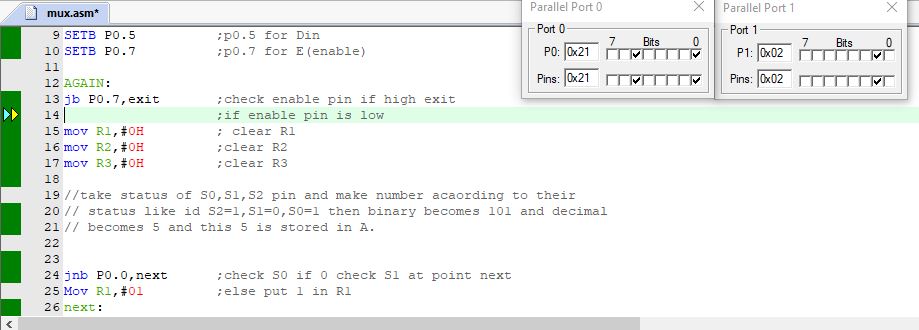
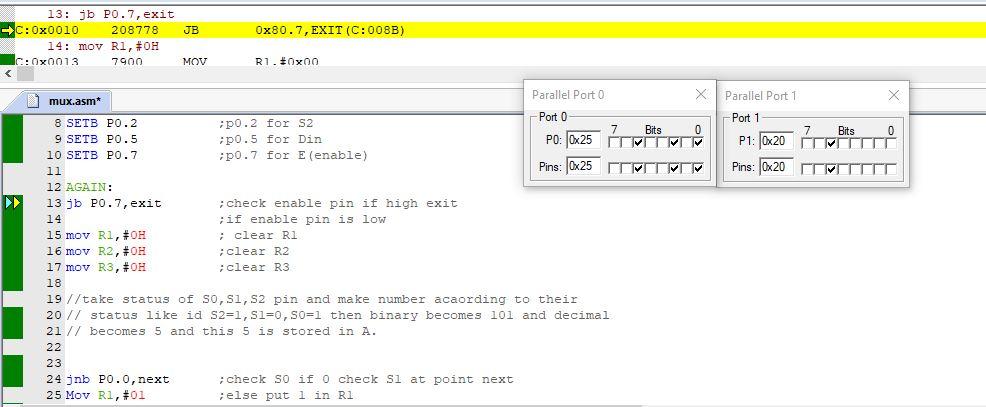
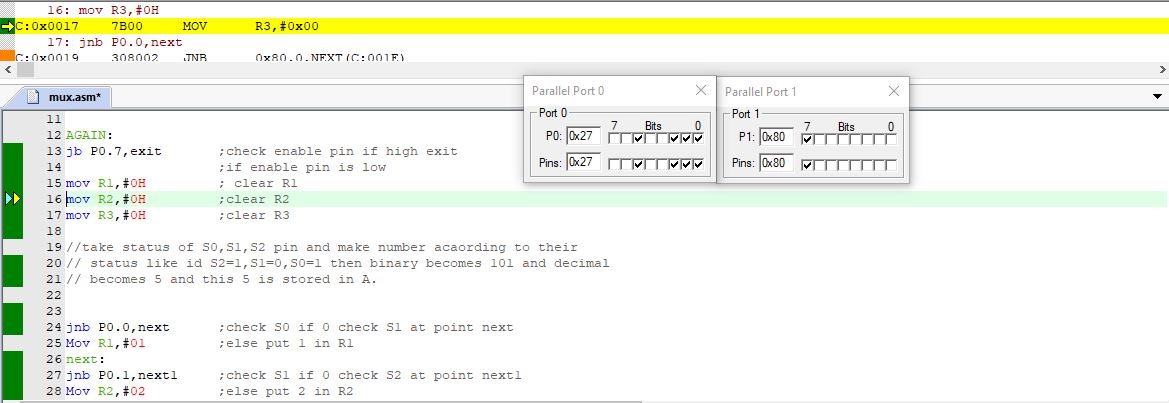
**Question No 1:**

**Code output Snip:**

**When S2=0, nS1=0, S0=1 and Din=1:**

**When S2=1, S1=0, S0=1 and Din=1:**

**When S2=1, S1=1, S0=1 and Din=1:**

****

**Code:**

ORG 00h ; starting of code

MOV A,#00H ; P1 as output port

MoV P1,A ;mov 00 to A

MoV P3,A

// declare pins as input

SETB P0.0 ;p0.0 for S0

SETB P0.1 ;p0.1 for S1

SETB P0.2 ;p0.2 for S2

SETB P0.5 ;p0.5 for Din

SETB P0.7 ;p0.7 for E(enable)

AGAIN:

jb P0.7,exit ;check enable pin if high exit

;if enable pin is low

mov R1,#0H ; clear R1

mov R2,#0H ;clear R2

mov R3,#0H ;clear R3

//take status of S0,S1,S2 pin and make number acaording to their

// status like id S2=1,S1=0,S0=1 then binary becomes 101 and decimal

// becomes 5 and this 5 is stored in A.

jnb P0.0,next ;check S0 if 0 check S1 at point next

Mov R1,#01 ;else put 1 in R1

next:

jnb P0.1,next1 ;check S1 if 0 check S2 at point next1

Mov R2,#02 ;else put 2 in R2

next1:

jnb P0.2,addition ;check S2 if 0 do addition

Mov R3,#04 ;else put 4 in R3

addition:

Mov A,R1 ;mov R1 in A

add A,R2 ;add R2 to A

add A, R3 ;add R3 to A

// the number stored in A is compared with the all numbers

// from 1 to 8 and from which it comapres we take data from

//Din pin and send it to respective pin of P1 and then exit

//from that condition to loop all process again

cjne A,#00,K1 // compare A with 00 if not equal goto K1

CLR A

mov P1,A // else clear port P1

Mov C,P0.5 // take data from Din

MOV P1.0,C //send it to Y1

Sjmp exit

K1:

cjne A,#01,K2 // compare A with 00 if not equal goto K2

CLR A

mov P1,A // else clear port P1

Mov C,P0.5 // take data from Din

MOV P1.1,C //send it to Y2

Sjmp exit

K2:

cjne A,#02,K3

CLR A

mov P1,A

Mov C,P0.5

MOV P1.2,C

Sjmp exit

K3:

cjne A,#03,K4

CLR A

mov P1,A

Mov C,P0.5

MOV P1.3,C

Sjmp exit

K4:

cjne A,#04,K5

CLR A

mov P1,A

Mov C,P0.5

MOV P1.4,C

Sjmp exit

K5:

cjne A,#05,K6

CLR A

mov P1,A

Mov C,P0.5

MOV P1.5,C

Sjmp exit

K6:

cjne A,#06,K7

CLR A

mov P1,A

Mov C,P0.5

MOV P1.6,C

Sjmp exit

K7:

cjne A,#07,exit

CLR A

mov P1,A

Mov C,P0.5

MOV P1.7,C

exit:

jmp again

End