EE 109 Unit 18 – Noise Margins, Interfacing, and Tri-States
Signal Types

• Recall even digital signals are *just voltages*...

• Analog signal
  – Continuous time signal where each voltage level has a unique meaning

• Digital signal
  – Continuous signal where voltage levels are mapped into 2 ranges meaning 0 or 1
Signals and Meaning

Each voltage value has unique meaning.

Analog

5.0 v

Digital

5.0 v

Logic 1

2.0 v

Logic 0

0.8 v

Illegal

0.0 v

Threshold Range

Each voltage maps to ‘0’ or ‘1’
(There is a small illegal range where meaning is undefined since threshold can vary based on temperature, small variations in manufacturing, etc.)
NOISE MARGINS, LEVEL SHIFTERS, & DRIVE STRENGTH
A Motivating Example

Example 1

- You connect an output port to an LED (light emitting diode) and connect everything correctly. The light should turn on when you set your output bit to a high voltage (logic '1').
- When you turn the system on the LED does not glow. You measure the voltage at the gate output with a voltmeter and find it is not 5V but 2.3V? Why isn't it a logic 1?
- The maximum current output ability from the output port is not high enough to adequately supply the LED which then drags the voltage down.

Example 2

- You have correctly built a circuit using chips provided by your instructor and verified its outputs
- You then attempt to interface it to a specific microprocessor
- When you connect them the microprocessor indicates that it never senses your circuit producing logic '1'. Why?
- Different circuit implementation techniques use different voltage levels to indicate '1' or '0' and may be incompatible

Lesson To Be Learned: Not all 1's or 0's are created equal!
The Digital Abstraction

• Digital is a nice abstraction of voltage and current
  – Lets us just think 'on' or 'off' but not really worry about the voltages and currents underneath

• Until NOW!!!

• Not all 1's and 0's are created equal
  – A '1' can be any 'HIGH' voltage (maybe in the range 2V-5V)
  – A '0' can be any 'LOW' voltage (maybe in the range 0V-0.8V)
  – So 3V and 5V both mean '1' but they aren't equal

• Similarly certain outputs of a chip may connect to other devices that require more current than the output can produce
  – Think of connecting a fire hose to your garden spigot
  – Or even worse your garden hose to a fire hydrant...it would shred it
  – In the same way, inputs and outputs of different devices must be matched to the demands/requirement of what they connect to
Digital Voltage Noise Margins

- Consider one digital gate feeding another

As long as $V_{OH} > V_{IH}$ and $V_{OL} < V_{IL}$ we are in good shape...

Electromagnetic interference & power spikes can cause this to break down

**Output** Range Interpretation

<table>
<thead>
<tr>
<th>Voltage</th>
<th>Logic</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_{OH}$</td>
<td>Logic 1</td>
<td>High</td>
</tr>
<tr>
<td>$V_{OL}$</td>
<td>Logic 0</td>
<td>Low</td>
</tr>
</tbody>
</table>

**Input** Range Interpretation

<table>
<thead>
<tr>
<th>Voltage</th>
<th>Logic</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_{IH}$</td>
<td>Logic 1</td>
<td>High</td>
</tr>
<tr>
<td>$V_{IL}$</td>
<td>Logic 0</td>
<td>Low</td>
</tr>
</tbody>
</table>

$NM_H = V_{OH} - V_{IH}$

$NM_L = V_{IL} - V_{OL}$

**Symbols:**
- $OH = Output High$
- $OL = Output Low$
- $IH = Input High$
- $IL = Input Low$
- $NM = Noise Margin$
Class Activity

• Do an internet search for "74LS00 datasheet" (this is a chip w/ some 2-input NAND gates) and try to find any PDF and open it

• Skim the PDF and try to find:
  – VOH, VIH, VOL, VIL
Analogy

- Consider a sprinkler system...what will happen if you add 100 new sprinklers to your backyard?
- Pressure (voltage) will go way down and reduced water (current) flow coming out of each
Current Limitations

- When a circuit outputs a 'HIGH' ('1') it can only supply \( \textbf{source} \) so much current (think of your garden hose spigot) = \( I_{OH} \)
- When a circuit outputs a 'LOW' ('0') it can only suck up \( \textbf{sink} \) so much current = \( I_{OL} \)
- When a circuit receives a 'HIGH' signal on the input side it may need a certain amount of current to recognize the input as 'HIGH' = \( I_{IH} \)
- When a circuit receives a 'LOW' signal on the input side it may need a certain amount of current to recognize the input as 'LOW' = \( I_{IL} \)
Consideration

• If we attach too many gates to one output it may not be enough to drive those gates
• Need to make sure the current requirements and capabilities match
• Let's say we connect one of the NAND gates on the 74LS00 chip to an input of N other NAND gates...
• Can it produce/suck up the required current...
• ...if N = 6?
• ...if N = 12?

If $I_{OH}$ or $I_{OL}$ is too low we can split the loads by place intermediate buffers
All In the Family

• There are many families of circuit devices that talk different language (Each has a different VOH, VIH, VOL, VIL, IOL, IIL, etc.)

• Examples:
  – CMOS
  – TTL
  – ECL

• Must make sure if you interface two different devices that they are compatible (i.e. VOH of device A is greater than VIH of device B) or use a buffer/amplifier/level shifter circuit to help them talk to each other

A VOH=2.2V  B VIH=3.5V
Arduino Limits

• Arduino outputs can sink (suck up) and source (produce) around a maximum of 20 mA on a pin

• Do an internet search for "Standard Servo Motor Datasheet" and find the maximum current it may need

• It doesn't seem like the Arduino would be able to drive the servo motor.
  How is it working?
  – Remember the 3-pin interface: R = Power, B = Ground, W = Signal
  – The signal is separate from the power
  – The power source is used to amplify the signal
Another Example

• Now consider a speaker system where the power and signal are provide together
  – Given our Arduino use 5V = Vcc and its current limitations per pin, how much power can we supply to the speaker?
  – 5V * 20 mA = 0.1W
  – You need an amplifier...

Power & Signal together
TRI-STATE GATES
Tri-State Gates

The output of a gate is normally in one of two logic levels, 0 or 1, as determined by the gate’s logic.

Based on the inputs the internal logic decides whether the output is connected a logic 1 or a logic 0.
Tri-State Gates

In a Tri-State gate, we have a third option of not having the output connected to anything.

The Tri-State control can be used to disconnect the gate output from the internal logic level of the gate. In this case the output is said to be “floating”.
Tri-State Gates

When in the disconnected state, the output is said to be in the “High impedance” or “High-Z” state.

Tri-State outputs are commonly used on the output of multi-bit registers. Multiple registers can be connected together as long as only one output is active at a time.

Big advantage: don’t have to know in advance how many devices will be connected together. Just have to make sure only one is enabled (output active) at any one time.
Tri-State Gates

Tri-State gates give us the option of connecting together the outputs of many devices without requiring a circuit to multiplex many signals into one.
Tri-State Gates

When used in place of MUXs, the benefits of using the Tri-State method becomes greater as size of data increases from one to 8 to 32 bits, etc.

Each MUX handles one output bit, so doubling the width of the data doubles the number of multiplexors required.

Tri-State requires logic to create the output enable signals, but this doesn’t change as the data width increases.
Tri-State Gates

Problem: How can you use the serial I/O lines of the Arduino, which are also used for programming it?

Two active devices, both trying to output a signal, collide here.
Solution: Use a Tri-State gate to isolate the MAX232 received data from the μC until programming is over.

Output of gate is floating until μC program makes Pxx a zero.