Program:

ORG 0000h

//Unconditional- Long Jump

MOV A, #05h

LJMP NEXT

MOV A, #00h

NEXT: MOV B, A

//Unconditional- Short Jump

MOV A, #0Ah

SJMP SKIP

MOV A, #00h

SKIP: MOV B, A

//Unconditional- Absolute Jump

MOV A, #0Fh

AJMP TARGET

MOV A, #00h

TARGET: MOV B, A

//Conditional- Jump if Zero

MOV A, #00h

JZ ZERO\_LABEL

MOV A, #0FFh

ZERO\_LABEL: MOV B, A

//Conditional- Jump if Not Zero

MOV A, #0Ah

SJMP NOT\_ZERO

MOV A, #00h

NOT\_ZERO: MOV B, A

//Conditional- Jump if Carry Set

MOV A, #0FFh

ADD A, #01h

JC CARRY\_LABEL

MOV B, A

MOV A, #00h

CARRY\_LABEL: MOV B, #00h

//Conditional- Jump if No Carry

MOV A, #0Ah

ADD A, #05h

JNC NO\_CARRY

MOV B, A

MOV A, #00h

NO\_CARRY: MOV B, #00h

//Conditional- Jump if Bit is Set

MOV A, #0FFh

JB ACC.7, BIT\_SET

MOV B, A

MOV A, #00h

BIT\_SET: MOV A, #0Fh

//Conditional- Jump if Bit is Not Set

MOV A, #0Fh

JNB ACC.7, BIT\_NOT\_SET

MOV B, A

MOV A, #00h

BIT\_NOT\_SET: MOV A, #0F0h

//Conditional- Jump if Bit is Set and Clear the Bit

SETB P1.0

JBC P1.0, BIT\_CLEAR

MOV A, #0FFh

BIT\_CLEAR: MOV A, #0Ch

END

Conclusion: Various Conditional and unconditional jumps were implemented successfully.