Basic I/O Interface

Dr. Mohammed Morsy
Basic I/O Interface

• This chapter will discuss:
  o I/O instructions.
  o Handshaking.
  o I/O Decoding.
  o 82C55 programmable parallel interface.
  o 8254 programmable interval timer.
Basic I/O Instructions

- IN and OUT transfer data between an I/O device and the microprocessor's accumulator (AL, AX or EAX).

- The I/O address is stored in:
  - Register DX as a 16-bit I/O address (variable addressing).
  - The byte, p8, immediately following the opcode (fixed address).

- Only 16-bits (A0 to A15) are decoded.
  - Address connections above A15 are undefined for I/O instructions.
  - INS and OUTS transfer to I/O devices using ES:DI and DS:SI, respectively.

```
IN AL, 19H ;8-bits are saved to AL from I/O port 19H.
IN EAX, DX ;32-bits are saved to EAX.
OUT DX, EAX ;32-bits are written to port DX from EAX.
OUT 19H, AX ;16-bits are written to I/O port 0019H.
```
Isolated and Memory-Mapped I/O

- In the Isolated scheme, IN, OUT, INS and OUTS are required.
- In the Memory-mapped scheme, any instruction that references memory can be used.

Disadvantage:
- A portion of the memory space is used for I/O devices.

Advantage:
- IORC and IOWC not required.
- Any data transfer instruction.

Disadvantage:
- Hardware using M/IO and W/R needed to develop signals IORC and IOWC.
- Requires IN, OUT, INS and OUTS.
Basic I/O Interface

- The basic input device (to the microprocessor) is a set of tri-state buffers.

8-bit input port

Toggle switches are data source.

When tri-states are enabled, microprocessor can read state of toggle switches into AL (using IN instruction).

I/O port address decoded to SEL

74ALS244

1A_1, 1A_2, 1A_3, 1A_4, 2A_1, 2A_2, 2A_3, 2A_4

1Y_1, 1Y_2, 1Y_3, 1Y_4, 2Y_1, 2Y_2, 2Y_3, 2Y_4

Data Bus

VCC

10KΩ
Basic I/O Interface

- The basic output device (from the microprocessor) is a set of latches.
Handshaking

- I/O devices are typically slower than the microprocessor.
- Handshaking is used to synchronize I/O with the microprocessor.
  - A device indicates that it is ready for a command or data (through some I/O pin or port).
  - The processor issues a command to the device, and the device indicates it is busy (not ready).
  - The I/O device finishes its task and indicates a ready condition, and the cycle continues.
- There are two basic mechanisms for the processor to service a device.
  - Polling (Processor initiated): Device indicates it is ready by setting some status bit and the processor periodically checks it.
  - Interrupts (Device initiated): The act of setting a status bit causes an interrupt, and the processor calls an ISR to serve the device.
I/O Port Decoding

• For memory-mapped I/O, decoding is identical to memory decoding.
• For isolated I/O, $\overline{TORC}$ and $\overline{IOWC}$ are developed using $M/\overline{TO}$ and $W/\overline{R}$ pins of the microprocessor.
• The 8-bit and 16-bit I/O port address decoding are similar to the memory decoding.
• The I/O banks on the 8086 through the 80386SX are also set up like the memory with bank enable $\overline{BEx}$ signals.
8086 I/O Port Decoding Example

• Write the PAL equations for this I/O interface?

Note that only an 8-bit I/O port address is decoded.
Programmable Peripheral Interface (82C55)

- The 82C55 is a popular interfacing component, that can interface any TTL-compatible I/O device to the microprocessor.
- It is used to interface the keyboard and parallel printer port in PCs (usually as part of an integrated chipset).
- Requires insertion of wait states if used with a microprocessor using higher than an 8 MHz clock.
82C55 PPI

- PPI has 24 pins for I/O that are programmable in groups of 12 pins.
- It has three distinct modes of operation.
  - Mode 0: is the basic input/output mode.
  - Mode 1: is the strobed operations
  - Mode 2: is a bidirectional mode of operation
- In the PC, an 82C55 or its equivalent is decoded at I/O ports 60H-63H.
82C55 PPI Pinout

Group A
Port A (PA7-PA0) and upper half of port C (PC7 - PC4)

Group B
Port B (PB7-PB0) and lower half of port C (PC3 - PC0)

I/O Port Assignments

<table>
<thead>
<tr>
<th>$A_1$</th>
<th>$A_0$</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Port A</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Port B</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Port C</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Command Register</td>
</tr>
</tbody>
</table>
Interfacing the 82C55 PPI
Programming the 82C55

Command Byte A (Programs ports A, B, C)

Group A
- Port C (PC7 - PC4)
  - 1 = input
  - 0 = output
- Port A
  - 1 = input
  - 0 = output
- Mode
  - 00 = mode 0
  - 01 = mode 1
  - 1x = mode 2

Group B
- Port C (PC3 - PC0)
  - 1 = input
  - 0 = output
- Port B
  - 1 = input
  - 0 = output
- Mode
  - 0 = mode 0
  - 1 = mode 1

Command Byte B (Sets or resets any bits in port C)

- Bit set/reset
  - 1 = set
  - 0 = reset
- Selects a bit
82C55: Mode 0 Operation - Seven Segment LED Display Example

8-Digit Seven Segment LED Display Interface
**82C55: Mode 0 Operation - Seven Segment LED Display**

- Mode 0 operation causes the 82C55 to function as a buffered input device or as a latched output device.
- In previous example, both ports A and B are programmed as (mode 0) simple latched output ports.
- Port A provides the segment data inputs to display and port B provides a means of selecting one display position at a time.
- Different values are displayed in each digit via fast time multiplexing.
- The values for the resistors and the type of transistors used are determined using the current requirements.
Seven Segment LED Display Software

;Software to initialize the 8255
  MOV AL, 1000 0000B
  OUT COMMAND,AL

;Procedure to display the data
DISPLAY PROC NEAR
  MOV BX,8
  MOV AH,7FH
  MOV SI, OFFSET DISPLAY_RAM1
  Display1:  MOV AL, AH
             OUT PORTB,AL
             MOV AL, [BX+SI]        
             OUT PORTA, AL
             CALL DELAY
             RAR AH
             DEC BX
             JNZ DISPLAY1
  RET
DISPLAY ENDP
82C55: Mode 1 Strobed Input

- Port A and/or port B function as latching input devices.
- External data is stored in the ports until the microprocessor is ready.
- Port C used for control or handshaking signals (cannot be used for data).
82C55: Mode 1 Strobed Input

Signal definitions for Mode 1 Strobed Input

- **STB**: The strobe input loads data into the port latch.
- **IBF**: Input buffer full is an output indicating that the input latch contains information.
- **INTR**: Interrupt request is an output that requests an interrupt.
- **INTE**: The interrupt enable signal is neither an input nor an output; it is an internal bit programmed via the PC4 (port A) or PC2 (port B) bits.
- **PC7, PC6**: The port C pins 7 and 6 are general-purpose I/O pins that are available for any purpose.
Mode 1 Strobed Input Timing
Mode 1 Strobed Input

- Mode 1 Strobed Input Example
- Keyboard encoder debounces the key-switches, and provides a strobe whenever a key is depressed.
- DAV is activated on a key press strobing the ASCII-coded key code into Port A.
Mode 1 Strobed Input

• Procedure that read an ASCII character from the keyboard via port A.

BIT5 EQU 20H
READ_KEY PROC NEAR
    IN AL, PORTC
    TEST AL, BIT5
    JZ READ_KEY
    IN AL, PORTA
    RET
READ_KEY ENDP
Mode 1 Strobed Output

- Similar to Mode 0 output operation, except that handshaking signals are provided using port C.

- Signal Definitions for Mode 1 Strobed Output
  - $\text{OBF}$: Output buffer full is an output that goes low when data is latched in either port A or port B. Goes high on ACK.
  - $\text{ACK}$: The acknowledge signal causes the OBF pin to return to 1. This is a response from an external device.
  - INTR: Interrupt request is an output that requests an interrupt.
  - INTE: The interrupt enable signal is neither an input nor an output; it is an internal bit programmed via the PC6 (port A) or PC2 (port B) bits.
  - PC5, PC4: The port C pins 5 and 4 are general-purpose I/O pins that are available for any purpose.
Mode 1 Strobed Output Timing
Mode 1 Strobed Output

- Mode 1 Strobed Output Example:
Mode 1 Strobed Output

- Procedure that write an ASCII character to the printer.

```assembly
BIT1 EQU 2
PRINT PROC NEAR
;Check printer ready
IN AL, PORTC
TEST AL, BIT1
JZ PRINT

;Send character
MOV AL, AH
OUT PORTB, AL

;Send DS
MOV AL, 0000 1000B
OUT COMMAND, AL
MOV AL, 0000 1001B
OUT COMMAND, AL
RET

PRINT ENDP
```
Mode 2 Bi-directional Operation

- Only allowed with port A.
- Bi-directional bused data used for interfacing two computers.
- Timing diagram is a combination of the Mode 1 Strobed Input and Mode 1 Strobed Output Timing diagrams.
Mode 2 Bi-directional Operation

- **INTR**: Interrupt request is an output that requests an interrupt.
- **OBF**: Output buffer full is an output indicating that the output buffer contains data for the bi-directional bus.
- **IFB**: Input buffer full is an output indicating that the input latch contains information for the external bi-directional bus.
- **ACK**: Acknowledge is an input that enables tri-state buffers which are otherwise in their high-impedance state.
- **STB**: The strobe input loads data into the port A latch.
- **INTE**: Interrupt enable are internal bits that enable the INTR pin. Bit PC6(INTE1) and PC4(INTE2)
- **PC2, PC1, and PC0**: These pins of port C are general-purpose I/O pins that are available for any purpose.
Programmable Interval Timer: 8254

- Three independent 16-bit programmable counters (timers).
- Each capable of counting in binary or BCD with a maximum frequency of 10MHz.
- Used for controlling real-time events such as real-time clock, event counter, and motor speed and direction control.
- Usually decoded at port address 40H-43H and has the following functions:
  - Generates a basic timer interrupt that occurs at approximately 18.2Hz. Interrupts the micro at interrupt vector 8 for a clock tick.
  - Causes DRAM memory system to be refreshed. Programmed with 15us on the PC/XT.
  - Provides a timing source to the internal speaker and other devices.
8254 Functional Description
8254 Pin Definitions

- $A_1, A_0$: The address inputs select one of the four internal registers with the 8254 as follows:
- CLK: The clock input is the timing source for each of the internal counters.
- $\overline{CS}$: Chip Select enables the 8254 for programming, and reading and writing.
- G: The gate input controls the operation of the counter in some modes.
- OUT: A counter output is where the wave-form generated by the timer is available.
- $\overline{RD}/\overline{WR}$: Read/Write causes data to be read/written from the 8254 and often connects to the $\overline{I/ORC}/\overline{I/OWC}$.

<table>
<thead>
<tr>
<th>$A_1$</th>
<th>$A_0$</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Counter 0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Counter 1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Counter 2</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Control Word</td>
</tr>
</tbody>
</table>
8254 Programming

- Each counter may be programmed with a count of 1 to FFFFFFFH.
- Minimum count is 1 all modes except 2 and 3 with minimum count of 2.
- Each counter has a program control word used to select the way the counter operates.
- If two bytes are programmed, then the first byte (LSB) stops the count, and the second byte (MSB) starts the counter with the new count.
Each counter is individually programmed by writing a control word, followed by the initial count.

The control word allows the programmer to select the counter, model of operation, binary or BCD count and type of operation (read/write).
8254 Modes of Operation

There are 6 modes of operation for each counter:

- **Mode 0**: An events counter enabled with G. The output becomes a logic 0 when the control word is written and remains there until N plus the number of programmed counts.

![Diagram showing clock pulses and output transitions for Mode 0 operation.](image)
8254 Modes of Operation

• Mode 1: One-shot mode.
The G input triggers the counter to output a 0 pulse for a predetermined number of clock cycles. Counter reloaded if G is pulsed again.

Trigger with count of 5
8254 Modes of Operation

- Mode 2: Counter generates a series of pulses 1 clock pulse wide.

The separation between pulses is determined by the count.

The cycle is repeated until reprogrammed or G pin set to 0.

CLK

OUT

Count of 5 loaded
8254 Modes of Operation

- Mode 3: Generates a continuous square-wave with G set to 1.
If count is even, 50% duty cycle otherwise OUT is high 1 cycle longer.
8254 Modes of Operation

- Mode 4: Software triggered one-shot (G must be 1).
- Mode 5: Hardware triggered one-shot. G controls similar to Mode 1 (G is active-high).
Example: Generating a Waveform with the 8254

- Write an assembly program to generate a 100KHz square-wave using the following circuit. (A book example)
Reading a Counter

- Each counter has an internal latch read with the read counter port operation.
  - The latches will normally follow the count.
- If counter contents are needed, the latch can remember the count by programming the counter latch control word.
  - Counter contents are held in a latch until read.

```
  7 6 5 4 3 2 1 0
SC1 SC0 0 0 X X X X
```

Select counter:
- 00 = counter 0
- 01 = counter 1
- 10 = counter 2
- 11 = read-back command
Reading a Counter

• When a read from the latch or counter is programmed, the latch tracks the contents.
• When necessary for contents of more than one counter to be read at the same time, the read-back control word is used.
Reading a Counter

- With the read-back control word, the $CNT$ bit is logic 0 to cause the counters selected by CNT0, CNT1, and CNT2 to be latched.
- If the status register is to be latched, then the $ST$ bit is placed at logic 0.
- The status register shows:
  - the state of the output pin
  - whether the counter is at its null state (0)
  - how the counter is programmed
Programmable Communication Interface: 16550

- A universal asynchronous receiver/transmitter (UART).
- Operation speed: 0 - 1.5M Baud (Baud is # of bits transmitted/sec, including start, stop, data and parity).
- UART contains:
  - A programmable Baud rate generator.
  - Separate FIFO buffers for input and output data (16 bytes each).
Programmable Communications Interface: 16550

- Asynchronous serial data:
  Transmitted and received without a clock or timing signal.

Two 10-bit frames of asynchronous data.
7- or 8-bit ASCII, e.g. w or w/o parity, is possible.
Programmable Communications Interface: 16550

- Two separate sections are responsible for data communications:
  - Receiver
  - Transmitter

- Can function in:
  - simplex: transmit only
  - half-duplex: transmit and receive but not simultaneously
  - full-duplex: transmit and receive simultaneously
Pinout of the 16550

- $A_0$, $A_1$, and $A_2$: Select an internal register for programming and data transfer.

<table>
<thead>
<tr>
<th>$A_2$</th>
<th>$A_1$</th>
<th>$A_0$</th>
<th>Register</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Receiver buffer (read) and transmitter holding (write)</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>Interrupt enable</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>Interrupt identification (read) and FIFO control (write)</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>Line control</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>Modem control</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>Line status</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>Modem status</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>Scratch</td>
</tr>
</tbody>
</table>
Pinout of the 16550

- **ADS**: Address strobe used to latch the address and chip select. Not needed on Intel systems -- connected to ground.
- **BAUDOUT**: Clock signal from Baud rate generator in the transmitter.
- **CS₀, CS₁, CS₂**: Chip selects.
- **CTS**: Clear to send -- indicates that the modem or data set is ready to exchange information (Used in half-duplex to turn the line around).
- **D₇-D₀**: The data bus pins are connected to the microprocessor data bus.
- **DCD**: The data carrier detect -- used by the modem to signal the 16550 that a carrier is present.
Pinout of the 16550

- **DDIS**: Disable driver output -- set to 0 to indicate that the microprocessor is reading data from the UART. Used to change direction of data flow through a buffer.
- **DSR**: Data set ready is an input to 16550 -- indicates that the modem (data set) is ready to operate.
- **DTR**: Data terminal ready is an output -- indicates that the data terminal (16550) is ready to function.
- **INTR**: Interrupt request is an output to the micro -- used to request an interrupt. (Receiver error, Data received, Transmit buffer empty)
- **MR**: Master reset -- connect to system RESET
- **OUT₁, OUT₂**: User defined output pins for modem or other device.
- **RCLK**: Receiver clock -- clock input to the receiver section of the UART. Always 16X the desired receiver Baud rate.
Pinout of the 16550

- **RD, RD**: Read inputs (either can be used) -- cause data to be read from the register given by the address inputs.
- **RI**: Ring indicator input -- set to 0 by modem to indicate telephone is ringing.
- **RTS**: Request-to-send -- signal to modem, indicating UART wishes to send data.
- **SIN, SOUT**: Serial data pins, in and out.
- **RXRDY**: Receiver ready -- used to transfer received data via DMA techniques.
- **TXRDY**: Transmitter ready -- used to transfer transmitter data via DMA.
- **WR, WR**: Write (either can be used) -- connects to micro write signal to transfer commands and data to 16550.
- **XIN, XOUT**: Main clock connections -- a crystal oscillator can be used.
Programming the 16550

- Two phases: Initialization, operation.
- Initialization:
  - After RESET, the line control register and baud rate generator need to be programmed.
  - Line control register sets the # of data bits, # of stop bits and the parity. Addressed at location 011.

Stop bits: $S = 1$, 1.5 stop bits used for 5 data bits, 2 used for 6, 7 or 8.
Programming the 16550

• Initialization (cont.):
  o ST, P and PE used to send even or odd parity, to send no parity or to send a 1 or a 0 in the parity bit position for all data.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>No parity</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>Odd parity</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>No parity</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>Even parity</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>Undefined</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>Send/receive 1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>Undefined</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>Send/receive 0</td>
</tr>
</tbody>
</table>

  o SB = 1 causes a break to be transmitted on SOUT. A break is at least two frame of 0 data.
  o DL = 1 enables programming of the baud rate divisor.
Programming the 16550

• Initialization (cont.):
  o Baud rate generator is programmed with a divisor that sets baud rate of the transmitter.
  o Baud rate generator is programmed at 000 and 001. Port 000 used to hold least significant byte, 001 most significant.
  o Value used depends on external clock/crystal frequency.
    For 18.432MHz crystal, 10,473 gives 110 baud rate, 30 gives 38,400 baud.
  o Note, number programmed generates a clock 16X the desired Baud rate.
  o Last, the FIFO control register must be programmed at 010.
Programming the 16550

• Operation:
  o Status line register gives information about error conditions and state of the transmitter and receiver.
  o This register needs to be tested in software routines designed to use the 16550 to transmit/receive data.
  o For example, suppose a program wants to send data out SOUT.
  o It needs to poll the TH bit to determine if the transmitter is ready to receive data.
  o To receive information, the DR bit is tested.
Programming the 16550

• Operation (Cont.):
  • It is also a good idea to check for errors.
    o Parity error: Received data has a wrong bit-- transmission bit flip due to noise.
    o Framing error: Start and stop bits not in their proper places. This usually results if the receiver is receiving data at the incorrect baud rate.
    o Overrun error: Data has overrun the internal receiver FIFO buffer. Software is failing to read the data from the FIFO.
    o Break indicator bit: Software should check for this as well, i.e. two consecutive frames of 0s.
Example of 16550

- Write an assembly program to initialize, transmit, and receive data from the shown 16550 UART. (Book examples)