



DEPARTMENT OF INFORMATION SYSTEMS AND COMPUTER SCIENCE



```
0010111010100011101011110010011101010101001000101
1101010110101010000101010101001010101010101010
10100101001001001010101010101010101010101010101010
11100001111010110000000111101010101010000010101
111010101110010100010010111010100010100100111010
10101001010010010010000101010110101010101010010111
0010101001010100101010000001010101001111101000011001
1000110010000111100110101011000100110101010000101010
1100101010101000010011001010100010010101010101010
10100101001001001010101010101010101010101010101010
11100001111010110000000111101010101010000010101
0010010101001010010010100100010101010101001010010
10010100100001010100100101010010100101010010010
1001010010101001010010101001010010101001001001001
100101010101001010101010100101010101010010101010
```

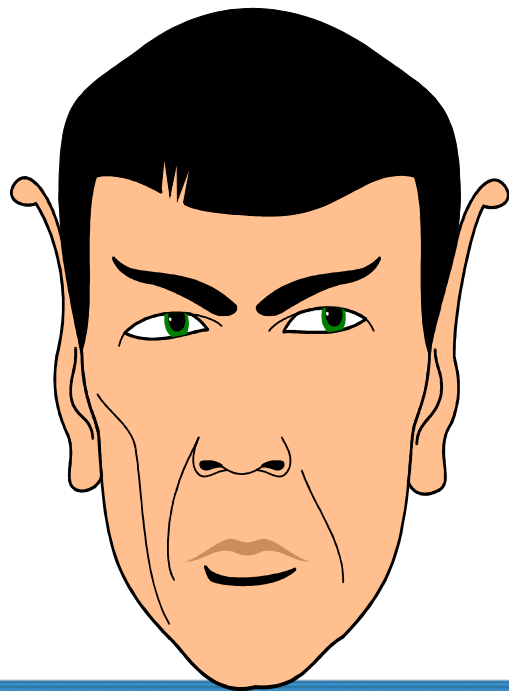
										01
										02
										03
										04
										05
A	B	C	D	E	F	G	H			

More CMOS Gates

And a Review of Digital Logic



+

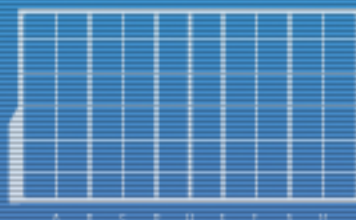


=



?

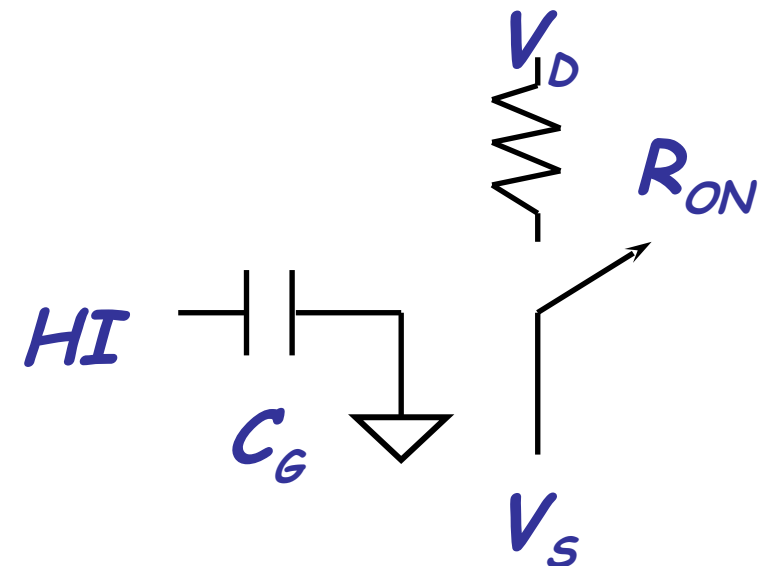
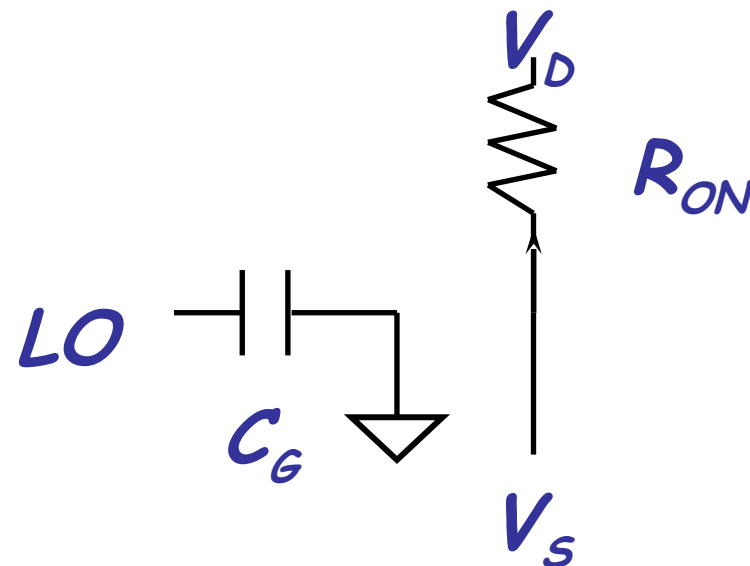
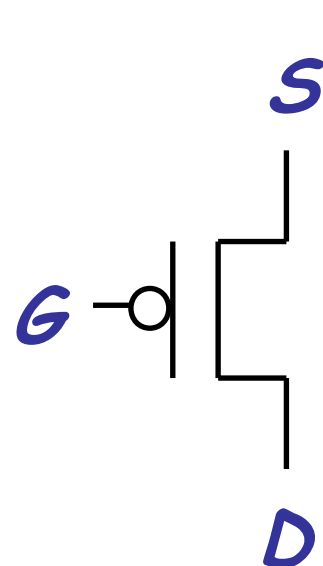
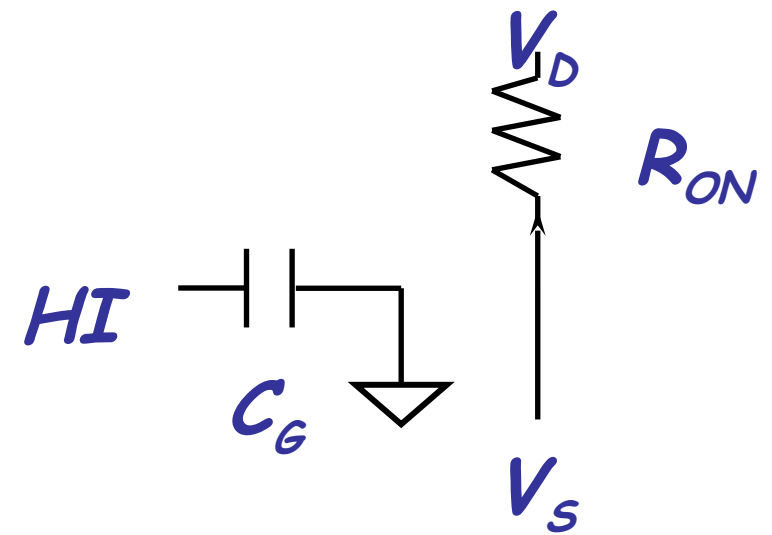
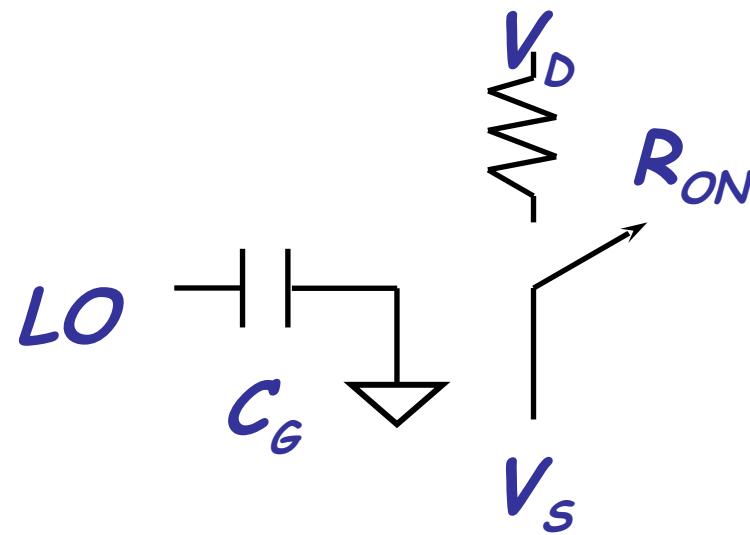
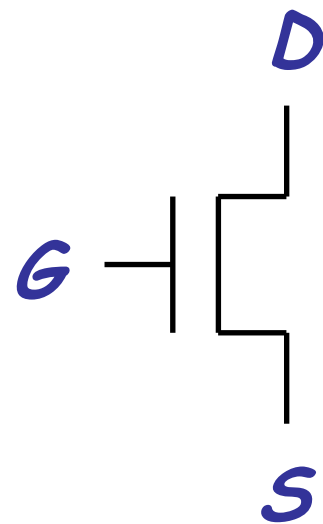
0010101001010100011110100001100
10001100100001111001101010010101
110010101010100001001100101010100
1001010010010010101010101010101
11100001111010110000000111101001
001001010100101001001010010010110
10010100100001010100100101001010
10010100101010010100101010010101
10010101010101010101010101010101



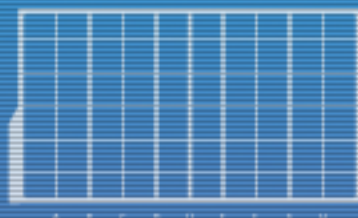
DISCS

More Transistor Limitations

► Resistance (R_{on}) and Capacitance (C_g)

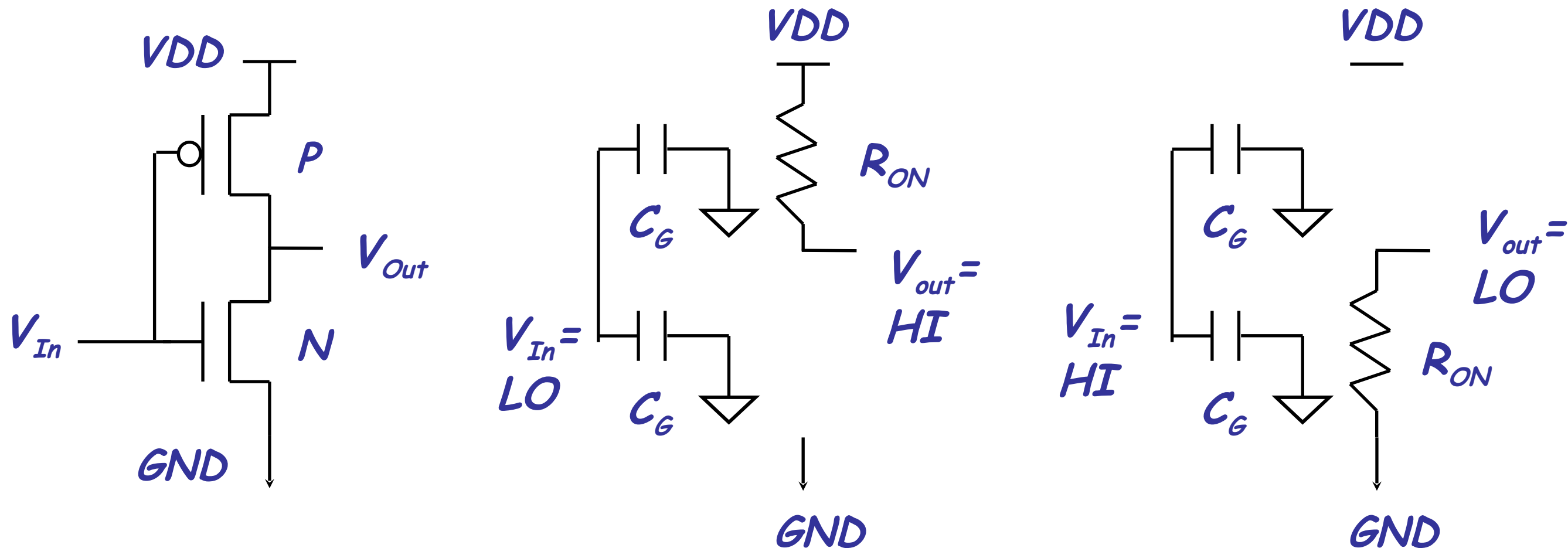


00101010010101000011110100001100
10001100100001111001101010010101
11001010101010100001001100101010100
100101001001001010101010101010101
11100001111010110000000111101001
00100101010010100100100100100110
10010100100001010100100101001010
10010100101010010100101010010101
100101010101010010101010010101



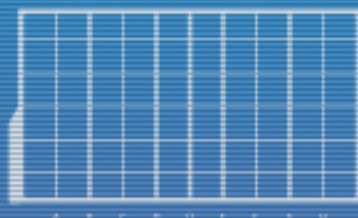
DISCS

Delays: CMOS Inverter

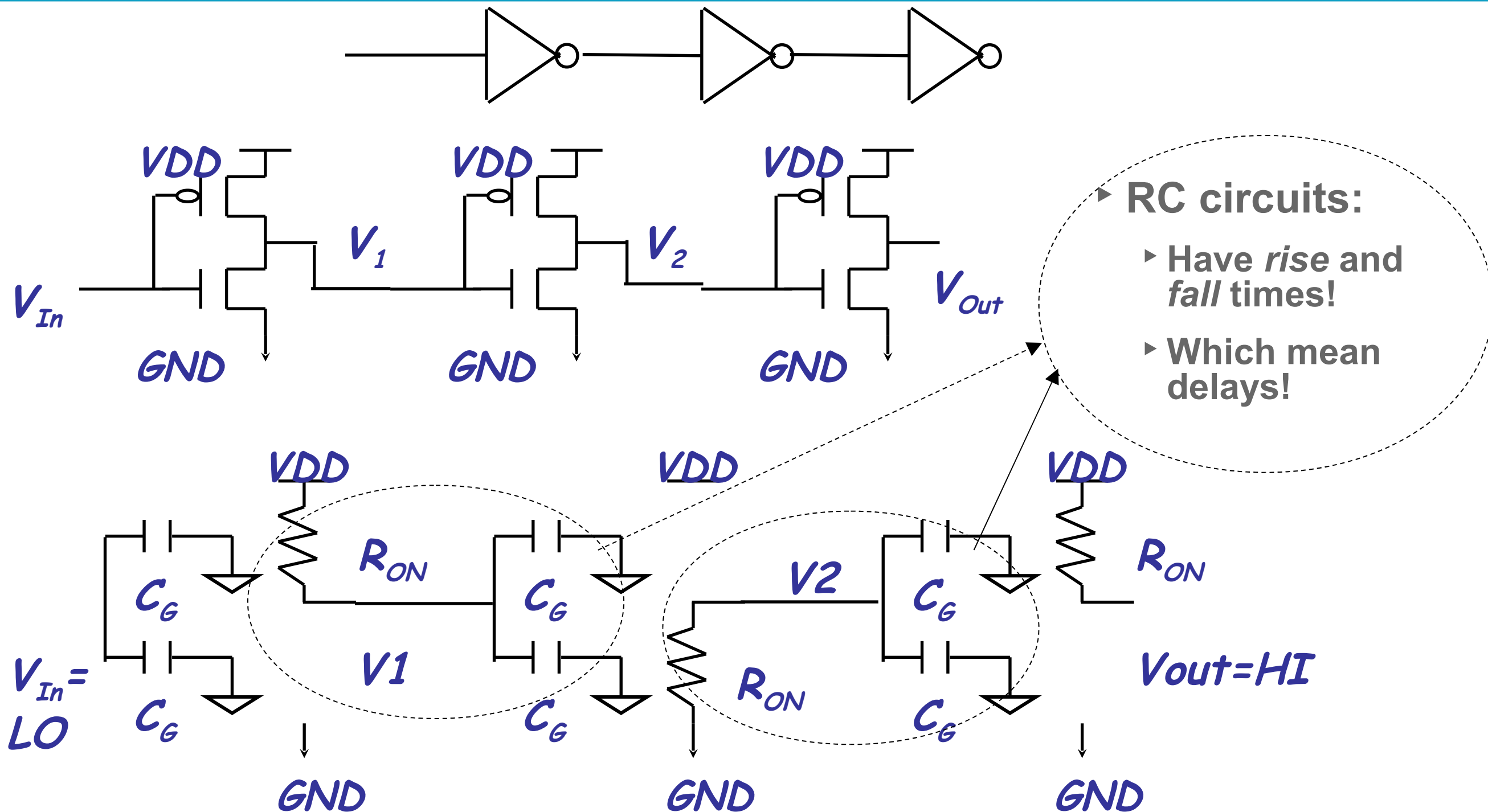


- Logic Gates have an Output Resistance and Input Capacitance.
- Form an RC circuit when chained together!

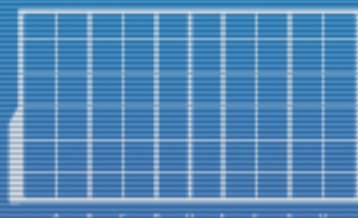
00101010010101000011110100001100
10001100100001111001101010010101
110010101010100001001100101010100
1001010010010010101010101010101
11100001111010110000000111101001
001001010100101001001010010010110
10010100100001010100100101001010
10010100101010010100101010010101
10010101010101010101010101010101



Example: Inverter Chain

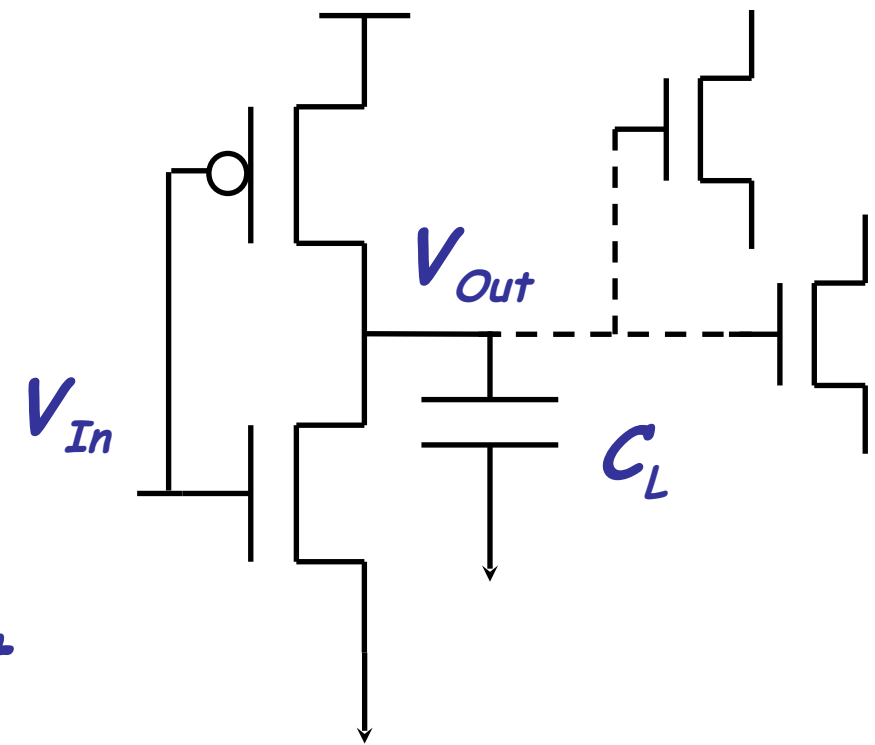
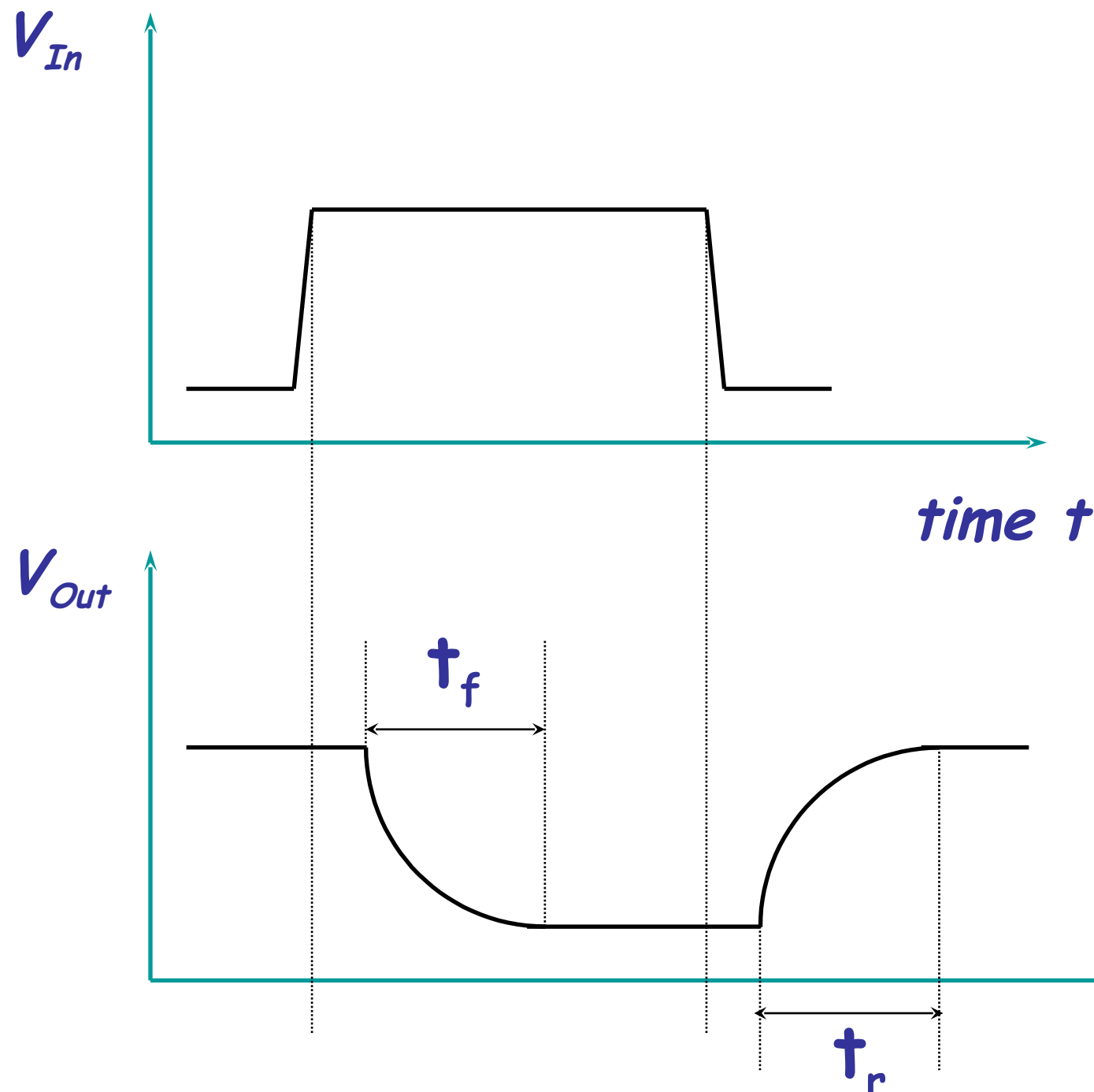


00101010010101000011110100001100
10001100100001111001101010010101
11001010101010100001001100101010100
100101001001001010101010101010101
11100001111010110000000111101001
00100101010010100100100100100110
10010100100001010100100101001010
10010100101010010100101010010101
10010100101010010100101010010101



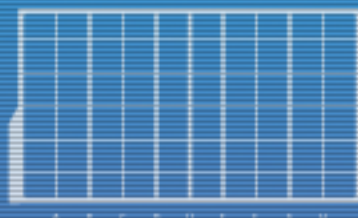
DISCS

Delays: Rise and Fall Time



Time constant
of t_f , t_r equals
 $R_{ON} C_L$

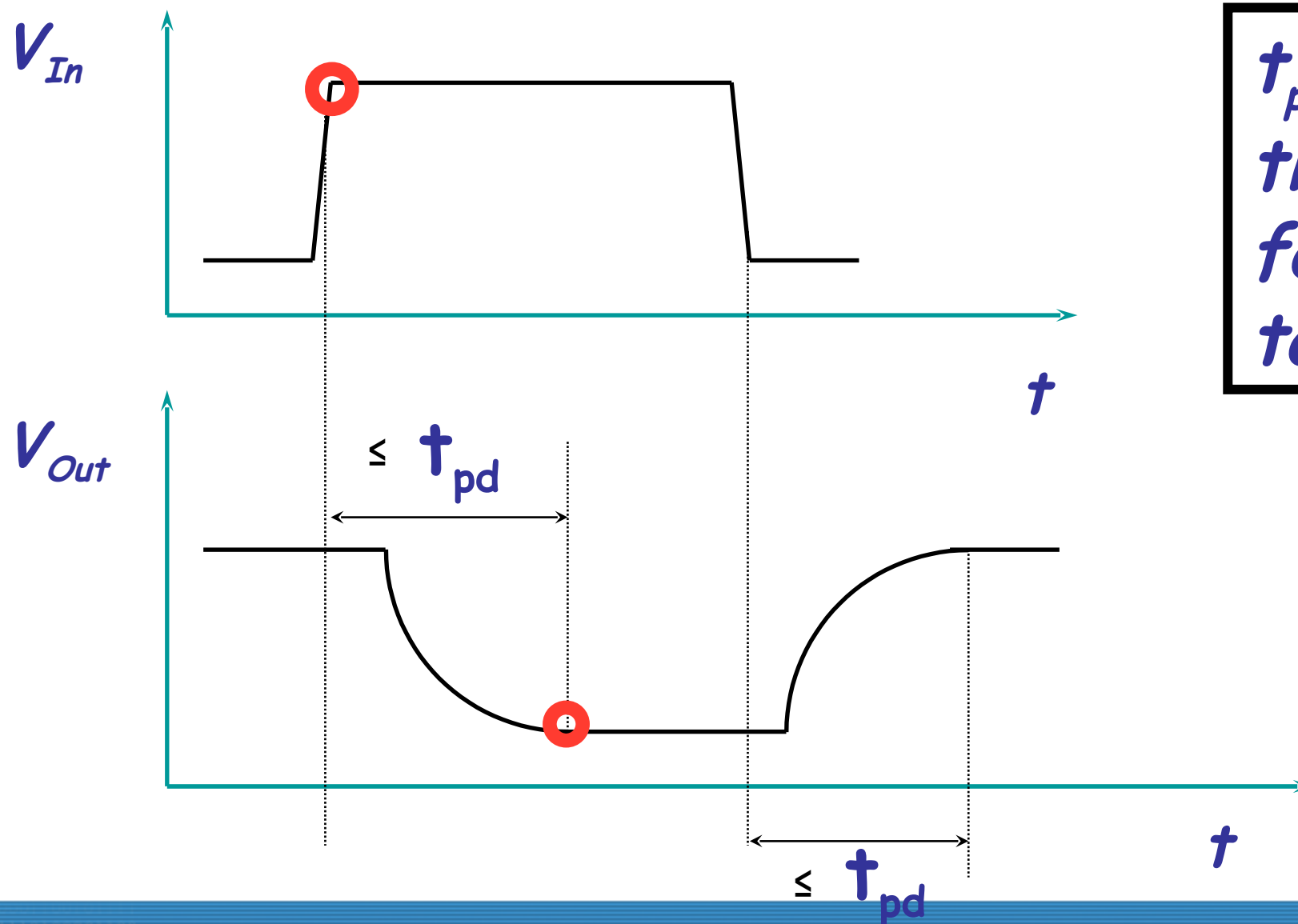
0010101001010100011110100001100
10001100100001111001101010010101
110010101010101000010011001010100
1001010010010010101010101010101
11100001111010110000000111101001
001001010100101001001010010010110
10010100100001010100100101001010
10010100101010010100101010010101
10010101010101010101010101010101



DISCS

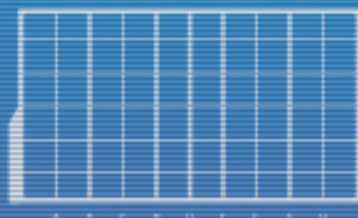
Delays: Propagation Delay

Propagation delay (t_{pd}) is upper bound between new valid inputs and new valid output, or how long it takes for the entire system to “settle” after inputs are changed.



t_{pd} includes time required for transistor to turn on/off.

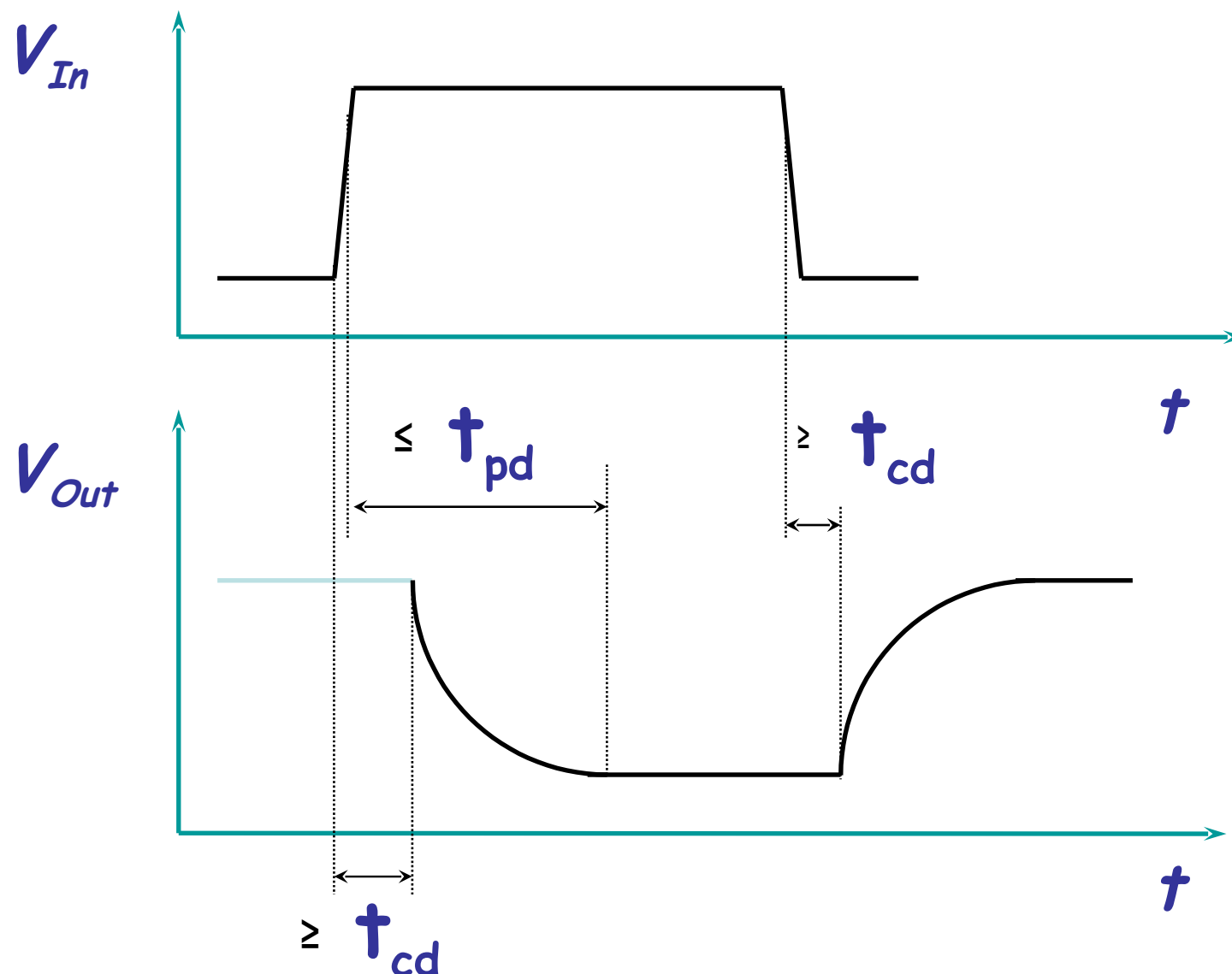
00101010010101000011110100001100
10001100100001111001101010010101
11001010101010100001001100101010100
1001010010010010101010101010101
11100001111010110000000111101001
001001010100101001001010010010110
10010100100001010100100101001010
10010100101010010100101010010101
10010101010101010101010101010101



DISCS

Signal Timing: Contamination Delay

Contamination delay t_{cd} is lower bound between invalid inputs and invalid output, or how long it takes for one invalid signal to mess up the result.

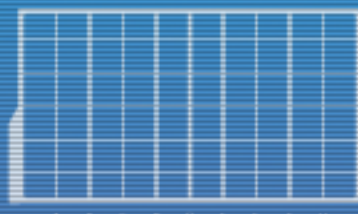


► t_{cd} is also known as *minimum tpd* or tpd_{min} but this obviously causes confusion.

- A device's tpd cannot go below its t_{cd} !

► If not known, assume $t_{cd} = 0$, which is the worst-case scenario: as soon as an invalid input enters, output becomes invalid.

00101010010101000011110100001100
10001100100001111001101010010101
11001010101010100001001100101010100
100101001001001010101010101010101
11100001111010110000000111101001
001001010100101001001010010010110
10010100100001010100100101001010
10010100101010010100101010010101
1001010010101001010101010010101



DISCS

Formal Definitions

*additive:
Device t_{pd}
and t_{cd} is
sum of
its parts.*

- ▶ **Propagation delay t_{pd}** : Worst case delay from new valid inputs to new valid outputs.
- ▶ **Contamination delay t_{cd}** : Best case delay from invalid inputs to invalid outputs.

Over all input combinations, over all input-output paths!

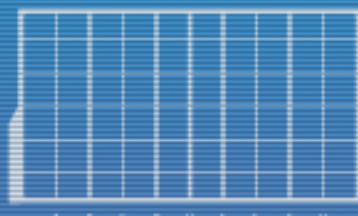
*Property
of last
stage only
(in an
acyclic
circuit).*

- ▶ **Rise time t_r** : Output signal transition time from invalid low to valid high.
- ▶ **Fall time t_f** : Output signal transition time from invalid high to valid low.

*Not quite true, but a
convenient assumption!*

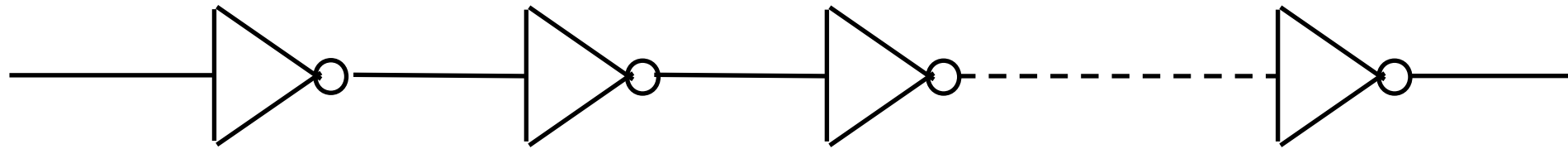
Note: Final output comes from the last stage of any given acyclic circuit, so rise/fall time measurement comes from there BUT these times ARE STILL affected by the rise/fall times of the preceding stages, don't forget!

00101010010101000011110100001100
10001100100001111001101010010101
11001010101010100001001100101010100
100101001001001010101010101010101
11100001111010110000000111101001
0010010101001010010010010010010110
10010100100001010100100101001010
10010100101010010100101010010101
10010101010101010101010101010101

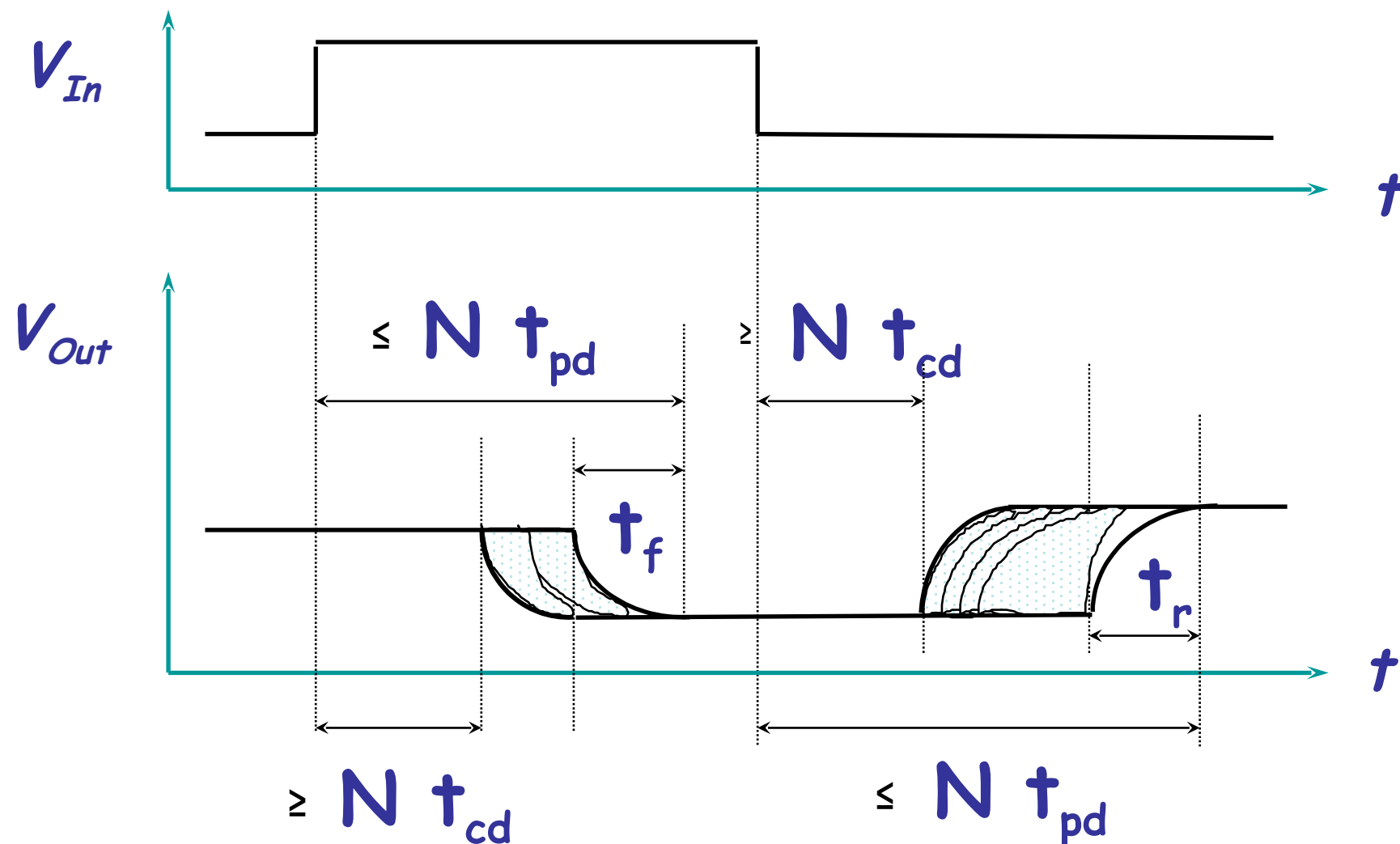


DISCS

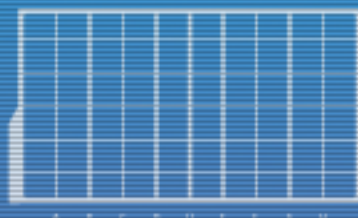
Cascaded Inverters



N inverters each with t_{pd} , t_{cd} , t_r , and t_f

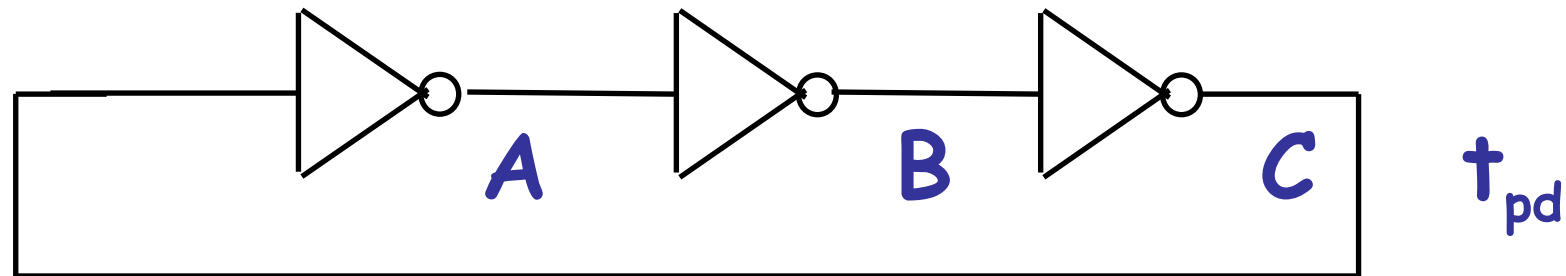


0010101001010100011110100001100
10001100100001111001101010010101
11001010101010100001001100101010100
100101001001001010101010101010101
11100001111010110000000111101001
001001010100101001001010010010110
10010100100001010100100101001010
10010100101010010100101010010101
10010101010101010101010101010101

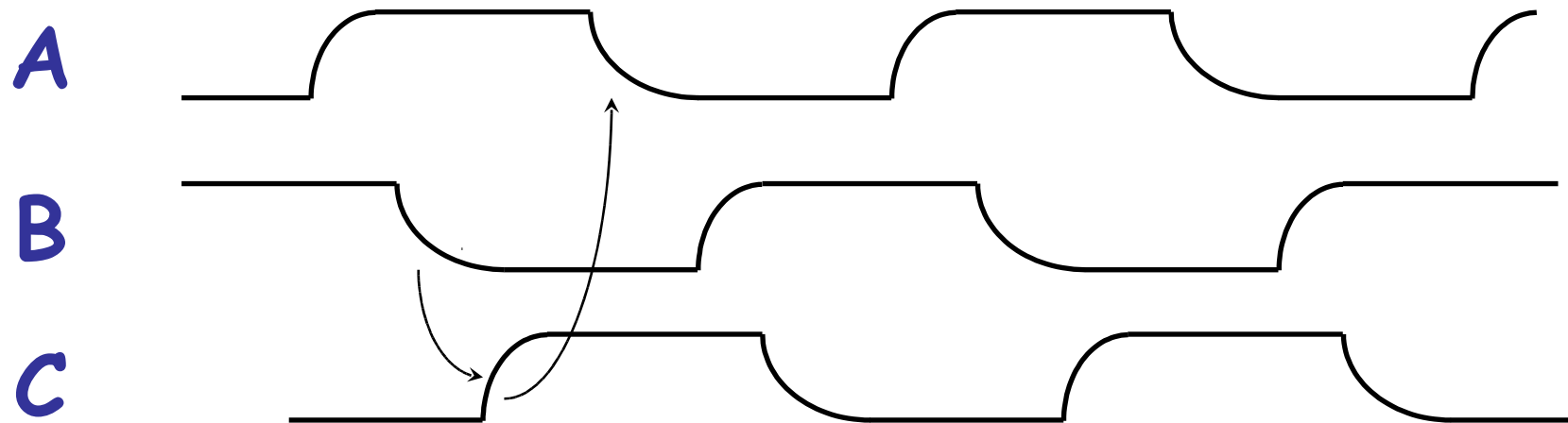


DISCS

Caveat: A Cyclic Circuit

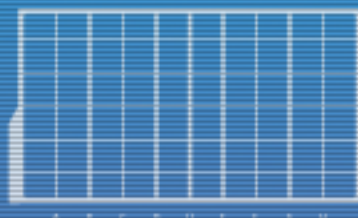


- ▶ Not combinational!
- ▶ Plausible behavior: Oscillates with period $6t_{pd}$



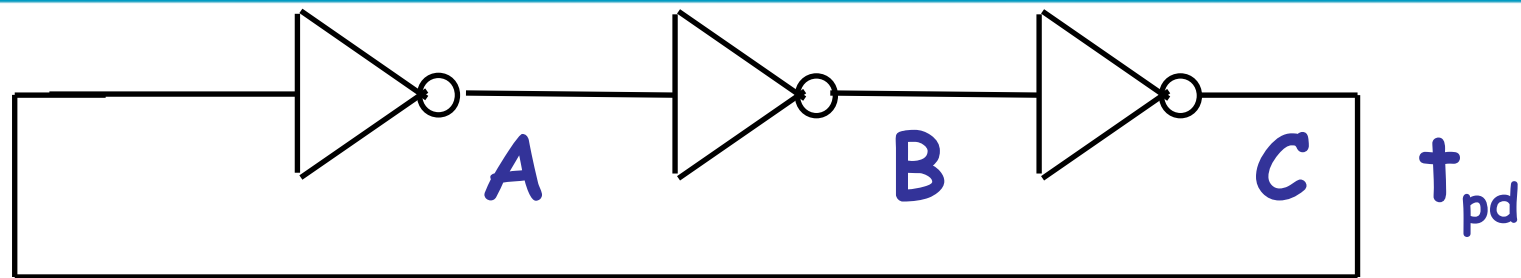
- ▶ Is the behavior guaranteed by static discipline?
Answer: NO!

00101010010101000011110100001100
10001100100001111001101010010101
11001010101010100001001100101010100
100101001001001010101010101010101
11100001111010110000000111101001
001001010100101001001010010010110
10010100100001010100100101001010
10010100101010010100101010010101
10010101010101010101010101010101

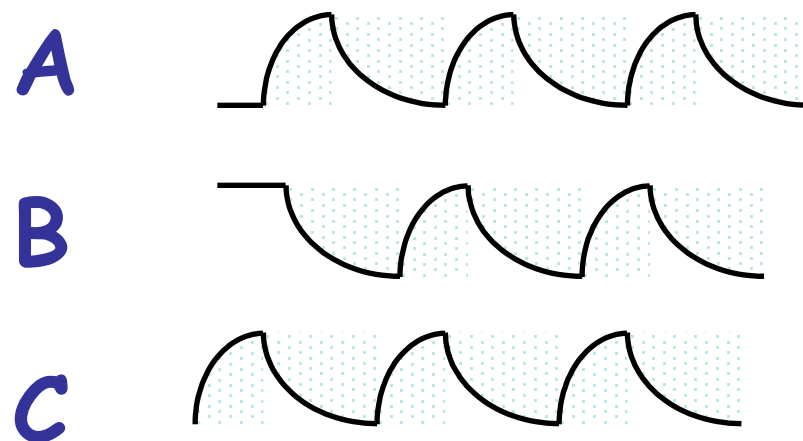


DISCS

Other Behaviors

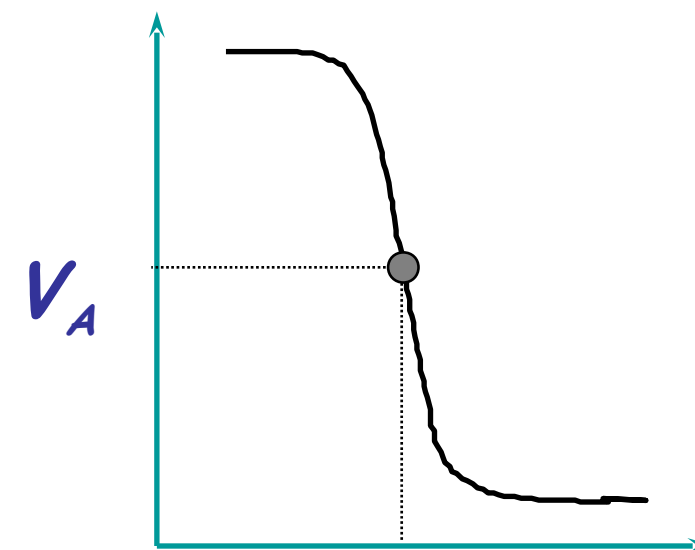


Continuous invalidity?



If no signal is ever valid for t_{pd}

Steady DC signal?



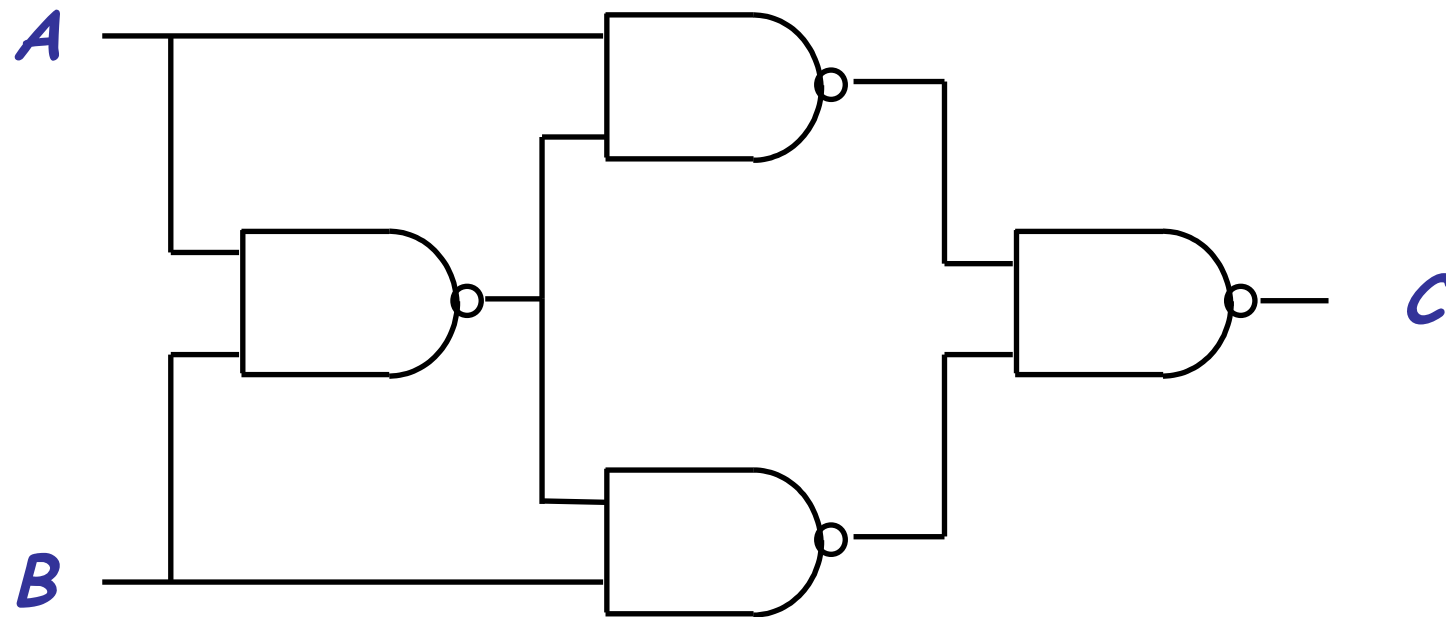
$V_A = V_B = V_C$,
and are in the forbidden zone!

00101010010101000011110100001100
10001100100001111001101010010101
11001010101010100001001100101010100
100101001001001010101010101010101
11100001111010110000000111101001
001001010100101001001010010010110
10010100100001010100100101001010
10010100101010010100101010010101
100101010101010010101010010101



DISCS

Delays in Combinational Circuits



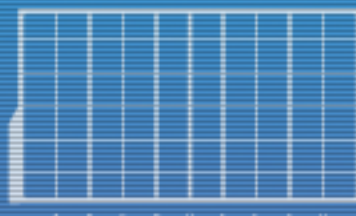
Static Discipline: Logically valid $A, B \Rightarrow$ Logically valid C (after delay)

$t_{pd} = 5\text{ns}$ and $t_{cd} = 1\text{ns}$ for each NAND gate

*Propagation delay for circuit is _____
(Worst case delay from new valid inputs to new valid outputs)*

*Contamination delay for circuit is _____
(Best case delay from invalid inputs to invalid outputs)*

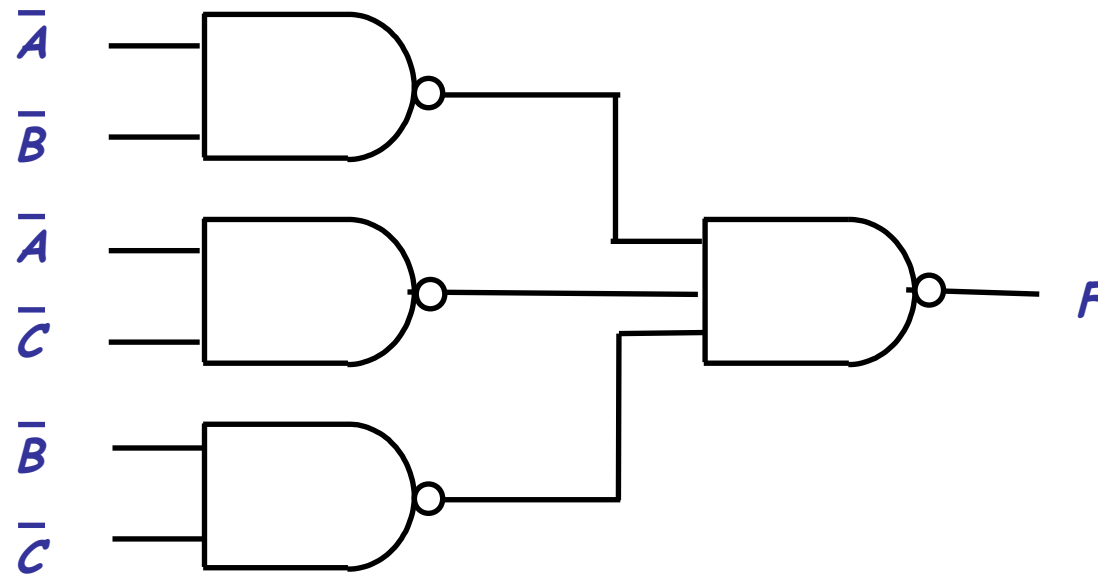
00101010010101000011110100001100
10001100100001111001101010010101
11001010101010100001001100101010100
100101001001001010101010101010101
11100001111010110000000111101001
001001010100101001001010010010110
10010100100001010100100101001010
10010100101010010100101010010101
1001010010101001010101001010101



DISCS

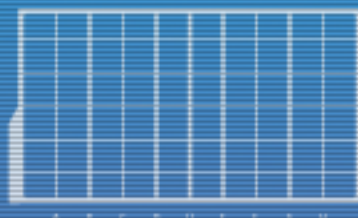
Simplistic Viewpoint

- All we need are two-level circuits!



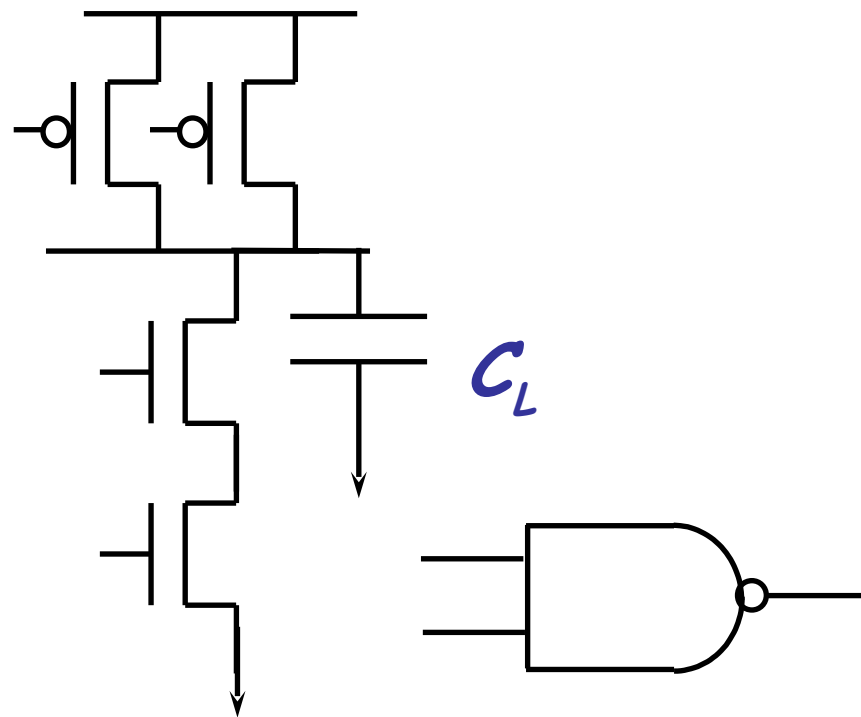
- Can implement all logic functions.
- Two-level circuits are fast because transitions only propagate through two levels of logic (ignoring input inverters).

00101010010101000011110100001100
10001100100001111001101010010101
110010101010100001001100101010100
1001010010010010101010101010101
11100001111010110000000111101001
001001010100101001001010010010110
10010100100001010100100101001010
10010100101010010100101010010101
10010101010101010101010101010101

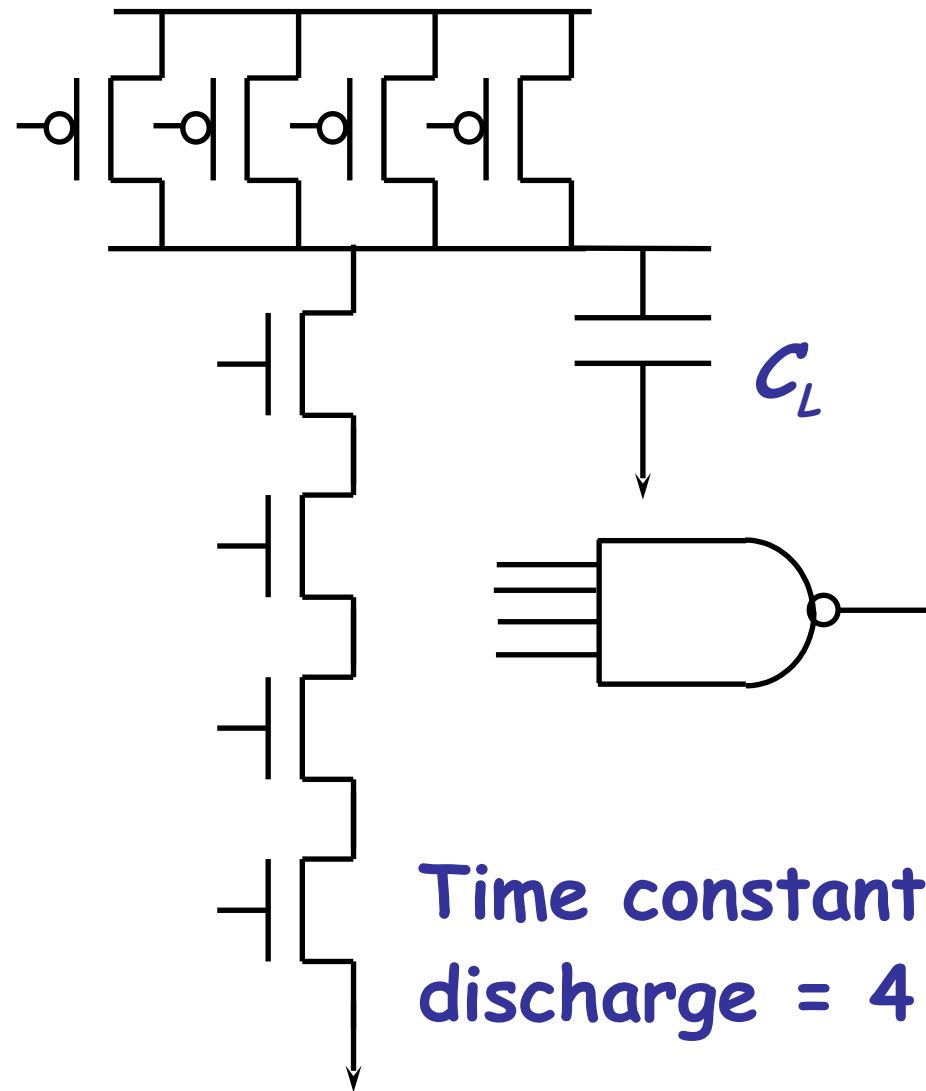


DISCS

Circuit Delay: Is Fanin Free?

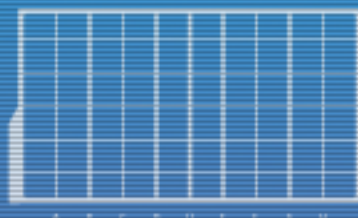


Time constant for C_L
discharge = $2 R_{ON} C_L$



Time constant for C_L
discharge = $4 R_{ON} C_L$

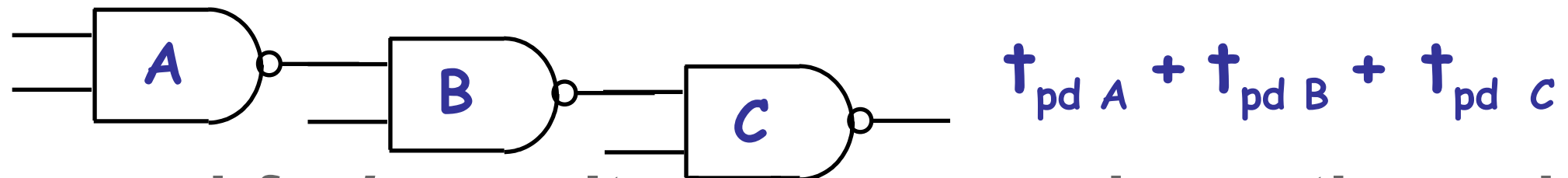
0010101001010100011110100001100
10001100100001111001101010010101
110010101010100001001100101010100
1001010010010010101010101010101
11100001111010110000000111101001
001001010100101001001010010010110
10010100100001010100100101001010
10010100101010010100101010010101
10010101010101010101010101010101



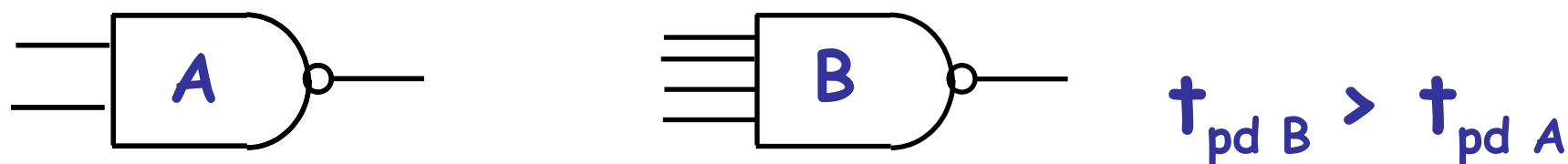
DISCS

Circuit Delay: Synopsis

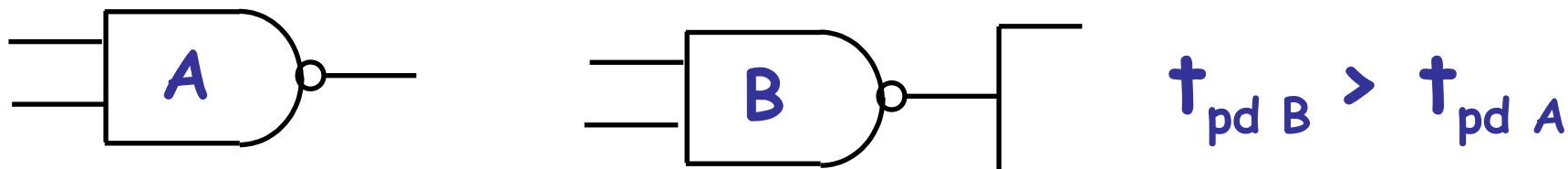
- ▶ Propagation delay accumulates with each level.



- ▶ Increased *fanin* results in longer series paths and therefore increased delay.

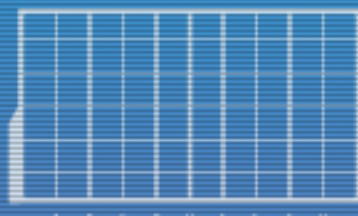


- ▶ Increased *fanout* results in increased load capacitance and therefore increased delay.



- ▶ Chip makers usually specify max fanout (to guarantee tpd, as well as voltage and current).

0010101001010100011110100001100
10001100100001111001101010010101
110010101010101000010011001010100
1001010010010010101010101010101
11100001111010110000000111101001
0010010101001010010010100100110
10010100100001010100100101001010
10010100101010010100101010010101
10010101010101010101010101010101



DISCS

How do we specify what we want?

Truth Tables (Can you handle the truth?)

k inputs

output

rows

input combinations

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

Shortcut
Assume unspecified input combinations produce 0.

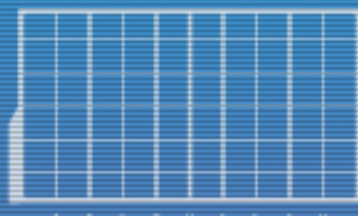
A	B	C	F
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	1	0	1

- # of distinct Truth

Tables with k inputs is _____

- same as # of distinct k -input boolean functions

00101010010101000011110100001100
10001100100001111001101010010101
110010101010101000010011001010100
1001010010010010101010101010101
11100001111010110000000111101001
001001010100101001001010010010110
10010100100001010100100101001010
10010100101010010100101010010101
10010101010101010101010101010101



Truth Tables and Logic Equations

- Truth tables define Boolean functions

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

$F = 1$ if $A=0$ AND $B=0$ AND $C=0$ OR
 $A=0$ AND $B=0$ AND $C=1$ OR
 $A=0$ AND $B=1$ AND $C=0$ OR
 $A=0$ AND $B=1$ AND $C=1$ OR
 $A=1$ AND $B=1$ AND $C=0$
 $F = 0$ otherwise

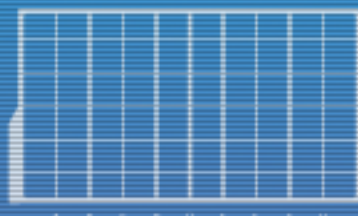
Write as a logic equation (Boolean function):

$$F = \bar{A} \cdot \bar{B} \cdot \bar{C} + \bar{A} \cdot \bar{B} \cdot C + \bar{A} \cdot B \cdot \bar{C} + \bar{A} \cdot B \cdot C + A \cdot B \cdot \bar{C}$$

or

$$F = \bar{A} \bar{B} \bar{C} + \bar{A} \bar{B} C + \bar{A} B \bar{C} + \bar{A} B C + A B \bar{C}$$

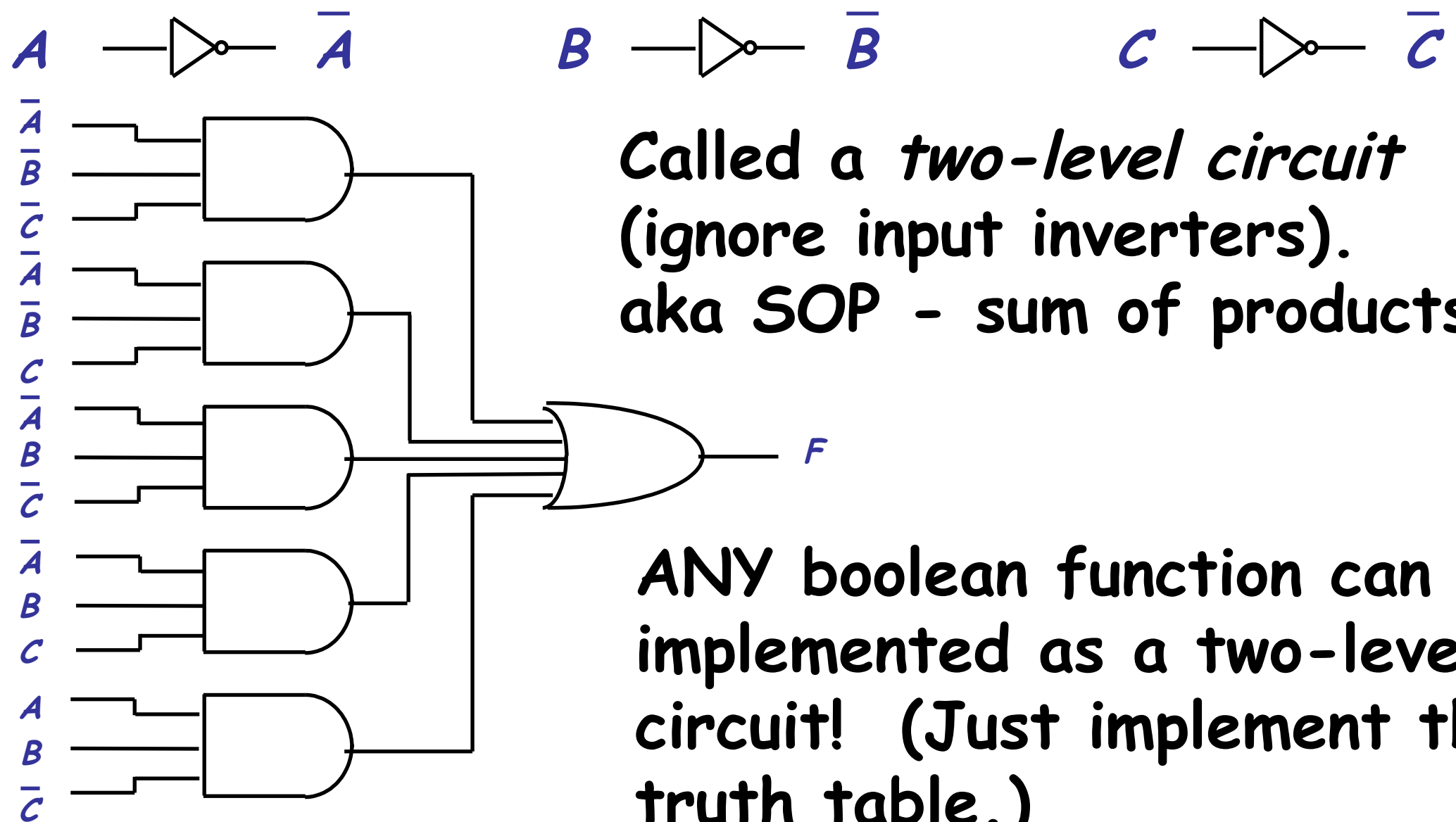
00101010010101000011110100001100
 10001100100001111001101010010101
 110010101010101000010011001010100
 1001010010010010101010101010101
 11100001111010110000000111101001
 001001010100101001001010010010110
 10010100100001010100100101001010
 10010100101010010100101010010101
 10010101010101010101010101010101



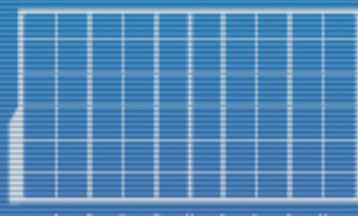
DISCS

Logic Equations and AND-OR Circuits

$$F = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}B\bar{C} + \bar{A}BC + A\bar{B}\bar{C}$$



00101010010101000011110100001100
10001100100001111001101010010101
11001010101010100001001100101010100
100101001001001010101010101010101
11100001111010110000000111101001
001001010100101001001010010010110
10010100100001010100100101001010
10010100101010010100101010010101
1001010010101001010101001010101



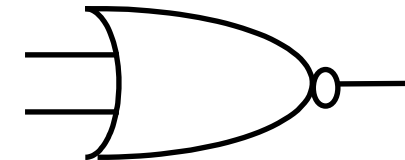
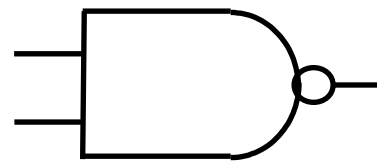
DISCS

NAND-NAND and NOR-NOR Circuits

- ▶ All circuits can be implemented with NOT, AND, OR!
- ▶ In fact, all circuits can be implemented using just NANDs or just NORs!

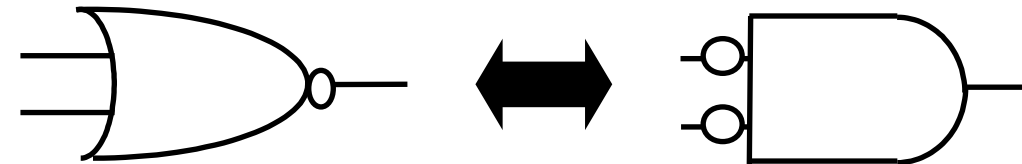
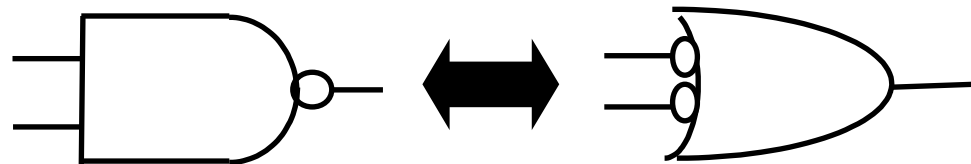
- ▶ **Proof:**

- ▶ **N = NOT**



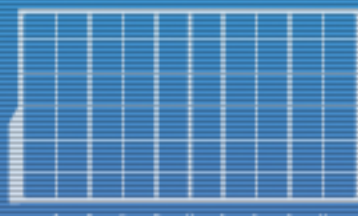
- ▶ **De Morgan's Law ("pushing bubbles")**

- ▶ **Change the gate when pushing bubbles through it.**



- ▶ **Can implement AND and OR.**
 - ▶ **NAND is just OR with inverted inputs.**
 - ▶ **NOR is just AND with inverted inputs.**

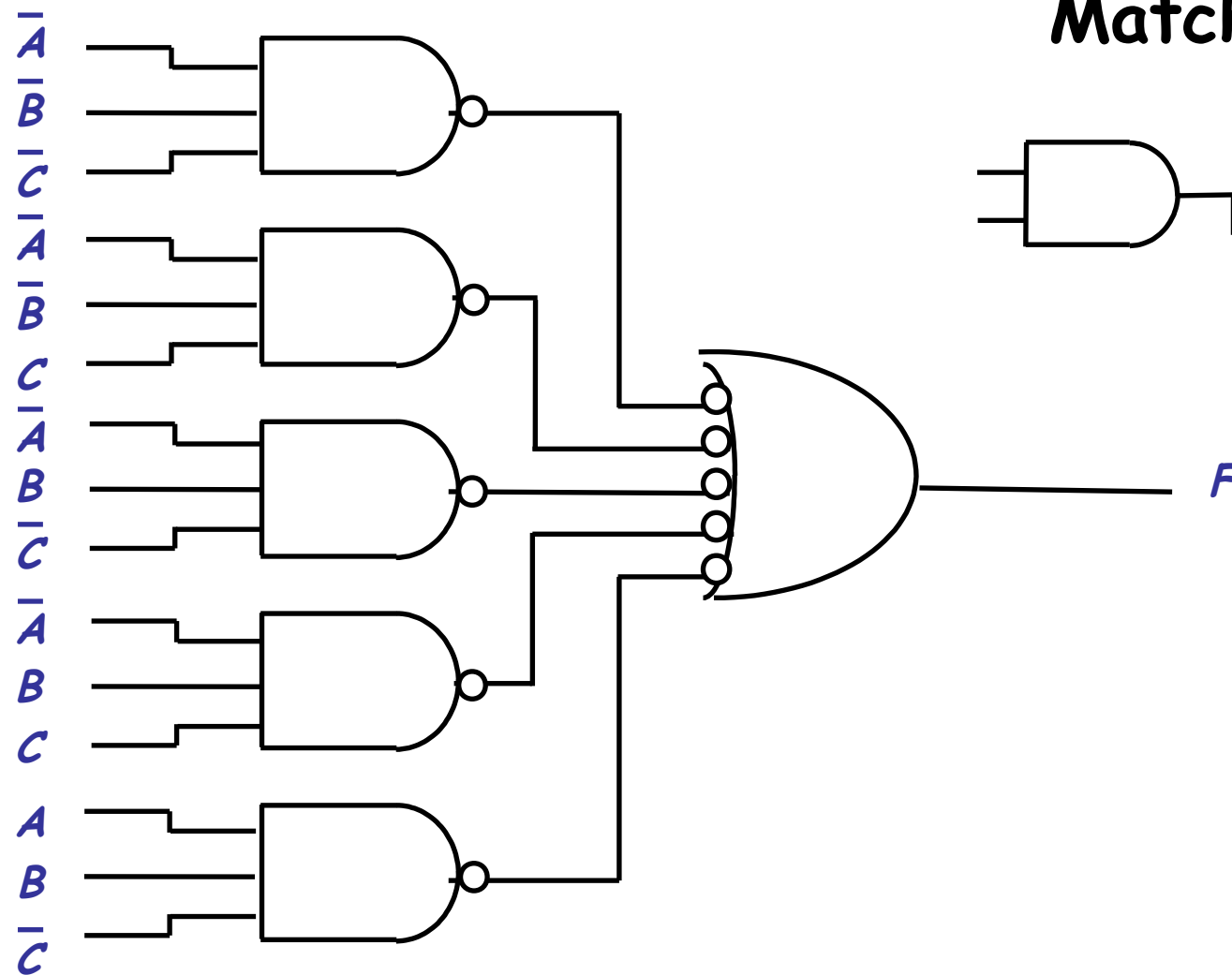
00101010010101000011110100001100
10001100100001111001101010010101
11001010101010100001001100101010100
100101001001001010101010101010101
11100001111010110000000111101001
001001010100101001001010010010110
10010100100001010100100101001010
10010100101010010100101010010101
10010101010101010101010101010101



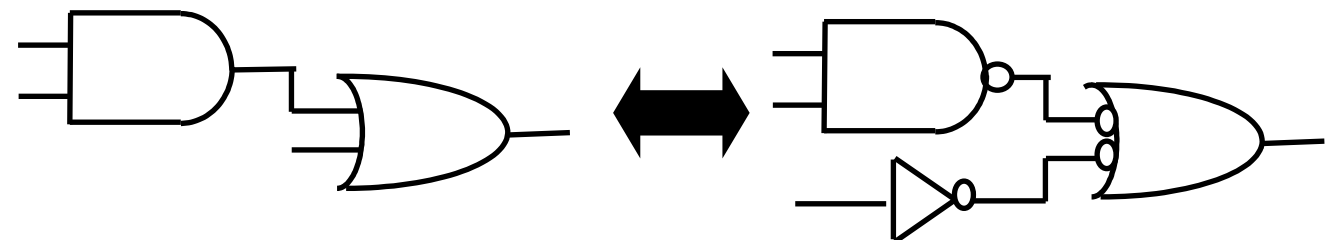
DISCS

NAND-NAND Circuit

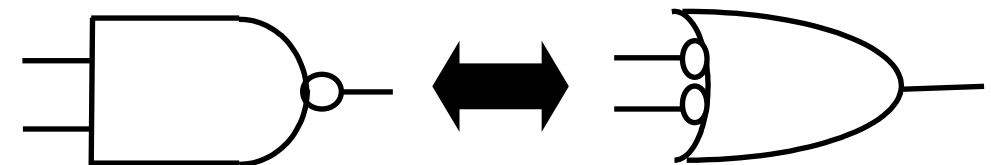
$$F = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}B\bar{C} + \bar{A}BC + A\bar{B}\bar{C}$$



Matching bubbles (negative logic)

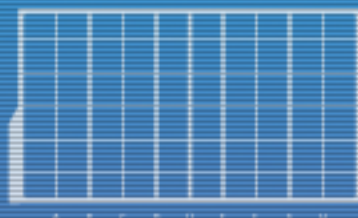


DeMorgan's law:



Number of transistors in above circuit (ignoring input inverters):
 40 = 5*6 + 1*10 (3-NAND = 6 trans., 5-NAND = 10 trans.)

0010101001010100011110100001100
 10001100100001111001101010010101
 110010101010101000010011001010100
 1001010010010010101010101010101
 11100001111010110000000111101001
 001001010100101001001010010010110
 10010100100001010100100101001010
 10010100101010010100101010010101
 10010100101010010100101010010101



DISCS

Simplifying Logic: Review

Given logic functions f, g, h

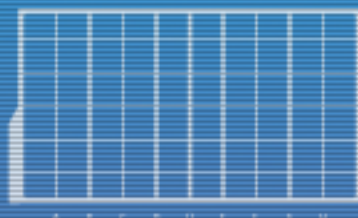
$$\begin{array}{ll} f \cdot f & = f \\ f \cdot 1 & = f \\ f \cdot 0 & = 0 \\ f \cdot \bar{f} & = 0 \end{array} \qquad \begin{array}{ll} f + f & = f \\ f + 1 & = 1 \\ f + 0 & = f \\ f + \bar{f} & = 1 \end{array}$$

Distributive laws

$$f (g + h) = f g + f h$$

$$(f + g) (f + h) = f + g h$$

00101010010101000011110100001100
10001100100001111001101010010101
11001010101010100001001100101010100
100101001001001010101010101010101
11100001111010110000000111101001
001001010100101001001010010010110
10010100100001010100100101001010
10010100101010010100101010010101
10010101010101010101010101010101



DISCS

Simplifying Logic

Apply identities to simplify equation for F

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

$$= \underline{\hspace{2cm}} + \underline{\hspace{2cm}} + A B \overline{C}$$

$$= \underline{\hspace{2cm}} + A B \overline{C}$$

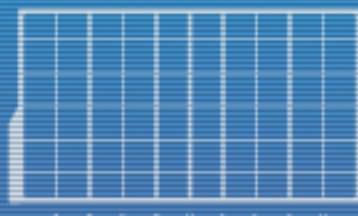
$$= \underline{\hspace{2cm}}$$

$f \cdot f = f$	$f + f = f$
$f \cdot 1 = f$	$f + 1 = 1$
$f \cdot 0 = 0$	$f + 0 = f$
$f \cdot f = 0$	$f + f = 1$

$$f(g + h) = fg + fh$$

$$(f + g)(f + h) = f + gh$$

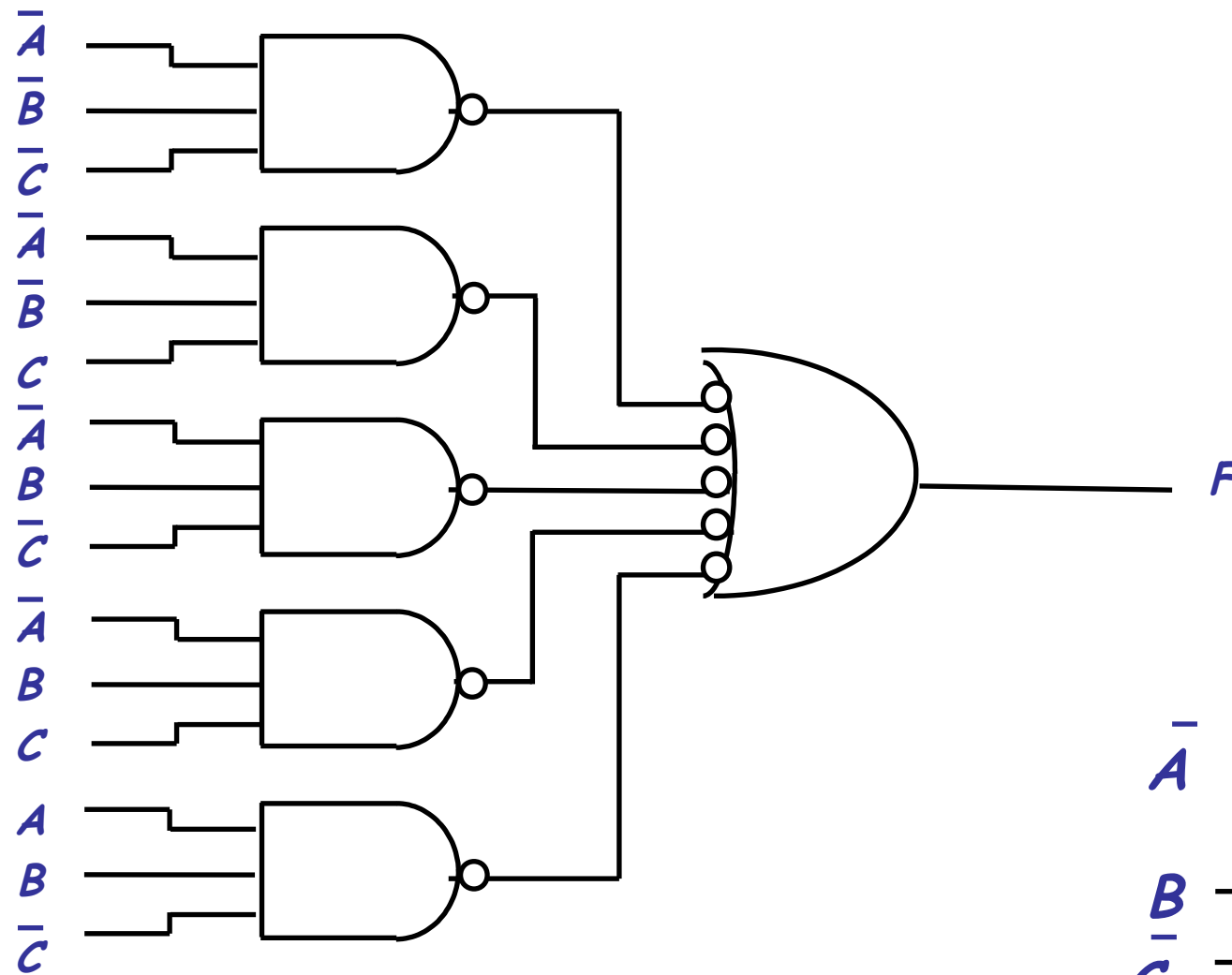
00101010010101000011110100001100
10001100100001111001101010010101
110010101010100001001100101010100
1001010010010010101010101010101
11100001111010110000000111101001
001001010100101001001010010010110
10010100100001010100100101001010
10010100101010010100101010010101
10010101010101010101010101010101



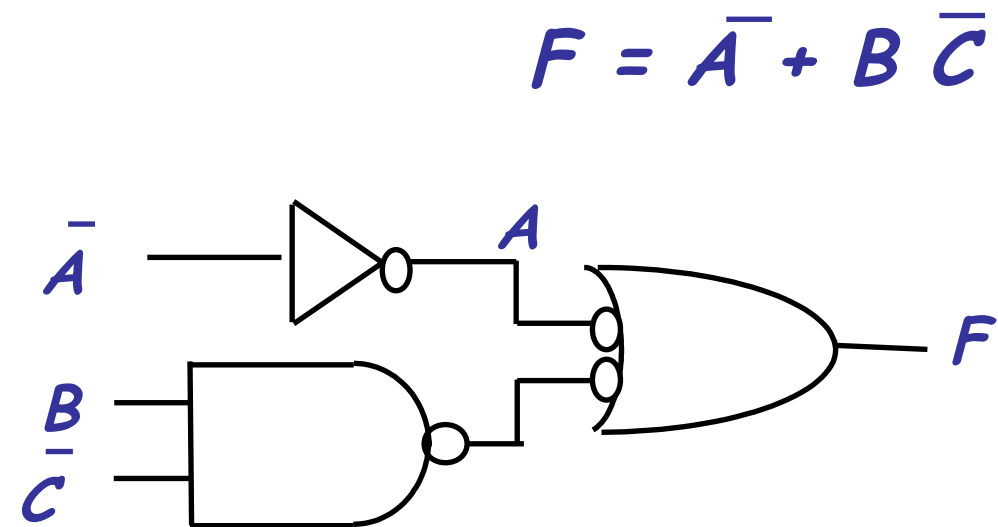
DISCS

Simplifying Logic: Comparison

$$F = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}B\bar{C} + \bar{A}BC + A\bar{B}\bar{C}$$

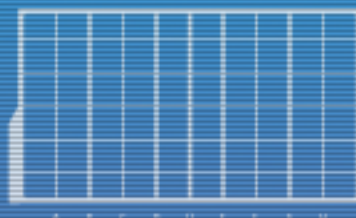


40 Transistors



8 Transistors

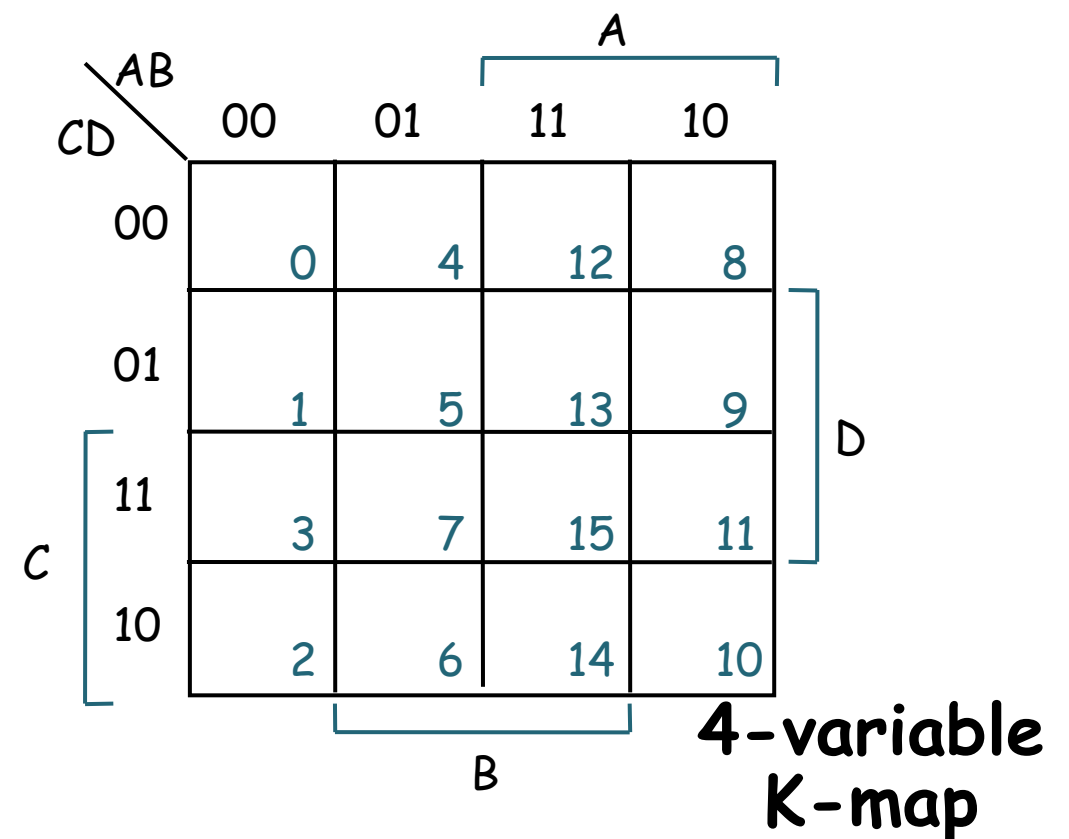
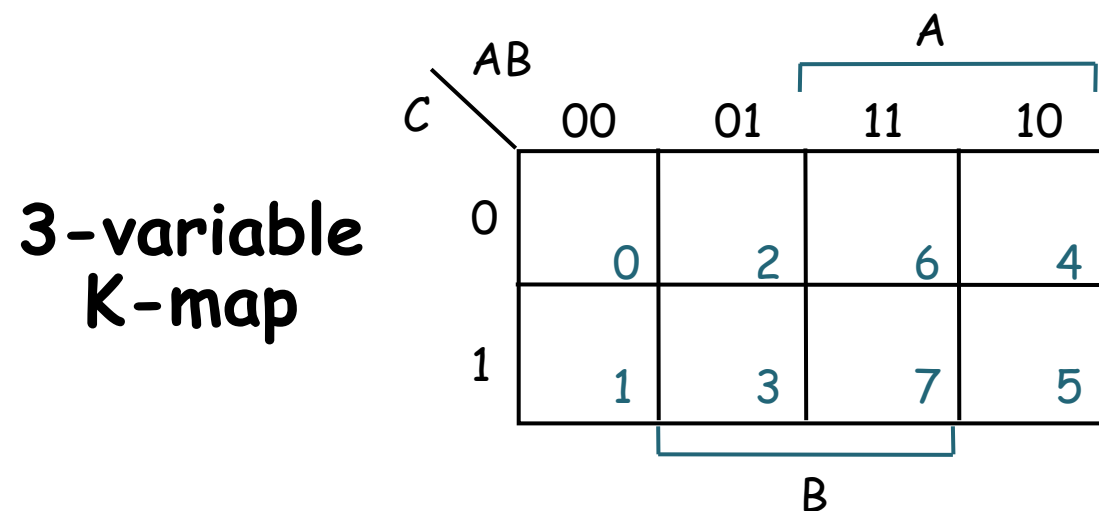
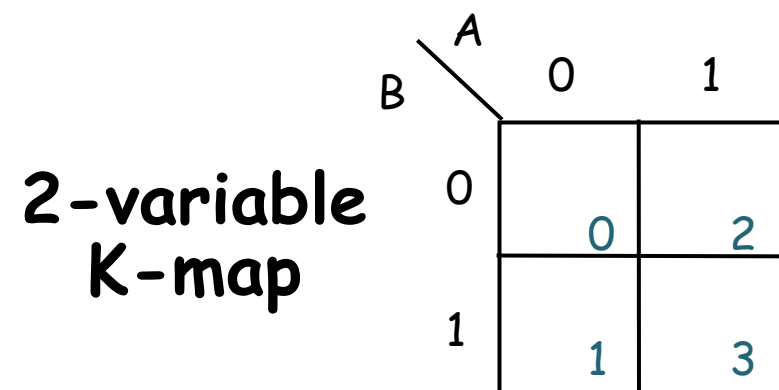
00101010010101000111110100001100
10001100100001111001101010010101
11001010101010100001001100101010100
100101001001001010101010101010101
11100001111010110000000111101001
001001010100101001001010010010110
10010100100001010100100101001010
10010100101010010100101010010101
10010101010101010101010101010101



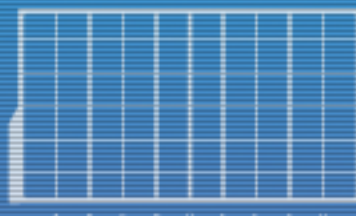
DISCS

Review: Karnaugh Maps

- ▶ Graphical representation of truth table.
- ▶ Gray code: 00, 01, 11, 10
 - ▶ Only a single bit changes bet. adjacent code words.
 - ▶ Adjacent boxes on K-map represent AND-terms (products) w/ only 1-bit difference.



0010101001010100011110100001100
 10001100100001111001101010010101
 110010101010101000010011001010100
 1001010010010010101010101010101
 11100001111010110000000111101001
 001001010100101001001010010010110
 10010100100001010100100101001010
 10010100101010010100101010010101
 10010101010101010101010101010101



DISCS

Example: 3-variable K-Map

$$F = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}B\bar{C} + \bar{A}BC + AB\bar{C}$$

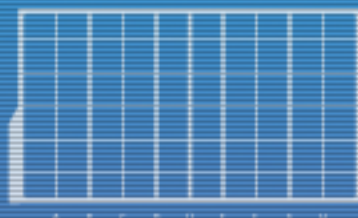
		A			
		00	01	11	10
C \ AB	0	1	1	1	0
	1	1	1	0	0

B

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

$$F = \bar{A} + B\bar{C}$$

00101010010101000011110100001100
 10001100100001111001101010010101
 110010101010101000010011001010100
 1001010010010010101010101010101
 11100001111010110000000111101001
 001001010100101001001010010010110
 10010100100001010100100101001010
 10010100101010010100101010010101
 100101010101001010101010010101



DISCS

Example: 4-variables

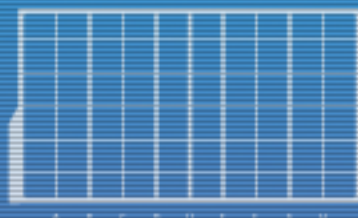
AB		A			
		00	01	11	10
CD	00	1	0	0	1
	01	0	1	0	0
	11	1	1	1	1
	10	1	1	1	1

Diagram illustrating a 4-variable Karnaugh map for variables A, B, C, and D. The map is a 4x4 grid with rows labeled CD (00, 01, 11, 10) and columns labeled AB (00, 01, 11, 10). The map shows the ON-set (1s) and OFF-set (0s). The variables A, B, C, and D are indicated by blue bars: A is a horizontal bar above the columns 11 and 10; B is a horizontal bar below the columns 00 and 01; C is a vertical bar to the left of the rows 11 and 10; D is a vertical bar to the right of the rows 01 and 11.

$F(A,B,C,D) =$

- Adjacencies wrap-around left-right, top-bottom, and around corners.
- Find the smallest number of the largest possible squares that cover the ON-set.

00101010010101000011110100001100
 10001100100001111001101010010101
 110010101010101000010011001010100
 1001010010010010101010101010101
 11100001111010110000000111101001
 00100101010010100100100100100110
 10010100100001010100100101001010
 10010100101010010100101010010101
 1001010010101001010101010010101



DISCS

Example: 4-variables

AB \ CD		A			
		00	01	11	10
C	00	1	0	0	1
	01	0	1	0	0
	11	1	1	1	1
	10	1	1	1	1
		B			
		00	01	11	10

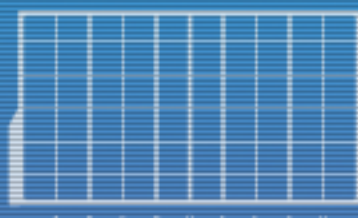
Diagram illustrating a 4-variable Karnaugh map for variables A, B, C, and D. The map is a 4x4 grid with rows labeled by CD (00, 01, 11, 10) and columns labeled by AB (00, 01, 11, 10). The values in the cells are: (00,00)=1, (01,00)=0, (11,00)=0, (10,00)=1, (00,01)=0, (01,01)=1, (11,01)=0, (10,01)=0, (00,11)=1, (01,11)=1, (11,11)=1, (10,11)=1, (00,10)=1, (01,10)=1, (11,10)=1, (10,10)=1. Blue boxes highlight the prime implicants: a horizontal box covering (00,00), (01,00), (11,00), (10,00); a vertical box covering (01,00), (01,01), (01,11), (01,10); a horizontal box covering (00,11), (01,11), (11,11), (10,11); a horizontal box covering (00,10), (01,10), (11,10), (10,10); a vertical box covering (00,00), (00,11), (00,10), (00,01); a vertical box covering (10,00), (10,11), (10,10), (10,01); a horizontal box covering (00,00), (00,11), (00,10), (00,01); and a horizontal box covering (10,00), (10,11), (10,10), (10,01).

$$F(A,B,C,D) =$$

$$C + \bar{A} B D + \bar{B} \bar{D}$$

- ▶ Adjacencies wrap-around left-right, top-bottom, and around corners.
- ▶ Find the smallest number of the largest possible squares that cover the ON-set.

00101010010101000011110100001100
 10001100100001111001101010010101
 110010101010101000010011001010100
 1001010010010010101010101010101
 11100001111010110000000111101001
 00100101010010100100100100100110
 10010100100001010100100101001010
 10010100101010010100101010010101
 1001010010101001010101010010101



DISCS

Product-of-Sums form

AB		A			
		00	01	11	10
CD	00	1	0	0	1
	01	0	1	0	0
	11	1	1	1	1
	10	1	1	1	1

B

$$F = (\bar{B} + C + D) (\bar{A} + C + \bar{D}) (B + C + \bar{D})$$

- Circle 0's instead.
- Flip +'s and •'s.
- Flip (invert) variables.

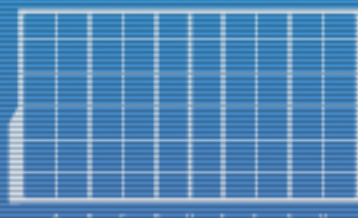
Side note: using DeMorgan's Law

$$\bar{F} = B \bar{C} \bar{D} + A \bar{C} D + B C D$$

$$\bar{F} = B \bar{C} \bar{D} + A \bar{C} D + B C D$$

$$F = (\bar{B} + C + D) (\bar{A} + C + \bar{D}) (B + C + \bar{D})$$

00101010010101000011110100001100
10001100100001111001101010010101
11001010101010100001001100101010100
100101001001001010101010101010101
11100001111010110000000111101001
001001010100101001001010010010110
10010100100001010100100101001010
10010100101010010100101010010101
10010101010101010101010101010101



DISCS

Don't Cares

Don't Cares can be treated as 1's or 0's if it is advantageous to do so

AB \ CD		A			
		00	01	11	10
C	00	0	0	X	0
	01	1	1	X	1
	11	1	1	0	0
	10	0	X	0	0

$$F = A'D + B'C'D \text{ w/o don't cares}$$

$$F = A'D + C'D \text{ w/ don't cares}$$

By treating this X as a "1", we save gates and literals

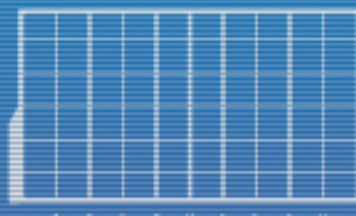
In PoS form: $F = D (A' + C')$

Same answer as above, but less literals.

(Note: don't cares are treated differently as convenient.)

AB \ CD		A			
		00	01	11	10
C	00	0	0	X	0
	01	1	1	X	1
	11	1	1	0	0
	10	0	X	0	0

00101010010101000011110100001100
10001100100001111001101010010101
110010101010100001001100101010100
1001010010010010101010101010101
11100001111010110000000111101001
001001010100101001001010010010110
10010100100001010100100101001010
10010100101010010100101010010101
10010100101010010100101010010101



DISCS

Another example: BCD +1

BCD = "Binary Coded Decimal"

only
care
about
0 to 9

A	B	C	D	W	X	Y	Z
0	0	0	0	0	0	0	1
0	0	0	1	0	0	1	0
0	0	1	0	0	0	1	1
0	0	1	1	0	1	0	0
0	1	0	0	0	1	0	1
0	1	0	1	0	1	1	0
0	1	1	0	0	1	1	1
0	1	1	1	1	0	0	0
1	0	0	0	1	0	0	1
1	0	0	1	0	0	0	0
1	0	1	0	x	x	x	x
1	0	1	1	x	x	x	x
1	1	0	0	x	x	x	x
1	1	0	1	x	x	x	x
1	1	1	0	x	x	x	x
1	1	1	1	x	x	x	x

		A				
		AB				
		00	01	11	10	
W	CD	00	0	0	X	1
		01	0	0	X	0
		11	0	1	X	X
		10	0	0	X	X

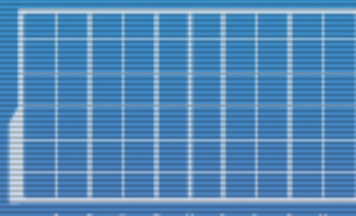
		A				
		AB				
		00	01	11	10	
X	CD	00	0	1	X	0
		01	0	1	X	0
		11	1	0	X	X
		10	0	1	X	X

		A				
		AB				
		00	01	11	10	
y	CD	00	0	0	X	0
		01	1	1	X	0
		11	0	0	X	X
		10	1	1	X	X

		A				
		AB				
		00	01	11	10	
Z	CD	00	1	1	X	1
		01	0	0	X	0
		11	0	0	X	X
		10	1	1	X	X

W = ? X = ? Y = ? Z = ?

0010101001010100011110100001100
10001100100001111001101010010101
110010101010101000010011001010100
1001010010010010101010101010101
11100001111010110000000111101001
0010010101001010010010100100110
10010100100001010100100101001010
10010100101010010100101010010101
10010100101010010100101010010101



DISCS