Computer Logic and Digital Circuit Design (PHYS306/COSC330): Week 1

Jordan Hanson

August 24, 2021

Whittier College Department of Physics and Astronomy

Summary

Summary

1. Logic functions
   - Basic gate ideas
   - Combinatorial logic functions
   - Serial and parallel
2. Programmable logic (PL) vs. fixed integrated circuits (ICs)
3. Test instruments
   - Oscilloscope
   - Signal generator
   - Digital voltmeter and DC power supply

Analog Circuits

First of all, what is an *analog* circuit?

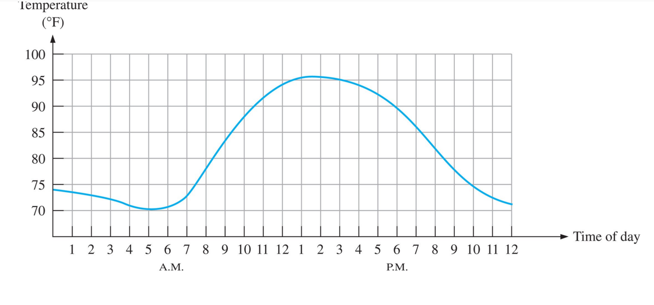


Figure 1: An example of an analog signal from a temperature sensor, converted from voltage.

Analog Circuits

First of all, what is an *analog* circuit?

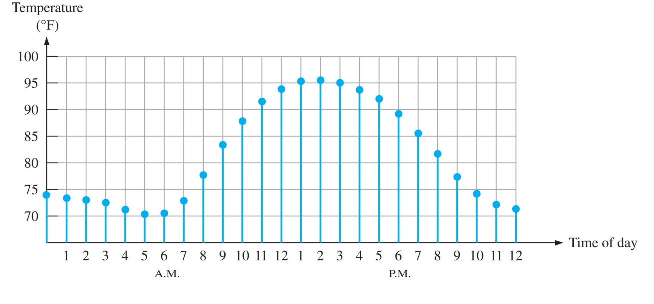


Figure 2: An example of that same signal, digitized and sampled.

Digital data forms the basis of computation:

- Noise issues, lossless transmission
- Constructed from *digits* ... 1 and 0

Analog Circuits

First of all, what is an *analog* circuit?

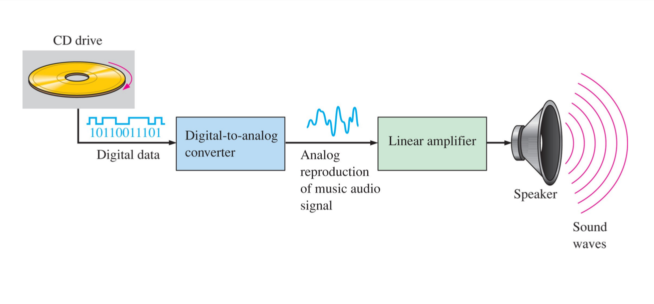


Figure 3: An example of a digital signal converted from binary to analog voltage signal.

One set of examples of analog circuits is RC/LC filters. We must apply *Ohm's Law* (Eq. 1) and *Kirchhoff's Rules* (Eqs. 2-3) to understand these circuits, so let's review those.

$$V = IR \tag{1}$$

$$\sum_{i,node}^{N} j_i = 0 \tag{2}$$

$$\sum_{i,loop}^{N} v_i = 0 \tag{3}$$

A node: a point in a circuit where conductors meet.

A loop: a path in a circuit that returns to the same node.

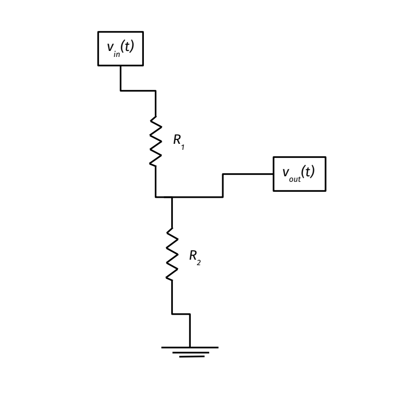


Figure 4: A two-resistor voltage divider.

- Using Ohm's Law, find expressions for $v_{\mathrm{in}}(t)$ and $v_{\mathrm{out}}(t)$ in terms of the resistances and the current.
- Take the ratio of the output voltage to the input voltage to find the **transfer function**, $H(t)$.

Answer:

$$H(t) = \frac{R_2}{R_1 + R_2} \tag{4}$$

Only the amplitude is affected:

- Original signal: $s_{in}(t) = A\cos(2\pi ft + \phi)$
- New signal: $s_{out}(t) = A\left(\frac{R_1}{R_1+R_2}\right)\cos(2\pi ft + \phi)$

Other actions we can do in the time-domain:

- *Delay*: shift the signal in time
- *Filter*: change which frequencies are most powerful
- *Integral/derivative*: circuit output is proportional to the integral or derivative of input

Logic Functions

How do we build up digital data from analog signals?

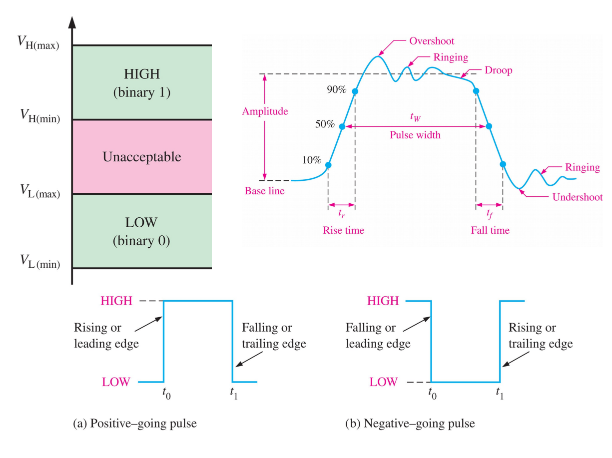


Figure 5: Logical “1” and “0.”

Terminology for digital signals:

1. **Frequency**, f and **period**, T: Signals per second, time between signals ($f = 1/T$).
2. **Pulse width**, t_{W}: time duration a pulse is HIGH.
3. **Duty cycle**: $t_{\mathrm{W}}/T \times 100\%$

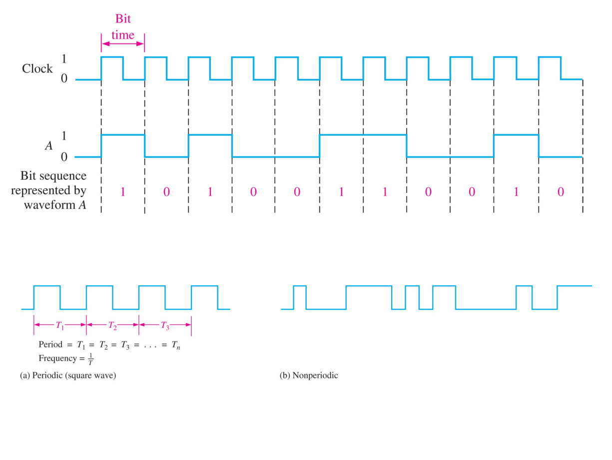


Figure 6: A clock signal is an example of a digital bitstream: alternating 1 and 0. It has a period and a frequency. Data can be *periodic* or *non-periodic*. (Professor: do some examples here).

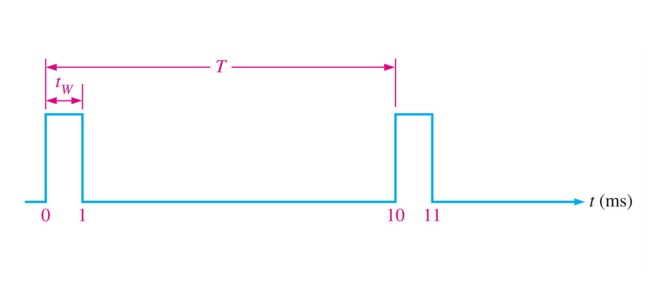


Figure 7: A periodic pulse demonstrating the concept of duty cycle. (Professor: do an example here and vary duty cycle).

What is the duty cycle in Fig. 7? What is the frequency?

Logic Functions

Logic function: mapping digital inputs to outputs. A *timing diagram* displays three digital inputs.

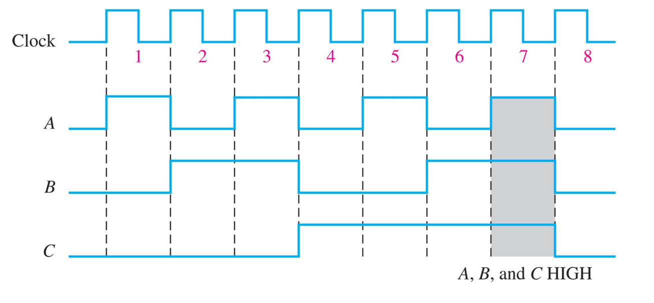


Figure 8: A logic function can be displayed as a timing diagram. Add a forth input, labeled OUT. Suppose OUT is only high when A, B, and C are all high. Draw OUT versus time.

Logic gates: fundamental blocks of logic functions.

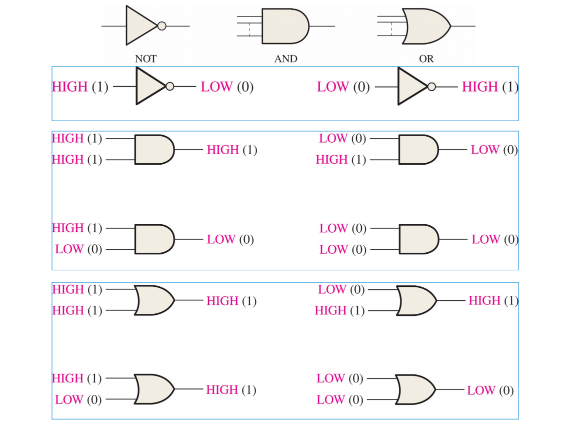


Figure 9: NOT, AND, and OR. These gates are built from transistors, and have *truth tables.*

Logic gates: combinatoric functions are built from gates.

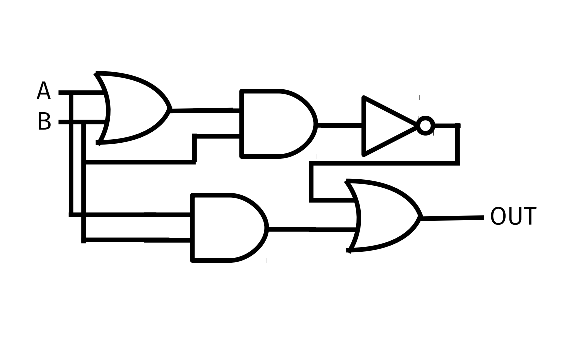


Figure 10: NOT, AND, and OR gates build up other functions. What function is being described here? What three states (values for A and B) would make OUT go high?

Logic Functions

Logic gates: combinatoric functions are built from gates.

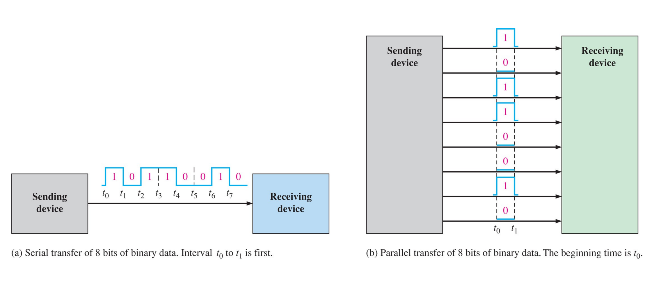


Figure 11: Combinatoric functions can exchange data in serial and parallel form.

PL versus IC

PL versus IC

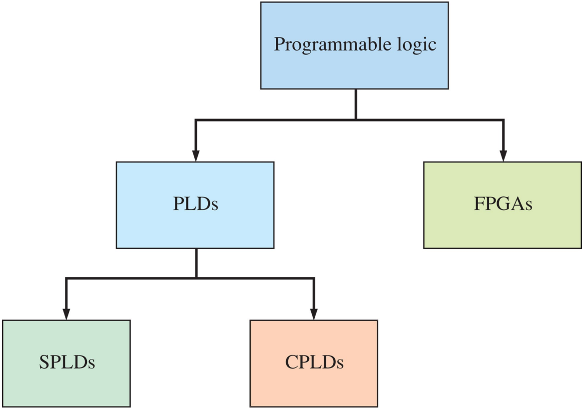


Figure 12: Programmable logic falls into several categories. (a) Programmable logic devices (PLDs), simple and complex. (b) Field Programmable Gate Arrays (FPGAs).

PL versus IC

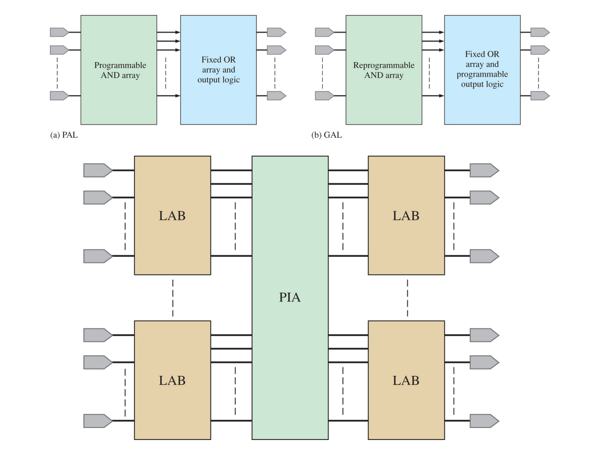


Figure 13: Programmable logic devices (PLDs), simple and complex. Combinatoric logic functions can be built using these systems.

PL versus IC

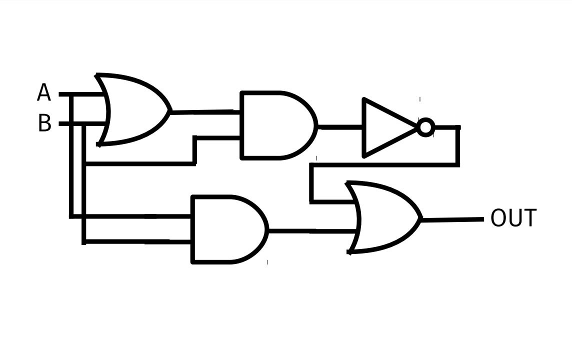


Figure 14: This function reduces to $\bar{B} + AB$.

SPLD could be used to temporarily create this function.

PL versus IC

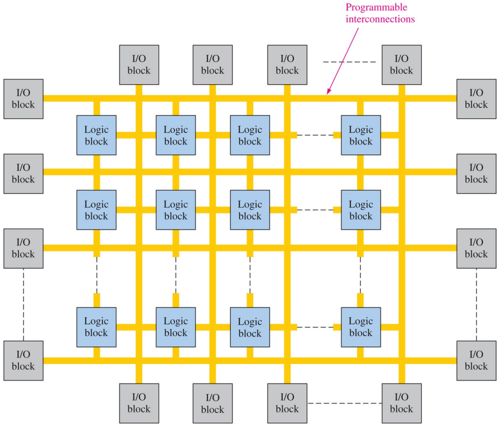


Figure 15: Field Programmable Gate Arrays (FPGAs) can be used to design vast, complex systems of logic functions.

PL versus IC

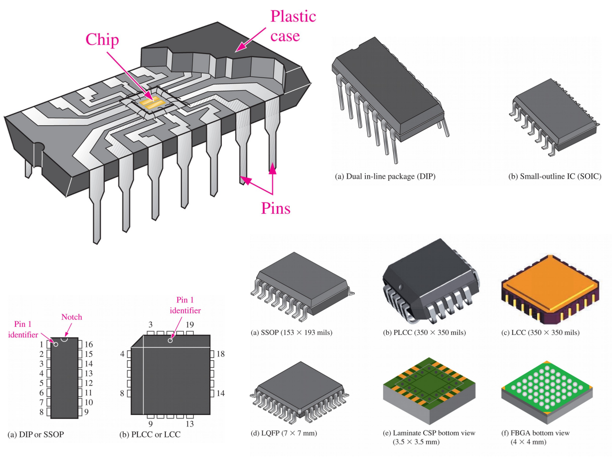
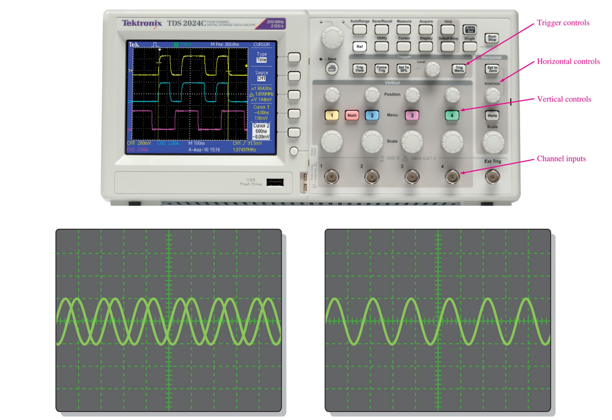


Figure 16: Various types of integrated circuits (ICs), that represent fixed combinatoric functions. They come in through-hole and surface mount varieties.

Test Instruments



(a) Untriggered waveform display

(b) Triggered waveform display

Figure 17: The oscilloscope plots digital and analog signals versus time. The scope needs to *trigger*.

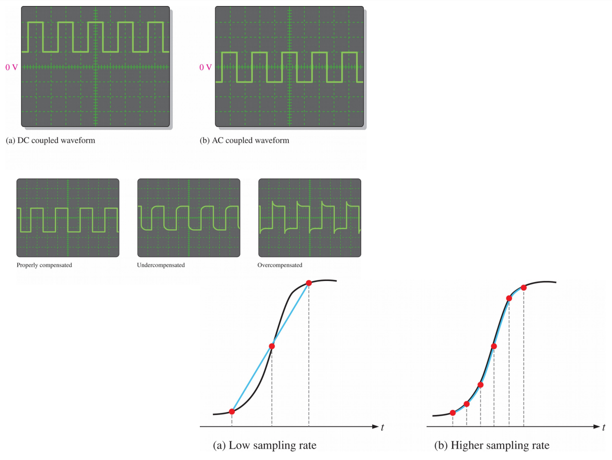


Figure 18: The scope needs to *sample* at a high enough sampling rate to compensate for frequency of data. There's also the issue of AC versus DC coupling. (Professor: do some examples here).

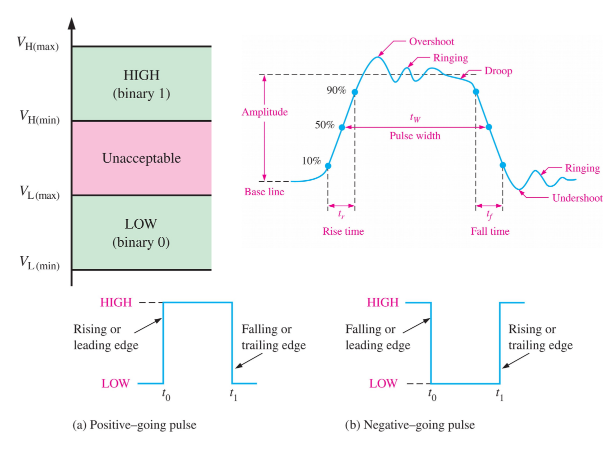


Figure 19: With this issue, we begin our laboratory activity.

Conclusion

Summary

1. Logic functions
    - Basic gate ideas
    - Combinatorial logic functions
    - Serial and parallel
2. Programmable logic (PL) vs. fixed integrated circuits (ICs)
3. Test instruments
    - Oscilloscope
    - Signal generator
    - Digital voltmeter and DC power supply