**Name:**

**Roll no :**

**Group A Lab Assignment:** 1

**Subject:** PSDL

**Title:** Addition of two hexadecimal numbers

**Assignment No: 1**

**Title :** Embedded C programming for Addition of two hexadecimal numbers

**Aim :** Write an embedded c program to add two hexadecimal numbers

**Objective:** To learn embedded C programming for PIC18FXXX microcontrollers and perform addition operation.

**Theory:**

1. **Programming of PIC microcontroller in Embedded C**

**1.1 Embedded C programming**

Earlier, many embedded applications were developed using assembly level programming which has many disadvantages, one of which was, they did not provide portability. This disadvantage was overcome by the advent of various high-level languages like C, Pascal, and COBOL. However, it was the C language that got extensive acceptance for embedded systems. C language was developed by Dennis Ritchie in 1969. The C code written is more reliable, scalable, and portable; and in fact, much easier to understand. Embedded C programming builds with a set of functions where every function is a set of statements that are utilized to execute some particular tasks. Both the embedded C and C languages are the same and implemented through some fundamental elements like a variable, character set, keywords, data types, declaration of variables, expressions, statements. All these elements play a key role while writing an embedded C program. Embedded C language is most frequently used to program the PIC microcontroller.

The main features of the C language include the following:

* C language is software designed with different keywords, data types, library functions, variables, constants, etc.
* Programming in embedded C is less tedious, less time consuming, easier to modify and update compared to assembly lamguage..
* Embedded C is an extension to the C language with some additional header files. These header files may change from controller to controller.
* For PIC18F4550 header file #include<PIC18F4550.h> is used.

1. **Development Tools for PIC18** **FXXX Microcontrollers:**

The tools for developing software and hardware for PIC 18FXXX microcontrollers include editors, assemblers, compilers, debuggers, simulators, emulators, and device programmers. A typical development cycle starts with writing the application program using a text editor, then translating into an executable code with the help of an assembler or compiler. If the program has several modules, a linker is used to combine them into a single application. Any syntax errors are detected by the assembler or compiler and must be corrected before the executable code can be generated. Next, a simulator is used to test the application program without the target hardware. Simulators are helpful in checking the correctness of an algorithm or a program with limited or no input-outputs, and most errors can be removed during simulation. Here we are using MPLAB IDE (v5.20) software and MPLAB XC8 (v2.36) compiler for code editing, testing, debugging and executing our application programs for PIC18FXXX microcontroller

.

**2.1) MPLAB Integrated Development Environment Software:**

MPLAB X IDE is a free, integrated toolset for the development of embedded applications on PIC MCU families. It is called an Integrated Development Environment, or IDE, because it provides a single integrated environment to develop code for embedded microcontrollers**.** The MPLAB IDE provides the ability to:

* Create and edit source code using the built-in editor.
* Assemble, compile and link source code.
* Debug the executable logic by watching program flow with the built-in simulator or in real time with in-circuit emulators or in-circuit debuggers.
* Make timing measurements with the simulator or emulator.
* View variables in Watch windows.
* Program firmware into devices with device programmers.

**2.2) MPLAB XC8**

MPLAB IDE by default does not include any compiler. We can use it to write assembly language programs without the need for any compiler. MPLAB X supports MPLAB XC8 which is C compiler for 8-bit PIC and AVR devices. It integrates with MPLAB X IDE to provide a full graphical front end. Features of the MPLAB XC8 compiler include:

* Supports all 8-bit PIC and AVR MCUs
* Integrates with MPLAB X IDE to provide a full graphical front end:
* Editing errors and breakpoints match the corresponding lines in the source code
* Single step through C source code to inspect variables and structures at critical points
* Data structures with defined data types, including floating point, display in Watch windows
* Runs on Windows, Linux, and macOS X.

**Procedure:**

**Step 1:** Open MPLABX IDE on the PC for program development and create a new project and save it in a new folder.

**Step 2:** Write the program in C language to add two hexadecimal numbers

**Step 3:** Build the program and create hex file. In case of errors correct program and rebuild to create hex file.

**Step 4:** Select Window->Target memory views->**Configuration bits** from tool bar. Select appropriate settings, generate source code and paste the configurations in the main program.

**Step 5:** Select debug project and then Finish debugger session from the tool bar.

**Step 6:** Select Window->Target memory views->**SFRs to view the output.**

**Source code :**

#include<stdio.h>

#include<stdlib.h>

#include<pic18f4550.h>

#include <xc.h>

void main(void)

{

int sum; // Variable declaration of type int

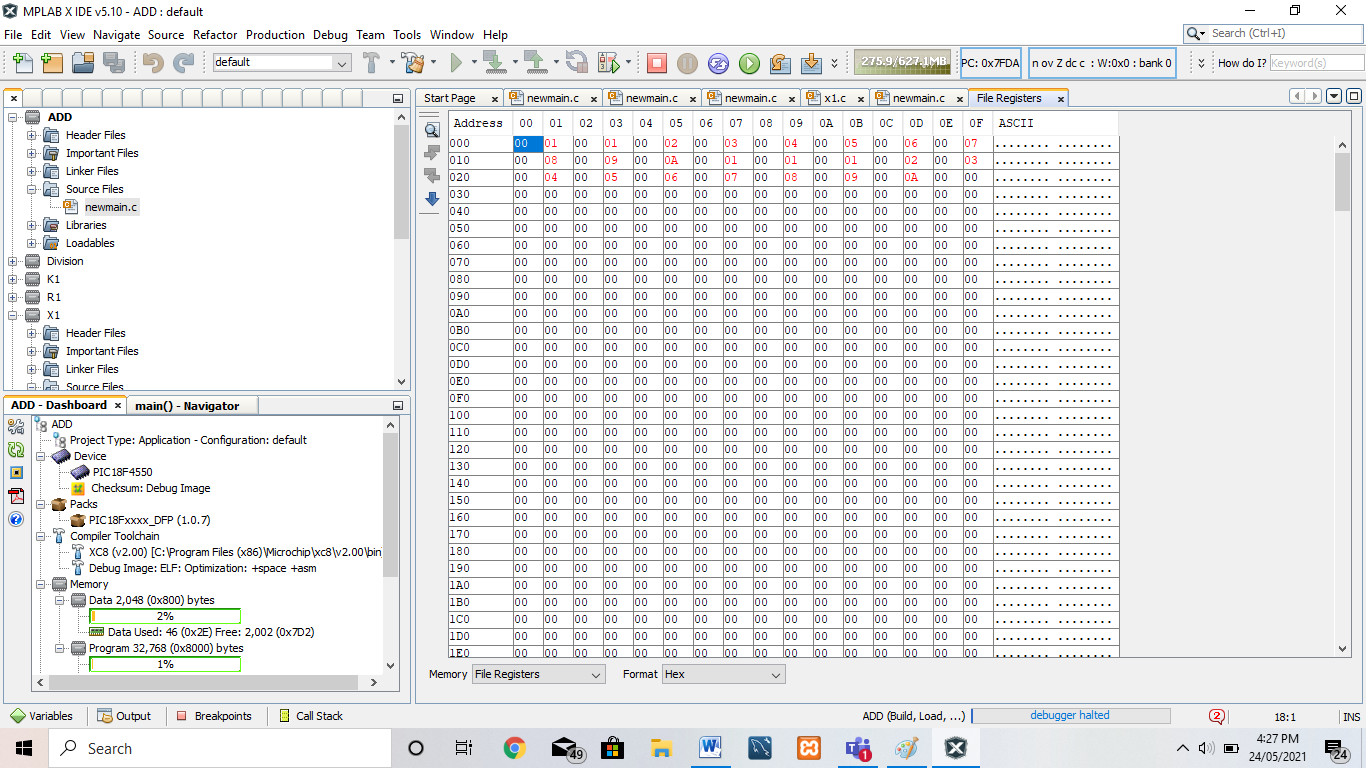
sum=0; // Initialize sum to zero

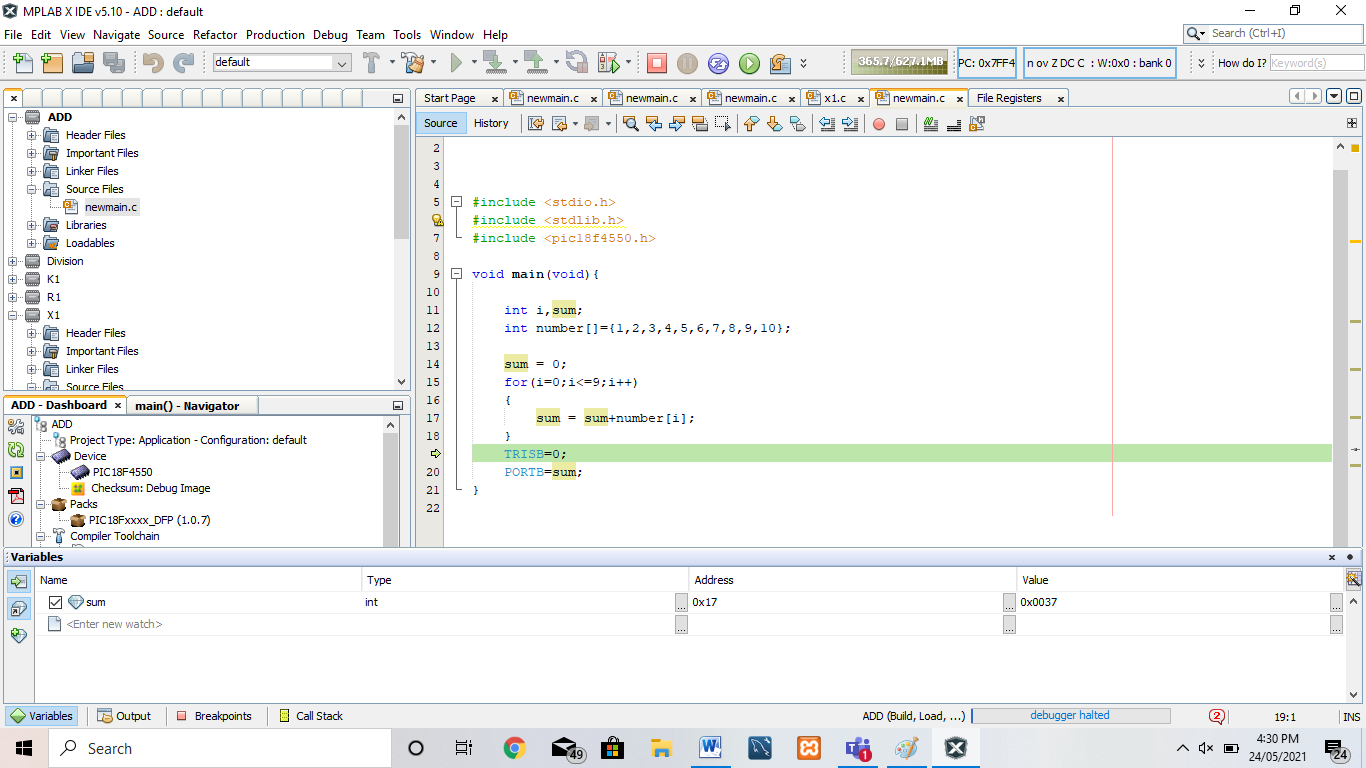
sum=0x0A+0x02;

TRISB=0; // Initialize port B for output

PORTB=sum; // Output written from sum to port B

}

**OUTPUT:**



**Conclusion:** Thus, we have studied embedded C programming for PIC18FXXX microcontrollers and perform addition operation.