# Introduction to Digital Computers
# Talk #2

Dr. Guy Even

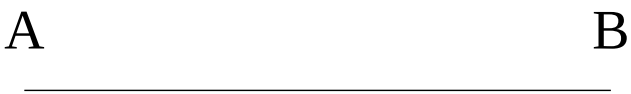
March 14, 2002

# Today's agenda

- The digital abstraction: noise models
- The zero noise model
  - Digital interpretation of analog signals
  - Definition of gates
- Bounded-noise model
  - Digital interpretation of analog signals
  - Definition of gates
- Combinational circuits
  - Gates as implementations of Boolean functions
  - Building blocks: gates and wires
  - Nets
  - Syntactic definition

# Noise models

A wire connecting points A and B:




**zero-noise model:** $\forall t \ : \ B(t) = A(t)$

**Noise signal:** models noise that wire accumulates

$$n(t) \stackrel{\triangle}{=} B(t) - A(t)$$

**bounded-noise model:** $\forall t \ : \ |n_B(t)| \leq \epsilon$

**uniform bounded-noise model:** same bound $\epsilon$ applies to all the noise signals in the circuit.

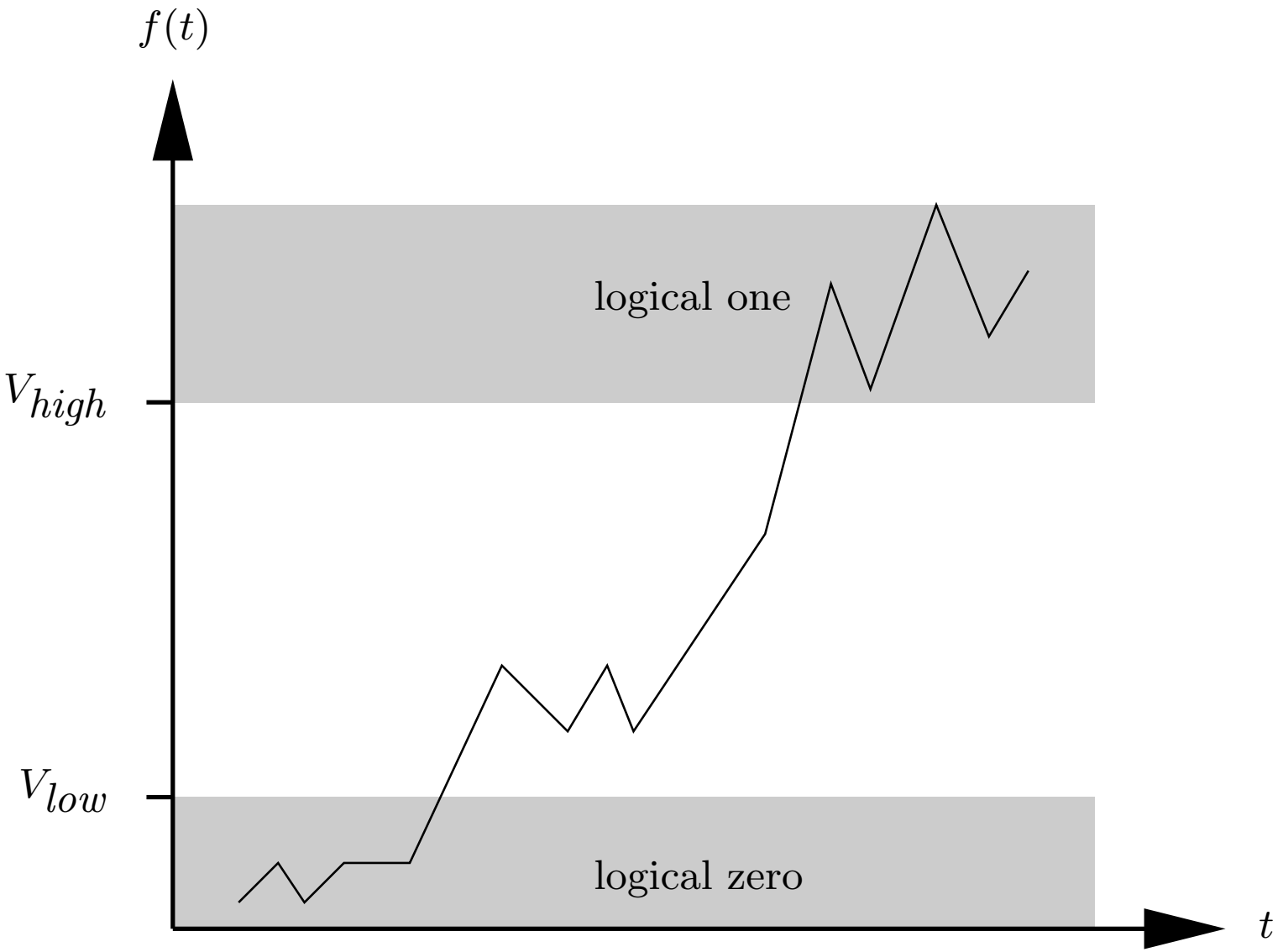
# zero-noise model

## Interpretation of analog signals as digital signals

Two voltage thresholds are defined: $V_{low} < V_{high}$.

Consider an analog signal $f(t)$. The digital signal $dig(f(t))$ is defined as follows.

$$dig(f(t)) \triangleq \begin{cases} 0 & \text{if } f(t) < V_{low} \\ 1 & \text{if } f(t) > V_{high} \\ \text{non-logical} & \text{otherwise.} \end{cases}$$


$f(t)$
logical one
$V_{high}$
$V_{low}$
logical zero
$t$

## zero-noise model: def. of inverter

**Definition 1 (inverter in zero-noise model)** *A gate $G$ with a single input $x$ and a single output $y$ is an inverter if its static transfer function $f(z)$ satisfies the following the following two conditions:*

1. *If* $z < V_{\text{low}}$, *then* $f(z) > V_{\text{high}}$.
2. *If* $z > V_{\text{high}}$, *then* $f(z) < V_{\text{low}}$.

*Equivalently:*

1. *If* $\text{dig}(z) = 0$, *then* $\text{dig}(f(z)) = 1$.
2. *If* $\text{dig}(z) = 1$, *then* $\text{dig}(f(z)) = 0$.

## Questions:

- Define a NAND-gate in the zero-noise model.
- Define a gate that implements a Boolean function $f : \{0,1\}^2 \to \{0,1\}$ in the zero-noise model.

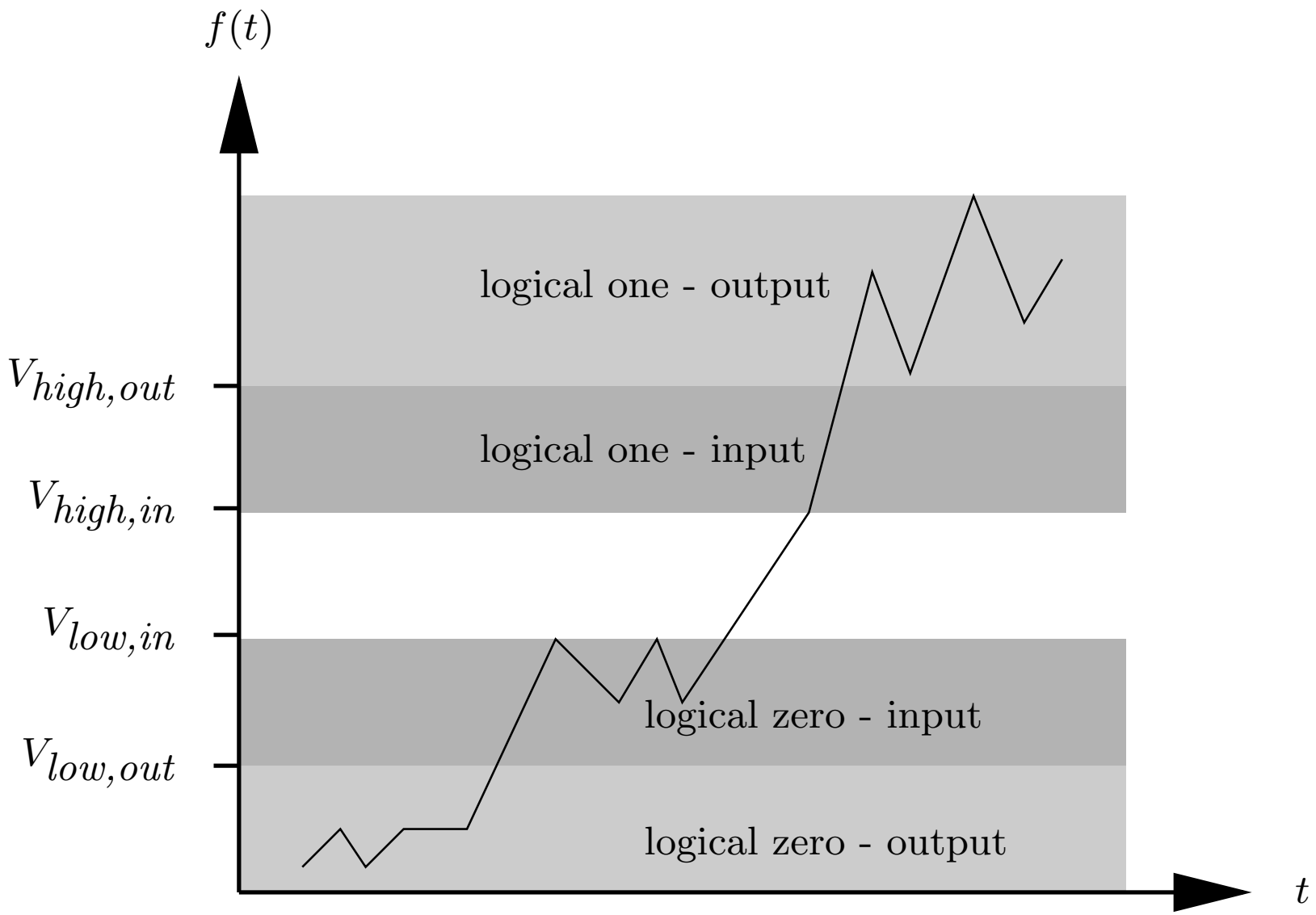
**bounded-noise model: analog signals $\Longrightarrow$ digital signals**

Four voltage thresholds are defined:

$$V_{low,out} < V_{low,in} < V_{high,in} < V_{high,out}$$

**Noise-margin:**

$$\min\{V_{low,in} - V_{low,out}, V_{high,out} - V_{high,in}\}$$


$f(t)$
logical one - output
$V_{high,out}$
logical one - input
$V_{high,in}$
$V_{low,in}$
logical zero - input
$V_{low,out}$
logical zero - output
$t$

**bounded-noise model: analog signals $\Longrightarrow$ digital signals**

Consider an input signal $f_{in}(t)$. The digital signal $dig(f_{in}(t))$ is defined as follows.

$$dig(f_{in}(t)) \triangleq \begin{cases} 0 & \text{if } f_{in}(t) < V_{low,in} \\ 1 & \text{if } f_{in}(t) > V_{high,in} \\ \text{non-logical} & \text{otherwise.} \end{cases} \quad (1)$$

**bounded-noise model: analog signals $\Longrightarrow$ digital signals**

Consider an output signal $f_{out}(t)$. The digital signal $dig(f_{out}(t))$ is defined analogously.

$$dig(f_{out}(t)) \triangleq \begin{cases} 0 & \text{if } f_{out}(t) < V_{low,out} \\ 1 & \text{if } f_{out}(t) > V_{high,out} \\ \text{non-logical} & \text{otherwise.} \end{cases} \quad (2)$$

## bounded-noise model: def. of inverter

**Definition 2 (inverter in the bounded-noise model)** *A gate $G$ with a single input $x$ and a single output $y$ is an inverter if its static transfer function $f(z)$ satisfies the following the following two conditions:*

1. *If $z < V_{\text{low,in}}$, then $f(z) > V_{\text{high,out}}$.*
2. *If $z > V_{\text{high,in}}$, then $f(z) < V_{\text{low,out}}$.*

## Questions:

- Define a NAND-gate in the bounded-noise model.
- Define a gate that implements a Boolean function $f : \{0,1\}^2 \to \{0,1\}$ in the bounded-noise model.

## Stable signals

- An analog signal $f(t)$ is logical at time $t$ if $dig(f(t)) \in \{0, 1\}$.
- An analog signal $f(t)$ is stable during the interval $[t_1, t_2]$ if $f(t)$ is logical for every $t \in [t_1, t_2]$.
- Continuity of $f(t)$ and $V_{low} < V_{high} \Longrightarrow$

**Claim 1** *If an analog signal $f(t)$ is stable during the interval $[t_1, t_2]$ then one of the following holds:*

1. $\mathrm{dig}(f(t)) = 0$, *for every* $t \in [t_1, t_2]$, *or*
2. $\mathrm{dig}(f(t)) = 1$, *for every* $t \in [t_1, t_2]$.

# Combinational circuits

Our goal is to prove two theorems:

(a) Every Boolean function can be implemented by a combinational circuit (very easy).

(b) Every combinational circuit implements a Boolean function.

Bonuses:

- Simulate a combinational circuit in linear time.
- Compute the propagation delay of a combinational circuit in linear time.

# Boolean functions

Let $\{0,1\}^n$ denote the set of $n$-bit strings.

A Boolean function is defined as follows.

**Definition 3** *A function $f : \{0,1\}^n \to \{0,1\}^k$ is called a Boolean function.*

## Consistent state of gates

Terminology: inputs and outputs of a gate are called *terminals*, *ports*, or even *pins*.

Simplification: consider a gate $G$ with 2 inputs, denoted by $x_1, x_2$, and a single output, denoted by $y$. We denote the digital signal at terminal $x_1$ by $x_1(t)$.

**Definition 4** *A gate $G$ is consistent with a Boolean function $f$ at time $t$ if the input values are digital at time $t$, and*

$$y(t) = f(x_1(t), x_2(t)).$$

## Gates as implementations of Boolean functions

The propagation delay of a gate $G$ is the amount of time that elapses till $G$ becomes consistent.

**Definition 5** *A gate $G$ implements a Boolean function $f : \{0,1\}^2 \to \{0,1\}$ with propagation delay $t_{pd}$ if the following holds.*

*For every $\sigma_1, \sigma_2 \in \{0,1\}$, if $x_i(t) = \sigma_i$, for $i = 1, 2$, during the interval $[t_1, t_2]$, then*

$$y(t) = f(\sigma_1, \sigma_2),$$

*for every $t \in [t_1 + t_{pd}, t_2]$.*

## Remarks on implementation of a Boolean function by a gate

**Compact form:** A gate $G$ implements a Boolean function $f : \{0,1\}^n \to \{0,1\}$ with propagation delay $t_{pd}$ if

stability of the inputs of $G$ in the interval $[t_1, t_2]$

$\Longrightarrow$

the gate $G$ is consistent with $f$ in the interval $[t_1 + t_{pd}, t_2]$.

## Remarks on implementation of a Boolean function by a gate

**What if $t_2 - t_1 < t_{pd}$?** Then gate does not have enough time to become consistent. Interval $[t_1 + t_{pd}, t_2]$ in definition is empty...

## Remarks on implementation of a Boolean function by a gate

**Contamination delay:** Consider a gate $G$ that is consistent with a Boolean function $f$. The contamination delay the amount of time that the outputs of $G$ retain their values after the inputs of $G$ become non-digital.

"Pessimistic" assumption: contamination delay is zero. (i.e. gates can notice instantly that an input becomes non-digital.)

## Remarks on implementation of a Boolean function by a gate

$t_{pd}$ is an upper bound on the time it takes to obtain consistency.

(If we buy a 2GHz microprocessor and operate it with a 1GHz clock, it should work. What about the converse?)

## Building blocks of combinational circuits

- gates
- nets (generalization of wires)

# Gates

**Fan-in:** the number of inputs of a gate.

We will restrict the fan-in of basic gates to at most $2-3$ input ports.

The basic gates that we consider are: inverter (NOT-gate), OR-gate, NOR-gate, AND-gate, NAND-gate, XOR-gate, NXOR-gate, multiplexer (MUX).

# Wires & Nets

Wires connect points to each other. Very often we need to connect several terminals (i.e. inputs and outputs of gates) together.

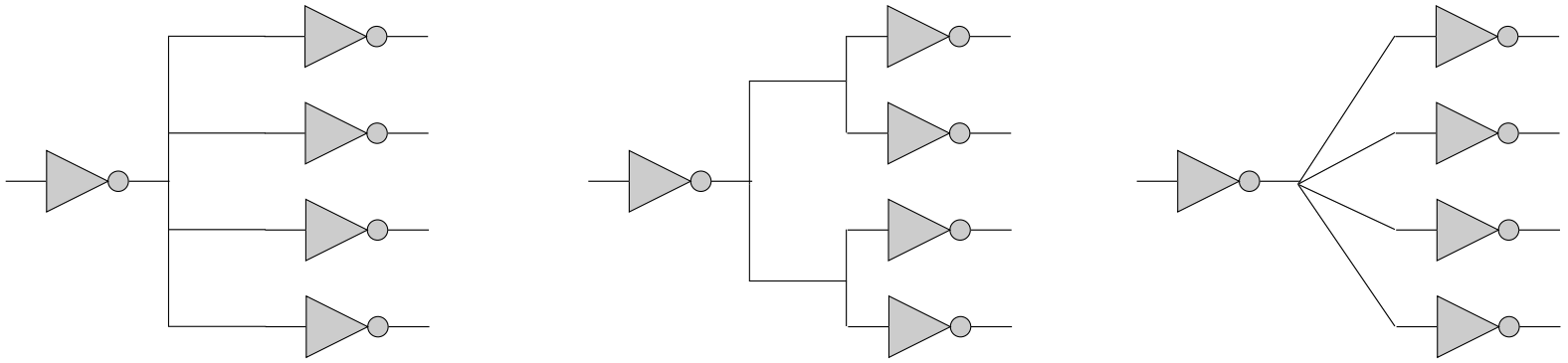
Ignore how connections are actually made.

A net is a subset terminals that are connected by wires. In the digital abstraction we assume that the signals all over a net are identical (why?).

The fan-out of a net $N$ is the number of terminals that are connected by $N$.

# Drawing nets

Three different drawings of the same net. We may draw a net in any way that we find convenient or aesthetic. The interpretation of the drawing is that terminals that are connected by lines or curves constitute a net.



# Digital signals for nets

We would like to define the digital signal $N(t)$ for a whole net $N$.

Noise creates different analog signals along the net.

Define $N(t)$ to logical only if there is a consensus among all the digital interpretations of analog signals at different terminals of the net.

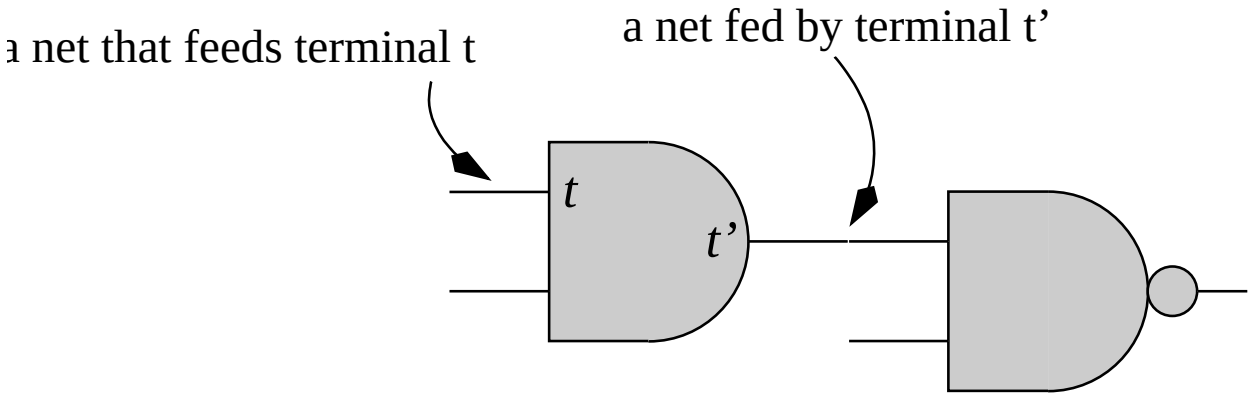
In other words:

- $N(t)$ is zero if the digital values of all the analog signals along the net are zero.
- $N(t)$ is one if the digital values of all the analog signals along the net are one.
- If there is no consensus, then $N(t)$ is non-logical.

# Directions in nets

A net $N$ feeds an input terminal $t$ if the input terminal $t$ in $N$.

A net $N$ is fed by an output terminal $t$ if $t$ is in $N$.




Information is "supplied" by output terminals and is "consumed" by input terminals. From an electronic point of view, output terminals are connected via resistors either the the ground or the the power. Input terminals are connected only to capacitors.

# Simple nets

**Definition 6** *A net $N$ is simple if (a) $N$ is fed by exactly one output terminal, and (b) $N$ feeds at least one input terminal.*

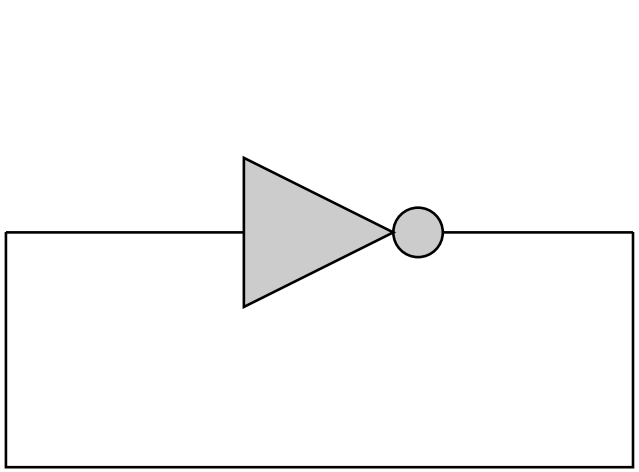
A simple net $N = \{t, t_1, t_2, \ldots, t_k\}$, where $t$ is an output terminal, and $\{t_i\}_{i=1}^{k}$ are input terminals can be modeled by wires $\{w_i\}_{i \in I}$. Each wire $w_i$ connects $t$ and $t_i$. We may regard each wire $w_i$ as a directed edge $t \rightarrow t_i$.

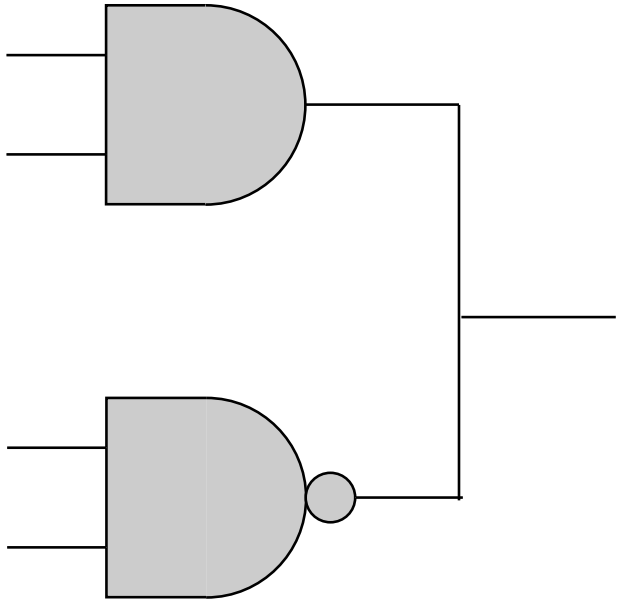
**Directed graph model of a circuit.** If all the nets are simple, then circuit may be modeled by a directed graph.

- vertices = gates
- directed edges = wires

If all the nets in $C$ are simple, then we denote the directed graph that models a circuit $C$ by $DG(C)$.

# Are these combinational circuits?





## Input gates & output gates

Input and output gates model communication with the "external world".

Input Gate

Output Gate

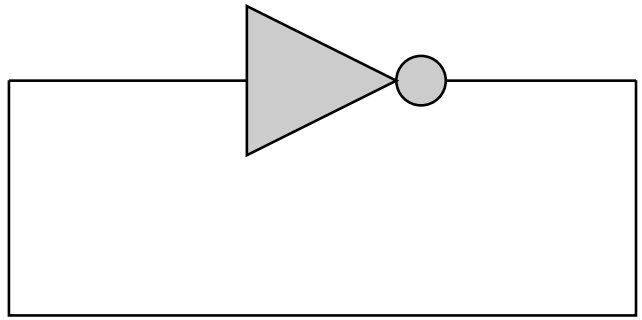
## Syntactic definition of combination circuits

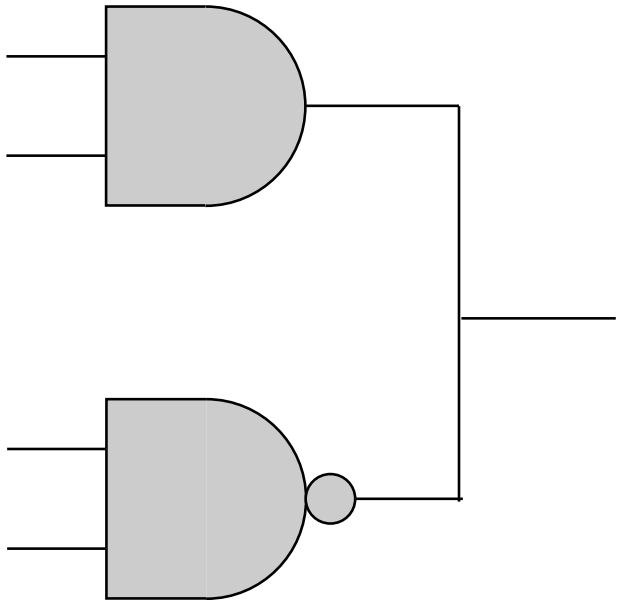
**Definition 7** *A combinational circuit is a pair* $C = \langle \mathcal{G}, \mathcal{N} \rangle$ *that satisfies the following conditions:*

1. $\mathcal{G}$ *is a set of gates.*
2. $\mathcal{N}$ *is a set nets over terminals of gates in* $\mathcal{G}$*.*
3. *Every terminal* $t$ *of a gate* $G \in \mathcal{G}$ *belongs to exactly one net* $N \in \mathcal{N}$*.*
4. *Every net* $N \in \mathcal{N}$ *is simple.*
5. *The directed graph* $DG(C)$ *is acyclic.*

# Back to the bad circuits

Which conditions are violated by these circuits?





## Combinational circuits: Syntax $\Rightarrow$ Semantics

**Completeness:** for every Boolean function $f$, there exists a combinational circuit that implements $f$.

**Soundness:** every combinational circuit implements a Boolean function. (NP-Complete to compute the Boolean function that is implemented by a given combinational circuit.)

**Simulation:** given the digital values of the inputs of a combinational circuit, one can simulate the circuit in linear time.

**Delay analysis:** given the propagation delays of all the gates in a combinational circuit, one can compute in linear time the propagation delay of the circuit.

## Simulation theorem of combinational circuits

**Theorem 2** *Let $C = \langle \mathcal{G}, \mathcal{N} \rangle$ denote a combinational circuit that contains $k$ input gates. Let $\{x_i\}_{i=1}^{k}$ denote the output terminals of the input gates in $C$. Assume that the digital signals $\{x_i(t)\}_{i=1}^{k}$ are stable during the interval $[t_1, t_2]$. Then, for every net $N \in \mathcal{N}$ there exist:*

1. *a Boolean function $f_N : \{0,1\}^k \rightarrow \{0,1\}$, and*
2. *a propagation delay $t_{pd}(N)$*

*such that*

$$N(t) = f_N(x_1(t), x_2(t), \ldots, x_k(t)),$$

*for every $t \in [t_1 + t_{pd}(N), t_2]$.*