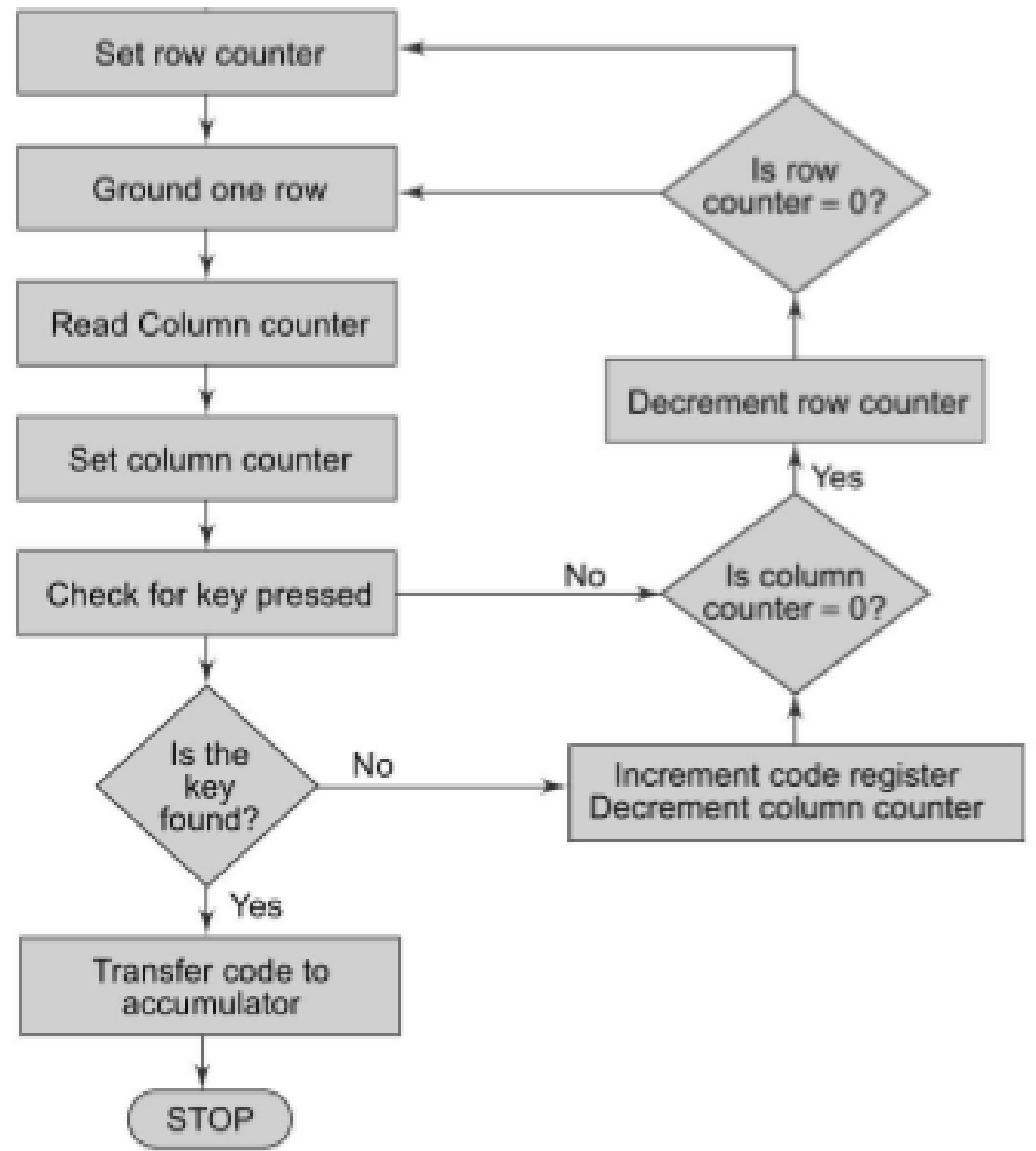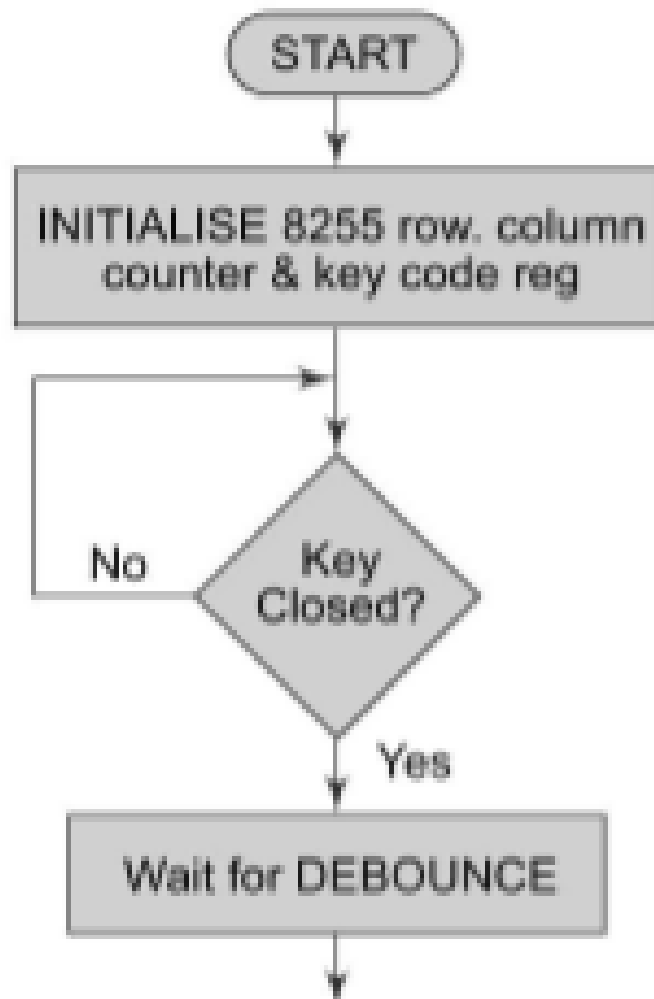## Problem 5.11
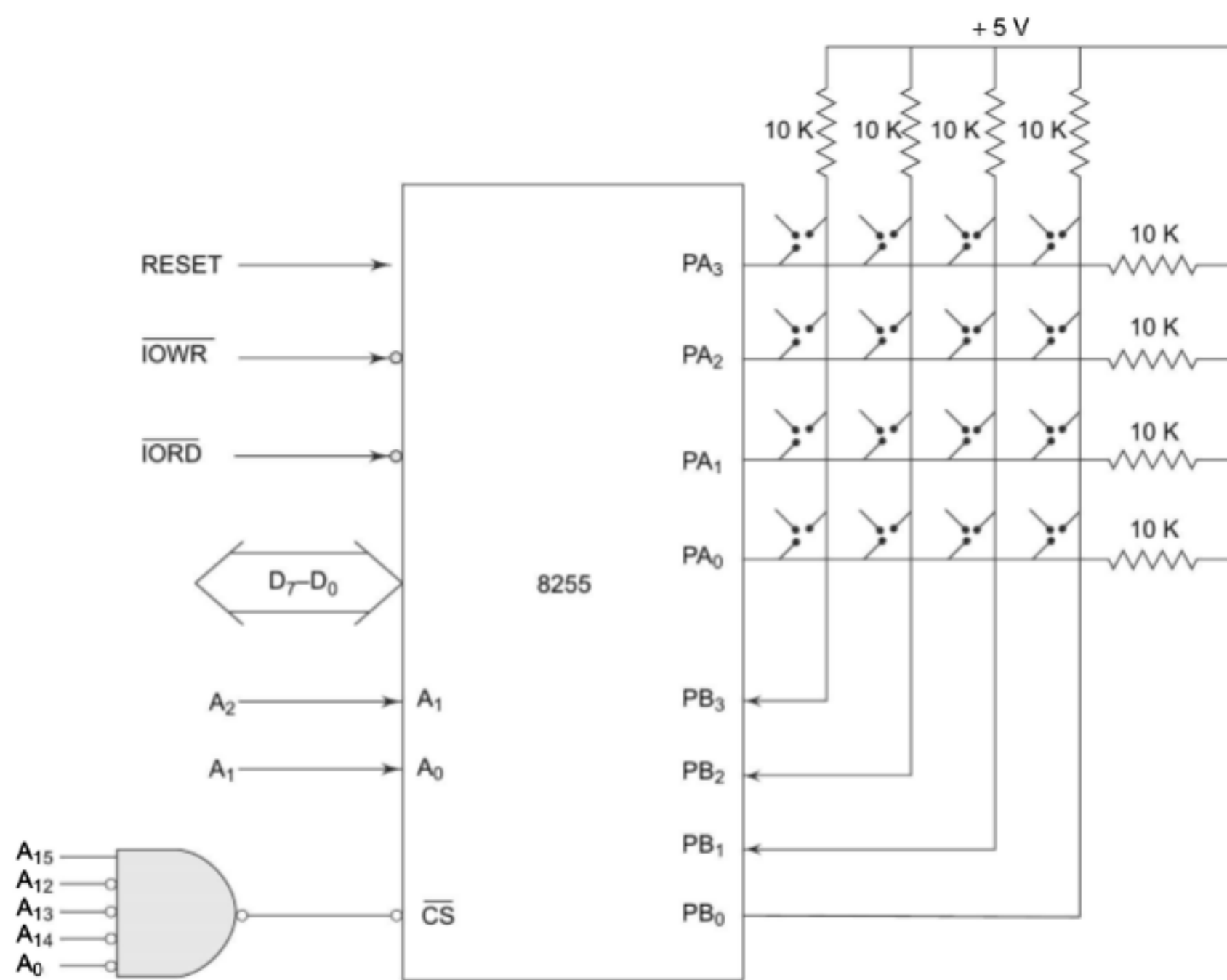
Interface a 4*4 Keyboard with 8086 using 8255. and write an ALP for detecting a key closure and return the key code in AL. The debouncing period for a key is 10 ms. Use software key debouncing technique. DEBOUNCE is an available 10 ms delay routine.

Solution   Port A is used as output port for selecting a row of keys while port B is used as an input port for sensing a closed key. Thus the keyboard lines are selected one by one through port A and the port B lines are polled continuously till a key closure is sensed. Then routine DEBOUNCE is called for key debouncing. The key code is decided depending upon the selected row and a low sensed column. The hardware circuit diagram is shown in Fig. 5.21.

The higher order lines of port A and port B are left unused. The addresses of port A and port B will be respectively 8000 H and 8002 H while the address of CWR will be 8006 H. The flow chart of the complete program is given in Fig. 5.22.

The ALP for the problem is given along with comments. The control word for this problem will be 82 H. Let us write this program using assembler directives. In this problem no major data is required hence only

START

INITIALISE 8255 row. column counter & key code reg

Key Closed? — No (loops back) / Yes

Wait for DEBOUNCE

Set row counter

Ground one row

Read Column counter

Set column counter

Check for key pressed — No → Is column counter = 0?

Is the key found? — No → Increment code register / Decrement column counter

Is column counter = 0? — Yes → Decrement row counter

Is row counter = 0?

Decrement row counter

Is the key found? — Yes

Transfer code to accumulator

STOP

Fig. 5.21 *Interfacing 4 × 4 Keyboard for Problem 5.11*

```
CODE            SEGMENT
ASSUME          CS : CODE
START:          MOV AL, 82H          ; Load CWR with

                MOV DX, 8006H        ; control word
                OUT DX, AL           ; required
                MOV BL, 00H          ; Initialize BL for key code
                XOR AX, AX           ; Clear all flags
                MOV DX, 8000H        ; Port Address in AX.
                OUT DX, AL           ; Ground all rows.
                ADD DX,02            ; Port B address in DX.
WAIT :          IN AL, DX            ; Read all columns.
                AND AL, OF H         ; Mask data lines $D_7$-$D_4$.
                CMP AL, OF H         ; Any key closed?
                JZ WAIT              ; If not, wait till key
                CALL DEBOUNCE        ; closure else wait for 10 ms
                MOV AL, 7FH          ; Load data byte to ground
                MOV BH, 04H          ; a row and set row counter.
```

```
NXTROW :    ROL AL, 01      ; Rotate AL to ground next row.
            MOV CH, AL      ; Save data byte to ground next row.
            SUB DX,02       ; Output port address is in DX.
            OUT DX, AL      ; Ground one of the rows.
            ADD DX,02       ; Input port address is in DX.
            IN AL, DX       ; Read input port for key closure.
            AND AL, OFH     ; Mask $D_4$-$D_7$.
            MOV CL, 04H     ; Set column counter.
NXTCOL :    ROR  AL, 01     ; Move $D_0$ in CF.
            JNC CODEKY      ; Key closure is found, if CF=0.
            INC BL          ; Increment BL for next binary
                            ; key code.
            DEC CL          ; Decrement column counter,
                            ; if no key closure found.
            JNZ NXTCOL      ; Check for key closure in next column
            MOV AL,CH       ; Load data byte to ground next row.
            DEC BH          ; if no key closer found in column
                            ; get ready to ground next row.
            JNZ NXTROW      ; Go back to ground next row.
            JMP WAIT        ; Jump back to check for key.
                            ; closure again.
CODEKY :    MOV AL, BL      ; Key code is transferred to AL.
            MOV AH, 4CH     ; Return to DOS prompt.
            INT 21 H
```

This procedure generates 10 ms delay at 5 MHz operating frequency.

```
DEBOUNCE  PROC NEAR
          MOV CL, 0E2H
BACK:     NOP
          DEC CL
          JNZ BACK
          RET
DEBOUNCE  ENDP
CODE      ENDS
          END START
```