I/O - 1

I/O Types, I/O Conditioning, 8288 Bus Controller, Timing

PHY 4635/5635

Spring 2009
I/O Types

**Isolated I/O**: I/O devices are treated separate from memory.
- 1st 64KB (0000h-FFFFh) in memory space = I/O.
- Ports can be byte-wide or word-wide (consecutive memory locations).
- Use IN and OUT instructions and AL (or AX) registers only.
I/O Types

**Memory-mapped I/O**: I/O devices are treated as a memory location. Some of the memory space (in addition to the isolated I/O ports) is wired to I/O ports – The I/O must behave like memory in terms of hardware timing & bus cycles!

- Ports can be byte-wide or word-wide (consecutive memory locations).
- All direct memory instructions apply (but it is slower and you lose some of your memory space).

Example: 4KB segment of memory E0000-E0FFFh is wired to I/O ports instead of memory chips.

This is done by splitting the address and data buses and using decoders, buffers, and latches as we will soon see.
Isolated I/O

- In the PC,
  - 0000h-03FFh I/O space is dedicated to system functions and ISA bus.
  - 0400h-FFFFh I/O space is for user functions and PCI bus.
  - 80287 coprocessor uses 00F8h-00FFh
  - **Fixed port** instructions: 0000h-00FFh
    Stored (e.g. in ROM) with the instruction (8-bits)
  - **Variable port** instructions: > 00FFh
    Stored in DX and then addressed
    e.g. MOV DX, 0303h

*N(OTE: In range 0000h-00FFh, variable*
I/O circuitry

- **Inputs** are BUFFERED
- Inputs need to be TTL
  
  \[0 = 0.0 - 0.8V\]
  \[1 = 2.0 - 5.0V\]
- Example: toggle switch

  Pull-up resistors insure that when switch is open, input=5V; when switch is closed, input=0V (GND)

Buffer selected based on I/O port decoding and possibly timed with peripheral clock (PCLK)

Tri-state buffer, goes to Hi-Z state when not selected.
Input conditioning - debouncing

- Hardware – common bounce elimination via asynchronous flip-flops.
- We did this using delay code.

"Bouncing" is common

Hardware debouncing is cheap and should be done when possible
I/O circuitry

- **Outputs** are LATCHED (often built into I/O device) – holds data, freeing data bus to tend to other work.

Latch selected based on I/O port decoding and possibly timed with peripheral clock (PCLK)
Output conditioning

- CPU output voltages are TTL compatible
  - “0”=0.0-0.4V and “1”=2.4-5.0V
- CPU cannot provide sufficient currents!
  - “0”=0.0-2.0 mA and “1”=0.0-400 μA
- Try transistor or TTL buffer to power output device.

How do we determine resistor values?
2N2222 – general purpose transistor
GAIN = 100

LED needs 10 mA (collector current)

So, base current = 1/100 of collector or 0.1 mA.

Use $\Delta V$ (resistor) =

$\Delta V$ (TTL) - $\Delta V$ (base-emitter)

$\Delta V$ = 2.4V – 0.7V = 1.7V

Current-limiting resistor = $\frac{1.7V}{0.1mA} = 17k\Omega \approx 18k\Omega$

7404 – Inverter
Provides up to 16mA (logic “0”)

LED needs 10 mA

$\Delta V$ (across LED) = 5.0V – 2.0V.

Current-limiting resistor =

$\frac{5.0V-2.0V}{10mA} = \frac{3V}{10mA} = 300\Omega \approx 330\Omega$

1.5V on your handout is incorrect

10 mA on your handout is incorrect
12V / 1A DC-motor

Can’t use an inverter! (16mA max)
Can’t use 2N2222 (250-500mA max)

Use a **Darlington-pair**

Darlington-pair has a minimum gain of 7000 and maximum current of 4A.

Diode prevents inductive kick-back (back EMF) from destroying transistors.

Resistor = \( \frac{0.9V}{1/7000} \) = 6.2kΩ

2 transistor b-e drops \sim 0.143mA
Isolated I/O - Minimum mode (8088)

- A0-A15 only (A16-A19=0)
- Functions performed by “interface circuitry”
  - I/O port decoding
  - Data latching/buffering
  - Synchronize data transfers
  - TTL V/I translation

- IO/M stays high during I/O operation – it can therefore be used to enable I/O latches & decoders in the interface circuitry
Isolated I/O - Minimum mode (8086)

- Similar to 8088 except:
  - D0-D15 (not just D0-D7)
  - M/IO is complemented
  - BHE instead of SS0

- Word-wide data transfers can occur in 1 bus cycle.
  - $A_0$ (BLE) and BHE are used to determine

<table>
<thead>
<tr>
<th>BHE</th>
<th>A0(BLE)</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Both banks (16-bit xfer)</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>High (ODD) bank (8-bits)</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Low (EVEN) bank (8-bits)</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>No banks enables</td>
</tr>
</tbody>
</table>

To ensure 1-bus cycle for word xfers, align I/O ports to even-address boundaries.
Isolated I/O - Maximum mode (8088)

- As in memory interfacing, the 8288 bus controller produces control signals for the I/O system.
- Decodes S0, S1, S2 (see next slide)
8288 Bus Controller

From 8088

<table>
<thead>
<tr>
<th>Status inputs</th>
<th>CPU cycle</th>
<th>8288 command</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\overline{S}_2$</td>
<td>$\overline{S}_1$</td>
<td>$\overline{S}_0$</td>
</tr>
<tr>
<td>0 0 0 0</td>
<td>Interrupt acknowledge</td>
<td>INTA</td>
</tr>
<tr>
<td>0 0 0 1</td>
<td>Read I/O port</td>
<td>IORC</td>
</tr>
<tr>
<td>0 1 1 0</td>
<td>Write I/O port</td>
<td>IOWC, AIOWC</td>
</tr>
<tr>
<td>0 1 1 1</td>
<td>Halt</td>
<td>None</td>
</tr>
<tr>
<td>1 0 0 0</td>
<td>Instruction fetch</td>
<td>MRDC</td>
</tr>
<tr>
<td>1 0 1 0</td>
<td>Read memory</td>
<td>MRDC</td>
</tr>
<tr>
<td>1 1 0 0</td>
<td>Write memory</td>
<td>MWTC, AMWC</td>
</tr>
<tr>
<td>1 1 1 0</td>
<td>Passive</td>
<td>None</td>
</tr>
</tbody>
</table>

RECALL: *Memory mapped I/O* treats all I/O transfers as memory operations (more soon)
Isolated I/O - Maximum mode (8086)

- Note the differences between this and the 8088
  - BHE is provided by the 8086 directly
  - D0-D15
Minimum Mode 8088 - I/O Input Cycle

- Same as for memory read cycle except for IO/M.
- T1: Address lines and ALE are placed on bus (A16-A19=0)

Tells I/O interface circuitry to place data on the bus
Minimum mode 8088 – I/O Output Cycle

- Same as the memory write cycle

Latch data to I/O device
8086 I/O Cycles