MICROPROCESSORS MODEL LAB EXAM

Shivanirudh S G
CSE-C
185001146

1. Write an ALP using 8051 to find number of ones and zeros in a 16 bit number.

Aim: To find the number of ones and zeros in a 16 bit number using 8051.

Algorithm:

- Assign the upper byte of number to be counted to register 0, lower byte to register 1 and 08H(number of bits in each byte) to register 2.
- Move values 00H to registers 3 and 4 to act as zero and one counters respectively.
- Move contents of register 0 to A.
- Under label HER, left rotate A using RLC A.
- Jump to label NOZER if CF = 1.
- Increment register 3 and jump to label DON.
- Under label NOZER, increment register 4.
- Under label DON, decrement register 2 and loop back to HER if it is not 0.

- Move value 08H to register 2, and contents of register 1 to A.
- Under label HERE, left rotate A using RLC A.
- Jump to label NOZERO if CF = 1.
- Increment register 3 and jump to label DONE.
- Under label NOZERO, increment register 4.
- Under label DONE, decrement register 2 and loop back to HERE if it is not 0.

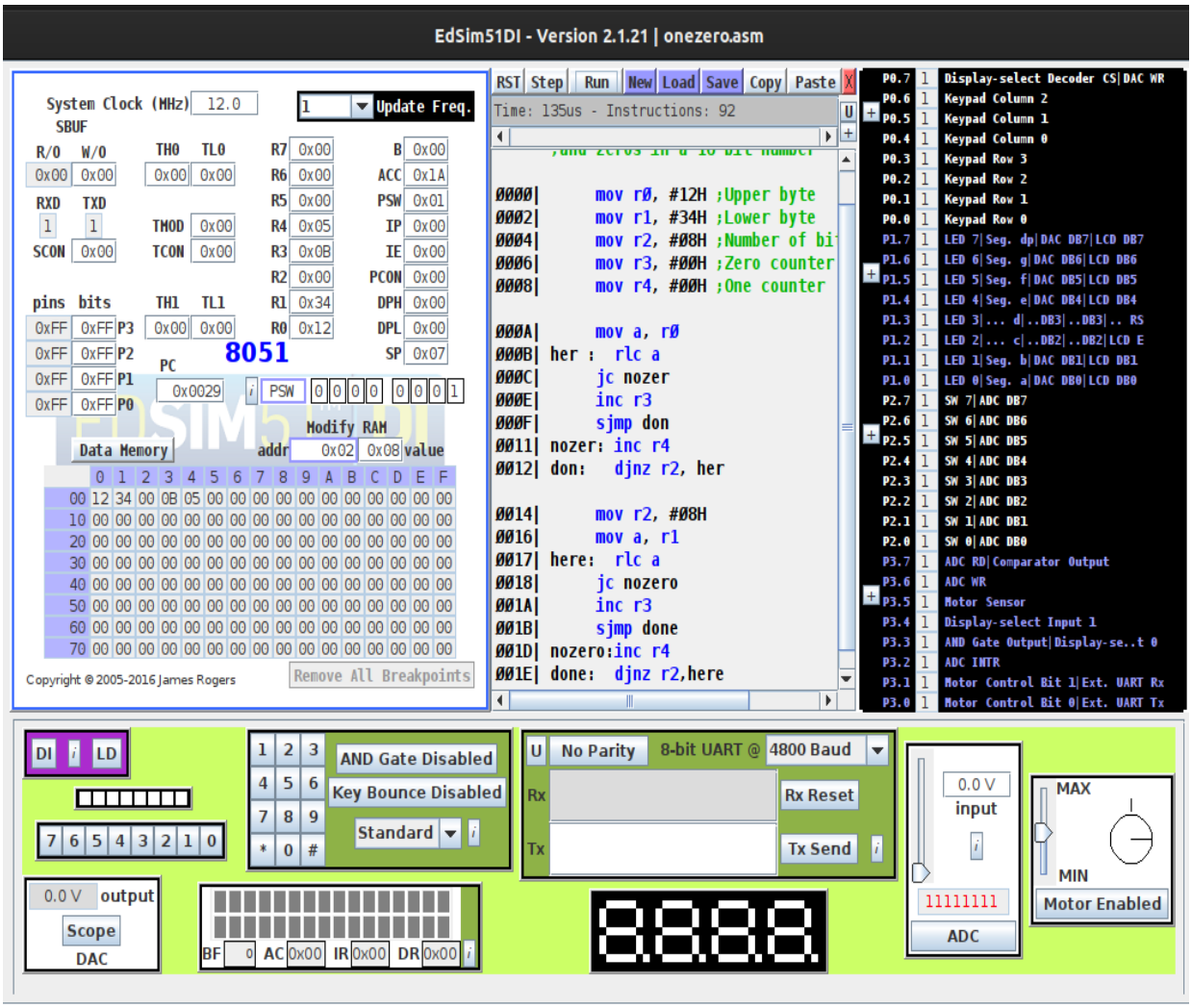
Program:

```
;Program to find number of ones
;and zeros in a 16 bit number

            mov r0, #12H ;Upper byte
            mov r1, #34H ;Lower byte
            mov r2, #08H ;Number of bits per byte
            mov r3, #00H ;Zero counter
            mov r4, #00H ;One counter

            mov a, r0       ;Load upper byte to A
her :       rlc a           ;Left rotate A
            jc nozer        ;Jump to label NOZER if CF = 1
            inc r3          ;Increment zero counter
            sjmp don
nozer:      inc r4          ;Increment one counter
don:        djnz r2, her    ;Repeat till r2 is 0
```

```
        mov r2, #08H ;Number of bits per byte
        mov a, r1    ;Load lower byte to A
here:   rlc a        ;Left rotate A
        jc nozero    ;Jump to label NOZERO if CF = 1
        inc r3       ;Increment zero counter
        sjmp done
nozero: inc r4       ;Increment one counter
done:   djnz r2,here ;Repeat till r2 is 0
```

Input/Output:

Input: 1234H
Output: Number of zeroes – 11 (0BH)
        Number of ones – 5 (05H)



2. Write an ALP using 8086 to count odd and even numbers in list.

Aim: To count the number of even and odd numbers in a list using 8086.

Algorithm:
- Move data to accumulator and then to data segment.
- Move offset of list to SI register, count to CL register.
- Move value 00H to registers DH and DL to act as even and odd counters respectively.
- Under label HERE, move value at address pointed by SI register to BL register.
- Mask everything but the last bit using AND BL, 01H.
- Jump to OCNT if BL is not equal to 0.
- Increment value of DL register, and jump to DONE.
- Under label OCNT, increment value of DH register.
- Under label DONE, increment SI register.
- Decrement CL register, jump to HERE if CL is not 0.
- Odd and even counts in DH, DL must be stored in odd and eve respectively.

Program:

```
;Program to count number of odd and even numbers in a list

assume ds:data, cs:code
data segment
org 00H
source db 01H, 12H, 23H, 33H, 44H, 56H, 67H, 77H, 80H, 91H    ;Input list
org 10H
eve db 00H                  ;Even counter
odd db 00H                  ;Odd counter
org 20H
count db 0AH                ;Number of elements
data ends
code segment
start: mov ax, data
mov ds, ax

mov si, offset source
mov cl, count
mov dh, 00H                 ;Even counter
mov dl, 00H                 ;Odd counter
here: mov bl, [si]
and bl, 01H                 ;Mask everything but lsb
jne ocnt
inc dh                      ;Increment even counter
jmp done
```

```
ocnt: inc dl                ;Increment odd counter
done: inc si
dec cl
jnz here                    ;Repeat till CL is 0
mov eve, dh
mov odd, dl
mov ah, 4ch
int 21h
code ends
end start
```

Input/Output:

Input:

01H, 12H, 23H, 33H, 44H, 56H, 67H, 77H, 80H, 91H

Output:

Number of even – 4 (04H)

Number of odd – 6 (06H)