6.004 Computation Structures
Spring 2009

The Digital Abstraction

1. Making bits concrete
2. What makes a good bit
3. Getting bits under contract

Handouts: Lecture Slides

Concrete encoding of information

To this point we've discussed encoding information using bits. But where do bits come from?

If we're going to design a machine that manipulates information, how should that information be physically encoded?

He said to his friend, "If the British march
By land or sea from the town to-night,
Hang a lantern aloft in the belfry arch
Of the North Church tower as a signal light,--
One if by land, and two if by sea;
And I on the opposite shore will be,
Ready to ride and spread the alarm
Through every Middlesex village and farm,
For the country folk to be up and to arm."

What makes a good bit?
- cheap (we want a lot of them)
- stable (reliable, repeatable)
- ease of manipulation
  (access, transform, combine, transmit, store)

Substrates for computation

We can build upon almost any physical phenomenon…

Wait!
Those last ones
might have potential...

lanterns
dominos
engraved stone tablets
Billiard balls
E. Coli
polarization of a photon

But, since we're EE's…

Stick with things we know about:

voltages	phase
currents	frequency

This semester we'll use voltages to encode information. But the best choice depends on the intended application...

Voltage pros:
- easy generation, detection
- lots of engineering knowledge
- potentially low power in steady state
  zero

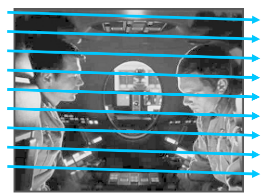
Voltage cons:
- easily affected by environment
- DC connectivity required?
- R & C effects slow things down

Representing information with voltage

Representation of each point (x, y) on a B&W Picture:

0 volts: BLACK
1 volt: WHITE
0.37 volts: 37% Gray
etc.

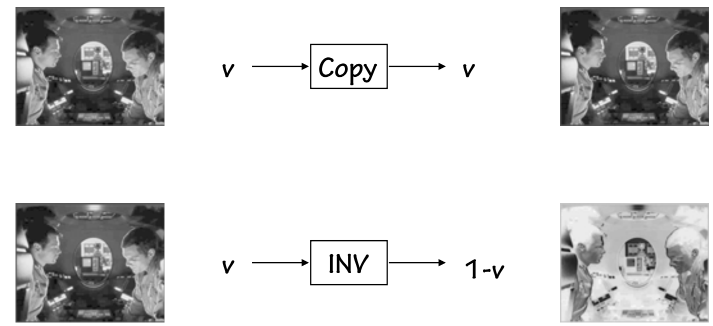
Representation of a picture:
Scan points in some prescribed raster order… generate voltage waveform



How much information at each point?

Information Processing = Computation

First let's introduce some processing blocks:



Why have processing blocks?

The goal of modular design: **Abstraction**

What does that mean anyway:

- Rules simple enough for a 6-3 to follow…
- Understanding BEHAVIOR
  without knowing IMPLEMENTATION
- Predictable composition of functions
- Tinker-toy assembly
- Guaranteed behavior,
  under REAL WORLD circumstances

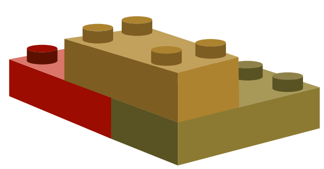
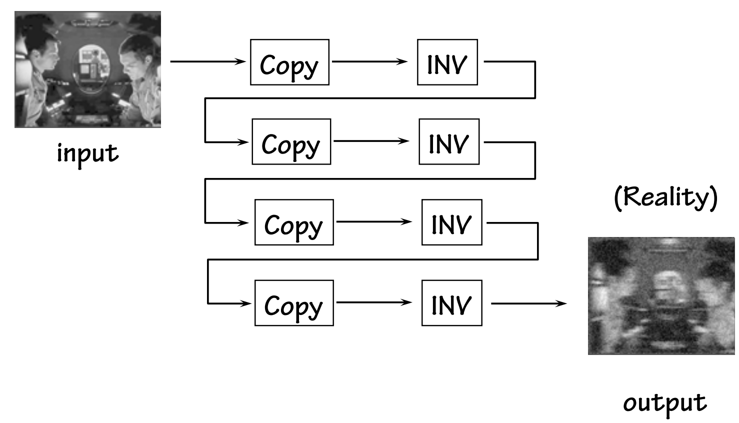


Figure by MIT OpenCourseWare.

Let's build a system!



Why did our system fail?

Why doesn't reality match theory?

1. COPY Operator doesn't work right
2. INVERSION Operator doesn't work right
3. Theory is imperfect
4. Reality is imperfect
5. Our system architecture stinks

ANSWER: all of the above!

Noise and inaccuracy are inevitable; we can't reliably reproduce infinite information-- we must design our system to tolerate some amount of error if it is to process information reliably.

The Key to System Design

A system is a structure that is guaranteed to exhibit a specified behavior, assuming all of its components obey their specified behaviors.

How is this achieved?

Contracts!

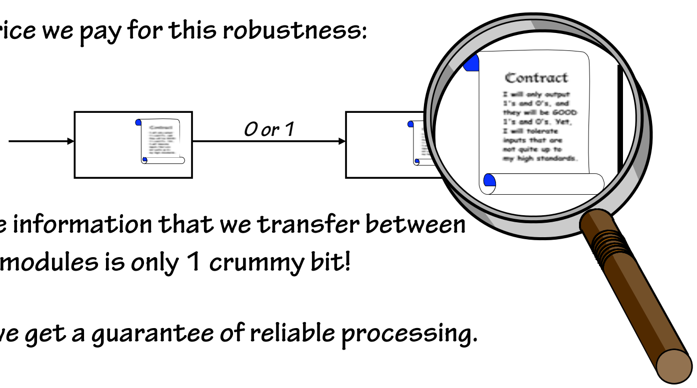
Every system component will have clear obligations and responsibilities. If these are maintained we have every right to expect the system to behave as planned. If contracts are violated all bets are off.

The Digital Panacea ...

Why digital?

... because it keeps the contracts simple!

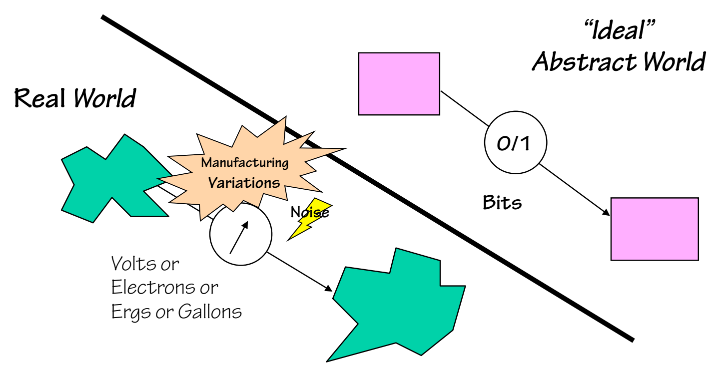
The price we pay for this robustness:



All the information that we transfer between modules is only 1 crummy bit!

But, we get a guarantee of reliable processing.

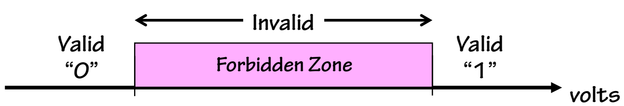
The Digital Abstraction

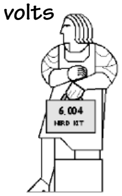


Keep in mind that the world is not digital, we would simply like to engineer it to behave that way. Furthermore, we must use real physical phenomena to implement digital designs!

Using Voltages "Digitally"

- Key idea: don't allow "0" to be mistaken for a "1" or vice versa
- Use the same "uniform representation convention" for *every* component and wire in our digital system
- To implement devices with high reliability, we outlaw "close calls" via a representation convention which forbids a range of voltages between "0" and "1".





CONSEQUENCE:

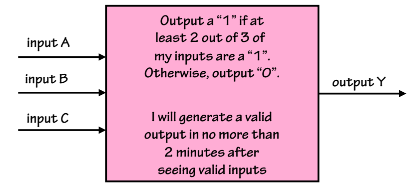
Notion of "VALID" and "INVALID" logic levels

A Digital Processing Element

A <u>combinational device</u> is a circuit element that has

Static discipline

- one or more digital *inputs*
- one or more digital *outputs*
- a *functional specification* that details the value of each output for every possible combination of valid input values
- a *timing specification* consisting (at minimum) of an upper bound t_{pd} on the required time for the device to compute the specified output values from an arbitrary set of stable, valid input values



A Combinational Digital System

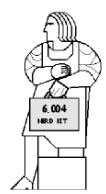
A set of interconnected elements is a combinational device if

- each circuit element is combinational
- every input is connected to exactly one output or to some vast supply of constant 0's and 1's
- the circuit contains no directed cycles

Why is this true?

Given an acyclic circuit meeting the above constraints, we can derive functional and timing specs for the input/output behavior from the specs of its components!

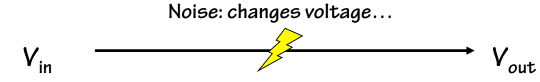
We'll see lots of examples soon. But first, we need to build some combinational devices to work with…



Wires: theory vs. practice

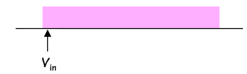
Does a wire obey the static discipline?

Noise: changes voltage…



(voltage close to boundary with forbidden zone)

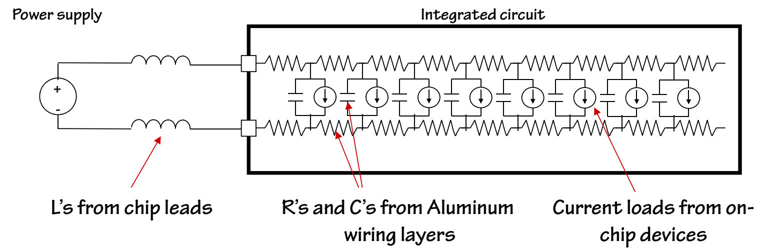
(voltage in forbidden zone: Oops, not a valid voltage!)



Questions to ask ourselves:

In digital systems, where does noise come from?
How big an effect are we talking about?

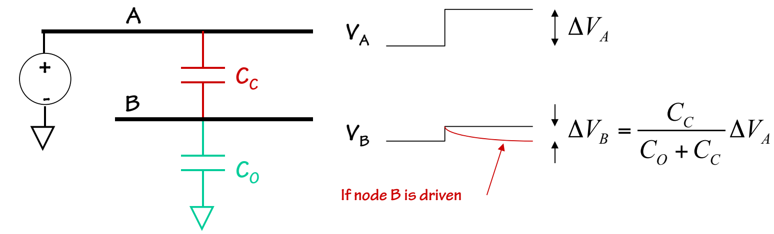
Power Supply Noise



ΔV from:

- IR drop
  (between gates: 30mV, within module: 50mV, across chip: 350mV)
- L(dI/dt) drop
  (use extra pins and bypass caps to keep within 250mV)
- LC ringing triggered by current "steps"

Crosstalk

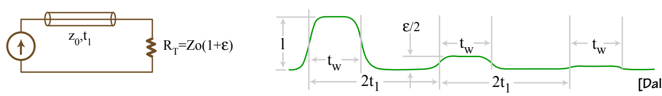


This situation frequently happens on integrated circuits where there are many overlapping wiring layers. In a modern integrated circuit ΔV_A might be 2.5V, C_O = 20fF and C_C = 10fF → ΔV_B = 0.83V! Designers often try to avoid these really bad cases by careful routing of signals, but some crosstalk is unavoidable.

Sequential Interference

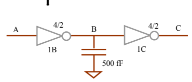
ΔV from energy storage left over from earlier signaling on the wire:

- transmission line discontinuities
  (reflections off of impedance mismatches and terminations)

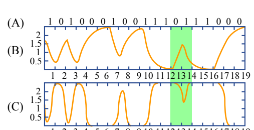


[Dally]Fig. 6-17

- charge storage in RC circuit
  (narrow pulses are lost due to incomplete transitions)

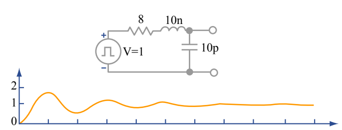


[Dally]Fig. 6-19



[Dally]Fig. 6-20

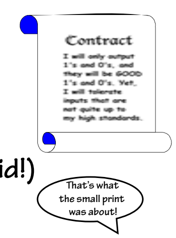
- RLC ringing (triggered by voltage "steps")



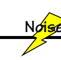
Fix: slower operation, limiting voltage swings and slew rates

Figures by MIT OpenCourseWare.

Needed: Noise Margins!

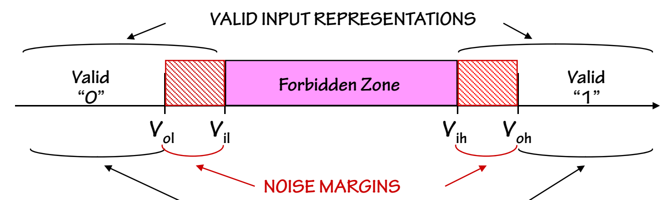


Does a wire obey the static discipline?



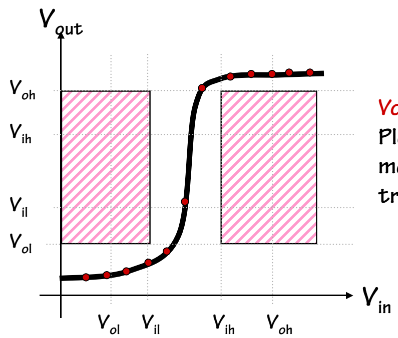
V_{in} (marginally valid) → V_{out} (invalid!)

No! A combinational device must restore marginally valid signals. It must accept marginal inputs and provide unquestionable outputs (i.e., to leave room for noise).



A Buffer

A simple *combinational device*:



Voltage Transfer Characteristic (VTC): Plot of V_{out} vs. V_{in} where each measurement is taken after any transients have died out.

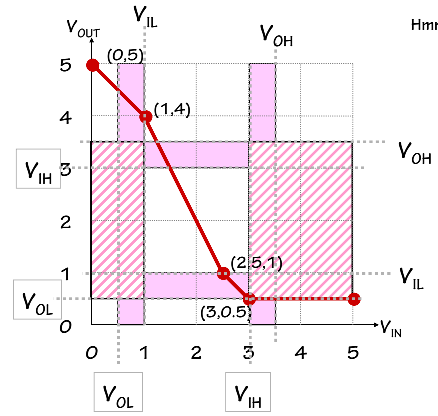
Note: VTC does not tell you anything about how fast a device is— it measures static behavior not dynamic behavior

Static Discipline requires that the VTC avoid the shaded regions (aka "*forbidden zones*"), which correspond to *valid* inputs but *invalid* outputs.

Net result: combinational devices must have GAIN > 1 and be NONLINEAR.

Can this be a combinational device?

Suppose that you measured the voltage transfer curve of the device shown below. Could we build a logic family using it as a single-input combinational device?



Hmmm, it had better be an *INVERTER*…

The device must be able to actually produce the desired output level. Thus, V_{OL} can be no lower than 0.5 V.

Try V_{OL} = 0.5 V

V_{IH} must be high enough to produce V_{OL}

Try V_{IH} = 3 V

Now, choose noise margins – find an N and set

$V_{OH} = V_{IH} + N$

$V_{IL} = V_{OL} + N$

Such that

V_{IH} IN generates V_{OL} or less out; AND

V_{IL} IN generates V_{OH} or more out.

Try N = 0.5 V

Summary

- Use voltages to encode information
- "Digital" encoding
  - valid voltage levels for representing "0" and "1"
  - forbidden zone avoids mistaking "0" for "1" and vice versa
  - gives rise to notion of signal VALIDITY.
- Noise
  - Want to tolerate real-world conditions: NOISE.
  - Key: tougher standards for output than for input
  - devices must have gain and have a non-linear VTC
- Combinational devices
  - Each logic family has Tinkertoy-set simplicity, modularity
  - predictable composition: "parts work → whole thing works"
  - static discipline
    - digital inputs, outputs; restore marginal input voltages
    - complete functional spec
    - valid inputs lead to valid outputs in bounded time

Next time:
Building Logic w/ Transistors