EECS 42 Introduction to Electronics for Computer Science

Andrew R. Neureuther

Lecture # 12 Physical Limits of Logic

- Timing Diagrams
- capacitance Loading

http://inst.EECS.Berkeley.EDU/~ee42/

Game Plan 03/10/03

Monday 03/10/04

- ❑ Monday: Physical Limits of Logic (11.3, 11.4 pp 403-422)

Wednesday 03/12/03:

- ❑ Dependent Sources (4.1 and 4.3)

Next (9th) Week:

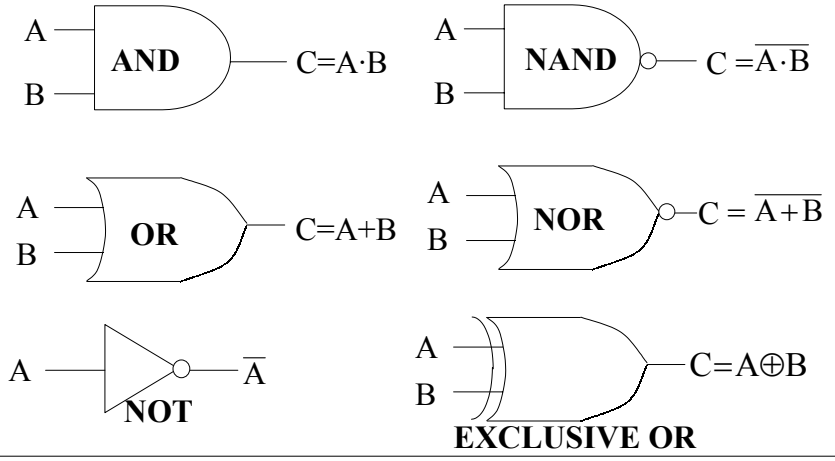
- ❑ Monday: Circuit analysis with dependent sources (4.1-4.3)
- ❑ Wednesday: Comparators and op-amps (handout)

Problem set #6: out Monday 3/3 and due at 2:30 3/10 in box in 240 Cory – logic functions, truth tables, synthesis, timing diagram

Problem set #7: out Monday 3/10 and due at 2:30 3/10 in box in 240 Cory – timing diagram, ca

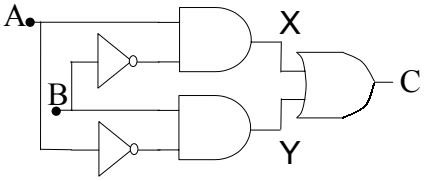
Logic Gates

These are circuits that accomplish a given logic function such as “OR”. We will shortly see how such circuits are constructed. Each of the basic logic gates has a unique symbol, and there are several additional logic gates that are regarded as important enough to have their own symbol. The set is: AND, OR, NOT, NAND, NOR, and EXCLUSIVE OR.



Logic Circuits

With a combination of logic gates we can construct any logic function. In these two examples we will find the truth table for the circuit.



It is helpful to list the intermediate logic values (at the input to the OR gate). Let’s call them X and Y.

Now we complete the truth tables for X and Y, and from that for C. (Note that $X = A \bullet \overline{B}$ and $Y = B \bullet \overline{A}$ and finally C = X + Y)

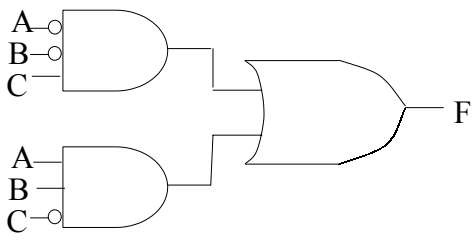
A	B	X	Y	C
0	0	0	0	0
0	1	0	1	1
1	0	1	0	1
1	1	0	0	0

Interestingly, this is the same truth table as the EXCLUSIVE OR

How to Combine Gate to Produce a Desired Logic Function? (More basic Logical Synthesis)

Example:

A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0



$$F=\overline{A}\,\overline{B}C+AB\overline{C}$$

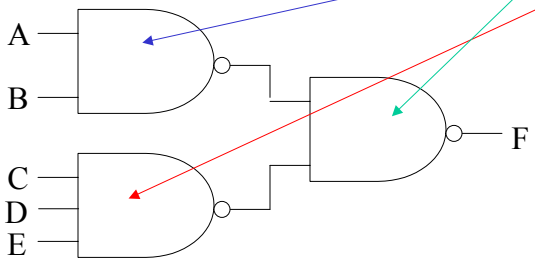
Logical Synthesis Guided by DeMorgan's Theorem

DeMorgan's Theorem :

$$A+B+C=\overline{\left[\overline{A}\,\overline{B}\,\overline{C}\right]} \quad \text{or} \quad \overline{A}+\overline{B}+\overline{C}=\overline{\left[A\,B\,C\right]}$$

Example of Using DeMorgan's Theorem:

$$F=A\bullet B+C\bullet D\bullet E=\overline{AB}\bullet\overline{CDE}$$



Thus any sum of products expression can be immediately synthesized from NAND gates alone

What Are Some Limitations of Digital?

It takes a lot of bits to represent even simple audio signals and if we convert a real-time audio signal to digital, we need to transmit a lot of bits every second.

Example: An ordinary audio signal is sampled every 50μs (to achieve 10KHz frequency performance) and evaluated (converted to digital form) to an accuracy of 1 part in 10,000. Every sample requires how many bits?

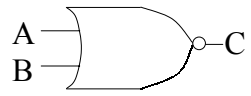
2^{13} = 8,192 and 2^{14} = 16,384 so we need 14 bits for each sample. Now we need to transmit these bits 20,000 times per second. The bit rate is 280,000 baud!}

The transmission of digital signals as pulses through circuits and over transmission media leads to pulse degradation, just as analog signals are degraded. As a consequence, we must (a) detect and regenerate the signal before it gets "buried in the noise," and (b) accept that propagation delays are intrinsic to signal flow, and take these delays into account in our design (e.g., to avoid making a control decision based on a piece of information that has not yet arrived!). The RC transient lectures treated pulse degradation in real circuits quantitatively.

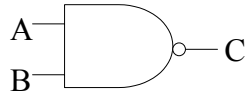
PHYSICAL LIMITATIONS OF LOGIC GATES

- Computer Datapath: Boolean algebraic functions using binary variables
- Symbolic representation of functions using logic gates

NOR $\overline{A+B}=C \rightarrow$



NAND $\overline{AB}=C \rightarrow$



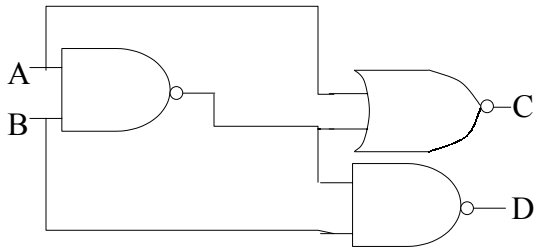
However:

- Every node has capacitance and interconnects have resistance. It takes time to charge these capacitances.

  Thus, output of all circuits, including logic gates is **delayed** from input.

PHYSICAL LIMITATIONS OF LOGIC GATES

Computer Datapath: Connected logic gates



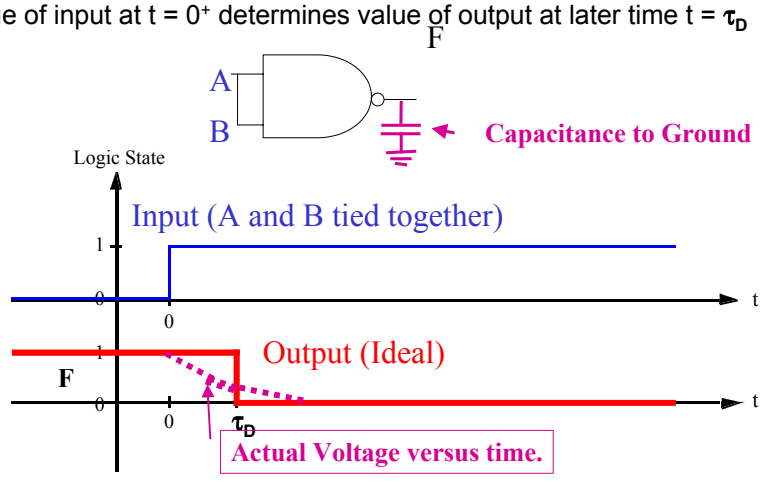
Every node in any circuit (such as the internal circuit of a NAND gate) has capacitance and all interconnects have resistance. Thus it takes time to charge these capacitances.

Thus, output of all circuits, including logic gates is **delayed** from input.

LOGIC GATE DELAY τ_D

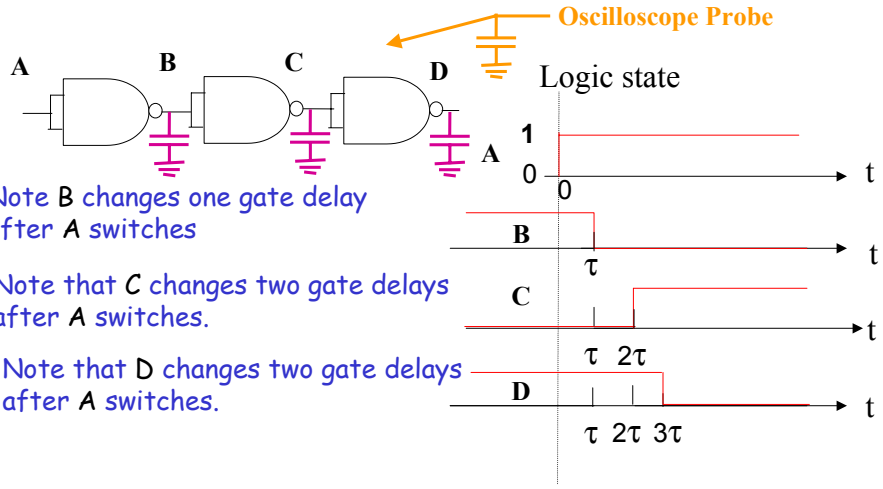
Time delay τ_D occurs between input and output: "computation" is not instantaneous

Value of input at t = 0^+ determines value of output at later time t = τ_D



SIGNAL DELAY: TIMING DIAGRAMS

Show transitions of variables vs time



Note B changes one gate delay after A switches

Note that C changes two gate delays after A switches.

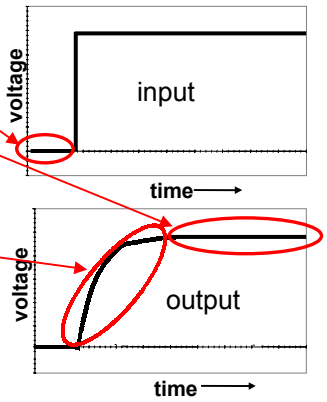
Note that D changes two gate delays after A switches.

Simplification for time behavior of RC Circuits

Before any input change occurs we have a dc circuit problem (that is we can use dc circuit analysis to relate the output to the input).

Long after the input change occurs things "settle down" …. Nothing is changing …. So again we have a dc circuit problem.

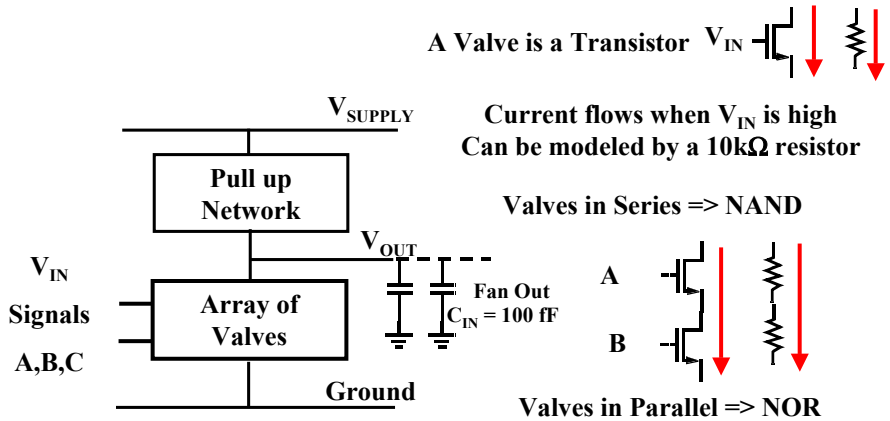
We call the time period during which the output changes the *transient*



We can predict a lot about the transient behavior from the pre- and post-transient dc solutions

Logic Gates – How are they built in practice?

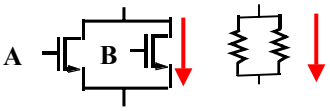
A Valve is a Transistor V_{IN}



Current flows when V_{IN} is high
Can be modeled by a 10kΩ resistor

Valves in Series => NAND

Valves in Parallel => NOR

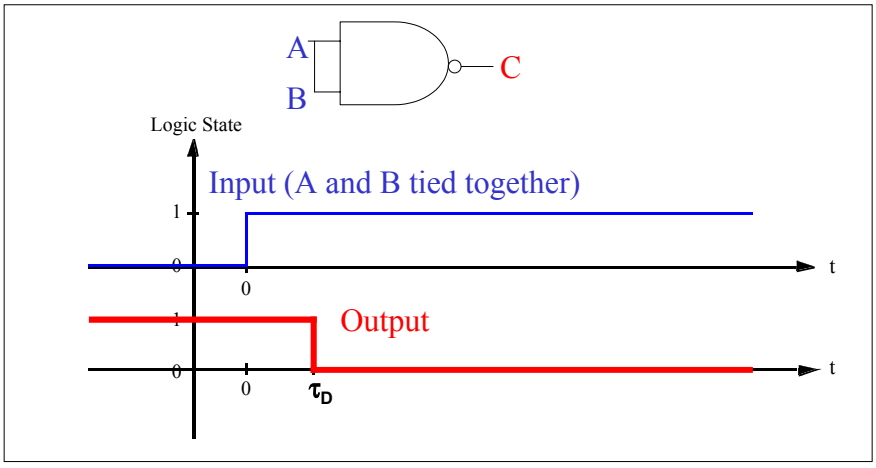


(You can learn about building gates in EE 141.)

UNIT GATE DELAY τ_D

Time delay τ_D occurs between input and output: "computation" is not instantaneous

Value of input at t = 0^+ determines value of output at later time t = τ_D



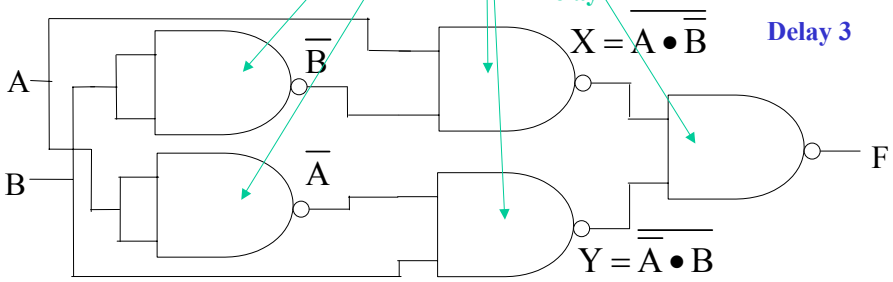
Logical Synthesis of XOR

A	B	F
0	0	0
0	1	1
1	0	1
1	1	0

Inputs have different delays, but we ascribe a single worst-case delay τ to every gate

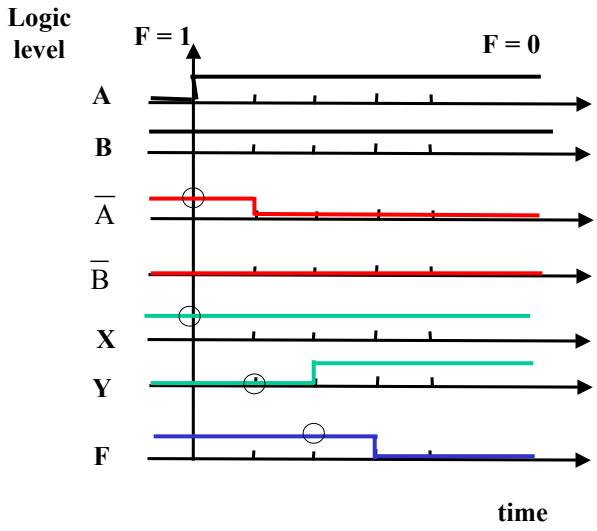
$$F = A \bullet \overline{B} + \overline{A} \bullet B$$

We Need a Timing Diagram!



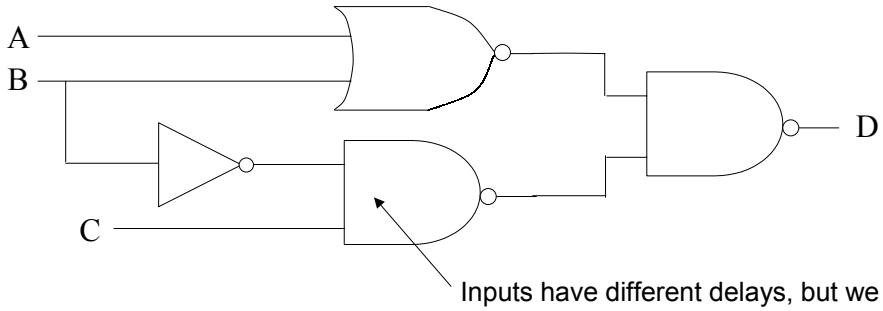
EECS 42 Intro. electronics for CS Spring 2003 Lecture 12: 03/10/03 A.R. Neureuther
Version Date 03/09/03

Timing Diagram for Delays in Logic



EFFECT OF GATE DELAY

Cascade of Logic Gates



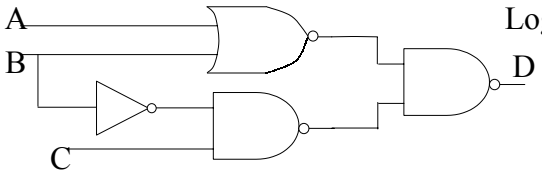
How many "gate delays for shortest path? ANSWER : 2

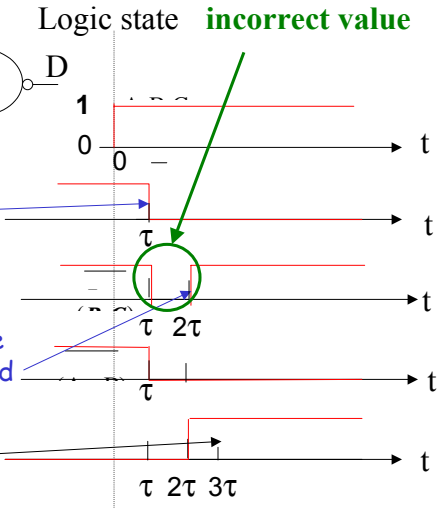
How many gate delays for longest path? ANSWER : 3

TIMING DIAGRAMS

Show transitions of variables vs time

Glitching: temporary switching to an incorrect value





Note $\overline{\ }$ becomes valid one gate delay after B switches

Note that $(\overline{B} \cdot C)$ becomes valid two gate delays after B&C switch, because the invert function takes one delay and the NAND function a second.

No change at t = 3τ