An-Najah National University
Faculty of Engineering and IT

جامعة النجاح الوطنية
كلية الهندسة وتكنولوجيا المعلومات

Computer Engineering Department

Course Name: Microprocessor Lab **Number: 10636392**

Lab Report Grading Sheet

Instructor: Dr. Aladdin Masri	**Experiment #**: 04
Academic Year: 2022/2023	**Experiment Name**: Keyboard/Display using 8279
Semester: 1st semester	

Students	
1- Sohaib Taqi Eldein Arafat	**2-** Mohammad Abd-Allateef Alawneh
3-	**4-**
Performed on: 14/10/2022	**Submitted on**: 24/10/2022

Report's Outcomes

ILO __ =( ) %	ILO __ =( ) %	ILO __ =( ) %	ILO __ =( ) %	ILO __ =( ) %

Evaluation Criterion	Grade	Points
Abstract answers of the questions: "What did you do? How did you do it? What did you find?"	0.5	
Introduction and Theory Sufficient, clear and complete statement of objectives. In addition to Presents sufficiently the theoretical basis.	1.5	
Apparatus/ Procedure Apparatus sufficiently described to enable another experimenter to identify the equipment needed to conduct the experiment Procedure sufficiently described.	2	
Experimental Results and Discussion (In-Lab Worksheet) Crisp explanation of experimental results. Comparison of theoretical predictions to experimental results, including discussion of accuracy and error analysis in some cases.	4	
Conclusions and Recommendations Conclusions summarize the major findings from the experimental results with adequate specificity. Recommendations appropriate in light of conclusions. Correct grammar.	1	
Appearance Title page is complete, page numbers applied, content is well organized, correct spelling, fonts are consistent, good visual appeal.	1	
Total	10	

Objectives

- To know the basic principle of 8279 and microcomputer interface.
- Use 8279 to interface the six 7-segment display array.
- Use 8279 to interface the 4x4 matrix keyboard.

Abstracts

Use 8279 keyboard/display to **show characters on the seven-segment displays** (Letters and Numbers), and to make a **one-bit BCD-Counter**. As well as to make **4X4 keyboard and display the pressed button on a seven-segment display**.

Introduction

The 8279 keyboard and display interface IC

Intel designed it to:

✓ **Interface a keyboard with the CPU.**

The IC first scans the keyboard and determines whether any keys have been pressed. After pressing a key, it sends the location of the pressed key back to the CPU (row & column) and vice versa.

✓ **Interface a six seven-segment displays with CPU.**

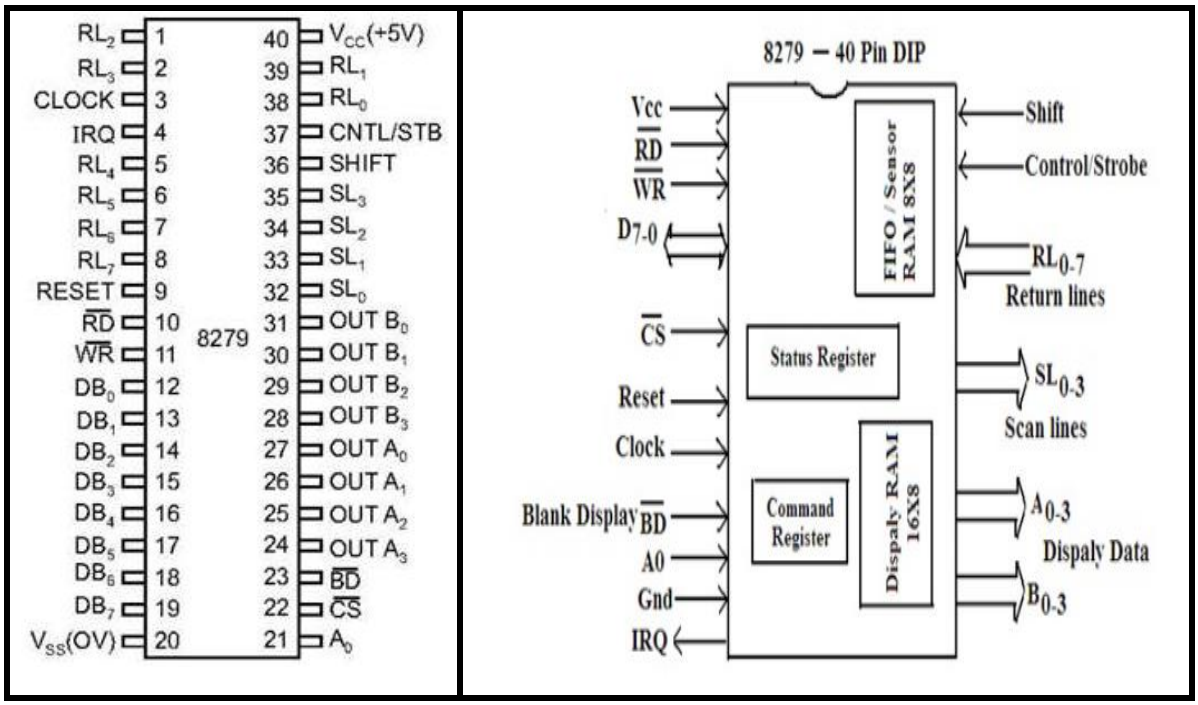
Some features:

- MCS-85 compatible 8279-5
- Simultaneous keyboard display operations
- Scanned keyboard mode
- Scanned sensor mode
- 16-character display.
- Right or left entry 16-byte display RAM.
- Programmable scan timing
- 8-character keyboard FIFO

Pins Definitions:

A0	Selects data (0) or control/status (1) for reads and writes between micro and 8279
BD	Output that blanks the displays
CLK	Used internally for timing. Max is 3 MHz
CN/ST	Control/strobe, connected to the control key on the keyboard
CS	Chip select that enables programming, reading the keyboard, etc.
DB7-DB0	Consists of bidirectional pins that connect to the data bus on micro
IRQ	Interrupt request, becomes 1 when a key is pressed, data is available
OUT A3-A0/B3-B0	Outputs that send data to the most significant/least significant nibble of display
RD/WR	Connects to micro's IORC or RD signal, reads data/status registers
RESET	Connects to system RESET.
RL7-RL0	Return lines are inputs used to sense key depression in the keyboard matrix
Shift	Shift connects to Shift key on keyboard
SL3-SL0	Scan line outputs scan both the keyboard and displays

- 2-key lockout or N-Key rollover with contact debounce



Programming 8279

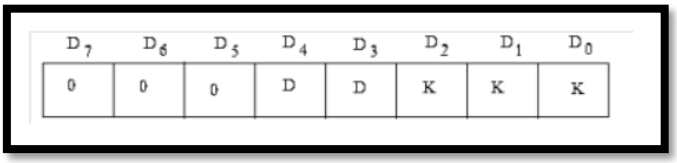
The **8279** is programmed using a command word register. We can specify which operation we want to select and the parameters related to it.

The D7-D5 pins are for selecting an operation as follows:

D_7	D_6	D_5	Function
0	0	0	Keyboard/Display Mode Set
0	0	1	Program Clock
0	1	0	Read FIFO/Sensor RAM
0	1	1	Read Display RAM
1	0	0	Write Display RAM
1	0	1	Display Write Inhibit/Blanking
1	1	0	Clear
1	1	1	End Interrupt/Error Mode Set

NOTE: In the Lab, we only use: **Mode Set**, **Program Clock** and **Clear**.

- **Mode Set**



KKK in D2 D1 D0 bit positions has the following four display mode options: (see the next page).

D_2	D_1	D_0	Keyboard Option
0	0	0	Encoded Scan Keyboard with 2-key lockout
0	0	1	Decoded Scan Keyboard with 2-key lockout
0	1	0	Encoded Scan Keyboard with N-key roll over
0	1	1	Decoded Scan Keyboard with N-key roll over
1	0	0	Encoded Scan Sensor Matrix
1	0	1	Decoded Scan Sensor Matrix
1	1	0	Strobed Input Encoded Display Scan
1	1	1	Strobed Input Decoded Display Scan

DD represents the display mode and KKK represents the keyboard mode. DD in D4 D3 bit positions has the following four display mode options:

D_4	D_3	Display Option
0	0	Eight 8-bit character display with left entry
0	1	Sixteen 8-bit character display with left entry
1	0	Eight 8-bit character display with right entry
1	1	Sixteen 8-bit character display with right entry

In our experiment we want to choose keyboard/display mode, encoded to key lockout, 8-bit left entry, so:

```
Mov dx, 8001h              ;address of cwr
Mov al,00000000b           ;program mode or (mov al,0)
Out dx,al
```

- **Program Clock**

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	0	1	P	P	P	P	P

The five bits (D0 to D4) of the command word define the scale factor.

In our experiment we want to program 8279 with clock factor scale 18:

```
Mov dx, 8001h              ;address of cwr
Mov al,00110010b           ;clock scale 18 or (mov al,32h)
Out dx,al
```

- **Clear**

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	1	0	C_D	C_D	C_D	C_F	C_A

We will replace C_D C_F C_A with ones.

In our experiment we want to clear 8279

```
Mov dx, 8001h              ;address of cwr
Mov al,11011111b           ;clear or (mov al,0dfh)
Out dx,al
```

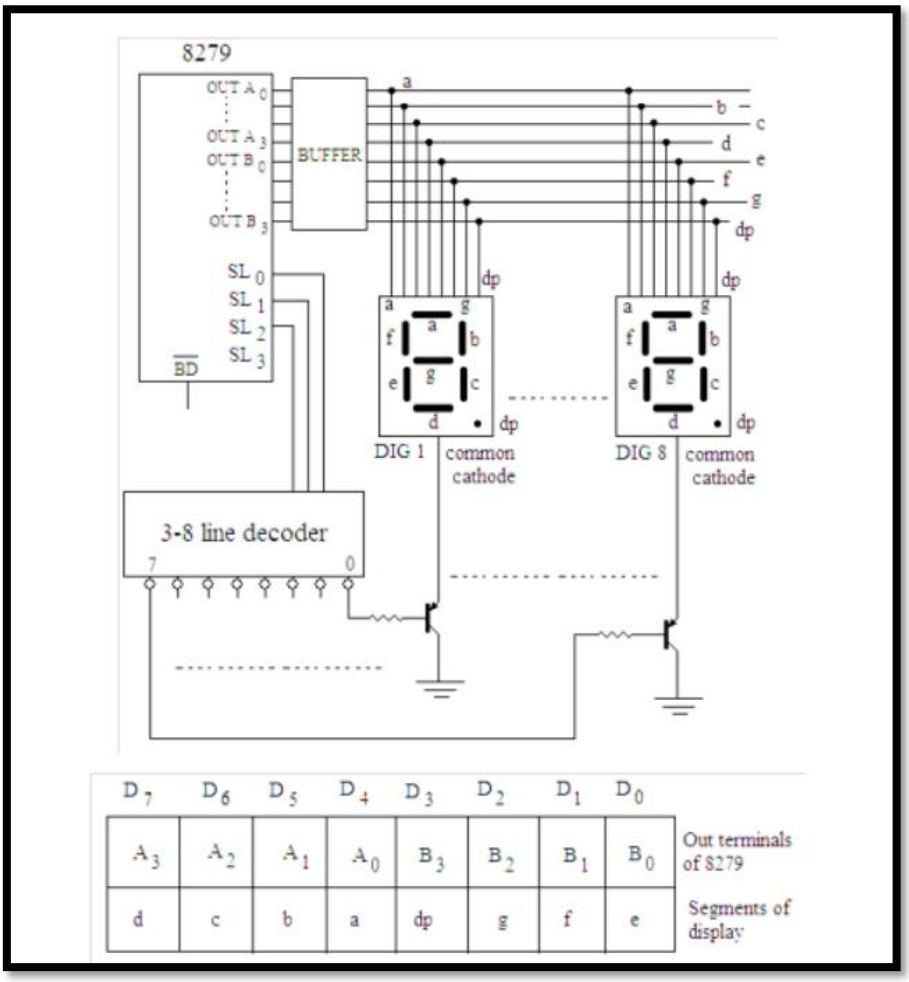
NOTE: We should clear 8279 before using it in our experiment, also, we should add some delay after clearing the IC.

Display Section

Generally, seven segment display devices are connected with 8279 using the multiplexing technique.

In the multiplexing technique the seven-segment code is sent to all the displays simultaneously, but

the particular segment to be illuminated is only grounded (in case of common cathode displays).



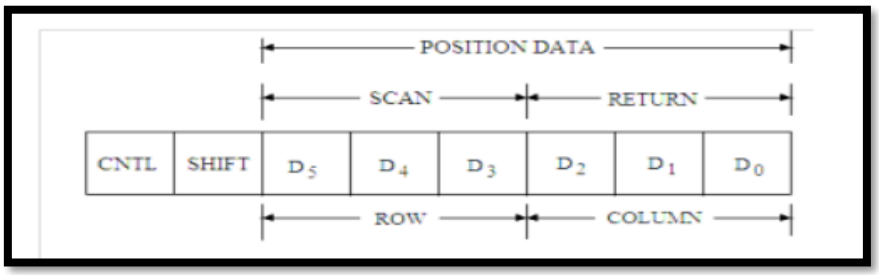
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	
A_3	A_2	A_1	A_0	B_3	B_2	B_1	B_0	Out terminals of 8279
d	c	b	a	dp	g	f	e	Segments of display

How to display 1 on a seven segment?

```
;1- choose the seven segment
Mov dx,8001h          ;address of cwr
Mov al,82h            ;choose a seven-segment
```

```
Out dx,al
;2- show the data
Mov dx,8000h          ;address of data
Mov al,10011111b      ;just b and c 0 (common cathode)
Out dx,al
```

Keyboard Section



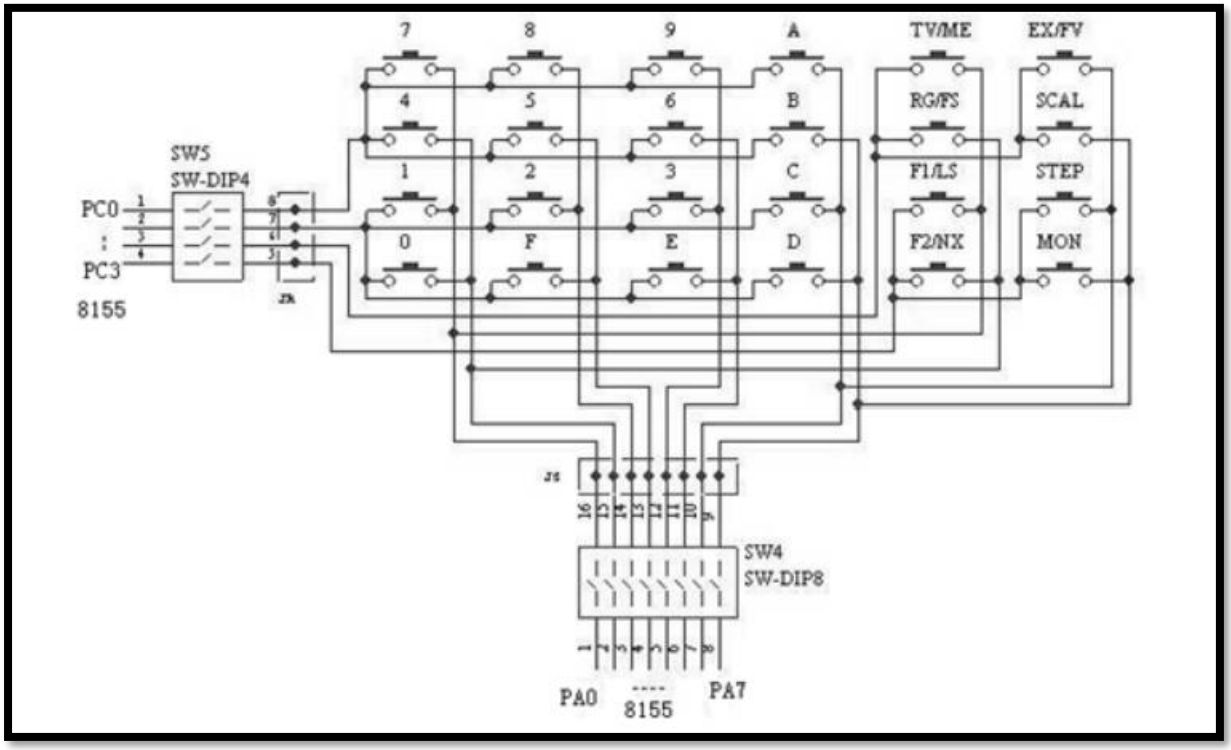


Table of the buttons -in the lab- and the corresponding values

	Col 0	Col 1
Row0	7	1
Row1	4	0
Row2	8	2
Row3	5	F
Row4	9	3
Row5	6	e

NOTE: before reading this register, we need to check if any button is pressed, by reading the control word register and check the least three bits, if they are zeros, no button is pressed.

For example: How to read check if button 7 is pressed?

```
checkForPressing: mov dx,8001h          ;address of cwr
                  in al,dx
                  and al,00000111b      ;masking for the least 3 bits
                  cmp al,0
                  je checkForPressing

            read: mov dx,8000h          ;address of data
                  in al,dx
                  and al,00111111b      ;masking for most 2 bits
                  cmp al,0              ;7 is column 0 and row 0
                  je btn7_isPressed
                  jmp checkForPressing
```

- **Procedures:**
  - **Part One: Display GP. 1** **(We are group 1)**

```
CODE SEGMENT
ASSUME CS:CODE
ORG 2000H

START:
;1- configue 8279 ------------------------------------------------------------
mov dx,8001h                    ;address of cwr
mov al,00000000b                ;keyboard/display + 8-bit left + encoded 2 keys
out dx,al

mov al,00110010b                ;clock with scale 18 (mov al,32h)
out dx,al

mov al,11011111b                ;clear
out dx,al

mov cx,0ffh                     ;delay after clear
delayLp1: nop
          loop delayLp1

;2- show data -----------------------------------------------------------------

;show P
mov dx,8001h                    ;address of cwr
mov al,81h                      ;choose a seven segment
out dx,al

mov dx,8000h                    ;address of data
mov al,10011111b                ;1
out dx,al

;show G
mov dx,8001h                    ;address of cwr
mov al,82h                      ;choose a seven segment
out dx,al

mov dx,8000h                    ;address of data
mov al,00001001b                ;G
out dx,al

;show 1
mov dx,8001h                    ;address of cwr
mov al,83h                      ;choose a seven segment
out dx,al

mov dx,8000h                    ;address of data
mov al,11001000b                ;P
out dx,al

END START

CODE ENDS
```

❖ **Part Two: one-bit BCD-counter (0-9)**

```
CODE SEGMENT
ASSUME CS:CODE
ORG 2000H

START:
;1- configue 8279 ------------------------------------------------------------
mov dx,8001h                    ;address of cwr
mov al,00000000b                ;keyboard/display + 8-bit left + encoded 2 keys
out dx,al

mov al,00110010b                ;clock with scale 18 (mov al,32h)
out dx,al

mov al,11011111b                ;clear
out dx,al

mov cx,0ffh                     ;delay after clear
delayLp1: nop
          loop delayLp1

;2- show data -----------------------------------------------------------------
lp: mov si,offset values            ;pointer to array
    mov cx,10                       ;10 iterations
    mov dx,8001h                    ;address of cwr
    mov al,82h                      ;choose seven segment
    out dx,al

    displayData: mov dx,8000h       ;address of data
                 mov al,[si]        ;value from values
                 out dx,al
                 inc si             ;next element in the array

                 push cx
                 mov cx,0fffh       ;delay after display
                 delayLp2: nop
                           loop delayLp2
              pop cx

              loop displayData
    jmp lp                          ;infinite loop


;3- declare array -------------------------------------------------------------
values db 0ch,9fh,4ah,0bh,99h,29h,28h,8fh,08h,89h


END START

CODE ENDS
```

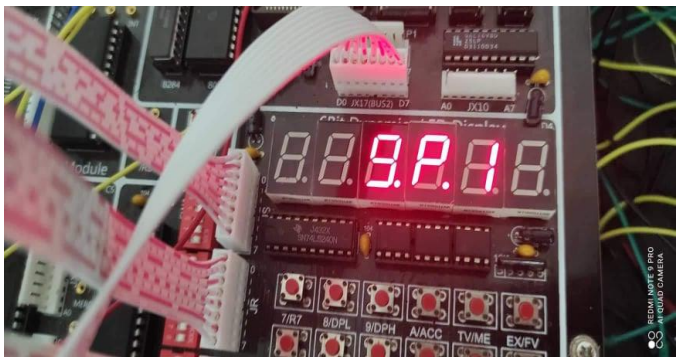
❖ **Part Three: 4*4 Keyboard and show the pressed key on a seven-segment display**

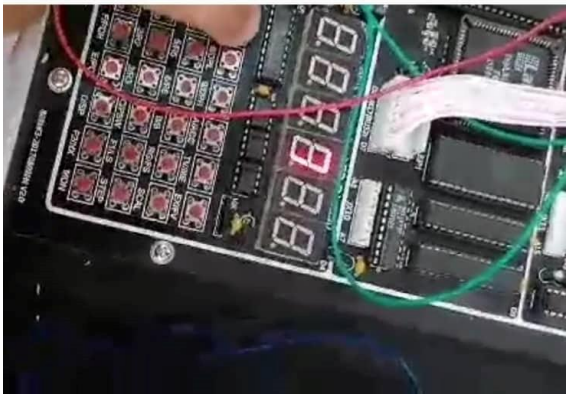
```
CODE SEGMENT
ASSUME CODE:CS
ORG 2000H

START:
;1- configue 8279 ------------------------------------------------------------------------
mov dx,8001h                    ;address of cwr
mov al,00000000b                ;keyboard/display + 8-bit left + encoded 2 keys
out dx,al

mov al,00110010b                ;clock with scale 18 (mov al,32h)
out dx,al

mov al,11011111b                ;clear
out dx,al

mov cx,0ffh                     ;delay after clear
delayLp1: nop
          loop delayLp1

;2- show data -----------------------------------------------------------------------------

;FIRST: check if any button is pressed
checkForPressing: mov dx,8001h          ;address of cwr
                  in al,dx
                  and al,00000111b      ;masking
                  cmp al,0
                  je checkForPressing
```

```
;SECOND: read the value of the pressed key, search about it, then display the corresponding value
            read: mov dx,8000h          ;address of data
                  in al,dx
                  and al,00111111b      ;masking
                  mov cx,16             ;16 iteration
                  lea si,keys           ;pointer to keys array
                  lea bx,values         ;pointer to values array

                  search: cmp al,[si]   ;compare what we read with a key
                          je displayData
                          inc si        ;next key
                          inc bx        ;next value
                          loop search
                  jmp checkForPressing  ;check for new action

            displayData: mov dx,8001h ;address of cwr
                         mov al,82h   ;select seven segment
                         out dx,al

                         mov dx,8000h ;address of data
                         mov al,[bx]  ;corresponding value
                         out dx,al

                  jmp checkForPressing  ;check for new action

;3- declare two arrays (keys + values) ------------------------------------------------------
keys db 09h, 01h, 11h, 21h, 08h, 18h, 28h, 0, 10h, 20h, 30h, 38h, 31h, 39h, 29h, 19h

values db 0ch, 9fh, 4ah, 0bh, 99h, 29h, 28h, 8fh, 08h, 89h, 88h, 38h, 6ch, 1ah, 68h, 0e8h

END START

CODE ENDS
```

Why to declare two arrays?
In order to make **synchronized** search about pressed button, and display the corresponding value on the seven-segment display.

- ## Results:





- ## Conclusion:

We can use the **8279** to interface two IO devices with the 8086 processor (seven-segment display and 4*4 keyboard). without any considerable delay. The 8279 is very versatile and useful due to its many modes and functions.

- ## References

- Microprocessors Lab (10636392) manual
- Microprocessors Course (10636322) slides
- Dr. Aladdin Masri lectures
- Tutorials Point 8279 - Programmable Keyboard (Press here)
- University of New Mexico Programmable Keyboard/Display Interface (Press here).