Embedded System Design and Advanced Digital Systems Design

SONEIL TECH

Contact no.9501476514

EMBEDDED SYSTEM

- Embedded System Design
  - Microcontrollers
  - Software Development for Microcontrollers
- Advanced Digital System Design
  - Finite State Machine (FSM)
  - Hardware Description Languages

What is an Embedded System?

- Computing systems embedded within electronic devices
- Hard to define
  - Nearly any computing system other than a desktop computer
  - "A computer that is a component in a larger system, and is not visible as a computer to a user of that system."
  - "A programmable component of subsystem providing some intelligence functions to the system of which it is a part."

What is an Embedded System?

- A microcontroller based system.
- Built into a device to control a function or a range of functions.
- Not designed to be programmed by the end-user (like a PC).
- Executes an in-built single program repeatedly.
- Tight coupling between hardware and software.
- Designed to work in highly constrained environments.
- Low cost, low power, small, fast, etc.
- Reactive and real-time
- Continually reacts to changes in the system's environment
- Must compute certain results in real-time without delay

Embedded Systems

- Key points:
  - Embedded systems have rigidly defined operational bounds
  - Not general purpose computers (PC, Unix workstation)
- Billions of units produced yearly vs millions of desktop units
- Perhaps 50 per household and per automobile

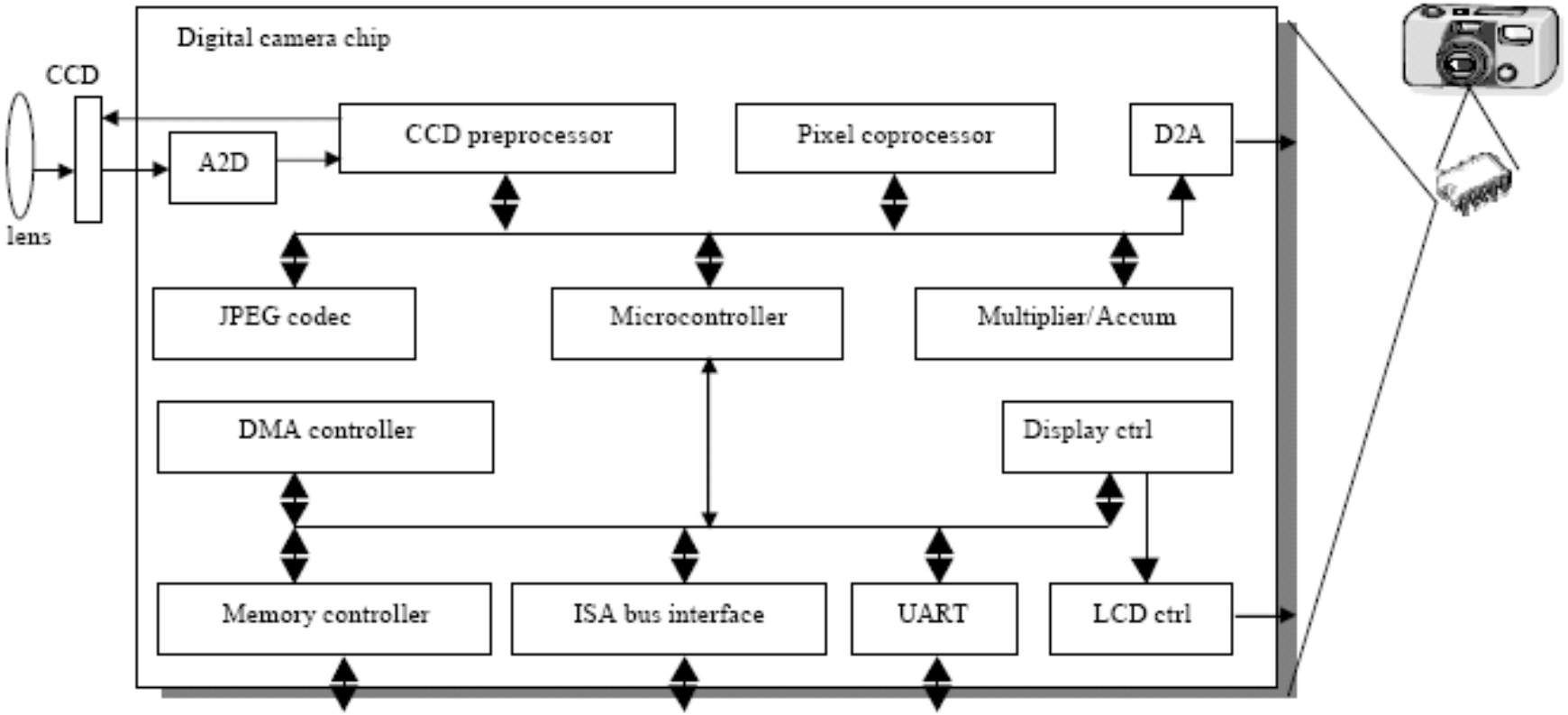
Embedded Systems

- Automotive systems
- Airplanes
- Toys
- Medical Devices

Example Embedded Systems

Anti-lock brakes
Auto-focus cameras
Automatic teller machines
Automatic toll systems
Automatic transmission
Avionic systems
Battery chargers
Camcorders
Cell phones
Cell-phone base stations
Cordless phones
Cruise control
Curbside check-in systems
Digital cameras
Disk drives
Electronic card readers
Electronic instruments
Electronic toys/games
Factory control
Fax machines
Fingerprint identifiers
Home security systems
Life-support systems
Medical testing systems
Modems
MPEG decoders
Network cards
Network switches/routers
On-board navigation
Pagers
Photocopiers
Point-of-sale systems
Portable video games
Printers
Satellite phones
Scanners
Smart ovens/dishwashers
Speech recognizers
Stereo systems
Teleconferencing systems
Televisions
Temperature controllers
Theft tracking systems
TV set-top boxes
VCR's, DVD players
Video game consoles
Video phones
Washers and dryers

ES Example - Digital Camera



Embedded System Design

- Design goal:
  - Construct an implementation with desired functionality
- Key design challenge:
  - Simultaneously optimize numerous design metrics
- Design metric:
  - A measurable feature of a system's implementation
  - Optimizing design metrics is a key challenge

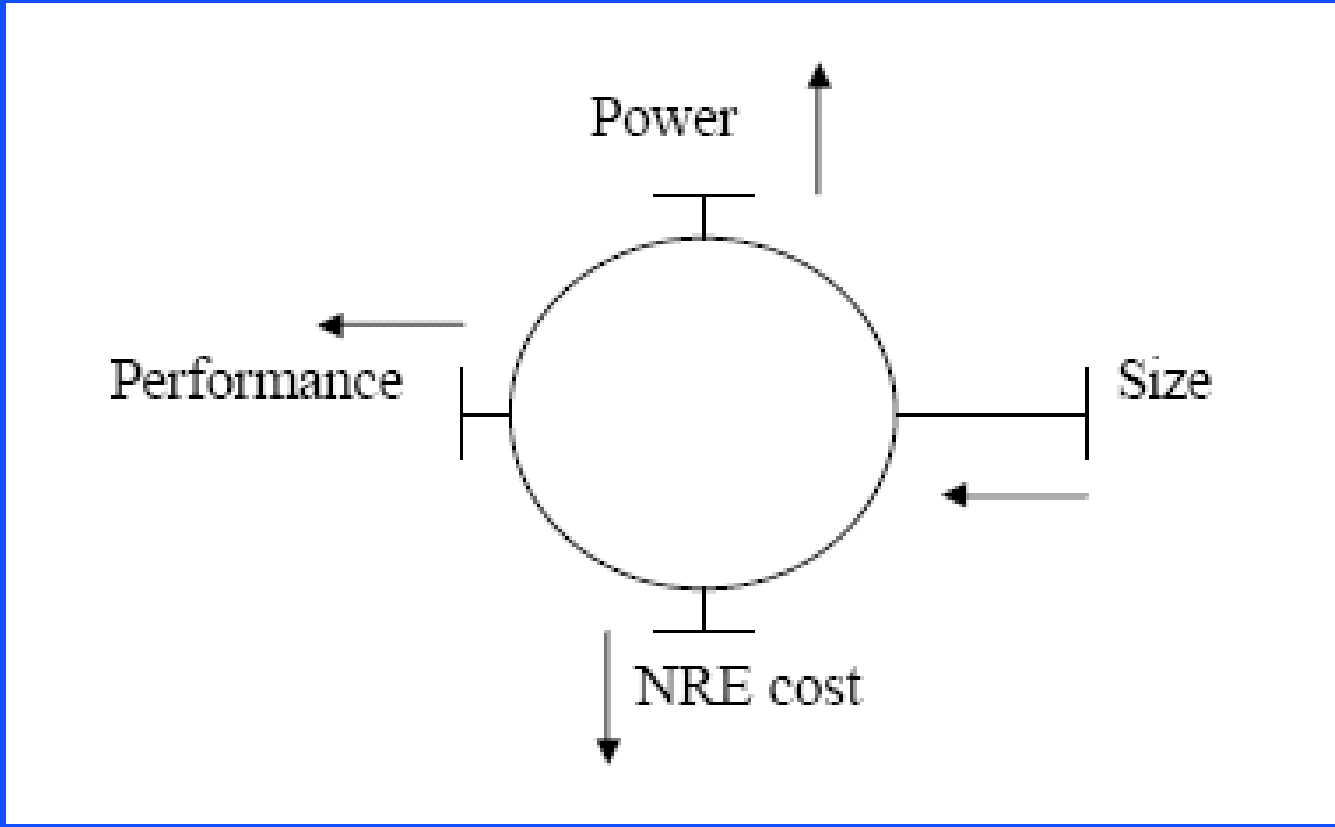
Common Design Metrics

- Unit cost: the monetary cost of manufacturing each copy of the system, excluding NRE cost
- NRE cost (Non-Recurring Engineering cost): The one-time monetary cost of designing the system
- Size: the physical space required by the system
- Performance: the execution time or throughput of the system
- Power: the amount of power consumed by the system
- Flexibility: the ability to change the functionality of the system without incurring heavy NRE cost

Common Design Metrics

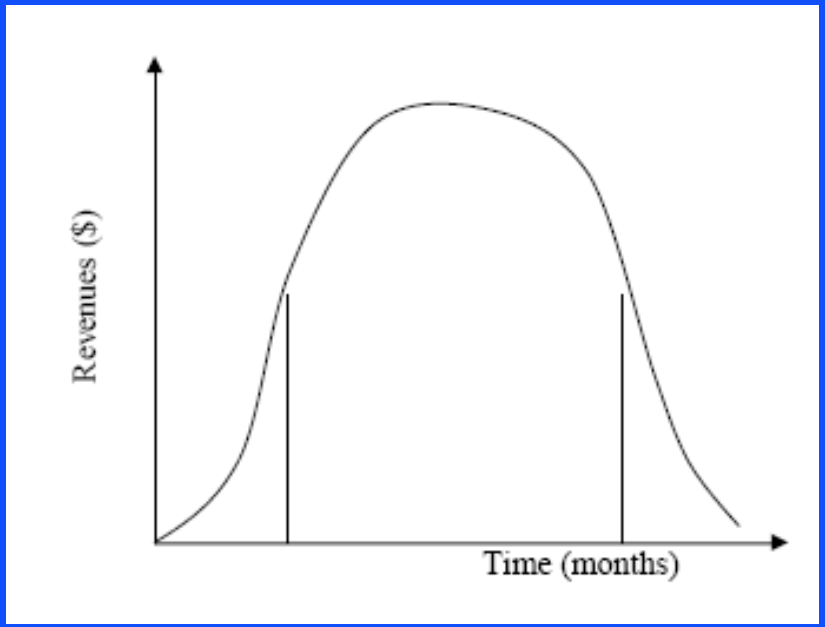
- Time-to-prototype: the time needed to build a working version of the system
- Time-to-market: the time required to develop a system to the point that it can be released and sold to customers
- Maintainability: the ability to modify the system after its initial release
- Correctness, safety and many more

Key Design Metrics



Time to Market

- Time required to develop a product to the point it can be sold to customers
- Market window - Period during which the product would have highest sales
- Delays can be costly



NRE and Unit Cost Metrics

- Costs:
  - NRE cost (Non-Recurring Engineering cost): The one-time monetary
    - Cost of designing the system
  - Unit cost: the monetary cost of manufacturing each copy of the system, excluding NRE cost
- Total cost = NRE cost + unit cost * # of units
- Per-product cost = Total cost / # of units

  = (NRE cost / # of units) + unit cost

Exercise

- NRE=$2000, unit=$100
- 10 units were sold

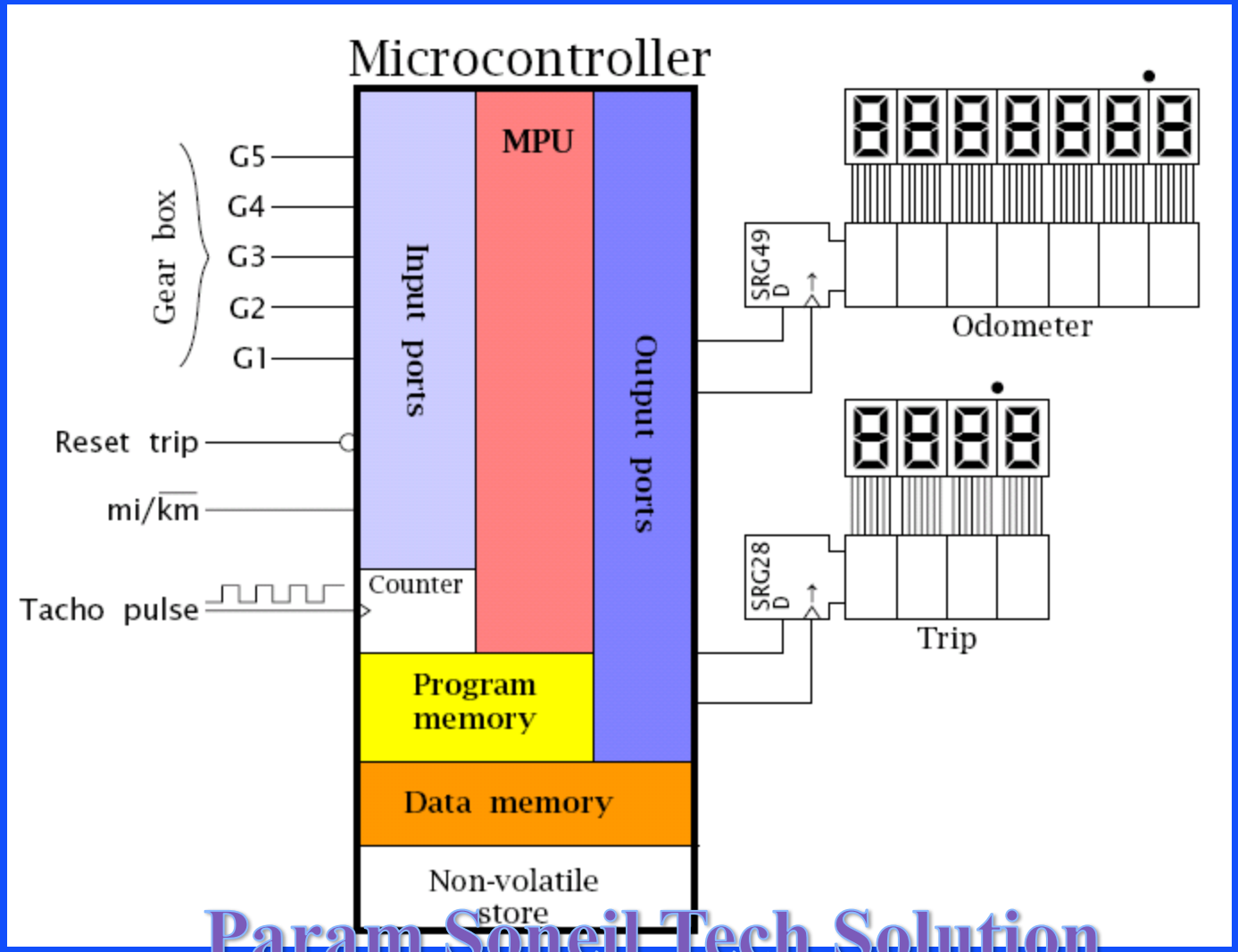
Exercise

- NRE=$2000, unit=$100
- For 10 units:
  - Total cost = $2000 + 10*$100 = $3000
  - Per-product cost = $2000/10 + $100 = $300
- *Amortizing NRE cost over the units results in an additional $200 per unit*

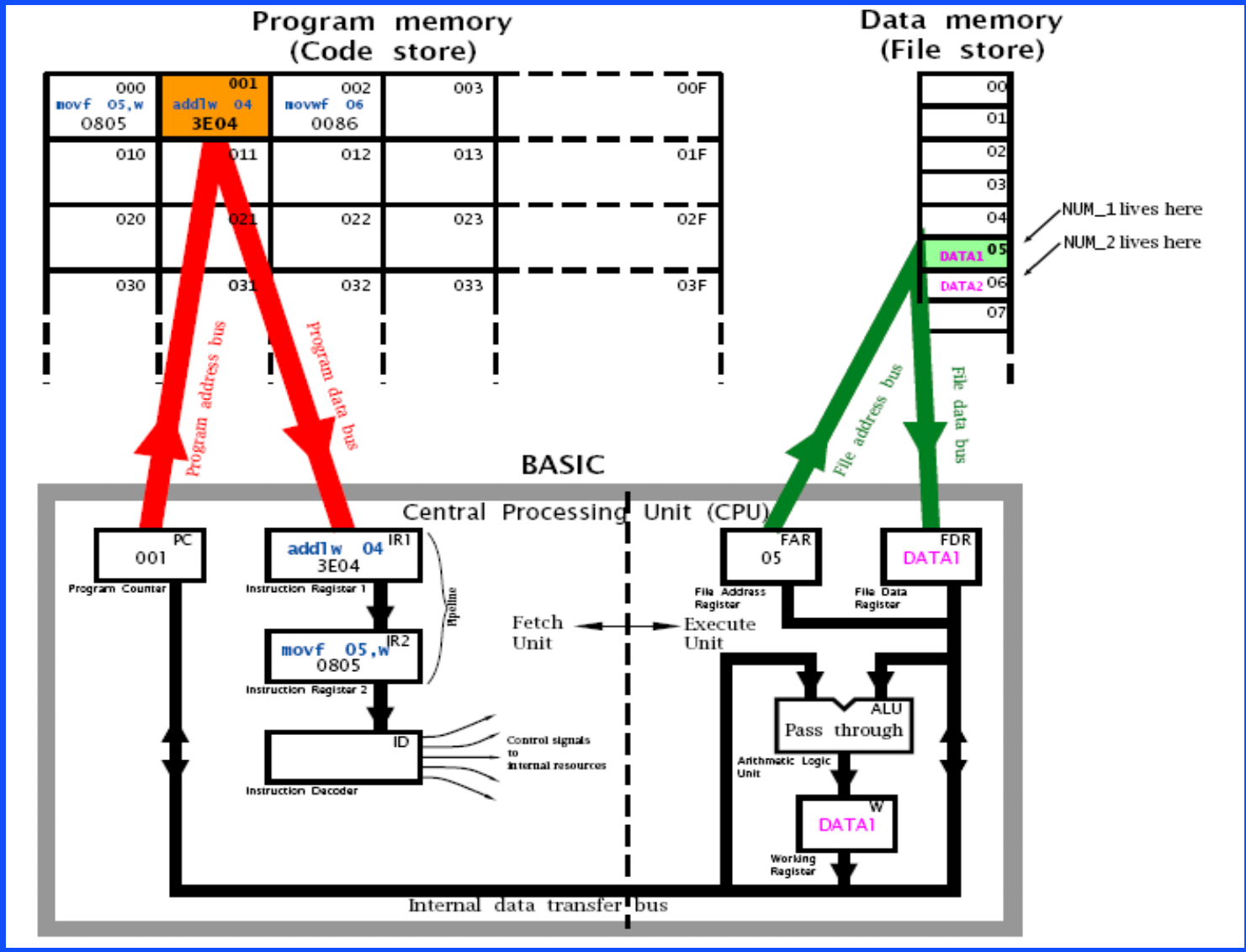
What is a Microcontroller?

- Microcontroller contains a microprocessor with additional peripheral devices integrated into a single package
- Peripheral devices may include:
  - Serial ports (COM), Parallel (Ports), Ethernet ports, A/D & D/A
  - Interval timers, watchdog timers, event counter/timers, real time clock
  - Other local processors (DSP, numeric coprocessor, peripheral controller)

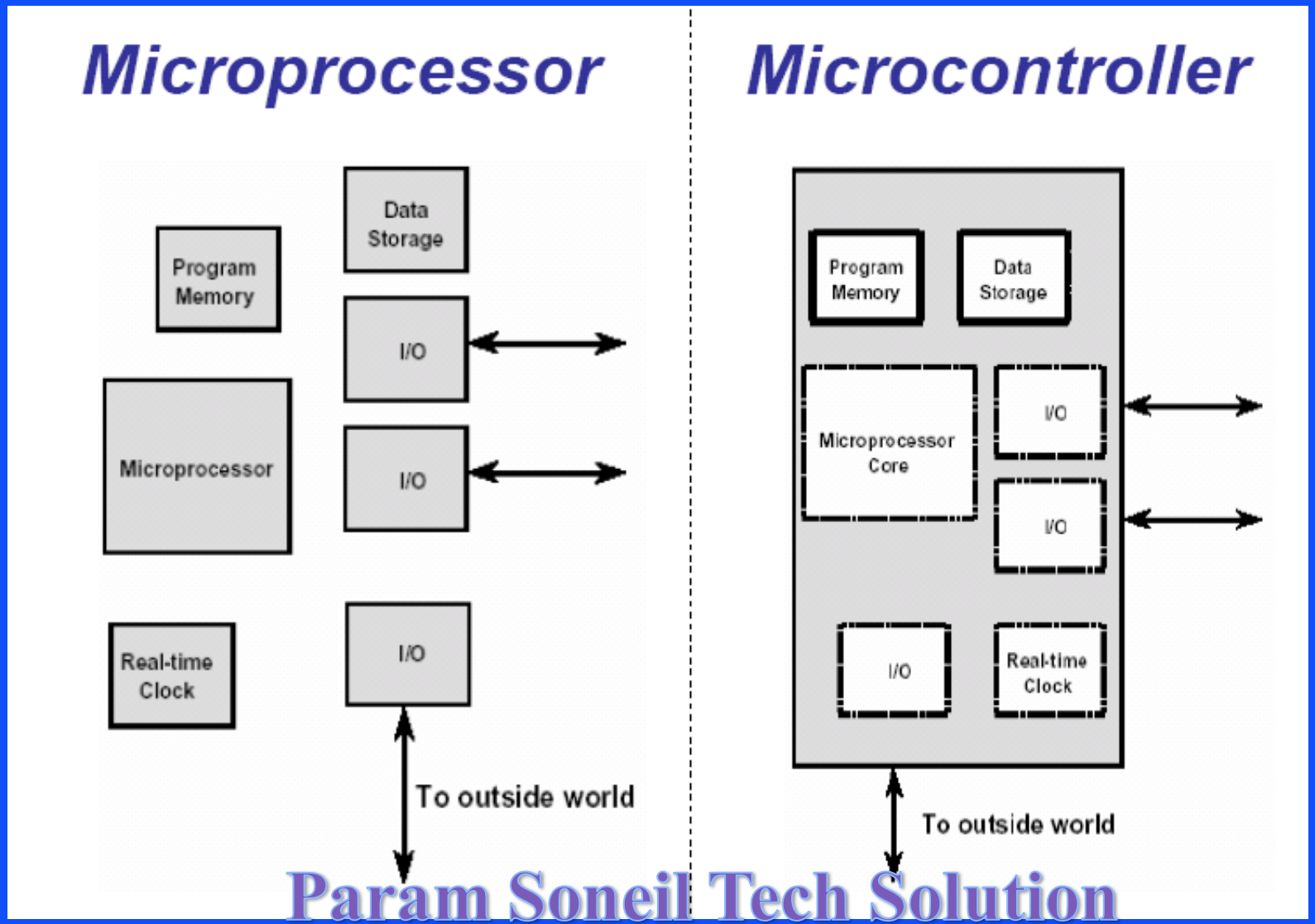
Microcontroller Example



Instruction Execution



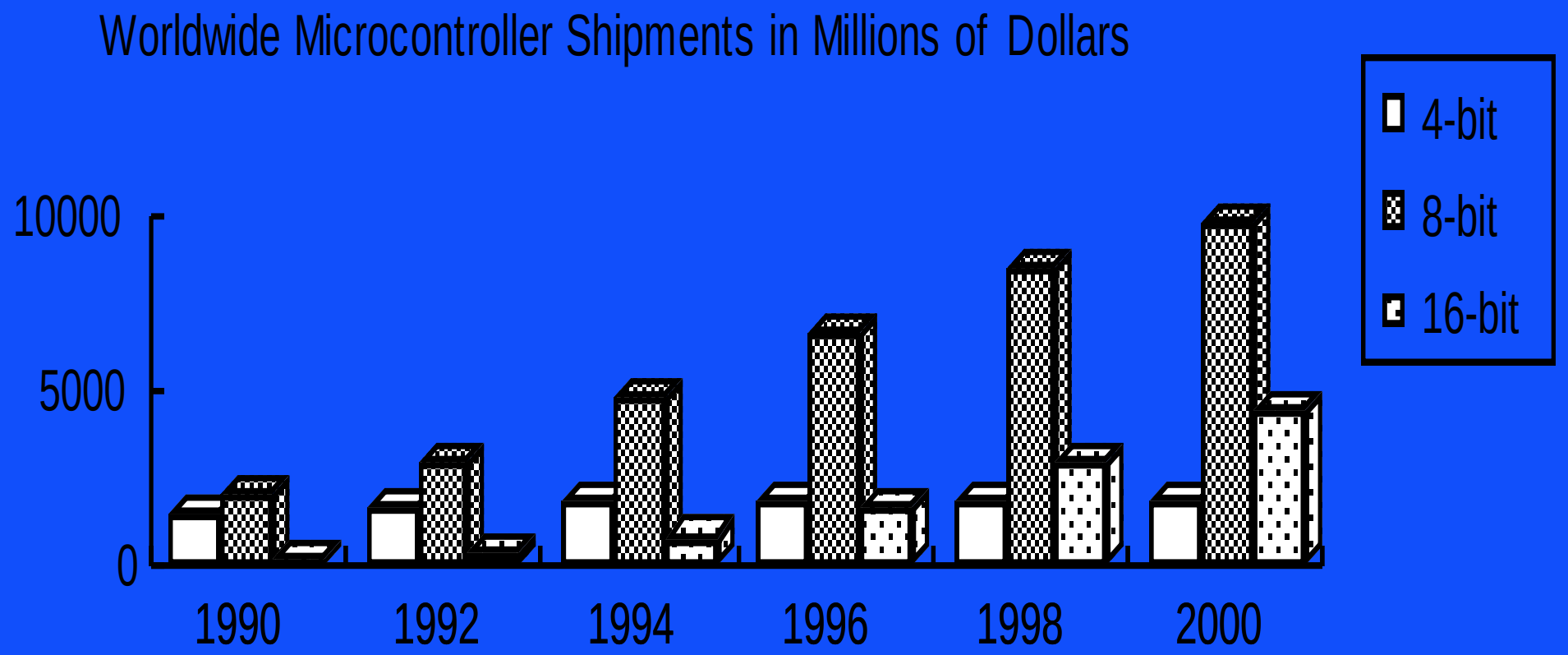
Microprocessor vs. Microcontroller



Microcontroller Market

- Shipments- > 16 Billion in 2000, 8 bit > 1/2 market
- Major Players:
  - Microchip 16Fxx
  - Intel 8051
  - Motorola MC68HCxx
  - National COP800
  - SGS/Thomson ST62
  - Zilog Z86Cxx

Shipments


Worldwide Microcontroller Shipments in Millions of Dollars
4-bit
8-bit
16-bit
10000
5000
0
1990
1992
1994
1996
1998
2000

Programming Languages

- ASM
  - Low level
  - Full Control
- BASIC, Forth, LOGO
  - Interpreted
  - Easy to use
  - Slow
- C
  - Most widely used
  - HiTech C
  - Microchip C
  - CCS PIC C

What you need

- **MPLAB IDE**
  - **Programmer's text editor**
  - **MPLAB SIM**, high speed software simulator for PICmicro and dsPIC MCUs with peripheral simulation, complex stimulus injection and register logging
  - **Full featured debugger**
  - **Graphical project manager**
  - **Visual Device Initializer (VDI)** to set up complex peripherals with a graphical point-and-click method
  - **Version control support** for MS Source Safe, CVS, PVCS, Subversion
  - **MPASM™** macro assembler with MPLINK™ linker and MPLIB™ librarian
  - **MPLAB ASM30 Assembler**, **MPLAB LINK30 and Utilities** for PIC24 and dsPIC devices
  - **PROCMD** command line programmer for MPLAB PM3 and PRO MATE® II
  - **Visual PROCMD** for simplified GUI control of MPLAB PM3 and PRO MATE® II
  - **CCS PCB C Compiler**
  - **Many Powerful Plug-ins:** AN851 Bootloader programmer, AN901 BLDC Motor Control Interface, AN908 ACIM Tuning Interface, KeeLoq, Data Monitor and Control, CMX Scheduler and RTOS viewer

What is the Process ?

1. Write you program in MPLAB IDE
   - C or ASM
2. Compile your program
   - CCS C Compiler
3. Transfer your program
   - Puts HEX file into the PIC
   - Use PICSTART and MPLAB
   - “Burns your app into the PIC”
4. Insert your PIC
5. Power it Up

What is the Process ?

6. Debug your program
    - Never works a the first time
7. Repeat step 1