The Digital Abstraction

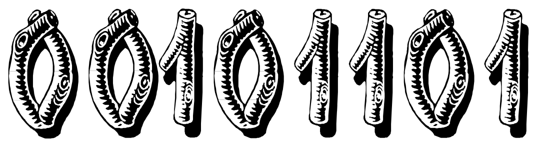
1. Making bits concrete
2. What makes a good bit
3. Getting bits under contract

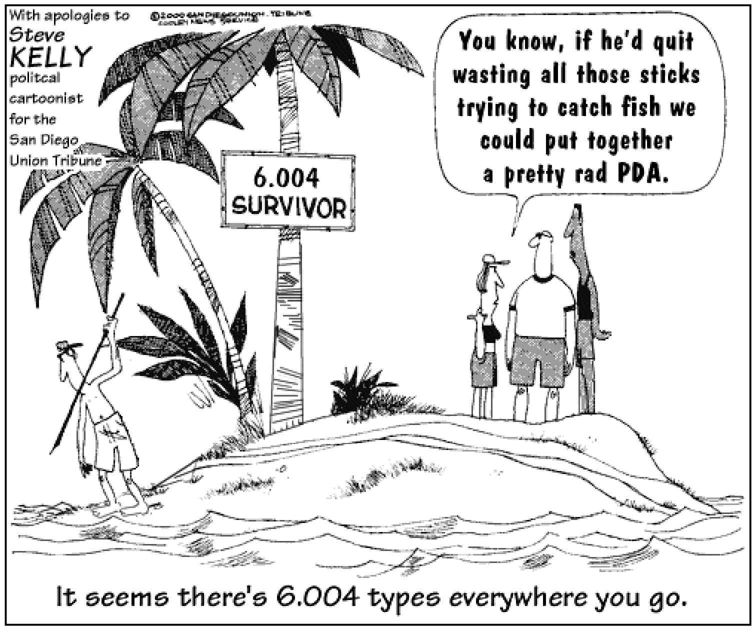
Handouts: Lecture Slides, Problem Set #1

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?




With apologies to Steve KELLY politcal cartoonist for the San Diego Union Tribune
6.004 SURVIVOR
You know, if he'd quit wasting all those sticks trying to catch fish we could put together a pretty rad PDA.
It seems there's 6.004 types everywhere you go.

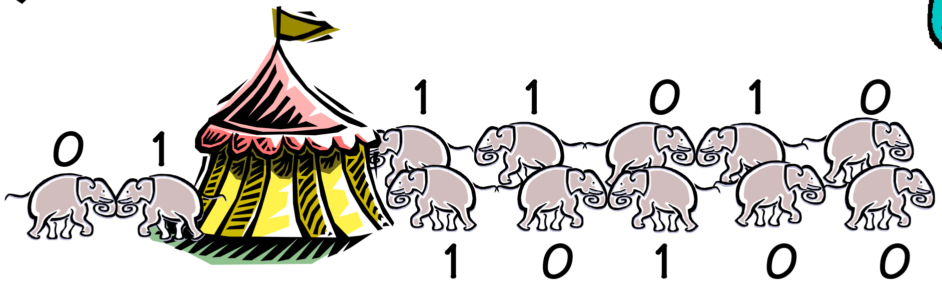
What makes a good bit?

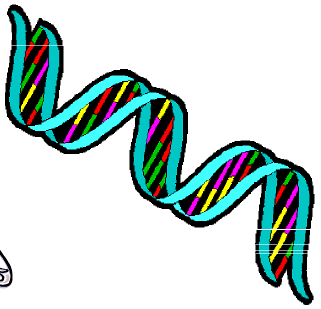
- cheap (we want a lot of them)
- stable (reliable, repeatable)
- ease of manipulation (access, transform, combine, transmit, store)

A substrate for computation

We can build upon almost any physical phenomenon

~~neutrino flux~~
~~elephants~~
~~engraved stone tablets~~
~~orbits of planets~~
~~sequences of amino acids~~
~~polarization of a photon~~





But, since we're EE's…

Stick with things we know about:

voltagesphase

currents frequency

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

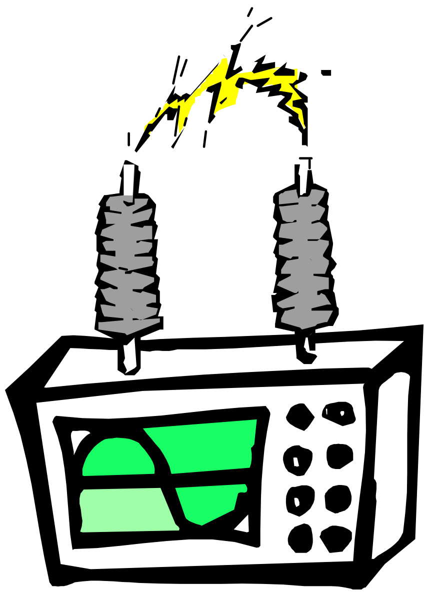
Voltage pros:

- 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

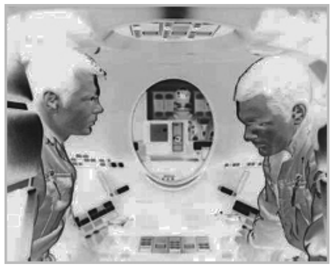
How much information at each point?

Information Processing = Computation

First let's introduce some processing blocks:

$$\begin{tikzcd} v \arrow[r] & \boxed{\text{Copy}} \arrow[r] & v \end{tikzcd}$$

$$\begin{tikzcd} v \arrow[r] & \boxed{\text{INV}} \arrow[r] & 1-v \end{tikzcd}$$

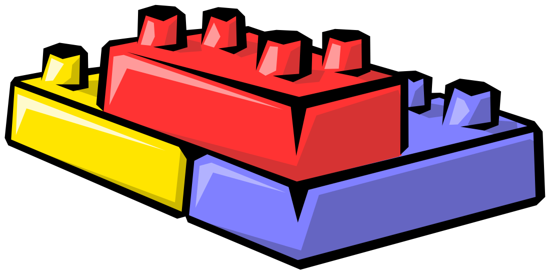


Why have processing blocks?

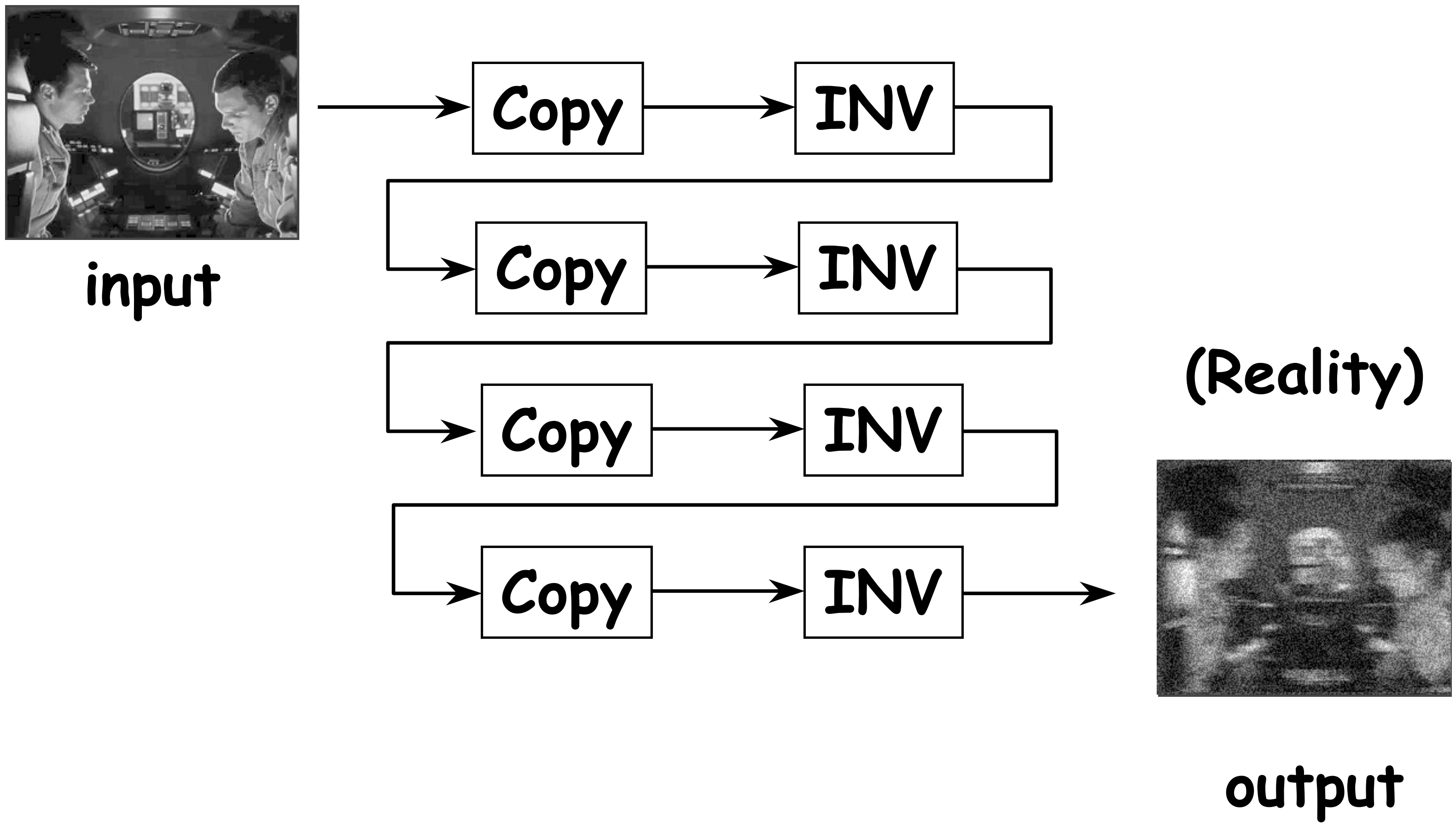
The goal of modular design: Abstraction

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



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

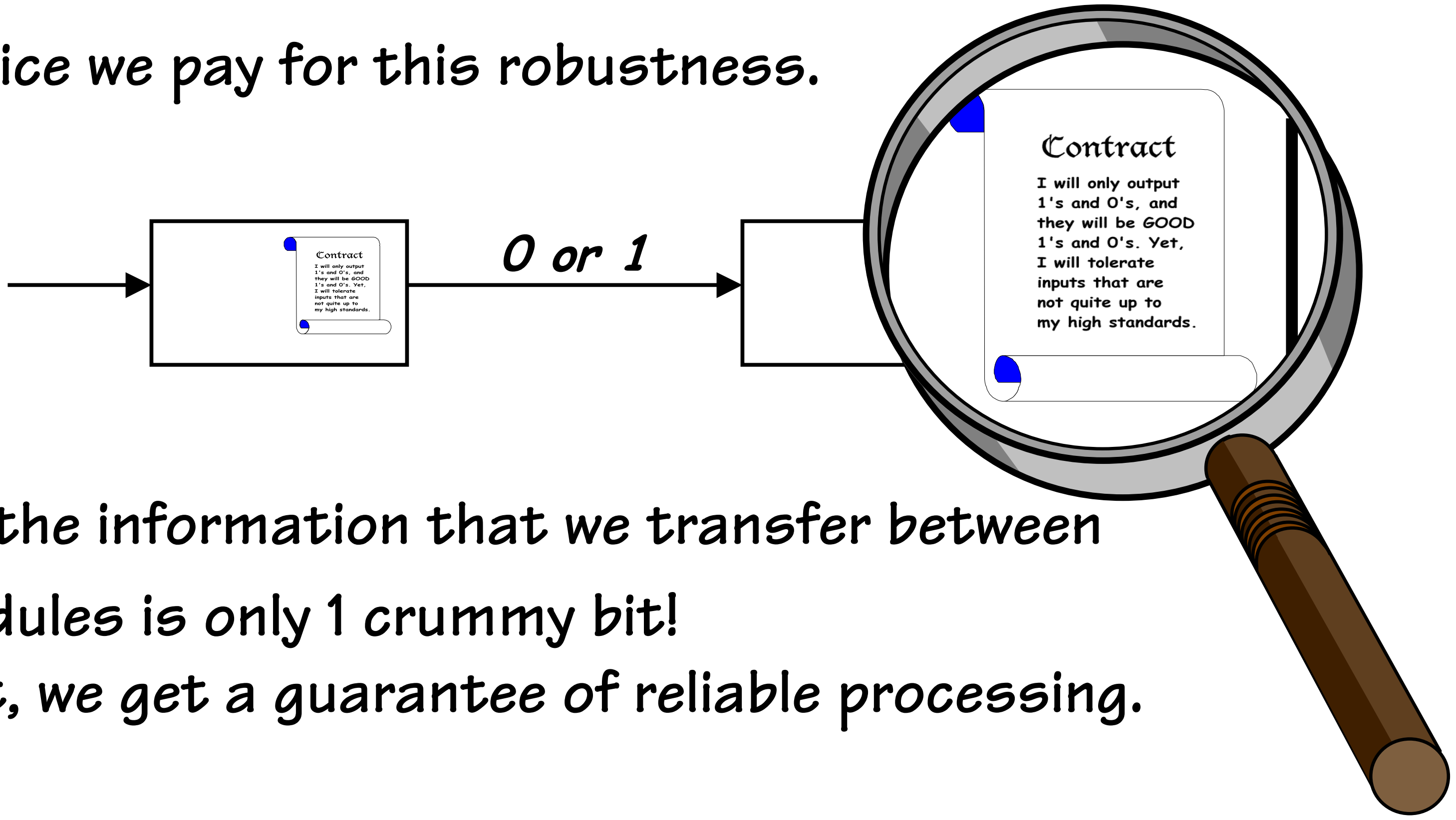
Every system component will have clear obligations and responsibilities. If these are maintained we have every right to expect the system to behave as planned. If contracts are violated all bets are off.

The Digital Panacea ...

Why digital?

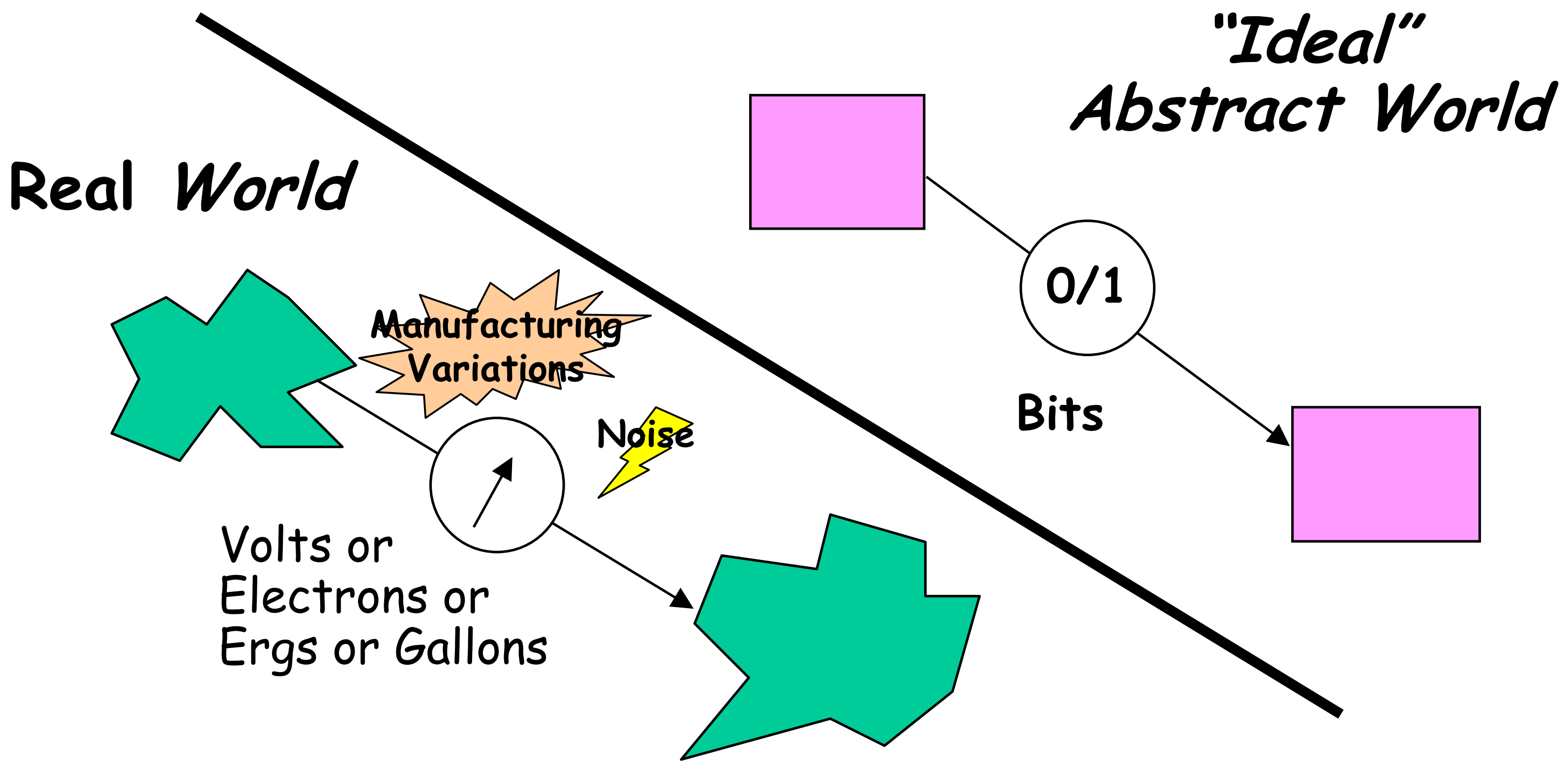
... because it keeps the contracts simple!

The price we pay for this robustness.


0 or 1
Contract
I will only output 1's and 0's, and they will be GOOD 1's and 0's. Yet, I will tolerate inputs that are not quite up to my high standards.

All the information that we transfer between modules is only 1 crummy bit!
But, we get a guarantee of reliable processing.

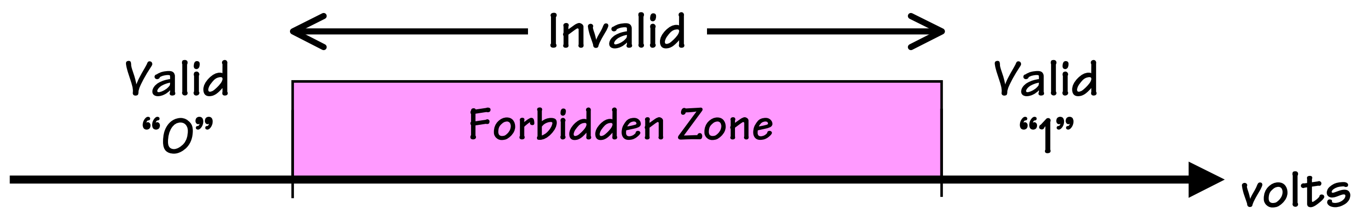
The Digital Abstraction

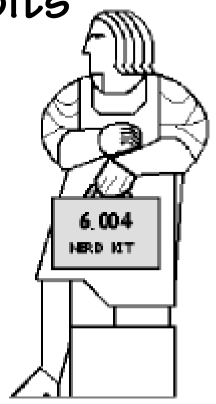


Keep in mind that the world is not digital, we would simply like to engineer it to behave that way. Furthermore, we must use real physical phenomena to implement digital designs!

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".





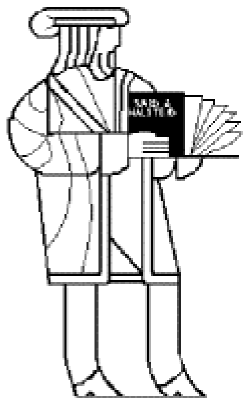
CONSEQUENCE:

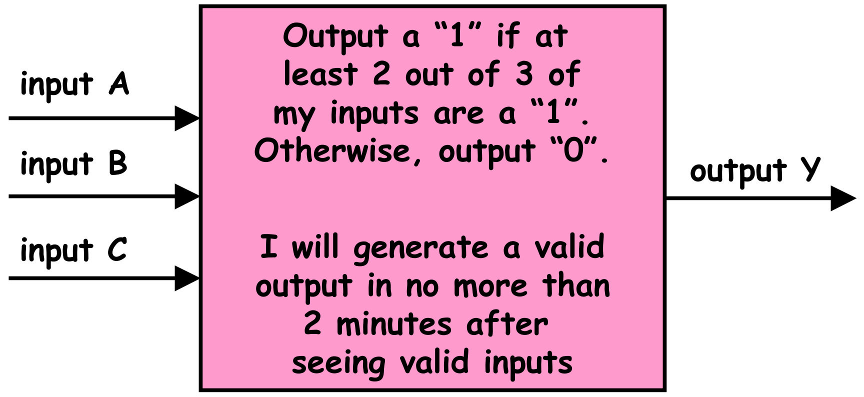
Notion of "VALID" and "INVALID" logic levels

A Digital Processing Element

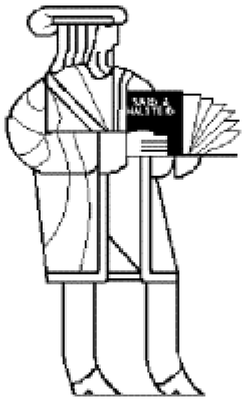
- A *combinational device* is a circuit element that has
  - one or more digital *inputs*
  - one or more digital *outputs*
  - a *functional specification* that details the value of each output for every possible combination of valid input values
  - a *timing specification* consisting (at minimum) 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

Static discipline




input A
input B
input C
Output a "1" if at least 2 out of 3 of my inputs are a "1". Otherwise, output "0".
I will generate a valid output in no more than 2 minutes after seeing valid inputs
output Y

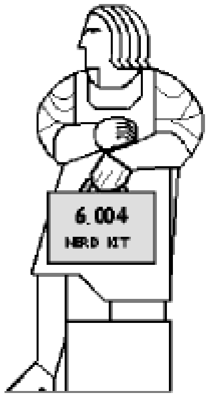
A Combinational Digital System



- 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 0's and 1's
  - the circuit contains no directed cycles

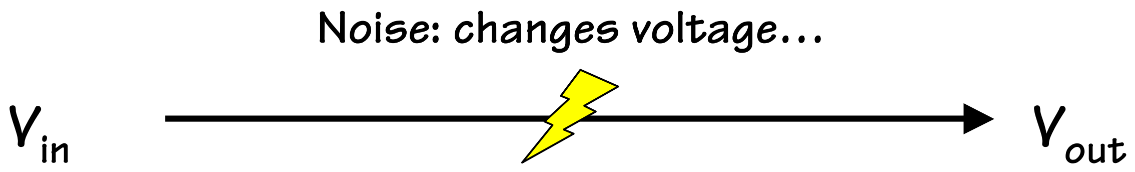
No feedback (yet!)

- But, in order to realize digital processing elements we have one more requirement!



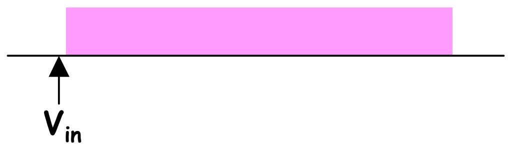
Wires: theory vs. practice

Does a wire obey the static discipline?



(voltage close to boundary with forbidden zone)

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

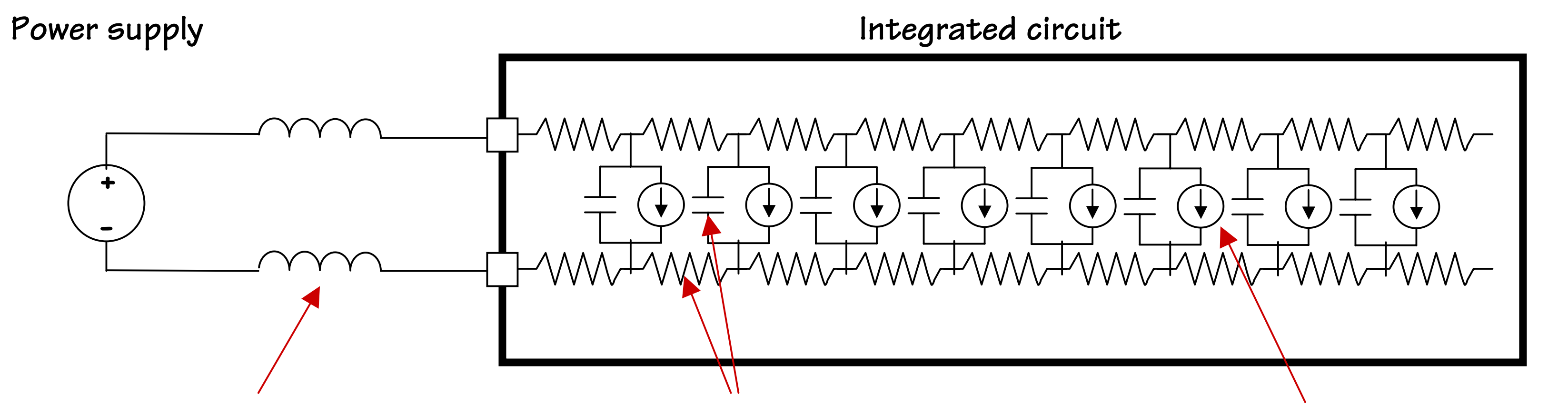


Questions to ask ourselves:

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

A good place to look for answers: [Dally] Ch. 5 & 6

Power Supply Noise



L's from chip leads

R's and C's from Aluminum wiring layers

Current loads from on-chip devices

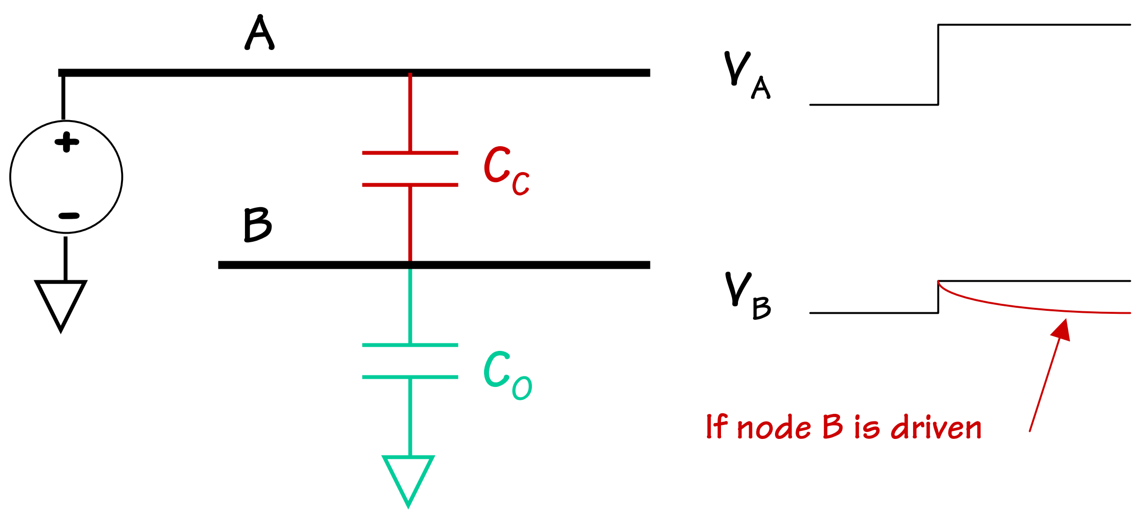
ΔV from:

- 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



$$\Delta V_A$$

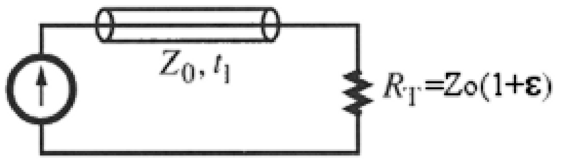
$$\Delta V_B = \frac{C_C}{C_O + C_C} \Delta V_A$$

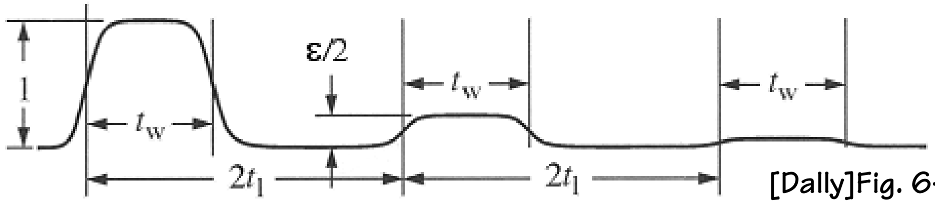
This situation frequently happens on integrated circuits where there are many overlapping wiring layers. In a modern integrated circuit ΔV_A might be 2.5V, C_O = 20fF and C_C = 10fF $\rightarrow$ ΔV_B = 0.83V! Designers often try to avoid these really bad cases by careful routing of signals, but some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

Intersymbol Interference

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

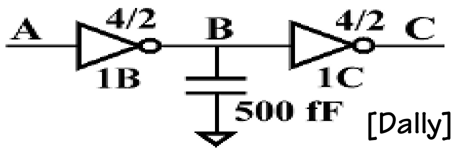
- transmission line discontinuities
  (reflections off of impedance mismatches and terminations)



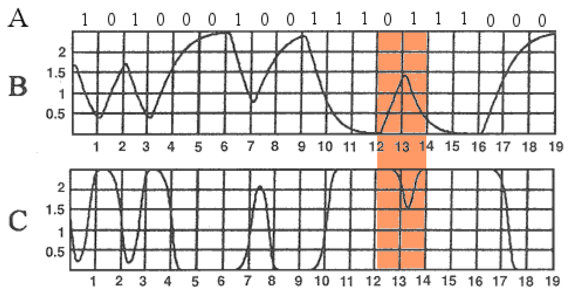


[Dally]Fig. 6-17

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

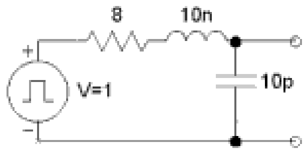


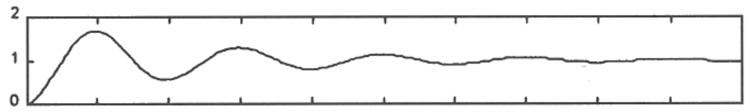
[Dally]Fig. 6-19



[Dally]Fig. 6-20

- RLC ringing (triggered by voltage "steps")

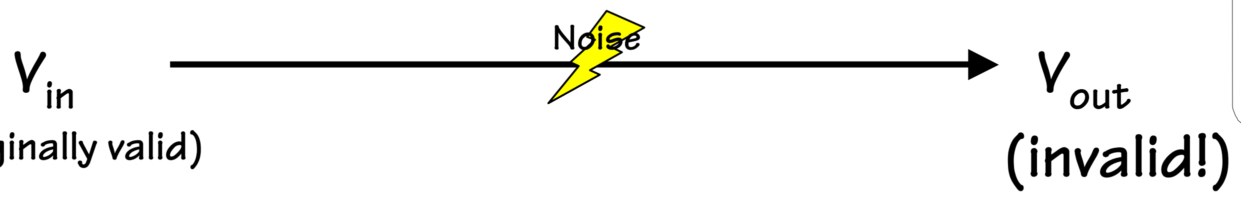


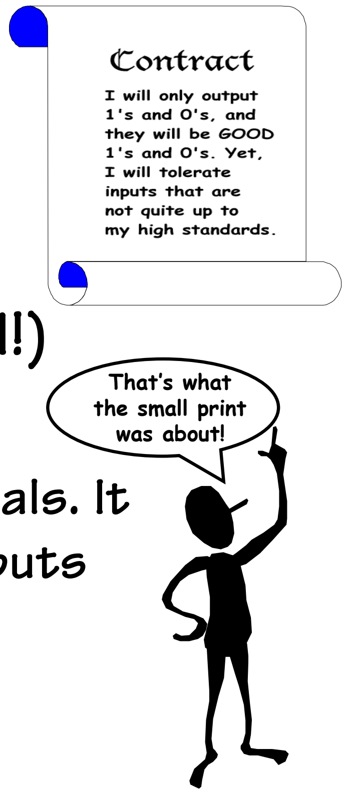


Fix: slower operation, limiting voltage swings and slew rates

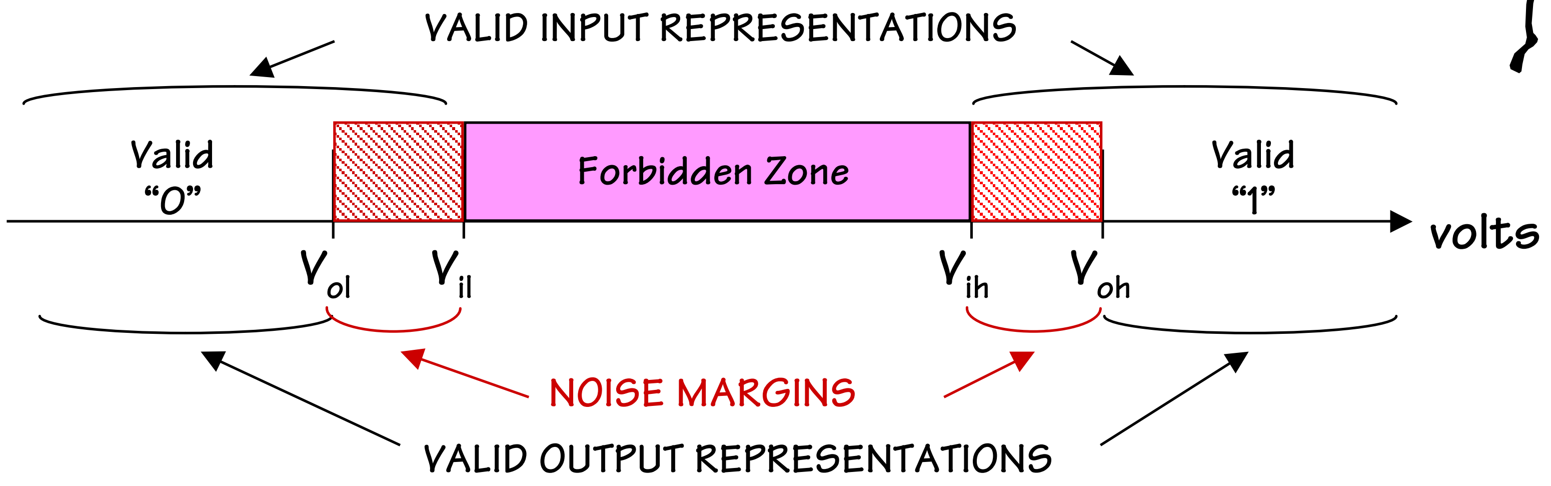
Noise Margins!

Does a wire obey the static discipline?



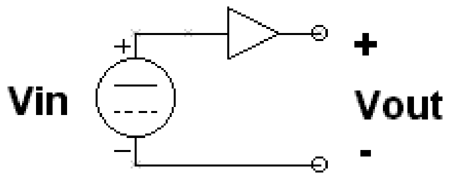

Contract
I will only output 1's and 0's, and they will be GOOD 1's and 0's. Yet, I will tolerate inputs that are not quite up to my high standards.
That's what the small print was about!

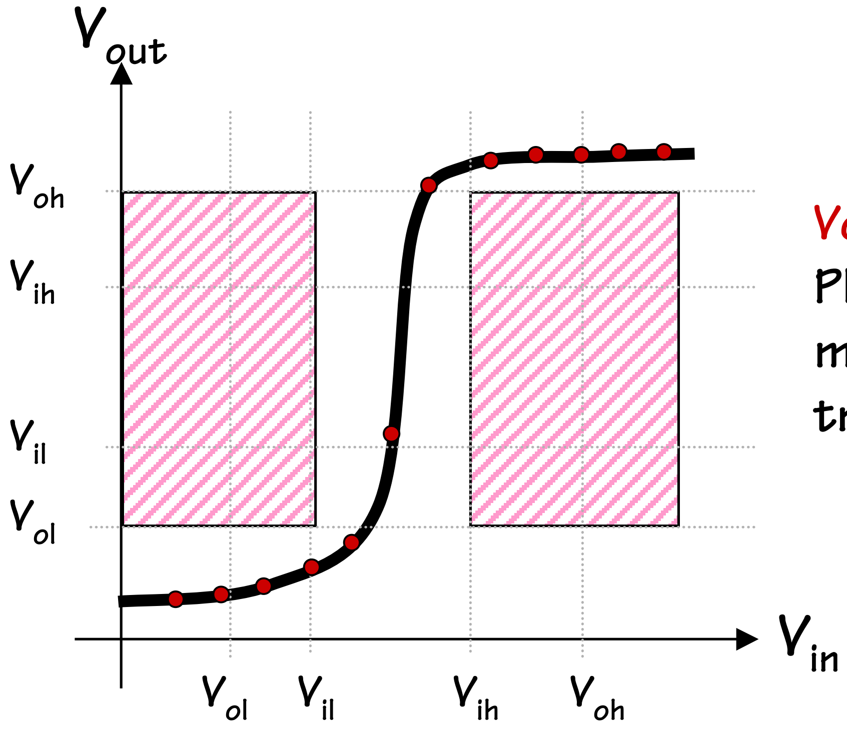
No! A combinational device must restore marginally valid signals. It must accept marginal inputs and provide unquestionable outputs (i.e., to leave room for noise).



An improved wire, a buffer

A simple BUFFER:





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

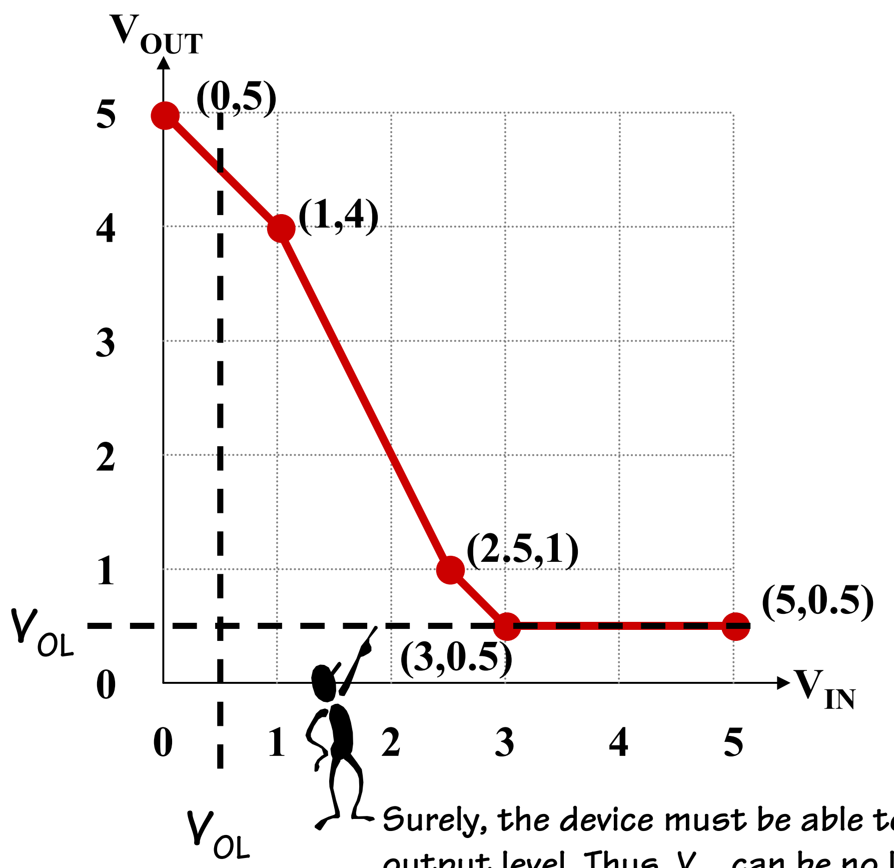
Note: VTC does not tell you anything about how fast a device is— it measures static behavior not dynamic behavior

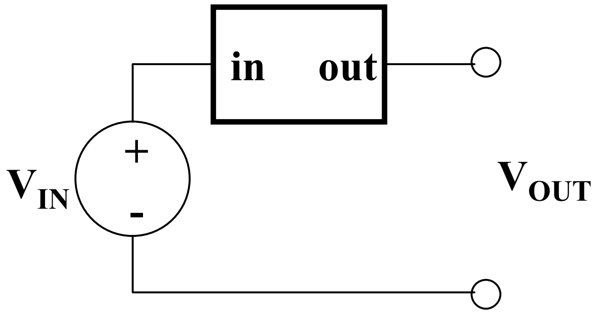
Static Discipline requires that we avoid the shaded regions (aka "forbidden zones"), which correspond to *valid* inputs but *invalid* outputs. Net result:

combinational devices must have GAIN > 1 and be NONLINEAR.

Example VTC

Suppose that you measured the voltage transfer curve of the device shown below. Could we build a digital system with such a device?





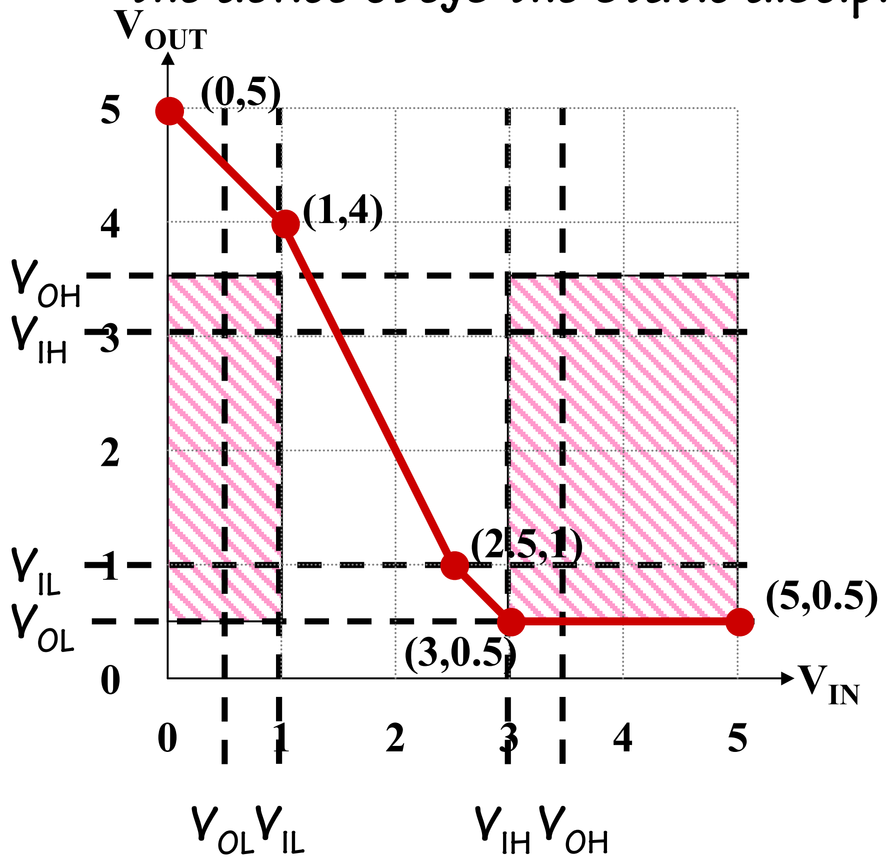
First let's consider the voltage that we will use as a valid "0" or "low" output. Can V_{OL} = 0V?

What's the smallest V_{OL} we can choose and still have our device obey our static discipline?

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

Example VTC (cont'd.)

Assuming that we want to have 0.5V noise margins for both "0" and "1" values, what are appropriate voltage levels for V_{OL}, V_{IL}, V_{IH}, and V_{OH} so that the device obeys the static discipline?

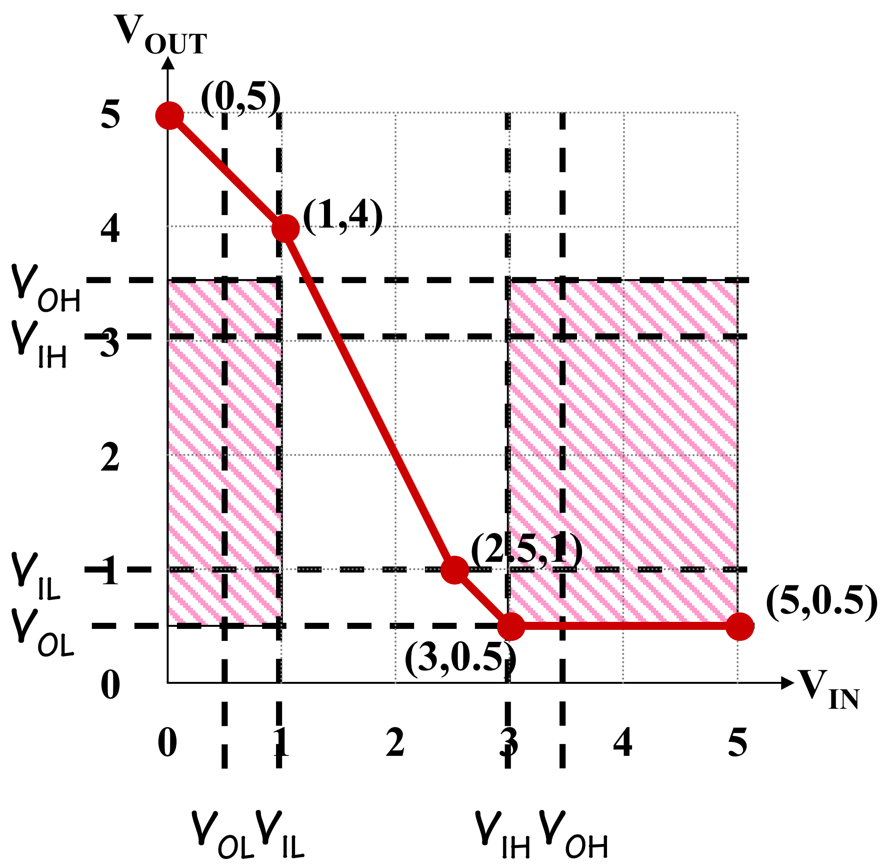


Noise margins are a measure of how far input voltages can be from the valid output voltage, and still be considered valid. If we stick with our value of V_{OL} = 0.5V then V_{IL} > 1.0 V.

Next comes the tricky part. We must select V_{IH} so that our device produces a valid output. For this device that output can be at most V_{OL}. The lowest input voltage that produces V_{OL} is V_{IN} = 3.0V, so V_{IH} can be no lower than 3.0V. V_{OH} must be at least 3.5V to assure our noise margin.

Example VTC (cont'd.)

Now that you get the idea, consider the following questions:



Assuming that we want to have 0.5V noise margins for both "0" and "1" values, what is the largest possible voltage level for V_{OL} that still results in a device that obeys the static discipline?

Assuming that we want to have equal noise margins for both "0" and "1" values, what is the largest noise margin we can achieve with this device and still obey the static discipline?

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
- Noise
  - natural consequence of using R's, L's, C's to connect components
  - add noise margins: $V_{OL} < V_{IL} < V_{IH} < V_{OH}$
  - devices must have gain and have a non-linear VTC
- Combinational devices
  - tinker-toy building blocks; understand behavior not implementation
  - predictable composition: "parts work $\rightarrow$ 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