Experiment No : 10

Aim : Arithmetic Operations

Procedure :

Write programs that perform the tasks listed in 8051. Use comments on each line of code and try to use as few lines as possible. All numbers may be considered to be unsigned numbers.

l. Add the bytes in RAM locations 34h and 35h: put the result in register R5 (LSB) and R6 (MSB).

2- Add the bytes in registers R3 and R4; put the result in RAM location 4Ah (LSB) and 4Bh (MSB).

3. Add the number 84h to RAM locations l 7h and 18h.

4. Add the byte in external RAM location 02CDh to internal RAM location l9h; put the result into external RAM location 0000h (LSB) and 00C1h (MSB).

5-8. Repeat Problems 1-4, assuming the numbers are in BCD format.

9. Subtract the contents of R2 from the number F3h; put the result in external RAM location 028Bh.

10. Subtract the contents of R l from R0; put the result in R7.

11. Subtract the contents of RAM location 13h from RAM location 2Bh: put the result in RAM location 3Ch.

12. Subtract the contents of TH0 from TH I: put the result in TL0.

13. Increment the contents of RAM location 13h, 14h and 15h using indirect addressing only.

14. Increment TLI by I Oh.

15. Increment external RAM locations 0100h and 0200h.

16. Add a 1 to every external RAM address from 00h to 06h.

17. Add a 1 to every external RAM address from 0100h to 0106h.

18. Decrement TL0. THO, TL1, and TH l.

19. Decrement external RAM locations 0l23h and 01 BDh.

20. Decrement external RAM locations 45h and 46h.

21. Multiply the data in RAM location 22h by the data in RAM location l5h; put the result in RAM locations 19h (low byte) and lAh (high byte).

Programs :

1.

```
MOV R6,#0
MOV R5,#0
MOV A,34h
ADD A,35h
JNC NOC
INC R6
NOC: MOV R5,A
```

2.

```
MOV 4Ah,#0
MOV 4Bh,#0
MOV A,R3
ADD A,R4
JNC NOC
INC 4Bh
NOC: MOV 4Ah,A
```

3.

```
MOV A, 84h
ADD A,17h
MOV 17h,A
MOV A, 84h
ADD A,18h
MOV 18h,A
```

4.

```
MOV R5,#0 ;local MSB
MOV DPTR, #02CDH
MOV A, #0
MOVX A, @DPTR ;get external value
ADD A, 19h
JNC NOC
INC R5 ;increment local msb
NOC:MOV DPTR, #0000h
MOVX @DPTR,A ;Move lsb to external memory
MOV DPTR, #00C1h
MOV A,R5
MOVX @DPTR, A ; move MSB to external memory
```

5.

```
MOV R6,#0
MOV R5,#0
MOV A,34h
ADD A,35h
DA A
JNC NOC
INC R6
NOC: MOV R5,A
```

6.

```
MOV 4Ah,#0
MOV 4Bh,#0
MOV A,R3
```

```
ADD A,R4
DA A
JNC NOC
INC 4Bh
NOINV: MOV 4Ah,A
```

7.

```
MOV A, 84h
ADD A,17h
DA A
MOV 17h,A
MOV A, 84h
ADD A,18h
DA A
MOV 18h,A
```

8.

```
MOV R5,#0 ;local MSB
MOV DPTR, #02CDH
MOV A, #0
MOVX A, @DPTR ;get external value
ADD A, 19h
DA A
JNC NOC
INC R5 ;increment local msb
MOV DPTR, #0000h
MOVX @DPTR,A ;Move lsb to external memory
MOV DPTR, #00C1h
MOV A,R5
MOVX @DPTR, A ; move MSB to external memory
```

9.

```
CLR C
MOV A,R2
SUBB A,#0f3h
JNC NOC
CPL A
INC A
NOC: Mov DPTR, #028Bh
MOVX @DPTR, A
```

10.

```
CLR C
MOV A,R1
SUBB A,R0
JNC NOC
CPL A
INC A
NOC: MOV R7,A
```

11.

```
CLR C
MOV A,13h
SUBB A,2bh
JNC NOC
```

```
CPL A
INC A
NOC: MOV 3ch,A
```

12.

```
CLR C
MOV A,TH1
SUBB A, TH0
JNC NOC
CPL A
INC A
NOC: MOV TL0,A
```

13.

```
CLR C
MOV R5, #2; counter
MOV R1, #13h; Address register
MOV A,@R1 ; Move first value
LOOP:
INC R1
SUBB A, @R1
JNC NOC
CPL A
INC A
DJNZ R5, LOOP
```

14.

```
ADD TL1, #10h
```

15.

```
MOV DPTR, #0100h
ACALL EXTINC
MOV DPTR, #0200h
ACALL EXTINC
JMP STOP
; increment external memory location
EXTINC:
MOVX A,@DPTR
INC A
MOVX @DPTR, A
RET
STOP:NOP
```

16.

```
MOV R2, #06h ;counter
MOV R0, #0
LOOP:
MOVX A,@R0
INC A
MOVX @R0, A
INC R0
DJNZ R2, LOOP
```

17.

```
MOV R2, #06h ;counter
MOV DPTR, #0100h
LOOP:
MOVX A,@DPTR
INC A
MOVX @DPTR, A
INC DPTR
DJNZ R2, LOOP
```

18.

```
DEC TL0
DEC TL1
DEC TH0
DEC TH1
```

19.

```
MOV DPTR, #0123h
ACALL EXT_DEC
MOV DPTR, #01BDh
ACALL EXT_DEC
JMP STOP
; decrement external memory location
EXT_DEC:
MOVX A,@DPTR
DEC A
MOVX @DPTR, A
RET
STOP:NOP
```

20.

```
MOV R0, #45h
ACALL EXT_DEC
MOV R0, #46h
ACALL EXT_DEC
JMP STOP
; decrement external memory location
EXT_DEC:
MOVX A,@R0
DEC A
MOVX @R0, A
RET
STOP:NOP
```

21.

```
CLR C
MOV B,15h; multiplicand
MOV A, 22h; multiplier
MUL AB
MOV 19h, A
JB OV ,NOC
MOV 1Ah, B
NOC:NOP
```

22.
```
CLR C
MOV A, R5; multiplier
MOV B, R5; multiplier
MUL AB
MOV R1,A
JB OV ,NOC
MOV R0, B
NOC:NOP
```

23.
```
CLR C
MOV A, 3Eh ; dividend
MOV B, #12h; divisor
DIV AB
MOV R4, A
MOV R5, B
```

24.
```
CLR C
MOV R0, #7ch; external ram location
MOV A, 15h ; dividend
MOV B, #16h; divisor
MOV @R0, A
```

25.
```
;;;;DIVISION STEP
CLR C
MOV A, 13h ; dividend
MOV B, #14h; divisor
DIV AB
MOV R5, B ; Move remainder to R5
CLR C
MOV B, #14h; get multiplier
MUL AB
ADD A, R5 ;add remainder
MOV 13h, A ; restore 13h
```

26.
```
ORG 0H
MOV DPTR, #MYDATA
MOV R4, #0h; counter
MOV R2, #0 ;clear
MOV R3, #0 ;clear
LOOP:
CLR C
MOV A, R4
MOVC A, @A+DPTR
JZ STOP ; If we have reached the end of MYDATA stop
ADD A, R2
JNC NOC
INC R3
NOC:
MOV R2, A
INC R4
```

```
SJMP LOOP
STOP:NOP
ORG 250H
MYDATA: DB 3, 94, 56, 92, 74, 65, 43, 23, 83, 0 ;Add extra zero to indicate end
```

27.

```
ORG 0H
MOV DPTR, #MYDATA
MOV R5, #09h; remaining counter
MOV R4, #0h; upcounter
MOV R2, #0 ;clear
MOV R3, #0 ;clear
LOOP:
CLR C
MOV A, R5;
JZ STOP ; If we have reached the end of MYDATA stop
DEC R5
MOV A, R4
MOVC A, @A+DPTR
ADD A, R2
JNC NOC
INC R3
NOC:
MOV R2, A
INC R4
SJMP LOOP
STOP:NOP
ORG 250H
MYDATA: DB 1,8,1,1,0,5,0,1,0 ;Add extra zero to indicate end
```

28.

```
MOV A, 84h
ADD A,17h
DA A
MOV 17h,A
MOV A, 84h
ADD A,18h
DA A
MOV 18h,A
```

29.

```
;;; Write values
MOV R0, #40h; Address register
MOV R1, #0Fh ; Counter
LOOP_WRITE:
MOV @R0, #55h
INC R0
DJNZ R1, LOOP_WRITE
;;; Add values
MOV 61h, #0; Zero higher bit
MOV R0, #40h; Address register
MOV R1, #0Fh ; Counter
MOV A, #0;
LOOP_ADD:
ADD A, @R0
```

```
JNC NOC
CLR C
INC 61h
NOC:
DJNZ R1, LOOP_ADD
MOV 60h, A
```

30.

```
ORG 0h
MOV R5, #3h; Counter
MOv R4, #0; offset register
LOOP:
;; get FIRST
MOV DPTR, #FIRST
MOV A, R4
MOVC A, @A+DPTR
MOV R0,A
;;Get SEC
MOV DPTR, #SEC
MOV A, R4
MOVC A, @A+DPTR
;;Add
ADD A, R0
JNC NOC
CLR C
INC 41h
NOC:
INC R4
DJNZ R5, LOOP
MOV 40h, A
ORG 300H
FIRST: DB 9Ah, 7Fh,89h
SEC: DB 48h, 0BCh,34h
```

Conclusion : Given 8051 programs were successfully written and executed

Deepraj Bhosale

181105016