I/O Peripherals - 8086 Microprocessor Project

This section documents the I/O peripheral interfaces used in the 8086-based microcomputer system. The peripherals include:

- 8255 Programmable Peripheral Interface for the printer interface (parallel communication).

- 8251 Universal Synchronous/Asynchronous Receiver Transmitter (USART) for serial communication.

- 8279 Programmable Keyboard Interface for a 64-key matrix keyboard.

- 8279 Display Interface for a 16-digit seven-segment display.

1. 8255 - Parallel Interface

Purpose

The Intel 8255 is a Programmable Peripheral Interface used to connect parallel I/O devices. It is used to interface with the printer.

Ports

- Port A: Used to send data to the printer.

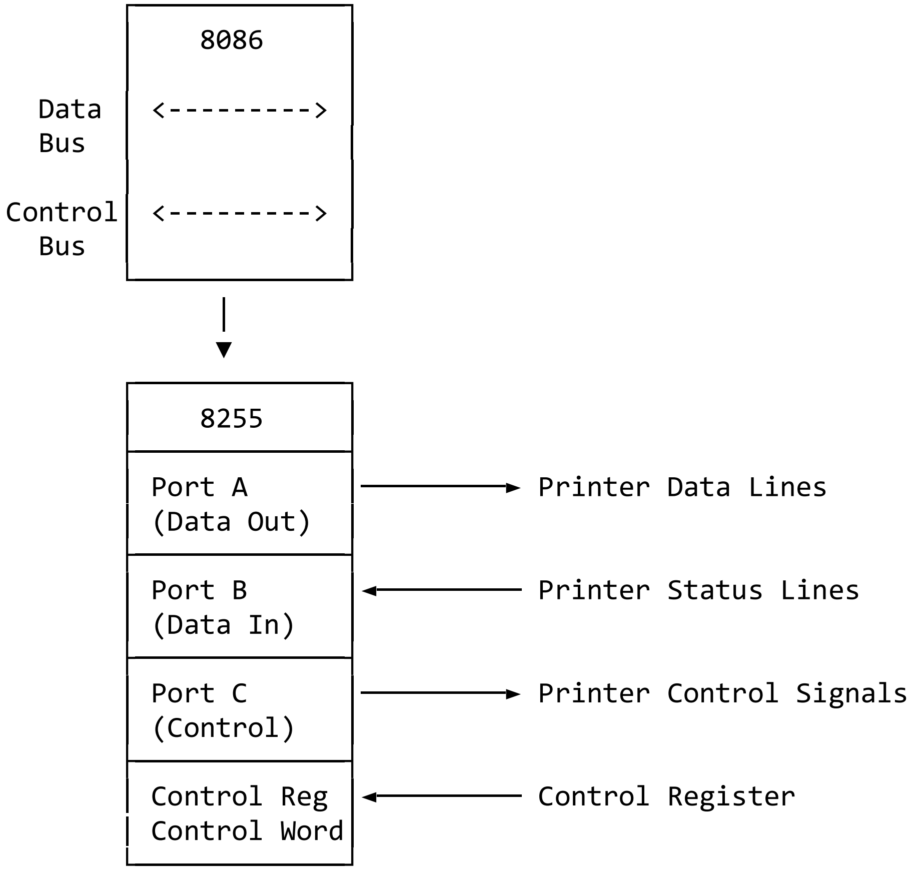
- Port B: Optional use for control signals or status.

- Port C: Used to send control signals.

Address Mapping

Register	Address
Port A	0xC0800
Port B	0xC0801
Port C	0xC0802
Control	0xC0803

Block Diagram



Code

```
; Initialize 8255
MOV AL, 10000000b ; Control Word: Port A & C output
OUT C0803h, AL ; Write control word to 8255 control port
; Send example data byte to printer via Port A
MOV AL, 0A5h ; Example data
OUT C0800h, AL ; Output data to Port A
; Generate strobe control signal on Port C
MOV AL, 01h ; Strobe bit set
OUT C0802h, AL ; Output control signals via Port C
```

2. 8251 - Serial Interface (Serial Port)

Purpose

The Intel 8251 is a USART for serial communication. It is used to send and receive data serially through a serial connector.

Components:

- *Transmit Buffer (TX Buffer):*

Temporarily holds data bytes that the CPU wants to send out serially.

- *Transmit Control:*

Manages the timing and framing of outgoing serial data, including start, stop bits, parity, and baud rate control.

- *Receive Buffer (RX Buffer):*

Holds incoming serial data received from an external device, buffering it for the CPU to read.

- *Receive Control:*

Handles synchronization, error detection, and framing of incoming serial data.

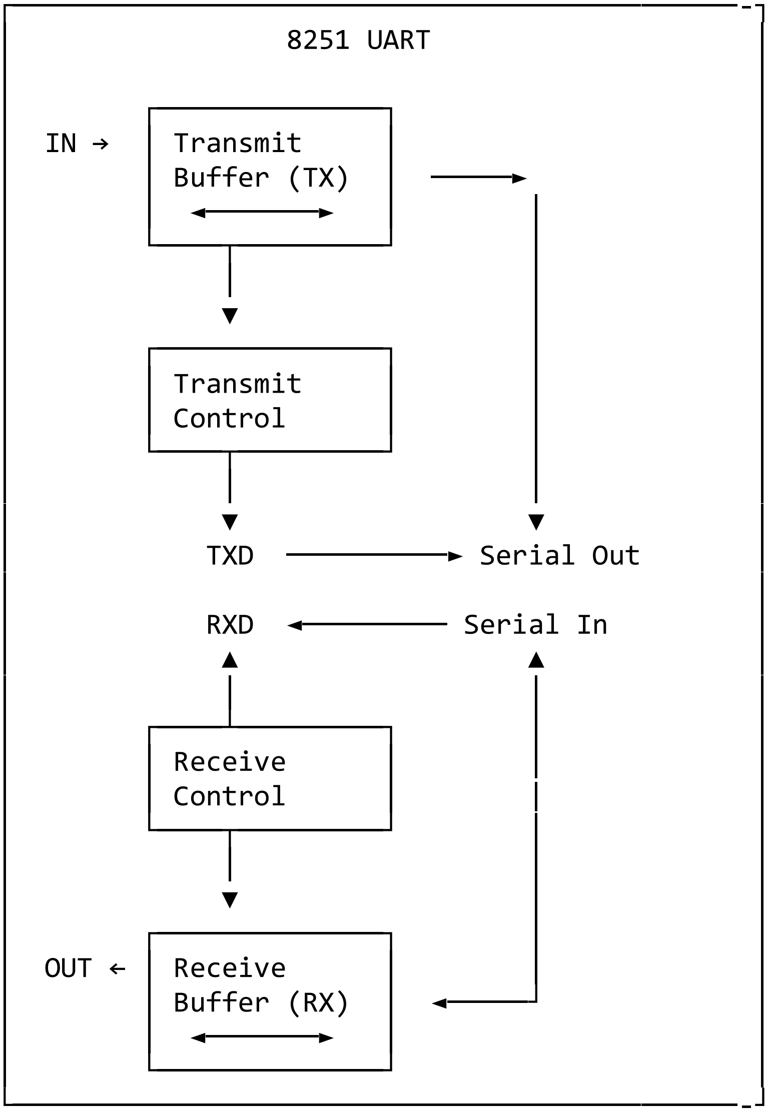
- *Baud Rate Generator (may be external or internal):*

Sets the speed (bits per second) of serial data transmission and reception.

Address Mapping:

Register	Address
Mode	0xC0200
Command	0xC0200
Transmit	0xC0201
Receive	0xC0201

Block Diagram



Code

```
; Initialize 8251
MOV AL, 01011110b ; Mode: async, 8-bit, 1 stop bit
OUT C0200h, AL ; Mode register
MOV AL, 00001011b ; Command: enable TX and RX
OUT C0200h, AL ; Command register
; Transmit example data byte ‘A’
MOV AL, 41h ; Data byte ACII ‘A’
```

```
OUT C0201h, AL ; Data register
IN AL, C0201h ; Read data from port
```

3. 8279 - Keyboard Interface

Purpose

The Intel 8279 is used to interface a 64-key matrix keyboard.

Components:

- *64-Key Matrix Scanning:*

The 8279 scans an 8-row by 8-column keyboard matrix, allowing it to detect key presses efficiently without needing one input line per key.

- *Debouncing:*

The controller automatically filters out mechanical key bounce, ensuring that each key press is registered cleanly as a single event.

- *FIFO Buffer:*

Key codes detected by the 8279 are stored in a First-In-First-Out buffer, allowing the CPU to read key presses asynchronously without losing data.

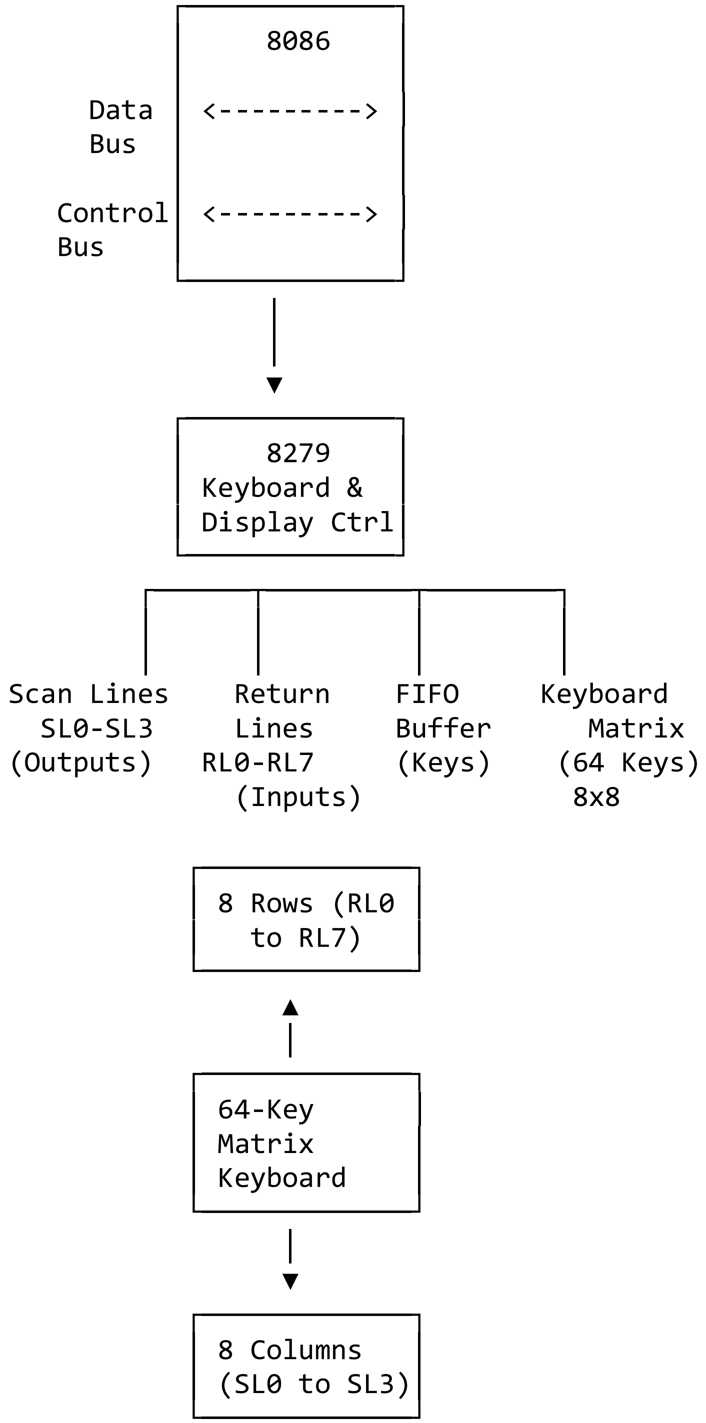
- *Interrupt Generation:*

The 8279 can generate an interrupt signal to the CPU when a key is pressed, enabling responsive, interrupt-driven input handling.

Address Mapping:

Register	Address
Command/Mode	0xC0400
Data	0xC0401

Block Diagram



Code

```
; Keyboard initialization
MOV AL, 00001111b ; Keyboard/display mode setup
OUT C0400h, AL ; Command port
MOV AL, 00000001b ; Enable auto-increment addressing
OUT C0400h, AL ; Command port
; Read key code from FIFO
IN AL, C0401h ; Keyboard data port
```

4. 8279 - Display Interface

Purpose

The Intel 8279 also controls the 16-digit 7-segment display.

Components:

- *16-Digit Multiplexed Display:*

Supports up to 16 digits, each with 7 segments (plus optional decimal point), using multiplexing.

- *Automatic Multiplexing:*

The controller automatically cycles through each digit rapidly, lighting one digit at a time in sequence.

- *Display RAM:*

Holds the digit codes for each of the 16 digits. The CPU writes the desired segment patterns into this RAM, which the 8279 uses to drive the display.

- *Segment Decoder:*

Converts the stored digit data into signals that turn on/off individual segments (a through g) on the 7-segment LEDs.

- *Digit Drivers:*

Enable the activation of each digit’s common line, controlling which digit is currently lit during multiplexing.

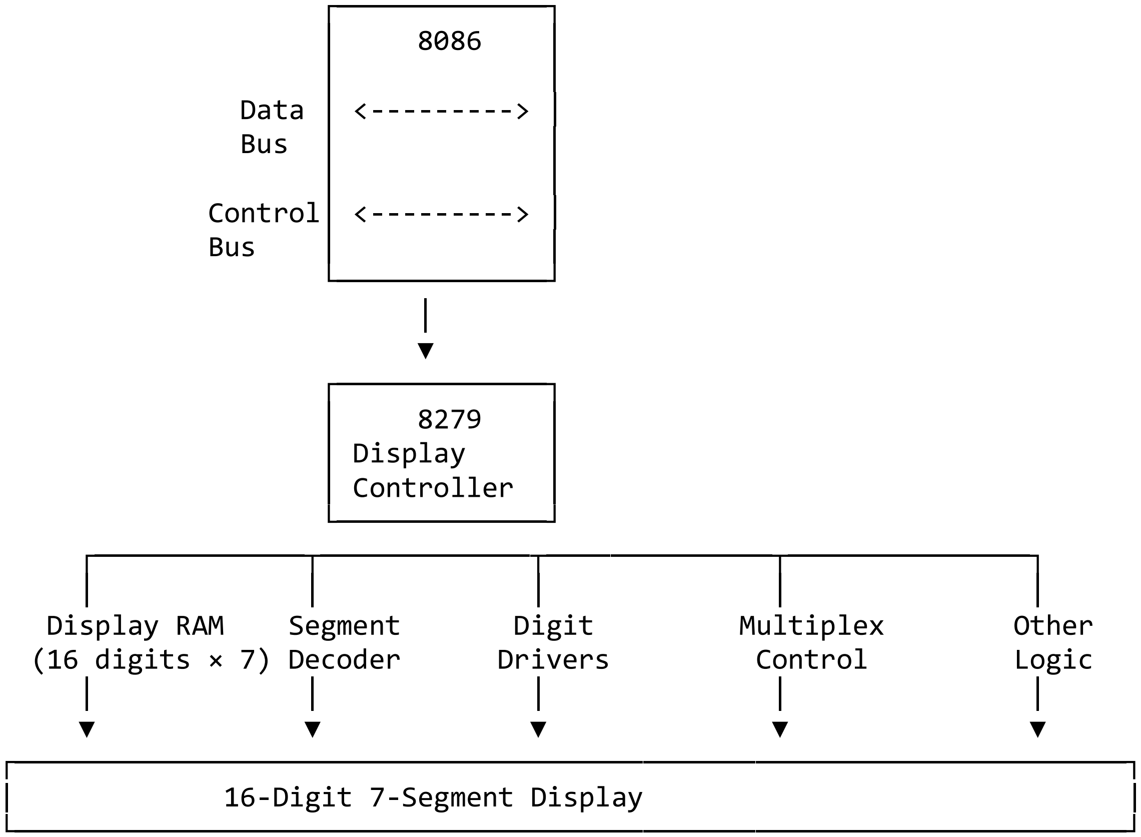
- *Brightness Control:*

Allows adjustment of display brightness by controlling the duty cycle of the multiplexing signal.

Address Mapping:

Register	Address
Command/mode	0xC0500
Data	0xC0501

Block Diagram



Code

```
; Display initialization
MOV AL, 00001111b ; Display mode setup
OUT C0500h, AL ; Display command port
MOV AL, 00000001b ; Enable auto-increment addressing
OUT C0500h, AL ; Display command port
; Output digit data to display RAM
```

```
MOV AL, 3Fh ; Digit ‘0’ 7-segment pattern

OUT C0501h, AL ; Display RAM data port
```

Conclusion

The IO Peripherals document provides the integration of key I/O peripherals essential to the functionality of the 8086 microcomputer system. Each peripheral is mapped into the memory space with proper address decoding and provides the necessary parallel, serial, keyboard, and display interfaces.