MB

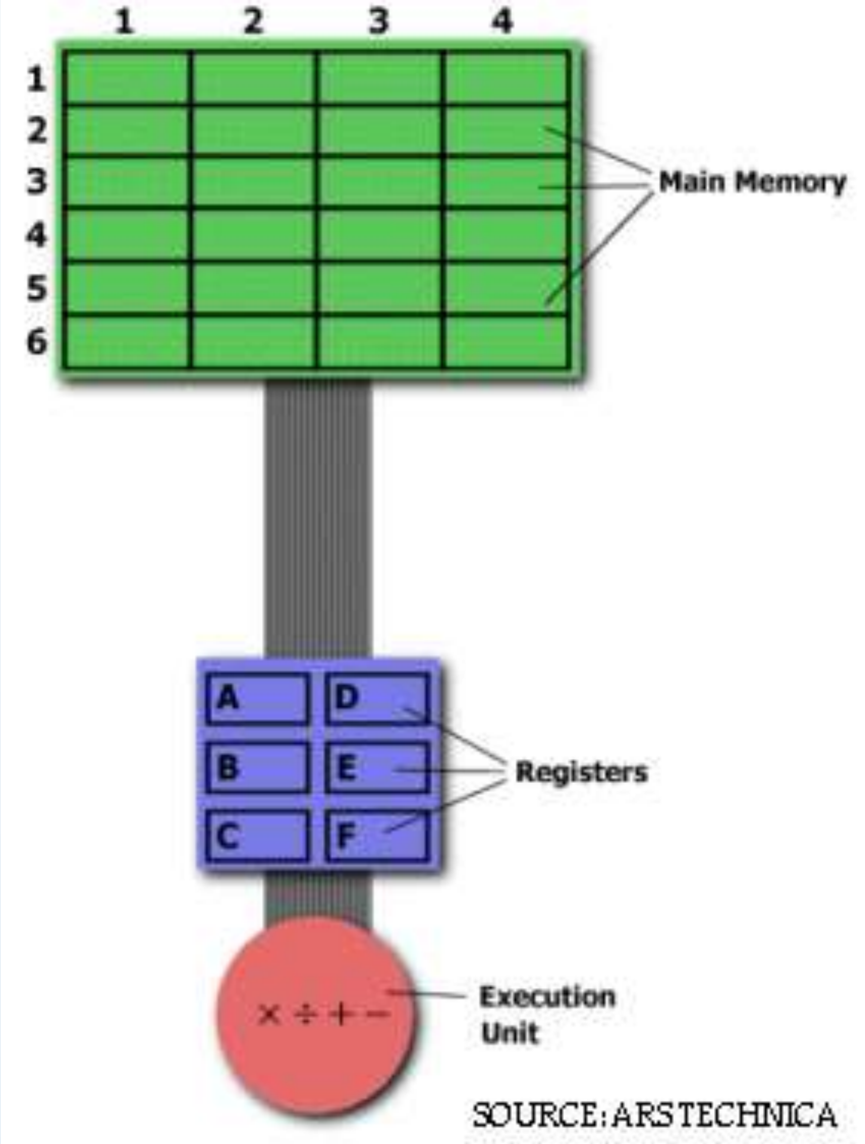
CISC vs RISC A race to increase the CPU performance

Example:

Multiplying 2 Nos. in Memory:

- The main memory is divided into locations numbered from (row) 1: (column) 1 to (row) 6: (column) 4.
- The execution unit can only operate on data that has been loaded into one of the six registers (A, B, C, D, E, or F).

- **Task**: Let's say we want to find the product of two numbers - one stored in location 2:3 and another stored in location 5:2 - and then store the product back in the location 2:3.



CISC vs RISC A race to increase the CPU performance

Multiplying 2 Nos. in Memory (CISC):

- The primary goal of CISC architecture is to complete a task in as few lines of assembly as possible.
- This is achieved by building processor hardware that is capable of understanding and executing a series of operations.
- For this particular task, a CISC processor would come prepared with a specific instruction (we'll call it "MULT").
- Thus, the entire task of multiplying two numbers can be completed with one instruction:

$$MULT\ 2{:}3, 5{:}2$$

- MULT is what is known as a "complex instruction." It operates directly on the computer's memory banks and does not require the programmer to explicitly call any loading or storing functions.

The primary advantages of this system is that:

i. The compiler has to do very little work to translate a high-level language statement into assembly.
ii. Because the length of the code is relatively short.
iii. Very little RAM is required to store instructions.

CISC vs RISC

A race to increase the CPU performance

Multiplying 2 Nos. in Memory (RISC):

- RISC processors only use simple instructions that can be executed within one clock cycle.
- Thus, the "MULT" command described above could be divided into three separate commands:
  - ✓ **LOAD**: which moves data from the memory bank to a register.
  - ✓ **PROD**: which finds the product of two operands located within the registers.
  - ✓ **STORE**: which moves data from a register to the memory banks.

$$LOAD\ A, 2:3$$
$$LOAD\ B, 5:2$$
$$PROD\ A, B$$
$$STORE\ 2:3, A$$

- At first, this may seem like a much less efficient way of completing the operation. Because there are more lines of code, more RAM is needed to store the assembly level instructions.

The primary advantages of this system is that:

i. Because each instruction requires only one clock cycle to execute, the entire program will execute in approximately the same amount of time as the multi-cycle "MULT" command.
ii. RISC "reduced instructions" require less transistors of hardware space than the complex instructions, leaving more room for general purpose registers.
iii. Pipelining is also possible.

CISC vs RISC

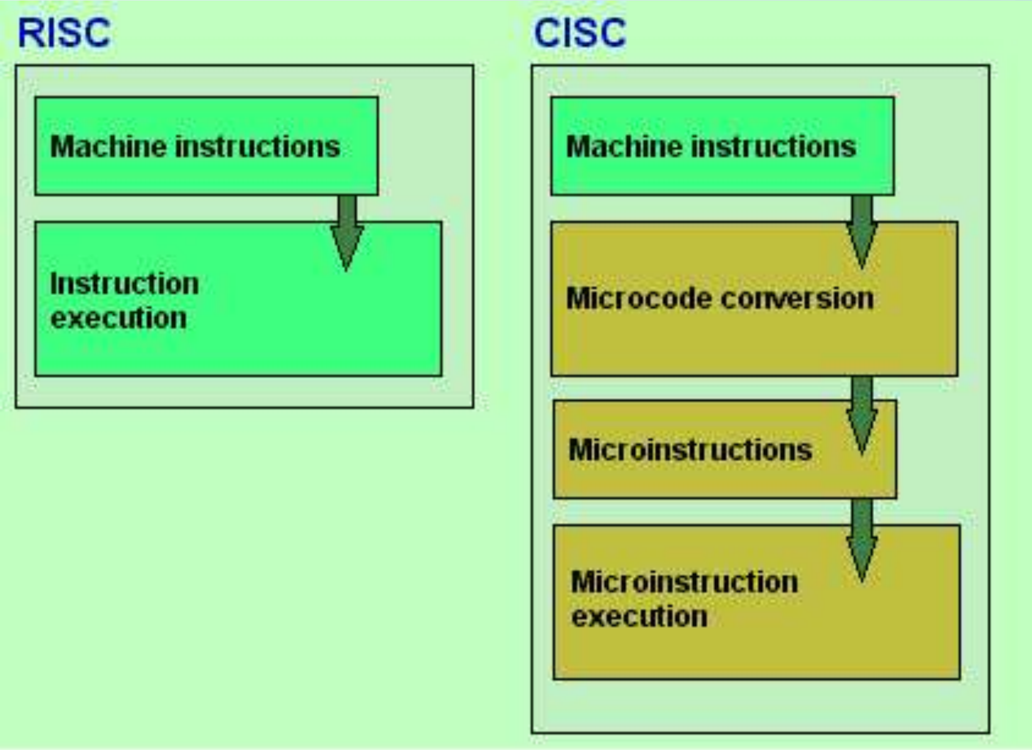
A race to increase the CPU performance

The Performance Equation:

$$\frac{\text{time}}{\text{program}} = \frac{\text{time}}{\text{cycle}} \times \frac{\text{cycles}}{\text{instruction}} \times \frac{\text{instructions}}{\text{program}}$$

CISC: Attempts to minimize the number of instructions per program, sacrificing the number of cycles per instruction.

RISC: Does the opposite, reducing the cycles per instruction at the cost of the number of instructions per program.



CISC vs RISC

A race to increase the CPU performance

RISC	CISC
Focus on software	Focus on hardware
Uses only Hardwired control unit	Uses both hardwired and micro programmed control unit
Transistors are used for more registers	Transistors are used for storing complex Instructions
Fixed sized instructions	Variable sized instructions
Can perform only Register to Register Arithmetic operations	Can perform REG to REG or REG to MEM or MEM to MEM
Requires more number of registers	Requires less number of registers
Code size is large	Code size is small
A instruction execute in single clock cycle	Instruction take more than one clock cycle
A instruction fit in one word	Instruction are larger than size of one word

CISC:

IBM 370/168 – It was introduced in the year 1970. CISC design is a 32 bit processor and four 64-bit floating point registers.
VAX 11/780 – CISC design is a 32-bit processor and it supports many numbers of addressing modes and machine instructions which is from Digital Equipment Corporation.
Intel 80486 – It was launched in the year 1989 and it is a CISC processor, which has instructions varying lengths from 1 to 11 and it will have 235 instructions.

RISC:

Power PC 601, 604, 615, 620

Special Processor

These are the processors which are designed for some special purposes.

Coprocessor

- A coprocessor is a specially designed microprocessor, which can handle its particular function many times faster than the ordinary microprocessor.

For example – Math Coprocessor.

- Some Intel math-coprocessors are –
- 8087-used with 8086
- 80287-used with 80286
- 80387-used with 80386

Special Processor

These are the processors which are designed for some special purposes.

Input/output Processor

- It is a specially designed microprocessor having a local memory of its own, which is used to control I/O devices with minimum CPU involvement.

For example –

- DMA (direct Memory Access) controller
- Keyboard/mouse controller
- Graphic display controller
- SCSI port controller

Special Processor

DSP (Digital Signal Processor)

- This processor is specially designed to process the analog signals into a digital form. This is done by sampling the voltage level at regular time intervals and converting the voltage at that instant into a digital form. This process is performed by a circuit called an analogue to digital converter, A to D converter or ADC.
- **A DSP contains the following components** –
  - **Program Memory** – It stores the programs that DSP will use to process data.
  - **Data Memory** – It stores the information to be processed.
  - **Compute Engine** – It performs the mathematical processing, accessing the program from the program memory and the data from the data memory.
- **Input/output** – It connects to the outside world.
- **Applications**
  - Sound and music synthesis
  - Audio and video compression
  - Video signal processing
  - 2D and 3d graphics acceleration.
- **For example** – Texas Instrument's TMS 320 series, e.g., TMS 320C40, TMS320C50.

Evolution of Microprocessor

Intel 4004

Year of introduction 1971

- 4-bit microprocessor
- 4 KB main memory
- 45 instructions
- PMOS technology

It was first programmable device which was used in calculators

Intel 8008

Year of introduction 1972

- 8-bit version of 4004
- 16 KB main memory
- 48 instructions
- PMOS technology
- Slow

Intel 8080

Year of introduction 1973

- 8-bit microprocessor
- 64 KB main memory
- 2 microseconds clock cycle time
- 500,000 instructions/sec.
- 10X faster than 8008
- NMOS technology
- Drawback was that it needed three power supplies.
- Small computers (Microcomputers) were designed in mid 1970’s using 8080 as CPU.

Intel 8085

Year of introduction 1975

- 8-bit microprocessor-upgraded version of 8080
- 64 KB main memory
- 1.3 microseconds clock cycle time
- 246 instructions
- Intel sold 100 million copies of this 8-bit microprocessor
- uses only one +5v power supply.

Intel 8086

Year of introduction 1978 for 8086 and 1979 for 8088

- 16-bit microprocessors
- Data bus width of 8086 is 16 bit and 8 bit for 8088
- 1 MB main memory
- 400 nanoseconds clock cycle time
- 6 byte instruction cache for 8086 and 4 byte for 8088
- Other improvements included more registers and additional instructions
- In 1981 IBM decided to use 8088 in its personal computer

Other Microprocessors & Comparison

Processor Name	Introduced Date	Transistors	Microns	Clock Speed	Size of Data	MIPS
8080	**1974**	**6000**	**6**	**2 MHz**	**8 bits**	**0.64**
8088	1979	29000	3	5 MHz	16 bits 8-bit bus	0.33
80286	1982	134,000	1.5	6 MHz	16 bits	1
80386	1985	275,000	1.5	16 MHz	32 bits	5
80486	1989	1,200,000	1	25 MHz	32 bits	20
Pentium	1993	3,100,000	0.8	60 MHz	32 bits 64-bit bus	100
Pentium II	1997	7,500,000	0.35	233 MHz	32 bits 64-bit bus	~300
Pentium III	1999	9,500,000	0.25	450 MHz	32 bits 64-bit bus	~510
Pentium 4	2000	42,000,000	0.18	1.5 GHz	32 bits 64-bit bus	~1700
Pentium 4 "Prescott"	2004	**125,000,000**	0.09	3.6GHz	32 bits 64-bit bus	~7000

Components of Computer System

Computer System

Three Main Component:

1. **Processor** [To execute Program]
2. **Memory** [To store/retrieve data/program]
3. **I/O** [To provide input/display result]

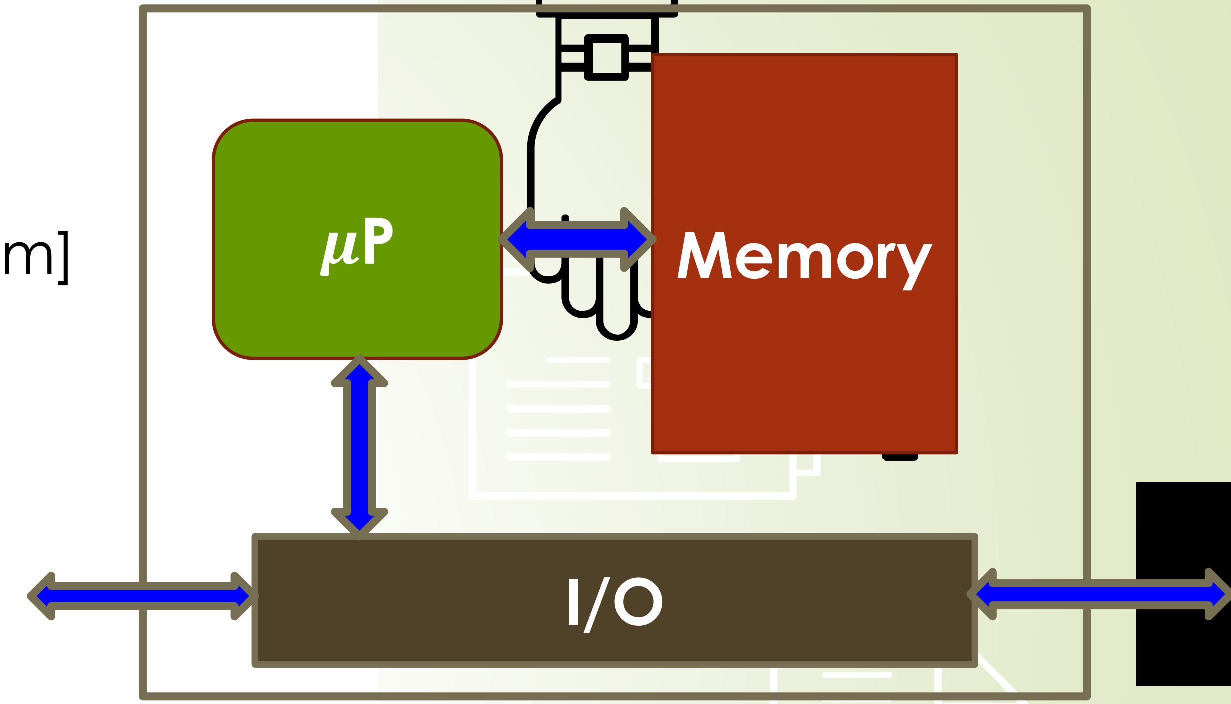


Fig.: Computing System

1. **Processor** : The computers are used for computing and the main task of computing is done by processor or CPU (that contains an ALU).

Computing needs ***Programing.***

Programming μP

- The main job of the CPU is to execute programs using the **fetch-decode-execute** cycle (also known as the **instruction cycle**).

I. **Fetch**: In this operation the instruction from the memory is fetched. Basically, the next instructio to be executed is fetched.

II. **Decode**: Decoding is not about converting instructions to binary (hex) equivalent.

Decoding: 3-types of programming language:

a = b + c; [High-level Language]

ADD B, C; [Assembly Language]

0001001001….. 10 [Machine Level Language]

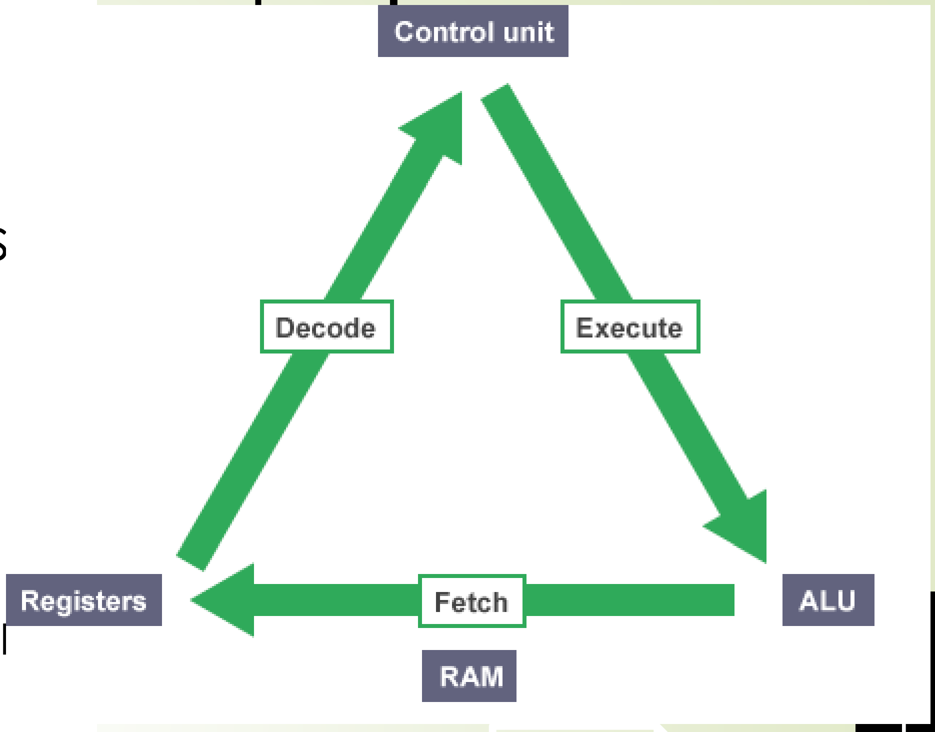


Fig.: Instruction Cycle

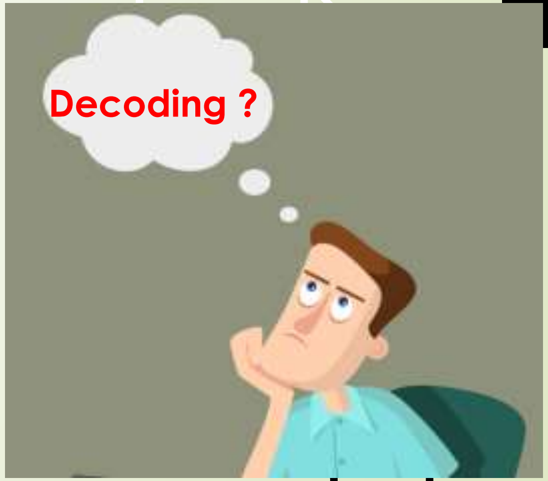
Decoding ....

High Level Language → Compiler → Machine Level Language

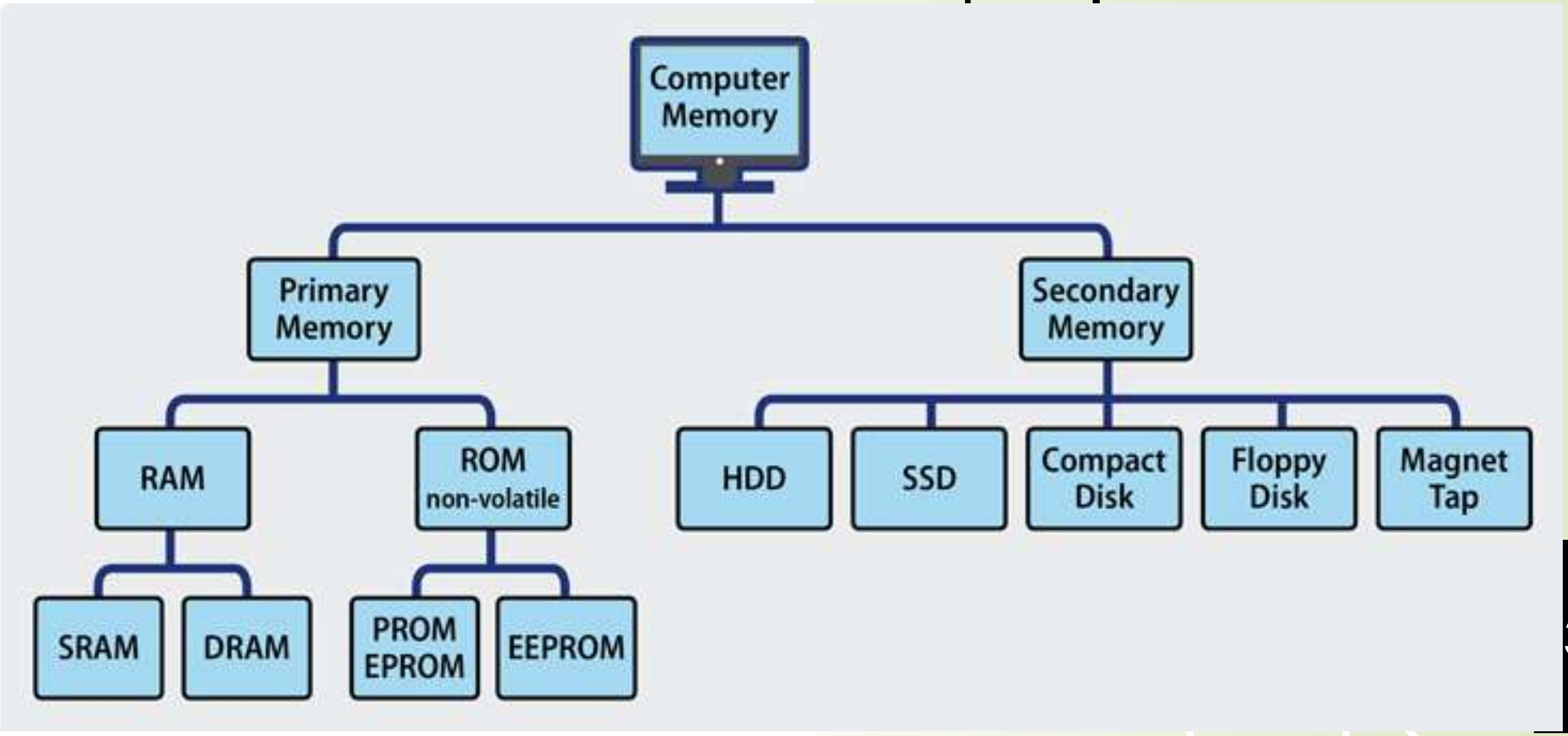
Assembly Language → Assembler → Machine Level Language

Assembler and compiler are used to convert the instruction mnemonics to equivalent binary form for μP redability.

- ✓ Every instruction has a **opcode** (***instruction machine code or instruction code***).
- ✓ After fetching of instruction processor will look at the binary number (opcode) to decode it to understand what is the instruction and what operation has to be performed.
- ✓ **Ex**: 8051 μC has 256 instructions; thus has 256 opcodes.
  - ✓ What would be the size of decoder to decode 256 instruction?

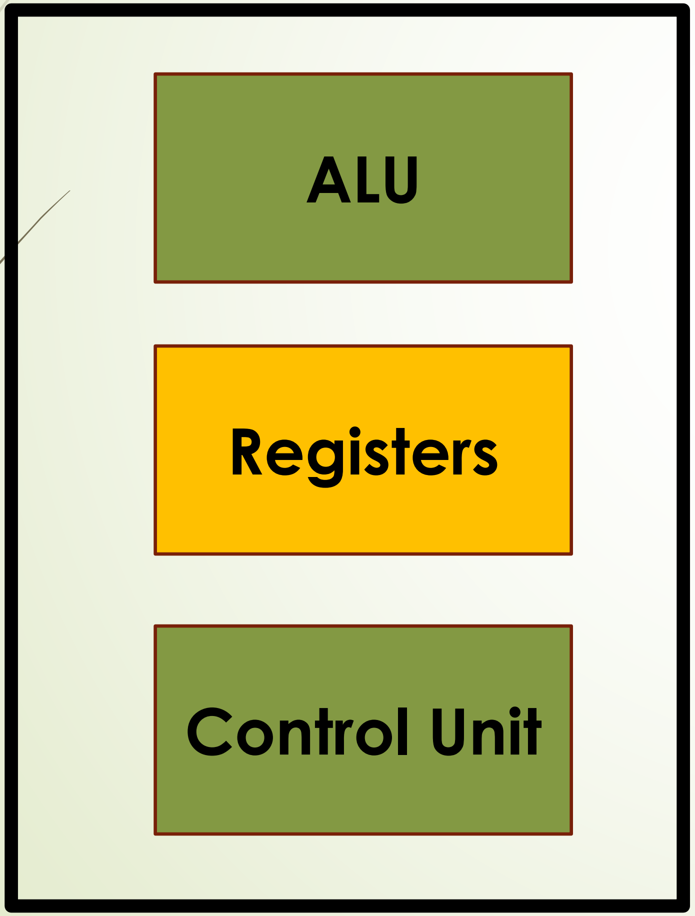


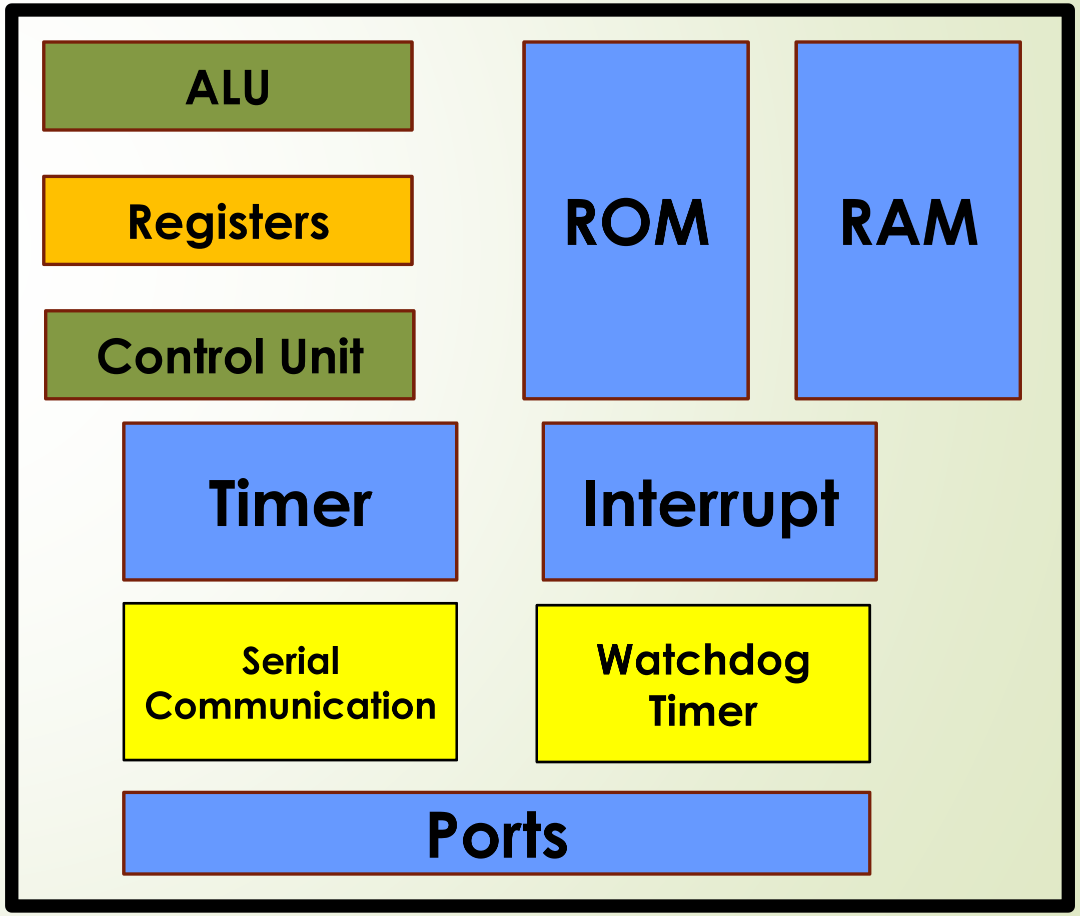
Memory



- **Primary memory** includes ROM and RAM, and is located close to the CPU on the computer motherboard, enabling the CPU to read data from primary memory very quickly indeed. It is used to store data that the CPU needs imminently so that it does not have to wait for it to be delivered.
- **Secondary memory** by contrast, is usually physically located within a separate storage device, such as a hard disk drive or solid state drive (SSD), which is connected to the computer system either directly or over a network. The cost per gigabyte of secondary memory is much lower, but the read and write speeds are significantly slower.


Microprocessor
VS
Microcontoller


ALU
Registers
Control Unit


ALU
Registers
Control Unit
ROM
RAM
Timer
Interrupt
Serial Communication
Watchdog Timer
Ports

Microprocessor Development Tools

Software Tools:

- Assembler
- Linker
- Loader
- Compiler
- Libraries
- Pre-compiler
- Simulator
- IDE - integrated development environment

Development Tools

Hardware Tools:

- In Circuit Emulator (ICE)
- Logic Analyzer
- Microprocessor Development Systems

Assembler

- Translates the assembly language mnemonics for instructions to corresponding binary codes.
- Reads the source file of your program from the disk, then determines the displacement of named data items, the offset of labels, etc. and puts this information in a symbol table.
- Produces the binary code for each instruction and inserts the offsets, etc. that it calculated during the first pass.
- The assembler generates 2 files on the floppy disk or hard disk. The first file is called object file (.obj).
- The second file generated by assembler is called the assembler list file and is given extension (.lst).

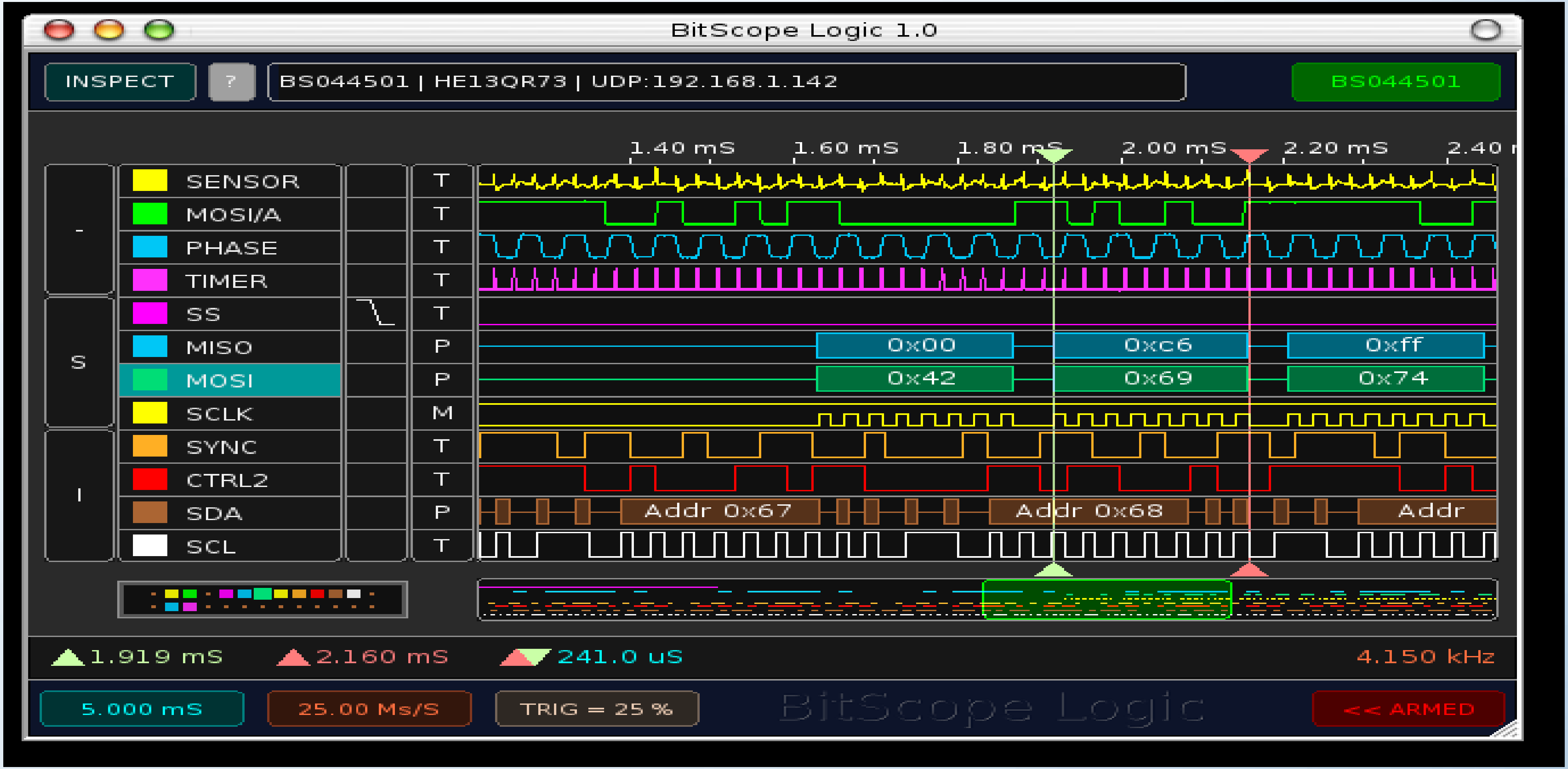
Linker

- ✓ A linker is a program used to join several object files into one large object file.
- ✓ The linker produces a link file which contains the binary codes for all the combined modules.
- ✓ linker also produces a link map file which contains the address information about the linked files (.exe).

Logic Analyzer

- ✓ Developed out of the need to be able de-bug and undertake fault-finding on microprocessor based systems, need for simultaneous monitoring of many lines and test points.
- ✓ Widely used to develop and debug complex digital electronic logic circuits
- ✓ Displays traces of multiple logic channels and reveal the circuit operation.
- ✓ Displays relative timing of a large number of signals, enabling traces of logic signals to be seen in such a way that the operation of several lines in a digital circuit can be monitored and investigated.
- ✓ Have much greater levels of flexibility and processing power when connected with computer.

Logic Analyzer



Logic Analyzer Characteristics

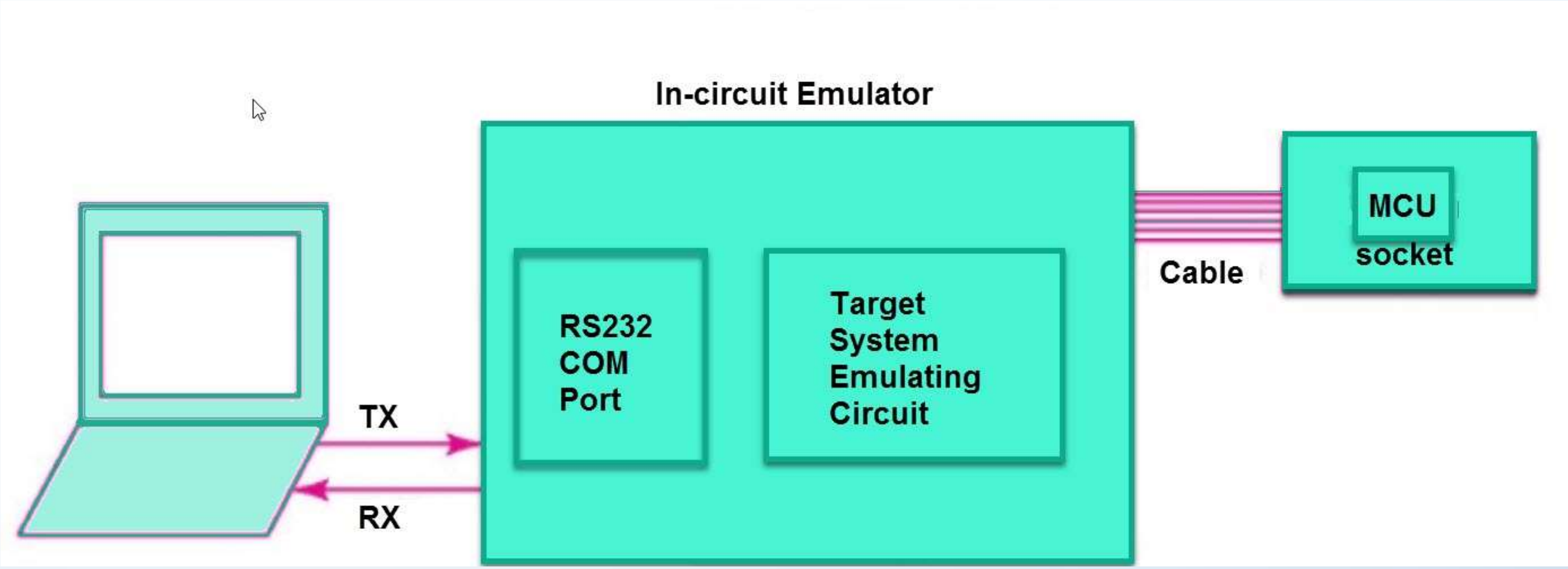
- Multiple channels
  - ❖ About 32 and 200+ channels, each channel monitoring one digital line
- Provide a time display of logic states
  - ❖ Horizontal time axis and a vertical axis to indicate a logic high or low states
- Display logic states
  - ❖ Signals entering various channels are converted into a high or low state for further processing within the analyzer. It provides a logic timing diagram of the various lines being monitored.
- Does NOT display analogue information
  - ❖ Monitors the logic operation of the system.

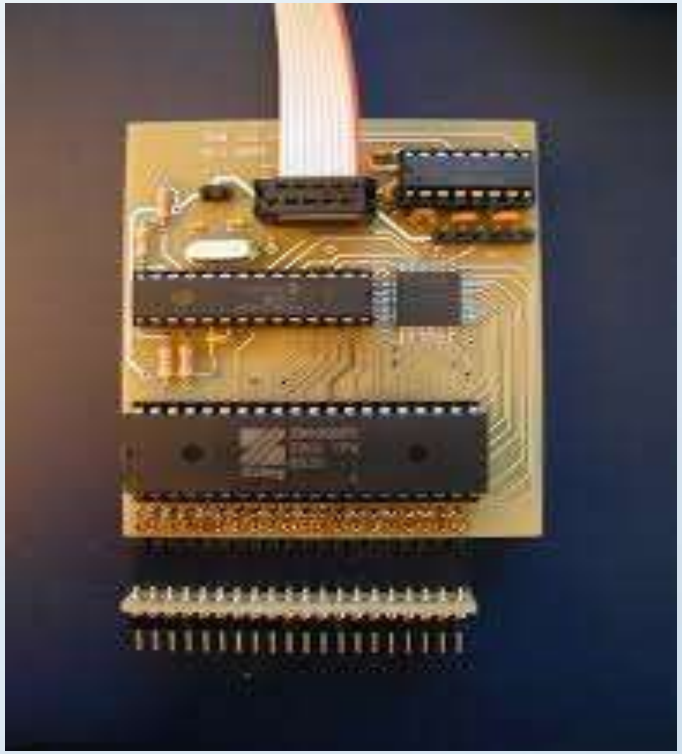
Logic Analyzer Applications

- Verifying and debugging the operation of digital designs looking a logic states and timings.
- Correlate a large number of digital signals
- Investigate the system operation.
- Detect timing violations
- Trace embedded software operation.

In Circuit Emulation (ICE)

- **ICE** is the use of a hardware device or **in-circuit emulator** used to debug the software of an embedded system.
- It operates by using a processor with the additional ability to support debugging operations, as well as to carry out the main function of the system.
- Particularly for older systems, with limited processors, this usually involved replacing the processor temporarily with a hardware emulator: a more powerful although more expensive version.
- It was historically in the form of bond-out processor which has many internal signals brought out for the purpose of debugging. These signals provide information about the state of the processor.


In-circuit Emulator
RS232
COM
Port
Target
System
Emulating
Circuit
TX
RX
Cable
MCU
socket



Thank You
For Your Attention

End of Module - 01