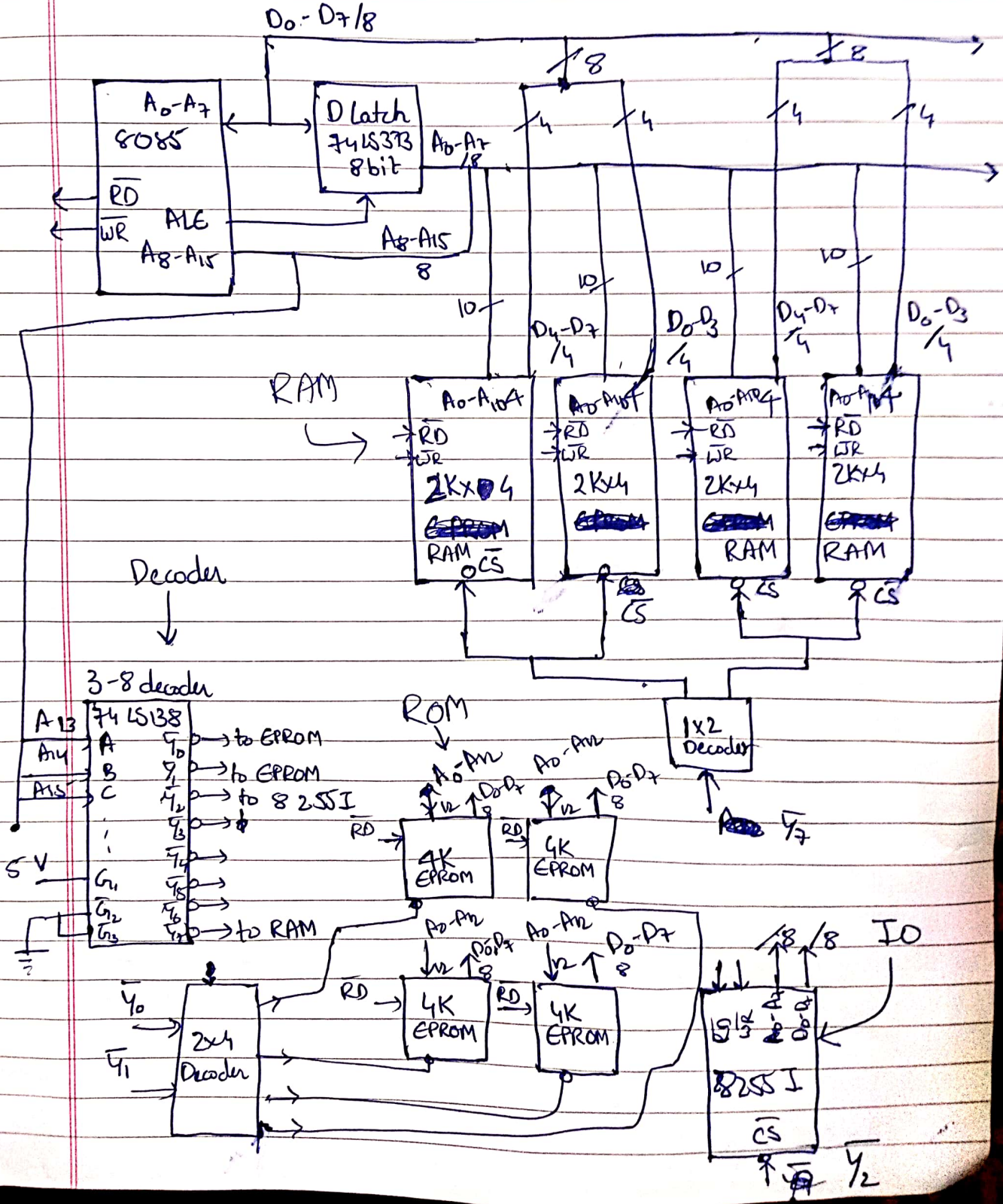
Part 2 : U18CO081 Krunal Rank 1/8

Ans 1/2



Part 2 U18CO081 2/8
Kunal Rank

Ans 2.

Part 2 [illegible] : 1/1
Round Rank : 3/8

Ans 3:

i) ADD BYTE PTR[DI] :-

The add instruction is used to add the byte value present at PTR address with an offset of DI.

ii) IMUL BYTE PTR[BX]

The signed multiplication done with AL with memory content at address PTR with offset BX.
The multiplication result is stored in AX.

Ans 4: DD is directive is used to define Double Dword allocating a memory of 4 bytes for each variable.

Hence DWORD-VAR has 5 Double Dwords: (5×4 = 20 bytes) allocated to it, followed by some more values later.

The values assigned are as follows:

00 01 02 00 00 01 02 00 00 01 02 00 00 01 02 00
00 01 02 00 XX XX XX XX FB FF FF FF Hex(B) Hex(A) 00 00
Hex(5) Hex(6) 00 00 56 02 00 00.

XX = Garbage values
Hex(x) = Hex value of X.

Part 2 U18CO081 4/8
Kunal Rank

Ans 5:

```
.model small
.stack 10H
.data
num1 dw 8001H
res dw ?        ; Answer stored here, res=0 Not palindrome  res=1 Palindrome
.code
start: mov ax,@data
mov ds, ax
mov cx, 10h
mov ax, num1
getint num1 ;Macro for num1 → call ispal.
ispal procedure near ~~; Store BX = Reverse of AX~~
rcr ax,1        ;Declaring procedure.
rcl bx,1
dec cx
jnz doit

ender:          ; Compare AX=BX
mov ax,num1
cmp ax,bx
jz equal
~~mov res,~~
mov res,00h
jmp exit
```

Part 2 U18CO081
Kunal Rank 5/8

```
equal:                  ; If equal, res=1.
 mov res,1h
ispal endp.
exit:
 mov ah,4ch
  int 21h
 end start
```

We can enclose the labels doit and ender inside a procedure.
Now, to use macro instead of inbuilt value,
use the following code:-

```
get int macro num
local loop1
local next
loop1: mov ah,01h
 int 21h
 cmp al,30h          ; Check AL < "Enter" 0
 jc next
 cmp al,3Ah          ; Check AL > 9
 jnc next
 mov bl,al
 sub bl,30h
  mov ax,num
  mov cx,0ah         ; num= 10*num +al
  mul cx
  add ax,bx
  mov num,ax
next: cmp al,10h     ; Check AL = "Enter"
 jnz loop1
 endm.
```

Part 2 U18CO081 6/8
Krunali Rank

Ans 6:

```
.model small

;macro for printing string.        ; Macro for
print macro m                      ; Printing string
mov ah, 09h
mov dx, offset m
int 21h
endm


.data                              ; Data values.
empty db 10,13," $"
str1 db 25,?,25 dup("$")
str2 db 25,?,25 dup("$")
len db ?
mstring db 10,13,"Enter String:-$"
notpalin db "String is not a Palindrome!$"
ispalin db "String is a Palindrome!$"


.code
mov ax,@data
mov ds,ax

again: print mstring
call accept_string    ; get input
mov si,offset str1
mov cl,str1+1
mov ch,00h
mov len,cl
```

Part 2 U18CO081 Kunal Rank 7/8

```
        inc si
        add si, cx ; si points to end

        mov di, offset str1     ; di → start of string
        add di, 02h             ; add 2h to reach actual start.
        mov cl, len

cmpagain: mov al, [si]          ; start comparing first and last
        mov ah, [di]
        inc di
        dec si
        jne nopalin             ; jump to exit message "No Palindrome".
        dec cl
        jnz cmpagain

        print ispalin
        jmp exit.

no palin: print notpalin

exit:
        mov ah, 4ch
        int 21h

accept proc near
        mov ah, 01
        int 21h
        ret
display1 proc near              ; Display number procedure
        mov al, bl
        mov bl, al
```

```
        and al,0f0h
        mov cl,04h
        rol al,cl

        cmp al,09
        jbe number
        add al,07

number: add al,30h
        mov dl,al
        mov ah,02
        int 21h
        mov al,~~00f0h~~ b8h
        mov and al,00fh
        cmp al,09
        jbe number2
        add al,07
number2: add al,30h
        mov dl,al
        mov ah,02
        int 21h
        ret
display endp

accept_string proc near  ;Accept String Procedure;
mov ah,0ah
mov dx,offset str1
int 21h
ret
accept_string end
end start
end
```