BITS Pilani

Microprocessors & Interfacing

IO Interface

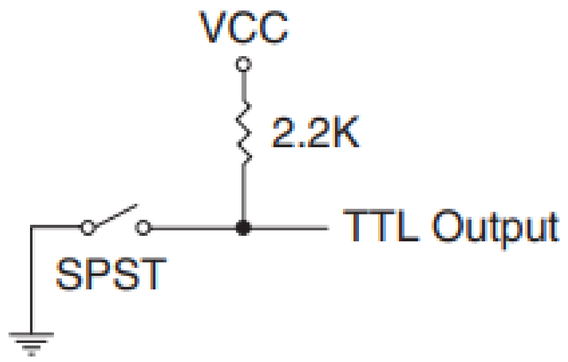
Dr. Gargi Prabhu
Department of CS & IS

Input Devices

- Input devices are already TTL and compatible, and can be connected to the microprocessor and its interfacing components.
  - or they are switch-based
- Switch-based devices are either open or connected; These are not TTL levels.
  - TTL levels are a logic 0 (0.0 V–0.8 V)
  - or a logic 1 (2.0 V–5.0 V)
- Using switch-based device as TTL-compatible input requires conditioning applied.

Input Devices

- A pull-up resistor ensures when the switch is open, the output signal is a logic 1.
  - when the switch is closed, it connects to ground, producing a valid logic 0 level
- A standard range of values for pull-up resistors is between 1K Ohm and 10K Ohm.



Input Devices

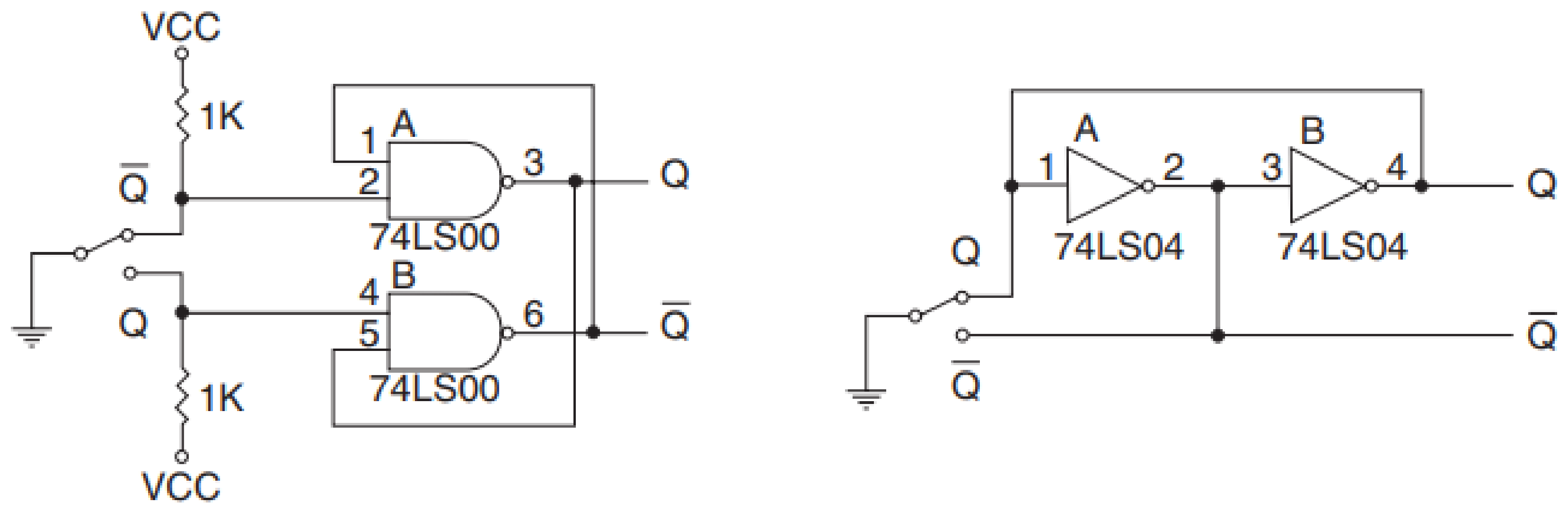
- Mechanical switch contacts physically bounce when they are closed,
  - which can create a problem if a switch is used as a clocking signal for a digital circuit
    - To prevent problems with bounces, following

circuits can be used

- The first version costs more to construct
  - the second costs requires no pull-up resistors

    and two inverters instead of two NAND gates

Input Devices



Debouncing switch contacts: (a) conventional debouncing and (b) practical debouncing.

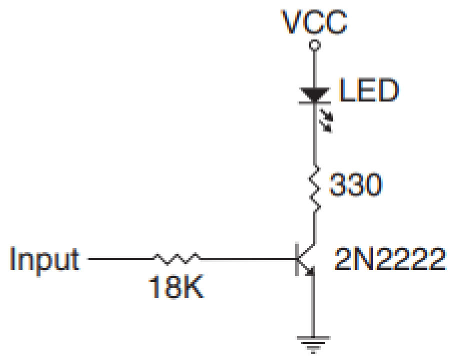
Output Devices

- Output devices are more diverse than input devices, but many are interfaced in a uniform manner.
- Before an output device can be interfaced, we must understand voltages and currents from the microprocessor or TTL interface.
- Voltages are TTL-compatible from the microprocessor of the interfacing element.
  - logic 0 = 0.0 V to 0.4 V
  - logic 1 = 2.4 V to 5.0 V

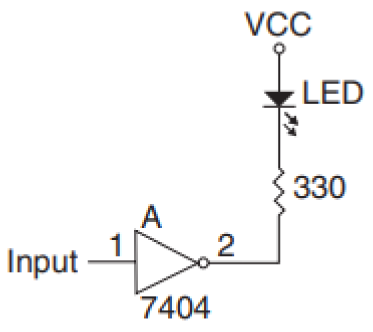
Output Devices

- Currents for a processor and many interfacing components are less than for standard TTL.
  - Logic 0 = 0.0 to 2.0 mA
  - logic 1 = 0.0 to 400 µA

Output Devices



(a)



(b)

- to interface a simple
  LED to a microprocessor peripheral pin.
  - a transistor driver is used in 11–8(a)
  - a TTL inverter is used in 11–8(b)
- The TTL inverter (standard version) provides up to 16 mA of current at a logic 0 level
  - more than enough to drive a standard LED

Output Devices

- TTL input signal has minimum value of 2.4 V
- Drop across emitter-base junction is 0.7 V.
- The difference is 1.7 V
  – the voltage drop across the resistor
- The value of the resistor is 1.7 V ÷ 0.1 mA or 17K W.
  – as 17K W is not a standard value, an 18K W resistor is chosen

Output Devices

- TTL input signal has minimum value of 2.4 V
- Drop across emitter-base junction is 0.7 V.
- The difference is 1.7 V
  – the voltage drop across the resistor
- The value of the resistor is 1.7 V ÷ 0.1 mA or 17K W.
  – as 17K W is not a standard value, an 18K W resistor is chosen

How I/O addresses are determined and assigned?

System Architecture Design: designed with a specific address space for I/O operations

Address Decoding: Address decoding logic maps specific address ranges to particular I/O devices.

Address Assignment: the system designer assigns specific I/O addresses to each peripheral device connected to the system. These assignments are typically documented in the system's hardware specifications or datasheets.

Configuration and Initialization: During system configuration and initialization, the microprocessor or system firmware may initialize the address decoding logic and configure the I/O addresses for each device.

How I/O addresses are determined and assigned?

Programming: In software development, programmers write device drivers or application software that interacts with the peripheral devices using the assigned I/O addresses. These software components use the addresses defined in the system hardware to access and control the devices.

Testing and Verification: Before deployment, the system undergoes testing and verification to ensure that the assigned I/O addresses function correctly and that the peripheral devices respond as expected to I/O operations.

I/O Port Address Decoding

- Very similar to memory address decoding, especially for memory-mapped I/O devices.
- The difference between memory decoding and isolated I/O decoding is the number of address pins connected to the decoder.
- In the personal computer system, we always decode all 16 bits of the I/O port address.

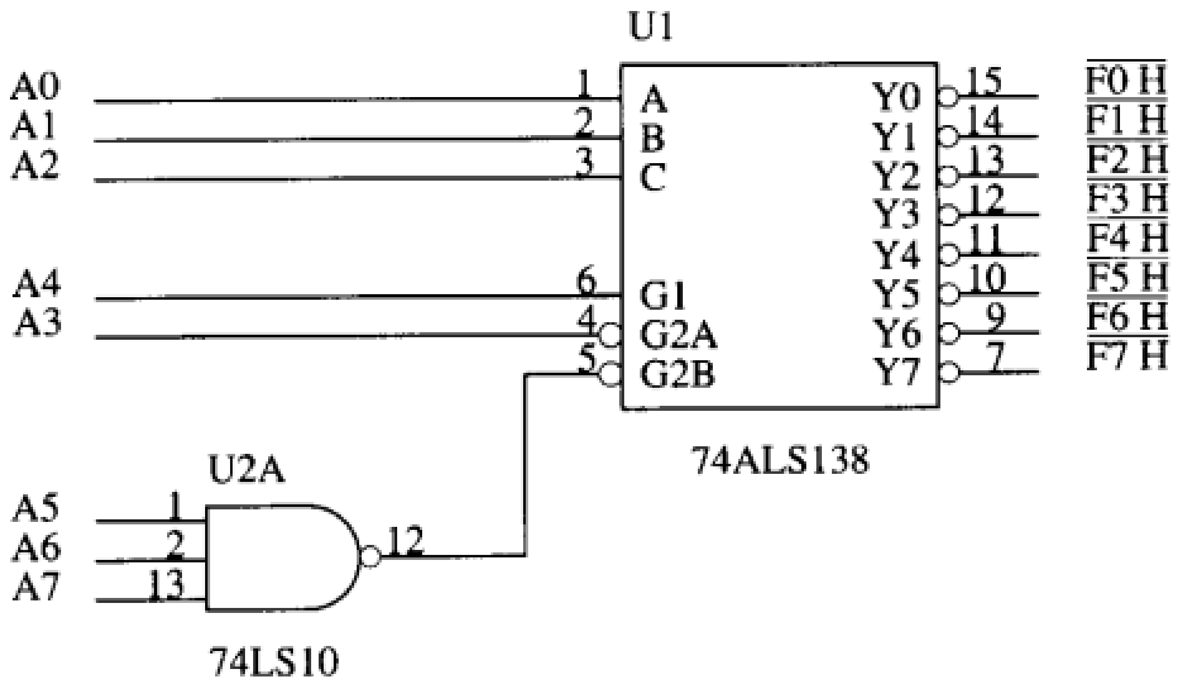
I/O Port Address Decoding

- Fixed I/O instruction uses an 8-bit I/O port address that on A_{15}–A_0 as 0000H–00FFH.
  - we often decode only address connections A_7–A_0 for an 8-bit I/O port address
- The DX register can also address I/O ports 00H–FFH.
- If the address is decoded as an 8-bit address, we can never include I/O devices using a 16- bit address.
  - the PC never uses or decodes an 8-bit address

I/O Port Address Decoding

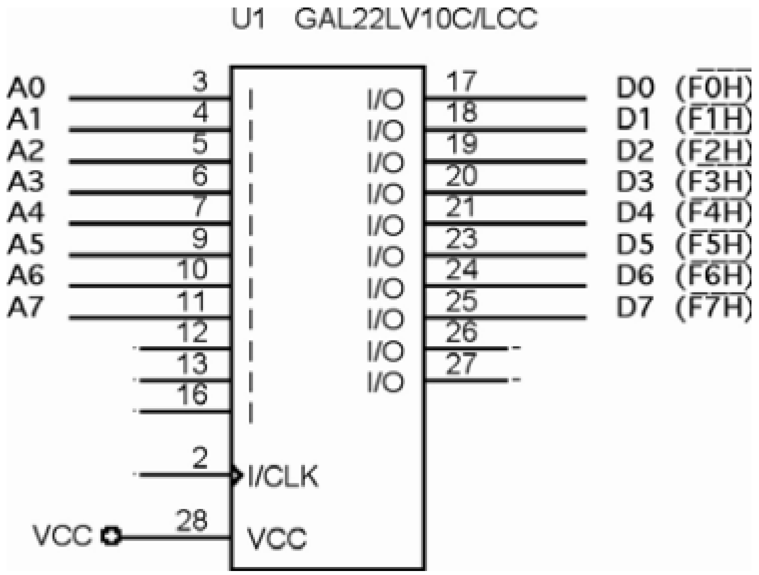
- Fixed I/O instruction uses an 8-bit I/O port address that on A_{15}–A_0 as 0000H–00FFH.
  - we often decode only address connections A_7–A_0 for an 8-bit I/O port address
- The DX register can also address I/O ports 00H–FFH.
- If the address is decoded as an 8-bit address, we can never include I/O devices using a 16- bit address.
  - the PC never uses or decodes an 8-bit address

I/O Port Address Decoding



A port decoder that decodes 8-bit I/O ports. This decoder generates active low outputs for ports F0H–F7H

I/O Port Address Decoding



A port decoder that decodes 8-bit 1/0 ports. This decoder generates active low outputs for ports FOH-F7H.

VHDL code for the decoder for PLD

```
library ieee;
use ieee.std_logic_1164.all;

entity DECODER_11_11 is

port (
      A7, A6, A5, A4, A3, A2, A1, A0: in STD_LOGIC;
      D0, D1, D2, D3, D4, D5, D6, D7: out STD_LOGIC
);

end;

architecture V1 of DECODER_11_11 is

begin

      D0 <= not( A7 and A6 and A5 and A4 and not A3 and not A2 and not A1 and
             not A0 );
      D1 <= not( A7 and A6 and A5 and A4 and not A3 and not A2 and not A1 and
             A0 );
      D2 <= not( A7 and A6 and A5 and A4 and not A3 and not A2 and A1 and
             not A0 );
```

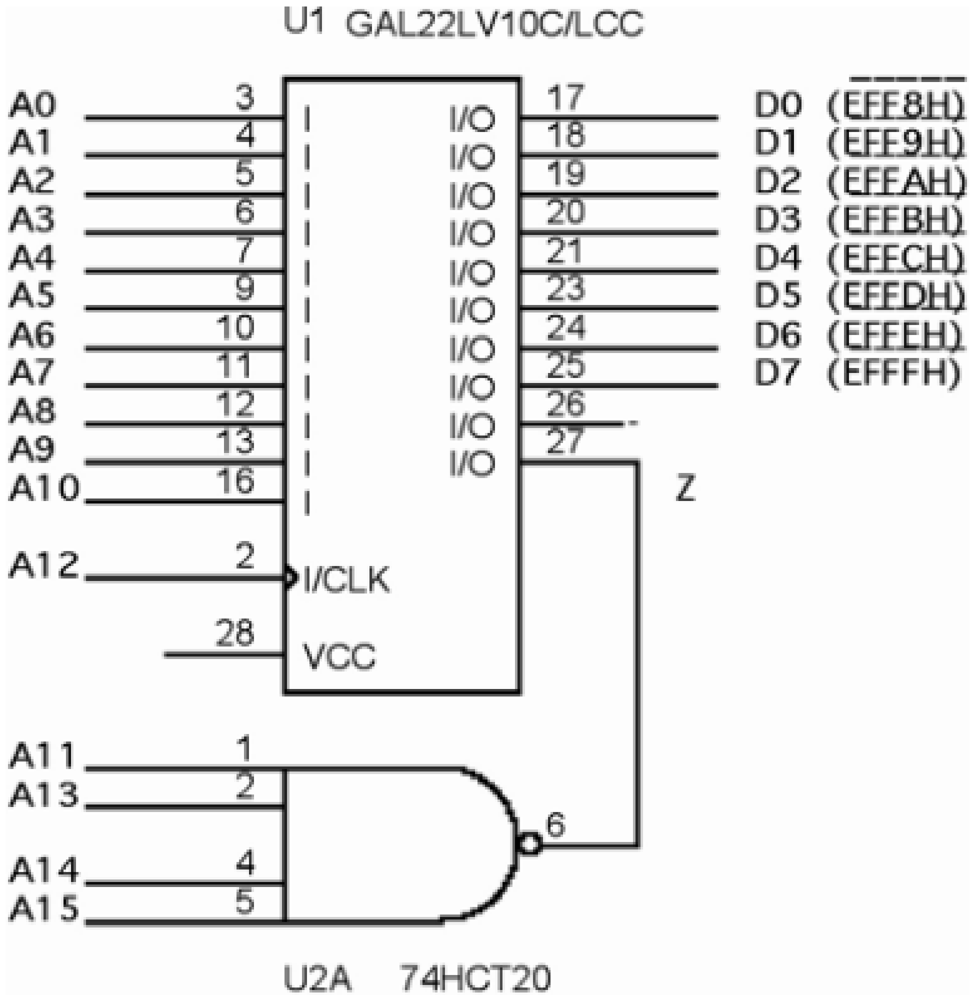
VHDL code for the decoder for PLD

```
        D3 <= not( A7 and A6 and A5 and A4 and not A3 and not A2 and A1 and
                 A0 );
        D4 <= not( A7 and A6 and A5 and A4 and not A3 and A2 and not A1 and
                 not A0 );
        D5 <= not( A7 and A6 and A5 and A4 and not A3 and A2 and not A1 and
                 A0 );
        D6 <= not( A7 and A6 and A5 and A4 and not A3 and A2 and A1 and
                 not A0 );
        D0 <= not( A7 and A6 and A5 and A4 and not A3 and A2 and A1 and
                 A0 );

end V1;
```

Decoding 16-bit I/O Port Addresses

- PC systems typically use 16-bit I/O addresses.
  - 16-bit addresses rare in embedded systems
- The difference between decoding an 8-bit and a 16-bit I/O address is that eight additional address lines (A_{15}–A_8) must be decoded.

Decoding 16-bit I/O Port Addresses



VHDL Code

```
entity DECODER_11_12 is

port (
      Z, A12, A10, A9, A8, A7, A6, A5, A4, A3, A2, A1, A0: in STD_LOGIC;
      D0, D1, D2, D3, D4, D5, D6, D7: out STD_LOGIC
);

end;

architecture V1 of DECODER_11_12 is

begin

      D0 <= not ( not Z and not A12 and A10 and A9 and A8 and A7 and A6 and A5
             and A4 and A3 and not A2 and not A1 and not A0 );
      D1 <= not ( not Z and not A12 and A10 and A9 and A8 and A7 and A6 and A5
             and A4 and A3 and not A2 and not A1 and A0 );
      D2 <= not ( not Z and not A12 and A10 and A9 and A8 and A7 and A6 and A5
             and A4 and A3 and not A2 and A1 and not A0 );
      D3 <= not ( not Z and not A12 and A10 and A9 and A8 and A7 and A6 and A5
             and A4 and A3 and not A2 and A1 and A0 );
      D4 <= not ( not Z and not A12 and A10 and A9 and A8 and A7 and A6 and A5
             and A4 and A3 and A2 and not A1 and not A0 );
      D5 <= not ( not Z and not A12 and A10 and A9 and A8 and A7 and A6 and A5
             and A4 and A3 and A2 and not A1 and A0 );
      D6 <= not ( not Z and not A12 and A10 and A9 and A8 and A7 and A6 and A5
             and A4 and A3 and A2 and A1 and not A0 );
      D7 <= not ( not Z and not A12 and A10 and A9 and A8 and A7 and A6 and A5
             and A4 and A3 and A2 and A1 and A0 );

end V1;
```

8- and 16-Bit Wide I/O Ports

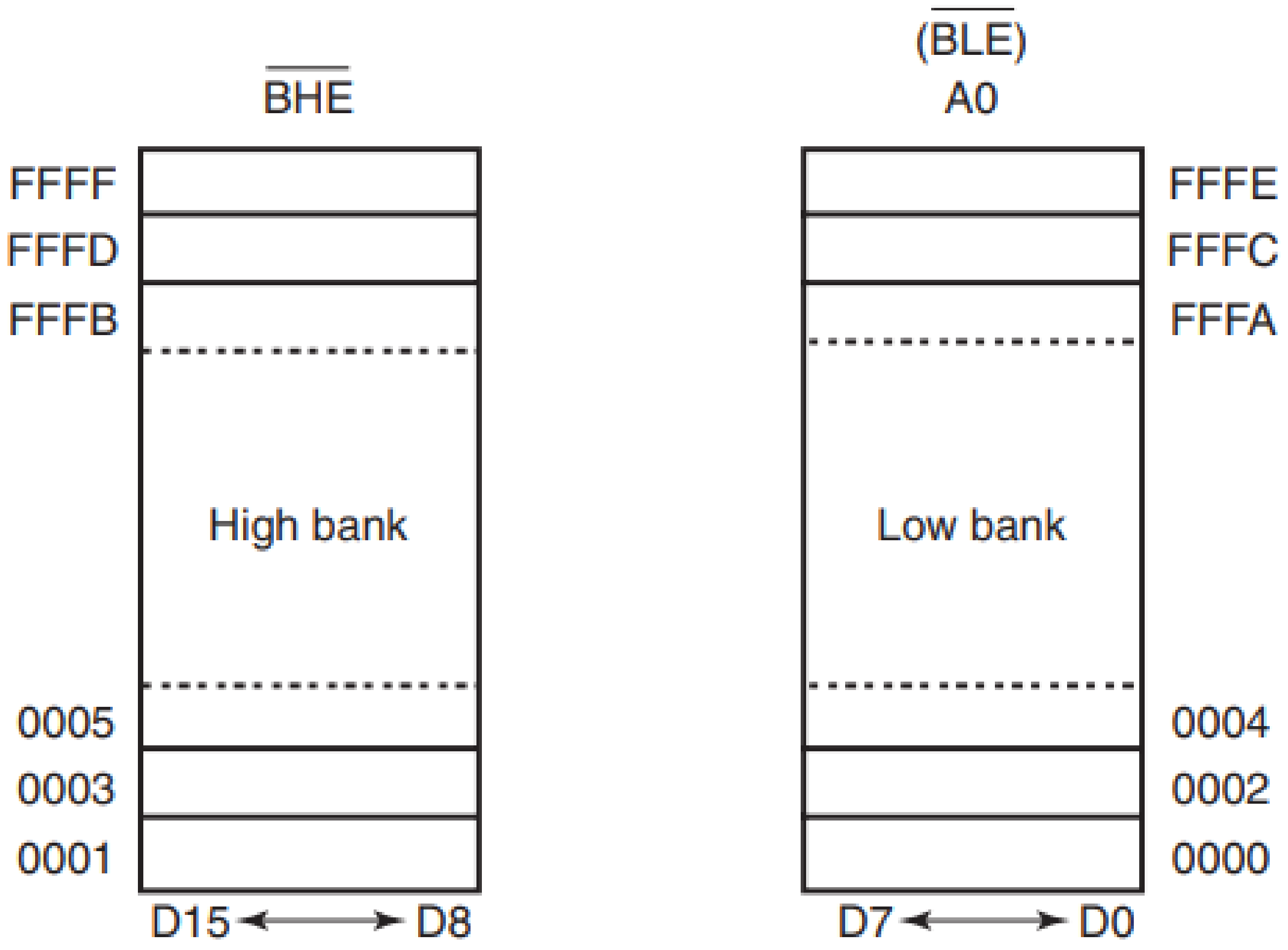
- Data transferred to an 8-bit I/O device exist in one of the I/O banks in a 16-bit processor such as 80386SX.
- The I/O system on such a microprocessor contains two 8-bit memory banks.
- Fig 11–13 shows separate I/O banks for a

16-bit system such as 80386SX.

- Because two I/O banks exist, any 8-bit I/O

write requires a separate write.

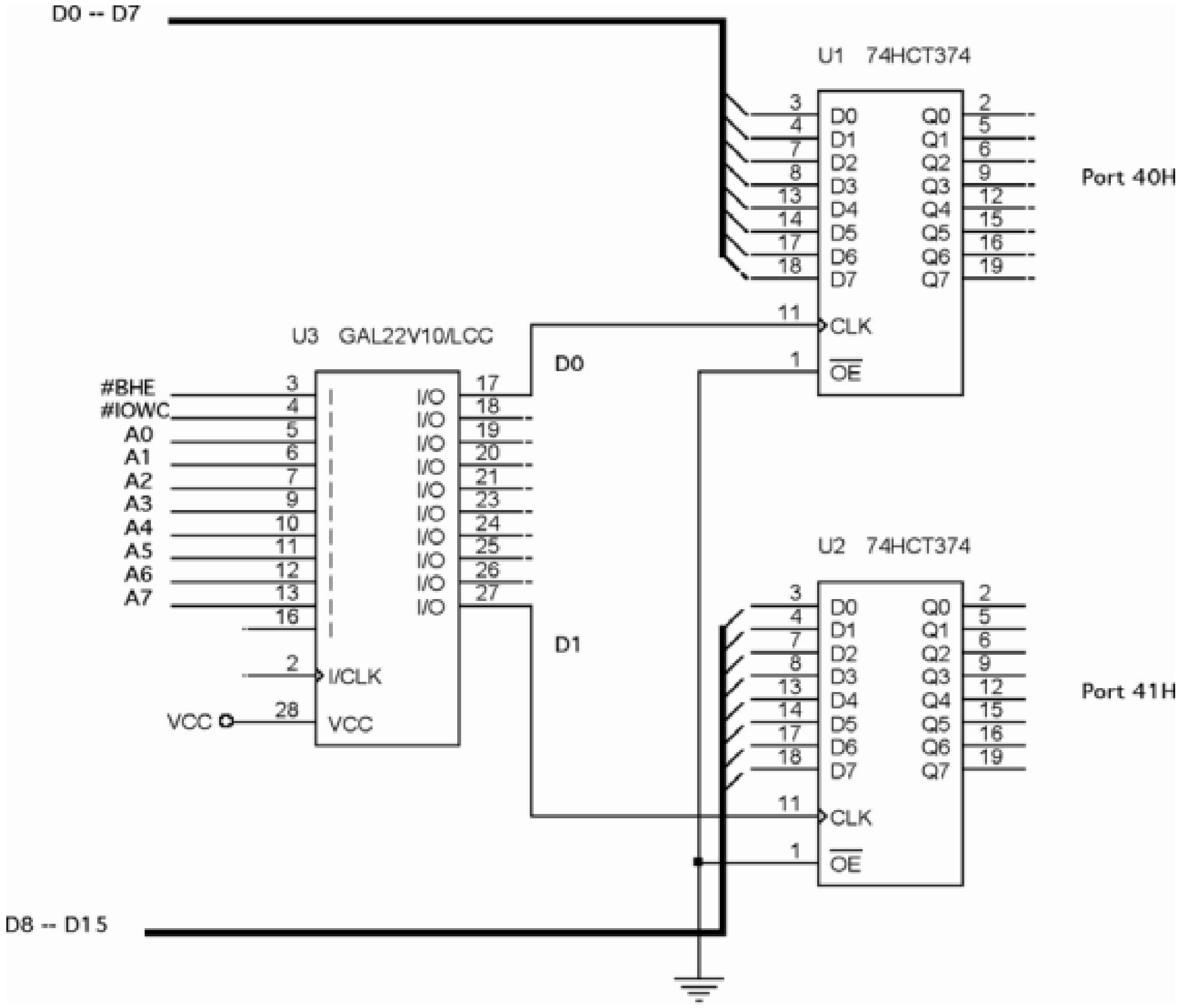
The I/O banks found in the 8086, 80186, 80286, and 80386SX



8- and 16-Bit Wide I/O Ports

- I/O reads don't require separate strobes
  - as with memory, the processor reads only the byte it expects and ignores the other byte
  - a read can cause problems when an I/O device responds incorrectly to a read operation

An I/O port decoder that selects ports 40H and 41H for output data



- all I/O ports use 8-bit addresses
- ports 40H & 41H can be addressed as separate 8-bit ports
- or as one 16-bit port

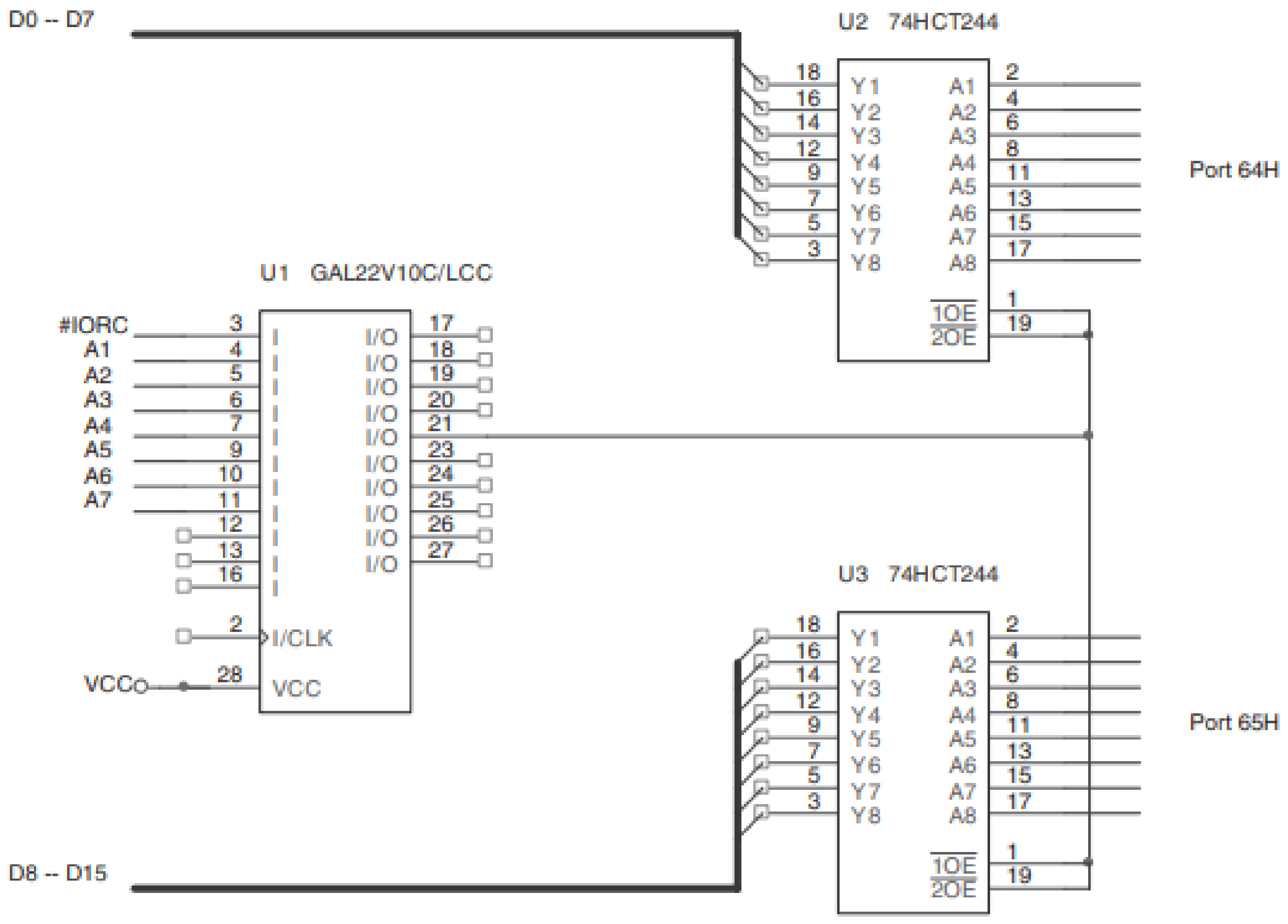
VHDL Code

```
-- VHDL code for the decoder of Figure 11-14

library ieee;

use ieee.std_logic_1164.all;
entity DECODER_11_14 is

port (
        BHE, IOWC, A7, A6, A5, A4, A3, A2, A1, A0: in STD_LOGIC;
        D0, D1: out STD_LOGIC
);

end;

architecture V1 of DECODER_11_14 is

begin

        D0 <= BHE or IOWC or A7 or not A6 or A5 or A4 or A3 or A2 or A1 or A0;
        D1 <= BHE or IOWC or A7 or not A6 or A5 or A4 or A3 or A2 or A1 or
              not A0;

end V1;
```

16-bit Port

- The PLD decoder does not have a connection for address bits BLE (A_0) and BHE because the signals don’t apply to 16-bit-wide devices.
- The program for the PLD shows how the enable signals are generated for the three-state buffers (74HCT244) used as input devices.

16-bit Port



D0 -- D7
U2 74HCT244
Port 64H
U1 GAL22V10C/LCC
#IORC
A1
A2
A3
A4
A5
A6
A7
I/CLK
VCC
VCC
U3 74HCT244
Port 65H
D8 -- D15
Y1 Y2 Y3 Y4 Y5 Y6 Y7 Y8
A1 A2 A3 A4 A5 A6 A7 A8
1OE
2OE

VHDL Code

```vhdl
-- VHDL code for the decoder of Figure 11-15

library ieee;
use ieee.std_logic_1164.all;

entity DECODER_11_15 is

port (
        IORC, A7, A6, A5, A4, A3, A2, A1: in STD_LOGIC;
        D0: out STD_LOGIC
);

end;

architecture V1 of DECODER_11_15 is

begin

        D0 <= IORC or A7 or not A6 or not A5 or A4 or A3 or not A2 or A1;

end V1;
```

innovate achieve lead

BITS Pilani

Pilani Campus

Thank You