Problem 1. **Combinational construction rules**

In lecture, we learned two basic principles regarding the class of combinational devices. The first allows us to build a combinational device from, e.g., electronic components:

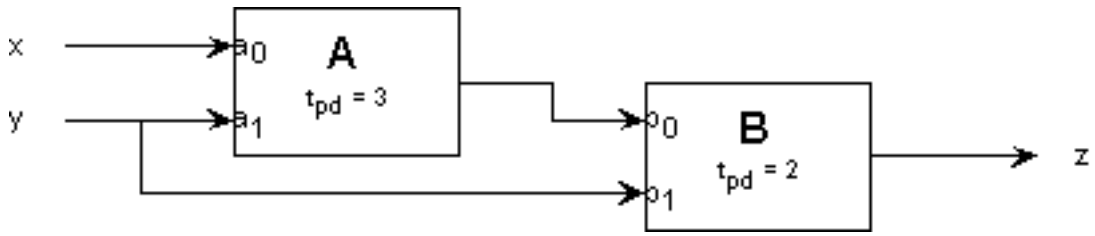
- A *combinational device* is a circuit element that has
    - one or more digital *inputs*
    - one or more digital *outputs*
    - a *functional specification* that details the value of each output for every possible combination of valid input values
    - a *timing specification* consisting (at minimum)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

while the second allows us to construct complex combinational devices from acyclic circuits containing simpler ones:

- 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

In this problem, we ask you to think carefully about why these rules work - in particular, why an acyclic circuit of combinational devices, constructed according to the second principle, is itself a combinational device as defined by the first. You may assume for the following that every input and output is a logical 0 or 1.

Consider the following 2-input acyclic circuit whose two components, A and B, are each combinational devices:



The propagation delay - the upper bound on the output settling time - for each device is specified in nanoseconds. The functional specifications for each component are given as truth tables detailing output values for each combination of inputs:

a_0	a_1	$A_{(a0, a1)}$
0	0	1
0	1	0
1	0	0
1	1	1

b_0	b_1	$B_{(b0, b1)}$
0	0	0
0	1	0
1	0	0
1	1	1

A. Give a truth table for the acyclic circuit, i.e. a table that specifies the value of z for each of the possible combinations of input values on x and y.

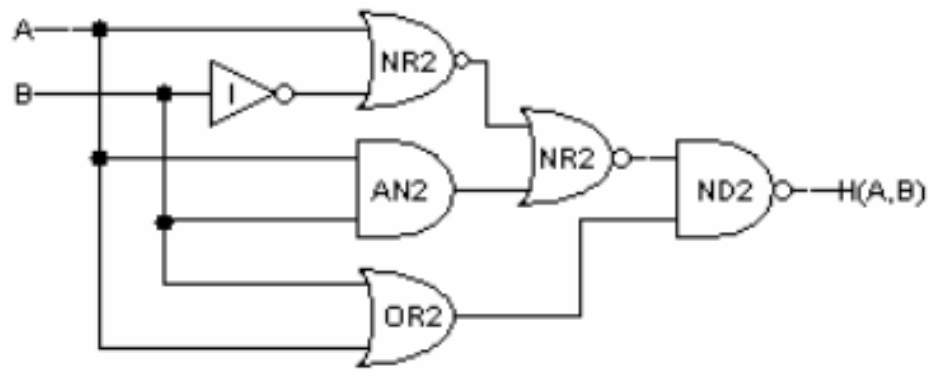
B. Describe a general procedure by which a truth table can be computed for each output of an arbitrary acyclic circuit containing only combinational components. [HINT: construct a functional specification to each circuit node].

C. Specify a propagation delay (the upper bound required for each combinational device) for the circuit.

D. Describe a general procedure by which a propagation delay can be computed for an arbitrary acyclic circuit containing only combinational components. [HINT: add a timing specification to each circuit node].

E. Do your general procedures for computing functional specifications and propagation delays work if the restriction to acyclic circuits is relaxed? Explain.

Problem 2. Consider the following circuit that implements the 2-input function H(A,B):



A. Fill in the following truth table for H. Give a sum-of-products expression that corresponds to the truth table below.

A	B	H
0	0	
0	1	
1	0	
1	1	

gate	t_{CD}	t_{PD}	t_R	t_F
I	3ps	15ps	8ps	5ps
ND2	5ps	30ps	11ps	7ps
AN2	12ps	50ps	13ps	9ps
NR2	5ps	30ps	7ps	11ps
OR2	12ps	50ps	9ps	13ps

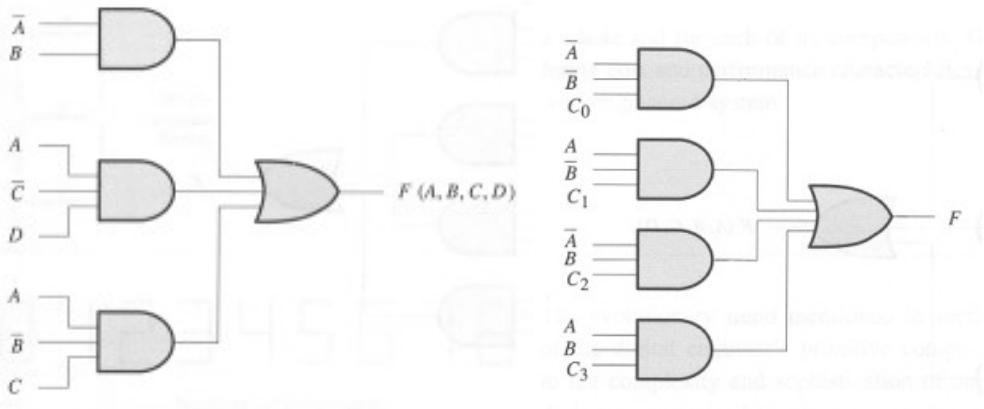
B. Using the table on the right of timing specifications for each component, what are t_{CD} and t_{PD} for the circuit shown above?

Problem 3. **Gates and Boolean equations**

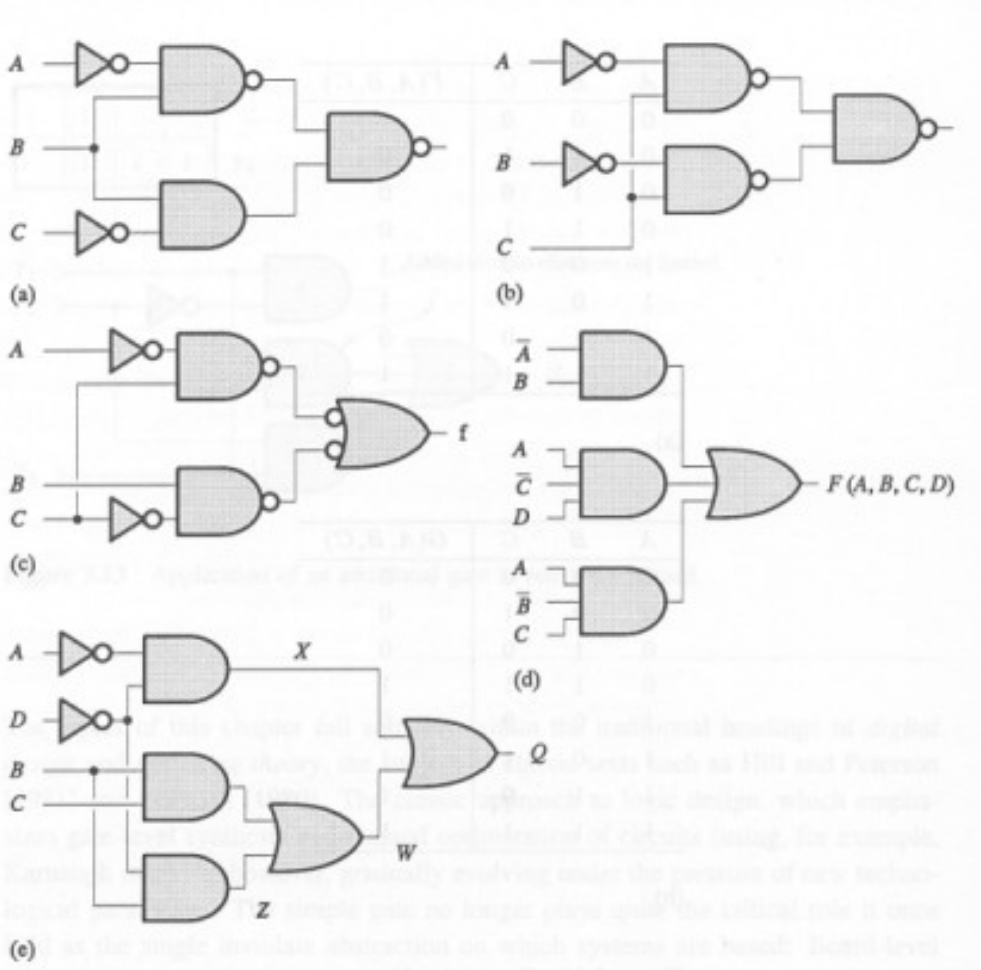
A. Show the Boolean equation for the function F described by the circuit on the left.

B. Consider the circuit shown on the right. Each of the control inputs, C0 through C3, must be tied to a constant, either 0 or 1. What are the values of C0 through C3 that would cause F to be the *exclusive OR* of A and B?

C. Can any arbitrary Boolean function of A and B be realized through appropriate wiring of the control signals C0 through C3?



D. Give a sum-of-products expression for each of the following circuits:



E. Give a canonical sum-of-products expression for the Boolean function described by each truth table below

A	B	C	$F(A,B,C)$
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

A	B	C	$G(A,B,C)$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

F. We've seen that there are a total of sixteen 2-input Boolean functions. How many 5-input Boolean functions are there?