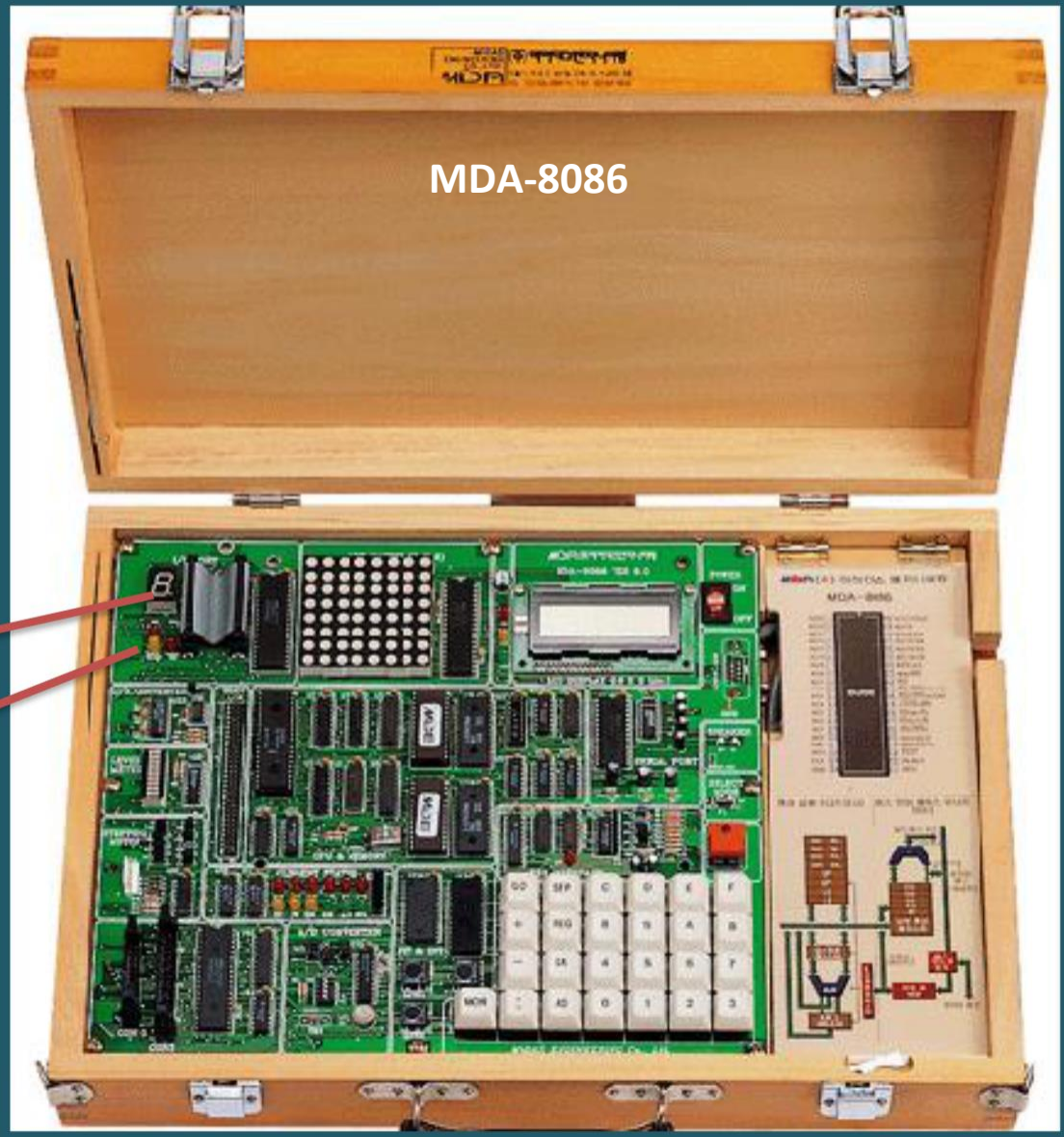
# ICE 3102
# Microprocessor and Interfacing Lab

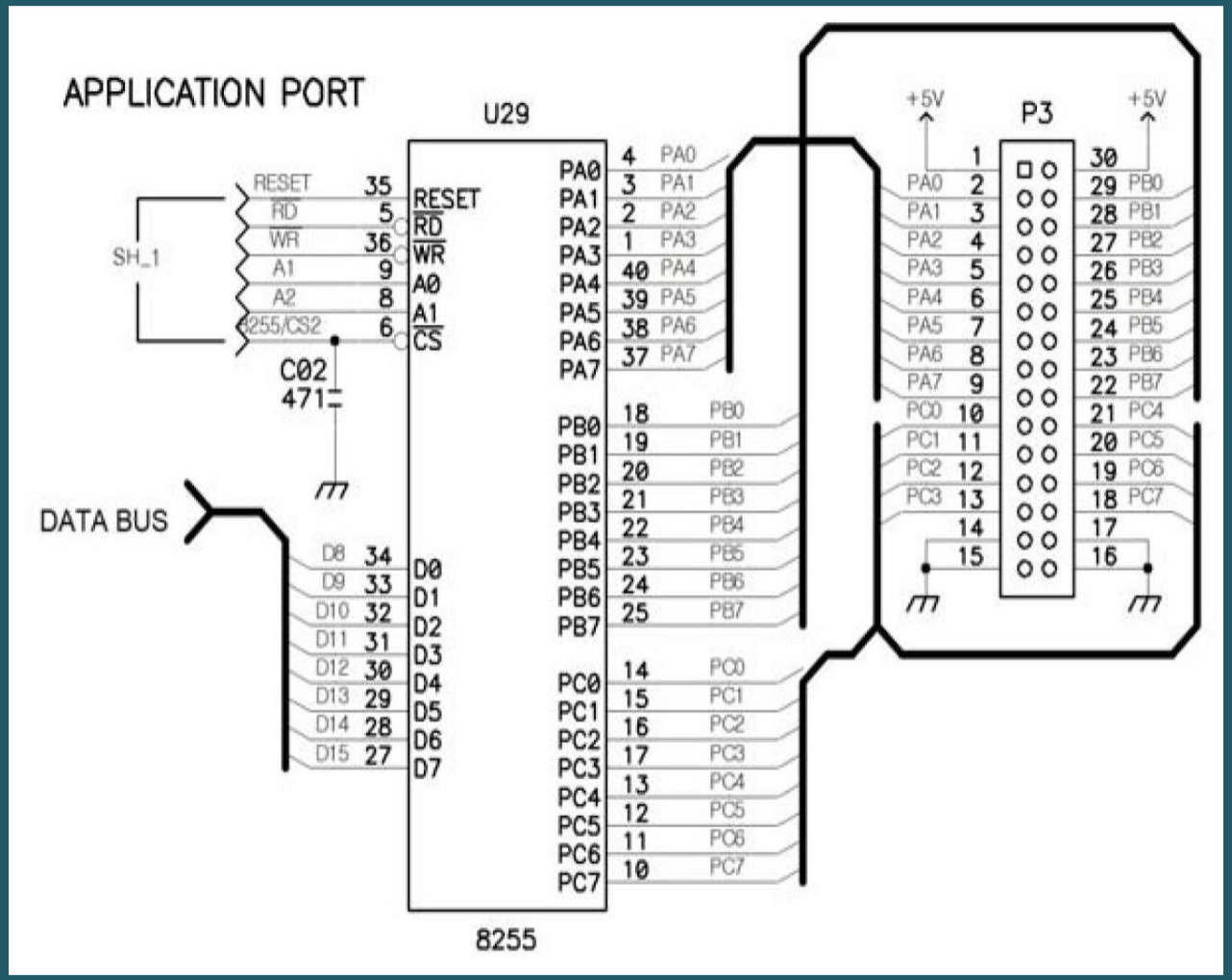
## Interfacing of LED and 7-segment display with 8086 microprocessor

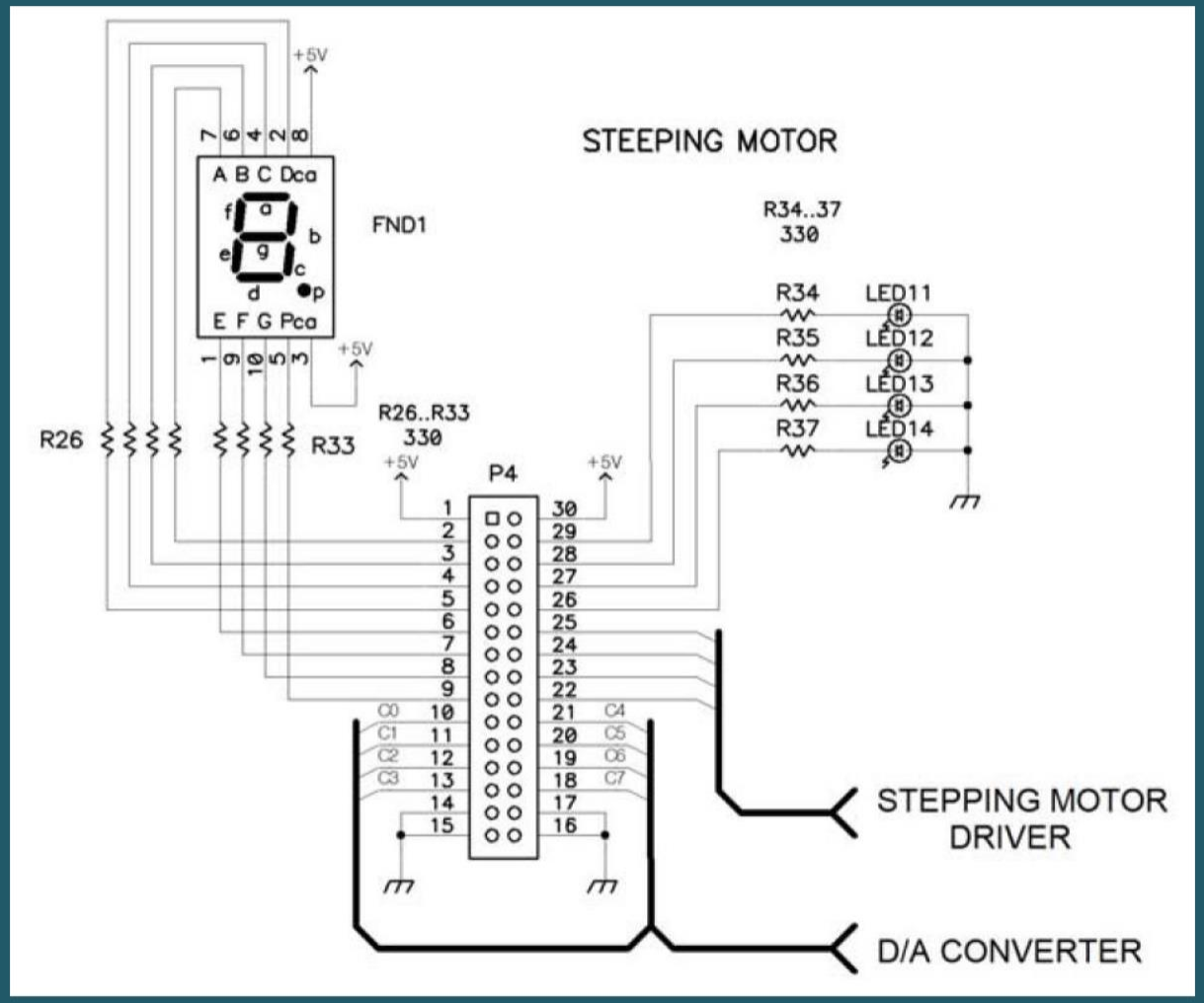
## Objectives

1) To interface LED with 8086 microprocessor by 8255 PPI.
2) To interface 7-segment display with 8086 microprocessor by 8255 PPI.



# Schematic of LED and 7-segment display interface with 8086

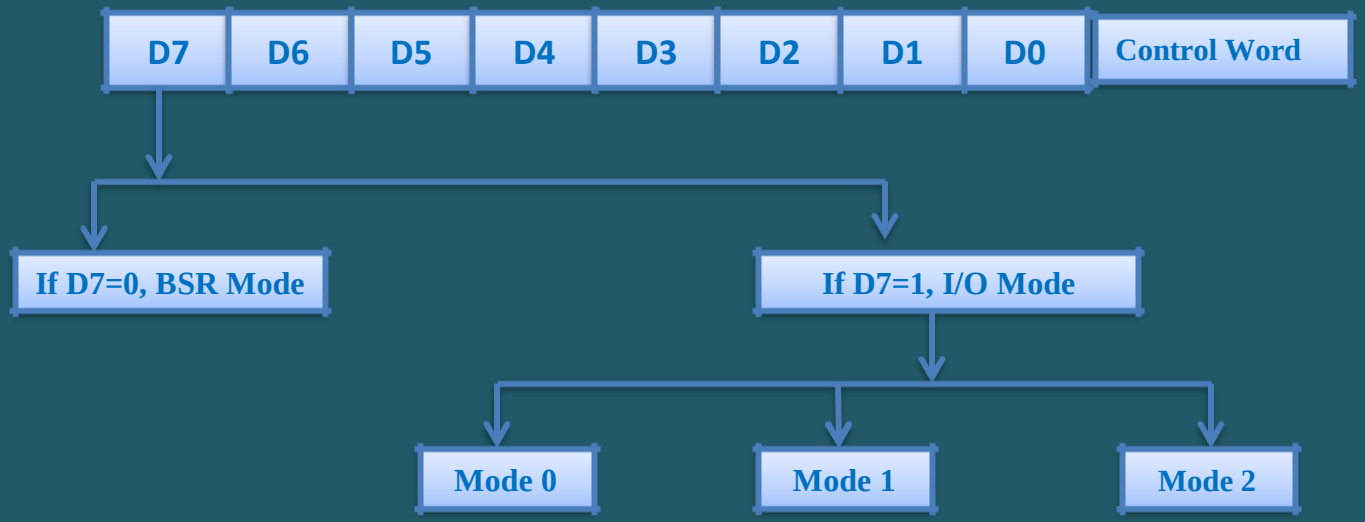


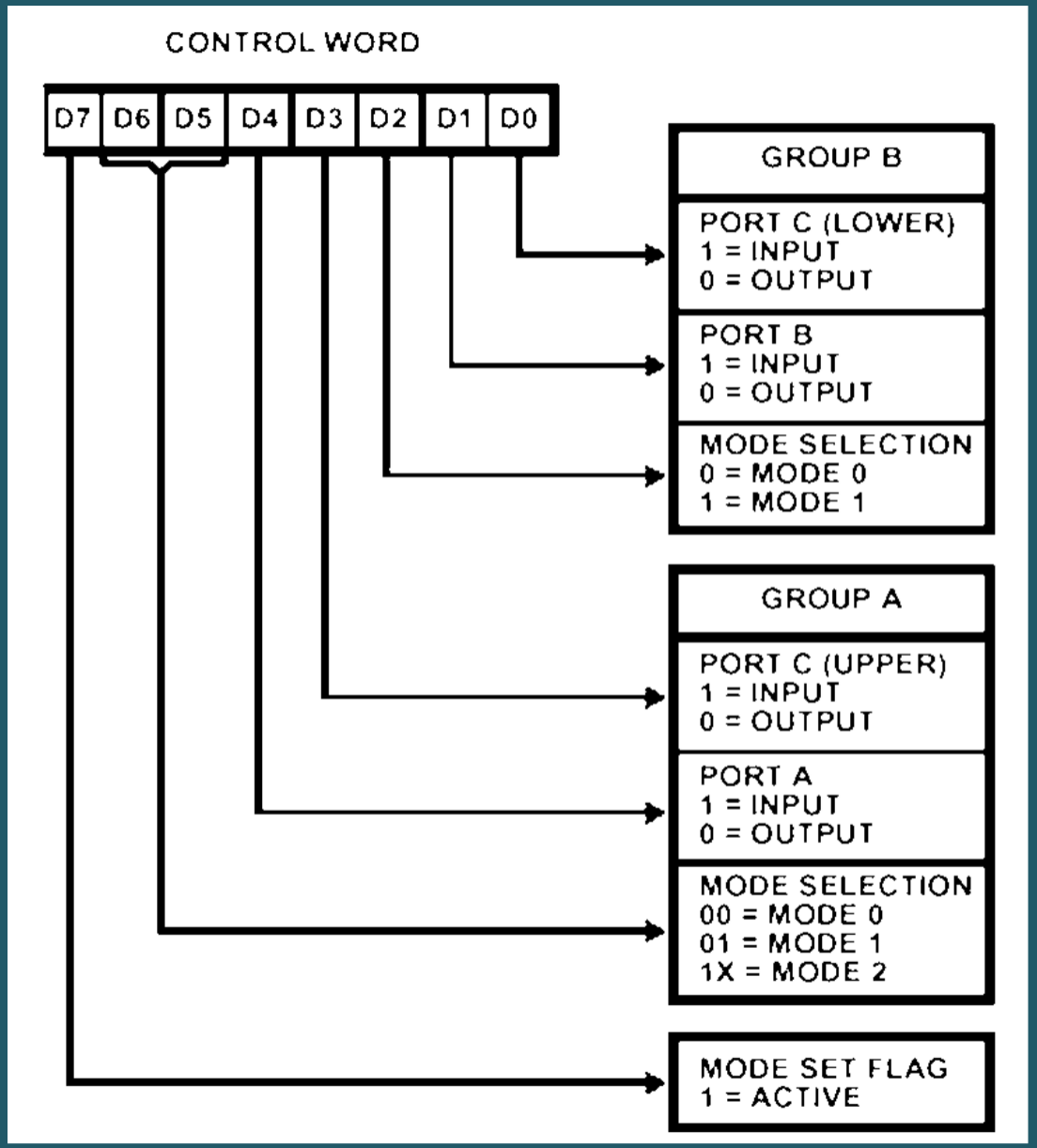



# Address of the internal registers of 8255 (In MDA-8086)

| | | |
|---|---|---|
| 18H ~ 1FH | 8255A-CS1/<br><br>8255A-CS2 | 8255A-CS1(DOT & ADC INTERFACE)<br>18H : A PORT DATA REGISTER<br>1AH : B PORT DATA REGISTER<br>1CH : C PORT CONTROL REGISTER<br>8255-CS2(LED & STEPPING MOTOR)<br>19H : A PORT DATA REGISTER<br>1BH : B PORT DATA REGISTER<br>1DH : C PORT CONTROL REGISTER<br>1FH : CONTROL REGISTER |

# Control Word for LED Interfacing



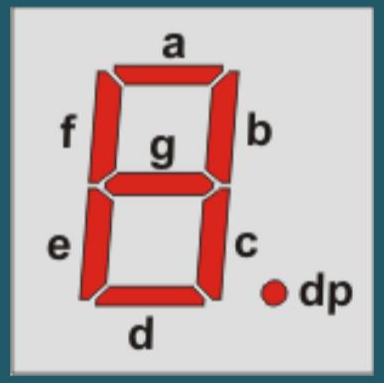
# LED Blinking

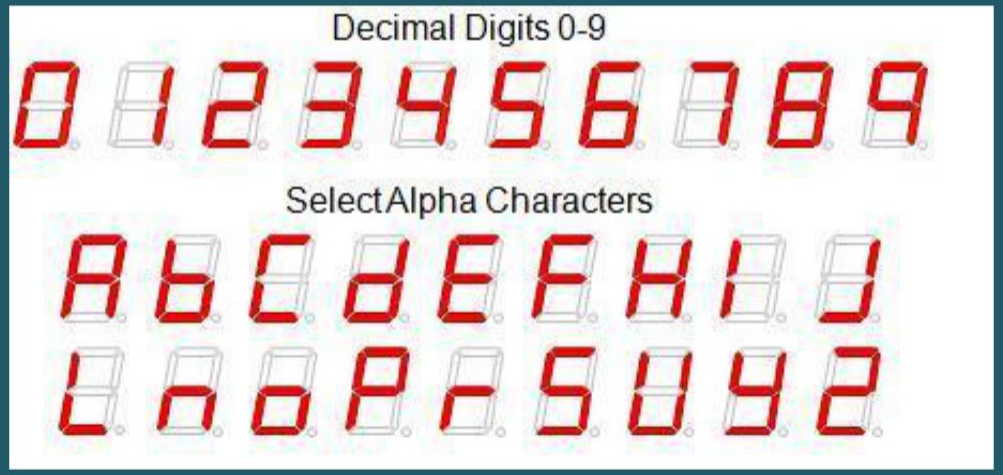
```
CODE    SEGMENT
        ASSUME  CS:CODE,DS:CODE,ES:CODE,SS:CODE

PPIC_C  EQU     1FH
PPIC    EQU     1DH
PPIB    EQU     1BH
PPIA    EQU     19H

        ORG     1000H
        MOV     AL,10000000B
        OUT     PPIC_C,AL

L1:     MOV     AL,00000001B
        OUT     PPIB,AL
        CALL    DELAY
        MOV     AL,00000000B
        OUT     PPIB,AL
        CALL    DELAY
        JMP     L1
```

```
DELAY:  MOV     CX,1111111111111111B
TIMER1: NOP
        NOP
        NOP
        NOP
        LOOP    TIMER1
        RET

CODE    ENDS
        END
```

# 7-segment Display

The 7-segment display, also written as “seven segment display”, consists of seven LEDs (hence its name) arranged in a rectangular fashion as shown. Each of the seven LEDs is called a segment because when illuminated the segment forms part of a **numerical digit (both Decimal and Hex) to be displayed**. An additional 8th LED is sometimes used within the same package thus allowing the indication of a decimal point, (DP) when two or more 7-segment displays are connected together to display numbers greater than ten.

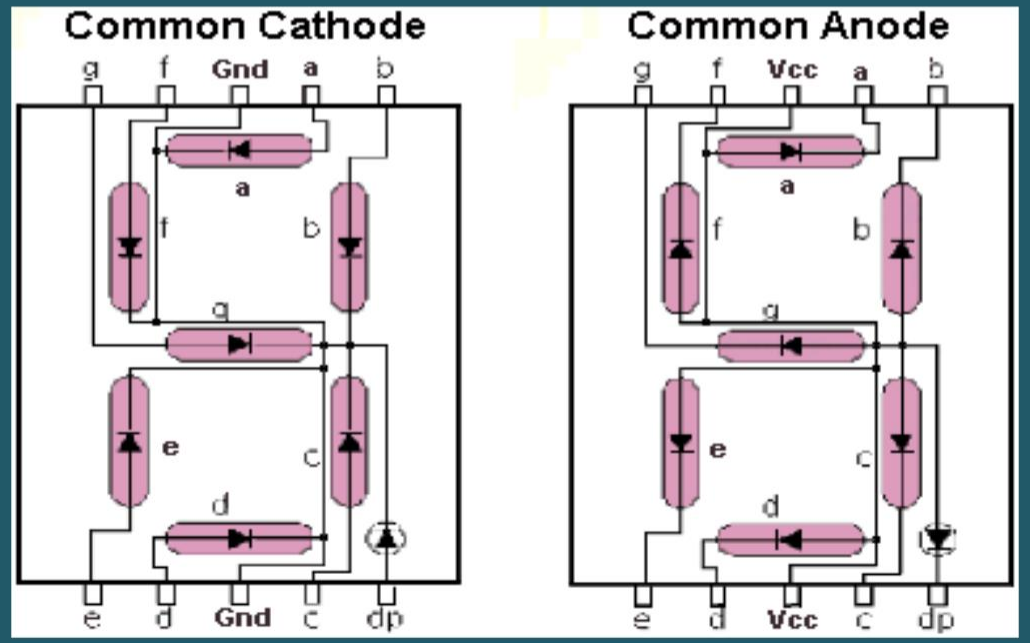


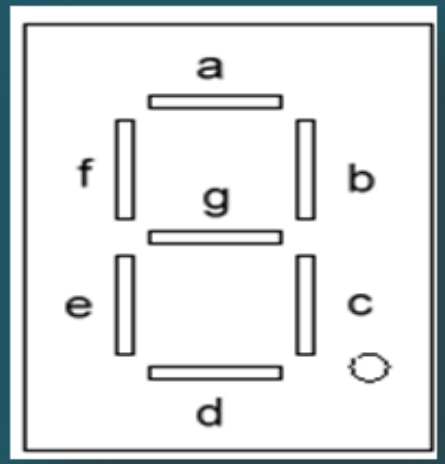



# Common Anode and Common Cathode 7-segment Display

**The Common Cathode (CC)** – In the common cathode display, all the cathode connections of the LED segments are joined together to logic “0” or ground. The individual segments are illuminated by application of a “HIGH”, or logic “1”

**The Common Anode (CA)** – In the common anode display, all the anode connections of the LED segments are joined together to logic “1”. The individual segments are illuminated by applying a ground, logic “0” or “LOW”

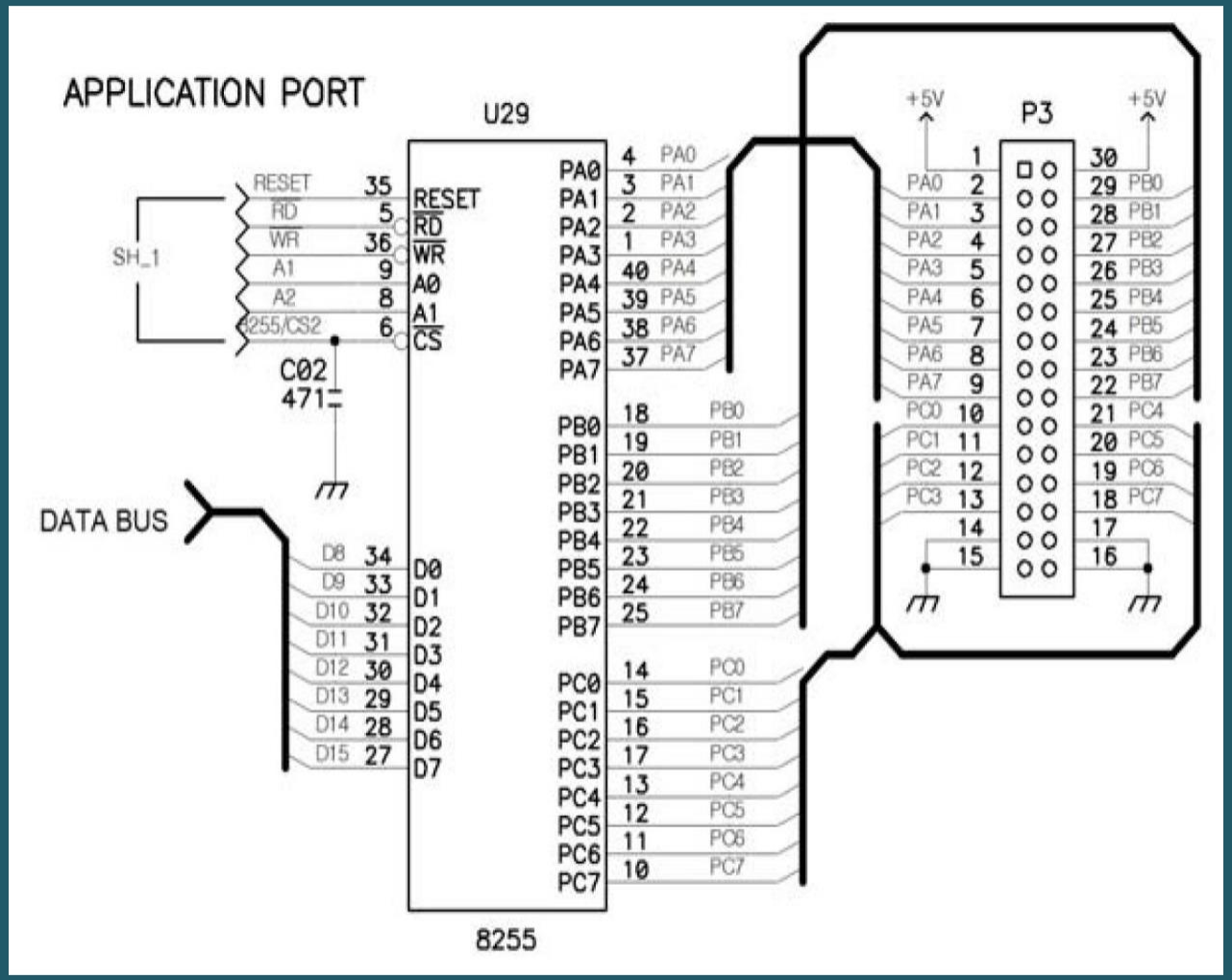


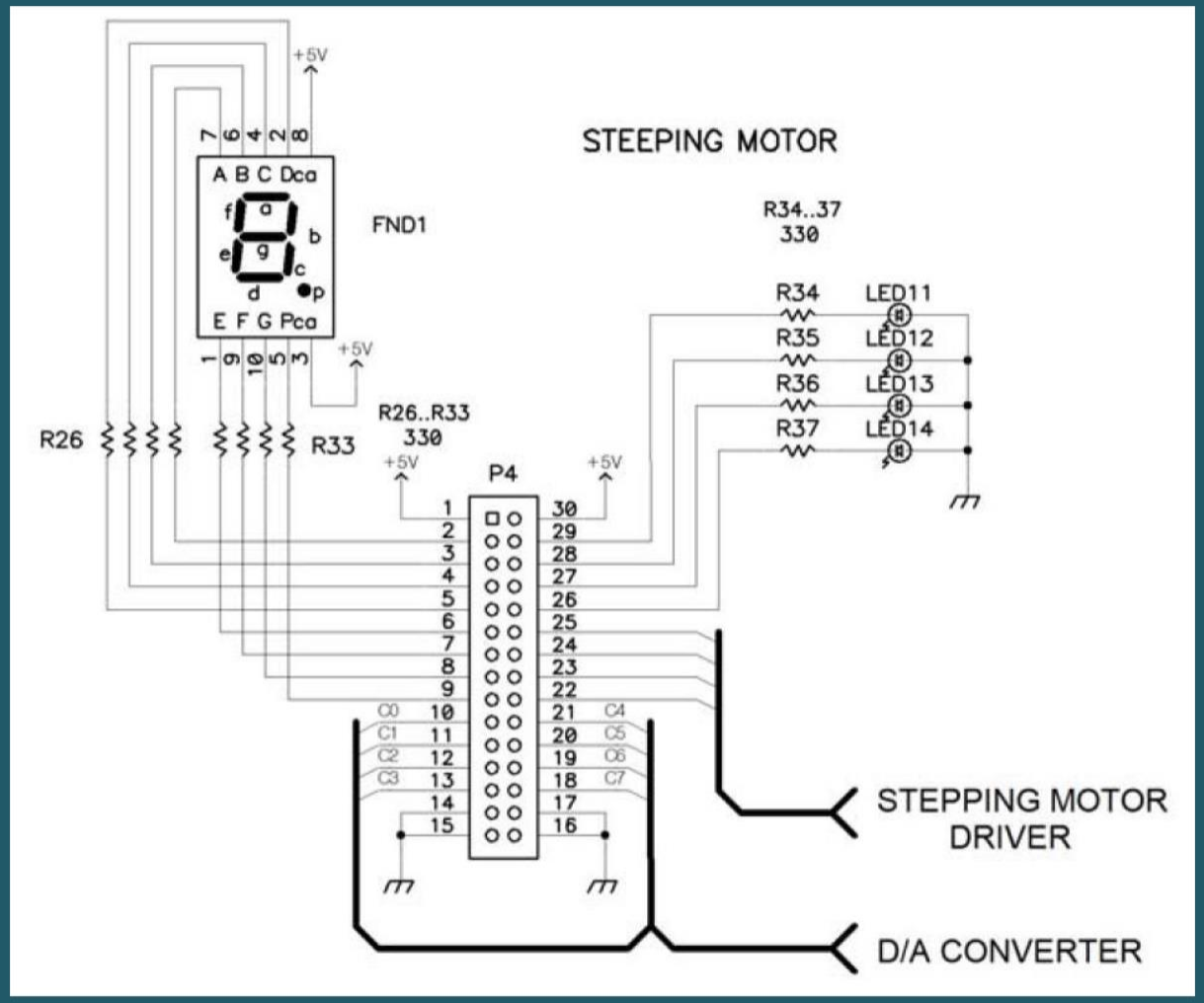


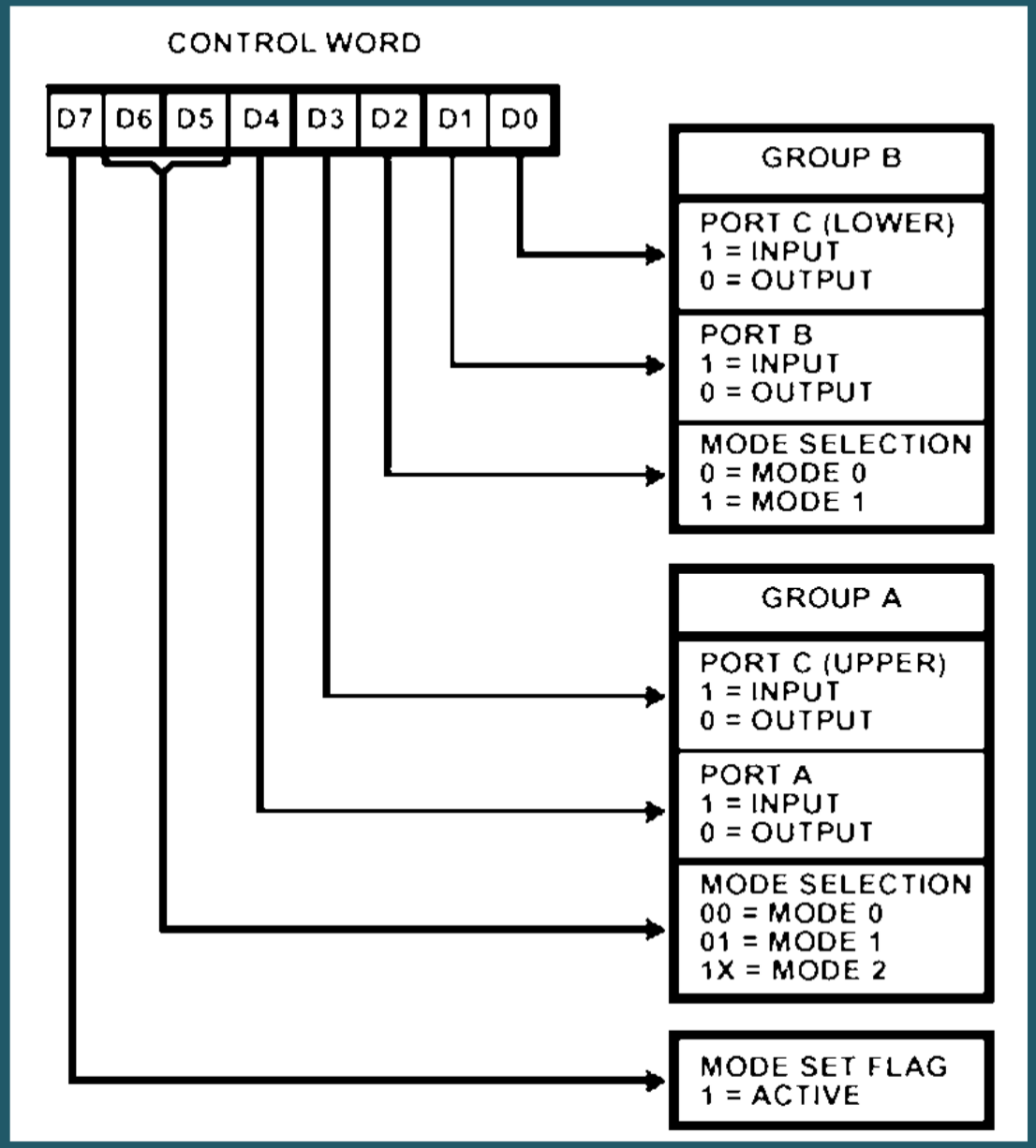

| 8 | 4 | 2 | 1 | 8 | 4 | 2 | 1 | | |
|---|---|---|---|---|---|---|---|---|---|
| dp | g | f | e | d | c | b | a | DECIMAL VALUE | HEX VALUE |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 00 |
| 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | F9 |
| 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 2 | A4 |
| 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 3 | B0 |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 4 | 99 |
| 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 5 | 92 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 6 | 82 |
| 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 7 | F8 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 8 | 80 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 9 | 90 |

**Figure: Displaying number for CA configuration**

# Schematic of LED and 7-segment display interface with 8086




CONTROL WORD
D7
D6
D5
D4
D3
D2
D1
D0
GROUP B
PORT C (LOWER)
1 = INPUT
0 = OUTPUT
PORT B
1 = INPUT
0 = OUTPUT
MODE SELECTION
0 = MODE 0
1 = MODE 1
GROUP A
PORT C (UPPER)
1 = INPUT
0 = OUTPUT
PORT A
1 = INPUT
0 = OUTPUT
MODE SELECTION
00 = MODE 0
01 = MODE 1
1X = MODE 2
MODE SET FLAG
1 = ACTIVE

# 7-Segment Display

```
CODE    SEGMENT
        ASSUME  CS:CODE,DS:CODE,ES:CODE,SS:CODE

PPIC_C  EQU     1FH
PPIC    EQU     1DH
PPIB    EQU     1BH
PPIA    EQU     19H

        ORG     1000H
        MOV     AL,10000000B
        OUT     PPIC_C,AL

L1:     MOV     AL,01000000B
        OUT     PPIA,AL
        CALL    DELAY
        MOV     AL,11111001B
        OUT     PPIA,AL
        CALL    DELAY
        JMP     L1
```

```
DELAY:  MOV     CX,1111111111111111B
TIMER1: NOP
        NOP
        NOP
        NOP
        LOOP    TIMER1
        RET

CODE    ENDS
        END
```