

UNIT - I

1

Introduction to Microcontrollers

Syllabus

Introduction, Brief history of microcontrollers, Difference between microprocessor and microcontroller, Criteria for selection of microcontroller.

Contents

- 1.1 *Introduction*
- 1.2 *Brief History of Microcontrollers*
- 1.3 *Difference between Microprocessor and Microcontroller* **May-11, Dec.-13, April-12**
- 1.4 *Advantages and Disadvantages of Microcontrollers* **April-14**
- 1.5 *Applications of Microcontrollers*
- 1.6 *Criteria for Selection of Microcontroller* **April-12**

1.1 Introduction

- Fig. 1.1.1 shows the simplified block diagram of a microprocessor. As shown in the Fig. 1.1.1 it consists of an Arithmetic and Logic Unit (ALU), general purpose registers, Stack Pointer (SP), Program Counter (PC), clock timing circuit and interrupt circuit.
- To make a complete microcomputer system only microprocessor is not sufficient. It is necessary to add other peripherals such as Read Only Memory (ROM), read/write memory (RAM), decoders, drivers, number of input/output devices to make a complete microcomputer system.

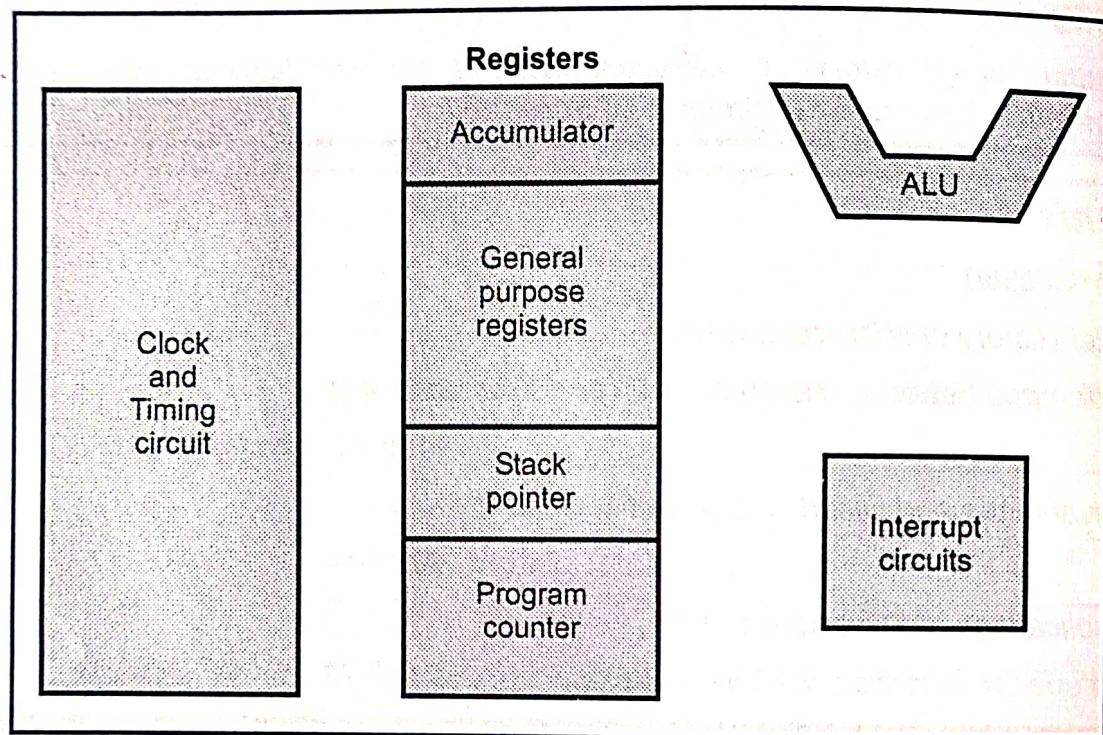


Fig. 1.1.1 Simplified block diagram of a microprocessor

- In addition, special purpose devices, such as interrupt controller, programmable timers, programmable I/O devices, DMA controllers may be added to improve the capacity and performance and flexibility of a microcomputer system.
- The key feature of microprocessor based computer system is that it is possible to design a system with a great flexibility. It is possible to configure a system as large system or small system by adding suitable peripherals.
- On the other hand, the microcontroller incorporates all the features that found in microprocessor. However, it has also added features to make a complete microcomputer system on its own.
- The microcontroller has built-in ROM, RAM, parallel I/O, serial I/O, counters and a clock circuit.

- Fig. 1.1.2 shows the simplified block diagram of a microcontroller. As shown in the Fig. 1.1.2, the microcontroller has on-chip (built-in) peripheral devices.

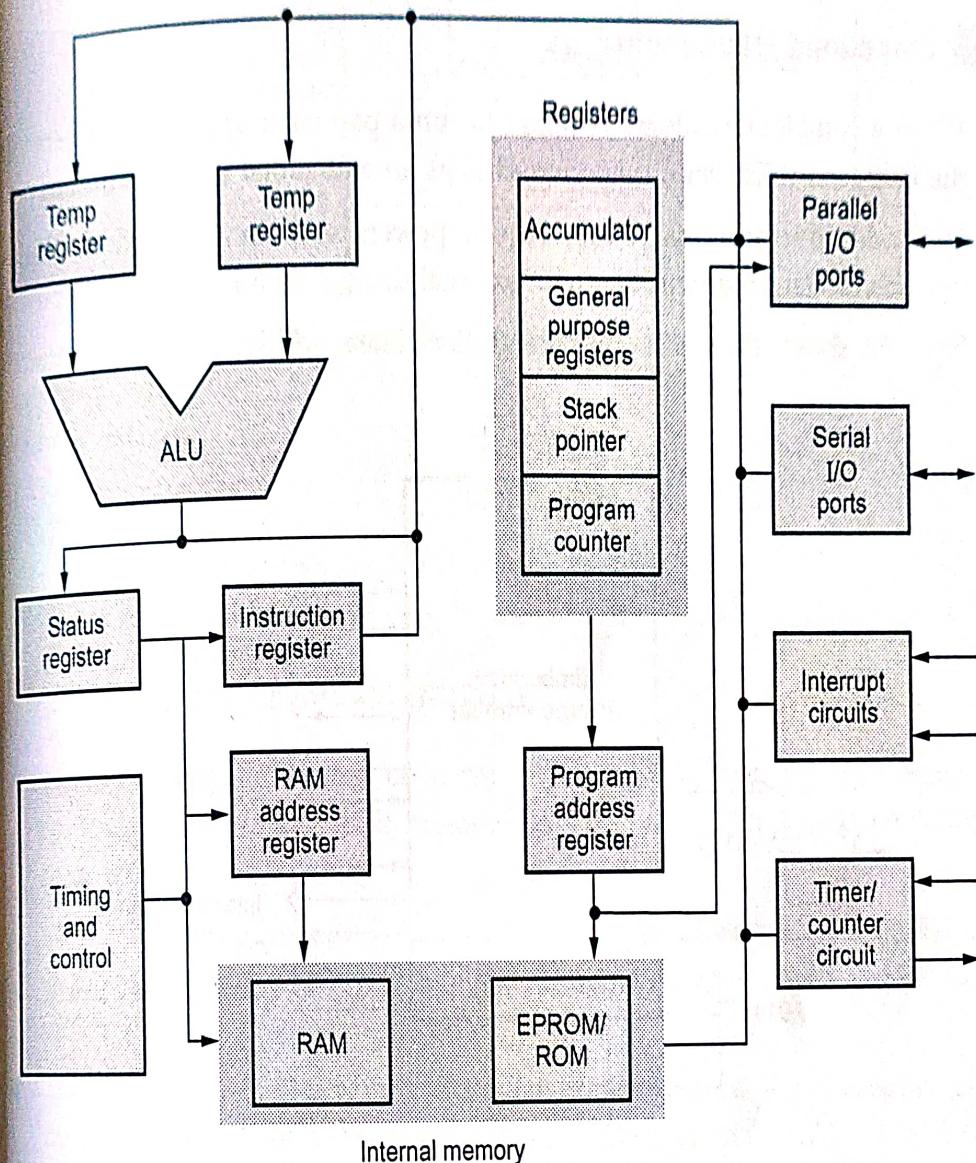


Fig. 1.1.2 Block diagram of microcontroller

- These on-chip peripherals make it possible to have single-chip microcomputer system.
- Most modern Microcontrollers might contain even more peripherals like SPI (Serial Peripheral Interface), I²C (Inter Integrated Circuit), ADC (Analog to Digital Converter), DAC (Digital to Analog Converter), CAN (Controlled Area Network), USB (Universal Serial Bus), and many more.
- The integration of features like ADC, DAC etc. on the same chip as the CPU makes it more efficient and cheaper than to use a separate ADC Chip.
- There are few more advantages of built-in peripherals :
 - Built-in peripherals have smaller access times hence speed is more.

- Hardware reduces due to single chip microcomputer system.
- Less hardware, reduces PCB size and increases reliability of the system.

1.1.1 Embedded Microcontrollers

- When a complete hardware required to run a particular application is provided on the microcontroller chip, it is referred to as an **embedded microcontroller**.
- Embedded microcontrollers only require power, reset circuit and clock. Embedded microcontrollers communicate with external devices with its digital I/O pins.
- Fig. 1.1.3 shows the typical microcontroller system with embedded microcontroller.

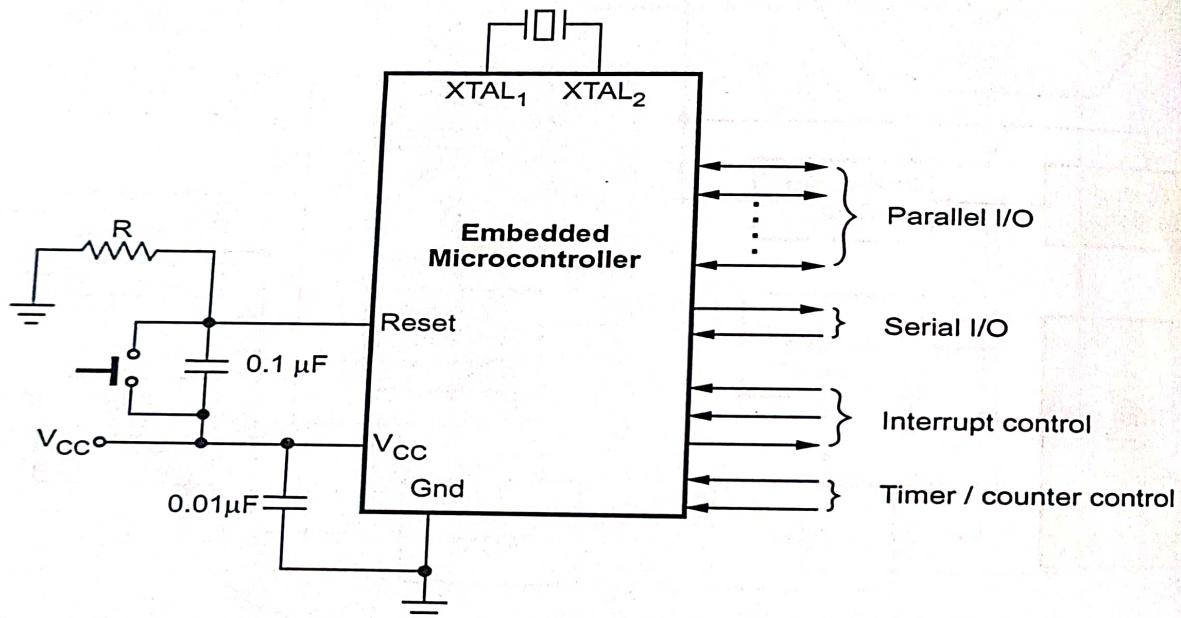


Fig. 1.1.3 Typical microcontroller system with embedded microcontroller

1.1.2 External Memory Microcontrollers

- Sometimes, for large systems, the built-in program memory and data memory are insufficient. To overcome this problem some microcontrollers allow the connection of external memory.
- For the connection of external memory some parallel port pins are used as address and data lines. Thus connecting external memory to the microcontroller reduces its parallel input/output capabilities.
- Fig. 1.1.4 shows the microcontroller with external memory connections. As shown in the Fig. 1.1.4, many times address and data lines are multiplexed and separated by external latch and ALE signal from the microcontroller.

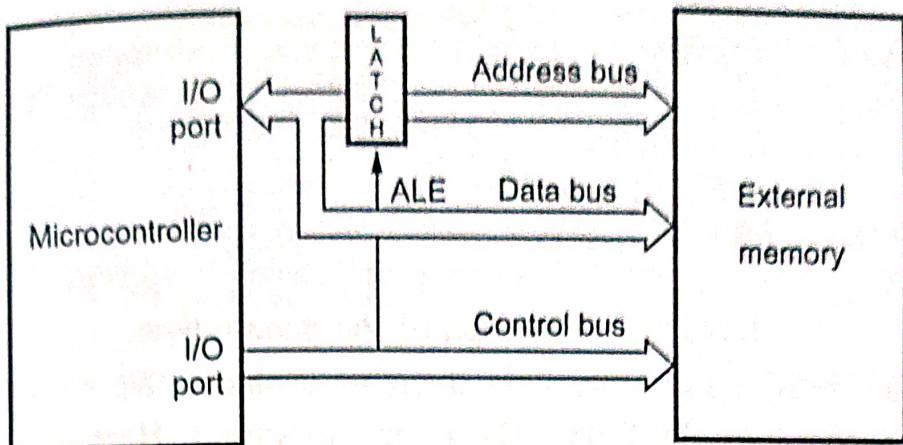


Fig. 1.1.4 Microcontroller with external memory connections

Review Questions

1. What is microprocessor ?
2. Draw and explain the simplified block diagram of microcontroller.
3. What do you mean by embedded microcontrollers and external memory microcontrollers ?

1.2 Brief History of Microcontrollers

- The first computer system on a chip optimized for control applications - micro controller was the Intel 8048 released in 1976, with both RAM and ROM on the same chip.
- Most microcontrollers at this time had two variants. One had an erasable EEPROM program memory, which was significantly more expensive than the PROM variant which was only programmable once.
- The Evolution of Microcontrollers has undergone a significant change. The development table of microcontrollers is as shown in Table 1.1.1.

Year	Microcontroller	Features
1976	Intel 8048	64 bytes RAM, 1 kB ROM and I/O ports.
1980	8031	128 bytes RAM, Two 16-bit timers, One serial port and I/O ports.
1980	Intel 8051	128 bytes RAM, 4 kB ROM, Two 16-bit timers, One serial port and I/O ports.
1984	Atmel 89C51	128 bytes RAM, 4 kB ROM, Two 16-bit timers, One serial port and I/O ports.
1985	Motorola 68HC11A8	256 bytes RAM, 8 kB ROM, 512 bytes EEPROM, 16-bit timer; 8 channel 8-bit A/D, SCI, SPI and I/O ports.

1996	ATtiny404 (AVR)	256 bytes RAM, 4 kB Flash memory, 128 bytes EEPROM, Two 16-bit Timers, One 16-bit Real-Time Counter (RTC), One USART, SPI, Configurable Custom Logic (CCL) and I/O ports.
1989	Microchip PIC18F458	1536 bytes RAM, 32 kB Flash memory, 256 bytes EEPROM, Four timers, ADC, USART and I/O ports.

Table 1.1.1 Evolution of Microcontrollers

- AVR is a family of microcontrollers developed since 1996 by Atmel, acquired by Microchip Technology in 2016. These are modified Harvard architecture 8-bit RISC single-chip microcontrollers.
- AVR was one of the first microcontroller families to use **on-chip flash memory** for program storage, as opposed to one-time programmable ROM, EPROM, or EEPROM used by other microcontrollers at the time.
- In 1989, Microchip Technology Corporation introduces an 8-bit microcontroller called PIC (Peripheral Interface Controller). These chips are designed with Harvard architecture, and are offered in various device families.
- This micro-controller had small amount of data RAM, a few hundred bytes of on-chip ROM for the program, one timer, and a few pins for I/O ports, all on a single chip. Some PIC Microcontroller families are 10xxx, 12xxx, 14xxx, 16xxx, 17xxx and 18xxx. They are all 8 bit processor means CPU can work on only 8 bit data at a time and data larger than 8 bit are broken into 8-bit pieces to be processed by the CPU.
- All current models use flash memory for program storage, and newer models allow the PIC to reprogram itself.
- **Program memory and data memory are separated.** Data memory is 8-bit, 16-bit and, in latest models, 32-bit wide.
- Program instructions vary in bit-count by family of PIC, and may be 12, 14, 16, or 24 bits long. The instruction set also varies by model, with more powerful chips adding instructions for digital signal processing functions.

Review Question

1. Write a note on brief history of microcontrollers.

1.3 Difference between Microprocessor and Microcontroller

SPPU : May-11, Dec.-13, April-12

- Fig. 1.3.1 shows comparison between microprocessor and microcontroller systems.

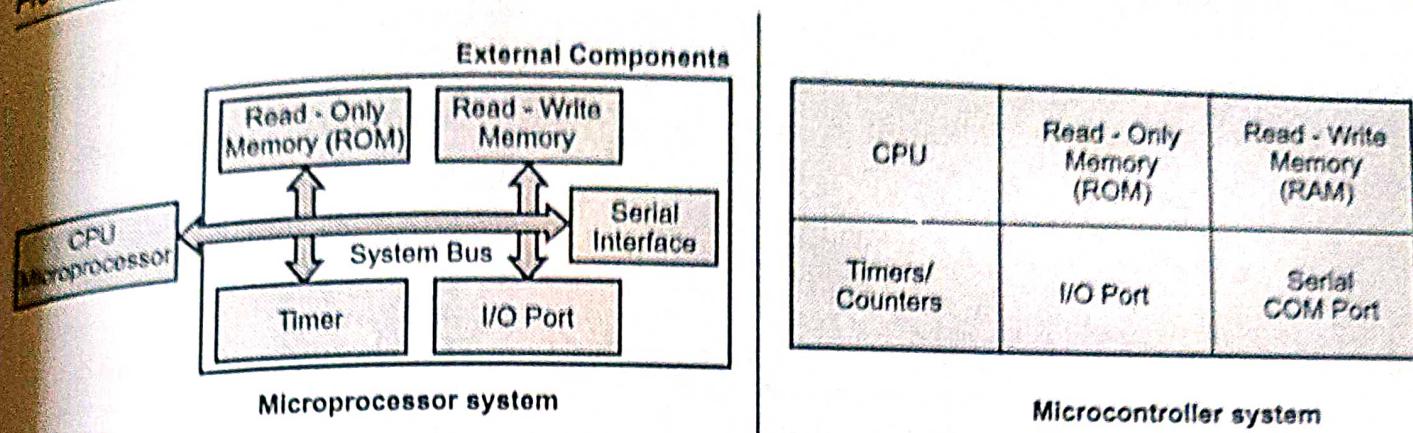


Fig. 1.3.1 Comparison between microprocessor and microcontroller systems

- Table 1.3.1 gives the comparison between microprocessor and microcontroller systems.

No.	Parameter	Microprocessor	Microcontroller
1	Structure	It has only the CPU on the chip. Other peripherals like I/O port, memory (RAM, ROM), timers and serial COM ports are connected externally.	CPU, Memory (RAM, ROM), I/O port timers and serial COM ports are connected on the single chip.
2	Architecture	Usually, CISC (Complex Instruction Set Computers). Architecture is based on Von Neumann model where data and instructions are stored in the same memory module.	May be CISC or RISC (Reduced Instruction Set Computers). Architecture is based on Harvard model where data and instructions are stored in the different memory modules.
3	Design Flexibility	High, users can decide the amount of memory, the number of I/O ports and other peripheral devices.	Less, once microcontroller is chosen user cannot change amount of memory, the number of I/O ports and other peripheral devices.
4	System Hardware	Requires more hardware components	Requires less hardware components.
5	Data bus width	Large up to 128 bits	Comparatively less up to 32-bits
6	Addressing capacity	Large	Comparatively less
7	Clock speed	The clock speed of the microprocessor is high. It is in terms of the GHz. It ranges between 1 GHz to 4 GHz.	The clock speed of the microcontroller is low. It is in terms of the MHz. it ranges between 1 MHz to 300 MHz.
8	I/O Interfacing	Needs external chip for interfacing I/O devices	Has in-built ports for interfacing I/O devices.
9	Bit level instructions	Less support	More support

11	Access time	High	Low
12	Power consumption	High	Low
13	Cost	High	Low
14	Examples	4004, 8085, 8086, 80386, 80486 and Pentium processors.	INTEL 8051, PIC, AVR family microcontrollers.
15	Applications	Designed for more generalized applications - Personal computers, laptops, etc. Hence called heart of computer.	In embedded devices such as digital camera, mobile phones, washing machine etc. Called heart of embedded systems.

Review Question

1. Differentiate between microprocessor and microcontrollers.

SPPU : May-11, Dec.-13, April-12

1.4 Advantages and Disadvantages of Microcontrollers

SPPU : April-14

1.4.1 Advantages of Microcontrollers

- The main advantages of microcontroller are :
 - A Microcontroller is a true device that fits the computer-on-a-chip idea.
 - The typical microcontroller is programmable, which means it is reusable. This is especially advantageous for prototyping control circuitry.
 - Consumes less power and hence suited for battery operated devices
 - No need for any external interfacing of basic components like Memory, I/O Ports, etc.
 - Microcontrollers don't require complex operating systems as all the instructions must be written and stored in the memory. (RTOS is an exception).
 - All the I/O Ports are programmable.
 - Integration of all the essential components reduces the cost, design time and area of the product (or application).

1.4.2 Disadvantages of Microcontrollers

- The main disadvantages of microcontroller are :
 - Microcontrollers are not known for their computation power.
 - The amount of memory limits the instructions that a microcontroller can execute.
 - Does not support complex instructions. Thus to implement complex operations such as multiplication requires multiple instructions.

Limitations of 8-bit microcontrollers

1. Limited Speed : Data bus width limits speed of execution.
2. Limited Addressing Capacity
3. Multiple instructions are needed to execute complex operations.
4. Limited size of ROM, RAM and Flash memory.
5. High power devices can not be interfaced directly.

Review Questions

1. State advantages and disadvantages of microcontroller.
2. Explain limitations of 8 bit microcontrollers.

SPPU : April-14

1.5 Applications of Microcontrollers

- There are a large number of applications of Microcontrollers. In fact, the entire embedded systems industry is dependent on Microcontrollers.
- The following are few applications of microcontrollers.
 1. Home appliances : Washing machines, refrigerators, microwave ovens, TVs, ACs etc.
 2. Electronic gadgets : Calculators, tablets, cameras, health bands, etc.
 3. Industry : Industrial controllers, data acquisition systems, etc.
 4. Computer accessories : Printers, Keyboards, etc.
 5. Communication systems : Modems, fax machines, intercoms, mobile phones etc.
 6. Automobiles : Engine control, speed sensed door locking systems, etc.
 7. Security systems : Smoke and Fire alarms, etc.
 8. Medical instruments : ECG, X-ray machines, etc.
 9. Military applications.
 10. Others : Traffic light control systems, flight control systems, electronic weighing machines, video games etc.

Review Question

1. List out the typical applications of microcontrollers.

1.6 Criteria for Selection of Microcontroller

SPPU : April-12

Criteria that designer should consider in choosing microcontrollers are :

1. It should satisfy the computing needs of the task efficiently and cost effective.
In analysing the needs of a microcontroller based system, we should consider:
 - Data bus size
 - Highest operating speed
 - Packing format
 - Power consumption
 - Amount of ROM, RAM, Flash RAM on the chip
 - On-chip peripherals such as timers, serial ports, ADC, DAC, etc.
 - Number of I/O pins
 - Cost per unit
2. Availability of software and hardware development tools such as compilers, assemblers and debuggers.
3. Availability in needed quantities both now and in the future.
4. Its ability to upgrade to higher-performance or lower power consumption versions.

Review Question

1. Explain the factors for selecting the microcontroller for the particular application.

SPPU : April-12

PIC18FXXX Microcontroller Architecture

Syllabus

PIC18FXXX : Features and architecture, comparison of PIC 18 series microcontrollers; PIC18F458/452 Pin out connection, Registers of PIC18F,

Program and data memory organization : The Program Counter and Programmable ROM space in the PIC, File register and Access bank, Bank switching in PIC18;

Addressing modes : Addressing modes with instruction example, Oscillator configurations, Reset operations, Brownout reset, Watchdog timer, Power down modes and Configuration registers.

Contents

2.1	Features of PIC18FXXX Series Microcontrollers	Dec.-12,13,15,16	Marks 6
2.2	Architecture of PIC18FXXX Series Microcontrollers	May-14,15,16,17	Marks 8
2.3	Comparison of PIC 18 Series Microcontrollers		
2.4	PIC18F458/452 Pin out Connection		
2.5	Registers of PIC18F	Dec.-12,13 May-13, 14, Aug.-14,15	Marks 4
2.6	Program and Data Memory Organization	May-12, 13, 15, Dec.-12,13, Aug.-14, 15,	Marks 8
2.7	File Register, Access Bank and Bank Switching in PIC18	Dec.-12,13, May-12,13,15,16,	Marks 8
2.8	Addressing Modes		
2.9	Oscillator Configurations	Aug.-14, May-19,	Marks 5
2.10	Reset Operations	Dec.-14,17,18, May-13,18, Aug.-15, 16,	Marks 8
2.11	Power Down Modes	Aug.-16, May-18, 19, Aug.-16,	Marks 9

2.1 Features of PIC18FXXX Series Microcontrollers

SPPU : Dec.-12,13,15,16

- Important Features of PIC18FXXX Series Microcontrollers are :
 1. Harvard architecture : PIC uses Harvard architecture. Harvard architecture has the program memory and data memory as separate memories and are accessed from separate buses. This improves bandwidth over traditional Von Neumann architecture in which program and data are fetched from the same memory using the same bus.
 2. High performance RISC processors
 - C-Language friendly architecture
 - PIC16 source code compatible
 - Linear program memory addressing to 2 Mbyte
 - Linear data memory addressing up to 4 Kbytes
 - Up to 10 MIPS operation
 - 16-bit wide instructions, 8-bit wide data path
 - Priority levels for interrupts
 - 8 x 8 Single cycle hardware multiplier
 3. Register file architecture : The register files/data memory can be directly or indirectly addressed. All special function registers, including the program counter, are mapped in the data memory.
 4. Instruction set simplicity
 - Reduced instruction set : The instruction set of PIC consists of only 77 instructions.
 - Orthogonal (symmetric) instructions : The instructions are orthogonal. It is possible to carry out any operation on any register using any addressing mode when the instructions are orthogonal.
 - Single word instructions
 5. FLASH program memory : In sizes from 8 to 128 Kbytes
 6. Data RAM : From 256 to 4 Kbytes
 7. Built-in power-on-reset : PIC has a built-in power-on-reset.
 8. Brown-out reset : PIC has a brown-out-reset. A brown-out-reset feature causes a reset of the PIC when the power supply voltage drops below 4 V or so.
 9. Watchdog Timer (WDT) : Watchdog timer is a special timer with specific function. It is used to prevent software crashes, i.e. endless loop.
 10. Power saving SLEEP mode : The PIC can put itself to sleep to save power during intervals when it has nothing to do. Thus, PIC supports a power saving SLEEP mode.

11. Flexible 8 - and 16-bit timers
12. Interrupt Control : The PIC can control up to 12 independent interrupt sources.
13. 10-bit ADC (Analog to digital converter)
14. Capture/Compare/PWM (CCP) modules : Up to five (CCP) modules.
15. Powerful output pin control
16. Built-in serial peripheral interface
17. EPROM/OTP/ROM Options
18. Low power consumption

Review Question

1. Explain the important features of PIC18FXXX series microcontrollers.

SPPU : Dec.-12,13,15,16 Marks 6

2.2 Architecture of PIC18FXXX Series Microcontrollers

SPPU : May-14,15,16,17

- Fig. 2.2.1 shows the simplified block diagram of PIC18 series microcontrollers.

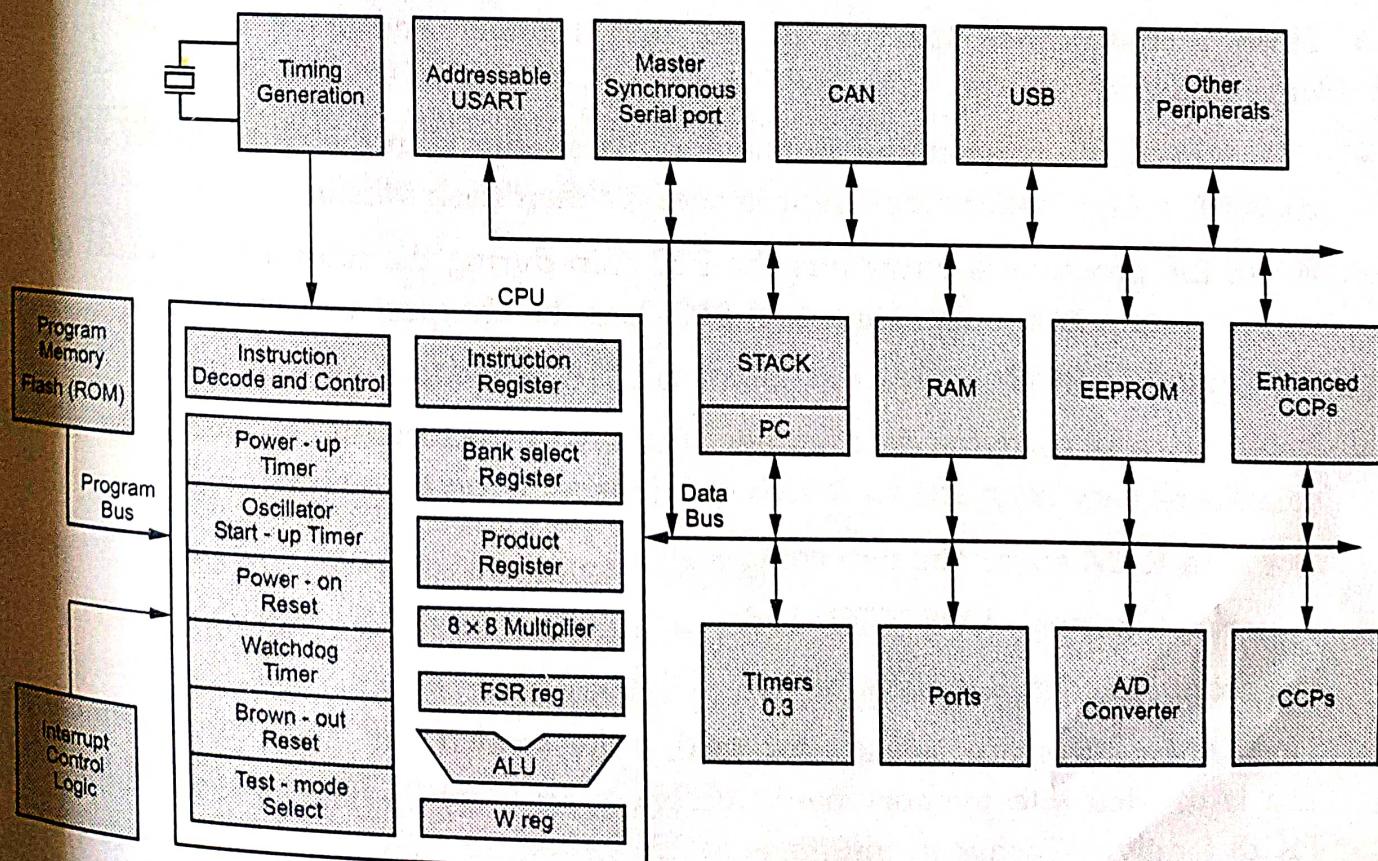


Fig. 2.2.1

Harvard architecture

- PIC18 uses Harvard architecture in which program and data are accessed from separate memories using separate buses.

ALU

- The PIC18 contains 8-bit ALU. It is capable of performing arithmetic operations such as add, subtract, multiply, shift and logical operations.

Multiplier 8 X 8 :

- An 8 x 8 hardware multiplier is included in the ALU of the PIC18 devices.
- By making the multiply a hardware operation, it completes in a single instruction cycle. This is an unsigned multiply that gives a 16-bit result. The result is stored in the 16-bit product register pair (PRODH : PRODL).

Program ROM

- ROM is used to store programs and hence the name. It is also known as code ROM. For PIC18 microcontroller family the size of program RAM vary from 4 to 128 K depending on the family. In PIC18 family microcontrollers, program ROM is available in different memory types :
 - Flash memory
 - OTP (One Time Programmable) memory
 - Masked memory
- Flash memory uses letter F in the part number and it is used for production development.
- OTP (One Time Programmable) memory uses letter C in the part number and is used for mass product because it is cheaper than flash version.

Data RAM and EEPROM

- RAM space is used for data storage. For PIC18 microcontroller family the size of data RAM vary from 256 to 4096 bytes depending on the family.
- The data RAM space has two components :
 - General Purpose RAM (GPR) and
 - Special Function Registers (SFRs)
- The RAM GPR space is used for read/write scratch pad and is divided into memory banks of 256 bytes. PIC18 family is always in ...

- The SFRs are used by the CPU and peripheral modules for controlling the desired operation of the device.
- The size of EEPROM on the chip vary with the family and it is used to store critical data that does not frequent access.

Interrupt Sources

- PIC18 devices support multiple interrupt sources.
- Each interrupt can be assigned with low/high priority.

I/O Pins

- In PIC18 microcontroller family I/O pins vary from 16 to 72 depending on the family.

Capture/Compare/PWM (CCP) Module

- The CCP (Capture/Compare/PWM) module contains a 16-bit register that can operate as a 16-bit capture register, as a 16-bit compare register or as a PWM duty cycle register.

Enhanced Capture/Compare/PWM (CCP) Module

- The operation of the ECCP module is identical to that of the CCP module. The ECCP module, on the other hand, has Enhanced PWM functionality and auto-shutdown capability.

CAN Module

- The Controller Area Network (CAN) module is a serial interface, useful for communicating with other peripherals or microcontroller devices.
- This interface/protocol was designed to allow communications within noisy environments.

ADC

ADC is 10-bit and the number of ADC channels in each chip varies from 5 to 16.

Timers

- PIC18 has 4/5 timers along with watch dog timer.

Master Synchronous Serial Port (MSSP) module

- The Master Synchronous Serial Port (MSSP) module is a serial interface useful for communicating with other peripheral or microcontroller devices. It has two modes of operation :
 - 3-wire SPI (Supports all 4 SPI modes)
 - I²C Master and Slave mode

Universal Synchronous Asynchronous Receiver Transmitter (USART) module

- The USART is a serial interface that can be configured in the following modes:
 - Asynchronous (full-duplex)
 - Synchronous - Master (half-duplex)
 - Synchronous - Slave (half-duplex)

Review Question

- Explain the architecture of PIC18FXXX with a suitable block diagram.

SPPU : May-14,15,16,17, Marks 8

2.3 Comparison of PIC 18 Series Microcontrollers

- Table 2.3.1 shows the comparison between some members of PIC 18 Family.

Device	Program memory		Data memory		I/O pins	ADC (10-bit) Channel	CCP/EC CP (PWM)	Timers 8/16 Bits	Number of Pins and package
	Flash (Bytes)	Single word Instructions	SRAM (Bytes)	EEPROM (Bytes)					
PIC18F1220	4 K	2048	256	256	16	7	0/1	1/3	18 DIP
PIC18F2420	16 K	8192	768	256	25	10	2/0	1/3	28 DIP
PIC18F2220	4 K	2048	512	256	25	10	2/0	2/3	28 DIP
PIC18F452	32 K	16384	1536	256	34	8	2/1	1/3	40 DIP
PIC18F4520	32 K	16384	1536	256	36	13	1/3	1/3	40 DIP
PIC18F458	32 K	16384	1536	256	34	8	1/3	1/3	40 DIP
PIC18F4580	32 K	16384	1536	256	36	11	1/3	1/3	40 DIP
PIC18F8722	128 K	65536	3936	1024	70	16	2/3	2/3	80 TQFP

Note : All above chips have USART/EUSART (enhanced USART) for serial communication

Table 2.3.1 Comparison between some members of PIC 18 Family

Review Question

- Compare various features of PIC 18 Family.

2.4 PIC18F458/452 Pin out Connection

- Fig. 2.4.1 shows the pin diagram of PIC18F458

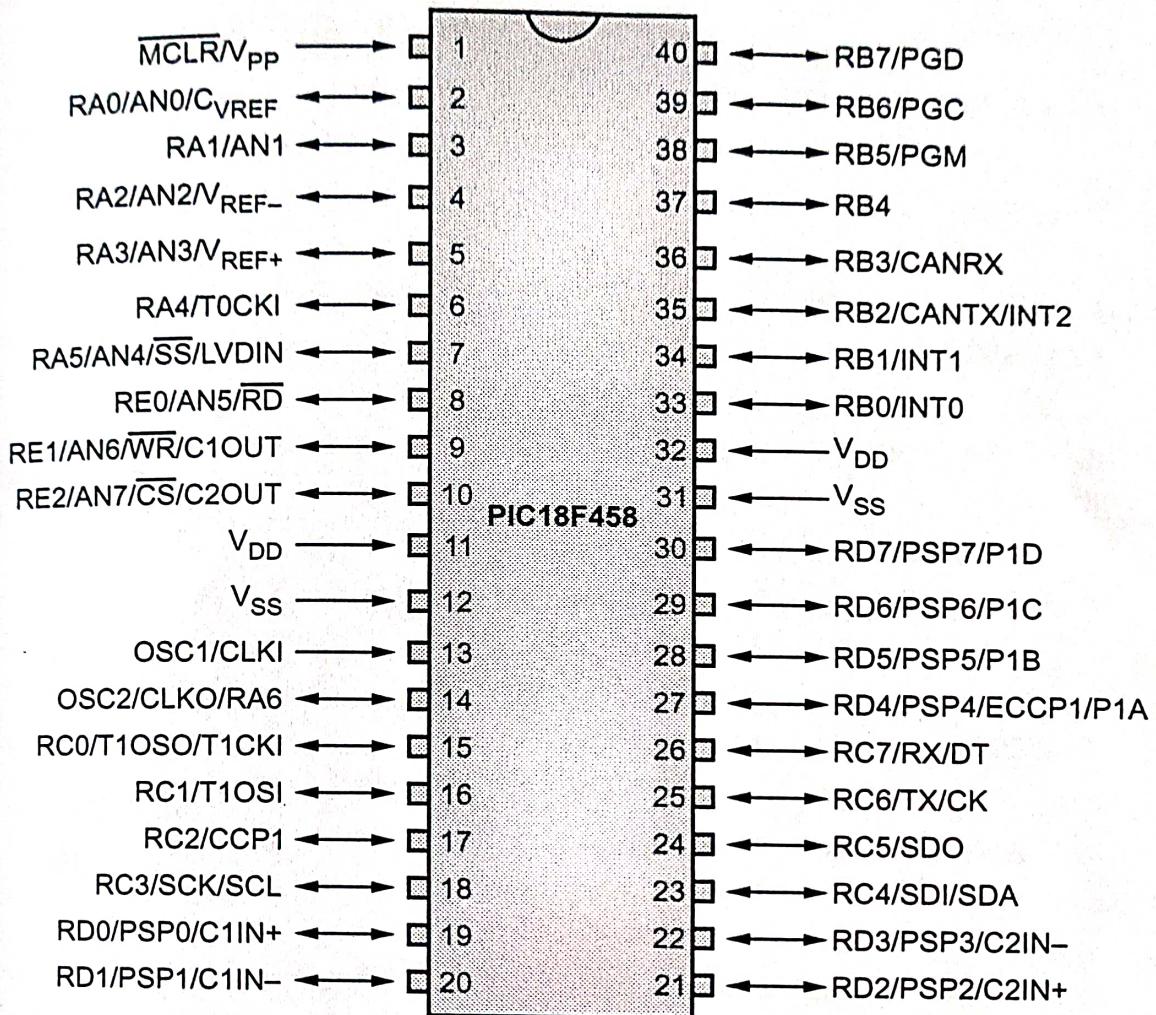


Fig. 2.4.1 PIC18F458 pin diagram

- As shown in the Fig. 2.4.1, PIC18F458 has total 40 pins. These pins can be classified as :
 - Port pins :** Out of 40 pins, 34 pins are dedicated to five ports A (RA0-RA6), B (RB0-RB7), C(RC0-RC7), D (RD0-RD7) and E (RE0-RE2), with alternate functions.
 - Oscillator pins :** OSC1 and OSC2. OSC2 and RA6 share the same pin.
 - Reset pin :** MCLR (Master clear)
 - Programming voltage input :** V_{PP}

- Power supply pins : Two V_{DD} and Two V_{SS} (GND).
- Table 2.4.1 shows PIC18F458 pins with description.

Pin No.	Pin name	Description
1	MCLR/VPP	$MCLR$: This pin is an active low Reset to the device, V_{PP} : Programming voltage input pin.
2	RA0/AN0/CVREF	RA0 : Digital I/O. AN0 : Analog input 0. CVREF : Comparator voltage reference output.
3	RA1/AN1	RA1 : Digital I/O. AN1 : Analog input 0.
4	RA2/AN2/ VREF?	RA2 : Digital I/O. AN2 : Analog input 0. VREF - : A/D reference voltage (Low) input.
5	RA3/AN3/ VREF+	RA3 : Digital I/O. AN3 : Analog input 0. V _{REF} + : A/D reference voltage (High) input.
6	RA4/T0CKI	RA4 : Digital I/O. T0CKI : Timer 0 external clock input.
7	RA5/AN4//LVDIN	RA5 : Digital I/O. AN4 : Analog input 0. SS / : Slave select input. LVDIN : Low-Voltage detect input.
8	RE0/AN5/	RE0 : Digital I/O. AN5 : Analog input 5. RD : Read control for Parallel Slave Port.
9	RE1/AN6//C1OUT	RE1 : Digital I/O. AN6 : Analog input 6. WR : Write control for Parallel Slave Port. C1OUT : Comparator 1 output
10	RE2/AN7//C2OUT	RE2 : Digital I/O. AN7 : Analog input 7. CS : Write control for Parallel Slave Port. C2OUT : Comparator 2 output
11, 32	V_{DD}	V_{DD} : Supply voltage, + 5 V.
12, 31	V_{SS} (GND)	V_{SS} (GND) : Ground reference
13	OSC1/CLK1	OSC1 : Oscillator crystal CLK1 : External clock input.

14	OSC2/CLK0/RA6	OSC2 : Oscillator crystal CLK0 : External clock input. RA6 : Digital I/O
15	RC0/T1OSO/T1CKI	RC0 : Digital I/O T1OSO : Timer1 oscillator output T1CKI : Timer1 external clock input
16	RC1/T1OSI	RC1 : Digital I/O T1OSI : Timer1 oscillator input.
17	RC2/CCP1	RC2 : Digital I/O CCP1 : Capture 1 input/Compare 1 output/PWM1 output
18	RC3/SCK/SCL	RC3 : Digital I/O SCK : Synchronous serial clock input/output for SPI mode SCL : Synchronous serial clock input/output for I2C mode
19	RD0/PSP0/C1IN +	RD0 : Digital I/O PSP0 : Parallel Slave Port data C1IN+ : Comparator 1 input
20	RD1/PSP1/C1IN -	RD1 : Digital I/O PSP1 : Parallel Slave Port data C1IN - : Comparator 1 input
21	RD2/PSP2/C2IN +	RD2 : Digital I/O PSP2 : Parallel Slave Port data C2IN + : Comparator 2 input
22	RD3/PSP3/C2IN -	RD3 : Digital I/O PSP3 : Parallel Slave Port data C2IN - : Comparator 2 input
23	RC4/SDI/SDA	RC4 : Digital I/O SDI : SPI data In. SDA : I ² C data I/O.
24	RC5/SDO	RC5 : Digital I/O SDO : SPI data Out.
25	RC6/TX/CK	RC6 : Digital I/O TX : USART asynchronous transmit. CK : USART synchronous clock.
26	RC7/RX/DT	RC7 : Digital I/O RX : USART asynchronous receive. DT : USART synchronous data.

14	OSC2/CLK0/RA6	OSC2 : Oscillator crystal CLK0 : External clock input. RA6 : Digital I/O
15	RC0/T1OSO/T1CKI	RC0 : Digital I/O T1OSO : Timer1 oscillator output T1CKI : Timer1 external clock input
16	RC1/T1OSI	RC1 : Digital I/O T1OSI : Timer1 oscillator input.
17	RC2/CCP1	RC2 : Digital I/O CCP1 : Capture 1 input/Compare 1 output/PWM1 output
18	RC3/SCK/SCL	RC3 : Digital I/O SCK : Synchronous serial clock input/output for SPI mode SCL : Synchronous serial clock input/output for I2C mode
19	RD0/PSP0/C1IN +	RD0 : Digital I/O PSP0 : Parallel Slave Port data C1IN+ : Comparator 1 input
20	RD1/PSP1/C1IN -	RD1 : Digital I/O PSP1 : Parallel Slave Port data C1IN - : Comparator 1 input
21	RD2/PSP2/C2IN +	RD2 : Digital I/O PSP2 : Parallel Slave Port data C2IN + : Comparator 2 input
22	RD3/PSP3/C2IN -	RD3 : Digital I/O PSP3 : Parallel Slave Port data C2IN - : Comparator 2 input
23	RC4/SDI/SDA	RC4 : Digital I/O SDI : SPI data In. SDA : I ² C data I/O.
24	RC5/SDO	RC5 : Digital I/O SDO : SPI data Out.
25	RC6/TX/CK	RC6 : Digital I/O TX : USART asynchronous transmit. CK : USART synchronous clock.
26	RC7/RX/DT	RC7 : Digital I/O RX : USART asynchronous receive. DT : USART synchronous data.

27	RD4/PSP4/ECCP1/ P1A	RD4 : Digital I/O PSP4 : Parallel slave port data. ECCP1 : ECCP1 capture/compare P1A : PWM output A
28	RD5/PSP5/P1B	RD5 : Digital I/O PSP5 : Parallel slave port data. P1B : PWM output B
29	RD6/PSP6/P1C	RD6 : Digital I/O PSP6 : Parallel slave port data. P1C : PWM output C
30	RD7/PSP7/P1D	RD7 : Digital I/O PSP7 : Parallel slave port data. P1D : PWM output D
33	RB0/INT0	RB0 : Digital I/O. INT0 : External interrupt 0.
34	RB1/INT1	RB1 : Digital I/O. INT1 : External interrupt 1.
35	RB2/CANTX/INT2	RB2 : Digital I/O. CANTX : Transmit signal for CAN bus. INT2 : External interrupt 2
36	RB3/CANRX	RB3 : Digital I/O. CANRX : Receive signal for CAN bus
37	RB4	RB4 : Digital I/O.
38	RB5/PGM	RB5 : Digital I/O. PGM : Interrupt-on-change pin. Low-voltage ICSP™ programming enable
39	RB6/PGC	RB6 : Digital I/O. PGC : Interrupt-on-change pin. ICSP programming clock
40	RB7/PGD	RB7 : Digital I/O. PGD : Interrupt-on-change pin. ICSP programming data

Table 2.4.1 PIC18F458 pins with description

Review Questions

1. Give the classification of PIC18F458 pins.
2. Give the alternate functions provided by Port A and Port E pins.
3. Give the alternate functions provided by Port B pins.
4. Give the alternate functions provided by Port C pins.

2.5 Registers of PIC18F

SPPU : Dec.-12,13 May-13,14, Aug.-14,15

- The memory of the PIC is divided into a series of registers. Each of the registers has its own address and memory locations.
- According to the type of working and usage, the registers in PIC are classified as :
- **Special Function Registers (SFRs)** : Used for control and status of the controller and peripheral functions
- **General Purpose Registers (GPRs)** : Used for data storage and scratchpad operations in the user's application.
- **WREG - working register** (acts as an Accumulator) - Used to perform arithmetic or logical functions.
- **Status register** that stores flags - Indicates the status of the operation done by ALU.
- **Registers - hold memory address**
 - **Bank Select Register (BSR)** : 4-bit register used in direct addressing the data memory.
 - **File Select Registers (FSRs)** : 16-bit registers used as memory pointers in indirect addressing data memory.
 - **Program Counter (PC)** : 21-bit register that holds the program memory address while executing programs. This means that the PIC18 family can access program addresses 000000 to 1FFFFFFH, a total of 2M bytes of code.
 - **Stack Pointer (SP)** : PIC18 has a 5-bit stack pointer. It is used to access the stack.

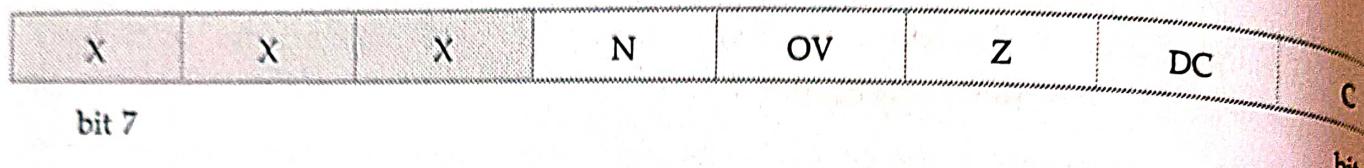
2.5.1 Working Register (WREG)

- It is 8-bits wide.
- It is used for ALU operations.
- The WREG register is the same as the accumulator in other microprocessors.
- The contents of WREG register are used for all arithmetic and logic one / two-operand instructions.
- It is not an addressable register.

2.5.2 PIC18 Status Register

- The status register is an 8-bit register. It is also referred to as **flag register**.
- The status register contains the arithmetic status of the ALU. It contains five flag bits Z, DC, C, OV and. These bits are set or cleared according to the result of arithmetic or logical instructions.

- Remaining three bits : 5, 6, and 7 are unimplemented and read as 0.
- CLRF STATUS instruction clears the upper three bits and set the Z bit. This leaves the Status register as 000u uuu (where u = Unchanged).
- Only BCF, BSF, SWAPF, MOVFF and MOVWF instructions are used to alter status register, because these instructions do not affect the Z, C, DC, OV or N from the status register.



C : Carry flag DC : Digital Carry flag OV : Overflow flag Z : Zero flag N : negative flag

Fig. 2.6.1 Status register

- **Carry (C) flag :** This flag is set if there is an overflow out of bit 7. The carry also serves as a borrow flag for subtraction.
- **Digital Carry (DC) flag :** This flag is set if there is an overflow out of bit 4, i.e., carry from lower nibble to higher nibble (bit 3 to bit 4). This flag is used for BCD operations and it is not available for the programmer. It is also known as Auxiliary Carry flag.
- **Zero (Z) flag :** The zero flag sets if the result of operation in ALU is zero and resets if result is non zero. The zero flag is also set if a certain register content becomes zero following an increment or decrement operation of that register.
- **Over (OV) flag :** This flag is set whenever the result of a signed number operation is too large, causing the high-order bit to overflow into the sign bit.
- **Negative (N) flag :** After the execution of arithmetic or logical operations, if bit 7 of the result is 1, the N flag is set indicating result is negative. If bit 7 is 0, the number will be considered as positive number.

2.5.3 Special Function Registers

- The Special Function Registers (SFRs) are registers used by the CPU and peripheral modules for controlling the desired operation of the device. These registers are implemented as static RAM.
- Table 2.5.1 shows the list of SFRs. The SFRs are classified into two sets :
 - Registers associated with the core function and
 - Registers associated with peripheral functions.
- Here, we will discuss the registers related to the core functions, while those related to the peripheral functions are discussed in the respective sections.

Address	Name	Address	Name	Address	Name	Address	Name
FFFh	TOSU	FDFh	INDF2 ⁽¹⁾	FBFh	CCPR1H	F9Fh	IPR1
FFEh	TOSH	FDEh	POSTINC2 ⁽¹⁾	FBEh	CCPR1L	F9Eh	PIR1
FFDh	TOSL	FDDh	POSTDEC2 ⁽¹⁾	FBDh	CCP1CON	F9Dh	PIE1
FFCh	STKPTR	FDCh	PREINC2 ⁽¹⁾	FBCh	ECCPR1H ⁽²⁾	F9Ch	-
FFBh	PCLATU	FDBh	PLUSW2 ⁽¹⁾	FBBh	ECCPR1L ⁽²⁾	F9Bh	-
FFAh	PCLATH	FDAh	FSR2H	FBAh	ECCP1CON ⁽²⁾	F9Ah	-
FF9h	PCL	FD9h	FSR2L	FB9h	-	F99h	-
FF8h	TBLPTRU	FD8h	STATUS	FB8h	-	F98h	-
FF7h	TBLPTRH	FD7h	TMR0H	FB7h	ECCP1DEL ⁽²⁾	F97h	-
FF6h	TBLPTRL	FD6h	TMR0L	FB6h	ECCPAS ⁽²⁾	F96h	TRISE ⁽²⁾
FF5h	TABLAT	FD5h	T0CON	FB5h	CVRCON ⁽²⁾	F95h	TRISD ⁽²⁾
FF4h	PRODH	FD4h	-	FB4h	CMCON ⁽²⁾	F94h	TRISC
FF3h	PRODL	FD3h	OSCCON	FB3h	TMR3H	F93h	TRISB
FF2h	INTCON	FD2h	LVDCON	FB2h	TMR3L	F92h	TRISA
FF1h	INTCON2	FD1h	WDTCON	FB1h	T3CON	F91h	-
FF0h	INTCON3	FD0h	RCON	FB0h	-	F90h	-
FEFh	INDF0 ⁽¹⁾	FCFh	TMR1H	FAFh	SPBRG	F8Fh	-
FEEh	POSTINCO ⁽¹⁾	FCEh	TMR1L	FAEh	RCREG	F8Eh	-
FEDh	POSTDEC0 ⁽¹⁾	FCDh	T1CON	FADh	TXREG	F8Dh	LATE ⁽²⁾
FECh	PREINC0 ⁽¹⁾	FCCh	TMR2	FACH	TXSTA	F8Ch	LATD ⁽²⁾
FEBh	PLUSW0 ⁽¹⁾	FCBh	PR2	FABh	RCSTA	F8Bh	LATC
FEAh	FSR0H	FCAh	T2CON	FAAh	-	F8Ah	LATB
FE9h	FSR0L	FC9h	SSPBUF	FA9h	EEADR	F89h	LATA
FE8h	WREG	FC8h	SSPADD	FA8h	EEDATA	F88h	-
FE7h	INDF1 ⁽¹⁾	FC7h	SSPSTAT	FA7h	EECON2	F87h	-
FE6h	POSTINC1 ⁽¹⁾	FC6h	SSPCON1	FA6h	EECON1	F86h	-
FE5h	POSTDEC1 ⁽¹⁾	FC5h	SSPCON2	FA5h	IPR3	F85h	-
FE4h	PREINC1 ⁽¹⁾	FC4h	ADRESH	FA4h	PIR3	F84h	PORTE ⁽²⁾
FE3h	PLUSW1 ⁽¹⁾	FC3h	ADRESL	FA3h	PIE3	F83h	PORTD ⁽²⁾
FE2h	FSR1H	FC2h	ADCON0	FA2h	IPR2	F82h	PORTC
FE1h	FSR1L	FC1h	ADCON1	FA1h	PIR2	F81h	PORTB
FE0h	BSR	FC0h	-	FA0h	PIE2	F80h	PORTA

1. This is not a physical register.
2. These registers are not implemented on the PIC18F248 and PIC18F258.

Table 2.5.1 Special Function Register map for PIC18FXX8 family

Review Questions

1. Write a note on PIC18 registers.
2. What is the function of WREG register in PIC Microcontroller ? **SPPU : Aug.-14,15 Marks 3**

3. Explain the status register of PIC18. **SPPU : Dec.-12,13 May-13, 14, Aug.-14,15 Marks 4**
4. What are SFRs ?

2.6 Program and Data Memory Organization

SPPU : May-12, 13, 15, Dec.-12,13, Aug.-14, 15

2.6.1 The Program Counter in the PIC18

- A program counter is a register in a CPU that contains the address (location) of the instruction being executed at the current time. After each instruction is fetched the program counter is incremented to point to the next instruction in the sequence.
- The PC can be accessed/modified by jump and branch instructions. Therefore, the destination address can be loaded to the program counter via branch instructions.
- The width of the program counter decides how many memory locations CPU can access. More the width, more the memory locations CPU can access.
- Table 2.6.1 shows the width of program counter in various PIC families and code memory access by them.

PIC family	PC width	Memory address range	Code memory
PIC12F	12 bit	000 - FFFFH	4 KBytes
PIC16F	14-bit	0000 – 3FFFFH	16 KBytes
PIC18F	21-bit	000000 - 1FFFFFFH	2 MBytes

Table 2.6.1

- Fig. 2.6.1 shows the 21 bit program counter in PIC18 Family. The first location of program memory in PIC has the address of 00000H; however the last location can be different depending on the size of the ROM on the chip.

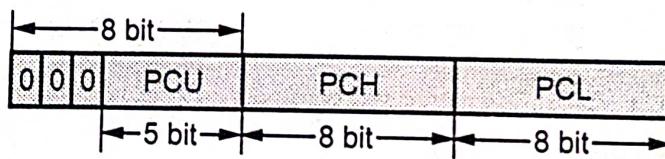


Fig. 2.6.1 Bit program counter of PIC18 family

- Program Counter (PC) - 21-bit register that holds the program memory address while executing programs. This means that the PIC18 family can access program addresses 000000 to 1FFFFFFH, a total of 2M bytes of code.

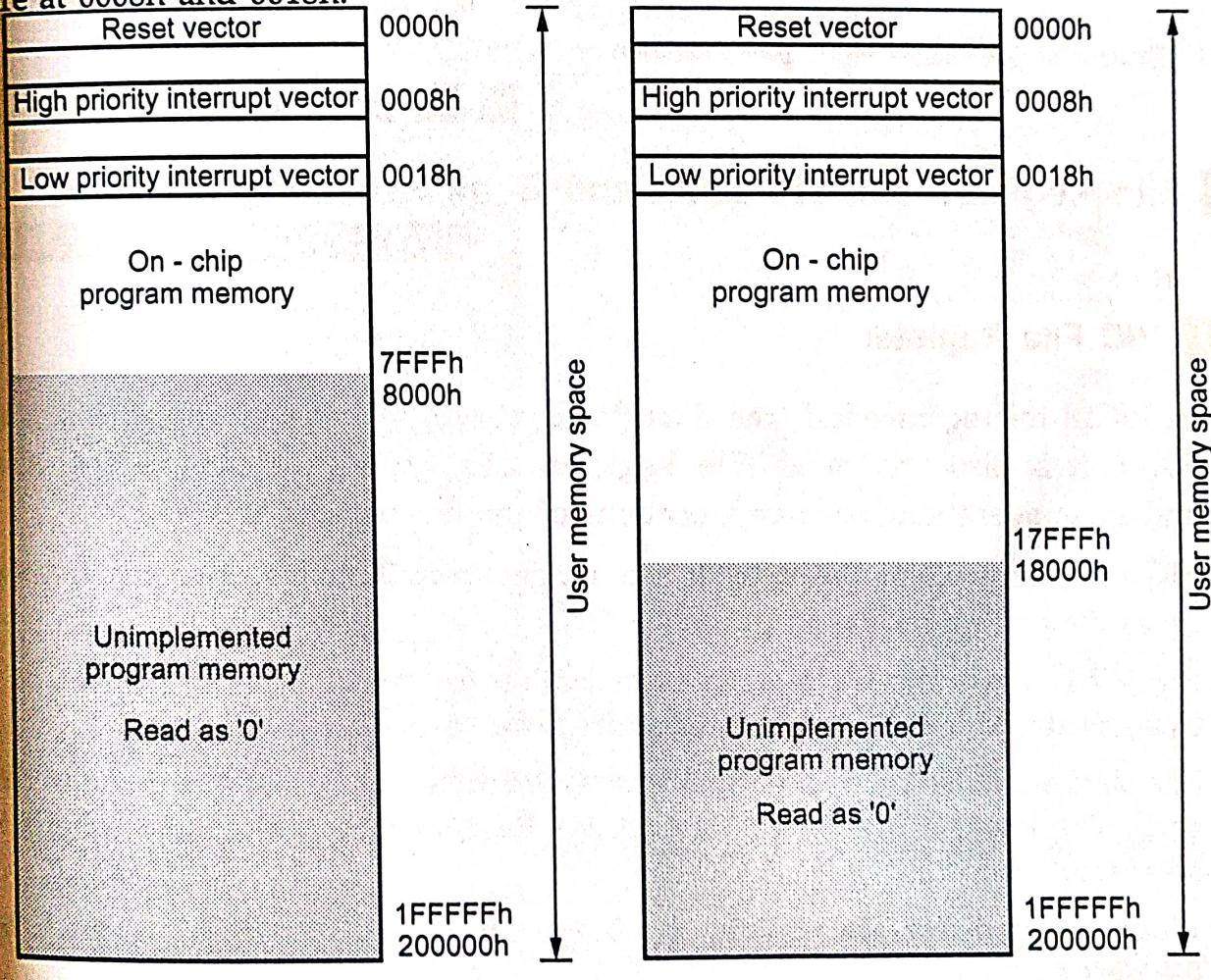
2.6.2 Programmable ROM Space In the PIC18 Family

Table 2.6.2 shows the code memory (ROM) size for various PIC18 family members.

PIC Family	Code (ROM) Memory	Address Range
PIC18F2220	4 kBytes	00000 - 00FFFFH
PIC18F2410	16 kBytes	00000 - 03FFFFH
PIC18F458	32 kBytes	00000 - 07FFFFH
PIC18F6680	64 kBytes	00000 - 0FFFFFH
PIC18F8722	128 kBytes	00000 - 1FFFFFH

Table 2.6.2

- Fig. 2.6.2 shows program ROM space for PIC18F458 and PIC18F8722. As shown in Fig. 2.6.2, the Reset vector address is at 0000h and the interrupt vector addresses are at 0008h and 0018h.



PIC18F458 Program ROM space

PIC18F8722 Program ROM space

Fig. 2.6.2 PIC18F Programmable ROM Space

- The PIC18F458 has 32 kbytes of flash program memory. This translates into 32768 single-word instructions, which can be stored in the program memory.
- On the other hand, the PIC18F8722 has 128 kbytes of Flash program memory. This translates into 131072 single-word instructions, which can be stored in the program memory.

- Accessing a location between the physically implemented memory and 2-Mbyte address will cause a read of all '0's (a NOP instruction).
- For the PIC18, the internal data bus between the code ROM and CPU is 16 b. Therefore, for PIC18F458, 32 kbyte code memory is presented as 16K x memory. As PIC18 family microcontrollers access 16 bits from a memory at given time, all instructions of PIC18 are either 2 bytes or 4 bytes.

Review Questions

- What is program counter ?
- Write a note on PIC18 Program Counter.
- Explain the programmable ROM space in PIC18.
- Draw and explain programmable memory map of PIC microcontroller

SPPU : May-12, Aug.-14, 15, Marks 5

- Explain programmable memory organization of PIC18FXXX microcontroller

SPPU : Dec.-12,13 May 13, 15, Marks 8

2.7 File Register, Access Bank and Bank Switching in PIC18

SPPU : Dec.-12,13, May-12,13,15,16,

2.7.1 PIC File Register

- In PIC18 microcontrollers, the data (Read/Write) memory is implemented as static RAM. It is also known as File Register. Like WREG we can perform arithmetic and logic operations on many locations of the file register data RAM.
- PIC microcontroller's file register size ranges from 32 bytes to several thousands of bytes depending on the chip.
- Fig. 2.7.1 shows the file register organization for the PIC18FXX8 devices. It is 4096 bytes of data memory with 12-bit address for each register.
- The data memory map is divided into as many as 16 banks that contain 256 bytes each. The lower 4 bits of the Bank Select Register (BSR) select which bank will be accessed.
- When using direct addressing, the Bank Select Register (BSR) is used to select the desired bank.
- BSR<3:0> holds the upper 4 bits of the 12-bit RAM address. These 4 bits are used to select the desired bank. The BSR<7:4> bits will always read '0's and writes will have no effect.
- The file register data memory of PIC is divided into two sections :
 - Special Function Registers (SFRs)** : Used for control and status of the controller and peripheral functions

- General Purpose Registers (GPRs) : Used for data storage and scratchpad operations in the user's application.

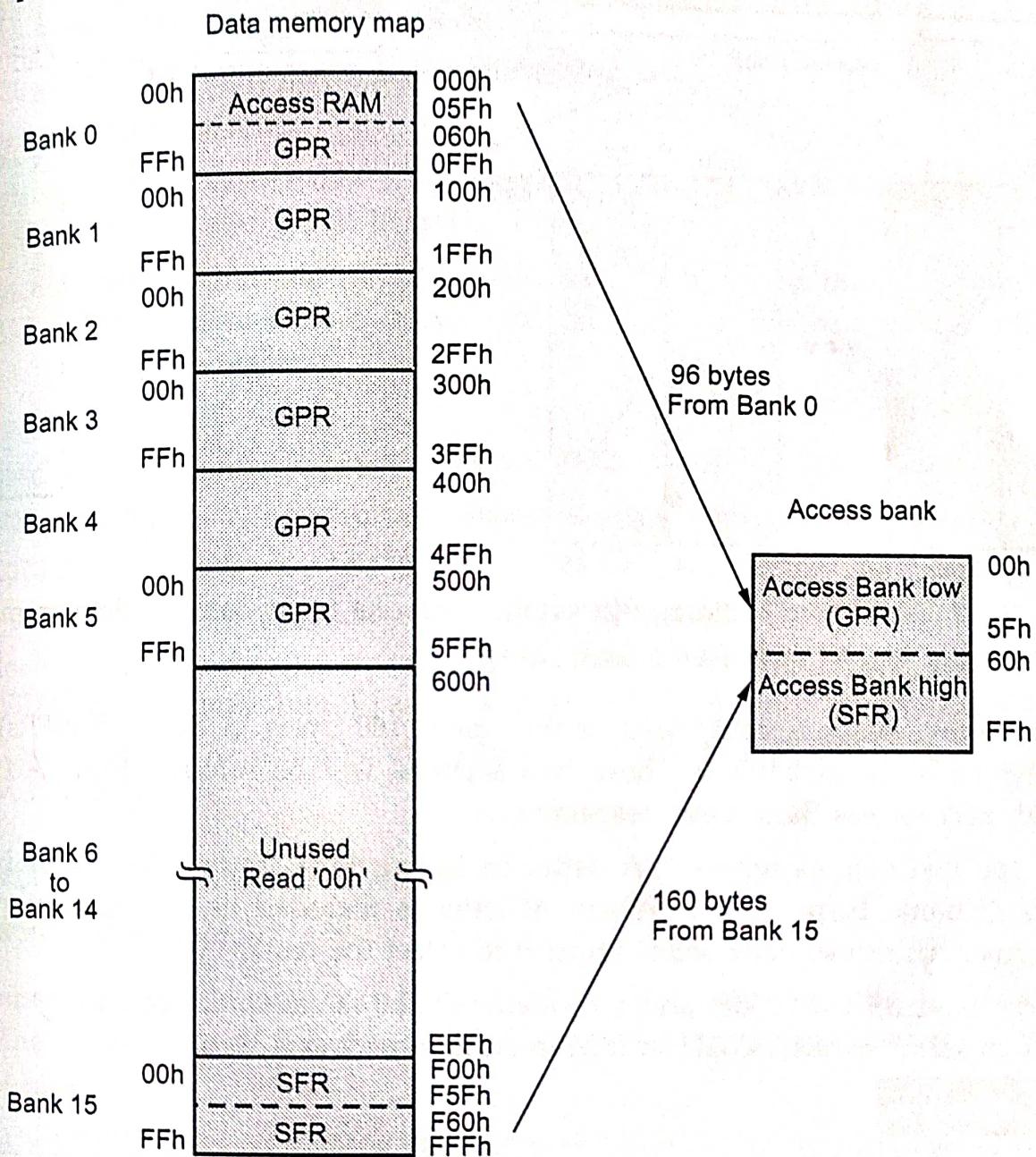


Fig. 2.7.1 File register data memory for PIC18F258/458

- The SFRs start at the last location of Bank 15 (FFFh) and grow downwards. GPRs start at the first location of Bank 0 and grow upwards.
- The entire data memory may be accessed directly or indirectly. Direct addressing may require the use of the BSR register. Indirect addressing requires the use of the File Select Register (FSR). Each FSR holds a 12-bit address value that can be used to access any location in the data memory map without banking. Fig 2.7.2 demonstrates direct and indirect addressing methods.
- File Select Registers :** FSR0, FSR1 and FSR2. These registers are composed of two 8-bit registers : FSRH and FSRL

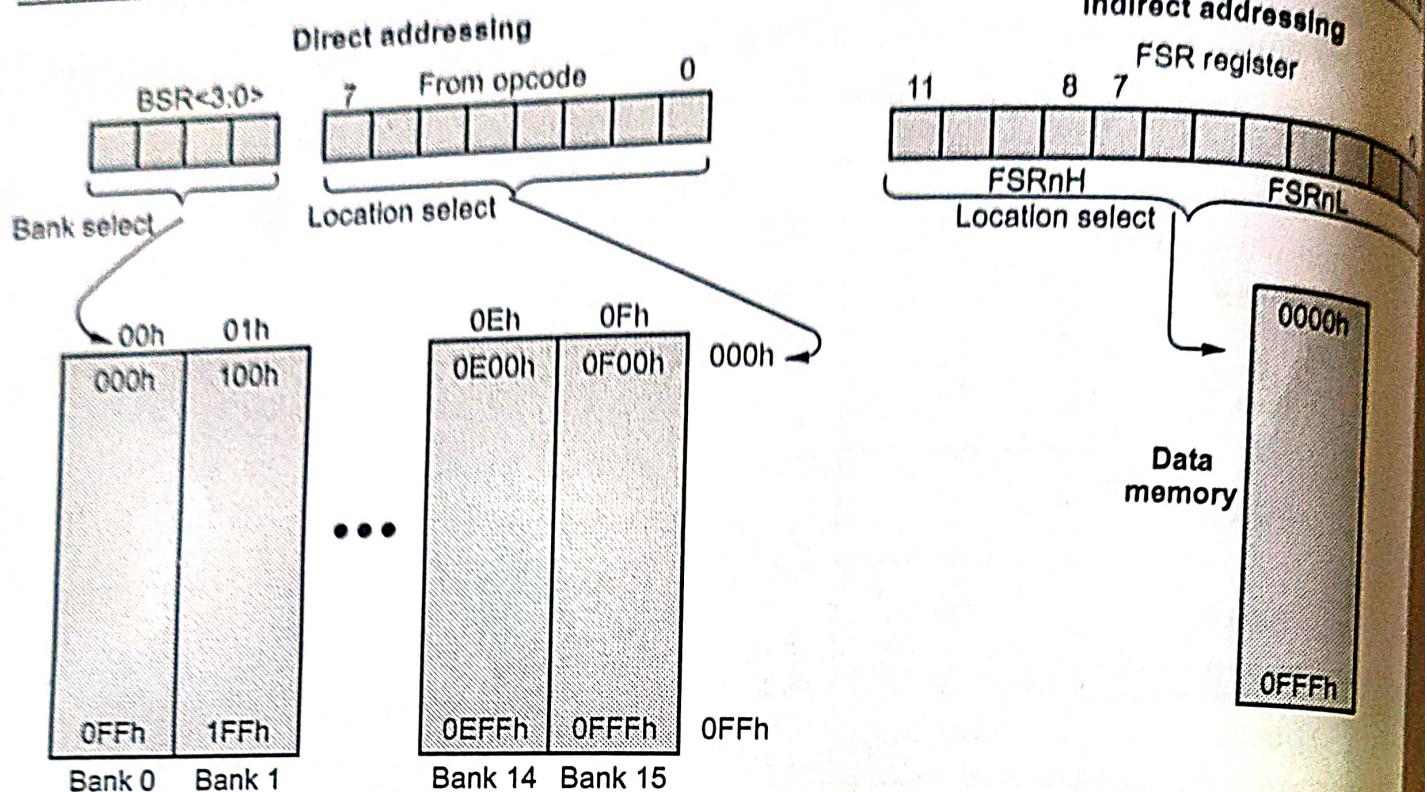


Fig. 2.7.2 Direct and indirect addressing methods for accessing data memory

2.7.2 Access Bank and Bank Switching

- The Access Bank is comprised of the upper 160 bytes in Bank 15 (SFRs) and the lower 96 bytes in Bank 0. These two sections will be referred to as Access Bank High and Access Bank Low, respectively.
- All the instructions without 'A' letter in instructions assume the access bank as default bank. Here, A = 0. When 'A' letter is included in the instruction, A is 1 and the instruction uses bank select register to select the bank.
- Upon reset BSR = 0000H and by default Bank0 is selected. We can load 0001H in BSR to select Bank1, 0002H in BSR to select Bank2 and so on.

Review Questions

- What is File Register in PIC18 ?
- Explain the File Register data memory of PIC18F458.
- Explain programmable memory organization of PIC18FXXX microcontroller.

SPPU : Dec.-12,13, May-12,13,15,16, Marks 8

- What is access bank ?
- Explain the Direct and indirect addressing methods for accessing data memory.
- What is BSR ? Explain its use.

2.8 Addressing Modes

- Part of the programming flexibility is the different kind of ways the programmer can refer to data stored in the memory. The different ways that can access data are referred to as **addressing modes**.

- The PIC 18 microcontroller supports following addressing modes :
 1. Immediate addressing mode
 2. Direct addressing mode
 3. Register indirect addressing mode
 4. Register Indirect/Indexed ROM addressing mode

1. Immediate addressing mode

- In immediate addressing mode, the immediate data is specified in the instruction. Immediate data is also called literal in PIC18.
- The immediate addressing mode is used to load the data into PIC registers and WREG register. However, we cannot use immediate addressing mode to load data into any of the file register.
- Examples :
 1. MOVLW 0x50 ; Load 50 H into WREG
 2. ANDLW D '18' ; Logically AND WREG with 18 decimal
 3. ADDLW B' 01100000' ; Add 60 H in WREG
- The letter 'L' in the instruction means literal (immediate).
- To load immediate data in file register data RAM, we have to load immediate data in WREG register and then it can be copied to file register data RAM location.
- We can use the EQU directive to access immediate data as shown below.
 NUM EQU 0x50
 ...
 MOVLW NUM ; Load 50H in WREG

2. Direct addressing mode

In direct addressing mode, the operand data is in the file register (RAM) data memory. The address of this memory location is specified in the instruction.

- Examples :
 1. MOVWF 0X10 ; Copy contents of WREG into File Register RAM location 10 H
 2. MOVFF 0X10, 0X20 ; Copy contents of location 10 H to 20 H
 3. MOVFF 0X30, PORTC ; Copy contents of location 30 H to PORTC
- The letter 'F' in the instruction means the address of the file register location.
- The letter 'FF' in the instruction means the both source and destination addresses are file register locations.

3. Register indirect addressing mode

- Register indirect addressing mode is used for accessing data stored in the file register (RAM) data memory.
- In this addressing mode, three file select registers - FSR0, FSR1, and FSR2 are used as pointers to the memory locations of the file register (RAM) data memory.

- Each of the FSR0, FSR1, and FSR2 registers has an INDF register associated with it, and these are called INDF0, INDF1, and INDF2. When we move data in INDFx we are moving data into a RAM location pointed to by the FSRx.
- Examples :
 - 1. LFSR 1, 0x55 ; Load FSR1 with 55H
 - MOVWF INDF1 ; copy contents of WREG into RAM location whose address is held by FSR1 register, i.e. 55H
 - 2. LFSR 0, 0x40 ; Load FSR0 with 40H
 - MOVF INDF0,W ; Copy the contents of RAM location pointed by FSR0 to WREG

4. Register Indirect/Indexed ROM addressing mode

- We use code space to store fixed data along with the code. This addressing mode is used for accessing the fixed data from look up tables that reside in the PIC program ROM. This process is often called **table processing**.
- Directive DB is used to define an 8-bit fixed data in data ROM. Here, we have use special function register to point the data to be fetched from the code space.
- Two registers are used in the table processing : TBLPTR (TaBLE PoinTR) and TABLAT (TABLE LATch).
- The 21 bit register TBLPTR is used to access byte from PIC18 program ROM. With 21 bit register TBLPTR we can cover the entire 2M program (code) space of PIC18.
- TBLPTR register is divided into three parts : TBLPTRL (Low), TBLPTRH (High) and TBLPTRU (Upper). These parts are of the SFRs.
- TABLAT register is used for keeping the byte read from code space pointed by TBLPTR.

• Example :

```

MOVlw 0x0          ; WREG = 0 Look-up Table low-byte address
MOVwf TBLPTRL     ; Load Look-up Table low-byte address in TBLPTRL register
MOVLW 0x10         ; WREG = 10 Look-up Table high-byte address
MOVwf TBLPTRH     ; Load Look-up Table high-byte address in TBLPTRH register
MOVLW 0x0          ; WREG = 0 Look-up Table upper-byte address
MOVwf TBLPTRU     ; Load Look-up Table upper-byte address in TBLPTRU register
TBLRD*            ; Read byte pointed by TBLPTR
MOVff TABLAT, PORTB ; Send the read byte to PORTB
  
```

Review Question

1. Explain the addressing modes supported by PIC18 with the help of suitable examples.

2.9 Oscillator Configurations

- The PIC18FXX8 can be operated in one of eight oscillator modes, programmable by three configuration bits : (FOSC2, FOSC1 and FOSC0).

SPPU : Aug.-14, May-19

No.	Configuration Bits			Oscillator Mode
	FOSC2	FOSC1	FOSC0	
1	0	0	0	LP Low-Power Crystal
2	0	0	1	XT Crystal/Resonator
3	0	1	0	HS High-Speed Crystal/Resonator
4	0	1	1	HS4 High-Speed Crystal/Resonator with PLL enabled
5	1	0	0	RC External Resistor/Capacitor
6	1	0	1	RCIO External Resistor/Capacitor with I/O pin enabled
7	1	1	0	EC External Clock
8	1	1	1	ECIO External Clock with I/O pin enabled

Table 2.9.1 Oscillator Configurations

2.9.1 Crystal Oscillator/Ceramic Resonators

- In XT, LP, HS or HS4 oscillator modes, a crystal or ceramic resonator is connected to the OSC1 and OSC2 pins to establish oscillation, as shown in Fig. 2.9.1. For these configurations, we can also connect an external clock source to OSC1 pin, as shown in Fig. 2.9.1.
- R_s is optional. It R_s may be required in HS mode, as well as XT mode, to avoid overdriving crystals with low drive level specification.

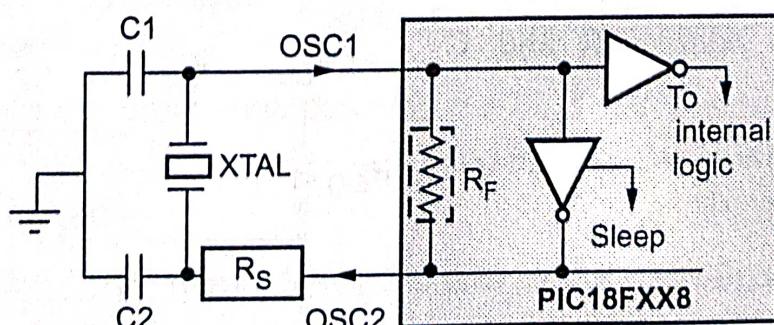


Fig. 2.9.1 Crystal/ceramic resonator operation

- The PIC18FXX8 oscillator design requires the use of a parallel cut crystal. Table 2.9.2 shows the capacitor selection for crystal oscillator.

OSC type	Crystal freq.	Cap. range C1	Cap. range C2
LP	32.0 kHz	33 pF	33 pF
	200 kHz	15 pF	15 pF
XT	200 kHz	47-68 pF	47 - 68 pF
	1.0 MHz	15 pF	15 pF
HS	4.0 MHz	15 pF	15 pF
	8.0 MHz	15 - 33 pF	15 - 33 pF
	20.0 MHz	15 - 33 pF	15 - 33 pF
	25.0 MHz	15 - 33 pF	15 - 33 pF

Table 2.9.2 Capacitor selection for crystal oscillator

2.9.2 RC Oscillator

- We always choose crystal oscillator when timing accuracy is required. However for timing insensitive applications, the "RC" and "RCIO" device options, we can choose RC oscillator which offers additional cost savings.
- The RC oscillator frequency is a function of the supply voltage, the resistor (R_{EXT}) and capacitor (C_{EXT}) values and the operating temperature.
- In addition to this, the RC oscillator frequency will vary from unit to unit due to
 - Normal process parameter variation,
 - Difference in lead frame capacitance between package types, especially for low C_{EXT} values.
 - Tolerance of external R and C components
- Fig. 2.9.2 shows RC Oscillator connections with PIC18.
- In the RC Oscillator mode, the oscillator frequency divided by 4 is available on the OSC2 pin. This signal may be used for test purposes or to synchronize other logic.

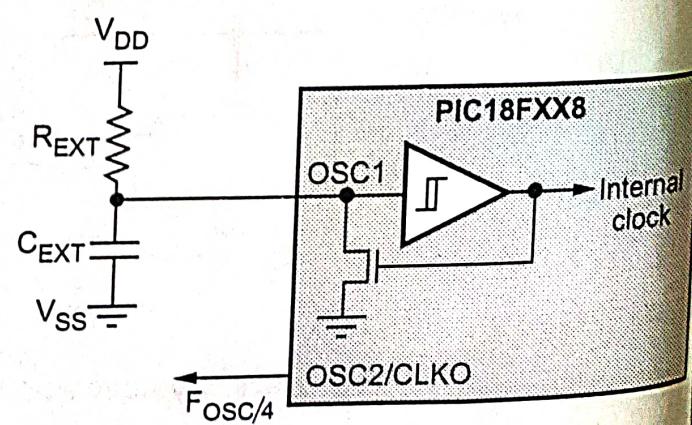


Fig. 2.9.2 shows RC Oscillator connections with PIC18

- The recommended values for RC are :
 - $3 \text{ k}\Omega \leq R_{\text{EXT}} \leq 100 \text{ k}\Omega$
 - $C_{\text{EXT}} > 20 \text{ pF}$
- The RCIO Oscillator mode functions like the RC mode, except that the OSC2 pin becomes an additional general purpose I/O pin. The I/O pin becomes bit 6 of PORTA (RA6).

2.9.3 External Clock Input

- The EC and ECIO Oscillator modes require an external clock source to be connected to the OSC1 pin. The feedback device between OSC1 and OSC2 is turned off in these modes to save current.
- In this mode, there is no oscillator start-up time required after a Power-on Reset or after a recovery from sleep mode.
- Like RC oscillator mode, in this mode, the oscillator frequency divided by 4 is available on the OSC2 pin. This signal may be used for test purposes or to synchronize other logic.
- Fig. 2.9.3 shows the pin connections for the EC Oscillator mode.

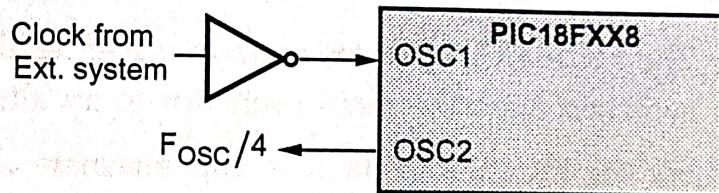


Fig. 2.9.3 Pin connections for EC Oscillator mode

- The ECIO Oscillator mode functions like the EC mode, except that the OSC2 pin becomes an additional general purpose I/O pin.

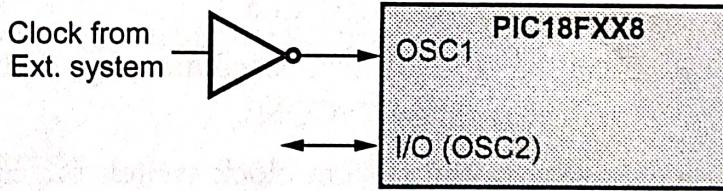


Fig. 2.9.4 Pin connections for ECIO Oscillator mode

2.9.4 HS4 (PLL)

- A Phase Locked Loop circuit is provided as a programmable option for users that want to multiply the frequency of the incoming crystal oscillator signal by 4. This is useful for users who are concerned with EMI due to high-frequency crystals.
- The PLL can only be enabled when the oscillator configuration bits are programmed for HS mode.
- The PLL is one of the modes of the FOSC2:FOSC0 configuration bits. The oscillator mode is specified during device programming.

- A PLL lock timer is used to ensure that the PLL has locked before device execution starts. The PLL lock timer has a time-out referred to as T_{PLL} .

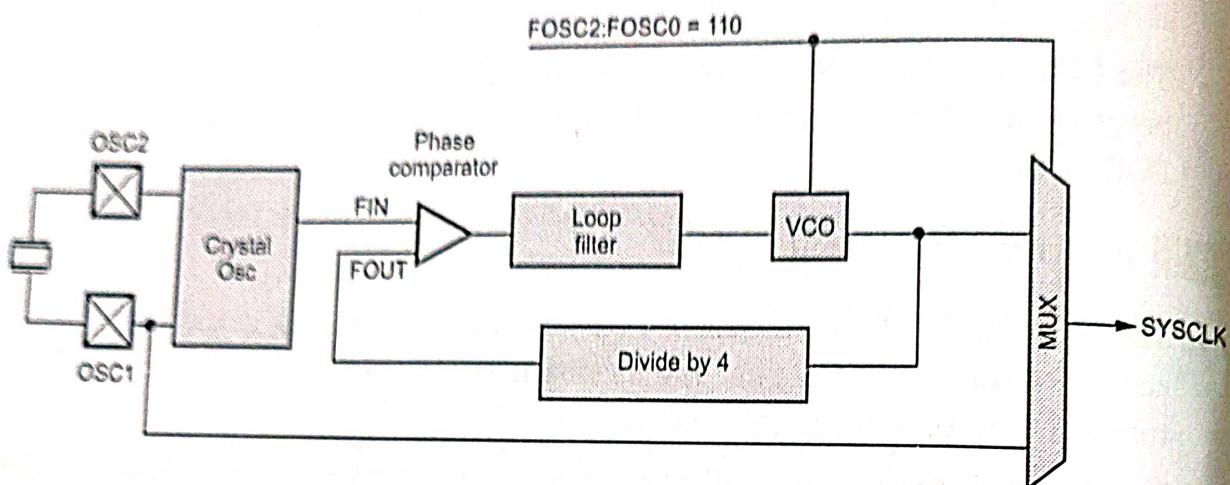


Fig. 2.9.5 PLL block diagram

2.9.5 Oscillator Switching Feature

- The PIC18FXX8 devices include a feature that allows the system clock source to be switched from the main oscillator to an alternate low-frequency clock source.
- For the PIC18FXX8 devices, this alternate clock source is the Timer1 oscillator.
- The clock switching can be enabled by
 - Programming the Oscillator Switching Enable (OSCSEN) bit in Configuration register, CONFIG1H, to a '0'.
 - Enabling the Timer1 oscillator by setting the T1OSCEN bit in the Timer1 Control register (T1CON).
 - By setting the system clock switch bit, SCS in the Oscillator Control (OSCCON) register to 1.
- When the SCS bit is '0', the system clock source comes from the main oscillator selected by the FOSC2:FOSC0 configuration bits.
- To avoid "glitches" when switching between oscillator sources, the circuitry in the PIC18FXX8 devices waits for eight rising edges of the clock source that the processor is switching to. This ensures that the new clock source is stable and that its pulse width will not be less than the shortest pulse width of the two clock sources.
- The sequence of events that takes place when switching from the Timer1 oscillator to the main oscillator will depend on the mode of the main oscillator. In addition to eight clock cycles of the main oscillator, additional delays may take place as follows :

- In XT, LP, or HS oscillator modes, the transition will take place after an oscillator start-up time (T_{OST}).
 - In HS4 (PLL) oscillator mode, the transition will take place after an oscillator start-up time (T_{OST}) plus an additional PLL time-out (T_{PLL}).
 - In RC, RCIO, EC or ECIO modes, there is no oscillator start-up time-out. Operation will resume after eight cycles of the main oscillator.
 - In the Sleep mode, the on-chip clocks and oscillator are turned off. This achieves the lowest current consumption of the device. The user can wake from Sleep through external Reset, Watchdog Timer Reset or through an interrupt.

2.9.6 CONFIG1H : Configuration Register 1 High (Byte Address 300001h)

- Bits : (FOSC2, FOSC1 and FOSC0) selects the one of the eight oscillator modes.
 - The clock switching feature is enabled by programming the Oscillator Switching Enable (OSCSEN) bit in Configuration register, CONFIG1H, to a '0'.

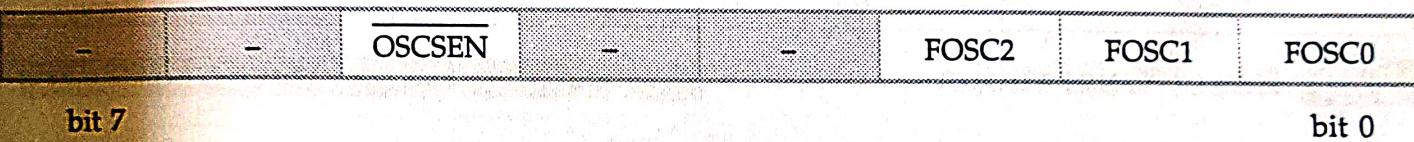


Fig. 2.9.6 Bit pattern of CONFIG1H register

bit 7-6 Unimplemented : Read as '0'

bit 5 (OSCSEN) : Oscillator System Clock Switch Enable bit

1 = Oscillator system clock switch option is disabled (main oscillator is source)

0 = Oscillator system clock switch option is enabled (oscillator switching is enabled)

bit 4-3 Unimplemented : Read as '0'

bit 2-0 FOSC2:FOSC0 : Oscillator Selection bits

111 = RC oscillator w/OSC2 configured as RA6

110 = HS oscillator with PLL enabled/clock frequency = (4 x FOSC)

101 = EC oscillator w/OSC2 configured as RA6

100 = EC oscillator w/OSC2 configured as divide-by-4 clock output

011 = RC oscillator

010 = HS oscillator

001 = XT oscillator

000 = LP oscillator

Review Questions

1. List various oscillator modes and corresponding configuration bit states, supported by PIC18FXX8 devices.
2. Explain the Crystal oscillator connections with the help of suitable diagram.
3. List the pros and cons of RC oscillator over crystal oscillator.
4. Explain the RC oscillator connections with the help of suitable diagram.
5. Write a note on EC and ECIO Oscillator modes of PIC18FXX8 devices.
6. Explain the PLL (HS4) oscillator mode of PIC18FXX8 devices.
7. Write a note on oscillator switching feature.
8. Explain the bit pattern of CONFIG1H register.
9. State the steps to enable clock switching in PIC18FXX8 devices.
10. State the sequence of events that takes place when switching from the Timer1 oscillator to the internal oscillator.

SPPU : Aug.-14, May-19 Marks

2.10 Reset Operations

SPPU : Dec.-14, 17, 18, May-13, 18, Aug.-15, 19

- The PIC18FXX8 devices support various RESET conditions. These are :
 1. Power-on Reset (POR)
 2. MCLR reset during normal operation
 3. MCLR reset during sleep
 4. Watchdog Timer (WDT) Reset during normal operation
 5. Programmable Brown-out Reset (PBOR)
 6. RESET Instruction
 7. Stack Full Reset
 8. Stack Underflow Reset
- Most registers are unaffected by a Reset. Their status is unknown on Power Reset (POR) and unchanged by all other Resets.
- The other registers are forced to a "Reset" state on Power-on Reset, (MCLR), WDT Reset, Brownout Reset, (MCLR) Reset during Sleep and by the RESET instruction.
- Fig. 2.10.1 shows a simplified block diagram of the On-Chip Reset Circuit.

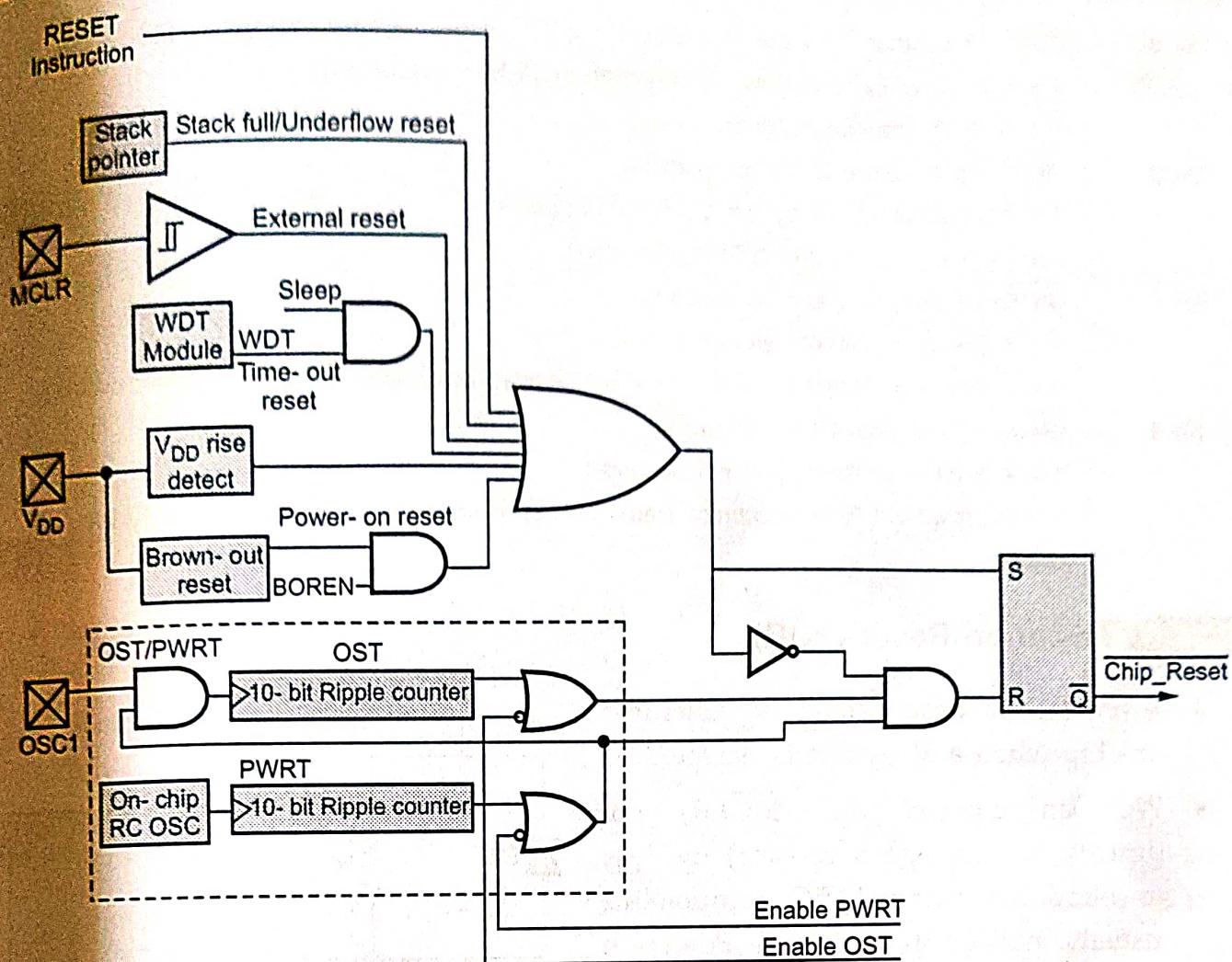


Fig. 2.10.1 Simplified block diagram of the On-Chip Reset Circuit

2.10.1 Reset Control (RCON) register

- The Reset Control (RCON) register contains flag bits that allow differentiation between the sources of a device Reset. These flags include the (**TO**), (**PD**), (**POR**), (**BOR**) and (**RI**) bits. This register is readable and writable.

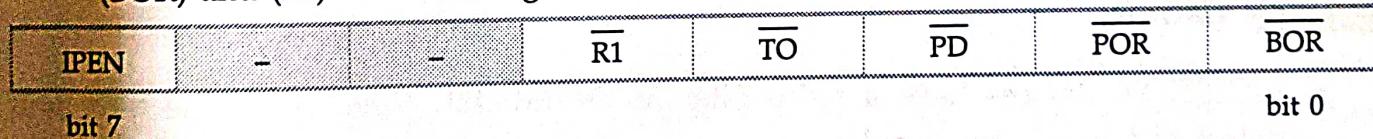


Fig. 2.10.2 Bit pattern of Reset Control (RCON) register

- bit 7 IPEN : Interrupt Priority Enable bit**
 1 = Enable priority levels on interrupts
 0 = Disable priority levels on interrupts (PIC16CXXX Compatibility mode)
- bit 6-5 Unimplemented : Read as '0'**
- bit 4 (**RI**) : RESET Instruction Flag bit**
 1 = The RESET instruction was not executed
 0 = The RESET instruction was executed causing a device Reset
 (must be set in software after a Brown-out Reset occurs)

bit 3	<u>(TO)</u> : Watchdog Time-out Flag bit 1 = After power-up, CLRWDT instruction or SLEEP Instruction 0 = A WDT time-out occurred
bit 2	<u>(PD)</u> : Power-down Detection Flag bit 1 = After power-up or by the CLRWDT instruction 0 = By execution of the SLEEP instruction
bit 1	<u>(POR)</u> : Power-on Reset Status bit 1 = A Power-on Reset has not occurred 0 = A Power-on Reset occurred (must be set in software after a Power-on Reset occurs)
bit 0	<u>(BOR)</u> : Brown-out Reset Status bit 1 = A Brown-out Reset has not occurred 0 = A Brown-out Reset occurred (must be set in software after a Brown-out Reset occurs)

2.10.2 Power-on Reset (POR)

- A Power-on Reset pulse is generated on-chip when a V_{DD} rise is detected.
- We can connect the (MCLR) pin directly (or through a resistor) to V_{DD} to eliminate external RC components usually needed to create a Power-on Reset delay. However, to provide slow rise times, we need to connect external RC components as shown in Fig. 2.10.3
- Recommended resistor values :
 $R < 40 \text{ k}\Omega$ and $R_1 = 100 \Omega$ to $1 \text{ k}\Omega$

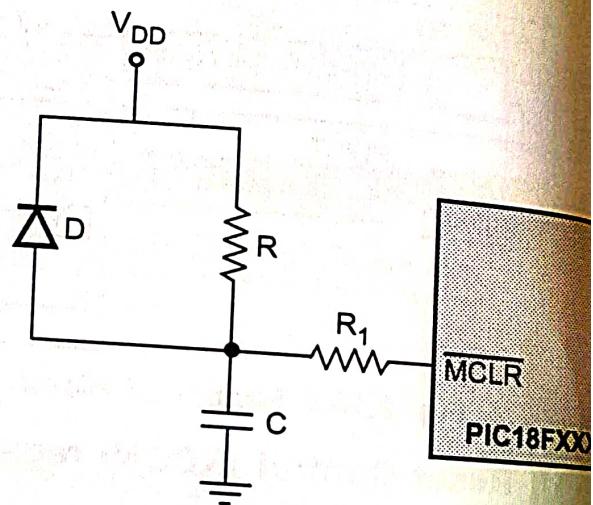


Fig. 2.10.3 Power on RESET circuit w/ external RC components

2.10.3 (MCLR)

- PIC18FXX8 devices have a noise filter in the (MCLR) Reset path. The filter detect and ignore small pulses.
- Voltages applied to the (MCLR) pin that exceed its specification can result in bounces during Resets and current draws outside of device specification during the Reset event. For this reason, it is recommended that to connect external RC components power on RESET instead of connecting MCLR pin directly to V_{DD} .

2.10.4 Power-up Timer (PWRT)

- The Power-up Timer provides a fixed nominal time-out, only on power-up from the POR. It operates on an internal RC oscillator.

- The chip is kept in Reset as long as the PWRT is active. The PWRT's time delay allows V_{DD} to rise to an acceptable level. A configuration bit (PWRTE in CONFIG2L register) is provided to enable/disable the PWRT.

2.10.5 Oscillator Start-up Timer (OST)

- The Oscillator Start-up Timer (OST) provides a 1024 oscillator cycle (from OSC1 input) delay after the PWRT delay is over. This additional delay ensures that the crystal oscillator or resonator has started and stabilized.
- The OST time-out is invoked only for XT, LP, HS and HS4 modes and only on Power-on Reset or wake-up from Sleep.

2.10.6 PLL Lock Time-out

- A portion of the Power-up Timer is used to provide a fixed time-out that is sufficient for the PLL to lock to the main oscillator frequency.
- This PLL lock time-out (T_{PLL}) is typically 2 ms and follows the oscillator start-up time-out (OST).

2.10.7 Brownout Reset

- A Brown-out Reset (BOR) is a circuit that monitors the V_{DD} level during operation by comparing it to a fixed threshold level. When V_{DD} drops below the threshold level, the Brown-out Reset is activated.
- When V_{DD} rises again, the controller is restarted after a specified delay. This ensures that the device does not continue program execution outside of its valid operation range, which can lead to improper code execution.
- Not all devices have BOR, but most do, and some have multiple voltage thresholds to select from.
- Many PIC microcontrollers have a BOR with configurable voltage levels. Device families, such as the PIC18F, PIC24F and dsPIC30F series, support the BOR threshold levels. Table 2.10.1 shows voltage thresholds for PIC18FXX8.

Brown-out Reset Voltage for PIC18FXX8	
BORV1:BORV0	Brown-out Reset Voltage
11	2.0 V
10	2.7 V
1	4.2 V
0	4.5 V

Table 2.10.1 Brown-out Reset voltage thresholds for PIC18FXX8

2.10.8 CONFIG2L : Configuration Register 2 Low (Byte Address 300002h)

- We can configure BOR threshold levels using BORV1 : BORV0 bits in the CONFIG2L register.
- CONFIG2L register also allows us to enable/disable Power-up Timer Enable bit.

-	-	-	-	BORV1	BORV0	BOREN	PWRTE
bit 7							bit

Fig. 2.10.4 Bit pattern of CONFIG2L register

bit 7-4 Unimplemented : Read as '0'

bit 3-2 BORV1 : BORV0 : Brown-out Reset Voltage bits

11 = VBOR set to 2.0 V

10 = VBOR set to 2.7 V

01 = VBOR set to 4.2 V

00 = VBOR set to 4.5 V

bit 1 BOREN : Brown-out Reset Enable bit

1 = Brown-out Reset enabled

0 = Brown-out Reset disabled

bit 0 (PWRTE) : Power-up Timer Enable bit

1 = PWRT disabled

0 = PWRT enabled

2.10.9 Watchdog Timer

- There are countless applications where the system cannot afford to get stuck at point (not even for a small duration of time). For example, in a radar system, the system hangs for 5 minutes, it can result in serious repercussions (an enemy plane or missile may go undetected resulting in huge losses).
- The watchdog timer is loaded with a timeout period which is dependent on the application. Whenever software failed to clear the watchdog timer before its timeout period, the watchdog timer resets the system.
- In PIC18FXX8, the Watchdog Timer is a free running, on-chip RC oscillator which does not require any external components. This RC oscillator is separate from the RC oscillator of the OSC1/CLKI pin.
- During normal operation, a WDT time-out generates a device Reset (Watchdog Timer Reset). If the device is in Sleep mode, a WDT time-out causes the device to wake-up and continue with normal operation (Watchdog Timer wake-up).

- Fig. 2.10.5 shows the block diagram of Watchdog Timer. As shown in Fig. 2.10.5, the Watchdog Timer is enabled/disabled by a device configuration bit. If the WDT is enabled, software execution may not disable this function. When the WDTEN (Watchdog Timer Enable) configuration bit is cleared, the SWDTEN (Software Controlled Watchdog Timer Enable) bit in the watchdog timer control register (WDTCON) enables/disables the operation of the WDT.
- WDTEN : Watchdog Timer Enable bit**
 - 1 = WDT enabled
 - 0 = WDT disabled (control is placed on the SWDTEN bit)
- SWDTEN : Software Controlled Watchdog Timer Enable bit**
 - 1 = Watchdog Timer is on
 - 0 = Watchdog Timer is turned off if the WDTEN configuration bit in the Configuration register = 0

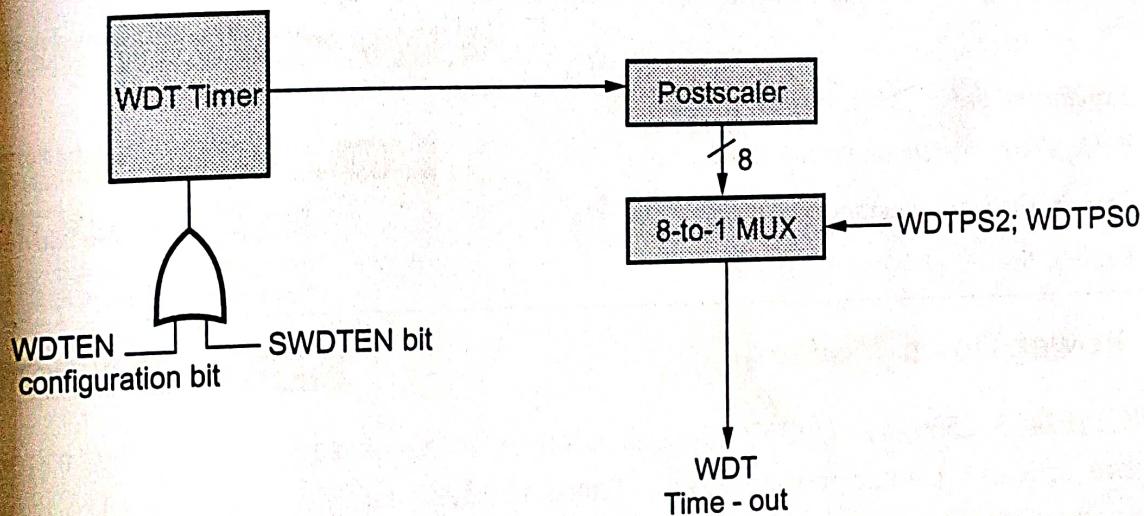


Fig. 2.10.5 Block diagram of Watchdog Timer

- The WDT has a **postscaler** that can extend the WDT Reset period. The postscaler is selected at the time of device programming by the value written into WDTPS2:WDTPS0 bits of the CONFIG2H Configuration register.

WDTPS2 : WDTPS0	Postscale
1 1 1	1 : 128
1 1 0	1 : 64
1 0 1	1 : 32
1 0 0	1 : 16
0 1 1	1 : 08
0 1 0	1 : 04
0 0 1	1 : 02
0 0 0	1 : 01

Table 2.10.2 Watchdog Timer Postscale Select bits

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
CONFIG2H	-	-	-	-	WDTPS2	WDTPS1	WDTPS0
RCON	IPEN	-	-	RI	TO	PD	POR
WDTCON	-	-	-	-	-	-	-

Fig. 2.10.6 Watchdog Timer Registers

Note : Shaded cells are not used by the Watchdog Timer.

- The (\overline{TO}) bit in the RCON register will be cleared upon a WDT time-out.

Review Questions

1. List out various RESET conditions supported by PIC18FXX8 devices.

2. Draw and explain the block diagram of On-Chip reset circuit.

SPPU : Dec.-17, 18, May-18, M

3. Explain the use of (MCLR), PWRT and OST.

4. Write short note on Brownout Reset.

SPPU : Dec.-14, Aug 15, M

5. Explain the block diagram of Watchdog Timer.

SPPU : May 13, M

6. Explain the use of Watchdog Timer.

SPPU : Aug.-16, M

2.11 Power Down Modes

SPPU : May-18, 19, A

- PIC18F2455/2550/4455/4550 devices offer a total of seven operating modes for more efficient power management. These modes provide a variety of options for selective power conservation in applications where resources may be limited (i.e., battery-powered devices).
- There are three categories of power-managed modes:
 - Run modes
 - Idle modes
 - Sleep mode
- These categories define which portions of the device are clocked and somewhat speed.
- The Run and Idle modes may use any of the three available clock sources (primary, secondary or internal oscillator block); however, the Sleep mode does not use a clock source.
- Selecting a power-managed mode requires two decisions: If the CPU is clocked or not and the selection of a clock source.

- The IDLEN (bit 7) of OSCCON register controls CPU clocking, while the SCS1 : SCS0 (bits1:0) of OSCCON register select the clock source.
- Table 2.11.1 lists the seven operating modes for more efficient power management.

Mode	OSCCON Bits	
	IDLEN	SCS1 : SCS0
Sleep	0	N/A
PRI_RUN	N/A	00
SEC_RUN	N/A	01
RC_RUN	N/A	1X
PRI_IDLE	1	00
SEC_IDLE	1	01
RC_IDLE	1	1X

Table 2.11.1 Power managed modes

2.11.1 Run Modes

- In the Run modes, clocks to both the core and peripherals are active. The difference between these modes is the clock source.

PRI_RUN Mode

- This is also the default mode upon a device Reset. It is the normal, full-power execution mode of the PIC microcontroller.

SEC_RUN Mode

- The SEC_RUN mode is the compatible mode to the "clock switching" feature offered in other PIC18 devices.
- In this mode, the CPU and peripherals are clocked from the Timer1 oscillator.
- This gives users the option of lower power consumption while still using a high-accuracy clock source.

RC_RUN mode

- In RC_RUN mode, the CPU and peripherals are clocked from the internal oscillator block using the INTOSC multiplexer and the primary clock is shut down.
- This mode provides the best power conservation of all the Run modes while still executing code.

- It works well for user applications which are not highly timing sensitive or do not require high-speed clocks at all times.

2.11.2 Idle Modes

- The Idle modes allow the controller's CPU to be selectively shut down while peripherals continue to operate.
- In these modes, the peripherals will be clocked from the clock source selected using the SCS1 : SCS0 bits; however, the CPU will not be clocked.
- Setting IDLEN (bit 7 of Oscillator Control Register) and executing a SLEEP instruction provides a quick method of switching from a given Run mode to corresponding Idle mode.
- Since the CPU is not executing instructions, the only exits from any of the modes are by interrupt, WDT time-out or a Reset.
- When the CPU begins executing code, it resumes with the same clock source as the current Idle mode.
- The IDLEN and SCS bits are not affected by the wake-up.

PRI_IDLE mode

- In this mode, the CPU is disabled but the peripherals continue to be clocked from the primary clock source specified by the FOSC3:FOSC0 configuration bits.
- This mode provides fastest resumption of device operation, with its more accurate primary clock source.
- This mode is entered from PRI_RUN mode by setting the IDLEN bit and executing a SLEEP instruction.
- If the device is in another Run mode, PRI_IDLE mode is entered by setting IDLEN first, then clearing the SCS (System Clock Select) bits to select the primary clock source and executing SLEEP instruction.

SEC_IDLE Mode

- In this mode, the CPU is disabled but the peripherals continue to be clocked from the Timer1 oscillator.
- This mode is entered from SEC_RUN by setting the IDLEN bit and executing SLEEP instruction.
- If the device is in another Run mode, SEC_IDLE mode is entered by setting IDLEN first, then setting the SCS (System Clock Select) bits to "01" to select the Timer1 oscillator and executing SLEEP instruction.

RC_IDLE Mode

- In this mode, the CPU is disabled but the peripherals continue to be clocked from the internal oscillator.
- This mode is entered from RC_RUN by setting the IDLEN bit and executing a SLEEP instruction.
- If the device is in another Run mode, RC_IDLE mode is entered by setting IDLEN first, then setting the SCS1 (System Clock Select) bit to select the internal oscillator and executing SLEEP instruction.

2.11.3 Sleep Mode

- In sleep mode, a PIC microcontroller is placed in its lowest current consumption state. PIC18FXX8 devices does not support idle modes, they support only sleep mode.
- PIC18FXX8 devices go into power-down mode by executing a SLEEP instruction. The (MCLR) ? pin must be at a logic high level for power mode to be active.
- PIC18F2455/2550/4455/4550 devices enter into sleep mode by clearing the IDLEN bit (the default state on device Reset) and executing the SLEEP instruction.
- In this mode, the device oscillator is turned off, so no system clocks are occurring in the device. However, the I/O ports maintain the status they had before the SLEEP instruction was executed.
- In order to minimize the power consumption in power-down mode, the output ports must not be sourcing or sinking the current before going into power-down mode. Besides, all the unused I/O pins should be configured as inputs and pulled either high (V_{DD}) or low (V_{SS}).

2.11.4 Wake-Up from Power-down

- An exit from Sleep mode or any of the Idle modes is triggered by an
 - **Interrupt** : Any of the available interrupt sources can cause the device to exit from an Idle mode, or the Sleep mode, to a Run mode.
 - **Reset**
 - **WDT time-out**
- In PIC18F458 devices several events can make the device wake up from the power-down (sleep) mode:
 1. External Reset input on MCLR pin.
 2. Watchdog Timer wake-up (if WDT was enabled).
 3. Interrupt from INT pin, RB port change or a peripheral interrupt.

- The following peripheral interrupts can wake the device from power-down mode :
 1. PSP (Parallel Slave Port) read or write.
 2. TMR1 interrupt. Timer1 must be operating as an asynchronous counter.
 3. TMR3 interrupt. Timer3 must be operating as an asynchronous counter.
 4. CCP Capture mode interrupt.
 5. Special event trigger (Timer1 in Asynchronous mode using an external clock).
 6. MSSP (Master Synchronous Serial Port) Start/Stop bit detect interrupt.
 7. MSSP transmit or receive in Slave mode (SPI/I2C).
 8. USART RX or TX (Synchronous Slave mode).
 9. A/D conversion (when A/D clock source is RC).
 10. EEPROM write operation complete.
 11. LVD (Low-Voltage Detection) interrupt.
- Other peripherals cannot generate interrupts, since during Sleep, no on-chip devices are present.

Review Questions

1. What are the different power down modes in PIC microcontroller ?

SPPU : May-18, Aug.-16, Marks

2. Explain the power down mode of PIC18F458.

SPPU : May-19, Marks

3. In PIC18F458, list various events that makes device to wake-up from power down mode.