**Assignment No.: 6**

**AIM:** Write a program to add n, 8 bits numbers found in internal ram location 40H onwards and store results in R6 and R7

**OBJECTIVE:** i) To perform arithmetic operations using 8051

ii) To learn internal memory access of 8051

S/W & H/W USED: 1. Keil uVision IDE

2. SST FlashFlex Programming tool

3. 8051 Development board.

**THEORY:**

Microcontroller is nothing but On-chip computer. It contains ALU, PC, SP, set of registers, RAM, ROM, I/O on a single chip. The prime use of microcontroller is to control the operation of a system in proper sequence using the program stored on ROM which does not change throughout the lifetime of the system.

8051 is Microcontroller and its basic features are :

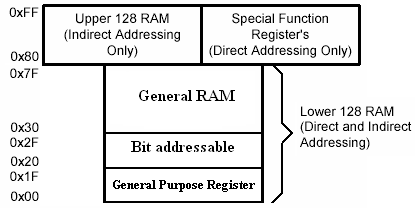
* 8 bit ALU and 16 bit address lines.
* 4K on chip ROM
* 128 bytes on chip Data RAM.
* 64K PM and 64K DM (External)
* Operating frequency = 10MHz
* 4 IO ports (32 bidirectional IO lines)
* Two 16 bit timers
* Full duplex UART
* On-Chip clock oscillator



**Fig.1 : 8051 Architecture**

The 8051 memory is divided into the following 4 physical parts

1. Internal RAM
2. Internal special function registers
3. External RAM
4. Internal and external ROM



**Fig.2 Internal RAM**

**Internal RAM:** Internal memory organization is shown in fig.1. It is divided into three distinct areas:

1) 32 bytes from address 00H to 1FH that make up 32 working registers organized as 4 banks of eight registers each. The 4 registers banks are numbered 0 to 3 and are made up of 8 registers named R0 to R7. Each register can be addressed by name or by its RAM address. Bits RS0 and RS1 in the PSW determine which bank of registers is currently in use at any time when the program is running. Bank 0 is selected on reset.

7 6 5 4 3 2 1 0

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| CY | AC | F0 | RS1 | RS0 | OV | ------ | P |

**The Program Status Word (PSW) Special Function Register**

|  |  |  |
| --- | --- | --- |
| Bit | Symbol | Function |
| 7 | CY | Carry flag |
| 6 | AC | Auxiliary Carry flag |
| 5 | F0 | User flag 0 |
| 4 | RS1 | Register bank select bit 1 |
| 3 | RS0 | Register bank select bit 0 |
|  | RS1 RS0 |  |
|  | 0 0 | Select register bank 0 |
|  | 0 1 | Select register bank 1 |
|  | 1 0 | Select register bank 2 |
|  | 1 0 | Select register bank 3 |
| 2 | OV | Overflow flag |
| 1 | ---- | Reserved for future use |
| 0 | P | Parity flag |

**Bit addressable as PSW.0 to PSW.7**

2) A bit addressable area of 16 bytes occupies RAM byte addresses 20H to 2FH forming a total of 128 addressable bits. An addressable bit may be specified by its bit address of 00H to 7FH or byte address from 20H to 2FH. Addressable bits are useful when the program need only remember a binary event.

3) A general purpose RAM area above the bit area from 30H to 7FH, addressable as bytes.

**8051 I/O Ports**

* + - 4 I/O pots – 32 Lines
    - All ports are output ports. To make them as I/P port, 1 has to be written on all pins of ports.
    - Port 1 (8 pins) has internal pull up registers.
    - Port 2 (8 pins) also has internal pull up registers and acts as a higher order address bus.
    - Port 0 does not have pull up registers. It also used as address or data bus. In case of external memory, port 0 provides both address and data. ALE used for de-multiplexing it.
    - Port 3 is 8 bit port and used as for other functionality.
* P 3.0 - RxD (Serial Data Reception)
* P 3.1 - TxD (Serial Data Transmission)
* P 3.2 - INT0 (Interrupt 0)
* P 3.3 - INT1 (Interrupt 1)
* P 3.4 - T0 (Timer 0)
* P 3.5 - T1 (Timer 1)
* P 3.6 - WR (Write)
* P 3.7 - RD (Read)

**Microcontroller 8051 assembly language programming tools:**

1. **Introduction to 8051 Microcontroller kit:**

SBC51 is a very user friendly and easy to use development kit for 8051 series of microcontrollers. It is designed keeping in mind the various requirements of industry and educational sector, thereby making it an extremely useful kit for students and industry professionals alike.

SBC51 development kit provides the user a basic 8051 microcontroller interfaced to various devices like LED's, 16x2 character LCD, 7 Segment Display, 4x4 Matrix Keypad, Stepper motor, 8 Bit single channel Analog to Digital Controller, Relay , I2C EEPROM, I2C RTC and a Serial Communication interface. It also has a provision to externally interface peripheral devices like 8 Bit Digital to Analog Controller, external memory, sensors and controller circuitry.

The SBC51 is powered by a 5V DC wall mount adapter having a current capacity of 1.5 Amps. After POWER ON a RED colored LED (LD10) will indicate a proper power supply. A Power ON-OFF slide switch is provided for the user. Its versatile design and programmable microcontroller lets user access numerous peripheral devices and program the board for multiple use. The board has many I/O connectors and supports various 8051 variants. It supports programming options including 8051 assembly and C. Flash programming for the microcontroller chip on the SBC51 is accomplished via the serial port with the help of Flash Magic Software.

**Features-**

* The SBC51 has the following features on-board,
* Philips P89V51RD2 as 8051 Microcontroller Unit (MCU) running at 12MHz
* A 16x2 character LCD
* 7 Segment Display
* Eight LEDs
* A 4x4 Matrix Keypad
* Analog to Digital Controller (single channel, 8-bit)
* Stepper motor interface
* EEPROM (256K) interfaced using I2C
* I2C based RTC
* Relay
* External Interrupt switch
* One RS232 compatible Serial port with DB9 connector
* A 26-pin and a 14-pin connectors for general purpose I/O using port P0, P1, P2 and P3
* Eight DIP switches for keypad selection logic
* Power indication LED
* RESET switch to reset the board

**PC Connection setup for SBC51 :**

The SBC51 connects to the PC via the serial port. The Hex files generated by the Keil IDE can be downloaded into the flash memory using the ISP feature on the microcontroller via the serial port. The SBC51 development kit has a serial port cable included in the package. Connect the female side of the connector to the PC and the male side to the SBC51.

1. **Software Programming Tools :**

SBC51 is a development kit that can be used by students and professionals alike. The SBC51 is generally provided with a microcontroller from Philips (P89V51RD2). This is a 8051 variant from Philips. The programs for the microcontroller can be written in Keil IDE. After Compiling a hex file will be generated for the microcontroller which can be downloaded into the internal Flash. The SBC51 has a provision for programming the on-chip flash via the serial port (In System Programming). The on-chip boot loader communicates with a flash programming utility (Flash Magic) on the host (PC) side to program the flash.

**Keil Programming IDE :**

Keil IDE is the most preferred development environment for 8051 series of microcontrollers. The IDE provides a very easy to use user interface for developing the programs in assembly and C languages. Following are the steps involved for developing a program for the SBC51.

Creating a New Project.

1. Open the Keil μVision IDE.

2. Go to the Project Tab.

• Left Click on New μVision Project. Name the project and save it in your preferred

location.

• Choose the CPU (NXP->P89V51RD2). And press ENTER.

It will ask the user if it should Copy Standard 8051 Startup code to project folder and

Add File to Project.

• Click Yes if you are developing a project in C language and No if you are developing

a project in Assembly language.

3. Go to File Menu and left click New.

4. Save the file in the folder created for the project. If the program is in C then put the

extension as .c and if the program is in assembly then save the file with .a51 extension.

5. Then right click on source group1 and click Add Files to Group source group1.

6. Select the file and press ADD. The file get added then press Close.

7. Now you can start writing you programs.

**Project Settings.**

8. Go to Project Tab and click Options for Target 'Target1'.

• In the Target tab set the crystal frequency as 12.0.

• In the Output tab Tick Create Hex File. Also select the folder where you want to save

the Hex file.

**Compile the Program**.

9. When the program is ready and all the settings are done then compile the program.

Go to Project tab and click Build Target.

1. **Flash Magic Flash Programming Tool :**

Once the HEX file is generated for the current target microcontroller, the next step is to program the flash. Following are the steps for programming the HEX file into the microcontroller.

1) Connect the SBC51 to the PC using the serial cable.

2) Open Flash Magic flash programming tool.

In the Step1:Communication Section.

3) Select the device as 89V51RD2.

4) Set COM Port as COM1.

5) Set Baud rate to 9600.

6) Set Interface as None(ISP).

In the Step2:Erase Section.

7) Untick the Erase all Flash option and Tick the Erase block used by Hex File option.

In Step3:Hex File Section.

8) Browse the hex file you want to program into the flash.

In the Step4:Options Section.

9) Untick all options.

**Very Important Note:**

1. In the main TAB near the ISP we have the Options tab.

2. In the Options Tab go to Advanced Options.

3. In the Advanced options tab go to Hardware Config Tab.

4. Untick the Use DTR to Control RST option and press OK tab.

Once these settings are done you can press the Start button on Flash magic and simultaneously press the reset button on the SBC51. This sequence has to be followed every time for programming the flash.

**ALGORITHM :**

1. Initialize counter to number of array elements.
2. Initialize pointer to the first element of an array.
3. Initialization of accumulated sum.
4. Addition of an elements to accumulated sum.
5. After each addition check whether carry is generated, if yes then increment carry counter by one.
6. Increment memory pointer by one to point to the next element.
7. Decrement count with each addition.
8. Go on adding numbers in internal memory till counter becomes zero. If not then go to step 4.
9. Display the result of addition on LEDs or R6 and R7

**CALCULATIONS :**

**INPUT:** n 8 bit numbers stored in internal memory

**OUTPUT:** Result of addition of n 8 bit numbers displayed on LEDs or in R6 and R7

**CONCLUSION:**