Microprocessor and Microcontrollers Based Design

EC-310

Why Study Microcontrollers and Microprocessors

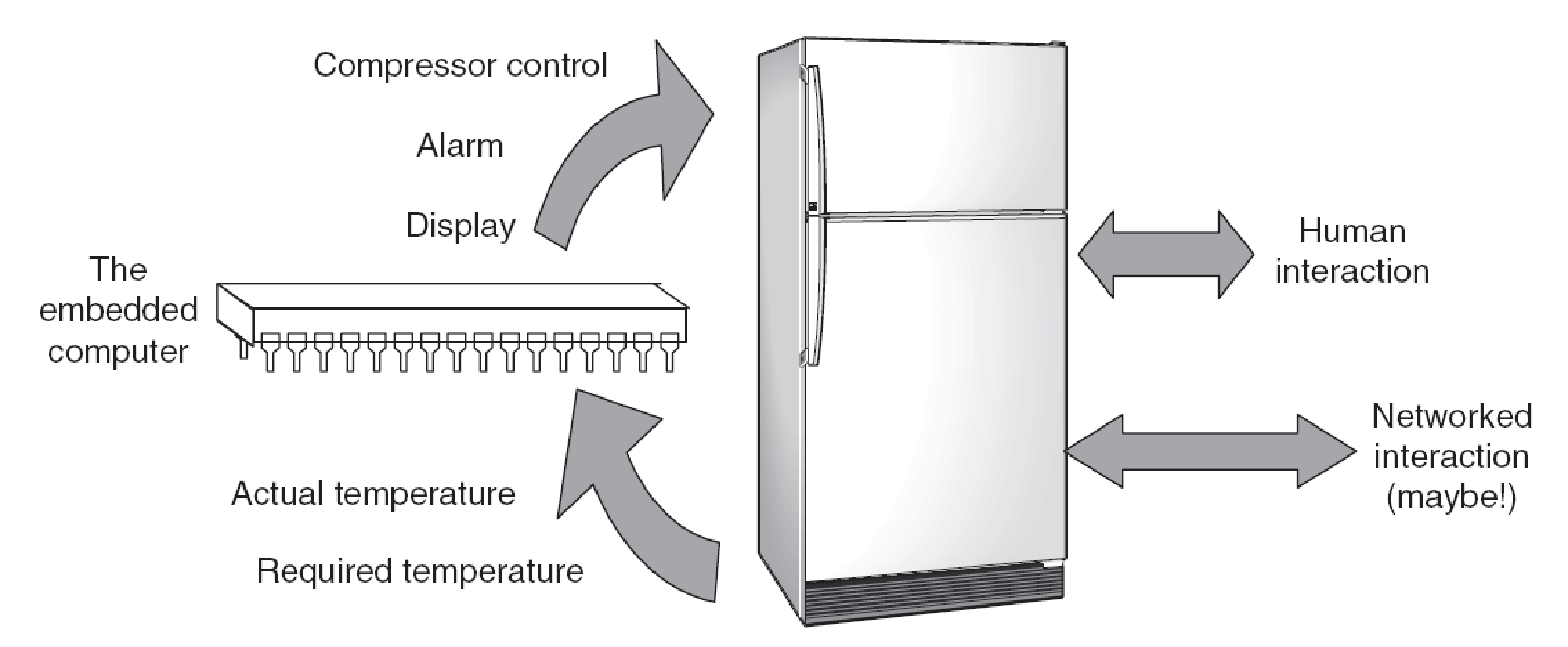
- Microprocessor is the brain of modern computing and communication machines such as PC, Laptop, Notepad, Phones, Servers etc
- Microcontroller is the brain of the modern embedded systems
- IoT is an emerging network with embedded systems being the core of the systems

Definition of Embedded Systems

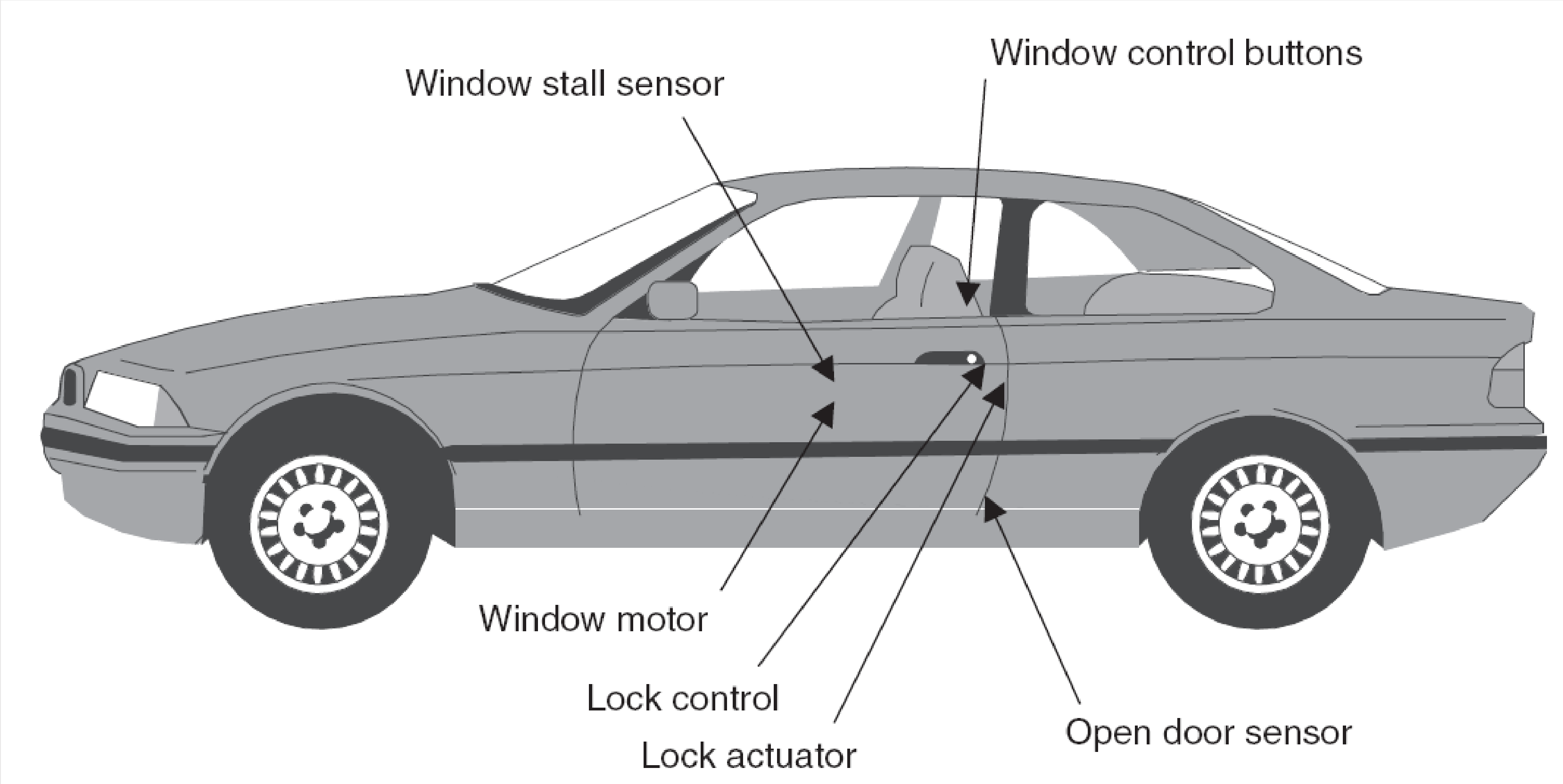
- An **embedded system** is a computer system with a dedicated function within a larger mechanical or electrical system, often with real-time computing constraints.

A Microcontroller and Sensors in a Refrigerator

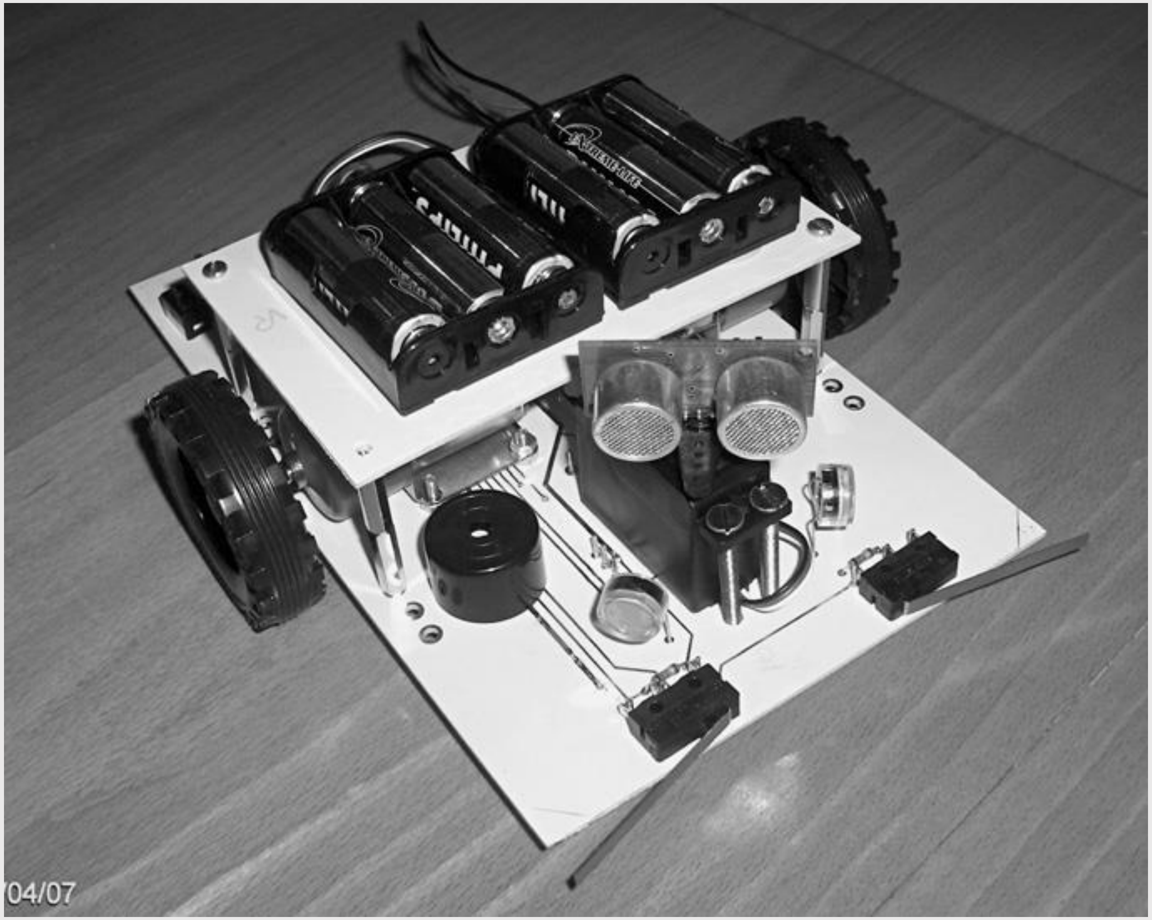
- Refrigerator as a larger electrical system



A Microcontroller and Actuator in a Car Door



Navigation System in an Autonomous Guided Vehicle



What is IoT?

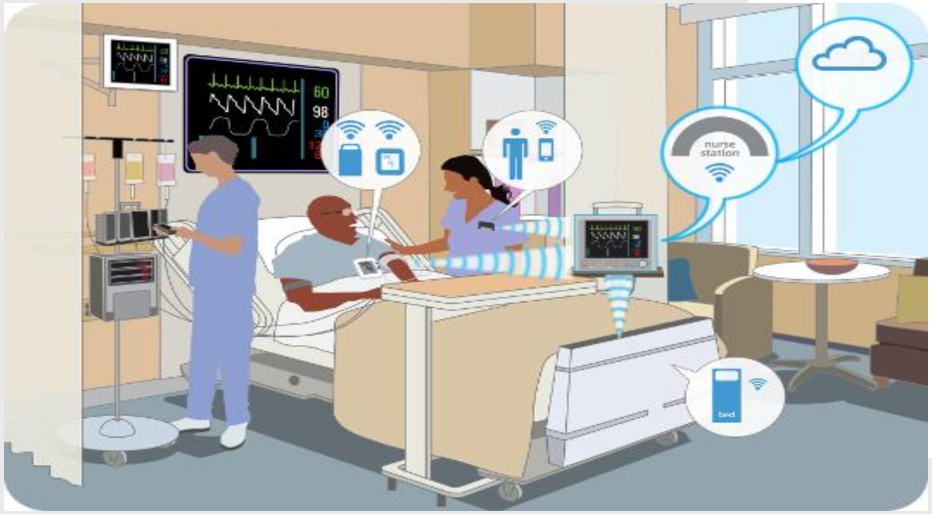
- The **Internet of Things (IoT)** is the network of physical objects—devices, vehicles, buildings and other items embedded with electronics, software, sensors, and network connectivity—that enables these objects to collect and exchange data.

IoT: An Example

Pg-Out

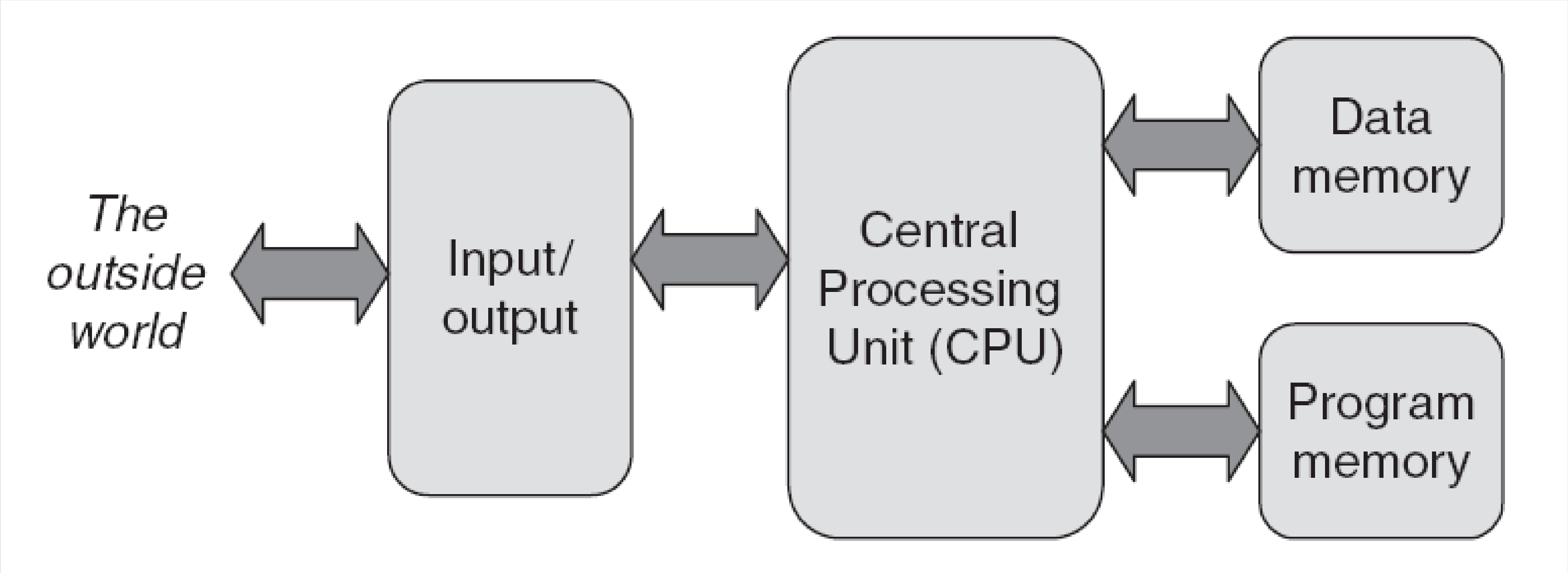
Smart Appliances

Wearable
Tech



Healthcare

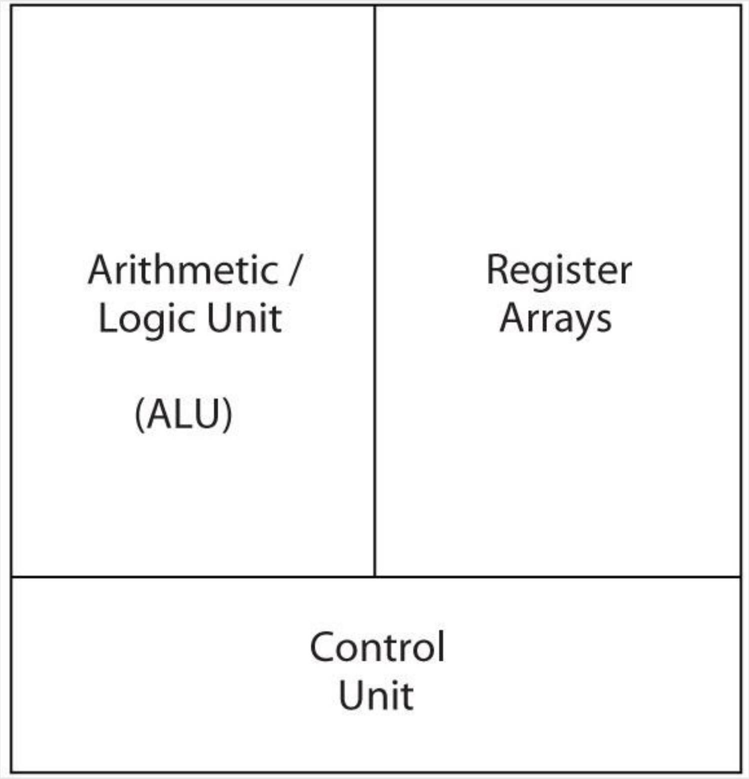
Computer Essentials



Microprocessor Unit

- Clock driven registered based IC
- Accepts binary data as input
- Process it according to the instruction
- Provide the respective results as output

Microprocessor Unit (MPU)



- MPU (CPU)
  - ALU,
  - Control Unit,
  - Registers

Example: General Purpose Processor (GPP)

- A CPU with Very High Speeds from few Megahertz to 4 Gegahertz
- Computationaly expensive tasks
- Running the PCs
- Costly
- Example Include: Pentium Series, Itium Series

Microcontrollers

- Microcontroller (MCU)
  - Integrated electronic computing device that includes three major components on a single chip
    - Microprocessor (MPU)
    - Memory
    - I/O (Input/Output) ports
  - Low to moderate clock speeds few kilohertz to few Megahertz

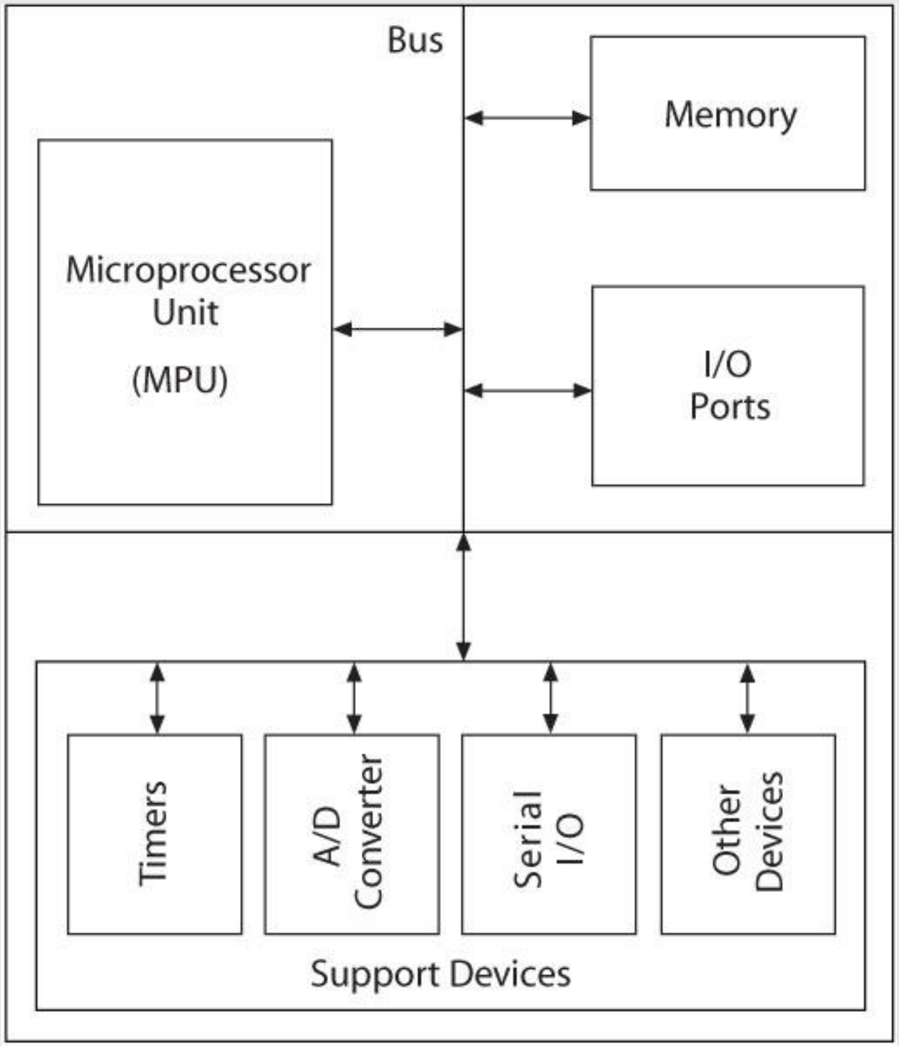
Microcontrollers

- Support Devices
  - Timers
  - A/D converter
  - Serial I/O
- Common communication lines
  - System Bus

If GPP Exists Why Microcontroller?

- Cost effective
- Already added additional circutry
- Easy to program
- Easy to communicate
- Less time to implement
- Easily available
- For simple tasks

Block Diagram of a Microcontroller



Popular Microcontrollers Manufacturers

- STMicroelectronics (STM microcontrollers)
  - Mostly use ARM Cortex Processor IP Core
- Renesas Electronics (RX microcontrollers)
  - Mostly use RXv1/RXv2 from Renesas
- Micro Chip (PIC)
  - Mostly use MIPS and also ARM Cortex IP core

Some Popular Microprocessor Manufacturers

- Intel → Pentium, Celeron, Itanium, Quark, etc
- AMD→ athlon, **Am29000**
- NXP → Coldfire

- Arm Holdings and Qualcom are the largest processor Cores Architecture, giving IP cores through licenses, e.g. Arm Cortex-M core

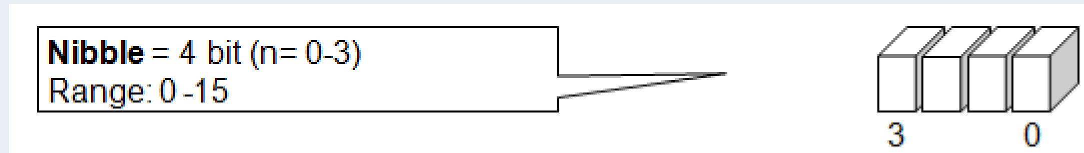
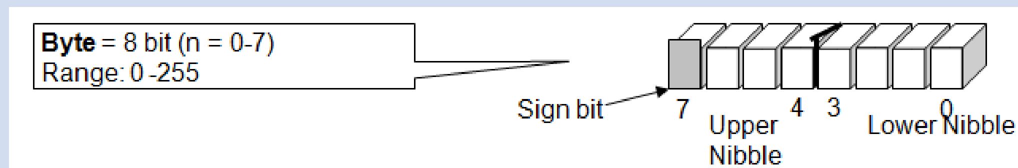
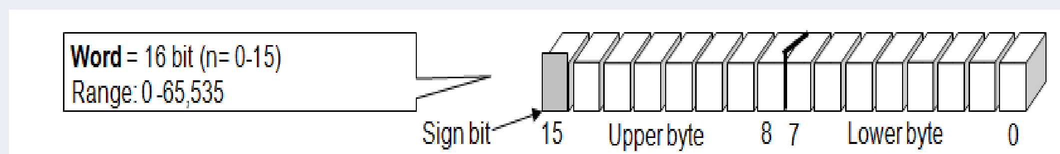
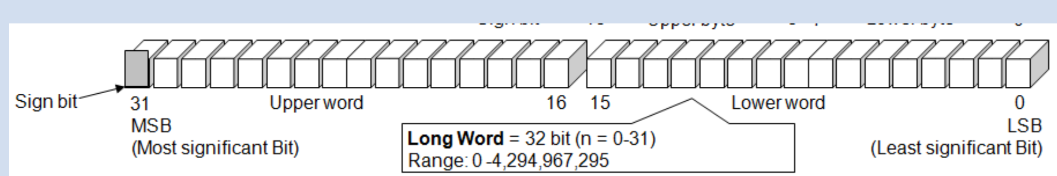
Word Size of a Processor

- The number of bits a processor can process at a single time
- E.g. An 8 bit processor can only process 8 bits at one time. Similarly a 16 bit or 32 bit processor can process 16 or 32 bits at one time.
- E.g. An 8 bit processor cannot add two numbers greater than 2^8-1 in one go
- It must split into small parts and add them in

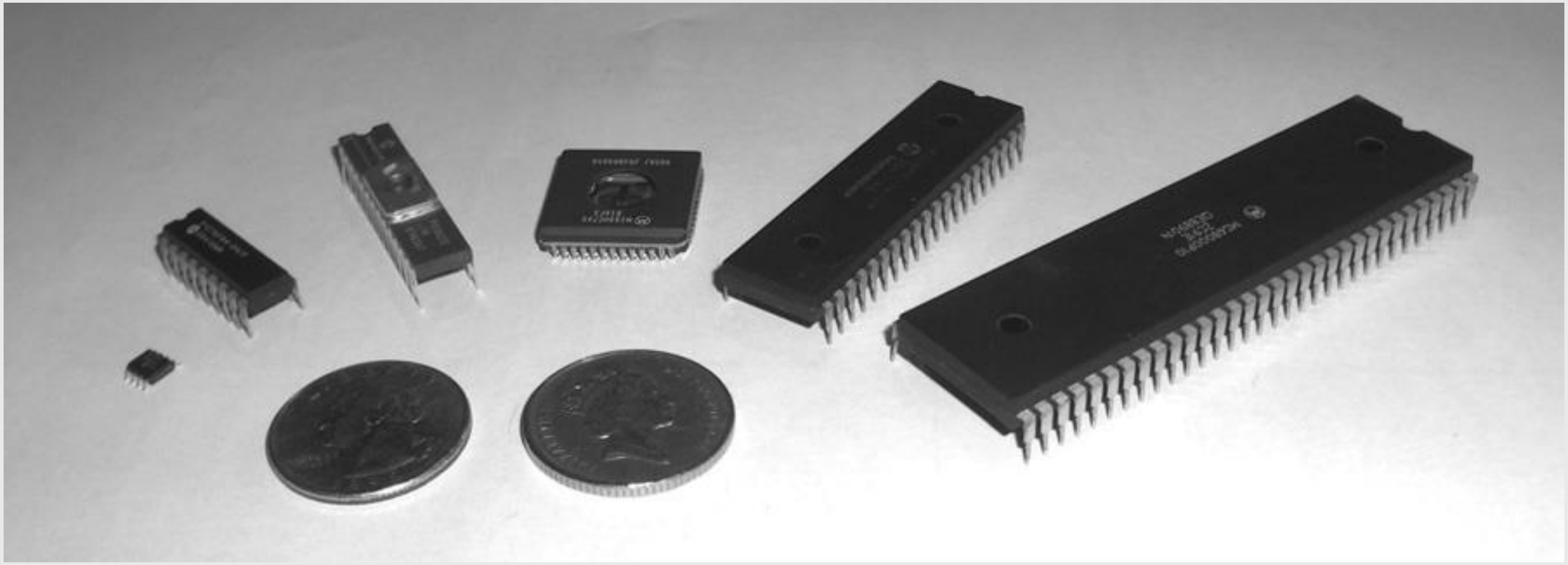
  Series

DATA SIZE

Pg-Out

<table>
<tr><td>Nibble</td><td>4 bit</td><td>
</td></tr>
<tr><td>Byte</td><td>8 bit</td><td>
</td></tr>
<tr><td>Word</td><td>16 bit</td><td>
</td></tr>
<tr><td>Long word</td><td>32 bit</td><td>
</td></tr>
</table>

Microcontroller Packaging and Appearance

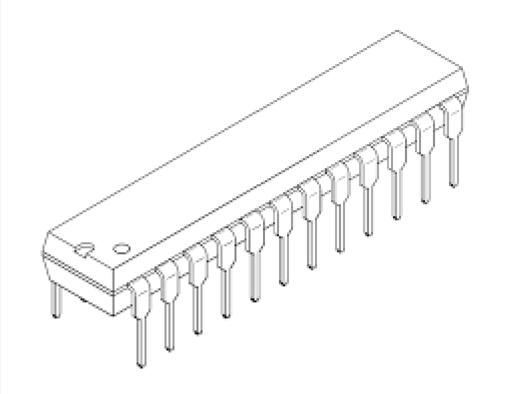


From left to right: PIC 12F508, PIC 16F84A, PIC 16C72, Motorola 68HC05B16, PIC 16F877, Motorola 68000

Packaging of a Microprocessor or Microcontroller

- The body/box of the IC containing the electronic circuitry is called the packaging
- Two microcontrollers from the same company with the same functionalities may have different packaging

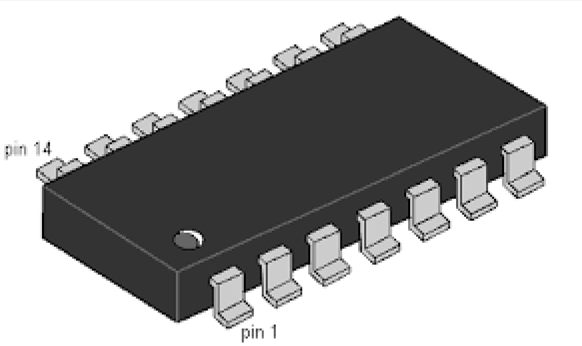
Packaging Basic Classes based On Connection with the Board



1. Through Holes Mounting (THM)

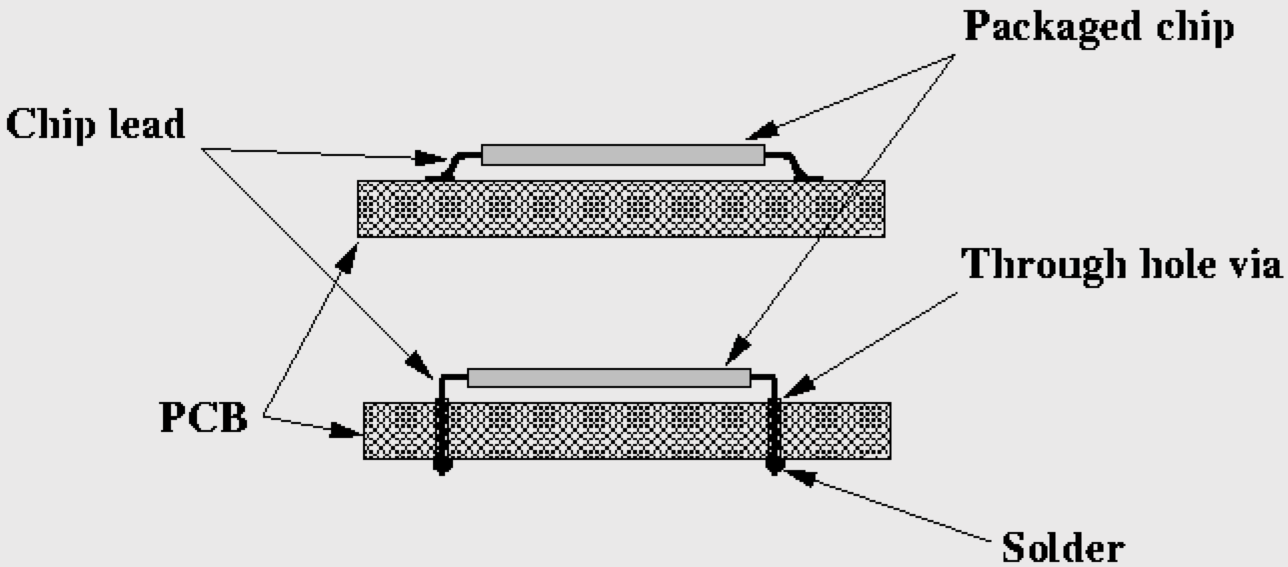
- Holes drilled and plated with copper
- Soldering
  - Chips placed inside holes
  - Bottom of board passed through a molten solder

2. Surface Mount Technology (SMT)

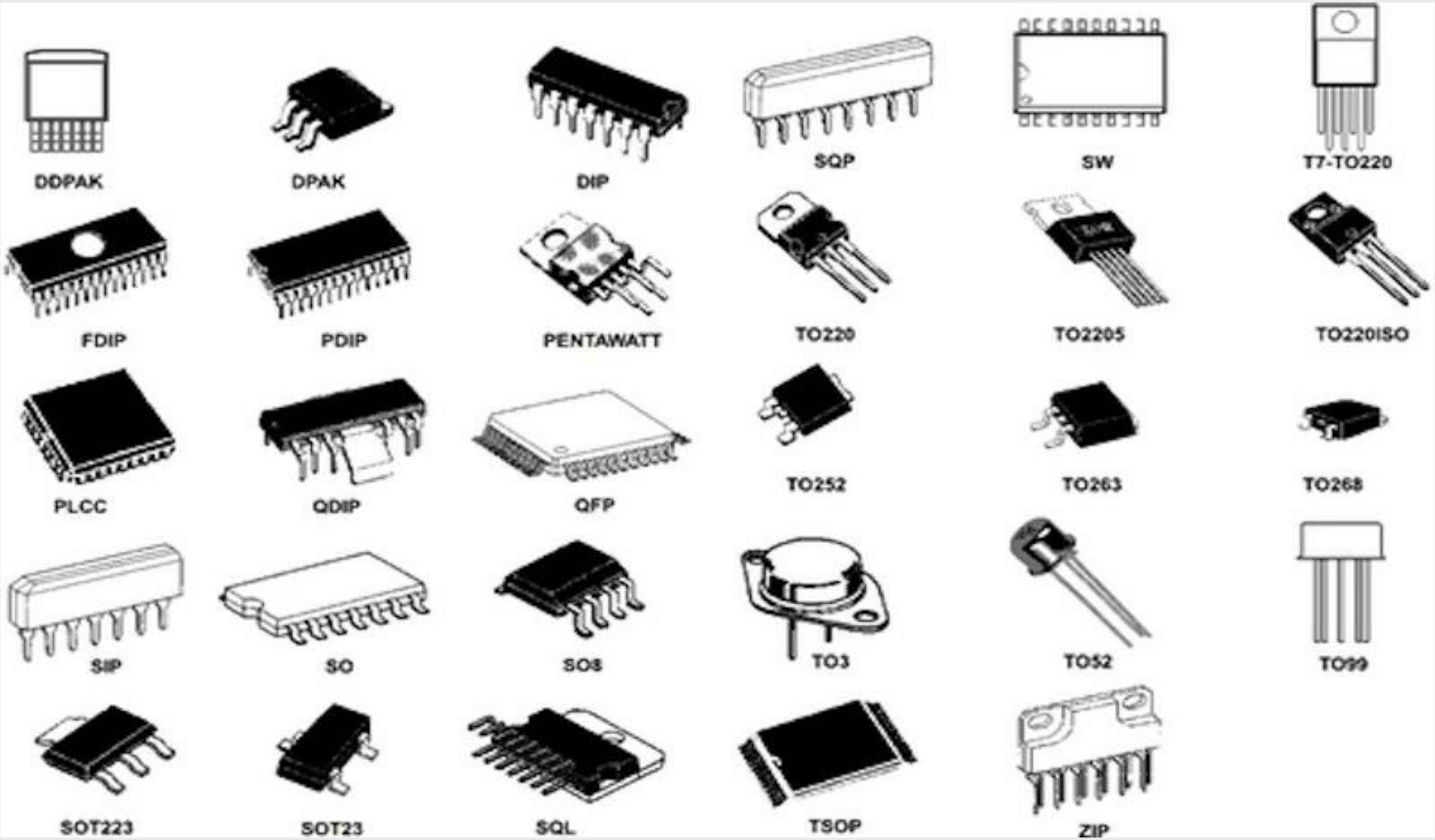


- More wiring room inside PC board
- Reduced space between package leads
- Chips on both sides of board
- Soldering
  - – Solder paste applied
  - – Heat supplied by intense infrared light, heated air,…

SMT and THM



Package Types Based on Structure



Pg-Out

Package Types Based on Structure

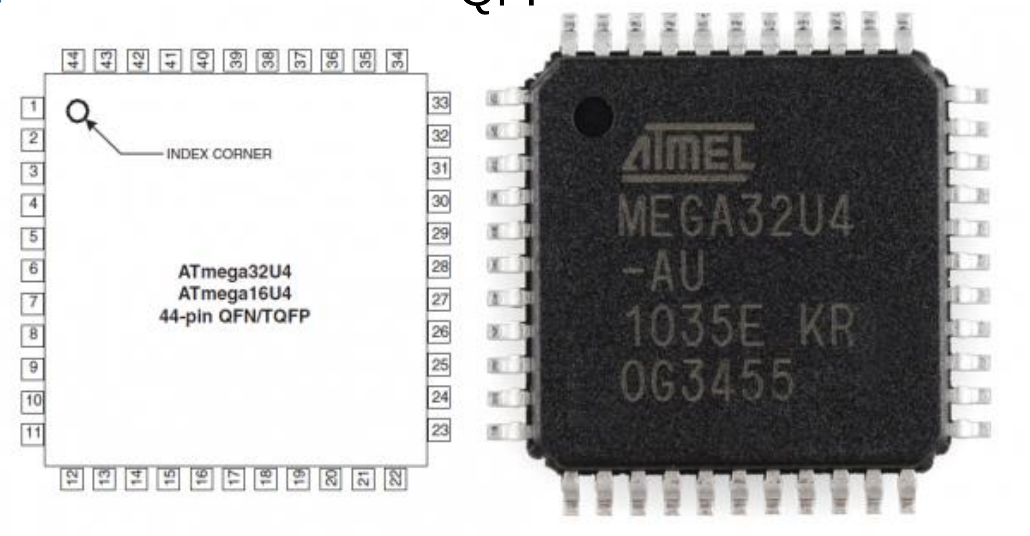
Fig. 3 :Surface Mount Packages

ICs Pin Numbering

- Dot and Notch
- Pin number starts from the pin with the dot in the anti-clockwise direction

DIP

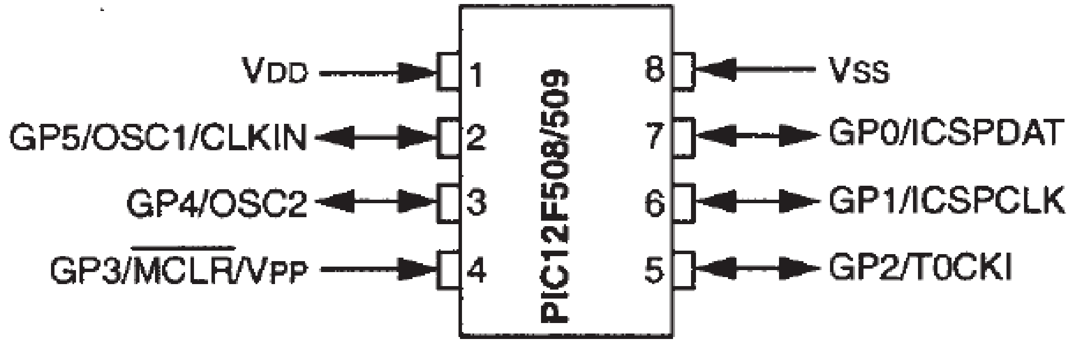
QFP



PIC Microcontrollers

- Peripheral Interface Controller (PIC) was originally designed by General Instruments
- In the late 1970s, GI introduced PIC® 1650 and 1655 – RISC with 30 instructions.
- PIC was sold to Microchip
- Features: low-cost, self-contained, 8-bit, Harvard structure, pipelined, RISC, single accumulator, with fixed reset and interrupt vectors.

PIC 12F508/509 pin connection diagram



Key

V_{DD}:	Power supply	V_{SS}:	Ground
V_{PP}:	Programming voltage input	MCLR:	Master clear
OSC1, OSC2:	Oscillator pins	CLKIN:	External clock input
GP0 to GP5:	General-Purpose input/output pins (bidirectional except GP3)		
CSPDAT:	In-Circuit Serial Programming™ data pin.		
CSPCLK:	In-Circuit Serial Programming™ clock pin.		

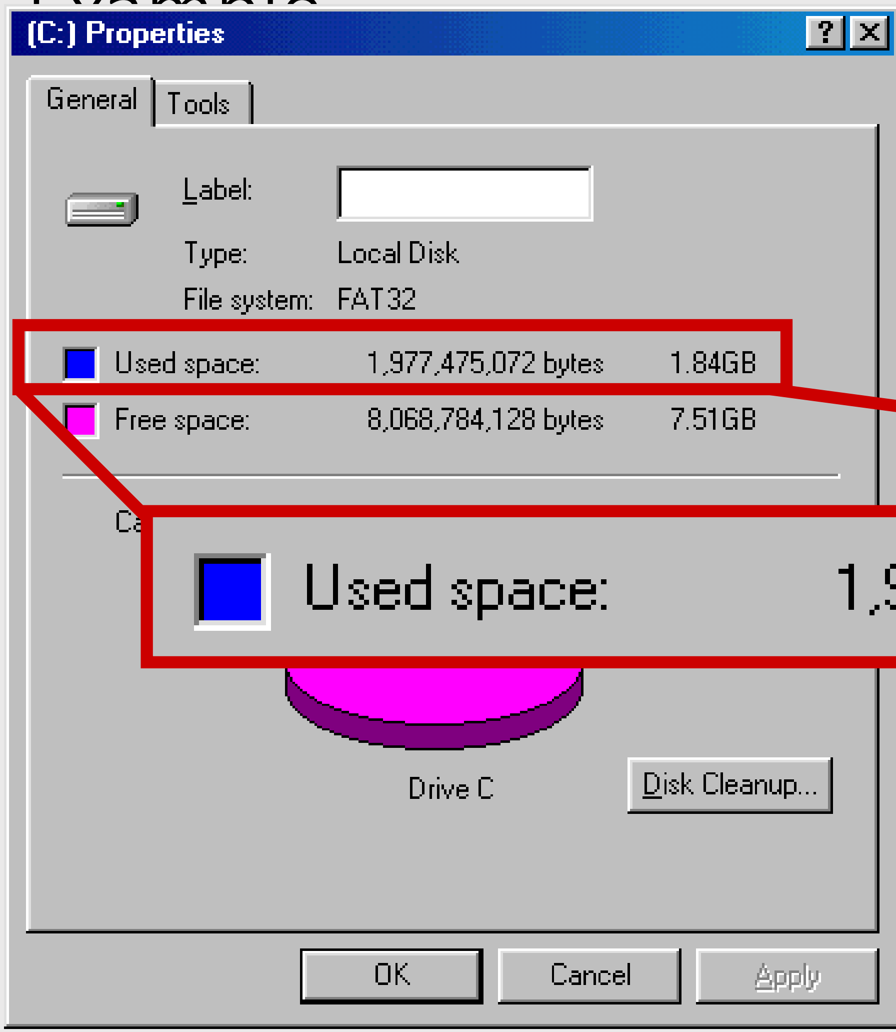
Common Powers (2 of 2)

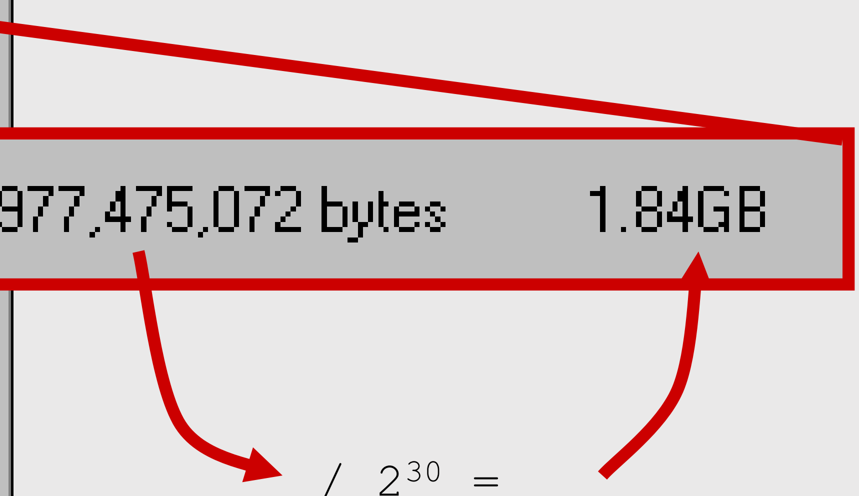
- Base 2

Power	Preface	Symbol	Value
2^{10}	kilo	k	1024
2^{20}	mega	M	1048576
2^{30}	Giga	G	1073741824

- What is the value of “k”, “M”, and “G”?
- In computing, particularly w.r.t. memory, the base-2 interpretation generally applies

Example

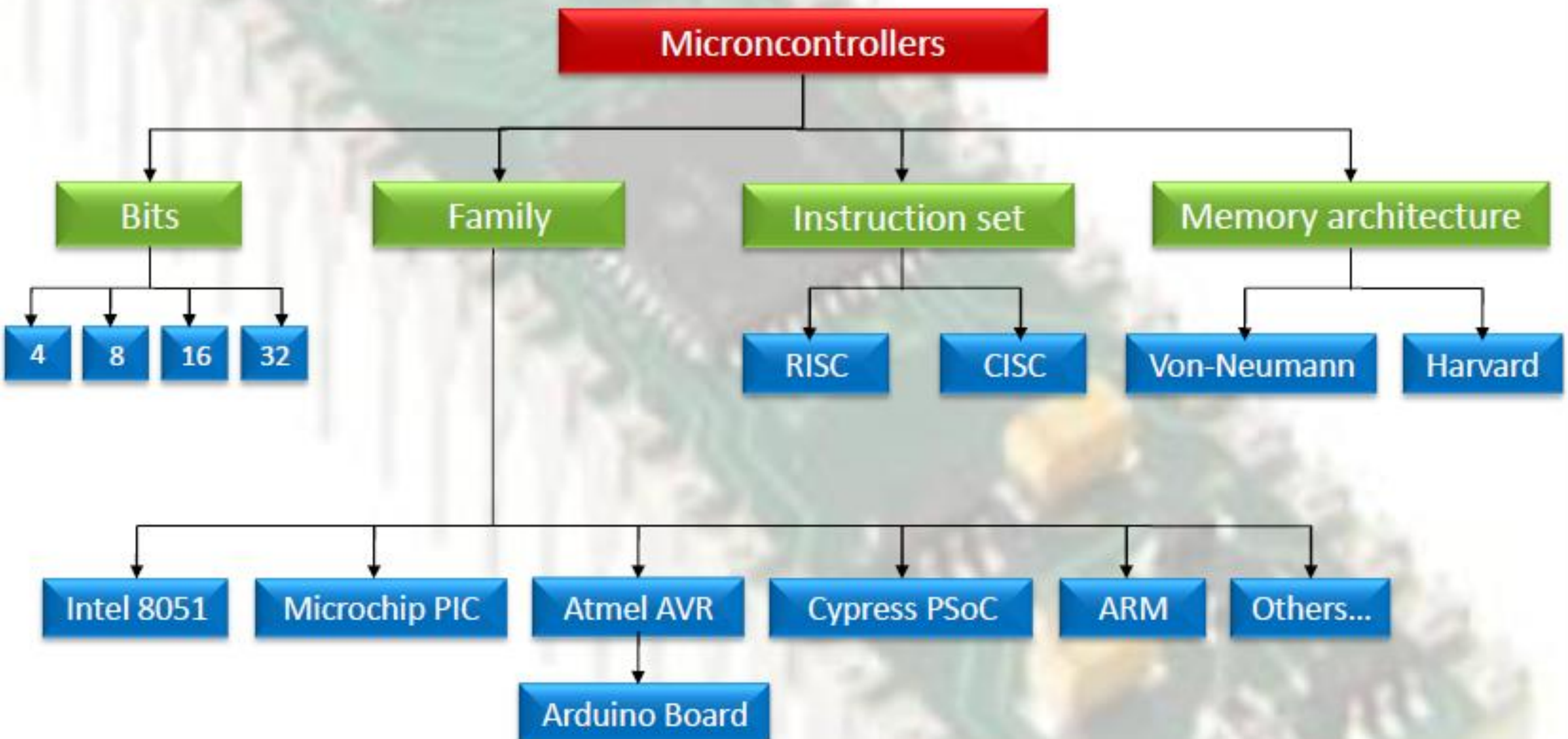


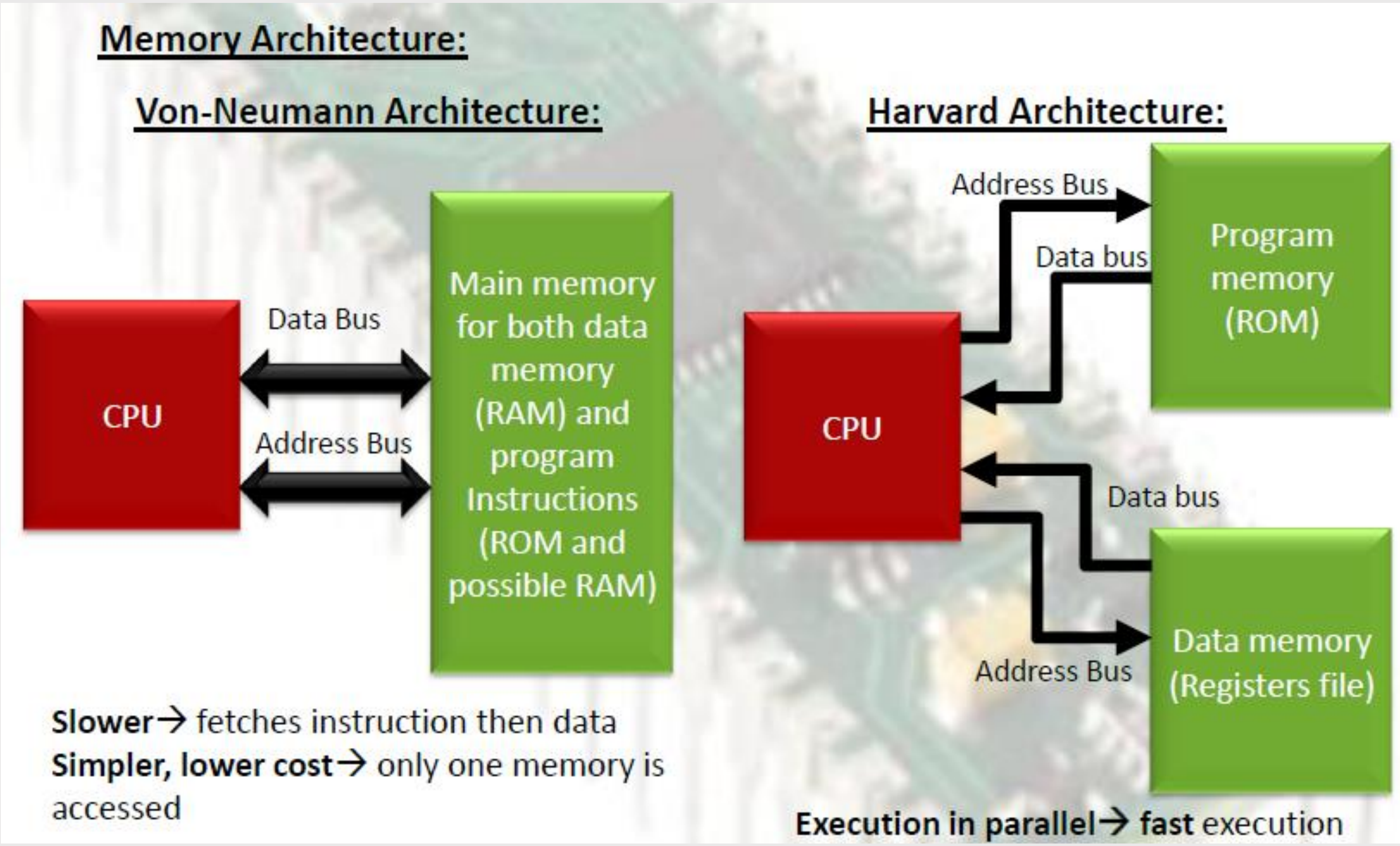


In the lab…

1. Double click on My Computer
2. Right click on C:
3. Click on Properties

Microcontrollers Classification




Memory Architecture:
Von-Neumann Architecture:
CPU
Data Bus
Address Bus
Main memory for both data memory (RAM) and program Instructions (ROM and possible RAM)
Slower→ fetches instruction then data
Simpler, lower cost→ only one memory is accessed
Harvard Architecture:
Address Bus
Data bus
Program memory (ROM)
CPU
Data bus
Address Bus
Data memory (Registers file)
Execution in parallel→ fast execution

Endianess

- Little endian – little end (least significant byte) stored first (at lowest address), e.g. Intel microprocessors (Pentium etc)
- Big endian – big end stored first at low address, e.g. SPARC, Motorola microprocessors

Criteria for choosing a microcontroller

- Speed
  - Unit?
- Packaging
  - Types (DIP, QFP)?
- Power consumption
- Amount of RAM and ROM on the chip
- Number of I/O pins
- Peripherals (Timers, Comm Ports, ADCs)
- Cost per unit

Criteria for choosing a microcontroller

- Ease of development
  - Language
  - SDK tools
- Availability in future

Architecture types

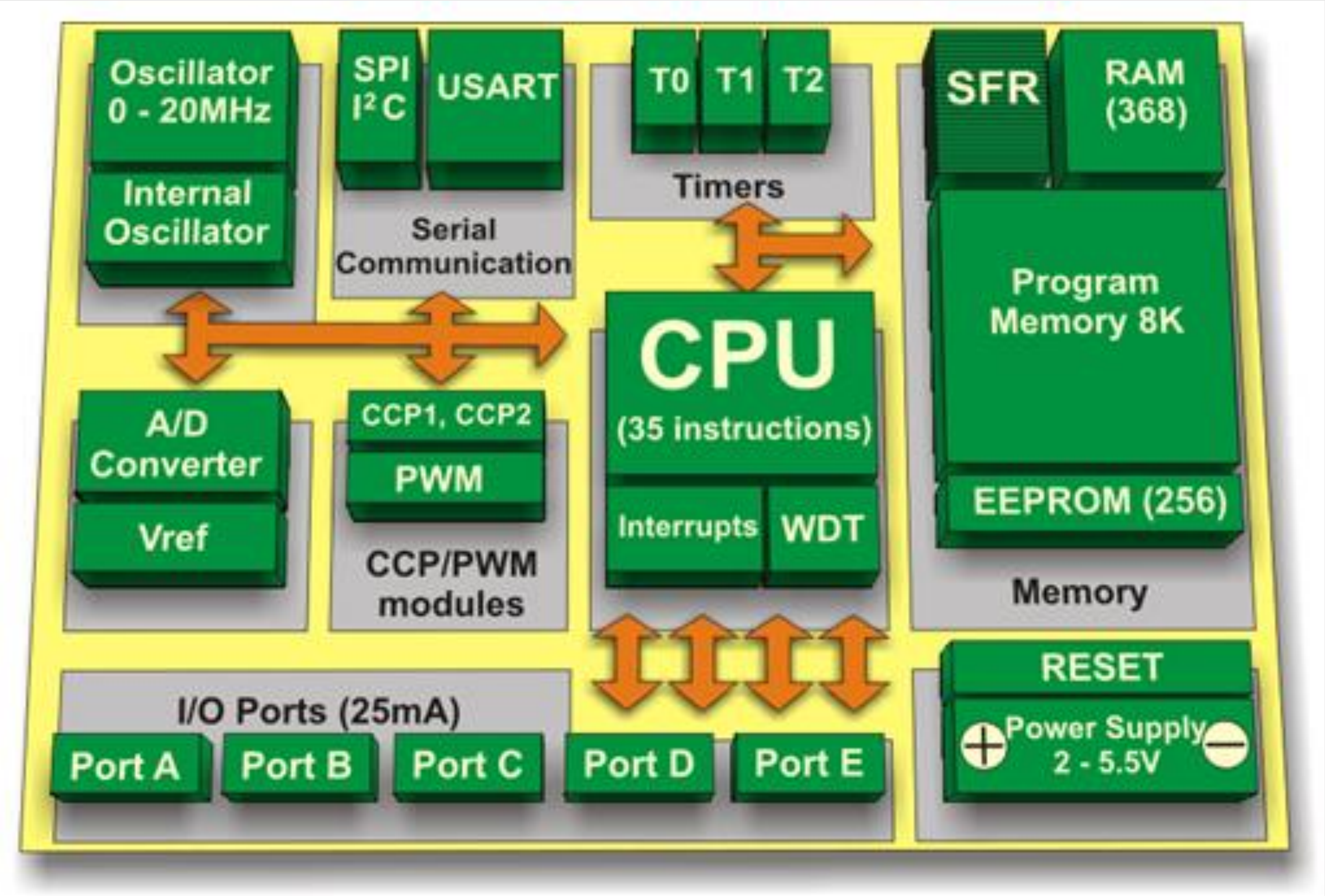
- Von Neumann/ Harvard
- Little endian / Big endian
- Fixed/ Floating (FPU)
- RISC/ CISC

Evaluation and development boards

LCD Explorer Development Board
(Part # DM240314)

PIC18 Architecture & Assembly Language Programming

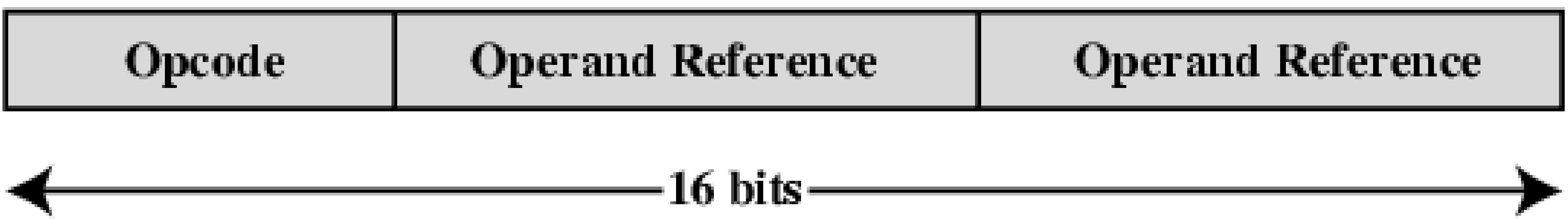
Before Architecture: Main Components of the PIC18



What is an Instruction Set?

- The complete collection of instructions that are understood by a CPU
- Machine language: binary representation of operations and (addresses of) arguments
- Assembly language: mnemonic representation for humans, e.g., OP A,B,C (meaning A <- OP(B,C))

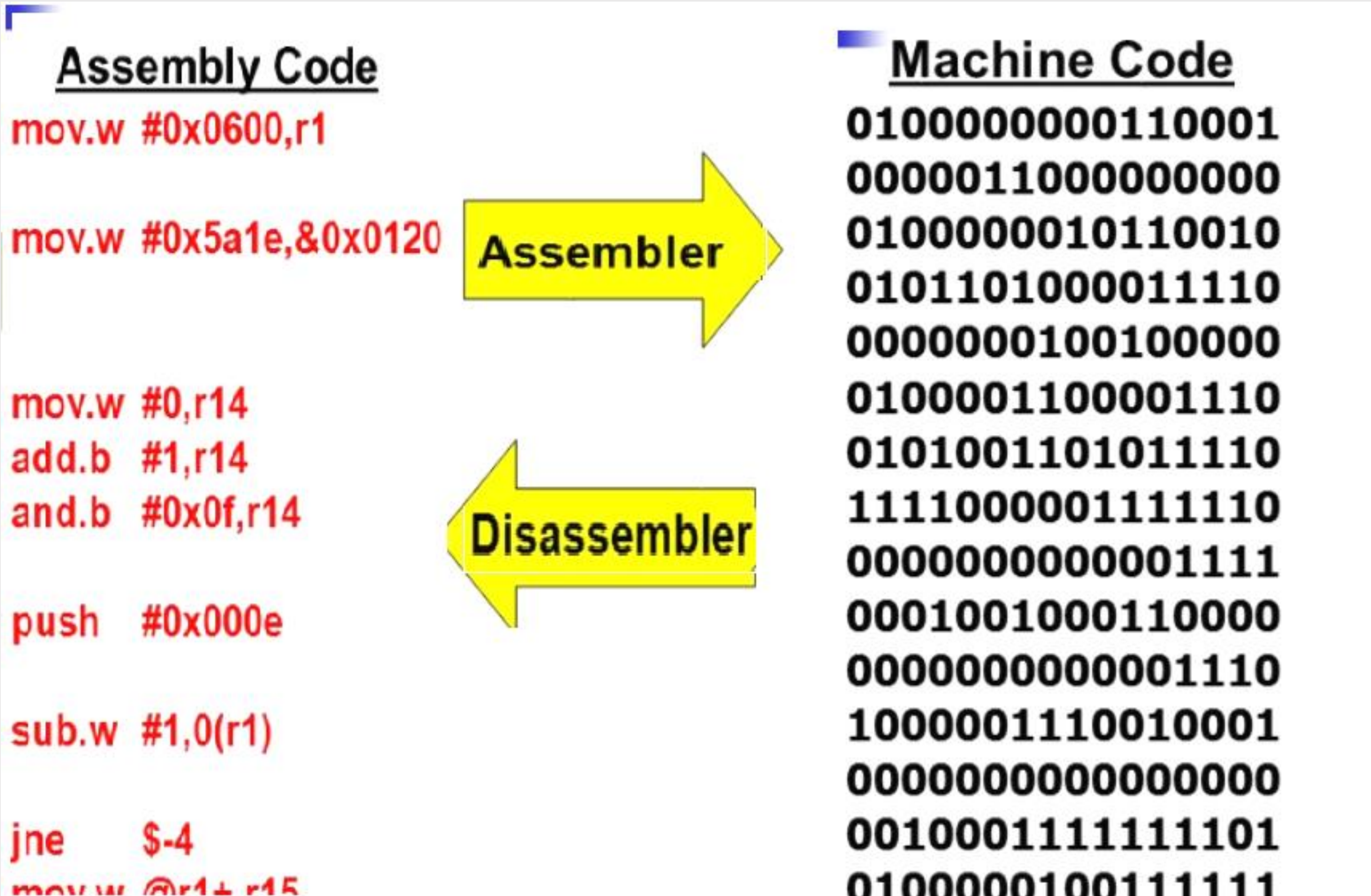
Simple Instruction Format in Machine Language (using two addresses)



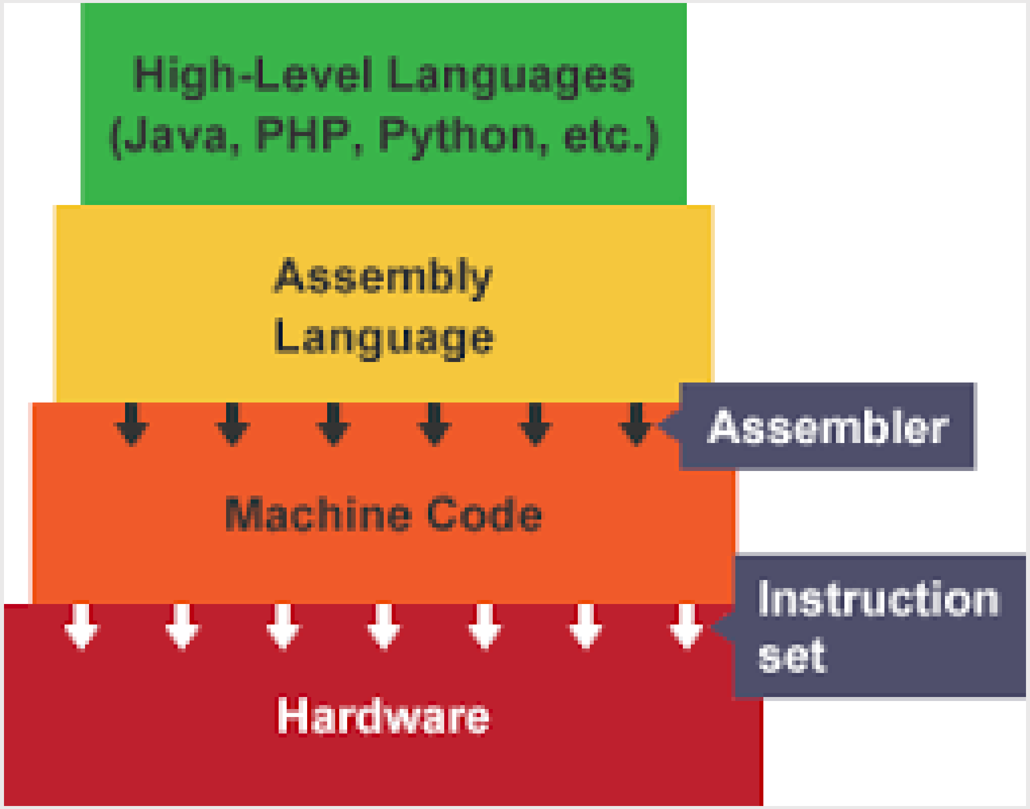
Elements of an Instruction in the form of Mnemonics (Assembly Language)

- Operation code (opcode)
  - Do this: ADD, SUB, MPY, DIV, LOAD, STOR
- Source operand reference
  - To this: (address of) argument of op, e.g. register, memory location
- Result operand reference
  - Put the result here (as above)

Machine and Assembly Code

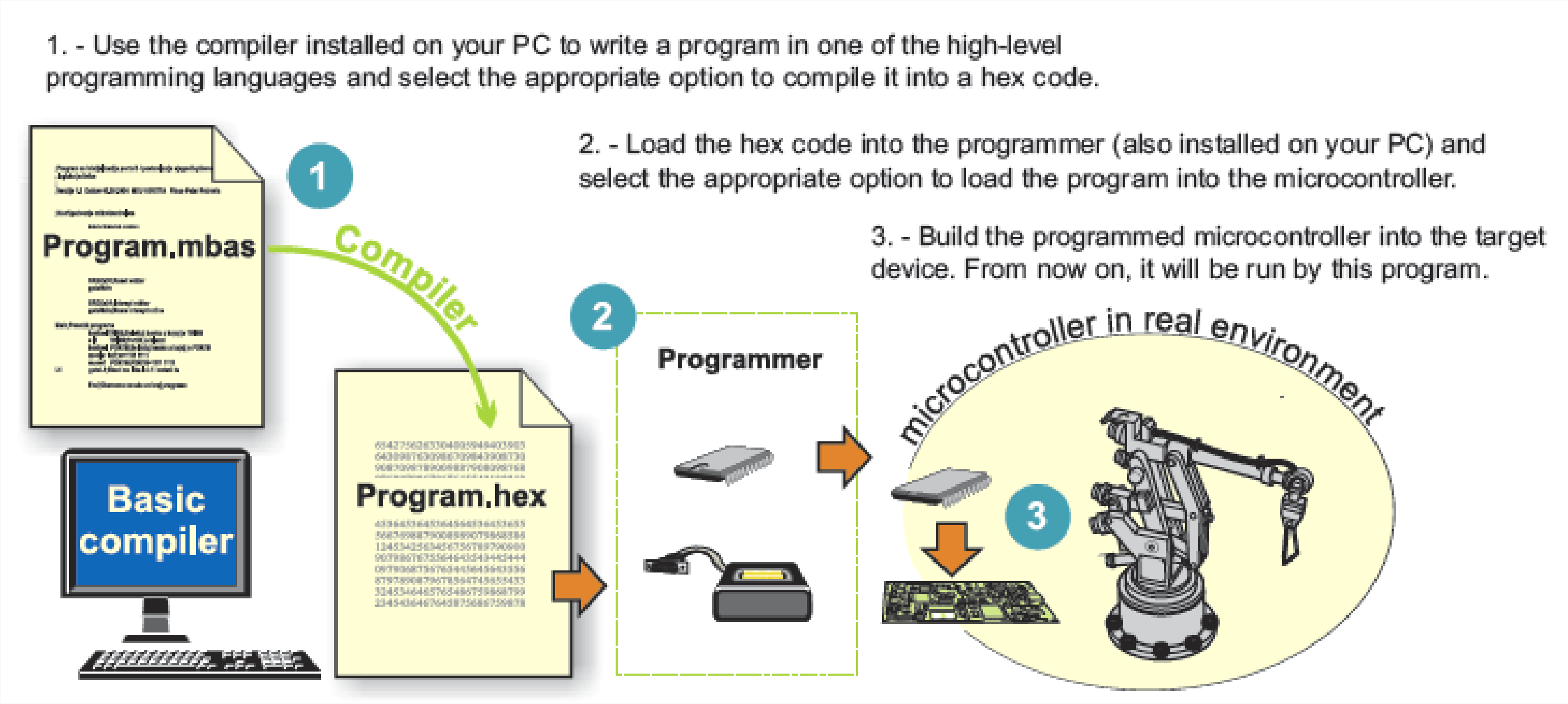


Levels of Abstraction

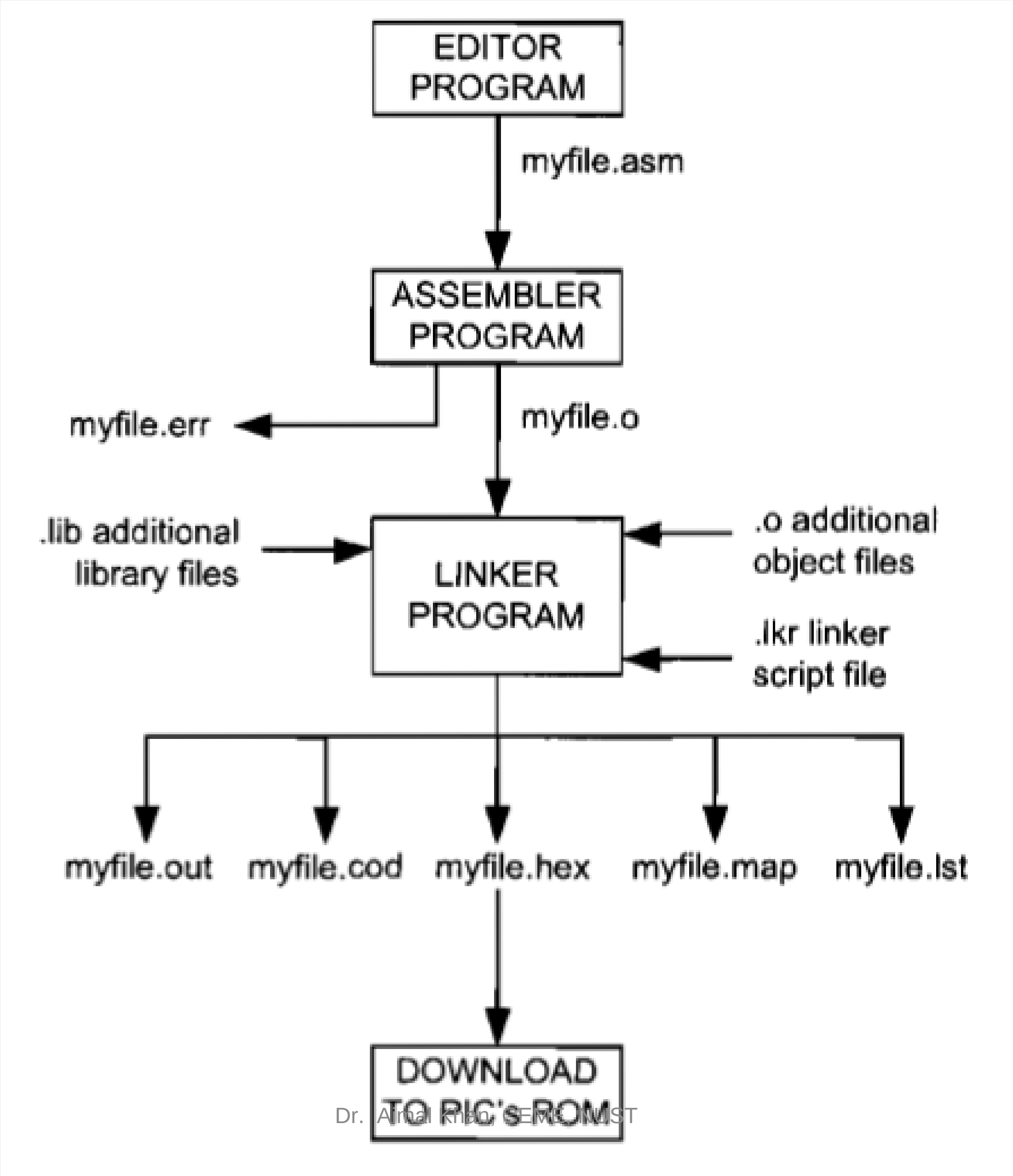


Compiler and Assembler

Pg-Out



Creating executable



Inside a computer

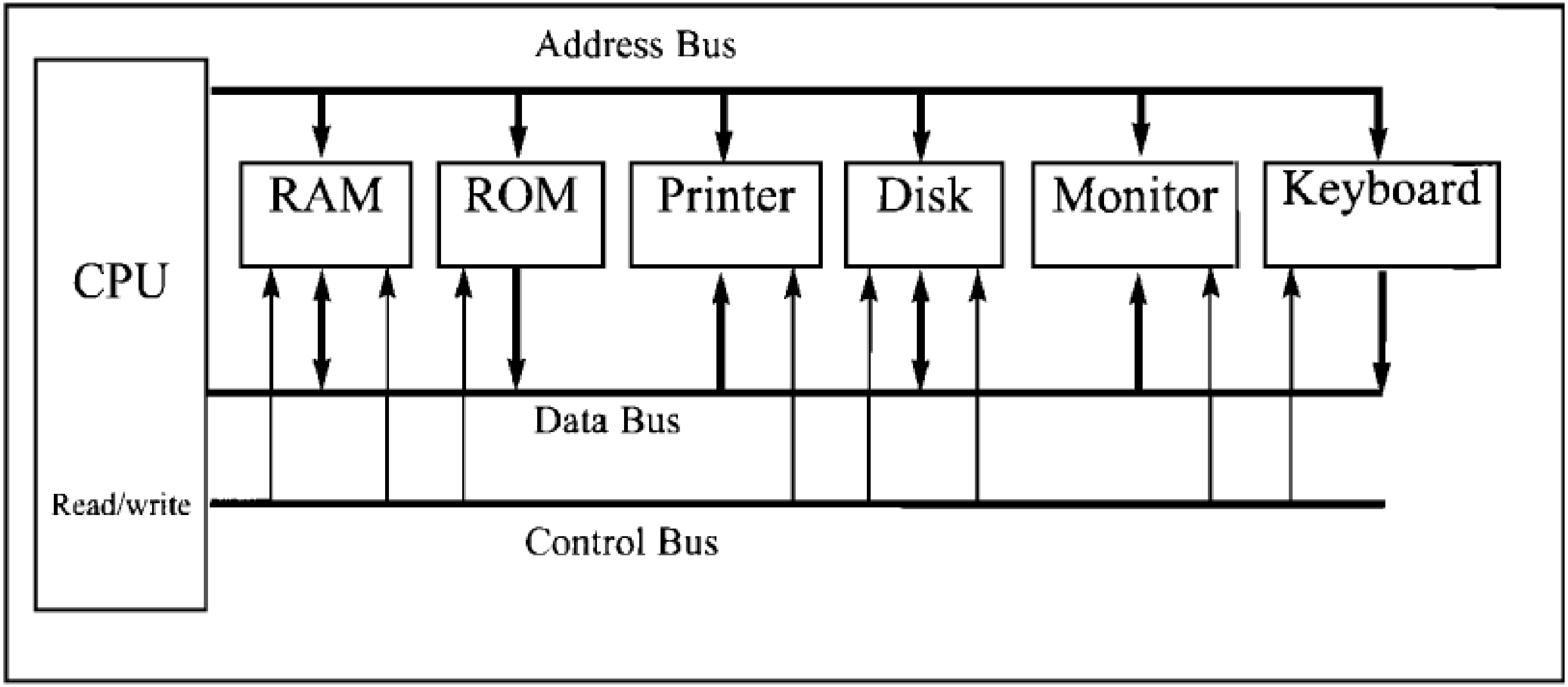


Figure 0-10. Internal Organization of a Computer

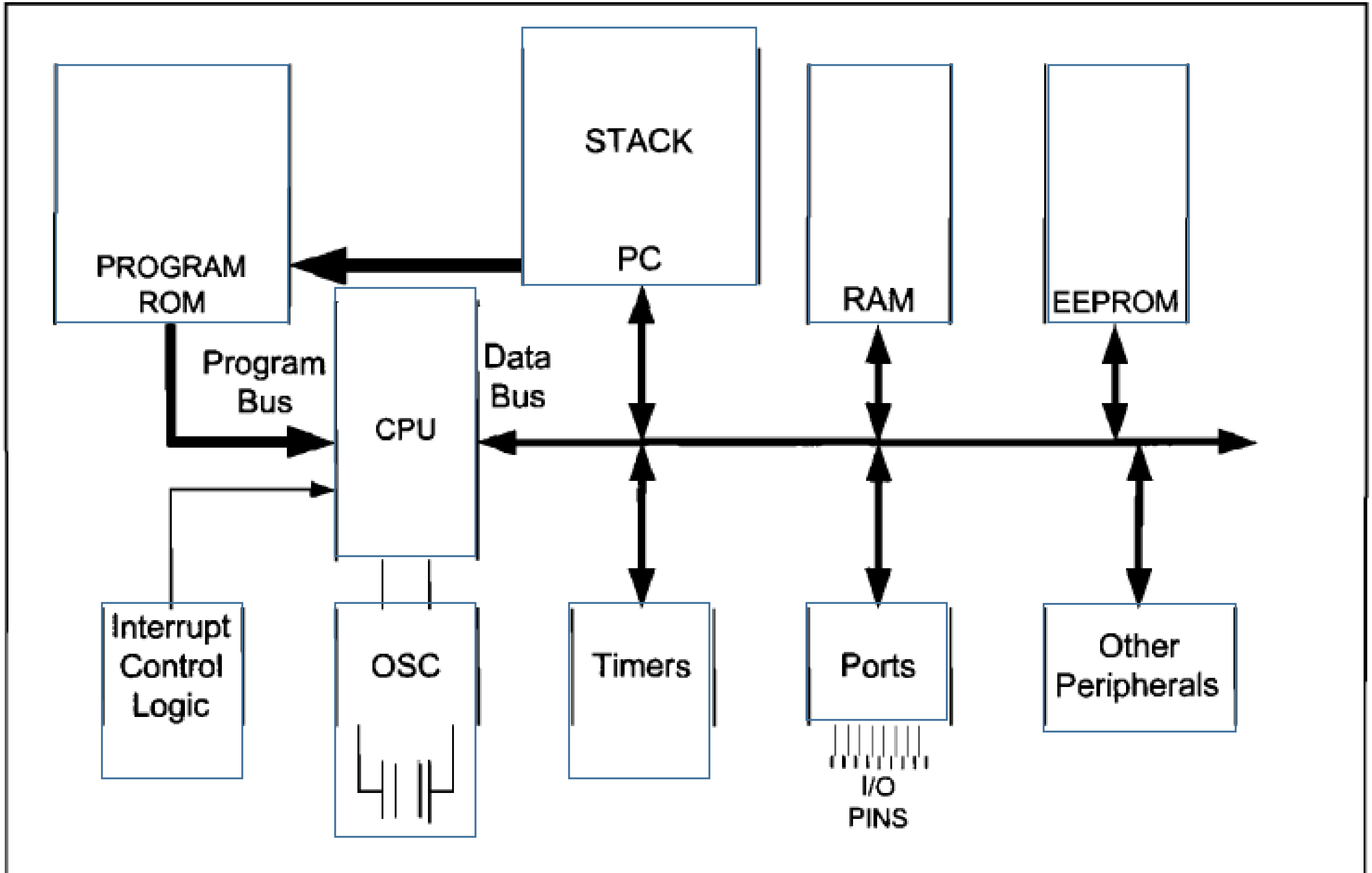
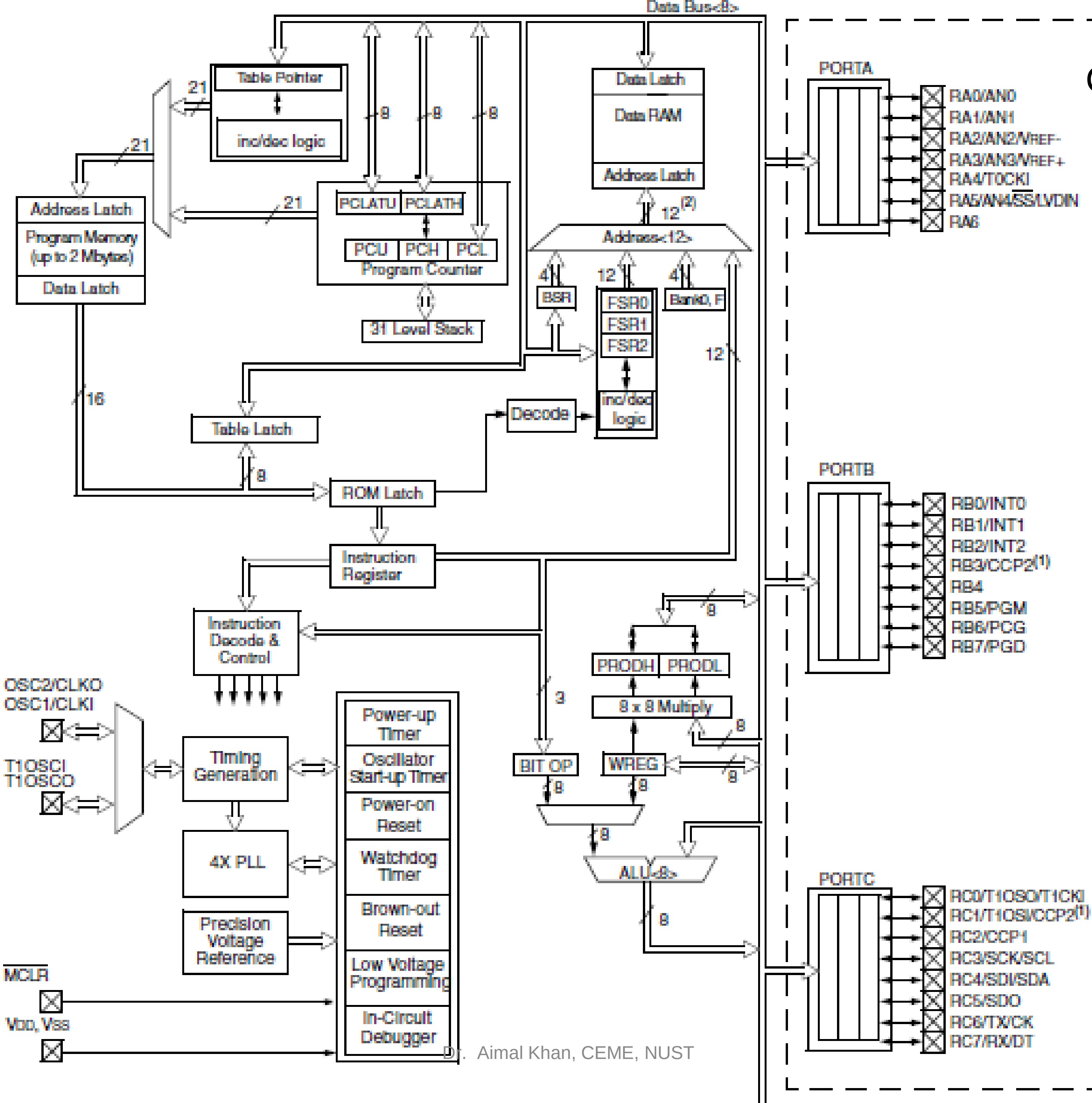


Figure 1-2. Simplified View of a PIC Microcontroller

List of Selected Microcontroller Families from Microchip

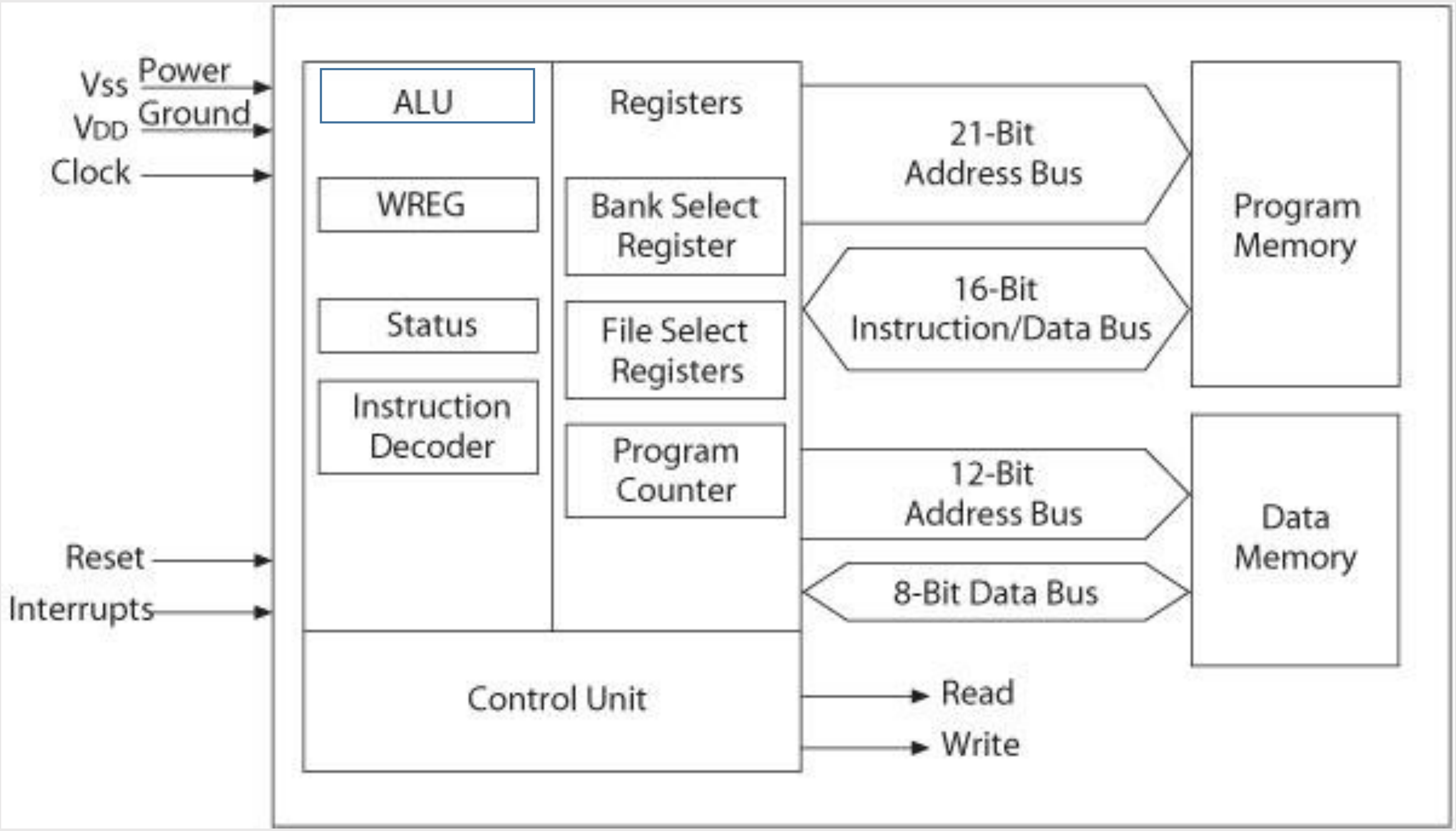
Part No.	Program OTP/Flash	EE PROM	RAM	Total Pins	I/O Pins	Analog ADC	Analog Comp.	Digital Timers/ WDT	Digital Serial I/O	CCP/ ECCP	Max Speed MHz	Instruc-tion Size	Total Instruc-tions
10F200	256x12 Flash		16	8	4			1-8 bit, 1-WDT			4	12-bit	33
10F220	256x12 Flash		16	8	4	2x8-bit	[illegible]	1-8 bit, 1-WDT			8	12-bit	33
12F510	1536x12 Flash		38	8	6	3x8-bit	1	1-8 bit 1-WDT			8	12-bit	33
16F506	1536x12 Flash		67	14	12	3x8-bit	2	1-8 bit 1-WDT			20	12-bit	33
16C55A	768x12 OTP		24	28	20			1-8 bit 1-WDT			40	12-bit	33
16CR58B	3072x12 ROM		73	18	12			1-8 bit 1-WDT			20	12-bit	33
12F683	2048x14 Flash	256	128	8	6	4x10-bit	1	1-16 bit, 2-8 bit, 1-WDT			20	14-bit	35
16F687	2048x14 Flash	256	128	20	18	12x10-bit	2	1-16 bit, 1-8 bit, 1-WDT	EU/I^2C/ SPI		20	14-bit	35
18F1230	2048x16 Enh Flash	128	256	18-28	16	4x10-bit	3	2-16 bit 1-WDT	EU		40	16-bit	77
18F4520	16384x16 Enh Flash	256	1536	40-44	36	13x10-bit	2	1-8 bit, 3-16 bit, 1-WDT	EU/ MI^2C /SPI	1/1	40	16-bit	77
18F6527	24576x16 Enh Flash	1024	3936	64	54	12x10-bit	2	2-8 bit, 3-16 bit, 1-WDT	2EU/ 2 - MI^2C /SPI	2/3	40	16-bit	77
18F8622	32768x16 Enh Flash	1024	3936	80	70	16x10-bit	2	2-8 bit, 3-16 bit, 1-WDT	2EU/ 2 - MI^2C /SPI	2/3	40	16-bit	77
18F96J60	32768x16 Flash		2048	100	72	16x10-bit	2	2-8 bit, 3-16 bit, 1-WDT	2EU/ 2 - MI^2C /SPI	2/3	42	16-bit	77
24FJ128GA-010	65536x16 Flash		8192	100-128	86	16x10-bit	2	5-16 bit, 1-WDT	2 –UART 2-I^2C/ SPI	5	32	16-bit	77

Abbreviations: 1) ADC: Analog-Digital Converter, 2) AUSART: Addressable USART, 3) CCP: Capture/Compare/PWM, 4) ECCP: Enhanced CCP, 5) EU: Enhanced USART, 6))Enh Flash: Enhanced Flash, 7) I^2C: Inter-integrated Circuit Bus, 8) MI^2C/SPI: Master I^2C /SPI, 9) OTP: One-Time Programmable, 10) SPI: Serial Peripheral Interface, 11) USART: Universal Synchronous/Asynchronous Receiver/Transmitter, 11) WDT: Watchdog Timer



Data Bus<8>
Table Pointer
inc/dec logic
21
Address Latch
Program Memory (up to 2 Mbytes)
Data Latch
16
PCLATU
PCLATH
PCU
PCH
PCL
Program Counter
31 Level Stack
Table Latch
ROM Latch
Instruction Register
Instruction Decode & Control
Data Latch
Data RAM
Address Latch
12(2)
Address<12>
BSR
FSR0
FSR1
FSR2
Bank0, F
inc/dec logic
Decode
PRODH
PRODL
8 x 8 Multiply
BIT OP
WREG
ALU<8>
OSC2/CLKO
OSC1/CLKI
T1OSCI
T1OSCO
Timing Generation
4X PLL
Precision Voltage Reference
Power-up Timer
Oscillator Start-up Timer
Power-on Reset
Watchdog Timer
Brown-out Reset
Low Voltage Programming
In-Circuit Debugger
MCLR
VDD, VSS
PORTA
RA0/AN0
RA1/AN1
RA2/AN2/VREF-
RA3/AN3/VREF+
RA4/T0CKI
RA5/AN4/SS/LVDIN
RA6
PORTB
RB0/INT0
RB1/INT1
RB2/INT2
RB3/CCP2(1)
RB4
RB5/PGM
RB6/PGC
RB7/PGD
PORTC
RC0/T1OSO/T1CKI
RC1/T1OSI/CCP2(1)
RC2/CCP1
RC3/SCK/SCL
RC4/SDI/SDA
RC5/SDO
RC6/TX/CK
RC7/RX/DT

PIC18F – MPU and Memory



PIC18F Microcontrollers

- Microcontroller Unit (MCU)
  - Microprocessor unit (MPU)
  - Harvard Architecture
    - Program memory for instructions
    - Data memory for data
  - I/O ports
  - Support devices such as timers

PIC18F Instructions

- 77 assembly language instructions
  - Earlier PIC families have 33 or 35 instructions
- PIC18F instruction set
  - Most instructions are 16-bit word length

Microprocessor Unit

- Includes Arithmetic Logic Unit (ALU), Registers, and Control Unit
  - Arithmetic Logic Unit (ALU)
  - Instruction Decoder

    -16 bit instruction
  - Status register that stores flags

    - 5-bits
  - WREG – Working Register

    8- bit accumulator

Microprocessor Unit

- Registers
  - Program Counter (PC)
    - 21-bit register that holds the Program Memory address
  - Bank Select Register (BSR)
    - 4-bit register used to select a bank in the memory
  - File Select Registers (FSRs)
    - 12-bit registers used as memory pointers in indirect addressing Data Memory
- Control unit
  - Provides timing and control signals
    - Read and Write operations

PIC18F - Address Buses

- Address bus
  - 21-bit address bus for Program Memory
    - Addressing capacity: 2 MB
  - 12-bit address bus for Data Memory
    - Addressing capacity: 4 KB

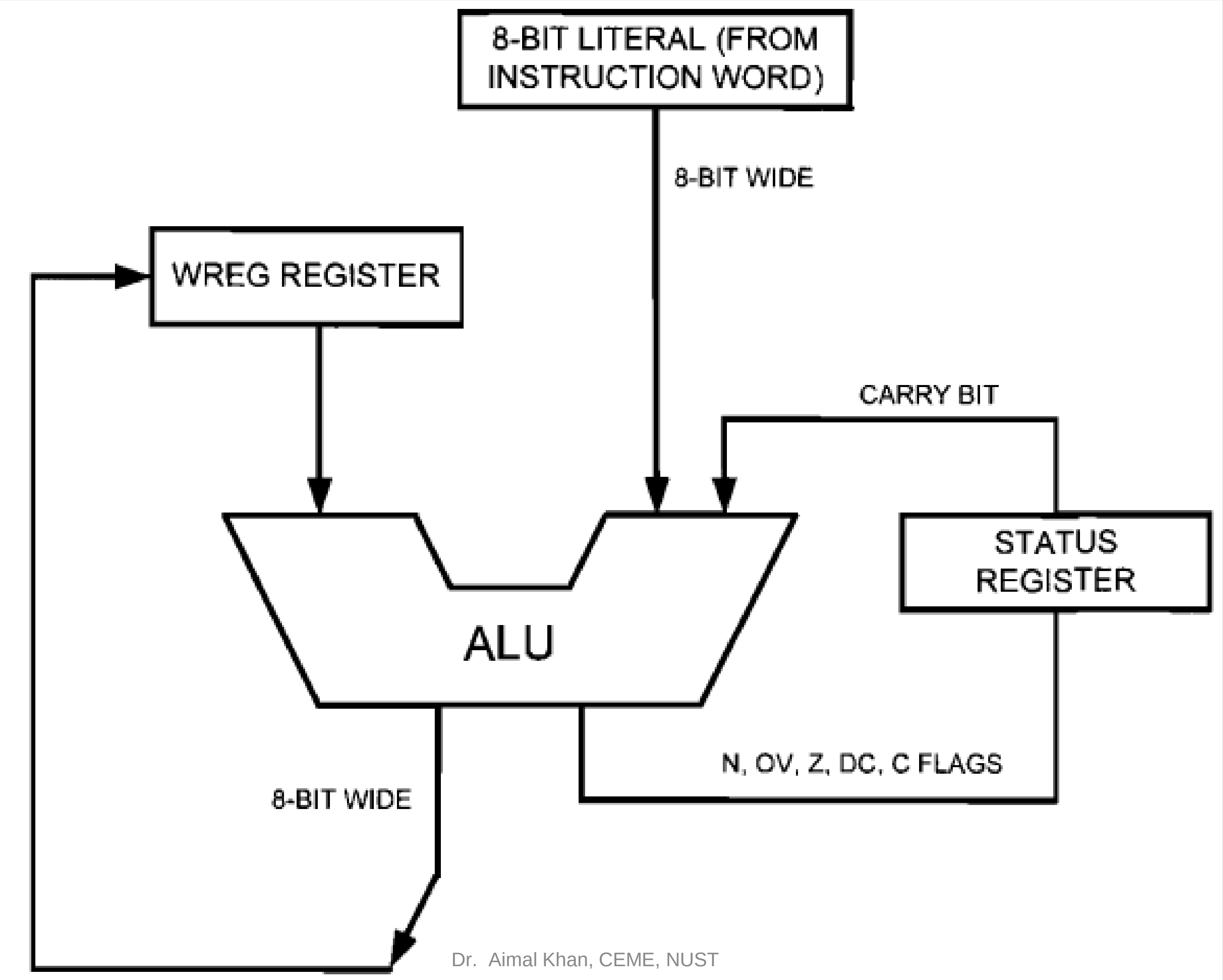
Data Bus and Control Signals

- Data bus
  - 16-bit instruction/data bus for Program Memory
  - 8-bit data bus for Data Memory
- Control signals
  - Read and Write

PIC18F452/4520 Memory

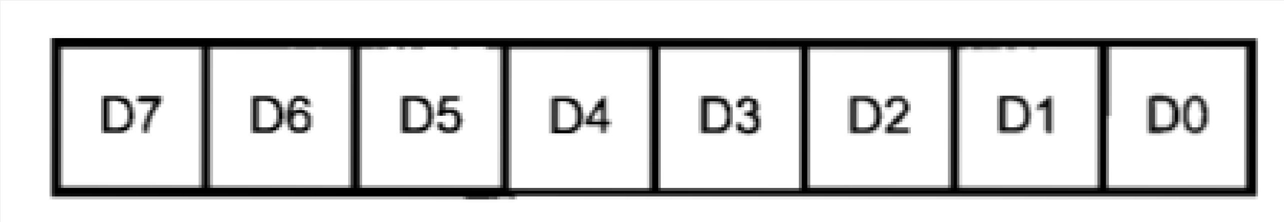
- Program Memory: 32 K
  - Address range: 000000 to $007FFF_H$
- Data Memory: 4 K
  - Address range: 000 to FFF_H

Datapath


8-BIT LITERAL (FROM INSTRUCTION WORD)
8-BIT WIDE
WREG REGISTER
CARRY BIT
STATUS REGISTER
ALU
N, OV, Z, DC, C FLAGS
8-BIT WIDE

PIC registers

- WREG (working register)



  - Aka as accumulator in microprocessors
  - Used for all arithmetic and logic instructions
  - Only one in PIC18 family

WREG instructions

- MOVLW
  - moves 8-bit data into the WREG register
- ADDLW
  - add the literal value K to register WREG and put the result back in the WREG register
- Example

```
MOVLW 25H   ;load 25H into WREG
ADDLW 34H   ;add value 34 to W(W = W + 34H)
```

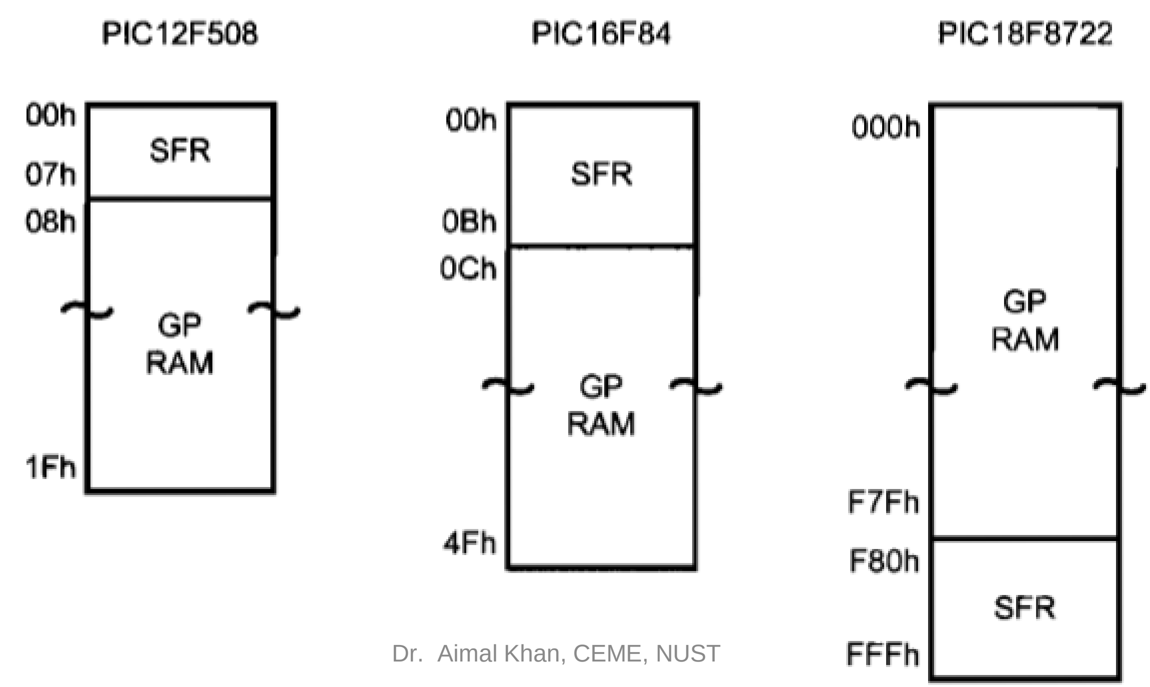
File register

- Divided in 2 sections
- Data RAM
  - used for data storage and scratch pad
- SFRs (Special function registers)
  - dedicated to specific functions such as ALU status, timers, serial communication, I/O ports, ADC

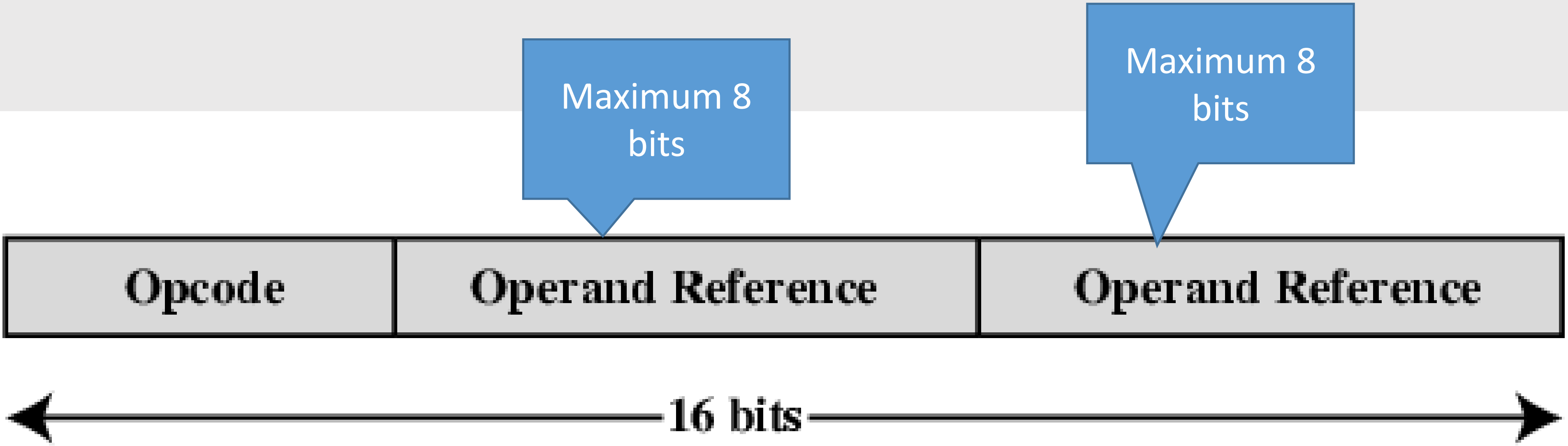
File register

Table 2-1: File Register Size for PIC Chips

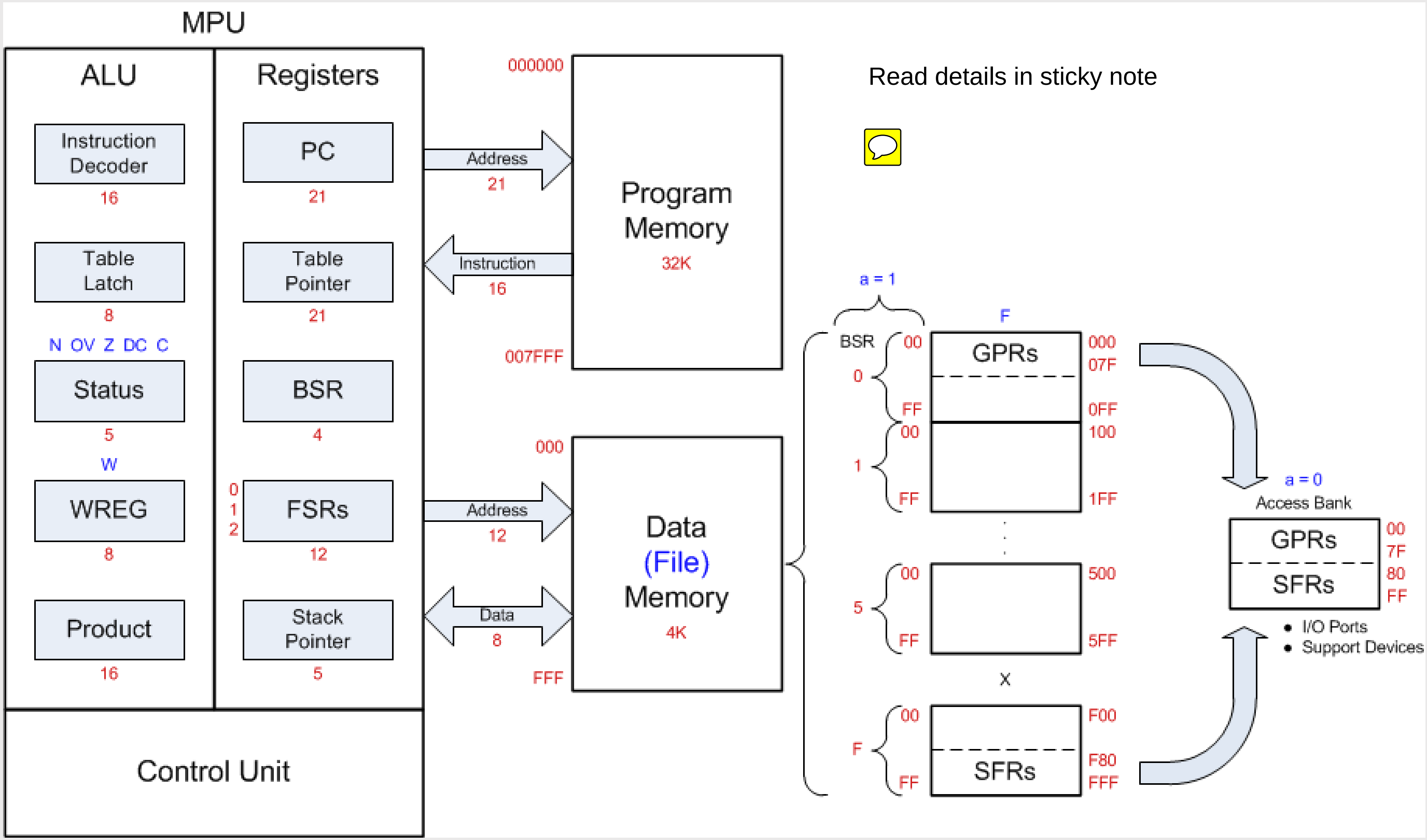
	File Register (Bytes)	=	SFR (Bytes)	+	Available space for GPR (Bytes)
PIC12F508	32		7		25
PIC16F84	80		12		68
PIC18F1220	512		256		256
PIC18F452	1792		256		1536
PIC18F2220	768		256		512
PIC18F458	1792		256		1536
PIC18F8722	4096		158		3938



Simple Instruction Format in Machine Language (using two addresses)



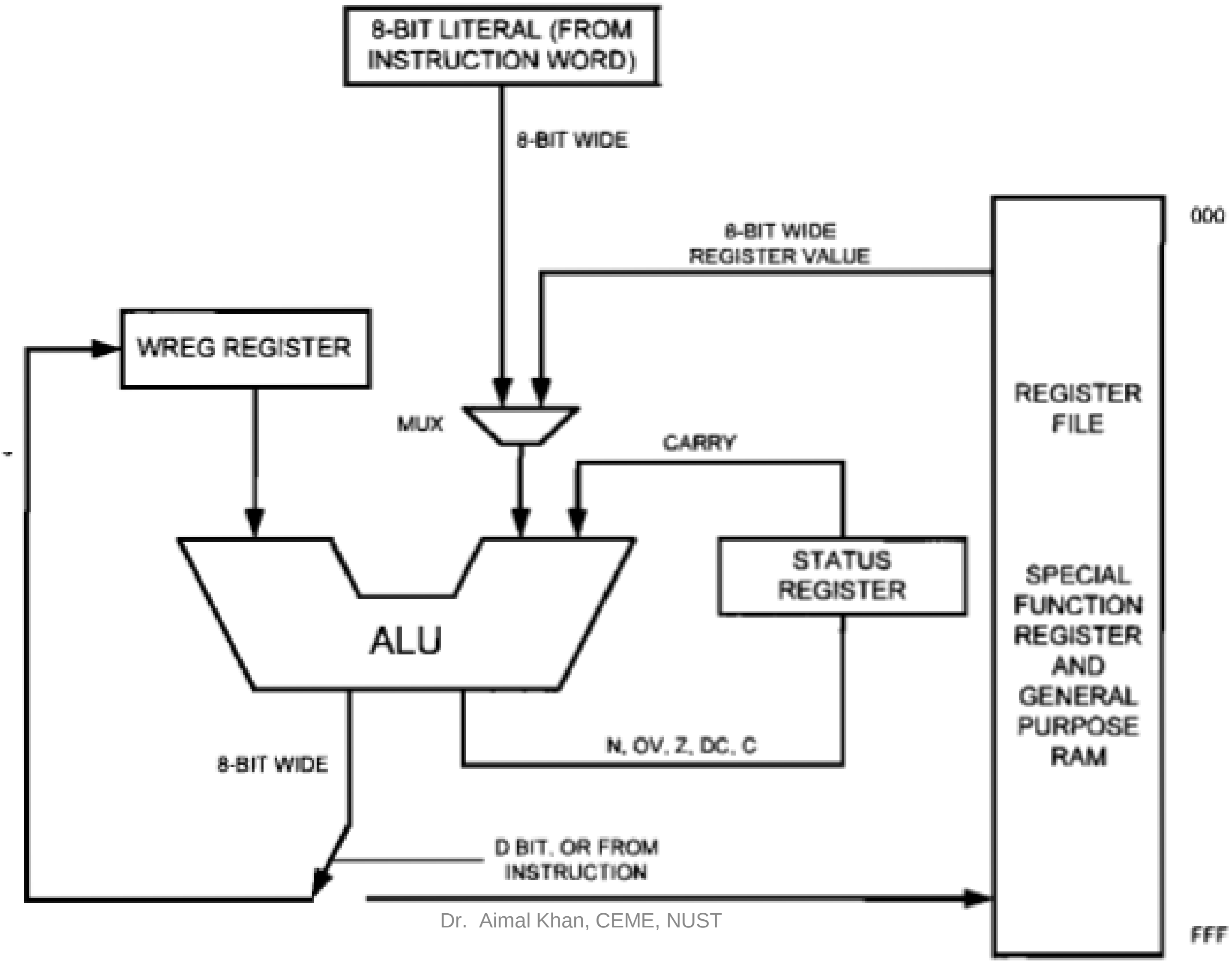
PIC18F452 Programming Model



SFRs in PIC18

F80h	PORTA	FA0h	PIE2	FC0h	----		FE0h	BSR	
F81h	PORTB	FA1h	PIR2	FC1h	ADCON1		FE1h	FSR1L	
F82h	PORTC	FA2h	IPR2	FC2h	ADCON0		FE2h	FSR1H	
F83h	PORTD	FA3h	----	FC3h	ADRESL		FE3h	PLUSW1	*
F84h	PORTE	FA4h	----	FC4h	ADRESH		FE4h	PREINC1	*
F85h	----	FA5h	----	FC5h	SSPCON2		FE5h	POSTDEC1	*
F86h	----	FA6h	----	FC6h	SSPCON1		FE6h	POSTINC1	*
F87h	----	FA7h	----	FC7h	SSPSTAT		FE7h	INDF1	*
F88h	----	FA8h	----	FC8h	SSPADD		FE8h	WREG	
F89h	LATA	FA9h	----	FC9h	SSPBUF		FE9h	FSR0L	
F8Ah	LATB	FAAh	----	FCAh	T2CON		FEAh	FSR0H	
F8Bh	LATC	FABh	RCSTA	FCBh	PR2		FEBh	PLUSW0	*
F8Ch	LATD	FACh	TXSTA	FCCh	TMR2		FECh	PREINC0	*
F8Dh	LATE	FADh	TXREG	FCDh	T1CON		FEDh	POSTDEC0	*
F8Eh	----	FAEh	RCREG	FCEh	TMR1L		FEEh	POSTINC0	*
F8Fh	----	FAFh	SPBRG	FCFh	TMR1H		FEFh	INDF0	*
F90h	----	FB0h	----	FD0h	RCON		FF0h	INTCON3	
F91h	----	FB1h	T3CON	FD1h	WDTCON		FF1h	INTCON2	
F92h	TRISA	FB2h	TMR3L	FD2h	LVDCON		FF2h	INTCON	
F93h	TRISB	FB3h	TMR3H	FD3h	OSCCON		FF3h	PRODL	
F94h	TRISC	FB4h	----	FD4h	----		FF4h	PRODH	
F95h	TRISD	FB5h	----	FD5h	T0CON		FF5h	TABLAT	
F96h	TRISE	FB6h	----	FD6h	TMR0L		FF6h	TBLPTRL	
F97h	----	FB7h	----	FD7h	TMR0H		FF7h	TBLPTRH	
F98h	----	FB8h	----	FD8h	STATUS		FF8h	TBLPTRU	
F99h	----	FB9h	----	FD9h	FSR2L		FF9h	PCL	
F9Ah	----	FBAh	CCP2CON	FDAh	FSR2H		FFAh	PCLATH	
F9Bh	----	FBBh	CCPR2L	FDBh	PLUSW2	*	FFBh	PCLATU	
F9Ch	----	FBCh	CCPR2H	FDCh	PREINC2	*	FFCh	STKPTR	
F9Dh	PIE1	FBDh	CCP1CON	FDDh	POSTDEC2	*	FFDh	TOSL	
F9Eh	PIR1	FBEh	CCPR1L	FDEh	POSTINC2	*	FFEh	TOSH	
F9Fh	IPR1	FBFh	CCPR1H	FDFh	INDF2	*	FFFh	TOSU	

Operation Procedure



ADDWF instruction

- adds together the contents of WREG and a file register location
- Used both working and file registers
- Format

ADDWF fileReg, D, a

  - Where D is the destination (‘w’ or ‘f’)
  - a is 1 for no access bank and 0 for access bank
  - If a is not specified then A =0 is considered

Default Access Bank Instructions

- MOVWF instruction
  - tells the CPU to copy the source register WREG to a destination in the file register (F)
  - literal (immediate) values cannot be moved directly into the general-purpose RAM locations

Example- MOVWF


```
MOVLW  99H        ;load WREG with value 99H
MOVWF  12H
MOVLW  85H        ;load WREG with value 85H
MOVWF  13H
MOVLW  3FH        ;load WREG with value 3FH
MOVWF  14H
MOVLW  63H        ;load WREG with value 63H
MOVWF  15H
MOVLW  12H        ;load WREG with value 12H
MOVWF  16H
```

Address	Data
012	99
013	85
014	3F
015	63
016	12

Example- file register instructions


```
MOVLW  0              ;move 0 WREG to clear it (WREG = 0)
MOVWF  12H            ;move WREG to location 12 to clear it
MOVLW  22H            ;load WREG with value 22H
ADDWF  12H, F         ;add WREG to loc 12H, loc 12 = sum
ADDWF  12H, F         ;add WREG to loc 12H, loc 12 = sum
ADDWF  12H, F         ;add WREG to loc 12H, loc 12 = sum
ADDWF  12H, F         ;add WREG to loc 12H, loc 12 = sum
```

- Memory contents

Address	Data
011	
012	22
013	

Address	Data
011	
012	44
013	

Address	Data
011	
012	66
013	

Address	Data
011	
012	88
013	