Lecture 11

8255 Programmable Peripheral Interface

Topics

- 8255 Programmable Peripheral Interface
- Application Examples

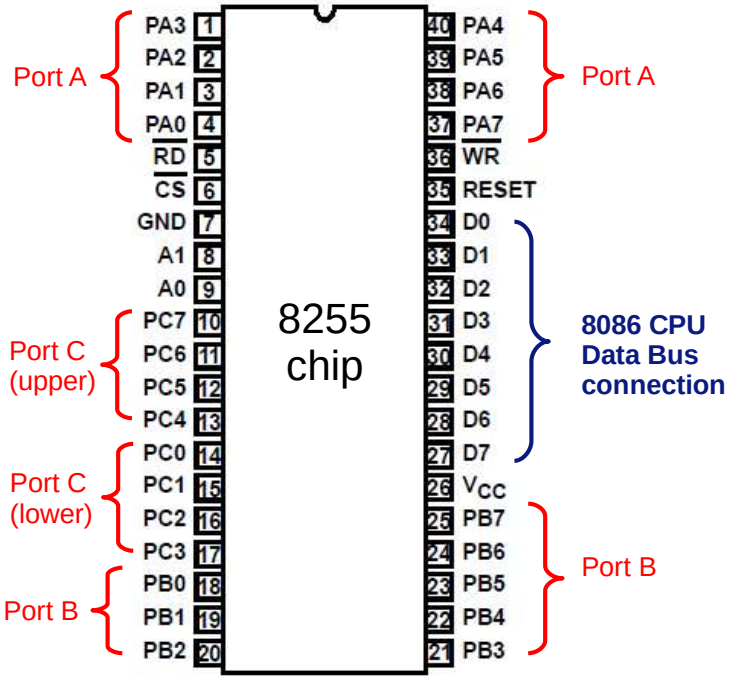
8255 Programmable Peripheral Interface Chip

- Intel 8255 is a general-purpose programmable Input/Output interfacing chip.
- Can be used for parallel I/O interfacing with many microprocessors such as Intel 8086, or with microcontrollers such as Intel 8051.
- 8255 chip has 40-pins, 24 of them are I/O pins.
- The 24 pins (I/O lines) are organized as 3 I/O ports labeled as Port A, Port B, and Port C.
- Ports are 8-bit registers.
- Port C can be divided in half, with the left four bits or the right four bits used as inputs or outputs.
- 8255 chip can be used in 3 modes of operation.
- The Control Port is used to configure the operation mode of the chip.

Device examples using with 8255 interface chip

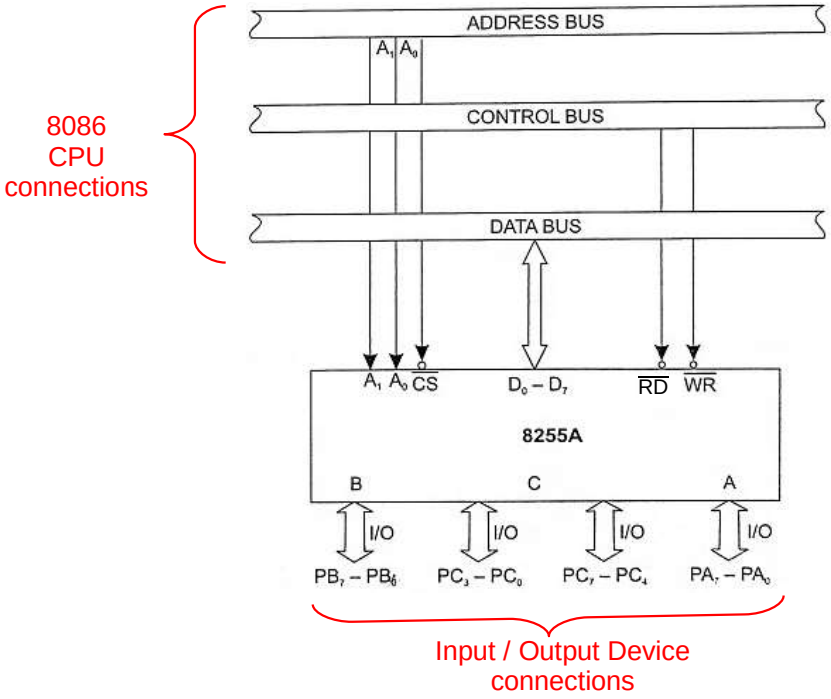
- LED interfacing
- 7-Segment LED Display device interfacing
- Switch, Button interfacing
- Matrix keypad interfacing
- Interfacing with motors
  - DC motor
  - Stepper motor
  - Servo motor
- Interfacing with A/D converter (Analog to Digital)
- Interfacing with D/A converter (Digital to Analog)

8255 Pins



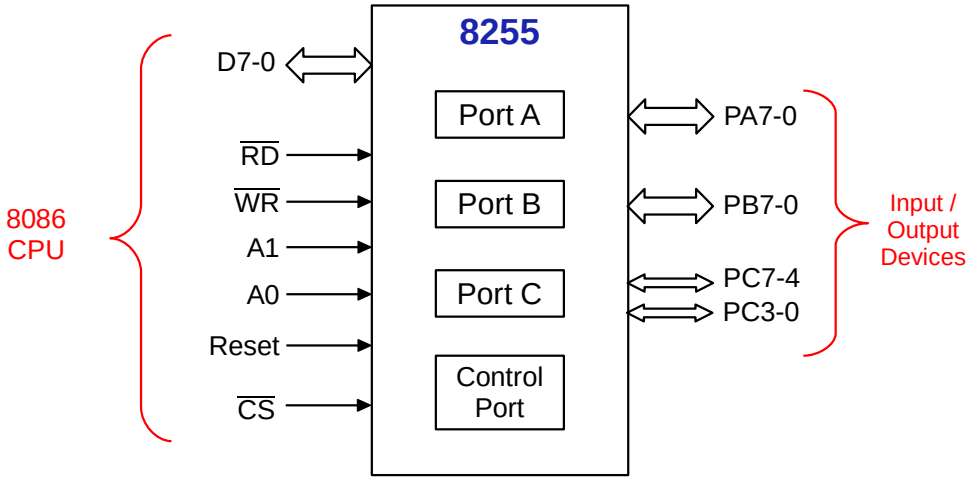
8255 and System Bus Connection

System Bus Connections: Outputs are latched, inputs are buffered.



Ports in 8255

- The Chip Select signal ($\overline{CS}$) should be connected to the Address Decoding logic.
- When CS (Chip Select) signal is 0, the 8255 chip is enabled.
- Port A, Port B, Port C and the Control Port have their own addresses.
- Each port is a 8-bit register.

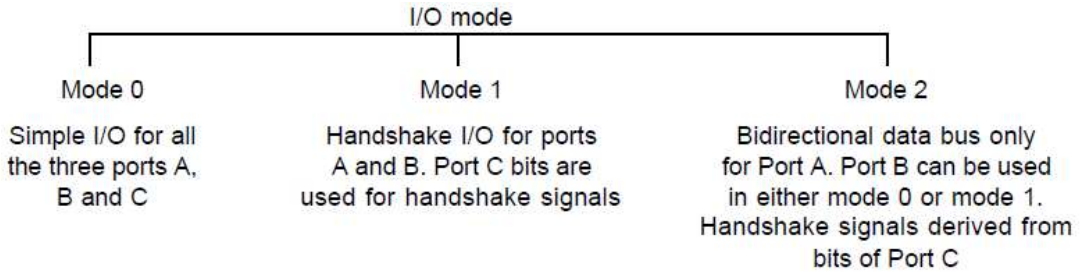


8255 Pin Descriptions

SYMBOL	TYPE	DESCRIPTION
V_{CC}		V_{CC}: The +5V power supply pin. A 0.1μF capacitor between V_{CC} and GND is recommended for decoupling.
GND		GROUND
D0-D7	I/O	DATA BUS: The Data Bus lines are bidirectional three-state pins connected to the system data bus.
RESET	I	RESET: A high on this input clears the control register and all ports (A, B, C) are set to the input mode with the "Bus Hold" circuitry turned on.
$\overline{CS}$	I	CHIP SELECT: Chip select is an active low input used to enable the 82C55A onto the Data Bus for CPU communications.
$\overline{RD}$	I	READ: Read is an active low input control signal used by the CPU to read status information or data via the data bus.
$\overline{WR}$	I	WRITE: Write is an active low input control signal used by the CPU to load control words and data into the 82C55A.
A0-A1	I	ADDRESS: These input signals, in conjunction with the $\overline{RD}$ and $\overline{WR}$ inputs, control the selection of one of the three ports or the control word register. A0 and A1 are normally connected to the least significant bits of the Address Bus A0, A1.
PA0-PA7	I/O	PORT A: 8-bit input and output port. Both bus hold high and bus hold low circuitry are present on this port.
PB0-PB7	I/O	PORT B: 8-bit input and output port. Bus hold high circuitry is present on this port.
PC0-PC7	I/O	PORT C: 8-bit input and output port. Bus hold circuitry is present on this port.

8255 Operating Modes

- The 8255 interface chip can be programmed to operate in three modes:
  - Mode 0 : Basic Input/Output
  - Mode 1 : Strobed Input/Output (Handshaked)
  - Mode 2 : Two-directional bus



Mode 0 (Basic Input/Output)

- Mode 0 gives the simplest form of I/O.
- No handshaking is required.
- Data is simply read from or written to the specified port.
- Any of the ports A, B, C (upper half), and C (lower half) can be set (configured) individually as input or output ports.
- Configuration is done by sending a control byte to the Control Port.
- There are 16 possible control words.

- **Example:** To configure all three ports (A,B,C) as output ports, you need to send the control word 80h to the Control port using the following instructions.

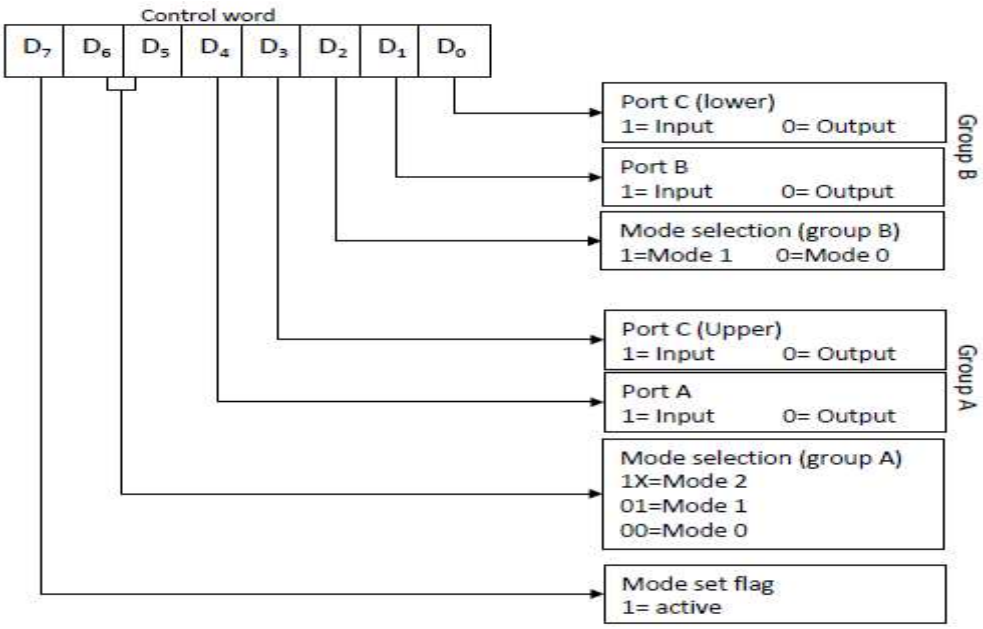
```
MOV AL, 80h     ; Load AL with the control word 80h.
OUT 63h, AL     ; Send the control word to port 63h address.
                ; (To the Control Port register)
```

8255 Control Port Register

- The control port register must be programmed to select the operation mode of the three ports A, B, and C.
- The CPU outputs a control word to the 8255 chip.
- The Control Port Register of 8255 is used to configure the individual ports as to be in input or output mode.
- The control word contains information such as operating mode that initializes the functional configuration of the 8255.

8255 Control Port Register

(Control Word Bits)



Mode 0 Control Words

(Table of all 16 possible configurations)

D7, D6, D5, and D2 bits of Control Register are fixed for Mode0.

Ports				Control Word								Hex Value
A	C (higher)	B	C (lower)	D7	D6	D5	D4	D3	D2	D1	D0	
OUT	OUT	OUT	OUT	1	0	0	0	0	0	0	0	80h
OUT	OUT	OUT	IN	1	0	0	0	0	0	0	1	81h
OUT	OUT	IN	OUT	1	0	0	0	0	0	1	0	82h
OUT	OUT	IN	IN	1	0	0	0	0	0	1	1	83h
OUT	IN	OUT	OUT	1	0	0	0	1	0	0	0	88h
OUT	IN	OUT	IN	1	0	0	0	1	0	0	1	89h
OUT	IN	IN	OUT	1	0	0	0	1	0	1	0	8Ah
OUT	IN	IN	IN	1	0	0	0	1	0	1	1	8Bh
IN	OUT	OUT	OUT	1	0	0	1	0	0	0	0	90h
IN	OUT	OUT	IN	1	0	0	1	0	0	0	1	91h
IN	OUT	IN	OUT	1	0	0	1	0	0	1	0	92h
IN	OUT	IN	IN	1	0	0	1	0	0	1	1	93h
IN	IN	OUT	OUT	1	0	0	1	1	0	0	0	98h
IN	IN	OUT	IN	1	0	0	1	1	0	0	1	99h
IN	IN	IN	OUT	1	0	0	1	1	0	1	0	9Ah
IN	IN	IN	IN	1	0	0	1	1	0	1	1	9Bh

Port selection

- The $\overline{CS}$ (chip select) pin selects the entire 8255 chip.
- Address pins A0 and A1 select the specific port within the chip.
- The A0 and A1 are bits as part of the port addresses.

CS	A1	A0	Selected Port
0	0	0	Port A
0	0	1	Port B
0	1	0	Port C
0	1	1	Control Port Register
1	x	x	8255 chip is disabled

8255 Port Addresses in Computer

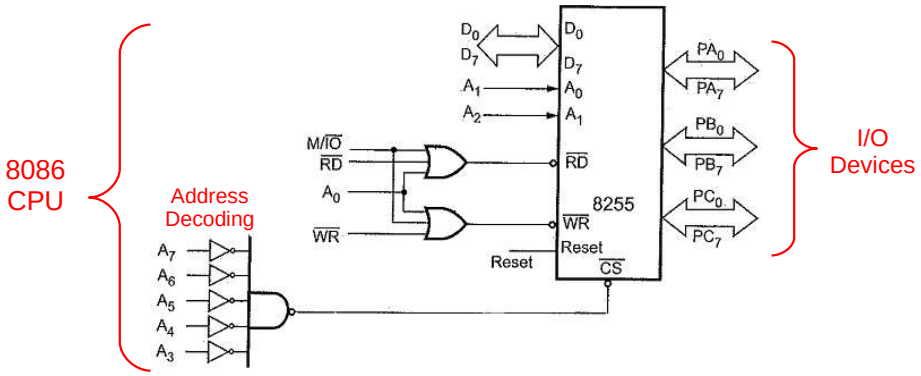
- In IBM-PC and compatible computers, the 8255 chip is decoded at Input/Output port addresses 60h through 63h.
- The smallest address 60h is called as Base Port address.

Port Names	Port Addresses	A1 and A0 address lines
A	60h	0 0
B	61h	0 1
C	62h	1 0
Control Register	63h	1 1

Example1: Interfacing of 8255 in Isolated I/O

(Using Logic Gates for address decoding)

- The 8086 CPU has special instructions IN and OUT to transfer data through the input/output ports in Isolated I/O method.
- The M / $\overline{IO}$ signal is always low, when 8086 CPU is executing IN or OUT instructions.

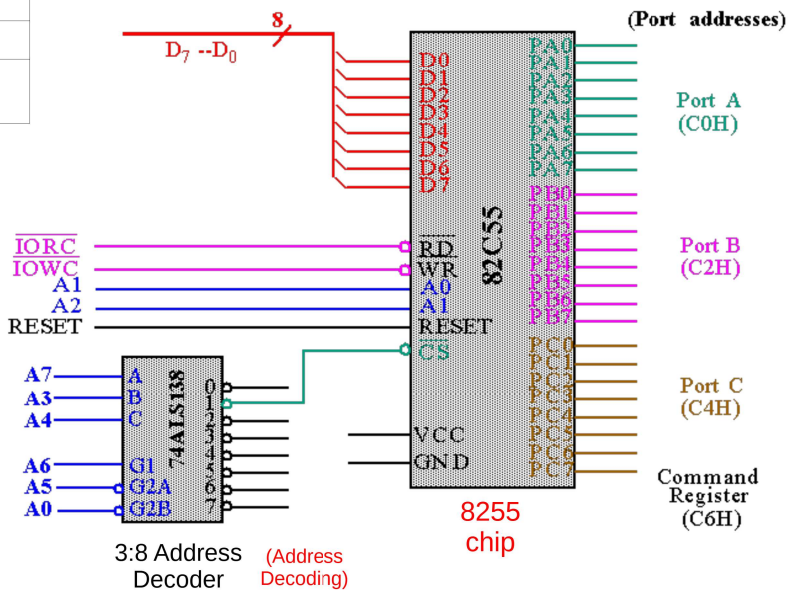


Port / control Register	Address lines								Address
	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	
Port A	0	0	0	0	0	0	0	0	00H
Port B	0	0	0	0	0	0	1	0	02H
Port C	0	0	0	0	0	1	0	0	04H
Control register	0	0	0	0	0	1	1	0	06H

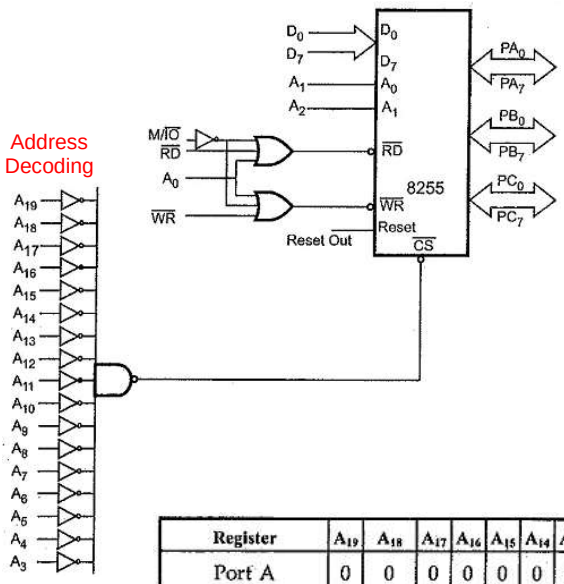
Example2: Interfacing of 8255 in Isolated I/O

(Using Address Decoder chip 74138)

Port	Address
A	C0h
B	C2h
C	C4h
Command Register	C6h



Example3: Interfacing of 8255 in Memory Mapped I/O



- The 8086 CPU uses 20 address lines to identify an I/O device.
- An I/O device is connected as if it is a memory register.
- 8086 CPU uses same control signals and instructions to access I/O as those of memory.
- RD and WR signals are activated when M / $\overline{IO}$ signal is high, indicating memory bus cycle.
- Address lines A0 and A1 are used by 8255 chip for internal decoding.

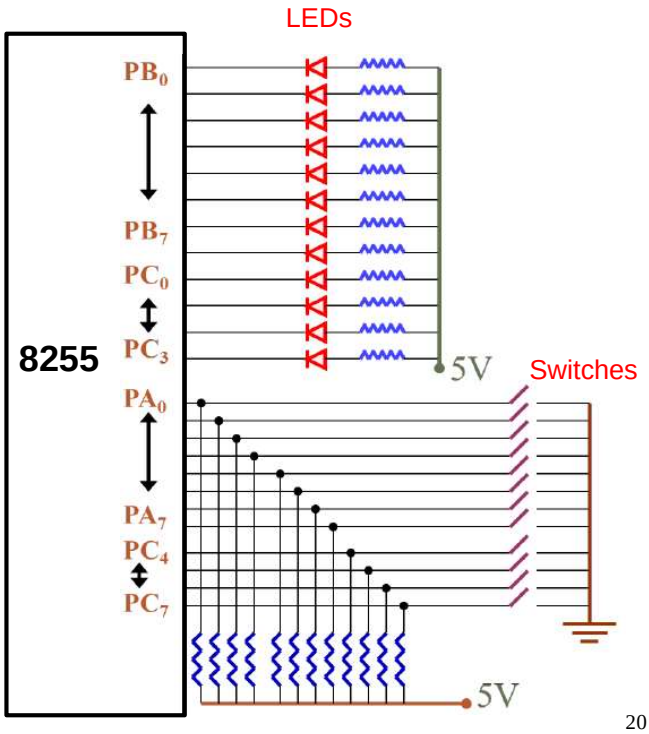
Register	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	Address
Port A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000H
Port B	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	00002H
Port C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	00004H
Control register	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	00006H

Topics

- 8255 Programmable Peripheral Interface
- Application Examples

Application Example1 : Switches and LEDs

- Write an Assembly program to continuously read 12 switches and display corresponding 12 LEDs.
- Each switch will control one LED.
- Suppose 12 LEDs are connected to all 8 pins of Port B, and 4 lower pins of Port C.
- Also 12 switches are connected to all 8 pins of Port A, and 4 upper pins of Port C.
- The Base Address (smallest address) of ports in 8255 chip is 00h.



Solution

Determine the Control Register bits (control word).

The Control Register in Mode 0

D7	D6	D5	D4	D3	D2	D1	D0
1=Mode 0 I/O Mode	Mode selection **00 = Mode 0** 01 = Mode 1 1X = Mode 2		Port A **1=input 0=output**	Port C (PC7-PC4) **1=input 0=output**	Mode Selection **0=Mode 0** 1=Mode 1	Port B **1=input 0=output**	Port C (PC3-PC0) **1=input 0=output**

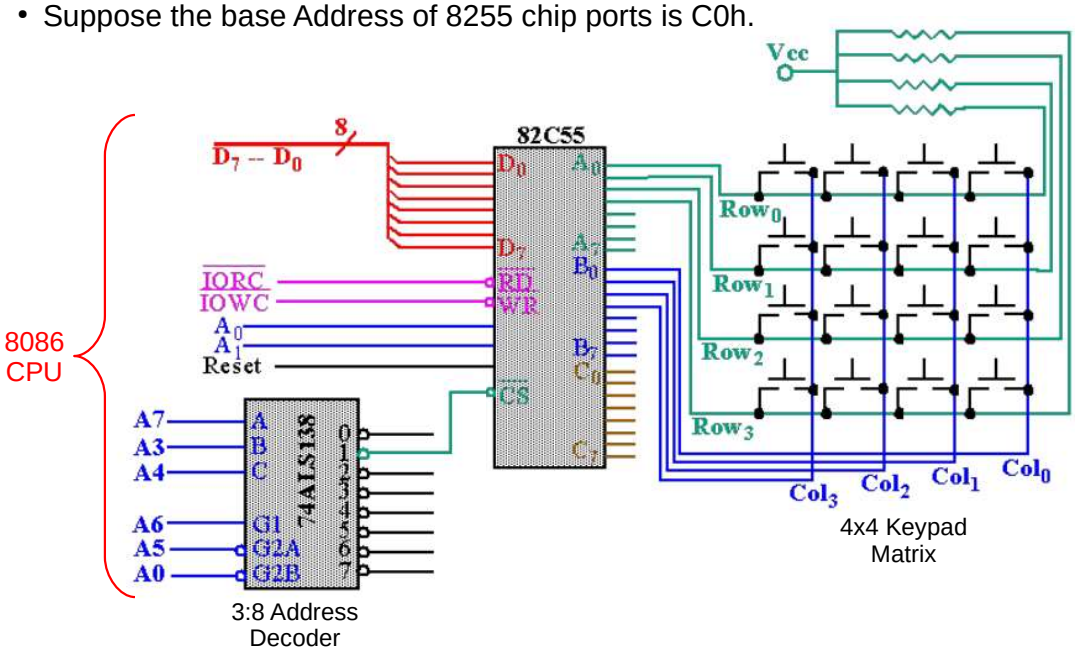
D7	D6	D5	D4	D3	D2	D1	D0
1	**0**	**0**	**1**	**1**	**0**	**0**	**0**

Program

```
; Define the port addresses of 8255 chip.
PortA          EQU  00h
PortB          EQU  02h
PortC          EQU  04h
KontrolPort    EQU  06h
.model small
.code
  mov AL, 10011000b          ; The control word
  out  KontrolPort, AL       ; Initialize the 8255 chip
MainDongu:                   ; Endless loop
  in   AL, PortA             ; Read 8 switches
  out  PortB, AL             ; Send to 8 LEDs
  in   AL, PortC             ; Read 4 switches
  and  AL, 0f0h              ; Upper 4 bits filtered (taken)
  mov  CL, 04                ; Counter for rotate operation
  ror  AL, CL                ; Rotate AL left 4 times
  out  PortC, AL             ; Send to 4 LEDs
  JMP MainDongu              ; Go to endless loop
END                          ; End of file
```

Application Example2 : Keypad Input

- Write an Assembly program to continuously read the keypad, and detect a button press.
- The 4 rows of keypad are connected to Port A, and the 4 columns are connected to Port B of 8255 chip.
- Suppose the base Address of 8255 chip ports is C0h.



Solution

Determine the Control Register bits (control word).

- Port A is used as input.
- Port B is used as output.
- Port C is not used.

The Control Register in Mode 0

D7	D6	D5	D4	D3	D2	D1	D0
1=Mode 0 I/O Mode	Mode selection **00 = Mode 0** 01 = Mode 1 1X = Mode 2		Port A **1=input 0=output**	Port C (PC7-PC4) **1=input 0=output**	Mode Selection **0=Mode 0** 1=Mode 1	Port B **1=input 0=output**	Port C (PC3-PC0) **1=input 0=output**

D7	D6	D5	D4	D3	D2	D1	D0
1	**0**	**0**	**1**	**x**	**0**	**0**	**x**

x : Assumed as zeros.

Keypad Layout

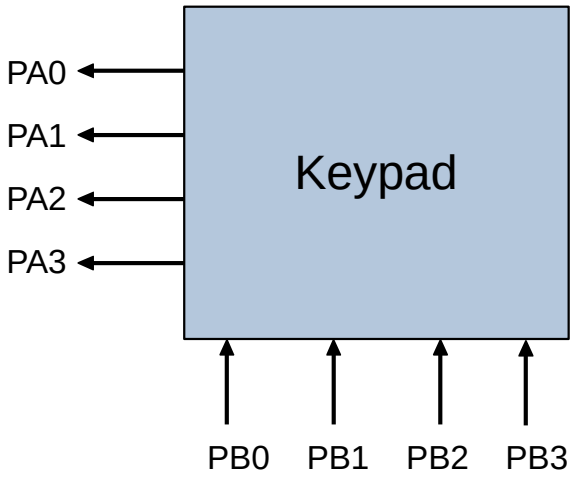
Keypad matrix (4x4)

0	1	2	3
4	5	6	7
8	9	A	B
C	D	E	F

Keypad Pin Connections to PortA and PortB

Columns are outputs to Keypad (Scan lines)
Port B is used as output :
PB0, PB1, PB2, PB3 pins are used, to select the columns.

Rows are inputs from Keypad (Return lines)
Port A is used as input :
PA0, PA1, PA2, PA3 pins are used, to read the rows.



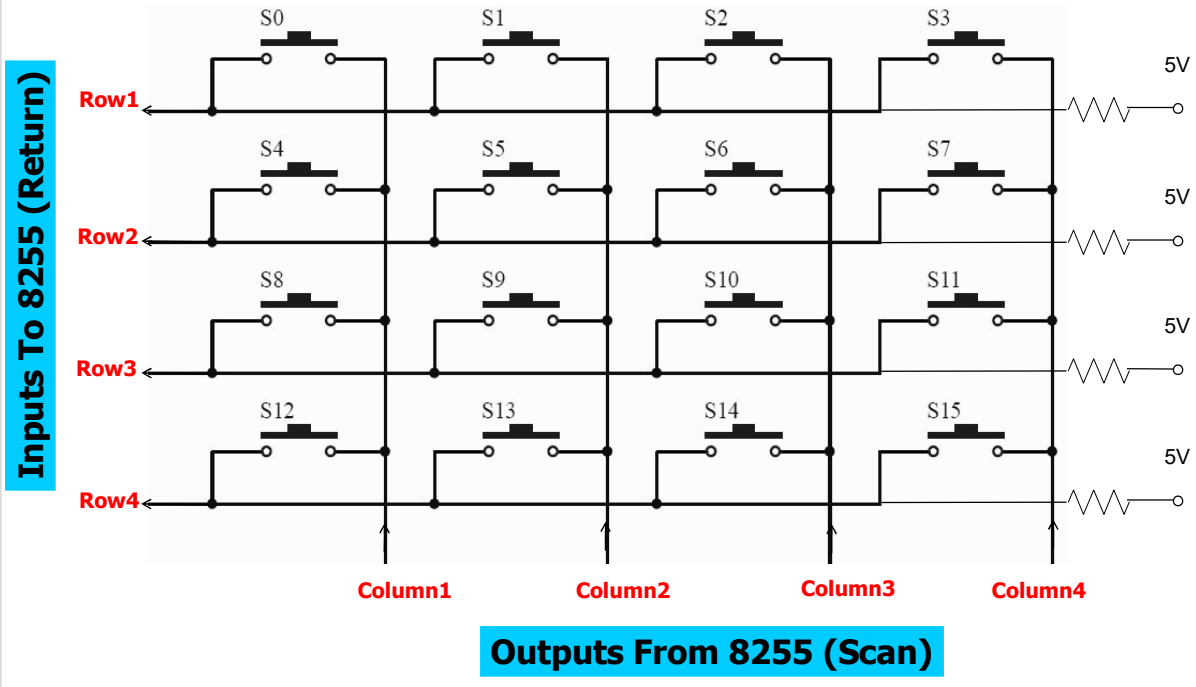
Selecting the Columns (Outputs) and Reading the Rows (Inputs)

- In order to use the port pins of the 8255 chip efficiently, only 8 pins of it will be used.
- The 4x4 button keypad normally requires 16 port lines.
- Since we want to use only 8 pins of 8255 chip for the keypad, looping and column selection in the program is necessary.
- Program will select the columns of keypad one-at-a-time, and also will read the rows one-at-a-time.

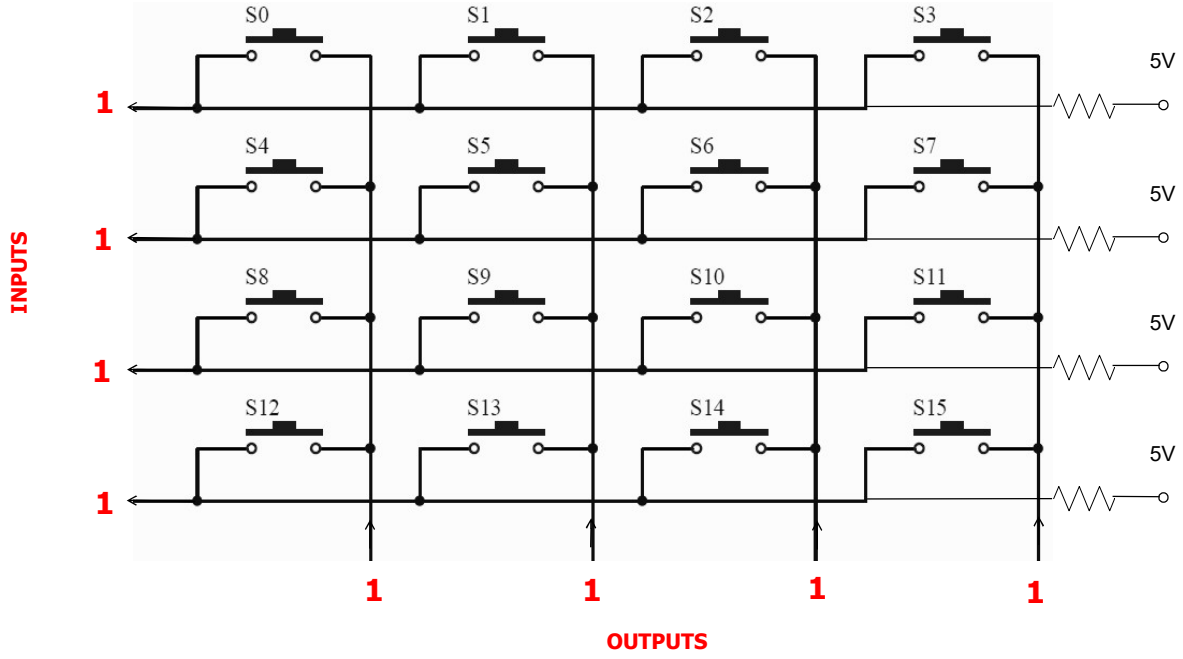
- **Port B (PB0-PB3)** lines of 8255 chip are used as **outputs (columns)**.
  - PB0-PB3 columns will be selected by program.
  - Key press is detected by rotating 0 among columns.

- **Port A (PA0-PA3)** lines are used as **inputs (rows)**.
  - If no key is pressed, then all rows (PA0-PA3) read high (1).
  - When a key is pressed, then one of the columns (PB0-PB3) is **shorted** with a row, and that row becomes low (0).

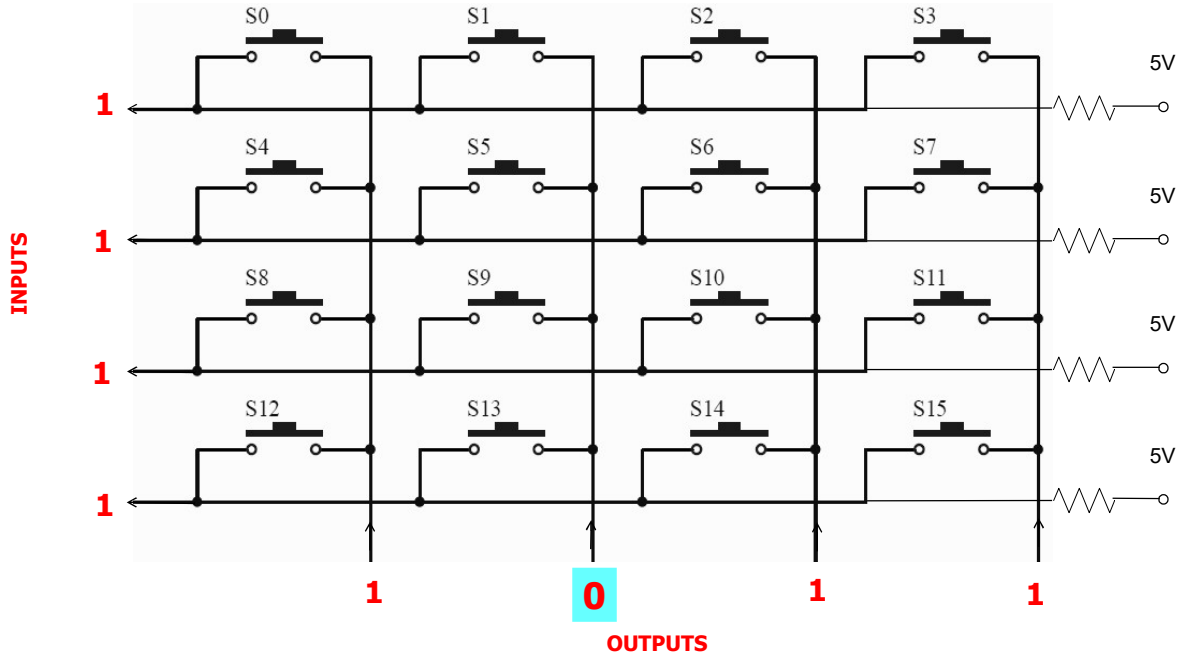
Connections of 4x4 Button Matrix



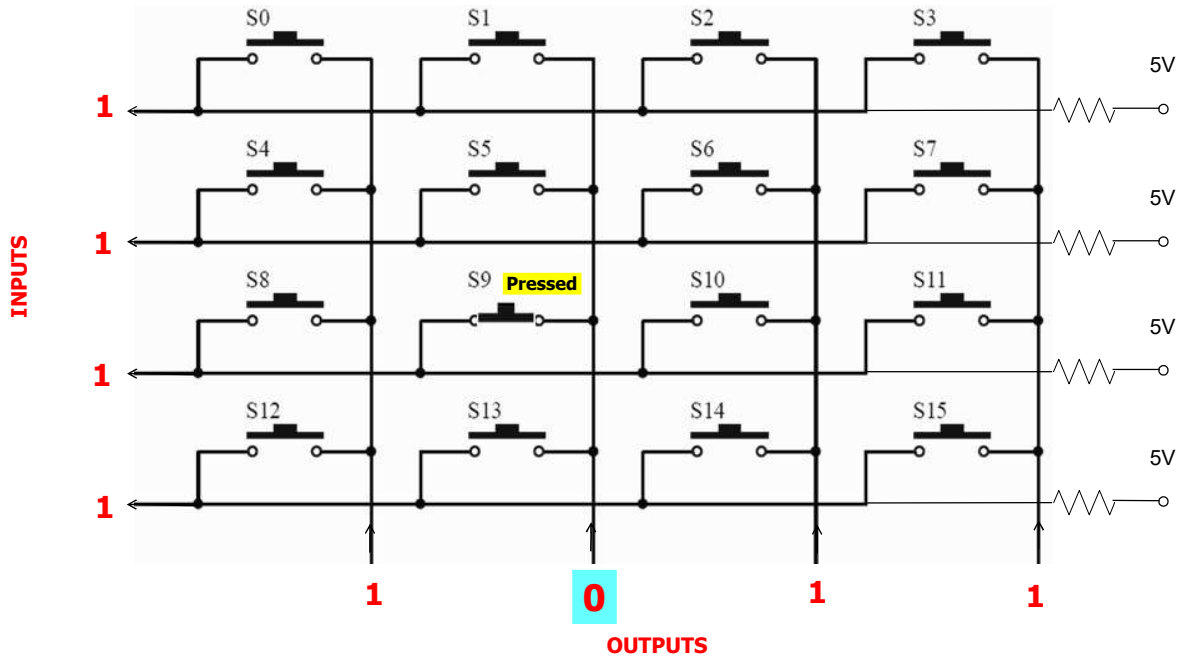
Initial status:
No buttons are pressed



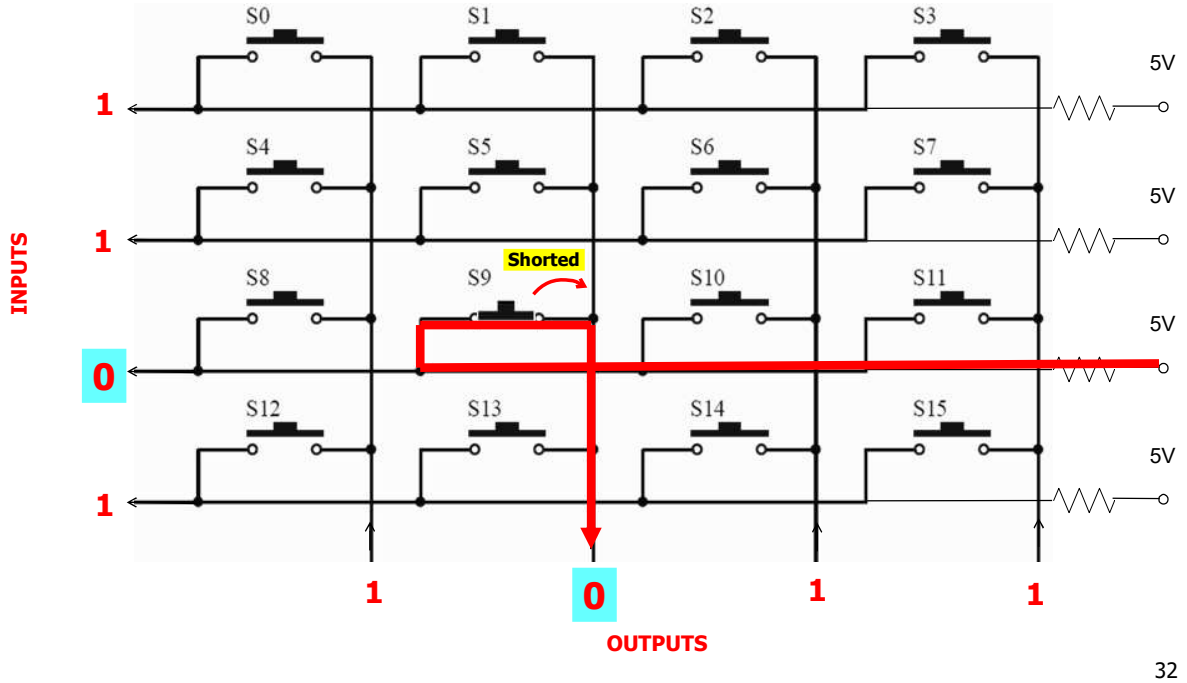
Column2 selected by program
(0 is written to Column2 output)



S9 button is pressed by user during Column2 selection



0 is read from Row3 input by program



Selecting the Columns and Reading the Rows of Keypad Matrix

- The way to read a button matrix is to **select (drive)** one column output at a time to the active low-level, and **read (scan)** all the input rows, looking to see if any row is at the active low-level.
- Then program goes to the next column output, and so on (endless looping).
- If a row input is at active low-level, then programs has detected a key press. In that case, program calculates the key value with a formula.
- The formula below is used to calculate the Key value pressed (between 0 and 15).

Key value = (Row - 1) * (Number of Columns) + Column - 1

- Both the Row and the Column variables are counted starting from 1.

Example : Calculating the Key Value by using row and column values

Suppose user pressed the B symbol on keypad.
Calculate the Key value for the B symbol.

0	1	2	3
4	5	6	7
8	9	A	B
C	D	E	F

Formula for Key value:

Key value = (Row-1)*(Number of Columns) + Column - 1

Solution:

Row = 3 Column = 4

Key value = (3-1)*4 + 4 – 1 = **11**

(The equivalent hex symbol on keypad is B.)

Selecting Columns of Keypad Matrix thru Port B of 8255 Chip

- Port B (PB0-PB3) 4 lines are connected to the columns of keypad.
- Port A (PA0-PA3) 4 lines are connected to the rows of keypad.
- Selecting a column requires writing **zero** to Port B pins (PB0-PB3 lines), which are connected to columns of keypad.
- To select a column, program writes "0" to a column one-at-a-time.

Inputs (Keypad rows) — Outputs (Keypad columns)

Row 4	Row 3	Row 2	Row 1	Column 4	Column 3	Column 2	Column 1
PA3	**PA2**	**PA1**	**PA0**	**PB3**	**PB2**	**PB1**	**PB0**
1	1	1	1	1	1	1	0
1	1	1	1	1	1	0	1
1	1	1	1	1	0	1	1
1	1	1	1	0	1	1	1

Reading Rows of Keypad Matrix thru Port A of 8255 Chip

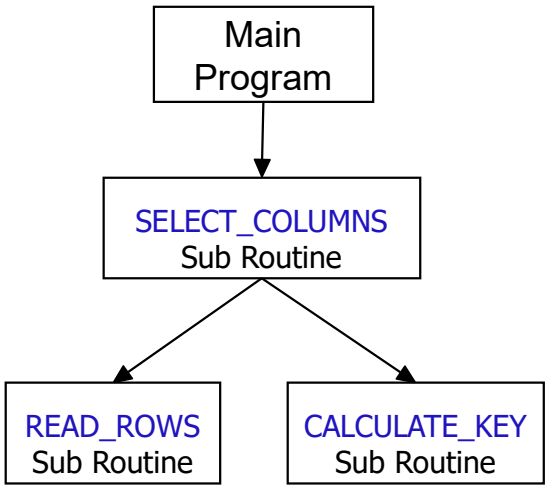
- Program reads rows and checks each of them one-at-a-time (checks whether which row is "0" or not).
- "0" in a row indicates that a key in that row is pressed by user.

Inputs (Keypad rows) — Outputs (Keypad columns)

Row 4	Row 3	Row 2	Row 1	Column 4	Column 3	Column 2	Column 1
PA3	**PA2**	**PA1**	**PA0**	**PB3**	**PB2**	**PB1**	**PB0**
1	1	1	0	0	0	0	0
1	1	0	1	0	0	0	0
1	0	1	1	0	0	0	0
0	1	1	1	0	0	0	0

Program Structure

- Program continuously reads the keypad matrix.
- Program detects any key press and stores its value (between 0 and 15) to a memory variable named KEY.
- Displaying the pressed key value on screen is not implemented in the program.



Program

Part1

```
; Program for reading 4x4 Matrix Keypad.
; Using the 8255 interface chip.

;---------------------------------
; MAIN PROGRAM
;---------------------------------
; Define the port addresses of 8255 chip.
PortA     EQU   0C0h
PortB     EQU   0C2h
Kontrol   EQU   0C6h

.model small

.stack     ; Default stack size is 1 KB.
; The CALL instruction uses the stack.

.data
ROW          DB  ?       ; Row index of pressed key
COLUMN    DB  ?       ; Column index of pressed key
KEY          DB  ?       ; Stores the calculated key number pressed
```

Part2

```
.code
BASLA PROC                  ; Main procedure

.STARTUP
mov AL, 10010000b           ; The control word
out  Kontrol, AL            ; Initialize 8255 chip

MainDongu:                  ; Endless main loop
Call  SELECT_COLUMNS        ; Call the subroutine
JMP MainDongu               ; Goto to main loop

BASLA ENDP                  ; End of main procedure
```


Part3

```
; SUB ROUTINES
;---------------------------------
; Subroutine selects (drives, scans) the columns one-by-one.
; Makes a column output bit zero, one at a time.
; Calls the other subroutines.
SELECT_COLUMNS  PROC
Kolon1:
      MOV AL, 1111111 0b   ; Select Column1 (rightmost bit)
      OUT PortB, AL        ; Drive column (Write to Port B)
      MOV COLUMN, 1        ; Assign current column
      call READ_ROWS       ; Call subroutine
      MOV AL, ROW          ; Get value of row variable
      CMP AL, 0            ; Check if any key pressed
      JE Kolon2            ; If zero, then no key pressed
      Call CALCULATE_KEY   ; Call subroutine
      RET                  ; Return
Kolon2:
      MOV AL, 111111 01b   ; Select Column2
      OUT PortB, AL
      MOV COLUMN, 2
      call READ_ROWS
      MOV AL, ROW
      CMP AL, 0
      JE Kolon3
      Call CALCULATE_KEY
      RET                  ; Return
```

Part4

```
Kolon3:
        MOV AL, 11111011b  ; Select Column3
        OUT PortB, AL
        MOV COLUMN, 3
        call READ_ROWS
        MOV AL, ROW
        CMP AL, 0
        JE Kolon4
        Call CALCULATE_KEY
        RET                         ; Return
Kolon4:
        MOV AL, 11110111b  ; Select Column4
        OUT PortB, AL
        MOV COLUMN, 4
        call READ_ROWS
        MOV AL, ROW
        CMP AL, 0
        JE Donus1
        Call CALCULATE_KEY

Donus1:
        RET                         ; Return
SELECT_COLUMNS  ENDP
```


Part5

```
; Subroutine reads (scans) the rows one-by-one.
; Determines which row is current if any key pressed.
READ_ROWS  PROC
        IN AL, PortA            ; Read from port A
        AND  AL, 11110000b  ; Filter left 4 bits
Satir1:
        CMP AL, 11111110b   ; Check Row1 bit
        JNE Satir2              ; If not equal, then check next row
        MOV ROW, 1              ; Row detected, save it to variable
        RET                     ; Return
Satir2:
        CMP AL, 11111101b   ; Check Row2 bit
        JNE Satir3
        MOV ROW, 2
        RET                     ; Return
Satir3:
        CMP AL, 11111011b   ; Check Row3 bit
        JNE Satir4
        MOV ROW, 3
        RET                     ; Return
Satir4:
        CMP AL, 11110111b   ; Check Row4 bit
        JNE Donus2
        MOV ROW, 4
Donus2:
        RET                     ; Return
READ_ROWS  ENDP
```

Part6

```
; Subroutine calculates the KEY value with
; the formula by using ROW and COLUMN.

CALCULATE_KEY   PROC
      MOV AX, 0             ; Clear accumulator
      MOV AL, ROW           ; Get row value
      DEC AL                ; Decrement row value by 1
      MOV BL, 4             ; Number of columns
      MUL BL                ; AL*BL executed, result is in AX
      ADD AL, COLUMN        ; Add column value to result
      MOV KEY, BL           ; Store calculated key to variable
      RET
CALCULATE_KEY   ENDP

END BASLA         ; End of file
```