UNIT 3

PROGRAM ROM ALLOCATION IN C-PIC16F877A

Memory of the PIC16F877 divided into 3 types of memories:

- Program Memory – A memory that contains the program(which we had written), after we've burned it. As a reminder, Program Counter executes commands stored in the program memory, one after the other.
- Data Memory – This is RAM memory type, which contains a special registers like SFR (Special Faction Register) and GPR (General Purpose Register). The variables that we store in the Data Memory during the program are deleted after we turn of the micro.

These two memories have separated data buses, which makes the access to each one of them very easy.

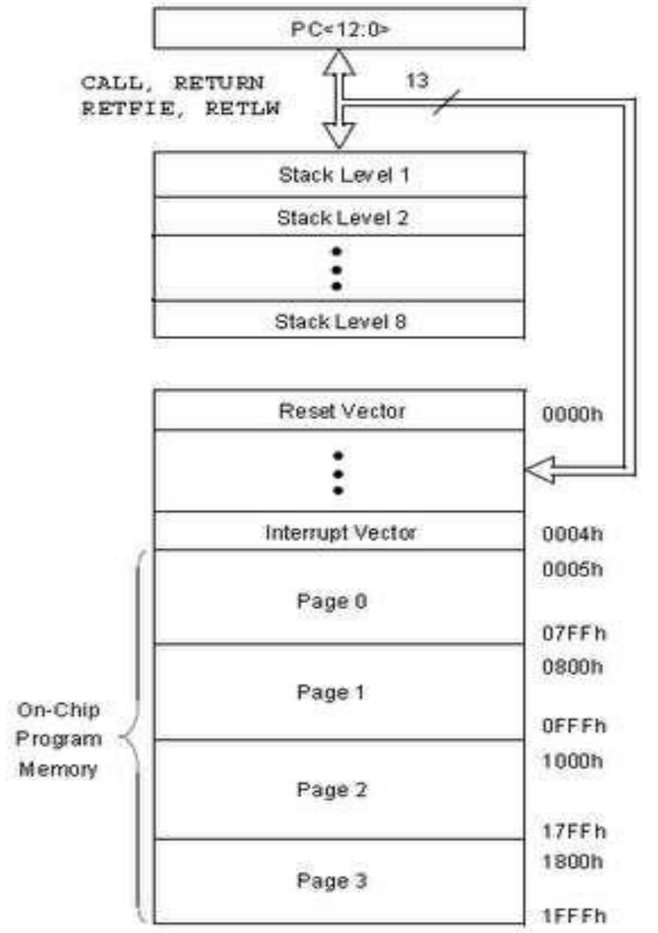
- Data EEPROM (Electrically Erasable Programmable Read-Only Memory) – A memory that allows storing the variables as a result of burning the written program.

Each one of them has a different role. Program Memory and Data Memory two memories that are needed to build a program, and Data EEPROM is used to save data after the microcontroller is turn off.

Program Memory and Data EEPROM they are non-volatile memories, which store the information even after the power is turn off. These memories called Flash Or EEPROM. In contrast, Data Memory does not save the information because it needs power in order to maintain the information stored in the chip.

PIC16F87XA Program Memory

The PIC16F87XA devices have a 13-bit program counter capable of addressing an 8K word x 14 bit program memory space. This memory is used to store the program after we burn it to the microcontroller. The PIC16F876A/877A devices have 8K words x 14 bits of Flash program memory that can be electrically erased and reprogrammed. Each time we burn program into the micro, we erase an old program and write a new one.



PIC16F876A/877A program memory map and stack

Program Counter (PC) keeps track of the program execution by holding the address of the current instruction. It is automatically incremented to the next instruction during the current instruction execution.

The PIC16F87XA family has an 8-level deep x 13-bit wide hardware stack. The stack space is not part of either program or data space and the stack pointer is not readable or writable. In the PIC microcontrollers, this is a special block of RAM memory used only for this purpose.

The CALL instruction is used to jump to a subroutine, which must be terminated with the RETURN instruction. CALL has the address of the first instruction in the subroutine as its operand. When the CALL instruction is executed, the destination address is copied to the PC. The PC is PUSHed onto the stack when a CALL instruction is executed, or an interrupt causes a branch. The stack is POP'ed in the event of a RETURN, RETLW or a RETFIE instruction execution.

The stack operates as a circular buffer. This means that after the stack has been PUSHed eight times, the ninth push overwrites the value that was stored from the first push. The tenth push overwrites the second push (and so on).

Each time the main program execution starts at address 0000 – Reset Vector. The address 0004 is “reserved” for the “interrupt service routine” (ISR).

If we plan to use an interrupt, our program will begin after the Interrupt Vector; and if not we can start to write from the beginning of the Reset Vector.