**SSN College of Engineering**

**Department of Computer Science and Engineering**

**UCS1512 – Microprocessors Lab**

**EX:12 – 8-bit arithmetic operations using 8051**

Exp No: 12 Name : Kshitij Sharma

Date: 22/10/2020 Reg No: 185001080

**AIM:**

To implement 8 bit arithmetic operations using 8051.

**8 BIT ADDITION:**

**ALGORITHM:**

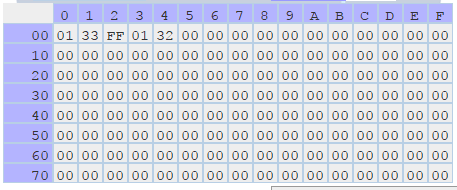
* Move 00H to R0.
* Move the value in R1 to register A.
* Add register A and the value in the R2. The result is stored in register A.
* Jump to LABEL if there is no carry.
* Else increment value in R0.
* LABEL: Move the register A’s value to R4 and move the value in R0 to R3.
* HERE: Infinite loop to HERE using SJMP HERE.

|  |  |  |
| --- | --- | --- |
|  | **Program** | **Comments** |
| START: | MOV R0, #00 | R0 <- 00H |
| MOV A, r1 | A <- R1 |
| ADD A, r2 | A <- A + R2 |
| JNC LABEL | Jump to LABEL, if no carry. |
| INC R0 | R0++ |
| LABEL: | MOV r4, a | R4 <- A |
| MOV 03, R0 | R3 <- R0 |
| HERE: | SJMP HERE | Transfers execution to HERE. |

**SNAPSHOT OF SAMPLE I/O:**

**R1=33 R2=FF**

**OUTPUT: R3=01 R4=32**



**8 BIT SUBTRACTION:**

**ALGORITHM:**

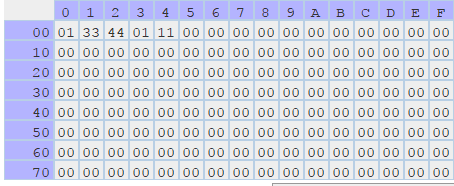
* Move 00H to R0.
* Clear the carry flag using CLR C.
* Move the value in R1 to register A.
* Sub register A with the value in the R2 using SUBB A, R2. The result is stored in register A.
* Jump to LABEL if there is no carry.
* Else increment value in R0 for carry. Take 2’s compliment of register A by using the instructions CPL A and INC A.
* LABEL: Move the register A’s value to R4 and move the value in R0 to R3.
* HERE: Infinite loop to HERE using SJMP HERE.

|  |  |  |
| --- | --- | --- |
|  | **Program** | **Comments** |
| START: | MOV R0, #00 | R0 <- 00H |
| CLR C | Clear the carry flag. |
| MOV A, r1 | A <- R1 |
| SUBB A, r2 | A <- A - R2 - C where C is Carry Flag |
| JNC LABEL | Jump to LABEL, if no carry. |
| INC R0 | R0++ |
| CPL A | Complient the value in Register A. |
| INC A | Increment A to get 2’s compliment. |
| LABEL: | MOV r4, a | R4 <- A |
| MOV 03, R0 | R3 <- R0 |
| HERE: | SJMP HERE | Transfers execution to HERE. |

**SNAPSHOT OF SAMPLE I/O:**

**R1=33 R2=44**

**OUTPUT: R3=01 R4=11**



**8 BIT MULTIPLICATION:**

**ALGORITHM:**

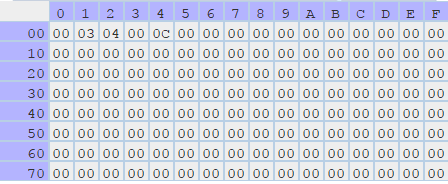
* Move the value in R1 to register A.
* Move the value in R2 to register B.
* Multiply A and B using MUL AB (BA <- A x B).
* Move the register A’s value to R4 and move the Register B's value to R3.
* HERE: Infinite loop to HERE using SJMP HERE.

|  |  |  |
| --- | --- | --- |
|  | **Program** | **Comments** |
| START: | MOV A, R1 | A <- R1 |
| MOV B, R2 | B <- R2 |
| MUL AB | BA = A x B |
| MOV R3, B | R3 <- B |
| MOV R4, A | R4 <- A |
| HERE: | SJMP HERE | Transfers execution to HERE. |

**SNAPHOT OF SAMPLE I/O:**

**Input:R1=03 R2=04**

**Output:R3=00 R4=0C**



**8 BIT DIVISON:**

**ALGORITHM:**

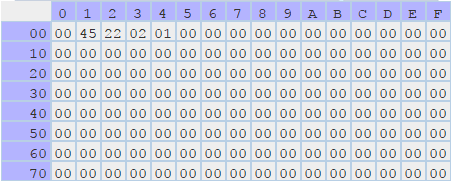
* Move the value in R1 to register A.
* Move the value in R2 to register B.
* Divide A by B using DIV AB with Quotient in A and Remainder in B.
* Move the register A’s value to R3 and move the Register B's value to R4.
* HERE: Infinite loop to HERE using SJMP HERE.

|  |  |  |
| --- | --- | --- |
|  | **Program** | **Comments** |
| START: | MOV A, R1 | A <- R1 |
| MOV B, R2 | B <- R2 |
| DIV AB | A / B; Quotient in A, Remainder in B |
| MOV R3, A | R3 <- A |
| MOV R4, B | R4 <- B |
| HERE: | SJMP HERE | Transfers execution to HERE. |

**SNAPHOT OF SAMPLE I/O:**

**Input:R1=45 R2=22**

**Output:R3=02(Quotient) R4=01(Remainder)**



**RESULT:**

Thus, 8 bit arithmetic operations have been successfully implemented using 8051.