Analog and Digital: What Isn't the Difference?

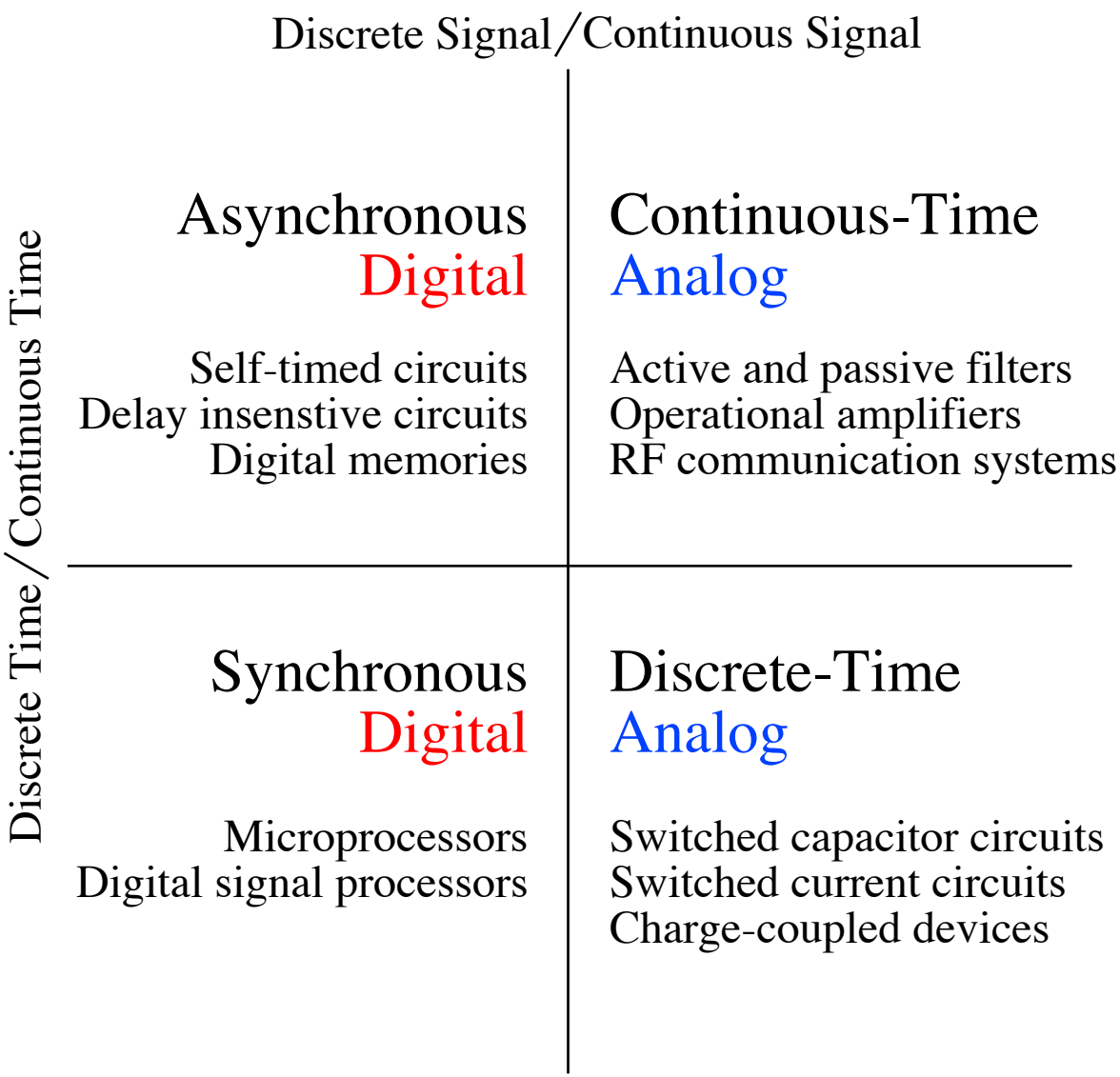
Everybody knows that...

Digital ≡ "Discrete"

Analog ≡ "Continuous"

...right?

Analog and Digital: What Isn't the Difference?



Analog and Digital: What Isn't the Difference?

So...

Digital $\neq$ "Discrete Time"

Analog $\neq$ "Continuous Time"

but is

Digital $\equiv$ "Discrete Signal"

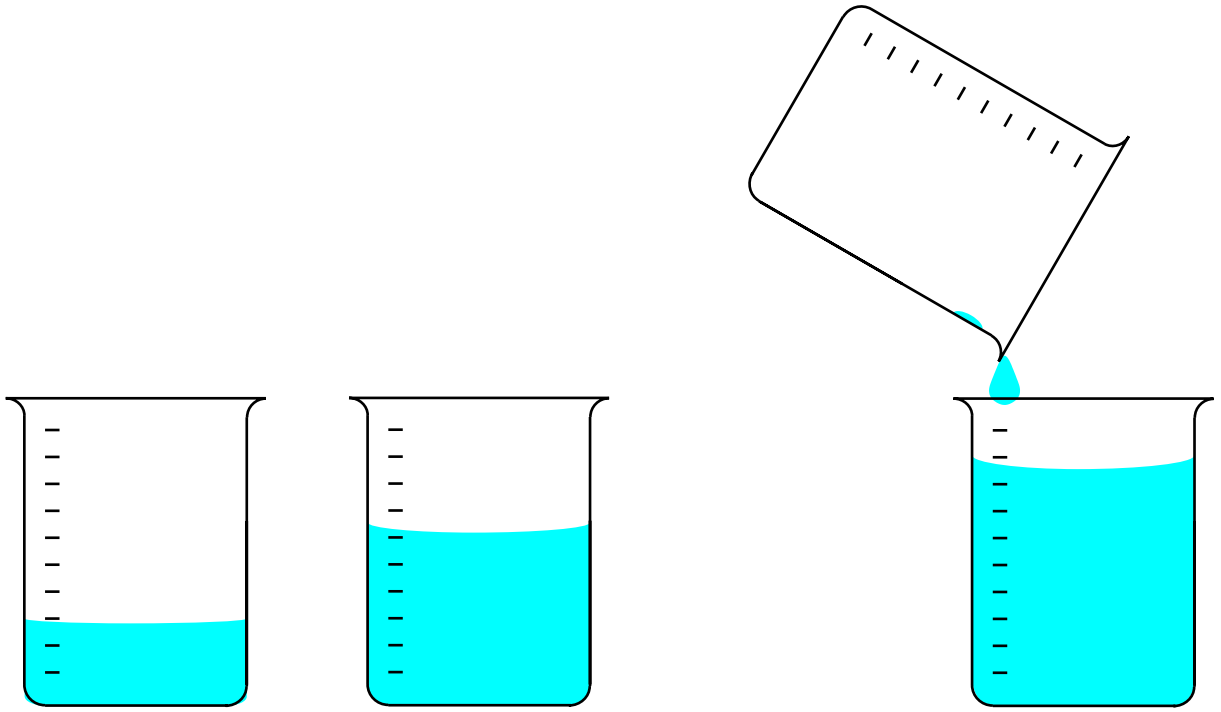
Analog $\equiv$ "Continuous Signal"

???

Analog Computation

- Numbers are represented by the magnitues of physical quantities. (e.g., length, weight, current, voltage)
- These quantities are combined or altered in ways that are *analogous* to the operations that we want to perform on the numbers that they represent.
- The computation is a kind of physical experiment and the result is ascertained by a measurement process.
- Precision is limited by noise, by device mismatch, and by measurement inaccuracies.

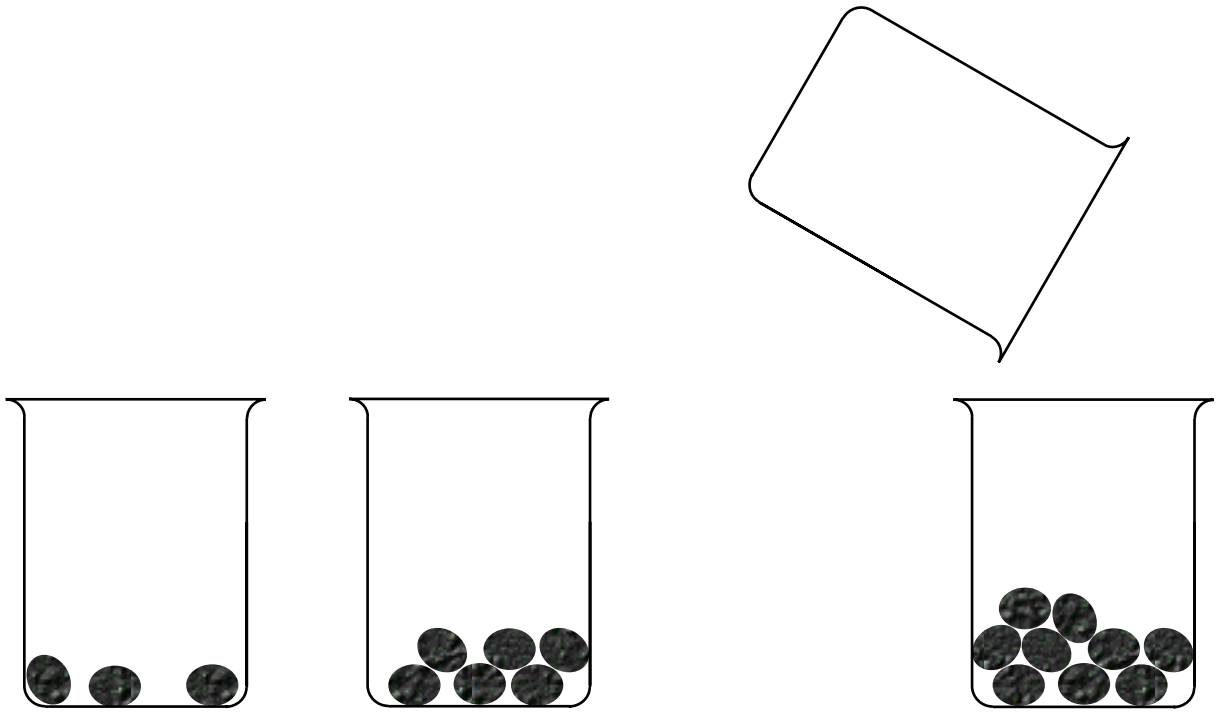
Analog Computation



Digital Computation

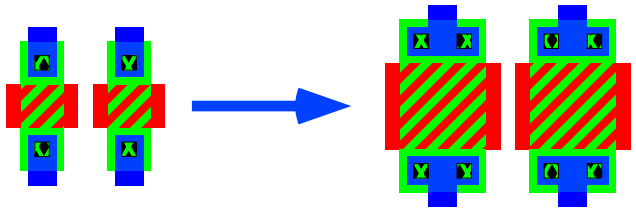
- Numerals are represented by the presence or absence of physical quantities.
- The presence or absence of these tokens are altered according to prescribed rules (e.g., a truth table) to perform some operation.
- The result is available in symbolic form and is ascertained by a *counting* process.
- Precision is limited only by the number of discrete states representable by the physical quantities used.

Digital Computation

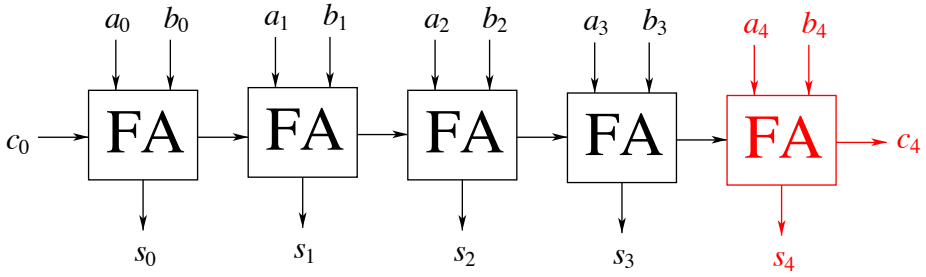


Analog vs. Digital: The Cost of Precision

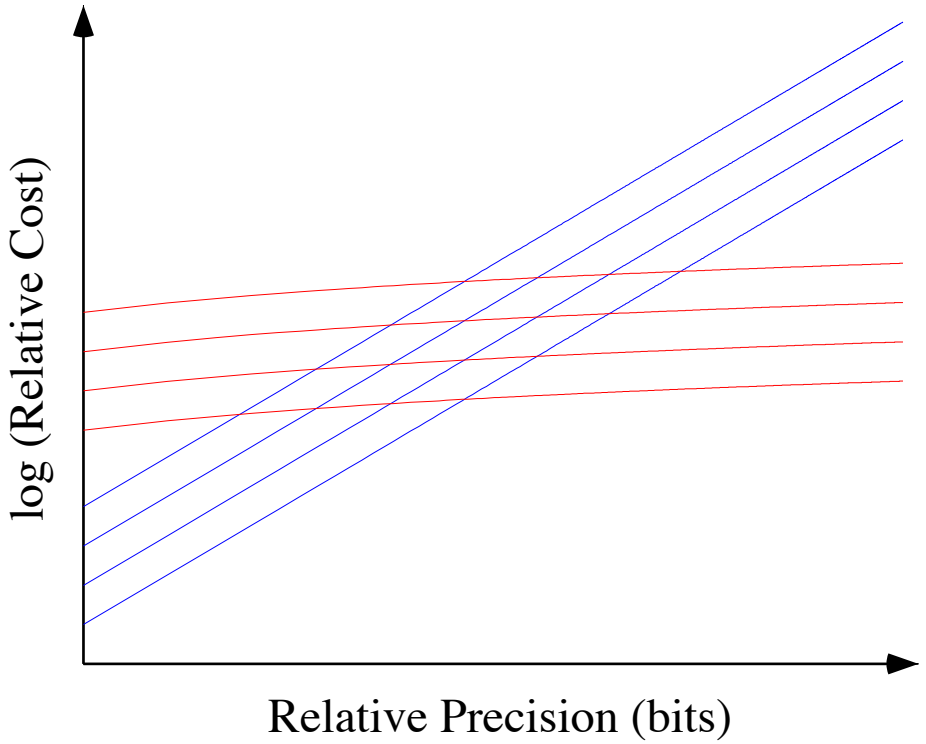
- Analog: Cost grows as a power law of precision.



- Digital: Cost grows as the logarithm of precision.

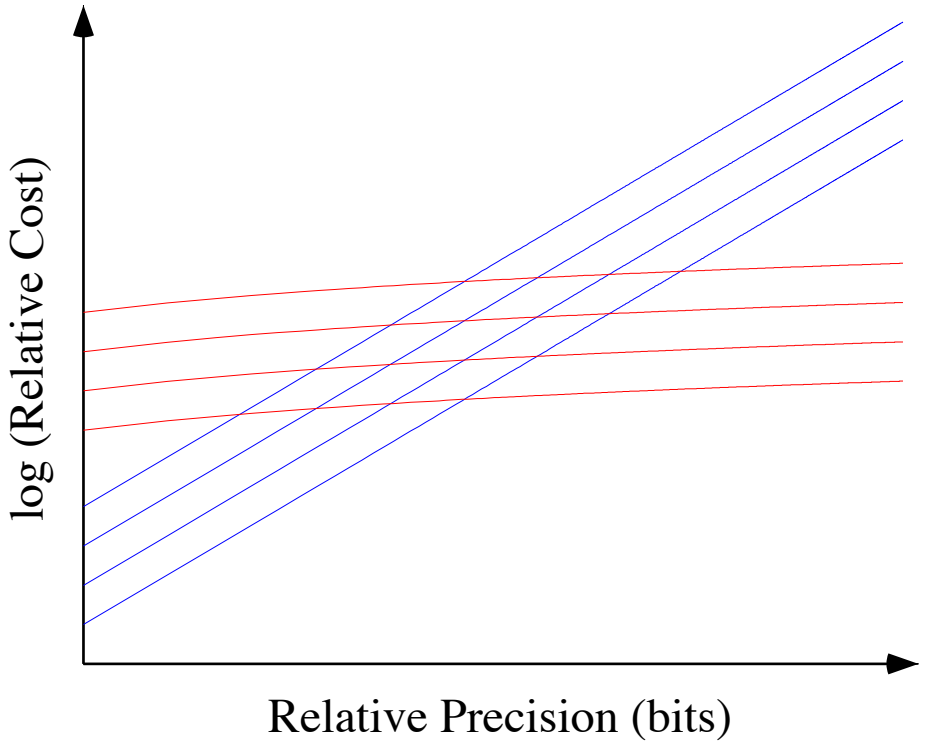


Analog vs. Digital: The Cost of Precision



- Analog: Low initial cost, high marginal cost.
- Digital: High initial cost, low marginal cost.

Analog vs. Digital: The Cost of Precision



⇒ When precision requirements are low, Analog can be very cost effective, but when precision requirements are high, Digital is your best bet...

Analog vs. Digital: The "Real" Cost (i.e., $$)

- Analog:
  - Full-custom designs
  - Functionality tied to device details
  - Not many high-level behavioral abstractions
  - Few EDA tools available
  - Not-so-rapid prototyping
  - Few devices, relatively long design time

- Digital:
  - Synthesized or semi-custom designs
  - Functionality not critically dependent on device details
  - Good high-level behavioral abstractions
  - Many EDA tools available
  - Rapid prototyping (e.g., FPGAs)
  - Many devices, relatively short design time