

Lab 17 – Using Logic & Shift Instructions

Objectives

The objective of this lab is to implement different programs related to logic and shift Instructions.

Lab Tasks

Task 1: Write a program to reverse bit pattern of the contents of AL register.

```
[org 0x0100]
mov al,00001100b
mov bl,0b
mov cx,0
start:
rcr al,1
rcl bl,1
add cx,1
cmp cx,8
jnz start
mov al,bl
mov ax, 0x4c00
int 21h
```

Task 2: Use ROL to count the number of 1 bits in BX, without changing BX. Put answer in AX.

```
[org 0x0100]
mov cx,0
mov bx,1010100110000101b
start:
rol bx,1
jnc nc
add cx,1
nc: add dx,1
```

```
cmp dx,16
```

```
jnz start
```

```
mov ax,cx
```

```
mov ax, 0x4c00
```

```
int 21h
```

Task 3: Use ROL to count the number of 0 bits in BX, without changing BX. Put answer in DX.

```
[org 0x0100]
```

```
mov cx,0
```

```
mov bx,1010100110000101b
```

```
start:
```

```
rol bx,1
```

```
jc yc
```

```
add cx,1
```

```
yc: add dx,1
```

```
cmp dx,16
```

```
jnz start
```

```
mov dx,cx
```

```
mov ax, 0x4c00
```

```
int 21h
```

Task 4: Write a single instruction that clears the high 8 bits of AX and does not change the low 8 bits.

And ah,0

Task 5: Write a single instruction that sets the high 8 bits of AX and does not change the low 8 bits.

Or ah,1

