**PIC12C508 Instruction Set Architecture**

Any of the given lists of the compilers can be used to write the program of the microcontroller. All of the compilers will write the program in assembly language. To write the code some instructions should be kept in mind. In PIC12C508 each instruction given to the controller should be in 12bits. The 12-bit gives the 4 types of instruction which are:

* Byte-Oriented for File Register Operation
* Bit Oriented for File Register Operation
* Literal and Control operations
	+ 8 – bit
	+ 9 – bit

During any operation, the 12-bit data will consist of multiple sets of instructions.

**Byte-Oriented for File Register Operation**

In **Byte-Oriented** file register, the least five bits will the file register address, the next one bit will represent the destination and the first 6-bits will be the OPCODE which is used by the machine learning instructions.

|  |  |  |
| --- | --- | --- |
| **OPCODE (6-bit)** | **DESIGNATION (1-bit)** | **FILE ADDRESS (5-bit)** |

**Bit Oriented for File Register Operation**

In the **Bit-Oriented** file register, the last 5 bits will be for file register address, the next 3 will be for bit address and the rest of the 4-bit code will be for OPCODE.

|  |  |  |
| --- | --- | --- |
| **OPCODE (4-bit)** | **BIT ADDRESS (3-bit)** | **FILE ADDRESS (5-bit)** |

**Literal and Control operations**

In literal and Control operations there will be only two parts part will be for OPCODE and rest of them will be for the literal value of control operation. In Literal and Control Operations, for GOTO instruction OPCODE will be of 3 bit and literal will be of 9 bits but in other cases expect GOTO operation the literal will be of 8 bits and OPCODE will be of 4 bits.

|  |  |
| --- | --- |
| **OPCODE (4-bit)** | **FILE ADDRESS (8-bit) (except GOTO instruction)** |

|  |  |
| --- | --- |
| **OPCODE (3-bit)** | **FILE ADDRESS (9-bit) (GOTO instruction)** |

**PIC12C508 Assembly Language Instructions Set**

Every single operation in the PIC12C508 will be handle by mostly Assembly language and by using assembly language data will be processed from register to register to perform each and every operation. Some main registers which will be mostly used are file address register and working register. These two registers will be represented by f and W. These two registers locations will be defined by Destination select bit in each instruction format. These two registers will be used for most every operation but the rest of the registers have also some uses. You can check tutorials for a complete guide of PIC microcontroller assembly language programming:

* **[pic microcontroller assembly language programming](https://microcontrollerslab.com/pic-microcontroller-assembly-language/%22%20%5Ct%20%22_blank)**

To specify the remain registers and operation we will need to use the specific symbol/values. Here’s the list of all the remaining operations and registers:



These all are the register and operations which will be used for multiple purposes. Some of them are literal, designation and rest of them are register address or data. The OPCODE will be used with them to perform the specific operation and for each and every operation OPCODE will be different. The list of the OPCODE is given below:



The OPCODES are in **assembly language** and they could be used by any given development tool, but maybe you find it difficult to understand and could find some better tools in high-level language on the internet