# PIC Microcontrollers

Looking at the range of PIC microcontrollers today, anyone can be forgiven for a sense of complete bewilderment. There are literally hundreds of different devices, offered in different packages, for different applications. Let us therefore try to identify the characteristics that all of these have in common. All PIC microcontrollers are low-cost, self-contained, 8-bit, Harvard structure, pipelined, RISC, single accumulator (the Working or W register), with fixed reset and interrupt vectors. Today, Microchip offers five main families of microcontrollers, whose features are summarized in the below table. It is possible to see clear evolution from one family to the other, so knowledge of one readily leads to knowledge of another. Every member of any one family shares the same core architecture and instruction set. The families are identified primarily by the first two digits of the device code. The alphabetic character that follows gives some indication of the technology used. The 'C' insert implies CMOS technology, where CMOS stands for Complementary Metal Oxide Semiconductor, the leading semiconductor technology for implementing low-power logic systems. The 'F' insert indicates incorporation of Flash memory technology (still using CMOS as the core technology). An 'A' after the number indicates a technological upgrade on the first issue device. An 'X' indicates that a certain digit can take a number of values, the one taken being unimportant to the overall number quoted. For example, the 16C84 was the first of its kind. It was later reissued as the 16F84, incorporating Flash memory technology. It was then reissued as the 16F84A, with certain further technological upgrades.

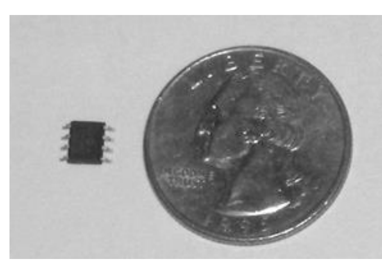
| PIC family | Stack size (words) | Instruction word size | Number of instructions | Interrupt vectors |
|---|---|---|---|---|
| 12CXXX/12FXXX | 2 | 12- or 14-bit | 33 | None |
| 16C5XX/16F5XX | 2 | 12-bit | 33 | None |
| 16CXXX/16FXXX | 8 | 14-bit | 35 | 1 |
| 17CXXX | 16 | 16-bit | 58, including hardware multiply | 4 |
| 18CXXX/18FXXX | 32 | 16-bit | 75, including hardware multiply | 2 (prioritised) |

**The 16C5X Series family**

This PIC microcontroller family represents the most direct descendant of its General Instrument ancestors and displays all the core features of the original PIC design. With only a two-level stack and no interrupts, there is significant limitation on the program and hardware complexity that can be developed. Particularly without interrupts there is restriction on the type of on-chip peripheral that can be included, as most peripherals use interrupts to enhance their interface with the CPU. The 16C5X family has also been issued with Flash memory, with 16F5X codes. While this family is well established, it has a limited number of members and is not being given much prominence by Microchip.

**The 12 Series family**

The 12 series microcontrollers are designed for really tiny applications, being packaged in small ICs (for example, 8- or 14-pin). They have a simple architecture and can be viewed as 'stripped-down' versions of the 16C5XX series, with the same instruction set. Despite their small size, 12 Series microcontrollers carry some interesting peripherals, including analog-to-digital converters and EEPROM (Electrically Erasable Programmable Read-Only Memory) data memory. Although a small family, there is strong interest at this end of the size range.



**The PIC 16 Series family**

This, the ‘mid-range’ family, represents an improved version of the 16C5XX Series, in which interrupts (albeit with a single interrupt vector) are introduced and the stack size increased. The instruction set is a slight extension of that of the 16C5X. A very wide range of family members exists, with many different peripherals and technical enhancements. The larger devices, with many peripherals and significant on-chip memory, are both powerful and versatile. There are four device “types” as indicated in the device number.

1. **F**, as in PIC16**F**84. These devices have Flash program memory and operate over the standard voltage range.
2. **LF**, as in PIC16**LF**84. These devices have Flash program memory and operate over an extended voltage range.
3. **CR**, as in PIC16**CR**83. These devices have ROM program memory and operate over the standard voltage range.
4. **LCR**, as in PIC16**LCR**84. These devices have ROM program memory and operate over an extended voltage range.

**The 17 Series family**

This family was introduced to give a real step-up in CPU performance compared with any of the 16 Series devices. While retaining the RISC strategy, the instruction set size is nearly doubled and the instruction word size increased to 16-bit. Thus, some programming activities that are awkward in the mid-range family, like table reads or data moves, are here much simpler. A hardware multiplier is also available. The single, often overloaded, interrupt vector of the mid-range family becomes four. Although much more powerful than the 16 Series, this family is limited in number, and Microchip appear to be focusing on the 18 Series family to move forward developments at the more powerful end of their range.

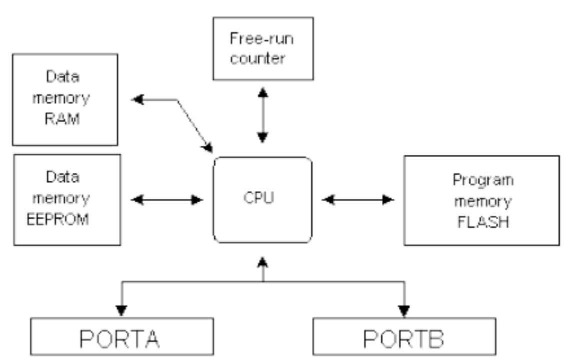
**The 18 Series family**

In this family Microchip comes to grips with some of the issues of sophisticated processors. The instruction set has increased again, now to 75 instructions, and is designed to facilitate use of the C programming language. In certain versions there is also an ‘extended’ instruction set, with a further small set of instructions. There are two interrupt vectors, which can be prioritized. This is an extremely powerful family of microcontrollers and a number of new members can be expected in the future.

## PIC 16F84

**PIC16F84** belongs to a class of 8-bit microcontrollers of RISC architecture. Its general structure is shown on the following map representing basic blocks.

**Program memory** (FLASH)- for storing a written program. Since memory made in FLASH technology can be programmed and cleared more than once, it makes this microcontroller suitable for device development.




**EEPROM** - data memory that needs to be saved when there is no supply. It is usually used for storing important data that must not be lost if power supply suddenly stops. For instance, one such data is an assigned temperature in temperature regulators. If during a loss of power supply this data was lost, we would have to make the adjustment once again upon return of supply. Thus our device looses on self-reliance.

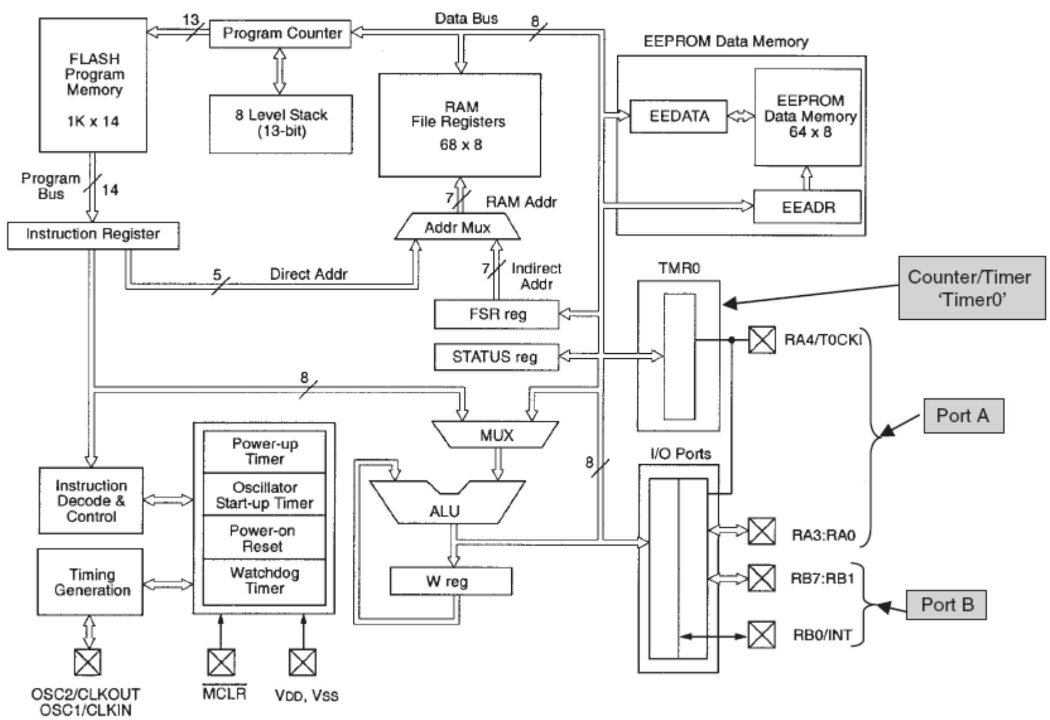
**RAM** - data memory used by a program during its execution. In RAM are stored all inter-results or temporary data during run-time.

**PORTA and PORTB** are physical connections between the microcontroller and the outside world. Port A has five, and port B eight pins.

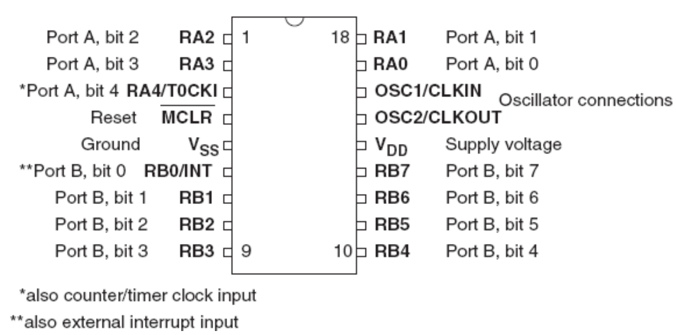
**FREE-RUN TIMER** is an 8-bit register inside a microcontroller that works independently of the program. On every fourth clock of the oscillator it increments its value until it reaches the maximum (255), and then it starts counting over again from zero. As we know the exact timing between each two increments of the timer contents, timer can be used for measuring time which is very useful with some devices.

**CENTRAL PROCESSING UNIT** has a role of connective element between other blocks in the microcontroller. It coordinates the work of other blocks and executes the user program.

The detail architecture is shown in the figure below




The pin connection diagram of the 16F84A is shown in Figure below




Pin no.1 **RA2** Second pin on port A. Has no additional function.
Pin no.2 **RA3** Third pin on port A. Has no additional function.

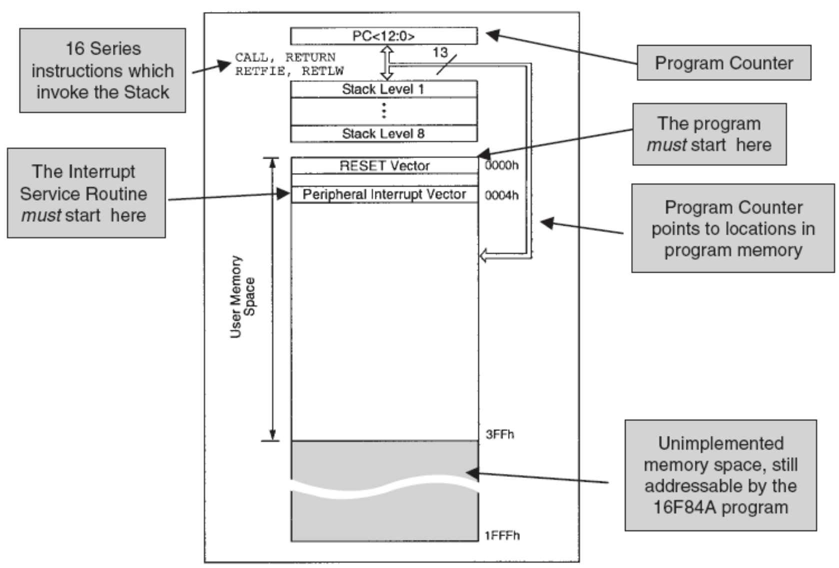
Pin no.3 **RA4** Fourth pin on port A. TOCK1 which functions as a timer is also found on this pin
Pin no.4 **MCLR** Reset input and Vpp programming voltage of a microcontroller.
Pin no.5 **Vss** Ground of power supply.
Pin no.6 **RB0** Zero pin on port B. Interrupt input is an additional function.
Pin no.7 **RB1** First pin on port B. No additional function.
Pin no.8 **RB2** Second pin on port B. No additional function.
Pin no.9 **RB3** Third pin on port B. No additional function.
Pin no.10 **RB4** Fourth pin on port B. No additional function.
Pin no.11 **RB5** Fifth pin on port B. No additional function.
Pin no.12 **RB6** Sixth pin on port B. 'Clock' line in program mode.
Pin no.13 **RB7** Seventh pin on port B. 'Data' line in program mode.
Pin no.14 **Vdd** Positive power supply pole.
Pin no.15 **OSC2** Pin assigned for connecting with an oscillator
Pin no.16 **OSC1** Pin assigned for connecting with an oscillator.
Pin no.17 **RA2** Second pin on port A. No additional function.
Pin no.18 **RA1** First pin on port A. No additional function.

# MEMORY ORGANIZATION

There are two memory blocks in the PIC16F8X. These are the program memory and the data memory. Each block has its own bus, so that access to each block can occur during the same oscillator cycle. The data memory can further be broken down into the general purpose RAM and the Special Function Registers (SFRs). These registers used to control and monitor the peripheral modules. The data memory area also contains the data EEPROM memory. This memory is not directly mapped into the data memory, but is indirectly mapped. That is, an indirect address pointer specifies the address of the data EEPROM memory to read/write. The 64 bytes of data EEPROM memory have the address range 0h-3Fh.

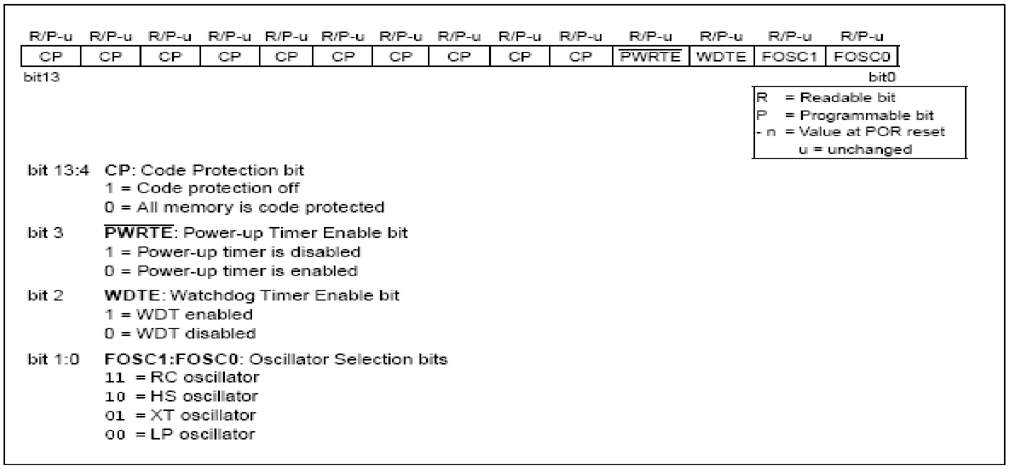
## a) Program Memory Organization

The PIC16FXX has a 13-bit program counter capable of addressing an 8K x 14 program memory space. For the PIC16F84 and PIC16CR84, the first 1K x 14 (0000h-03FFh) are physically implemented. Accessing a location above the physically implemented address will cause a wraparound. For example, for the PIC16F84 locations 20h, 420h, 820h, C20h, 1020h, 1420h, 1820h, and 1C20h will be the same instruction. The reset vector is at 0000h and the interrupt vector is at 0004h.



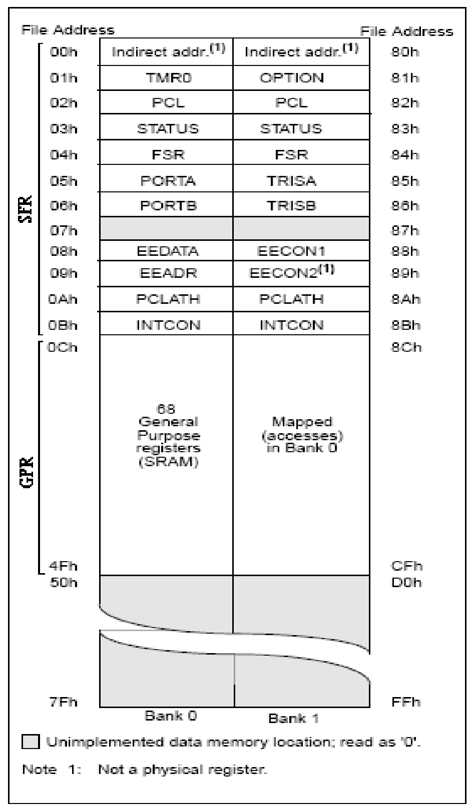

## The Configuration Word

A special part of the 16F84A program memory is its *Configuration Word* (Figure below). This allows the user to define certain configurable features of the microcontroller, at the time of program download. These are fixed until the next time the microcontroller is programmed. This is distinct from those many selectable features, like the setting of SFRs, which are under normal program control. While the Configuration Word is part of program memory, it is not accessible within the program or in any way while the program is running.




## b) Data Memory Organization

The data memory is partitioned into two areas. The first is the Special Function Registers (SFR) area, while the second is the General Purpose Registers (GPR) area. The SFRs control the operation of the device. Portions of data memory are banked. This is for both the SFR area and the GPR area. The GPR area is banked to allow greater than 116 bytes of general purpose RAM. The banked areas of the SFR are for the registers that control the peripheral functions. Banking requires the use of control bits for bank selection. These control bits are located in the STATUS Register. The programmer must ensure that the bank bit in the Status register is correctly set before making any access to memory.




***Why 'Banked' addressing?***

*A problem with any memory space is that the larger the memory is, the larger the address bus must be. One way of avoiding big address buses is to divide the memory into a number of smaller blocks – called banks – each identical in size. Now a smaller address bus can be used. It can access all banks in an identical way, with just one of the banks being identified at any one time as the target of the address specified.*

## c) Data EEPROM Memory

PIC16F84 has 64 bytes of EEPROM memory locations on addresses from 00h to 63h those can be written to or read from. The most important characteristic of this memory is that it does not loose its contents during power supply turned off. That practically means that what was written to it will be remaining even if microcontroller is turned off. Data can be retained in EEPROM without power supply for up to 40 years (as manufacturer of PIC16F84 microcontroller states), and up to 10000 cycles of writing can be executed. In practice, EEPROM memory is used for storing important data or some process parameters. One such parameter is a given temperature, assigned when setting up a temperature regulator to some process. If that data wasn't retained, it would be necessary to adjust a given temperature after each loss of supply. Since this is very impractical (and even dangerous), manufacturers of microcontrollers have began installing one smaller type of EEPROM memory. **EEPROM memory is placed in a special memory space** and can be accessed through special registers.

These registers are:

• **EEDATA** at address 08h, which holds read data or that to be written.

• **EEADR** at address 09h, which contains an address of EEPROM location being accessed.

• **EECON1** at address 88h, which contains control bits.

• **EECON2** at address 89h. This register does not exist physically and serves to protect EEPROM from accidental writing.