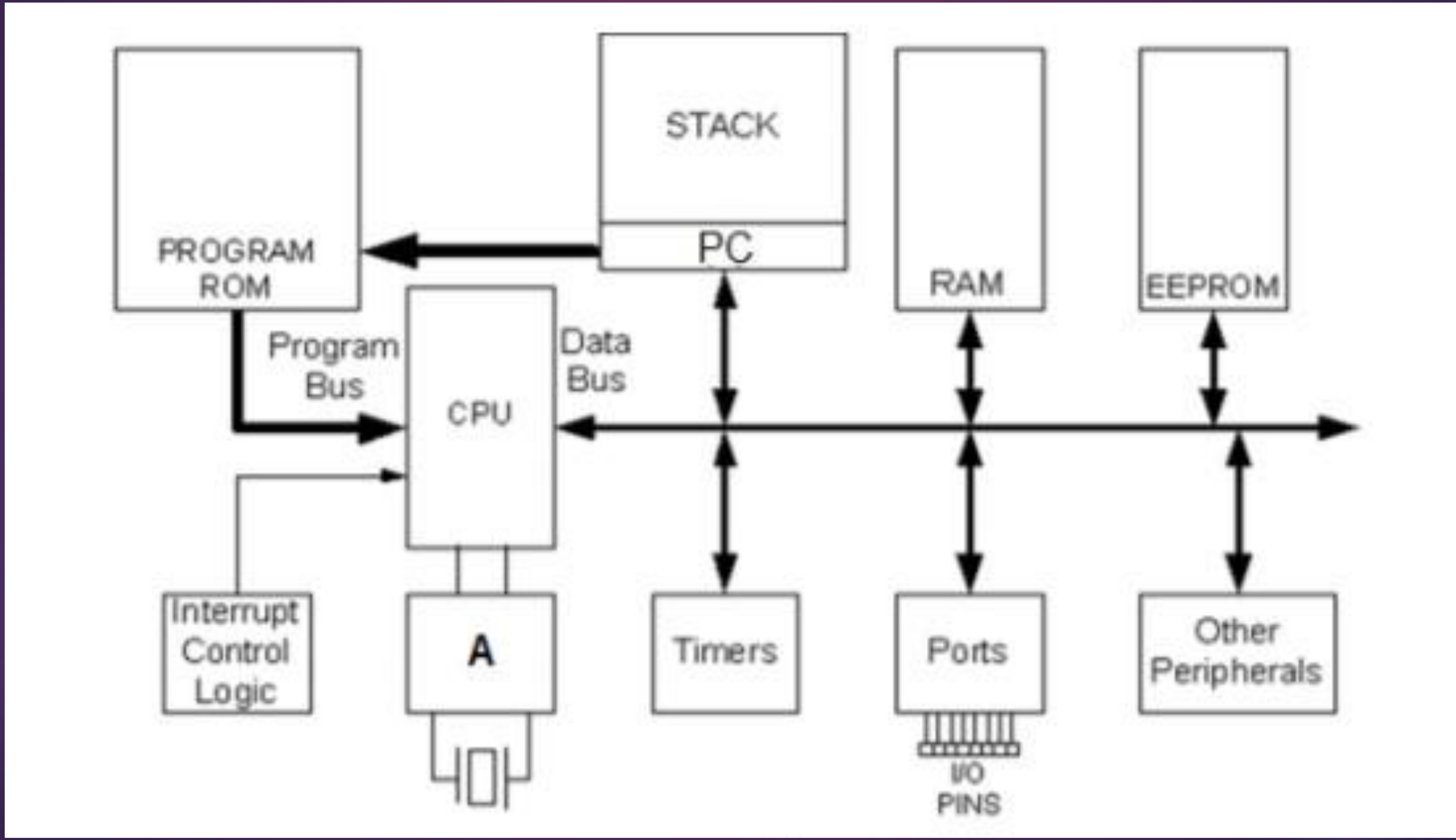
Microprocessor Interfacing & Programming

LECTURE 3&4

Overview of the PIC18 Family

- In 1989, Microchip Technology Corporation introduced an 8-bit microcontroller called the PIC, Peripheral Interface Controller.
- It has small amounts of RAM, few hundred bytes of on-chip ROM, one timer and a few pins for I/O ports.
- All PIC families are 8 bit processors, means that the processor can work on only 8 bits of data at a time.
- One major problem with PIC is that they are not completely compatible in terms of software when going from one family to another family.

Simplified View of a PIC Microcontroller



PIC18 Features

- RISC architecture
- ROM
- RAM
- EEPROM
- Timers
- ADC
- USART
- I/O ports.

Comparison of 8051 and PIC18 Family (40-pin package)

Feature	8051/52	PIC18xxx
Program ROM (maximum space)	64K	2M
Data RAM (maximum space)	256 bytes	4K
Timers	3	4
I/O pins	32	33
Serial port	1	1

Oscillator in microcontrollers

- An oscillator in a microcontroller is an electronic circuit that generates a continuous clock signal (a series of pulses). This clock signal is what the microcontroller’s CPU uses to synchronize and time all operations.
- Without an oscillator, the controller wouldn’t know when to execute the next instruction.
- The oscillator also determines how fast the microcontroller executes code.
- Example: A 16 MHz oscillator means 16 million clock pulses per second.

RISC vs CISC

- **RISC (Reduced Instruction Set Computer)**

Use a small set of simple instructions that execute very fast (usually in 1 clock cycle).

More instructions are needed to perform a complex task, but each instruction is simple and fast.

Examples: ARM, AVR (Arduino), MIPS, RISC-V.

- **CISC (Complex Instruction Set Computer)**

Use a large set of complex instructions, some of which can perform multi-step operations in a single instruction.

Fewer instructions are needed for a task, but each instruction may take multiple clock cycles.

Examples: Intel x86 (PC processors), laptops

Example

- **Task:**

C = A + B ;Where A, B, C are stored in memory.

RISC Example ; Assume A, B, C are in memory at addresses addrA, addrB, addrC

LW R1, addrA ; Load A into register R1

LW R2, addrB ; Load B into register R2

ADD R3, R1, R2 ; R3 = R1 + R2

SW R3, addrC ; Store R3 into C

- register to register operation

- **CISC Example**

```
MOV  AX, [addrA]    ; Load A directly into AX
ADD  AX, [addrB]    ; AX = AX + B (read B, add directly from memory)
MOV  [addrC], AX    ; Store AX into C
```

- RISC uses many general purpose registers while CISC uses fewer registers and relies more on memory.
- RISC or CISC defines how the processor is designed, what its instruction set looks like, and how it executes code.
- It is about processor design----How hardware executes instructions.

Difference between RISC and CISC

- CISC can use memory directly because its CPU is designed with complex hardware that decodes and executes heavy instructions internally.
- RISC keeps things simple and predictable by forcing all data to go through registers, which makes the chip easier to design and faster to clock.

PIC18 Family

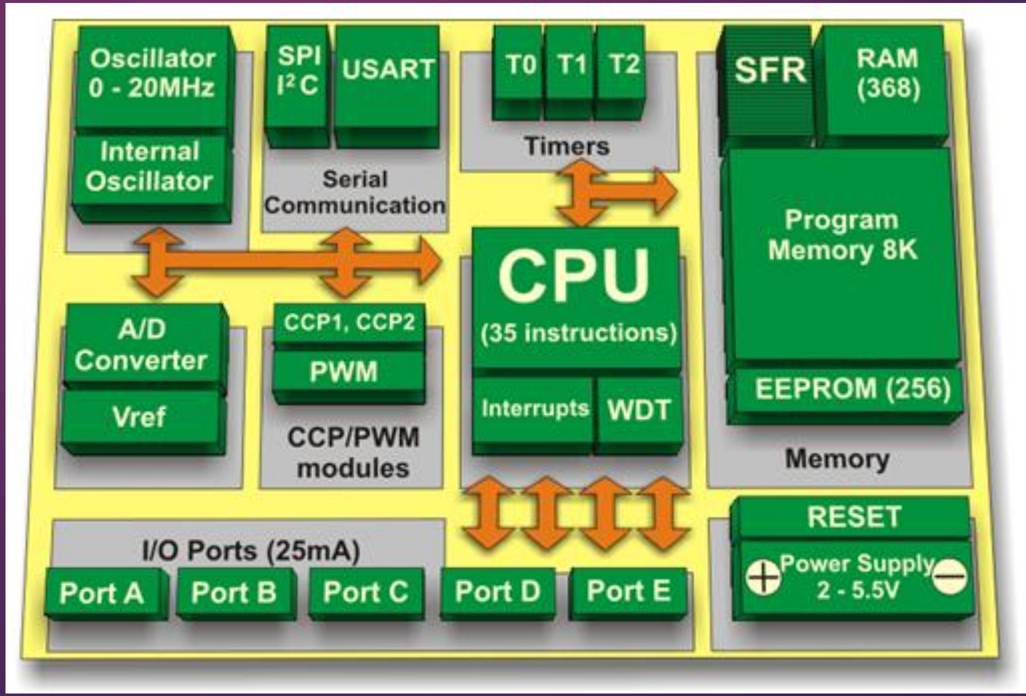
- ROM is used to store program (code).
- RAM is used for data storage.
- PIC18 has a maximum of 4096 bytes (4K) of data RAM space.
- In some PIC18 family members, there is a small amount of EEPROM to store critical data that does not need to be changed very often.
- They have from 16 to 72 pins dedicated for I/O.
- All the members of PIC18 family come with ADC, timers and USART as standard peripherals.
- PIC18 can have up to 4 timers besides the watchdog timer.

Watchdog Timer

- A Watchdog Timer (WDT) is a hardware timer inside a microcontroller that resets the system if the software stops responding or crashes.
- If your program gets stuck in an infinite loop or hangs, the watchdog will detect it and restart the controller automatically.

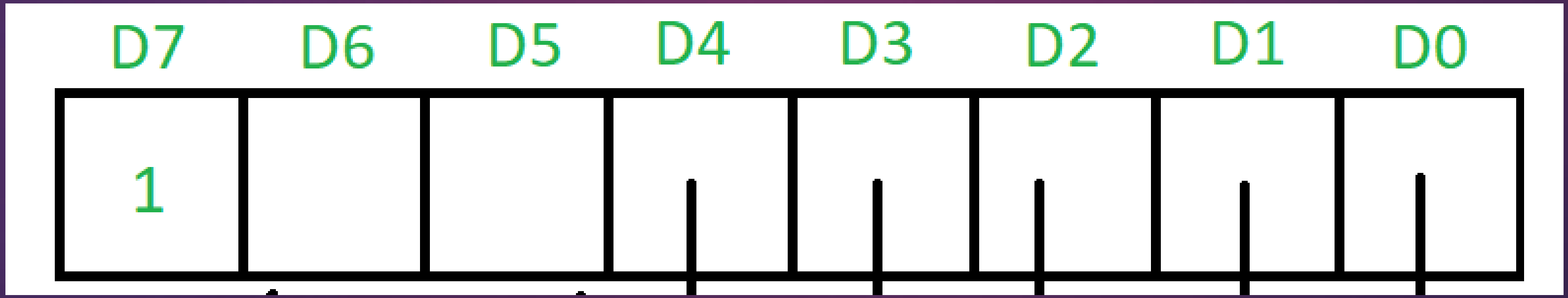
PIC Architecture

- Processor uses many registers to store data temporarily.
- We must understand the registers and architecture of a processor and the role they play in processing data.



WREG Register (Working register) in the PIC

- Among many registers for arithmetic and logic operations, there is a WREG register, the most widely used register.
- In processor, registers are used to store information temporarily. It could be a byte of data or an address pointing to the data to be fetched.
- In PIC, only one data type: 8-bit. D7 (MSB) to D0 (LSB)



Contd…

- With an 8-bit data type, any data larger than 8 bits must be broken into 8-bit chunks before it is processed.
- WREG is the same as the accumulator in other microprocessors.
- It is used for all arithmetic and logic instructions.
- All arithmetic, logic, and data movement operations go through WREG.
- It’s the “central register” for the CPU. Used for Arithmetic/Logic.

MOVLW Instruction

- This instruction moves 8 bit data into the WREG register.
- **Format:** **MOVLW K** ; move literal value K into WREG

K is an 8 bit value that can range from 0-255 in decimal, or 00-FF in hex.

MOVLW 25H ; move value 25H into WREG (WREG=25H)

MOVLW 87H

MOVLW 15H ; loads the WREG with value 15H (15 in hex and 21 in decimal)

ADDLW Instruction

- **Format: ADDLW K** ; add literal value K to WREG

The result will also be back in the WREG.

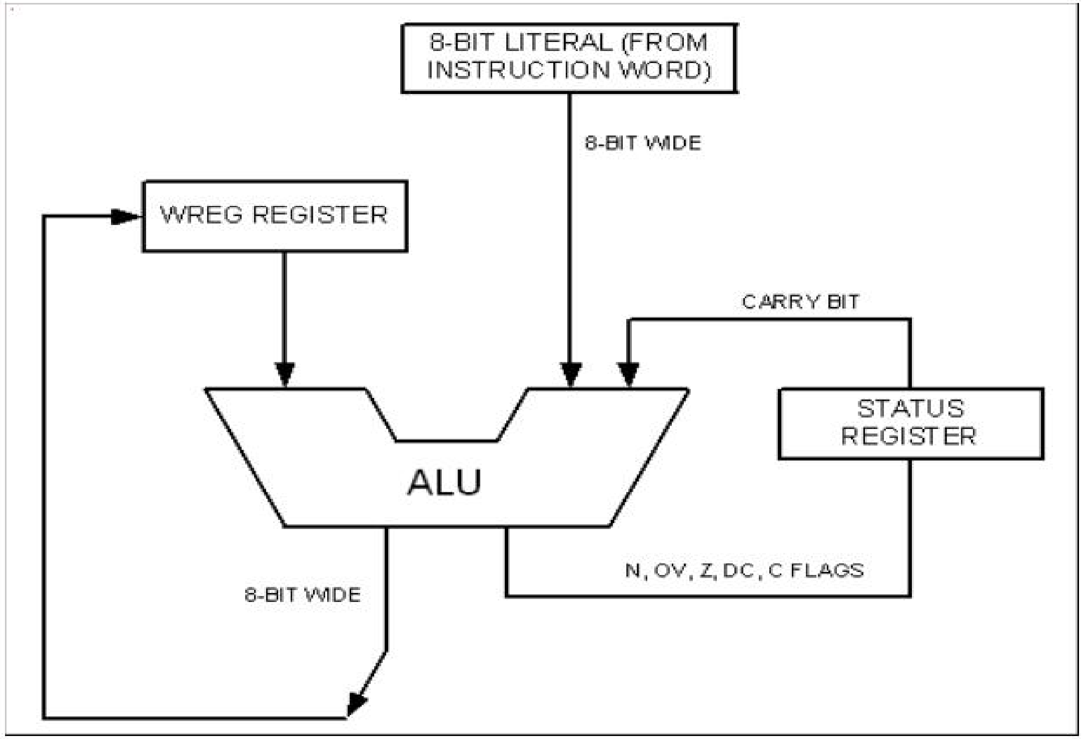
- To add the two numbers such that 25H and 34H:

MOVLW 25H ; load 25H into WREG

ADDLW 34H ; add value 34 to W (W=W+34H)

Executing the above lines results in WREG=59H (25H+34H=59H)

PIC WREG and ALU Using Literal Value



Question???

MOVLW 12H
ADDLW 16H
ADDLW 11H
ADDLW 43H

Answer?????

Precautions

- Moving a value larger than 255 (FF in hex) into the WREG register will truncate the upper byte and cause a warning in the .err file.

MOVLW 7F2H ; Illegal 7F2H > 8 bits (FFH) , becomes F2H

MOVLW 456H ; Illegal 456H > FFH, becomes 56H

MOVLW 60A5H ;Illegal but becomes A5H

- MOVLW 5H ; the result will be WREG = 05H ; that is WREG=00000101 in binary

Practice Questions

- Write instructions to move value 34H into the WREG register.
- Write instructions to add the values 16H and CDH. Place the result in the WREG register.

The PIC File Register

- In addition to WREG register, PIC has many other registers called **data memory space** to distinguish them from program (code) memory space.
- The data memory space is a read/write (static RAM) memory.
- The data memory is also called **file register**.
- It is used by the Processor for data storage and registers for internal use and functions.
- As with WREG, we can perform arithmetic and logic operations on many locations of the file register (data RAM).
- Size varies from 32 bytes to several thousand bytes.

File register (data RAM)

- In PIC, file register is divided into 2 sections

1. Special Function Registers (SFRs)-- hardware control registers
2. General Purpose Registers (GPRs) or general Purpose RAM (GP RAM)

Special Function Registers

- They are dedicated to specific functions sunch as ALU status, timers, serial communication, I/O ports, ADC and so on.
- They are used for control of the microcontroller or peripheral.
- PIC SFRs are 8-bit registers.
- The number of locations in the file register depends on the pin numbers and peripheral functions supported by that chip.

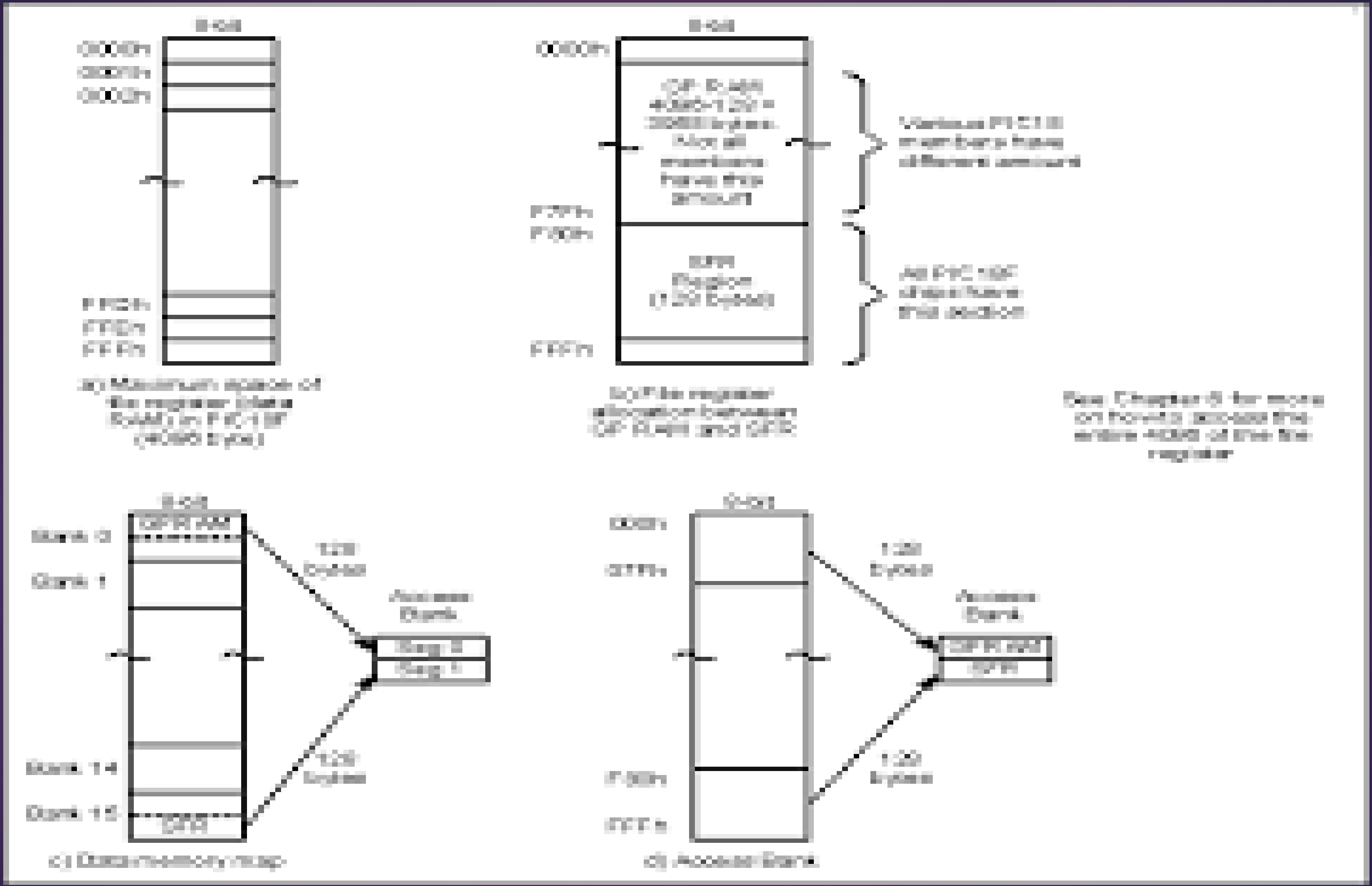
General-Purpose Registers

- They are a group of RAM locations in the file register that are used for data storage and scratch pad.
- A scratch pad is a small, fast storage area used to temporarily hold data or intermediate results while a program is running.
- Each location is 8 bits wide and can be used to store any data we want as long as it is 8-bit.
- The number of RAM locations in the file register for general purpose registers can vary from chip to chip.

File Register and access bank in PIC18

- The file register of PIC18 family can have a maximum of 4K (4096) bytes.
- With 4K Bytes, the file register has addresses of 000-FFFH.
- The file register is divided into 256 byte banks. So, we have up to a maximum of 16 banks (16 x 256=4096).
- Address range= 0 to (2^address bits -1)
- So, for PIC18, we have address range 0 x 000 to 0 x FFF.

- Every PIC 18 family member has at least one bank for the file register.
- This bank is called the **access bank** and is the default bank when we power up the PIC18 chip.
- See Figure 2-3 to examine the access bank.



Bank 0
Bank 1
Bank 14
Bank 15
Access Bank
c) Data memory map
d) Access Bank

- We are always working with file registers (RAM locations). The chip divides data memory into:
- SFRs and GPRs

- Because there is 4 KB of RAM but only 8 bits in the instruction to hold an address, Microchip uses **banking** to reach all memory.
- The Access Bank gives you direct access to 256 bytes of memory.
- The Access Bank is like a shortcut slice of memory:

  Top half = SFRs

  Bottom half = a small part of GPRs
- The rest of RAM is still there, but to use it you must use banked addressing.

Default Access Bank

- In PIC18 microcontrollers, the Access Bank is a 256-byte memory window that lets the CPU access certain registers directly without using a bank select instruction (BSR).

Why ???

- PIC18 has many banks of memory (large RAM space), and we must set BSR to select which bank to access.
- With Access Bank, we can always access the most commonly used SFRs and first 128 bytes of GPR directly.

Access Bank in PIC18 is:

- 128 bytes of GPR (0x00–0x7F)
- 128 bytes of SFR (0xF80–0xFFF)
- Physically, SFRs and RAM are not necessarily adjacent on silicon.
- They’re connected to the same address/data bus through a decoder.

- SFRs are not in RAM physically. They are memory-mapped registers
- They share the address space with RAM because it simplifies the MCU architecture.

- **Analogy:**

Your bedroom (RAM) is for storing stuff (data).

Your light switches (SFRs) are mounted on the wall but wired into your room layout so you can control the lights easily.

They're in the same house layout (address space).

Term	Meaning
Data memory	Memory used for runtime data storage (separate from program memory).
File register	Any byte in the data memory, whether it’s a GPR or an SFR.
GPR	File registers that are real RAM for variables.
SFR	File registers that control hardware.

In Short we can say,

- Data memory is made of file registers.
- Each byte in data memory is a file register, and depending on its address, it's either RAM (GPR) or a hardware control register (SFR).