इरिसेट
माइक्रोकंट्रोलर प्रयोगशाला
प्रयोग नं: एम सी 1

IRISET
Microcontroller Laboratory
EXPERIMENT NO : MC-1

नाम
Name : ________________

अनुक्रमांक
Roll No : ________________

प्राप्त अंक
Marks Awarded:

पाठ्यक्रम
Course : ________________

दिनांक
Date : ________________

अनुदेशक का अधाक्षर
Instructor Initial:

Experiment No. MC-1

Familiarization with 8051 Microcontroller Kit

Objective:

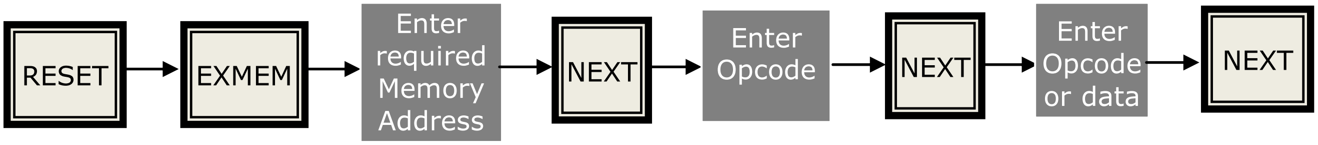
The main objective of this practical session is to introduce to the basic functions needed for practicing programming on microcontroller kits

Description of 8085 Basic Functions:

The following basic functions are required to be known by every trainee before being able to practice programming on 8051 kit.

1. Storing Program In Memory

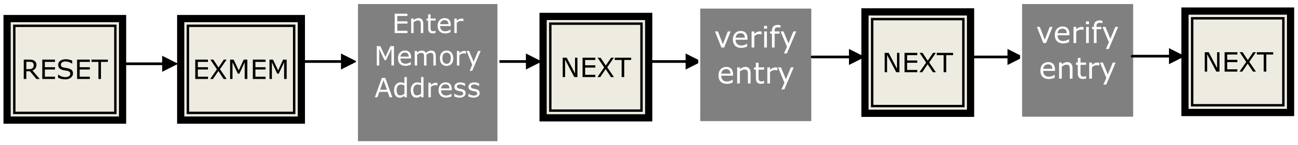
a) Press the following keys in the given sequence



b) When Reset key is pressed the display should be **‘UP 51’**
c) Even after entry of last opcode or data byte, the NEXT key must be pressed
d) That means this process should end with the pressing of NEXT key.
e) As an exercise enter the following data starting from 2000 memory register
f) 25, F6, 56, 75, 57, 99, 01

g) Then, verify the correctness of data entered into memory, by the below given key-sequence.

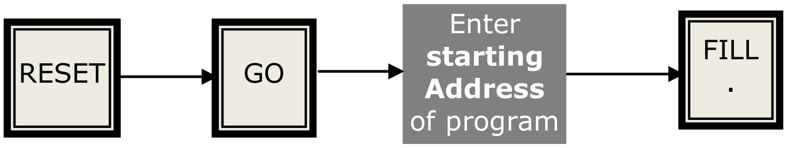
h) After every NEXT key press the entries can be verified.



2. Execution of Program

a) Entire Program Execution At One Go

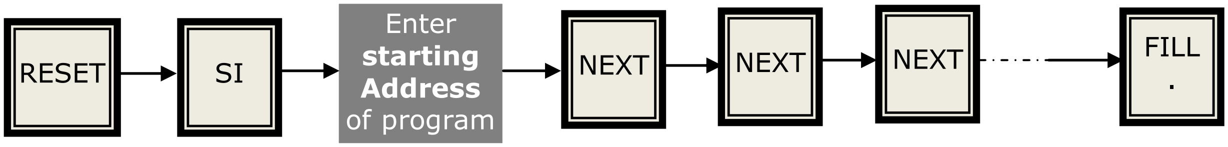
1. In this mode of **program execution** the entire program is executed at one stretch
2. For the execution of a program that is stored in memory by the above sequence of steps follow the sequence shown below



3. After the above sequence of operations the display shows **‘E’** which means the program is **executed** or execution is going on.

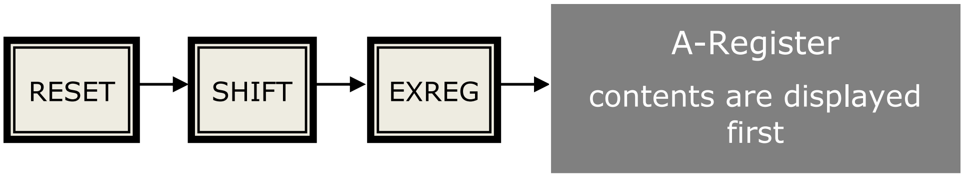
b) Single Instruction/Step (SI) Mode of Operation

1. In this mode the program is executed in steps of single instruction at a time. The below given key sequence is used for execution of program
2. After the execution of every instruction the NEXT key has to be pressed for going to the execution of next instruction or it can be terminated at that point itself with FILL key. Purpose of this process is for debugging a program in case of any errors.



3. To Examine the Contents of Internal Registers of 8051

a) Use the fallowing sequence to check the contents of any internal register of µC.



b) Contents of A - register appear on the display

c) By pressing NEXT key other next register contents can be examined one by one.

d) After executing a program in SI mode the contents of internal registers are affected.

Exercise:

1) Perform the following functions

   a) Enter the below given program-1 into memory from location 2000H onwards. Execute it as per the sequence given in function procedure **2 (b)**.

   b) Then verify the correctness of contents of each internal register of the microcontroller using EXREG function, the function sequence **3**.

Program-1:

```
        MOV   A,    #25          ;
        MOV   DPTR, #2250        ;
        MOV   R0,   #41          ;
        MOV   R1,   #55          ;
        MOV   R2,   #62          ;
        MOV   R3,   #22          ;
        MOV   R4,   #33          ;
        MOV   R5,   #44          ;
        MOV   R6,   #55          ;
        MOV   R7,   #66          ;
        MOV   F0,   #77          ; F0   means  B
        MOV   81,   #88          ; 81   means  SP
        MOV   88,   #99          ; 88   means  TCON
        MOV   89,   #AA          ; 89   means  TMOD
        MOV   D0,   #45          ; D0   means  PSW
        MOV   98,   #23          ; 98   means  SCON
        MOV   A8,   #CC          ; A8   means  IE
        MOV   B8,   #EE          ; B8   means  IP
        MOVX  @DPTR, A           ;
end:    SJMP  end                ; End of the program
```

2) The below given is a program for the ADDITION of two 8 bit values which are assumed to be available at 2050 and 2051 memory locations. Load this program (and

also any two 8 bit numbers at 2050 and 2051) and execute it. Verify and record the results for different 8bit values, as shown in table-1 below.

Program-2:

```
        MOV  DPTR,  # 2050  ; Point  to the 1st value  in memory
        MOV    R1, #00      ; Reserve Register  for  storing  Carry Flag
        MOVX   A,  @DPTR    ; Copy  value  at  2050  into  A – register
        MOV   R2,  A        ; Copy  this value into R2
        INC    DPTR         ; increment address in DPTR  to point to 2nd value.
        MOVX   A,  @DPTR    ; Copy  value  at  2050  into  A – register
        ADD   A, R2         ; add  R2  value  to  A
        JNC  Skip           ; if no carry go to skip
        INC   R1            ; increment  B  if   there  is  a carry
Skip:   MOV  DPTR   #2060   ; load new address in DPTR
        MOVX   @DPTR, A     ; store result at 2060 memory register
        INC    DPTR         ; point  to memory register  2061
        MOV   A,  R1        ; get carry into A
        MOVX   @DPTR, A     ; store carry at 2061 memory register
end:    SJMP   end          ; stop
```

S.No	1st value	2nd value	Sum @2060	Carry@2061
1	22	72		
2	98	83		
3				
4				

Table-1

Review Questions:

1) What is IP register in 8051? Could you find out the contents of it with EXREG function?
2) In the 1st program above, what is the data that is transferred with the instruction ‘ MOVX @DPTR, A’ and where can you find it ?

इरिसेट

माइक्रोकंट्रोलर प्रयोगशाला

प्रयोग नं : एम सी 2

IRISET

Microcontroller Laboratory

EXPERIMENT NO : MC-2

नाम
Name : ________________

अनुक्रमांक
Roll No : ________________

प्राप्त अंक
Marks Awarded:

पाठ्यक्रम
Course : ________________

दिनांक
Date : ________________

अनुदेशक का अधाक्षर
Instructor Initial:

Experiment No. MC- 2

Addition of Numbers of an Array

Objective: Summing up of all the numbers of an array using 8051

Program: Assume that the array starting address is **2050** and size is **Ten** bytes.

```
        MOV  DPTR, # 2050  ; Load  array  starting address into DPTR
        MOV    R1, #00     ; Reserve a Register  for  storing  Carry Flag
        MOV    R2, #09     ; Load array count  (n-1) into R2
        MOVX   A, @DPTR    ; Move 1st byte of array into  A - register
        MOV    R3,  A      ; shift  this value into R3
Rpt:    INC    DPTR        ; increment address in DPTR  to point to next value.
        MOVX   A, @DPTR    ; Bring  next value of array into  A
        ADD   A, R3        ; add  both the values
        JNC  Skip          ; if no carry go to ‘Skip’
        INC    R1          ; increment  R1  if   there is a carry
Skip:   MOV   R3, A        ;copy the sum into R3
        DJNZ  R2,  Rpt     ;decrement  R2 and jump to ‘Rpt’ if count  is not zero
        MOV  DPTR,  #2060  ; load new address in DPTR
        MOVX   @DPTR, A    ; store result at 2060 memory register
        INC    DPTR        ; point  to memory register  2061
        MOV   A,  R1          ; get carry into A
        MOVX  @DPTR, A        ; store carry at 2061 memory register
end:    SJMP  end             ; stop
```

Procedure:

1) Load the above program into memory in the 8051 kit, starting at 2000.
2) Also load an array of ten byte into memory, starting at 2050.
3) Execute the program in SI (single instruction) mode.
4) And verify the result and carry values at 2060 and 2061 respectively.

Exercise:

1) Increase the array size to 20. Execute the program and verify the results.

Review Questions:

1) Why the count value 09 is used (loaded into R2), in the above program?

2) What can be the maximum size of the array that can be handled by the above program? For arrays above this maximum size, what changes are needed in the program?

इ रि से ट

माइक्रोकंट्रोलर प्रयोगशाला

प्रयोग नं : एम सी 3

IRISET

Microcontroller Laboratory

EXPERIMENT NO : MC-3

नाम
Name : _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

अनुक्रमांक
Roll No : _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

प्राप्त अंक
Marks Awarded:

पाठ्यक्रम
Course : _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

दिनांक
Date : _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

अनुदेशक का अधाक्षर
Instructor Initial:

Experiment No. MC- 3

Flashing ‘IRISET’ on the Display

Objective: To show the letters IRISET flashing on the display of the kit

Program: Assume that the data for displaying characters IRISET is stored at 2050 and data for a blank display at 2060

```
Start:  MOV   DPTR, #2050       ; Character data address
        LCALL 06F7              ; Call display sub routine at 06F7
        MOV   R1, #FF           ; Flashing delay values
  Lp2:  MOV   R2, #FF           ;
  Lp1:  DJNZ  R2, Lp1           ;
        DJNZ  R1, Lp2           ;
        MOV  DPTR, #2060        ; Blank data address
        LCALL 06F7              ; Call display sub routine
  Lp4:  MOV   R2, #FF           ; Load array count into R2
  Lp3:  DJNZ  R2, Lp3           ;
        DJNZ  R1, Lp4           ;
  SJMP  Start                   ; continue the same
```

Data for Character Display:

```
        2050:       9F  F5  9F  49  61  E1
        2060:       FF  FF  FF  FF  FF  FF
```

Procedure:

1) Load the above program into memory in the 8051 kit starting at 2000.
2) Also load the given data for display of IRISET at 2050 and 2060.
3) Execute the program.
4) Now , look for a flashing IRISET on the display

Exercise:

1) Display the name of your Railway zone, (for example: SC RLY) using the same program.
2) Try to decrease the flashing rate of the display by suitably changing the program.

Review Questions:

1) Did you succeed in displaying the name of your railway? If yes, what is the data that you have entered at 2050?
2) What changes did you make in the program for decreasing flashing rate?

इरिसेट
माइक्रोकंट्रोलर प्रयोगशाला
प्रयोग नं : एम सी 4

IRISET
Microcontroller Laboratory
EXPERIMENT NO : MC-4

नाम
Name : ________________

अनुक्रमांक
Roll No : ________________

पाठ्यक्रम
Course : ________________

दिनांक
Date : ________________

प्राप्त अंक
Marks Awarded:

अनुदेशक का अधाक्षर
Instructor Initial:

Experiment No. MC- 4

Finding Out Even and Odd numbers

Objective: To find out whether an 8 bit number is odd or even and also to indicate the same through the display of the kit

Program:

```
       MOV   81, #65          ; Initialize stack
       LCALL  073E            ; Call key- board sub-routine at  073E
       INC   DPTR             ;
       MOVX   A, @DPTR        ;
       ORL   A, #FE           ;
       CJNE A,  #FE, Odd      ; check  LS bit,  if set output Odd
       MOV   DPTR, #201A      ; Point to display  Even
       SJMP  Out                  ;
Odd:   MOV  DPTR, #2020       ;  Point to display  Odd
Out:   LCALL  06F7            ; Call  display sub- routine
end:   SJMP  end              ; Stop
```

Data for displaying Even or Odd:

```
201A:    61  83  61  D5  FF  FF
2020:    03  85  85  FF  FF  FF
```

Procedure:

1) Load the above program into memory in the 8051 kit starting at 2000.
2) Also load the display data for Even/Odd at 201A and 2020.
3) Execute the program.
4) Then enter a 2-digit number of your choice and press the FILL key.
5) On the display appears either ‘Even’ or ‘Odd’ depending on the value entered by you.

Exercise:

1) Make changes in the above program/data to get display of Even and Odd in reverse order. That is NEVE for Even and DDO for odd.
2) Similarly get the display of your **Batch No**. for Even and **Roll No**. for Odd values, by suitably changing the program/data.

Review Questions:

1) Have you got the display of EVEN and ODD in reverse order? If yes how?
2) Write down the data that you have used for the display of RESET and SET in the above exercise.
3) Examine and write down the codes of both key board subroutine, available at 073E, and display subroutine, at 06F7 memory locations, in the kit.