**SSN College of Engineering**

**Department of Computer Science and Engineering**

**UCS1512 – Microprocessors Lab**

**End Semester Practical**

**Name : Kshitij Sharma Date: 16/11/2020 Reg No: 185001080**

**Question 1:**

**Aim:**

Write an ALP in 8086 to convert HEX TO BCD.

**Algorithm:**

* + Move the data segment to the AX register and then move it to the DS register.
  + Move HEX value to AL register.
  + Move 64h(100 in decimal) to CL register.
  + Move 00h to AH register.
  + Divide AX by BL.
  + The quotient in AL is now moved to DH register.
  + Move the remainder in AH to AL.
  + Move 00h to AH.
  + Move 0Ah(10 in decimal) to CL.
  + Divides AX by CL.
  + Left rotate the quotient in AL by 4 bits.
  + Bitwise OR of AL and AH.
  + Move AL to DL.
  + Move DX to BCD.

**Program:**

mov ax,data

mov ds,ax

mov al,hex

mov ah,00h

mov cl,064h

div cl

mov dh,al

mov al,ah

mov ah,00h

mov cl,00ah

div cl

mov cl,004h

rol al,cl

mov dl,ah

or dl, al

mov bcd,dx

mov ah,4ch

int 21h



**Result:**

An ALP using 8086 to convert HEX to BCD was implemented.

**Question 2:**

**Aim:**

Write an ALP in8051 to find number of odd and even umbers.

**Algorithm:**

* Mov the initial address to R0.
* Mov the size of the array to R1.
* Set R2 to 0 use as even count register.
* Set R3 to 0 use ad odd count register.
* Mov value of R0 in A.
* The RRC instruction rotates the eight bits in the accumulator and the one bit in the carry flag right one bit position.
* Jump if no carry to even .
* In even increment R2 register and increment R0 register to next position.
* In carry mov to odd inc R3 and R0.
* Short jump to l
* In l decrement R1 and jump to l.

**Program:**

MOV R0,#10h ; Addr

MOV R1,#05h ; Size of Arr

MOV R2,#00h ; Even

MOV R3,#00h ; Odd

loop1:

MOV A,@R0

RRC A

JNC even

odd:

INC R3

INC R0

SJMP l

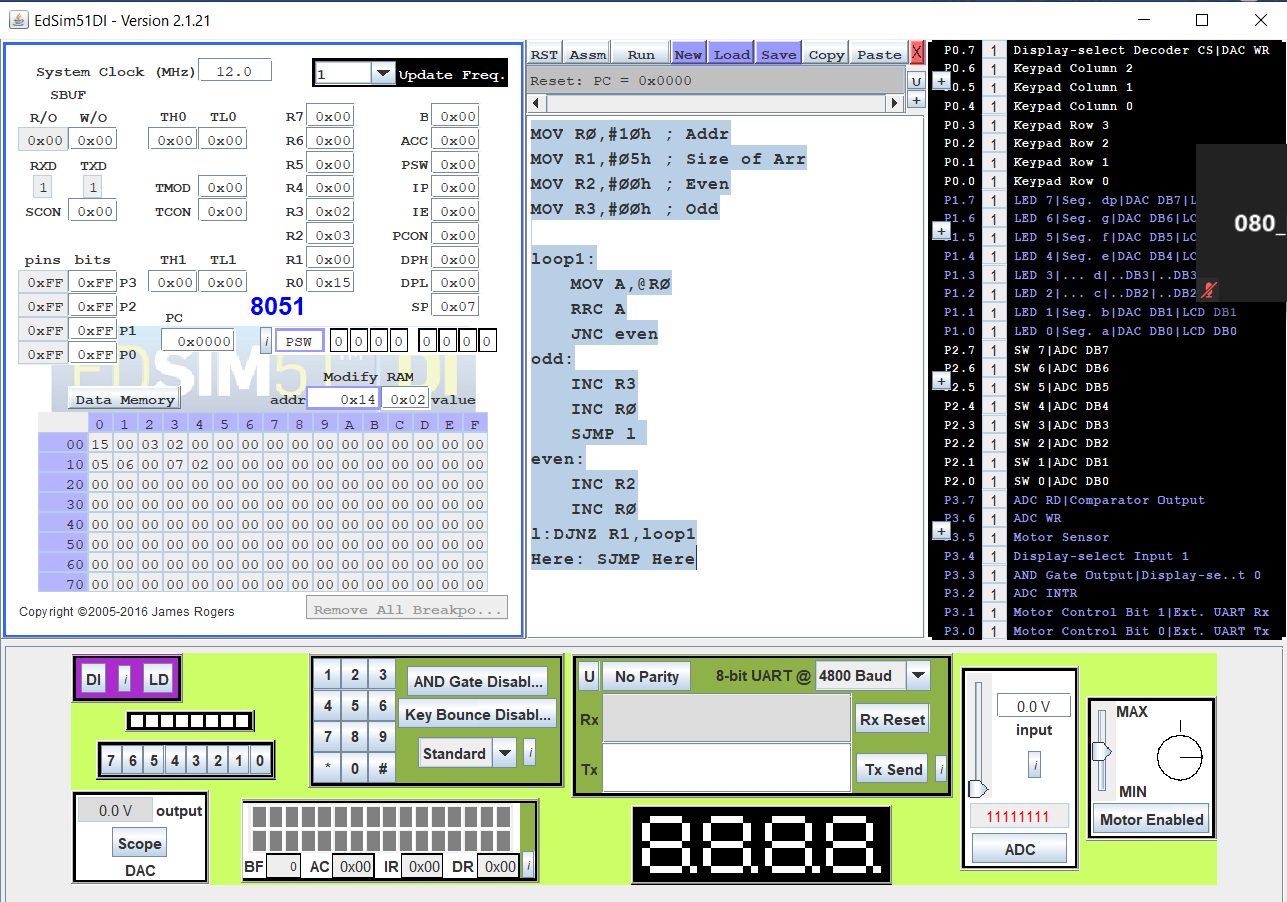
even:

INC R2

INC R0

l:DJNZ R1,loop1

Here: SJMP Here



**Result:**

An ALP using 8051 to find number of odd and even numbers was implemented.