

**LAPORAN PRAKTIKUM**  
**DASAR SISTEM KOMPUTER**



**DISUSUN OLEH:**  
**EKO RACHMAT SATRIYO (2100018142)**  
**JUM'AT 07.30-KELAS C**

**PROGRAM STUDI TEKNIK INFORMATIKA**  
**FAKULTAS TEKNOLOGI INDUSTRI**  
**UNIVERSITAS AHMAD DAHLAN**  
**DESEMBER 2021**

## POSTEST I

The screenshot displays the 8086 Assembler interface with the file `C:\Users\Windows7 64bit\Desktop\Apk DSK\smsv32\lift.asm` open. The main window shows the assembly code, and a "Lift on Port 06" window is overlaid on the left, illustrating a lift simulation.

**Lift Simulation Window:**

- Stopped:** The lift is currently stopped.
- Motor:** A circular motor icon is shown at the top.
- up/down:** Arrows indicate the direction of movement.
- MSB 00001000 LSB:** The current position is shown as 00001000.
- Repair Crashed Lift:** A button to reset the simulation.

**Assembly Code View:**

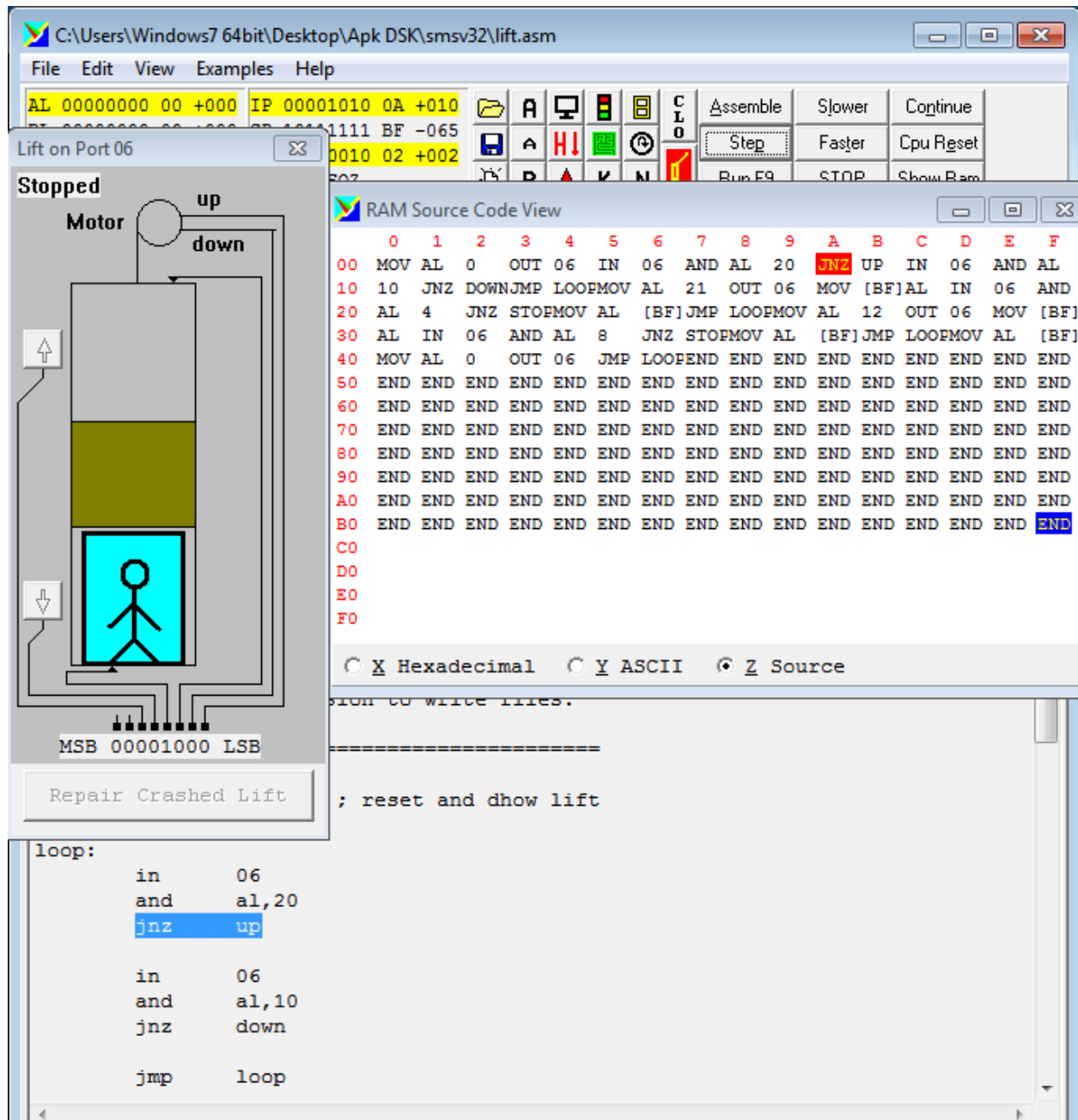
0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
MOV AL 0	OUT 06	IN 06	AND AL 20	JNZ UP	IN 06	AND AL									
10	JNZ DOWN	JMP LOOP	MOV AL 21	OUT 06	MOV [BF] AL	IN 06	AND								
AL 4	JNZ STOP	MOV AL [BF]	JMP LOOP	MOV AL 12	OUT 06	MOV [BF]									
AL IN 06	AND AL 8	JNZ STOP	MOV AL [BF]	JMP LOOP	MOV AL [BF]										
MOV AL 0	OUT 06	JMP LOOP	END	END	END	END	END	END	END	END	END	END	END	END	END
END	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END
END	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END
END	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END
END	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END
END	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END
END	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END
END	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END

**Assembly Code:**

```

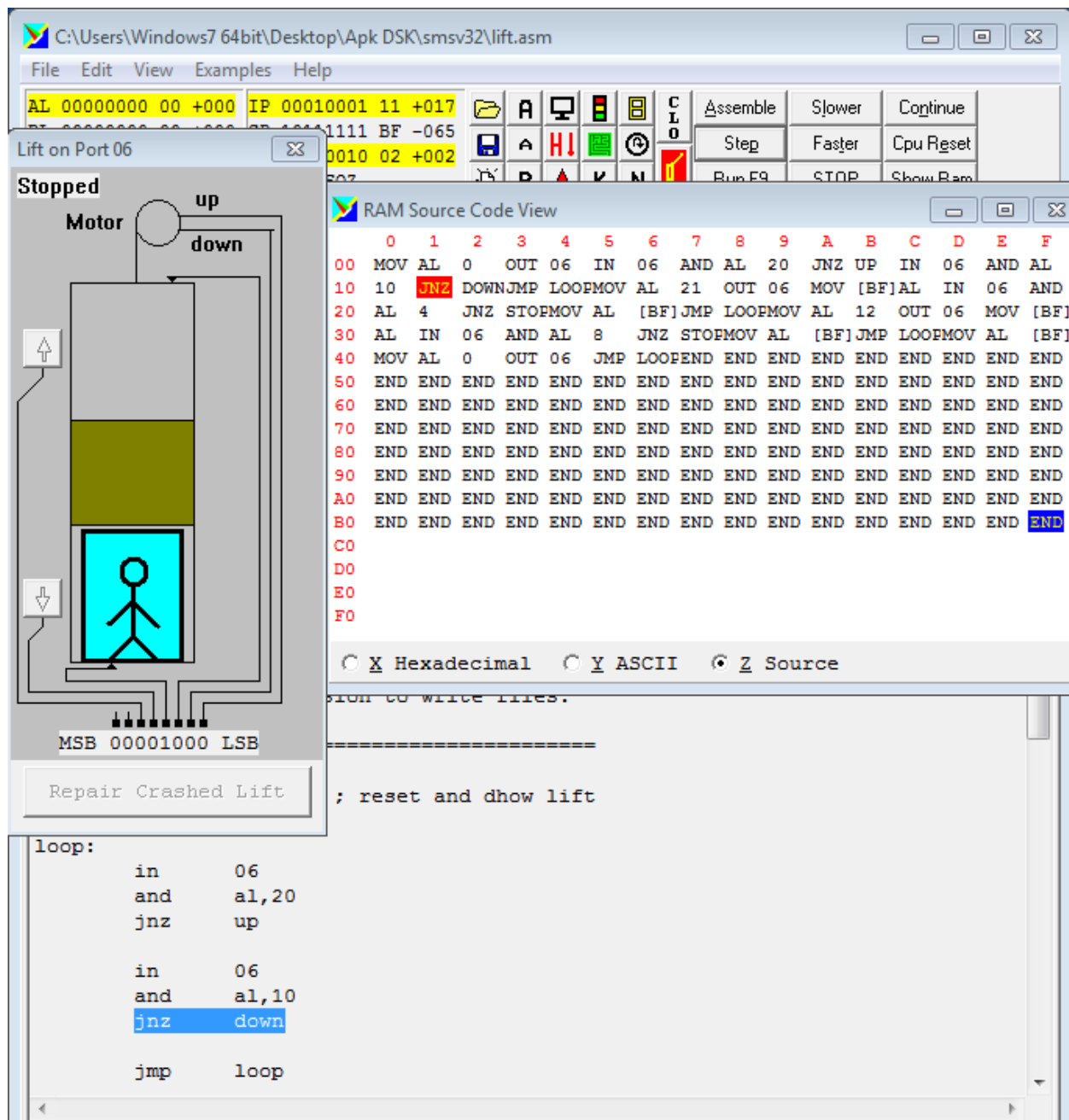
mov     al,0      ; reset and dhow lift
out     06
loop:
in      06
and     al,20
  
```

Kode in 06=memunculkan program pada layar



Pada bagian jnz up

Apabila user tidak mengklik panah atas(di lift) maka proses akan dilanjutkan ke in 06 dan seterusnya.



Begitu pula jnz down

Apabila user tidak mengklik panah bawah(di lift) maka proses akan dilanjutkan ke loop dan akan terus mengulang apabila tidak ada inputan(panah bawah/atas)

## POSTEST II

The screenshot shows an 8086 assembly simulator with the following components:

- File Edit View Examples Help**: Standard menu bar.
- Registers Window**: Shows AL=00000000, IP=00001010, and other registers.
- RAM Source Code View**: Displays assembly code. The instruction `JNZ UP` at address 0A is highlighted in red.
- Lift Diagram**: A schematic of a lift with a person inside. It has 'up' and 'down' buttons and a 'Motor' label. The status is 'Stopped'.
- Assembly Code**:
 

```

loop:
    in     06
    and    al,20
    jnz    up

    in     06
    and    al,10
    jnz    down

    jmp    loop
      
```

Karena saya mengklik tanda panah atas maka akan lompat ke perintah up:

karena

The screenshot shows an 8086 assembly simulator interface. The main window displays a lift diagram with a person inside, labeled "Going Up" and "Motor". A "Repair Crashed Lift" button is at the bottom. The assembly code window shows a program starting with "up:" and instructions like "mov al, 21", "out 06", "mov [BF], al", "in 06", "and al, 4", "jnz stop", "mov al, [BF]", and "jmp loop". The RAM Source Code View window shows a memory dump with hexadecimal and ASCII values.

File Edit View Examples Help

AL 00100001 21 +033 IP 00011010 1A +026

Lift on Port 06

Going Up

Motor

up

down

MSB 00100001 LSB

Repair Crashed Lift

RAM Source Code View

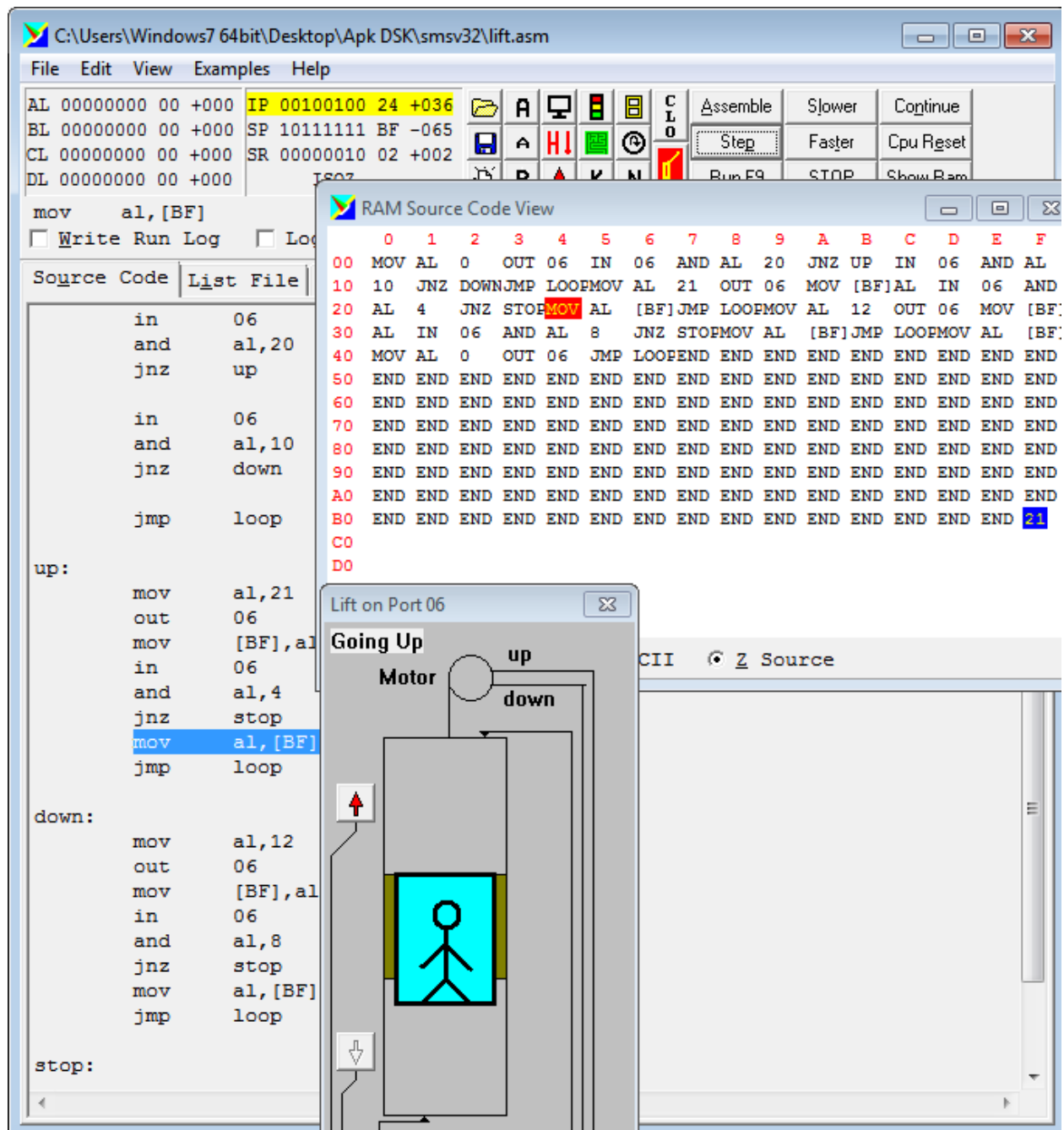
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	MOV	AL	0	OUT	06	IN	06	AND	AL	20	JNZ	UP	IN	06	AND	AL
10	10	JNZ	DOWN	JMP	LOOP	MOV	AL	21	OUT	06	MOV	[BF]	AL	IN	06	AND
20	AL	4	JNZ	STOP	MOV	AL	[BF]	JMP	LOOP	MOV	AL	12	OUT	06	MOV	[BF]
30	AL	IN	06	AND	AL	8	JNZ	STOP	MOV	AL	[BF]	JMP	LOOP	MOV	AL	[BF]
40	MOV	AL	0	OUT	06	JMP	LOOP	END	END	END	END	END	END	END	END	END
50	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END
60	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END
70	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END
80	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END
90	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END
A0	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END
B0	END	END	END	END	END	END	END	END	END	END	END	END	END	END	END	21
C0																
D0																
E0																
F0																

X Hexadecimal Y ASCII Z Source

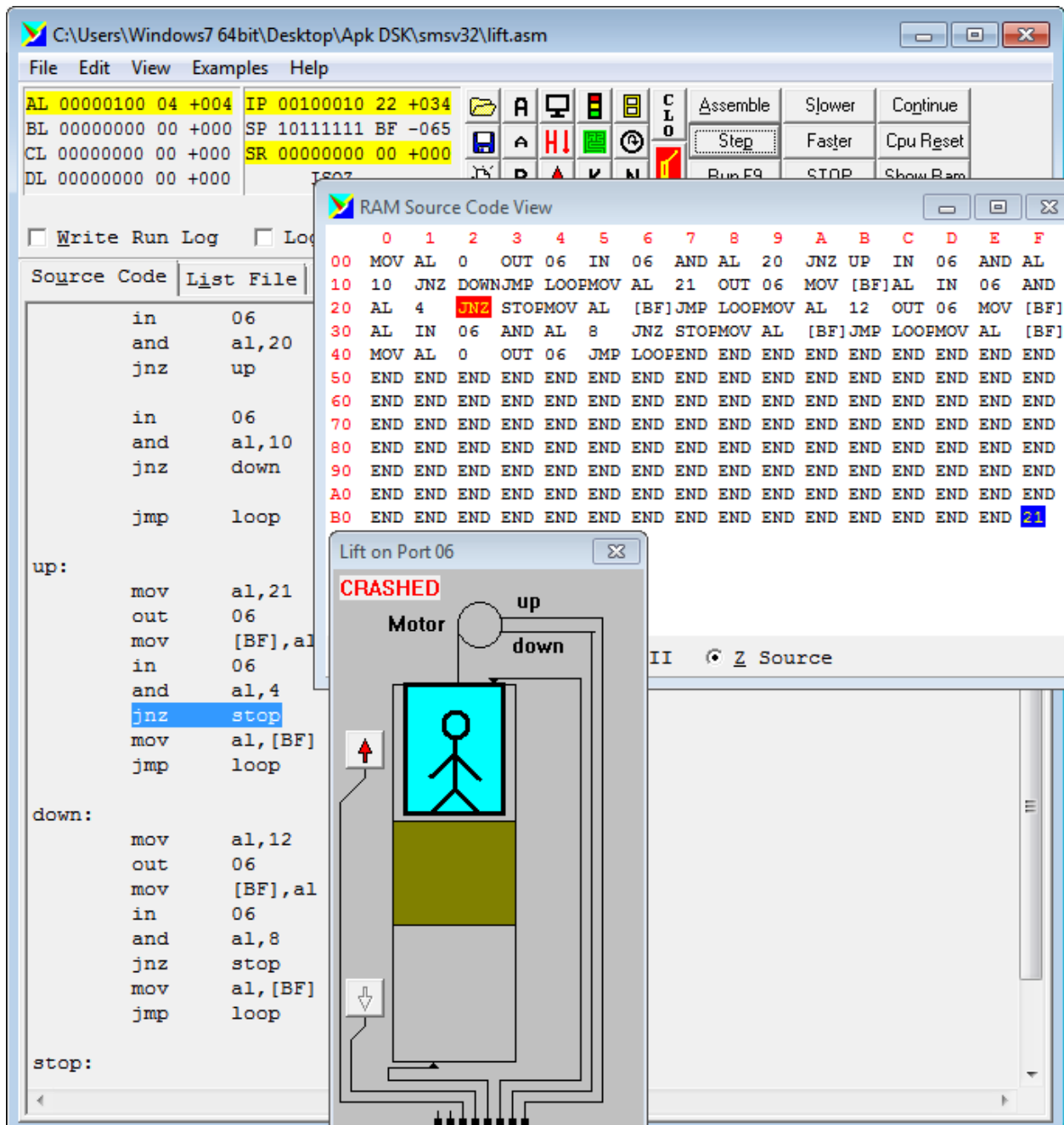
up:

```
mov    al,21
out    06
mov    [BF],al
in     06
and    al,4
jnz    stop
mov    al,[BF]
jmp    loop
```

mov [BF],al membuat lift naik ke atas

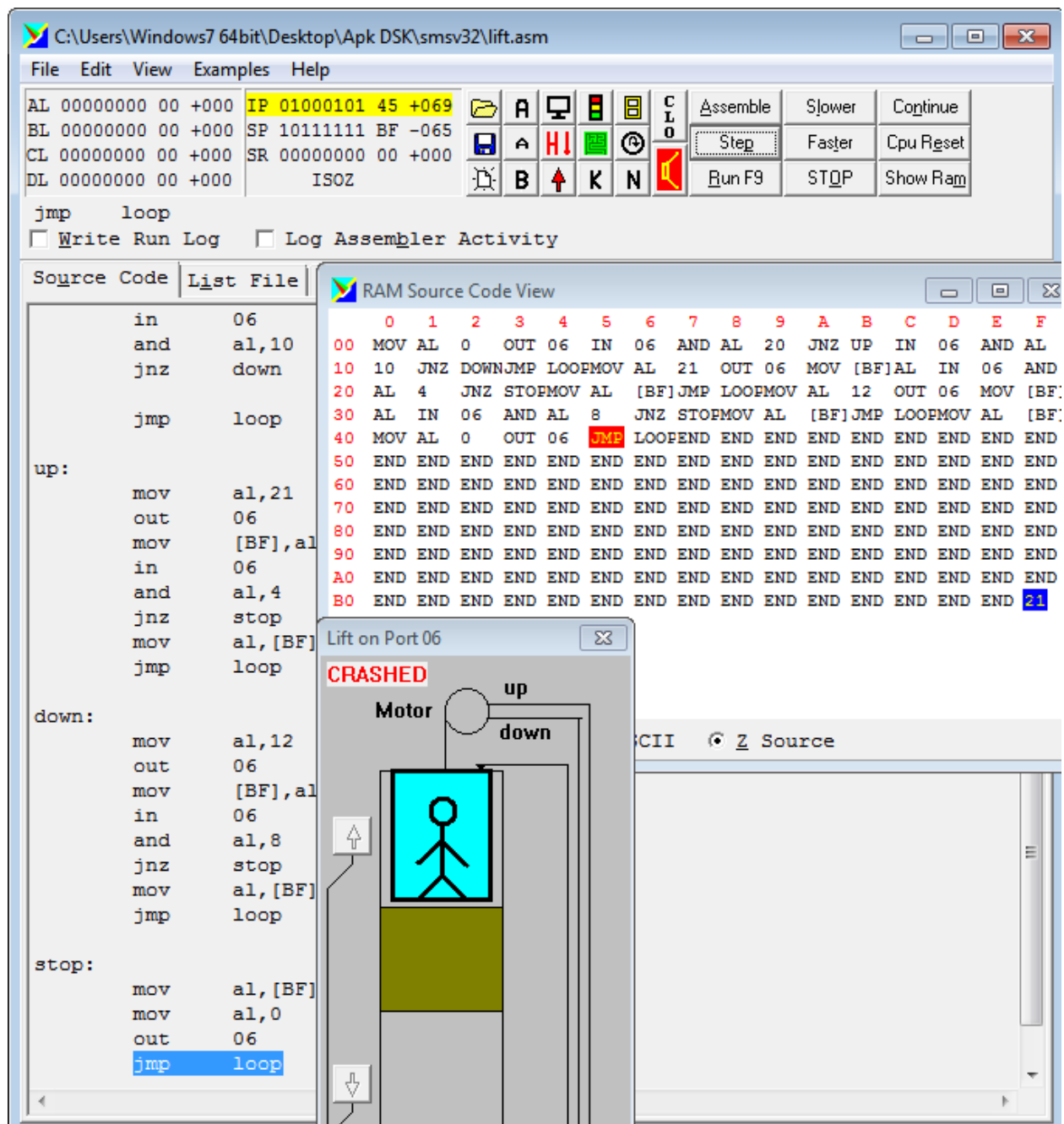


Jnz stop tidak akan berfungsi apabila lift belum crashed, sehingga akan dilanjutkan ke mov al,[bf]

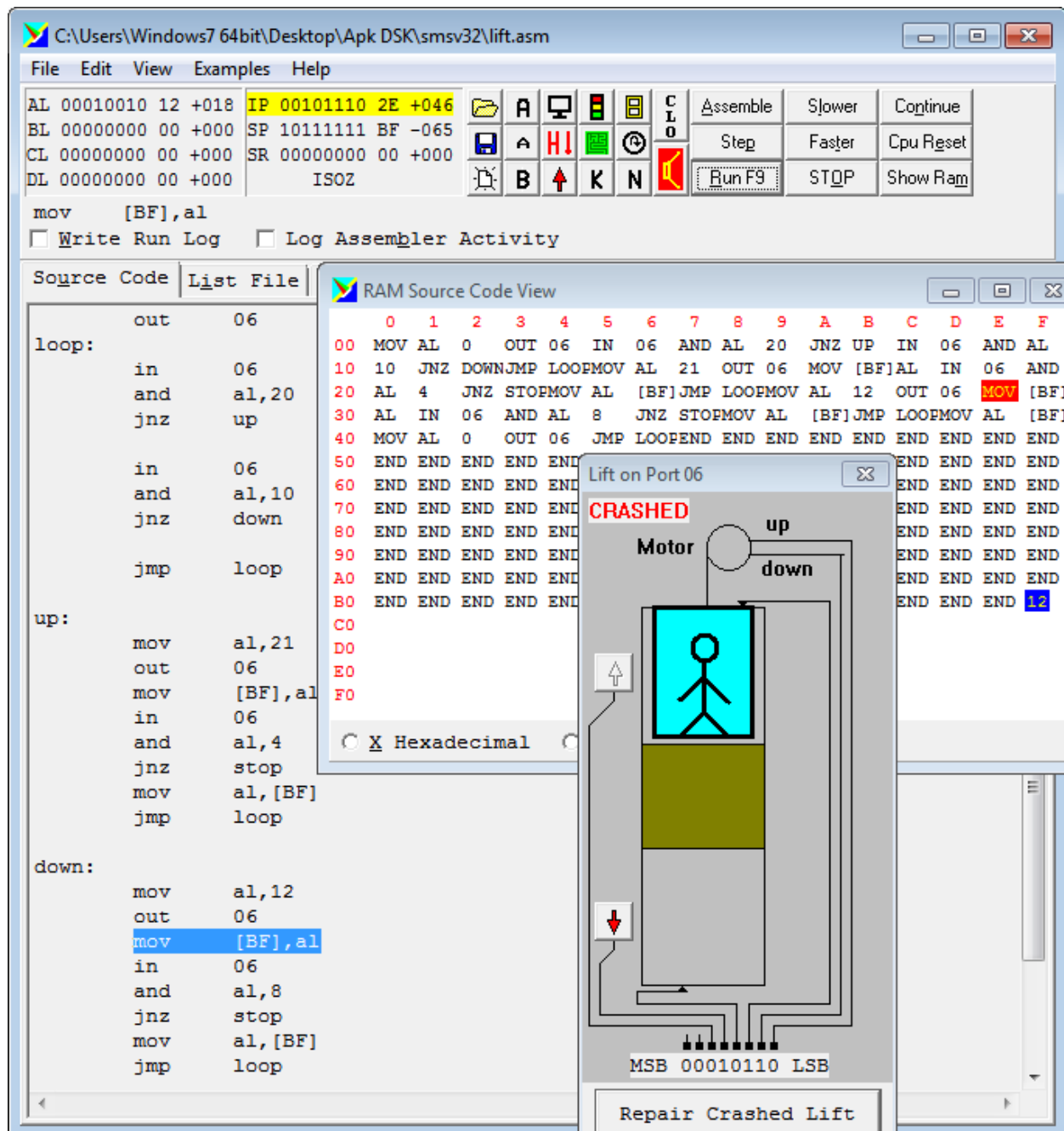


Jnz stop akan lompat ke perintah stop

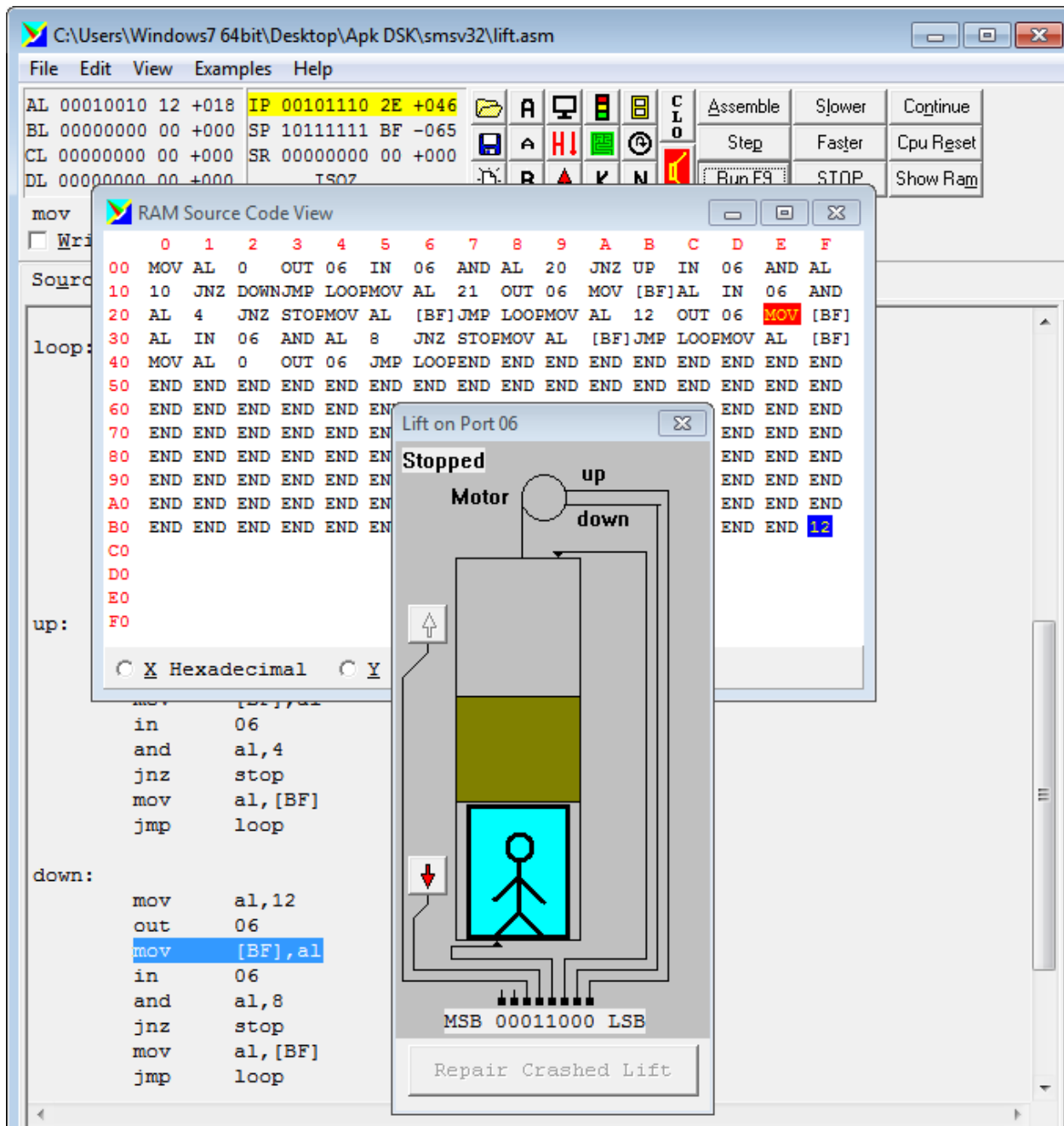




Jmp loop akan lompat ke loop,dan akan terus mengulang di loop menunggu input(atas/bawah)

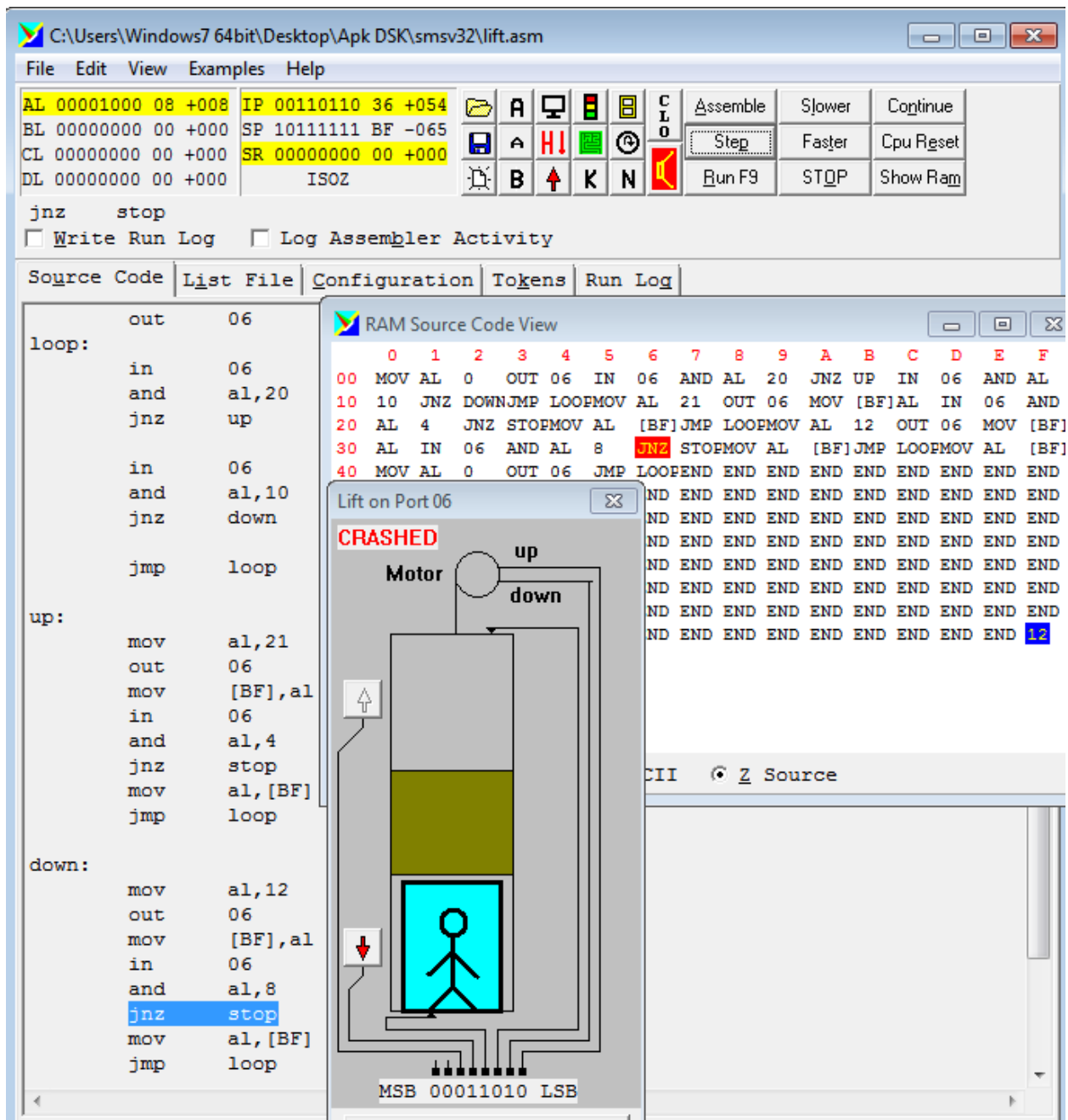


Walaupun user sudah memilih bawah,apabila lift belum diperbaiki,maka perintah mov [bf],al tidak akan bekerja

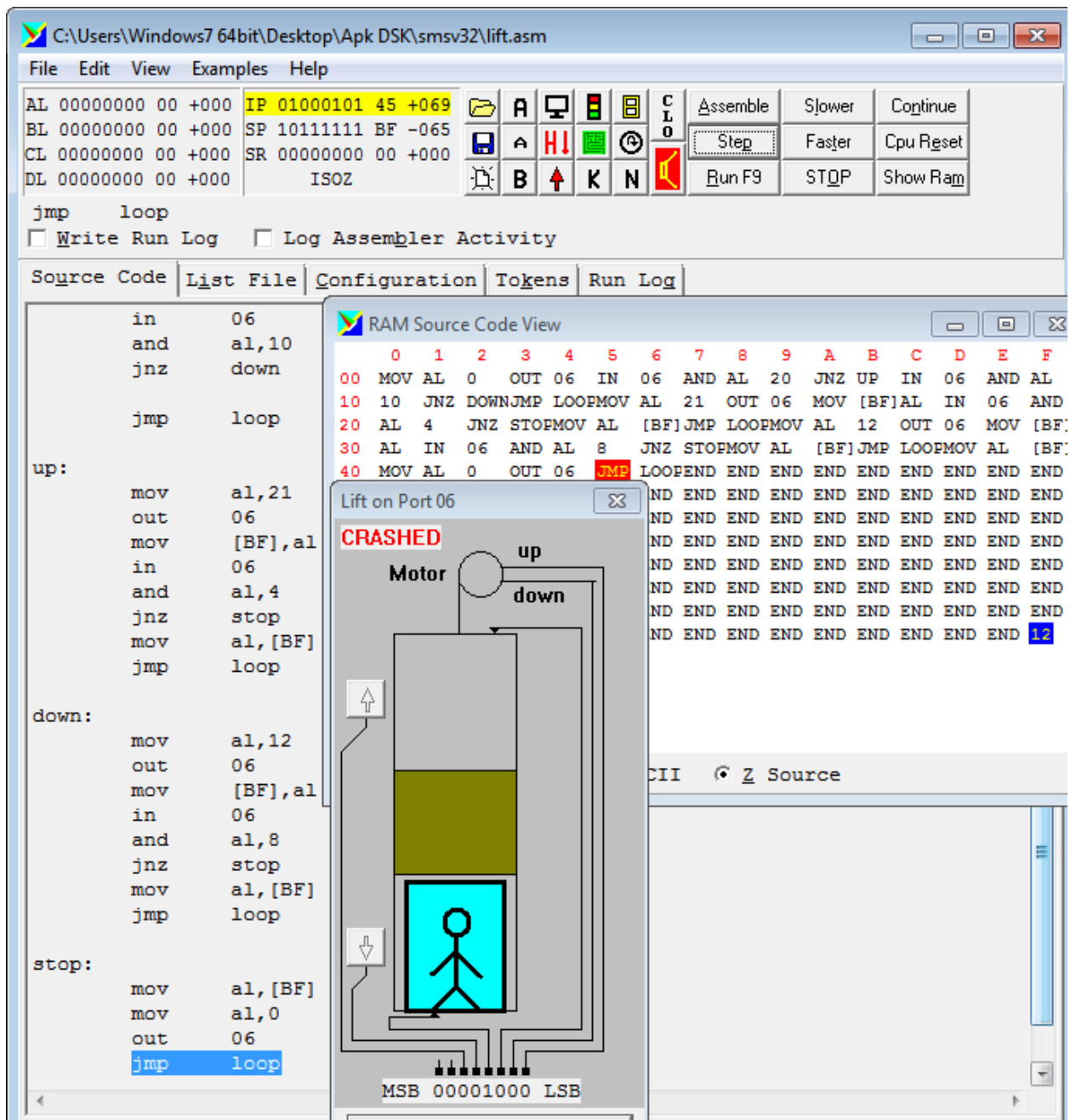


Mov [bf],al akan membuat lift bergerak turun hingga crashed

Untuk seterusnya mekanismenya sama dengan up



Jnz stop akan lompat ke perintah stop



Jmp loop akan lompat ke loop,dan akan terus mengulang di loop menunggu input(atas/bawah)