Chapter 5

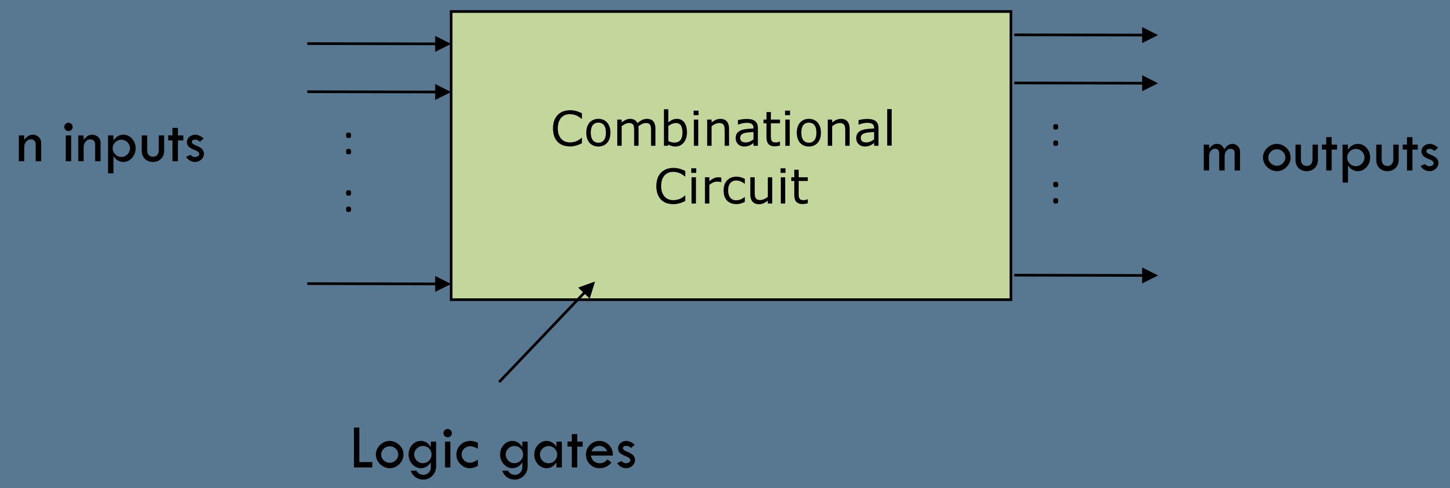
DESIGN of COMBINATIONAL CIRCUITS

Kinds of Logic Circuits

- Combinational circuit
  - output is determined by the current inputs with some delays of logic gates only
  - performs specific information-processing operation
- Sequential circuit
  - output is determined by the inputs and the state of the memory elements
  - output can change when clock ticks

Combinational Circuit

- General combinational circuit



Combinational Logic Design

- Specify the design problem
  – make sure you completely understand all the design requirements
- Generate a truth table
  – translate the word statement into an input/output table
- Generate an SOP equation
  – write a Boolean equation based from the truth table

Combinational Logic Design

- Simplify the equation
  - use Boolean algebra (or some other means) to reduce the equation
- Implement the expression using logic gates

Step One - Requirements

- Make sure you completely understand all the design requirements
  - functionality
  - performance
  - reliability
  - fault-tolerance
  - maintainability

Step One - Requirements

Example:

Design a combinational circuit with three inputs (a, b, c) and one output, F such that F is 1 when all three inputs are 1 or when a is 1 and either b or c is also 1 or when all three inputs are 0.

Step Two – Truth Table

- List all possible inputs and outputs
  - for the example there are 3 inputs A, B, and C so how many rows will there be in the truth table?

Step Two – Truth Table

- List all possible inputs and outputs
  - for the example there are 3 inputs A, B, and C so how many rows will there be in the truth table?

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

Step Two – Truth Table

- List all possible inputs and outputs
  - for the example there are 3 inputs A, B, and C so how many rows will there be in the truth table?

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

F is one when:
all three inputs are 0

A is 1 and either B or C is 1

All three inputs are 1

F is zero for all other inputs

Step Three - Equation

- SOP - Sum Of Products
- Look at those inputs in the truth table which produce a “1” on the output
  - each of these input conditions represents a product (AND) term in the SOP equation containing the input variable or its complement
  - if the input variable is “1” the term contains the variable
  - if it is “0” the term contains the complement of the variable
  - these product terms are ORed

Example

- Translate the truth table into an SOP function

A	B	C	F	
0	0	0	1	A'B'C'
0	0	1	0	
0	1	0	0	
0	1	1	0	
1	0	0	0	
1	0	1	1	AB'C
1	1	0	1	ABC'
1	1	1	1	ABC

Example

- Translate the truth table into an SOP function

$F = A'B'C' + AB'C + ABC' + ABC$

A	B	C	F	
0	0	0	1	$A'B'C'$
0	0	1	0	
0	1	0	0	
0	1	1	0	
1	0	0	0	
1	0	1	1	$AB'C$
1	1	0	1	ABC'
1	1	1	1	ABC

Step Four - Simplify

- Use standard Boolean algebra to reduce the expression

$$F = A'B'C' + AB'C + ABC' + ABC$$

Step Four - Simplify

- Use standard Boolean algebra to reduce the expression

$$F = A'B'C' + AB'C + ABC' + ABC$$

$$= A'B'C' + AC(B' + B) + ABC'$$

Step Four - Simplify

- Use standard Boolean algebra to reduce the expression

$$F = A'B'C' + AB'C + ABC' + ABC$$

$$= A'B'C' + AC(B' + B) + ABC'$$

$$= A'B'C' + AC + ABC'$$

Step Four - Simplify

- Use standard Boolean algebra to reduce the expression

$$F = A'B'C' + AB'C + ABC' + ABC$$

$$= A'B'C' + AC(B' + B) + ABC'$$

$$= A'B'C' + AC + ABC'$$

$$= A'B'C' + A(C + BC')$$

Step Four - Simplify

- Use standard Boolean algebra to reduce the expression

$$
\begin{aligned}
F &= A'B'C' + AB'C + ABC' + ABC \\
&= A'B'C' + AC(B' + B) + ABC' \\
&= A'B'C' + AC + ABC' \\
&= A'B'C' + A(C + BC') \\
&= A'B'C' + A(C + B)(C + C')
\end{aligned}
$$

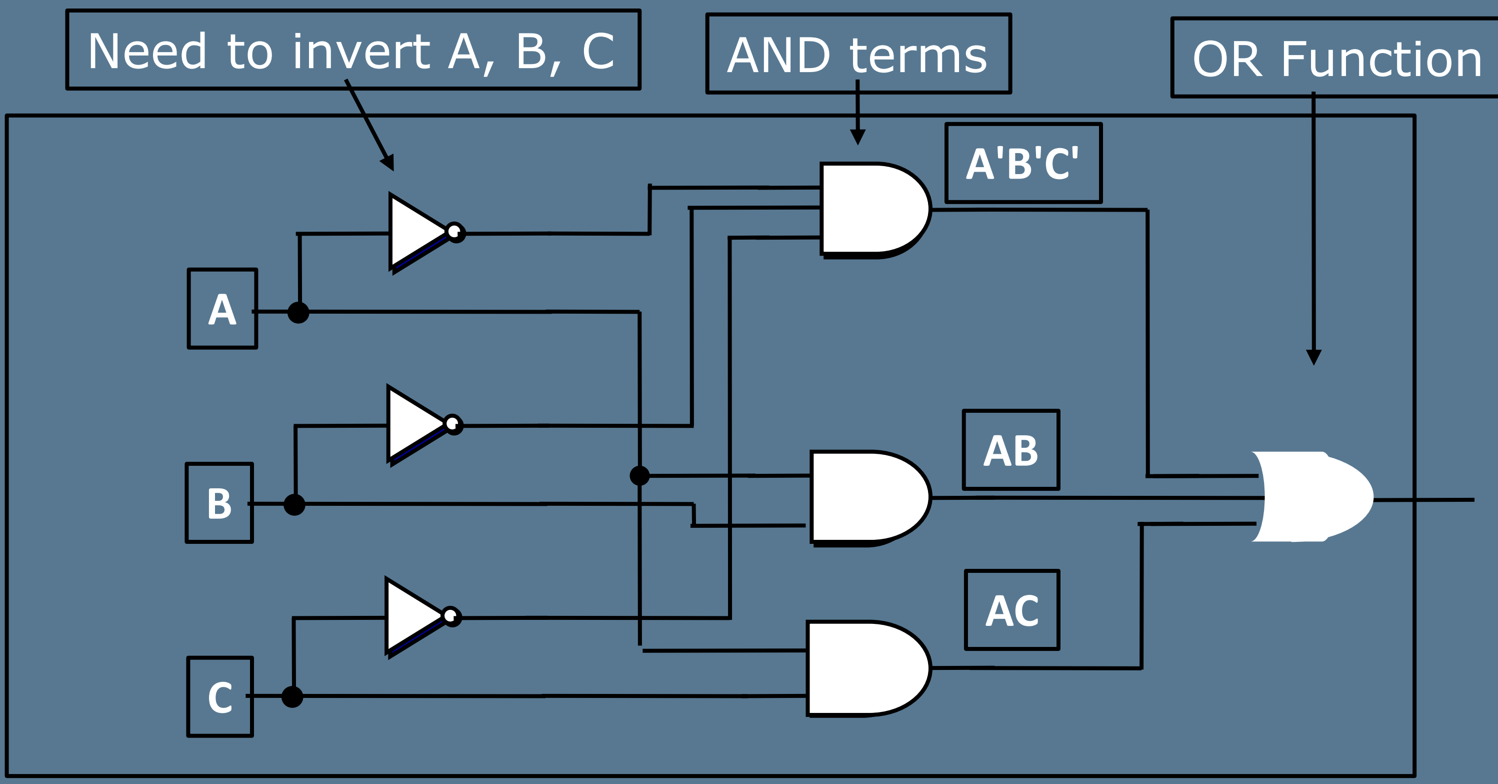
Step Four - Simplify

- Use standard Boolean algebra to reduce the expression

$$
\begin{aligned}
F &= A'B'C' + AB'C + ABC' + ABC \\
&= A'B'C' + AC(B' + B) + ABC' \\
&= A'B'C' + AC + ABC' \\
&= A'B'C' + A(C + BC') \\
&= A'B'C' + A(C + B)(C + C') \\
&= A'B'C' + AC + AB
\end{aligned}
$$

Step Five – The Circuit

- There is a direct correspondence between the equation and the circuit



Design Example

- Design a combinational circuit that generates the 9's complement of a BCD digit. For example, the complement of (0100) is (0101).

Truth Table

a b c d	
0 0 0 0	
0 0 0 1	
0 0 1 0	
0 0 1 1	
0 1 0 0	
0 1 0 1	
0 1 1 0	
0 1 1 1	

a b c d	
1 0 0 0	
1 0 0 1	
1 0 1 0	
1 0 1 1	
1 1 0 0	
1 1 0 1	
1 1 1 0	
1 1 1 1	

Truth Table

a b c d	W X Y Z
0 0 0 0	
0 0 0 1	
0 0 1 0	
0 0 1 1	
0 1 0 0	
0 1 0 1	
0 1 1 0	
0 1 1 1	

a b c d	W X Y Z
1 0 0 0	
1 0 0 1	
1 0 1 0	
1 0 1 1	
1 1 0 0	
1 1 0 1	
1 1 1 0	
1 1 1 1	

Truth Table

a b c d	W X Y Z
0 0 0 0	1 0 0 1
0 0 0 1	1 0 0 0
0 0 1 0	0 1 1 1
0 0 1 1	0 1 1 0
0 1 0 0	0 1 0 1
0 1 0 1	0 1 0 0
0 1 1 0	0 0 1 1
0 1 1 1	0 0 1 0

a b c d	W X Y Z
1 0 0 0	
1 0 0 1	
1 0 1 0	
1 0 1 1	
1 1 0 0	
1 1 0 1	
1 1 1 0	
1 1 1 1	

Truth Table

a b c d	W X Y Z
0 0 0 0	1 0 0 1
0 0 0 1	1 0 0 0
0 0 1 0	0 1 1 1
0 0 1 1	0 1 1 0
0 1 0 0	0 1 0 1
0 1 0 1	0 1 0 0
0 1 1 0	0 0 1 1
0 1 1 1	0 0 1 0

a b c d	W X Y Z
1 0 0 0	0 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

Equations

- $W = a'b'c'd' + a'b'c'd$
- $X = a'b'cd' + a'b'cd + a'bc'd' + a'bc'd$
- $Y = a'b'cd' + a'b'cd + a'bcd' + a'bcd$
- $Z = a'b'c'd' + a'b'cd' + a'bc'd' + a'bcd' + ab'c'd'$

K-map Simplification

- W = a’b’c’d’ + a’b’c’d

AB \ CD	0 0	0 1	1 1	1 0
0 0	1	1		
0 1				
1 1	X	X	X	X
1 0			X	X

K-map Simplification

- W = a’b’c’d’ + a’b’c’d

AB \ CD	0 0	0 1	1 1	1 0
0 0	1	1		
0 1				
1 1	X	X	X	X
1 0			X	X

K-map Simplification

- W = a’b’c’d’ + a’b’c’d = a’b’c’

AB \ CD	0 0	0 1	1 1	1 0
0 0	1	1		
0 1				
1 1	X	X	X	X
1 0			X	X

a’b’c’

K-map Simplification

- X = a'b'cd' + a'b'cd + a'bc'd' + a'bc'd

AB \ CD	0 0	0 1	1 1	1 0
0 0			1	1
0 1	1	1		
1 1	X	X	X	X
1 0			X	X

K-map Simplification

- X = a’b’cd’ + a’b’cd + a’bc’d’ + a’bc’d

AB \ CD	0 0	0 1	1 1	1 0
0 0			1	1
0 1	1	1		
1 1	X	X	X	X
1 0			X	X

K-map Simplification

- X = a’b’cd’ + a’b’cd + a’bc’d’ + a’bc’d

  = b’c + bc'

AB \ CD	0 0	0 1	1 1	1 0
0 0			1	1
0 1	1	1		
1 1	X	X	X	X
1 0			X	X

b'c

bc’

K-map Simplification

- Y = a'b'cd' + a'b'cd + a'bcd' + a'bcd

AB \ CD	00	01	11	10
00			1	1
01			1	1
11	X	X	X	X
10			X	X

K-map Simplification

- Y = a'b'cd' + a'b'cd + a'bcd' + a'bcd

AB \ CD	0 0	0 1	1 1	1 0
0 0			1	1
0 1			1	1
1 1	X	X	X	X
1 0			X	X

K-map Simplification

- Y = a’b’cd’ + a’b’cd + a’bcd’ + a’bcd

  = c

AB \ CD	0 0	0 1	1 1	1 0
0 0			1	1
0 1			1	1
1 1	X	X	X	X
1 0			X	X

c

K-map Simplification

- Z = a'b'c'd' + a'b'cd' + a'bc'd' + a'bcd' + ab'c'd'

AB \ CD	0 0	0 1	1 1	1 0
0 0	1			1
0 1	1			1
1 1	X	X	X	X
1 0	1		X	X

K-map Simplification

- Z = a'b'c'd' + a'b'cd' + a'bc'd' + a'bcd' + ab'c'd'

AB \ CD	0 0	0 1	1 1	1 0
0 0	1			1
0 1	1			1
1 1	X	X	X	X
1 0	1		X	X

K-map Simplification

- Z = a'b'c'd' + a'b'cd' + a'bc'd' + a'bcd' + ab'c'd' = d'

AB \ CD	0 0	0 1	1 1	1 0
0 0	1			1
0 1	1			1
1 1	X	X	X	X
1 0	1		X	X

d'

Simplification

- W = a’b’c’
- X = bc’ + b’c
- Y = c
- Z = d’

More Examples

1) Design a combinational circuit that converts BCD to Excess-3 code. For example, (0100) to (0111).

Example 1: Truth Table

a b c d	W X Y Z
0 0 0 0	
0 0 0 1	
0 0 1 0	
0 0 1 1	
0 1 0 0	
0 1 0 1	
0 1 1 0	
0 1 1 1	

a b c d	W X Y Z
1 0 0 0	
1 0 0 1	
1 0 1 0	
1 0 1 1	
1 1 0 0	
1 1 0 1	
1 1 1 0	
1 1 1 1	

Example 1: Truth Table

a b c d	W X Y Z
0 0 0 0	0 0 1 1
0 0 0 1	0 1 0 0
0 0 1 0	0 1 0 1
0 0 1 1	0 1 1 0
0 1 0 0	0 1 1 1
0 1 0 1	1 0 0 0
0 1 1 0	1 0 0 1
0 1 1 1	1 0 1 0

a b c d	W X Y Z
1 0 0 0	1 0 1 1
1 0 0 1	1 1 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

Equations

- $W(a,b,c,d) = \Sigma\ (5,6,7,8,9)$
- $X(a,b,c,d) = \Sigma\ (1,2,3,4,9)$
- $Y(a,b,c,d) = \Sigma\ (0,3,4,7,8)$
- $Z(a,b,c,d) = \Sigma\ (0,2,4,6,8)$

Simplification

- $W(a,b,c,d) = \Sigma(5,6,7,8,9)$

AB \ CD	0 0	0 1	1 1	1 0
0 0				
0 1		1	1	1
1 1	X	X	X	X
1 0	1	1	X	X

Simplification

- W(a,b,c,d) = Σ(5,6,7,8,9) = a + bc + bd

AB \ CD	0 0	0 1	1 1	1 0
0 0				
0 1		1	1	1
1 1	X	X	X	X
1 0	1	1	X	X

Simplification

- X(a,b,c,d) = Σ(1,2,3,4,9)

AB \ CD	0 0	0 1	1 1	1 0
0 0		1	1	1
0 1	1			
1 1	X	X	X	X
1 0		1	X	X

Simplification

- X(a,b,c,d) = Σ(1,2,3,4,9) = b'c + b'd + bc'd'

AB \ CD	0 0	0 1	1 1	1 0
0 0		1	1	1
0 1	1			
1 1	X	X	X	X
1 0		1	X	X

Simplification

- $Y(a,b,c,d) = \Sigma(0,3,4,7,8) = cd + c'd'$

AB \ CD	0 0	0 1	1 1	1 0
0 0	1		1	
0 1	1		1	
1 1	X	X	X	X
1 0	1		X	X

Simplification

- Z(a,b,c,d) = $\Sigma(0,2,4,6,8)$ = d'

AB \ CD	0 0	0 1	1 1	1 0
0 0	1			1
0 1	1			1
1 1	X	X	X	X
1 0	1		X	X

More Examples

1) Design a combinational circuit that converts BCD to Excess-3 code. For example, (0100) to (0111).
2) Design a circuit whose output is HIGH whenever A and B are both HIGH as long as C and D are either both LOW or both HIGH.

Example 2: Truth Table and Expression

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

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

Example 2: Truth Table and Expression

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

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

Example 2: Truth Table and Expression

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

$F(A,B,C,D) = ABC'D' + ABCD \quad = AB\,(C'D' + CD)$

More Examples

1) Design a combinational circuit that converts BCD to Excess-3 code. For example, (0100) to (0111).
2) Design a circuit whose output is HIGH whenever A and B are both HIGH as long as C and D are either both LOW or both HIGH.
3) Design a circuit which detects whether a four-variable input is a palindrome or not.

Example 3: Truth Table and Expression

w x y z	F
0 0 0 0	
0 0 0 1	
0 0 1 0	
0 0 1 1	
0 1 0 0	
0 1 0 1	
0 1 1 0	
0 1 1 1	

w x y z	F
1 0 0 0	
1 0 0 1	
1 0 1 0	
1 0 1 1	
1 1 0 0	
1 1 0 1	
1 1 1 0	
1 1 1 1	

Example 3: Truth Table and Expression

w x y z	F
0 0 0 0	1
0 0 0 1	0
0 0 1 0	0
0 0 1 1	0
0 1 0 0	0
0 1 0 1	0
0 1 1 0	1
0 1 1 1	0

w x y z	F
1 0 0 0	0
1 0 0 1	1
1 0 1 0	0
1 0 1 1	0
1 1 0 0	0
1 1 0 1	0
1 1 1 0	0
1 1 1 1	1

Example 3: Truth Table and Expression

w x y z	F
0 0 0 0	1
0 0 0 1	0
0 0 1 0	0
0 0 1 1	0
0 1 0 0	0
0 1 0 1	0
0 1 1 0	1
0 1 1 1	0

w x y z	F
1 0 0 0	0
1 0 0 1	1
1 0 1 0	0
1 0 1 1	0
1 1 0 0	0
1 1 0 1	0
1 1 1 0	0
1 1 1 1	1

$F(w,x,y,z) = w'x'y'z' + w'xyz' + wx'y'z + wxyz$

Other Tasks

- Need to determine the actual gates to use
- Find the speed
- Find the cost
- Find the power requirements

Example

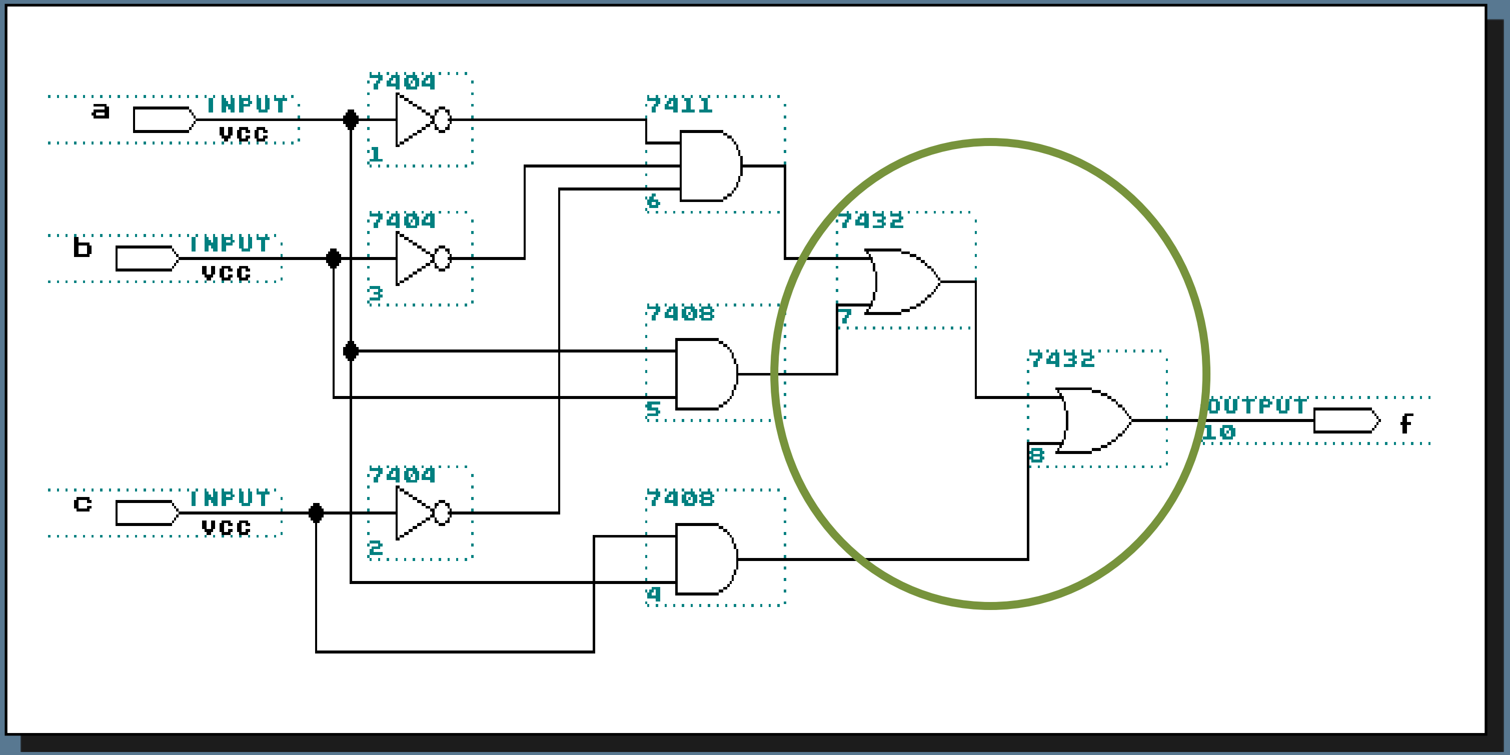
- Design a combinational circuit with three inputs (a, b, c) and one output, F such that F is 1 when all three inputs are 1 or when a is 1 and either b or c is also 1 or when all three inputs are 0.

$$F = a'b'c' + ac + ab$$

Solution One

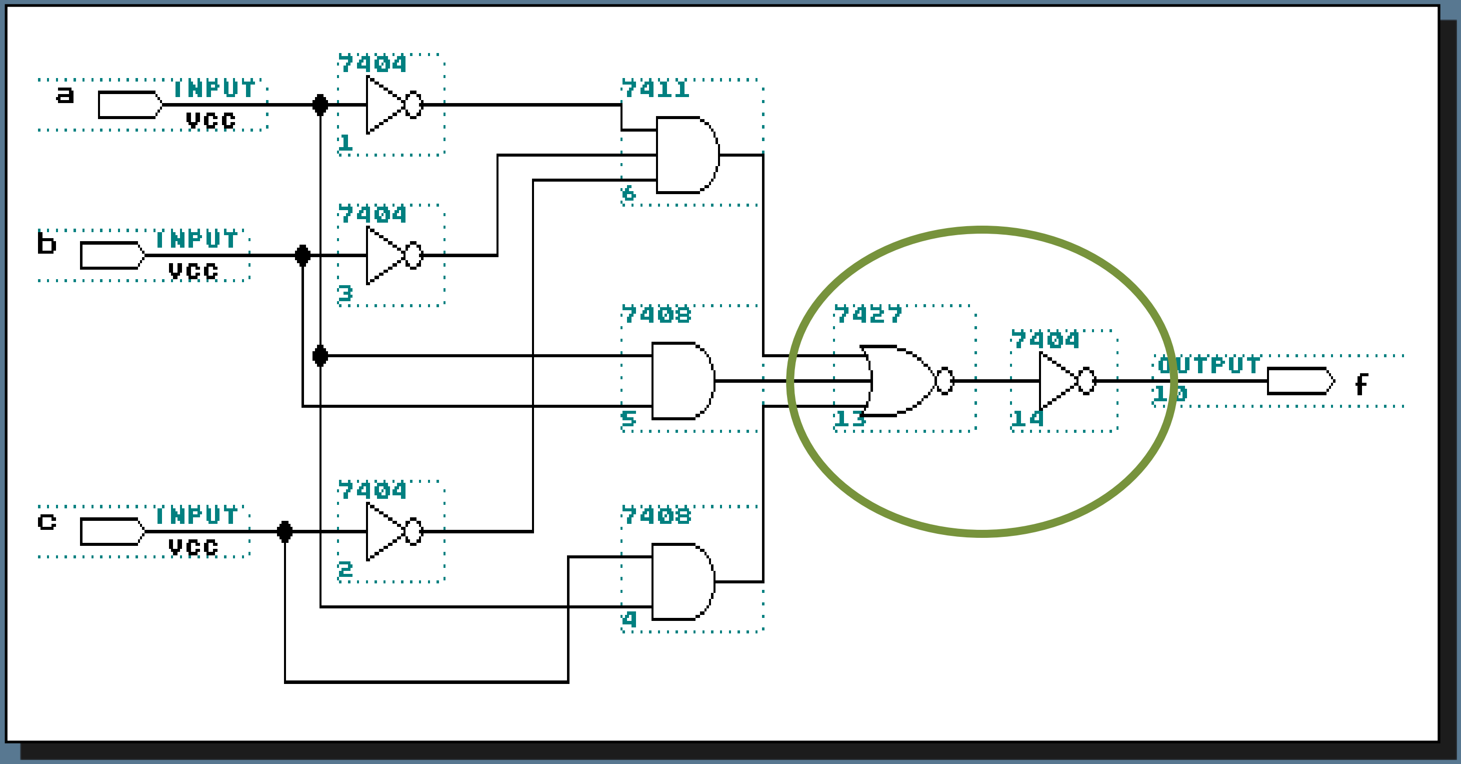
- Use two OR gates - each with two inputs for

  F = a'b'c' + ac + ab



Solution Two

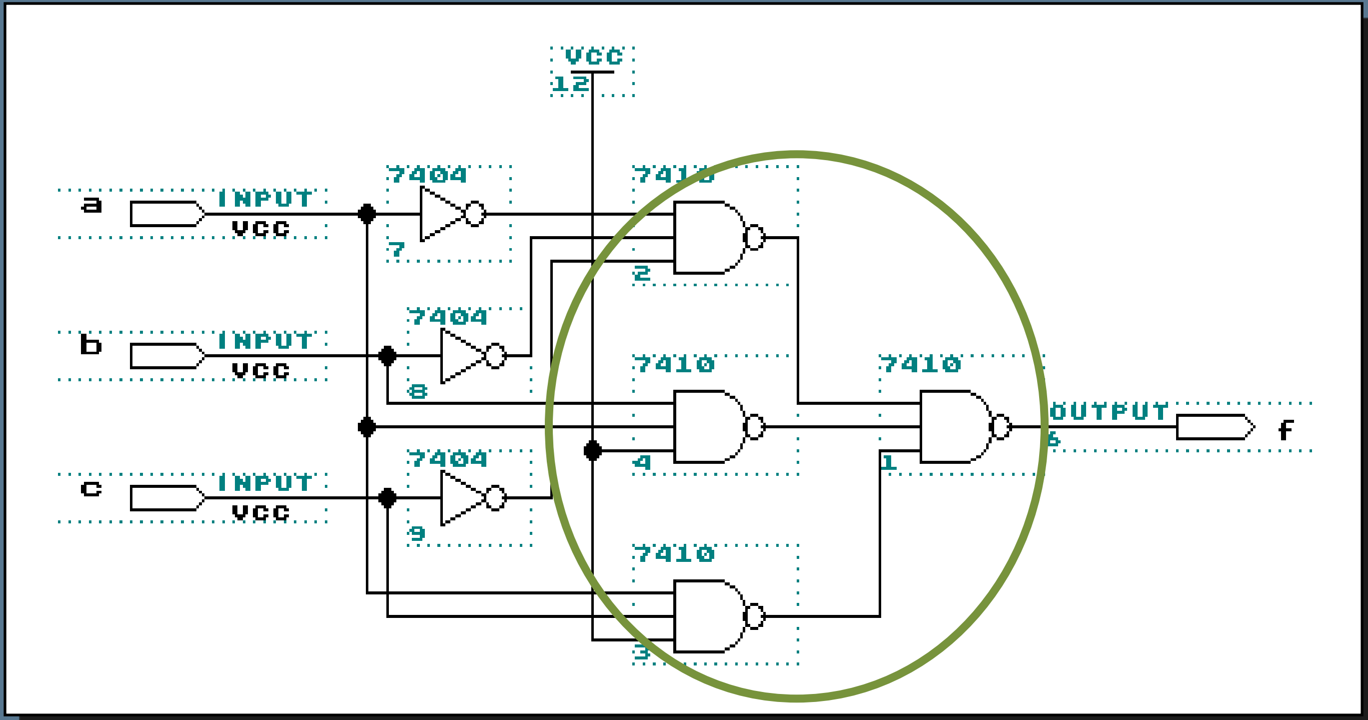
- Use a 3 input NOR gate with an inverter on the output for F = a'b'c' + ac + ab



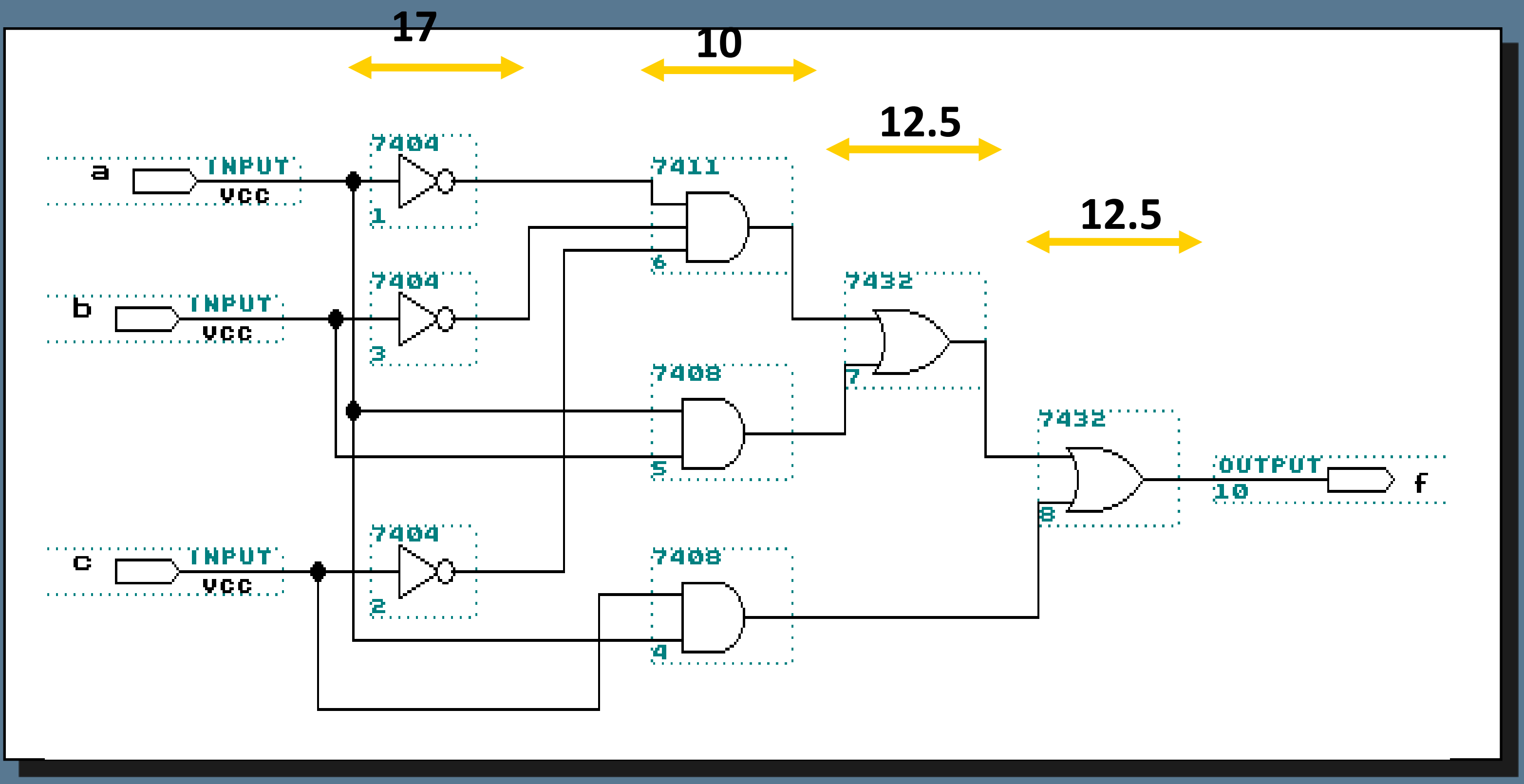
Solution Three

- Use NAND gates instead of AND-ORs for

  $F = a'b'c' + ac + ab$



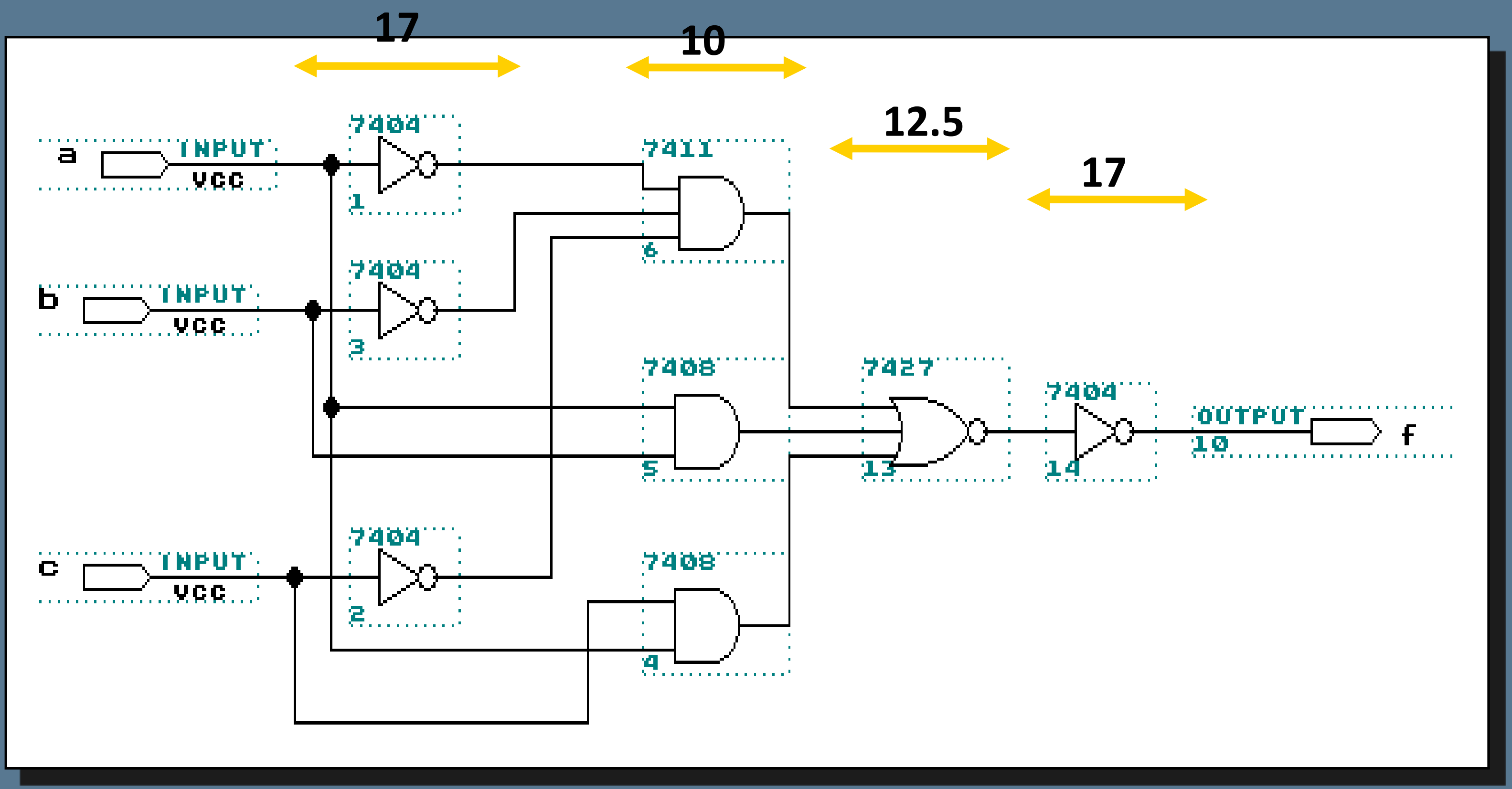
Solution One



SPEED
(worst case/ave)

Total = 52 nsec

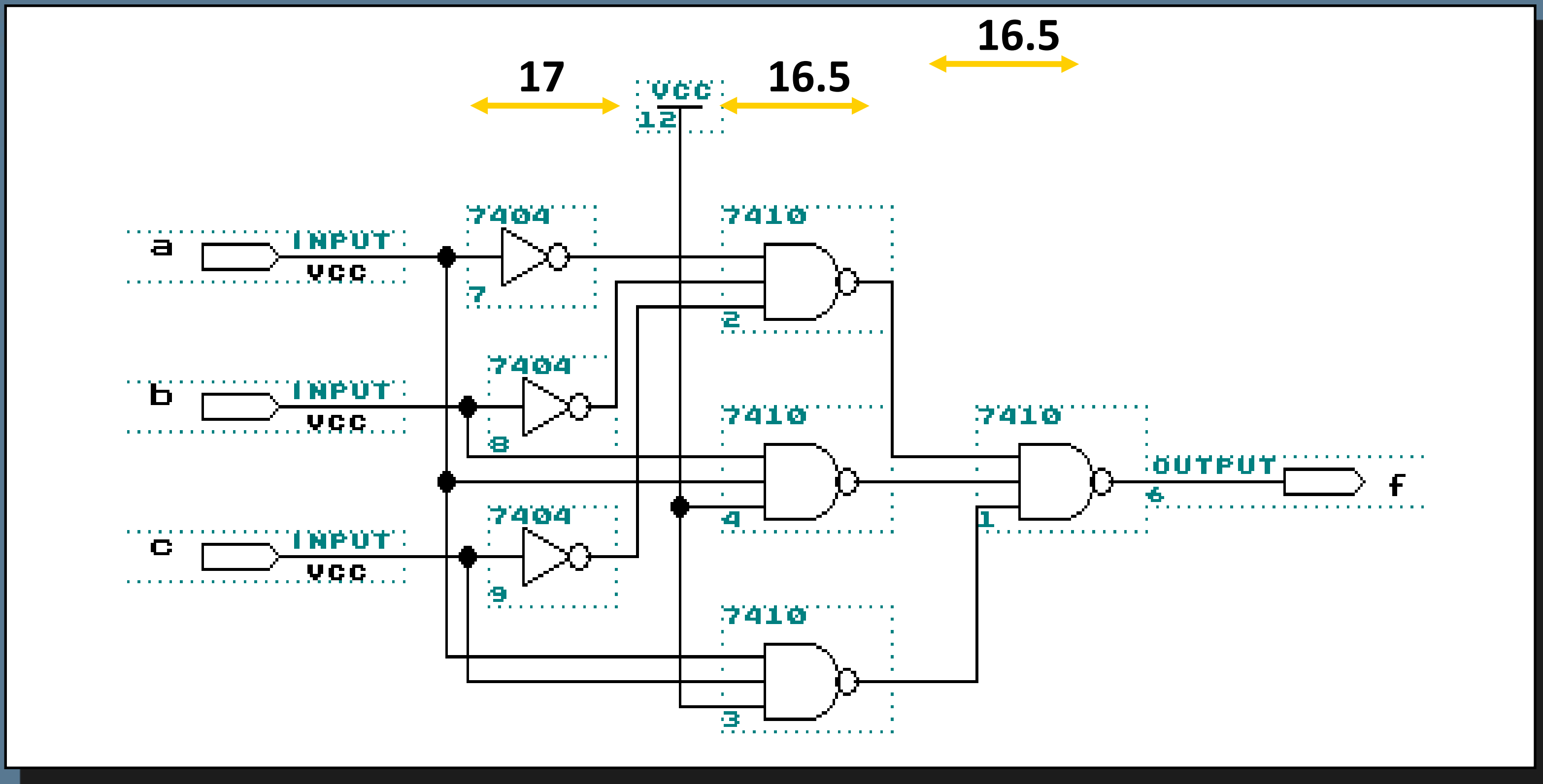
Solution Two



SPEED
(worst case/typical)

Total = 56.5 nsec

Solution Three



SPEED
(worst case/typical)

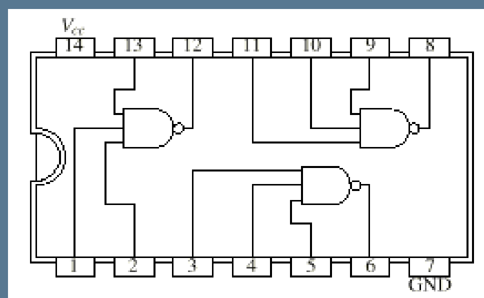
Total = 50 nsec

Prototype

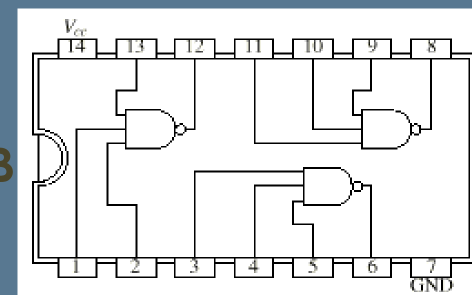
$F = a'b'c' + ab + ac$

a b c

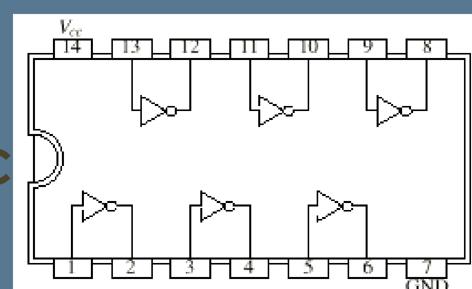
A



B

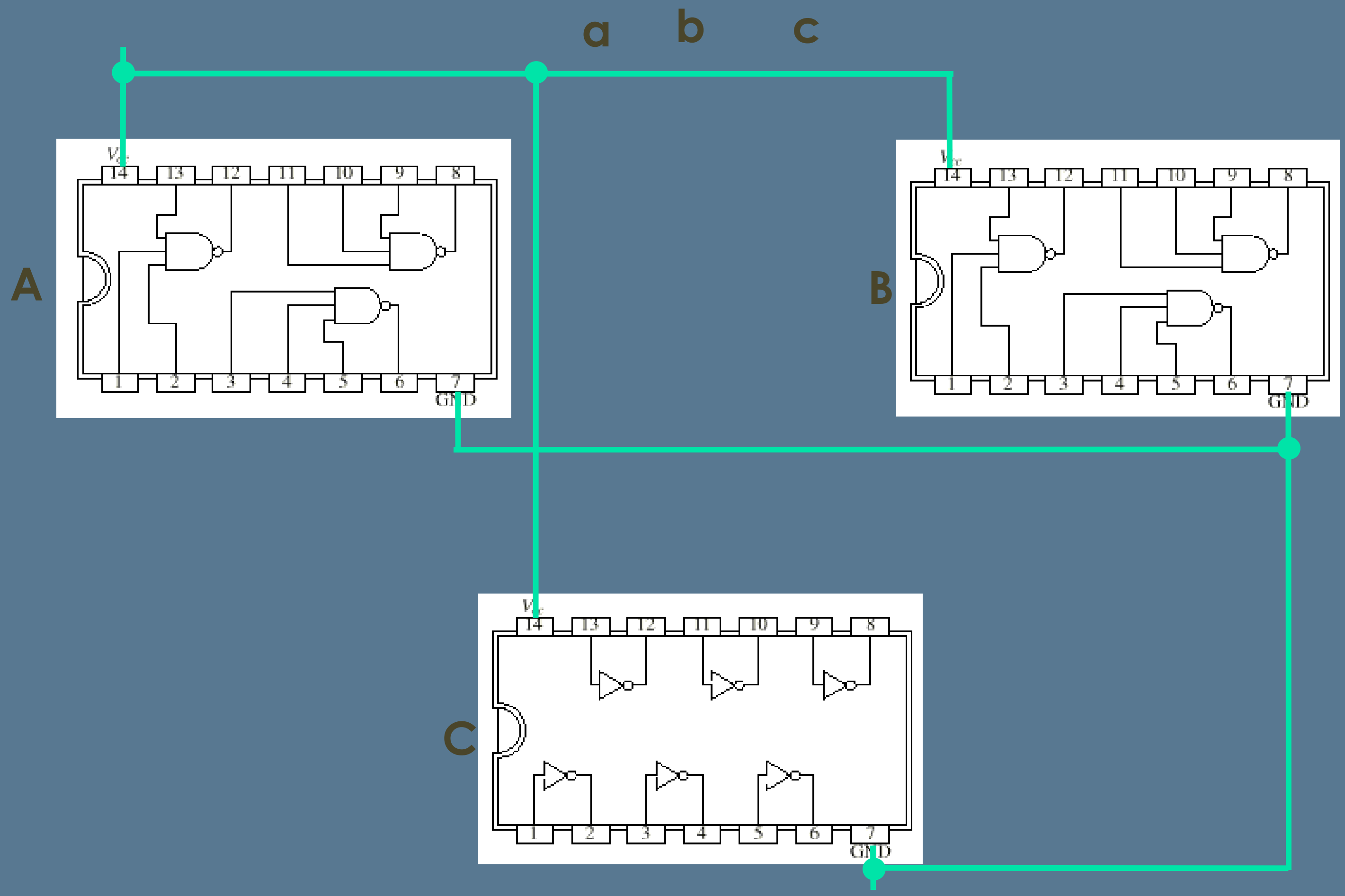


C



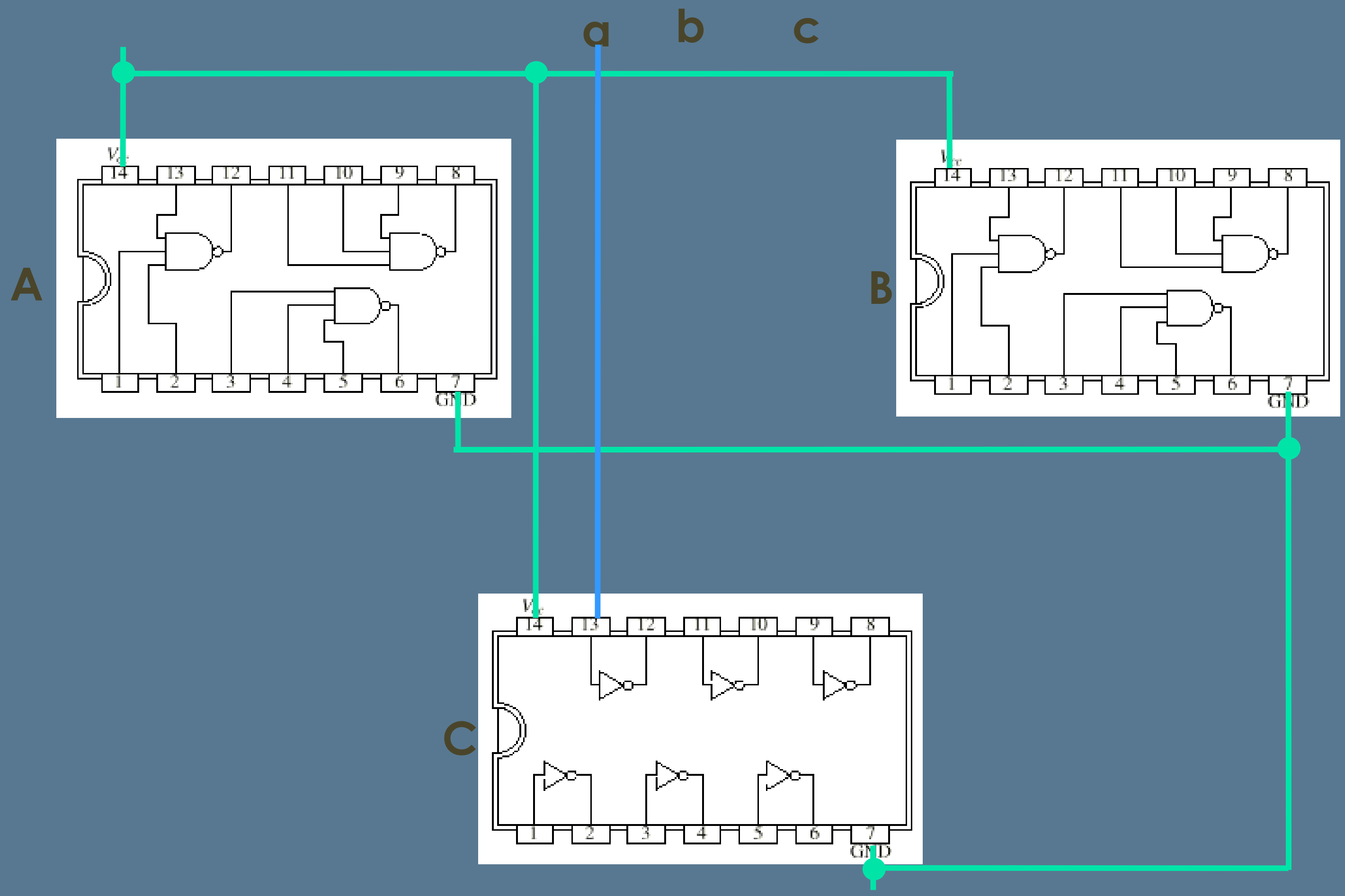
Prototype

$$F = a'b'c' + ab + ac$$



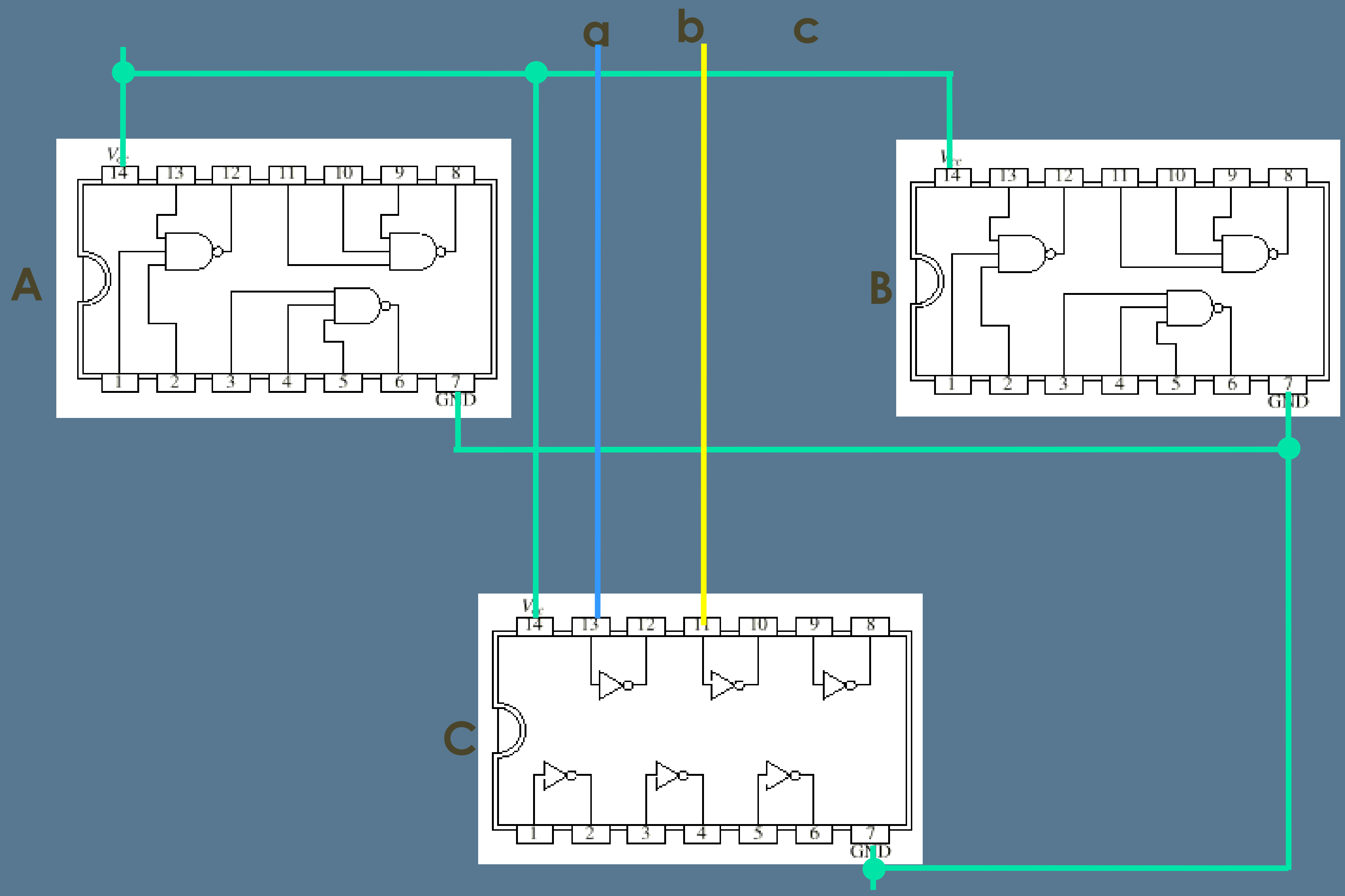
Prototype

$F = a'b'c' + ab + ac$



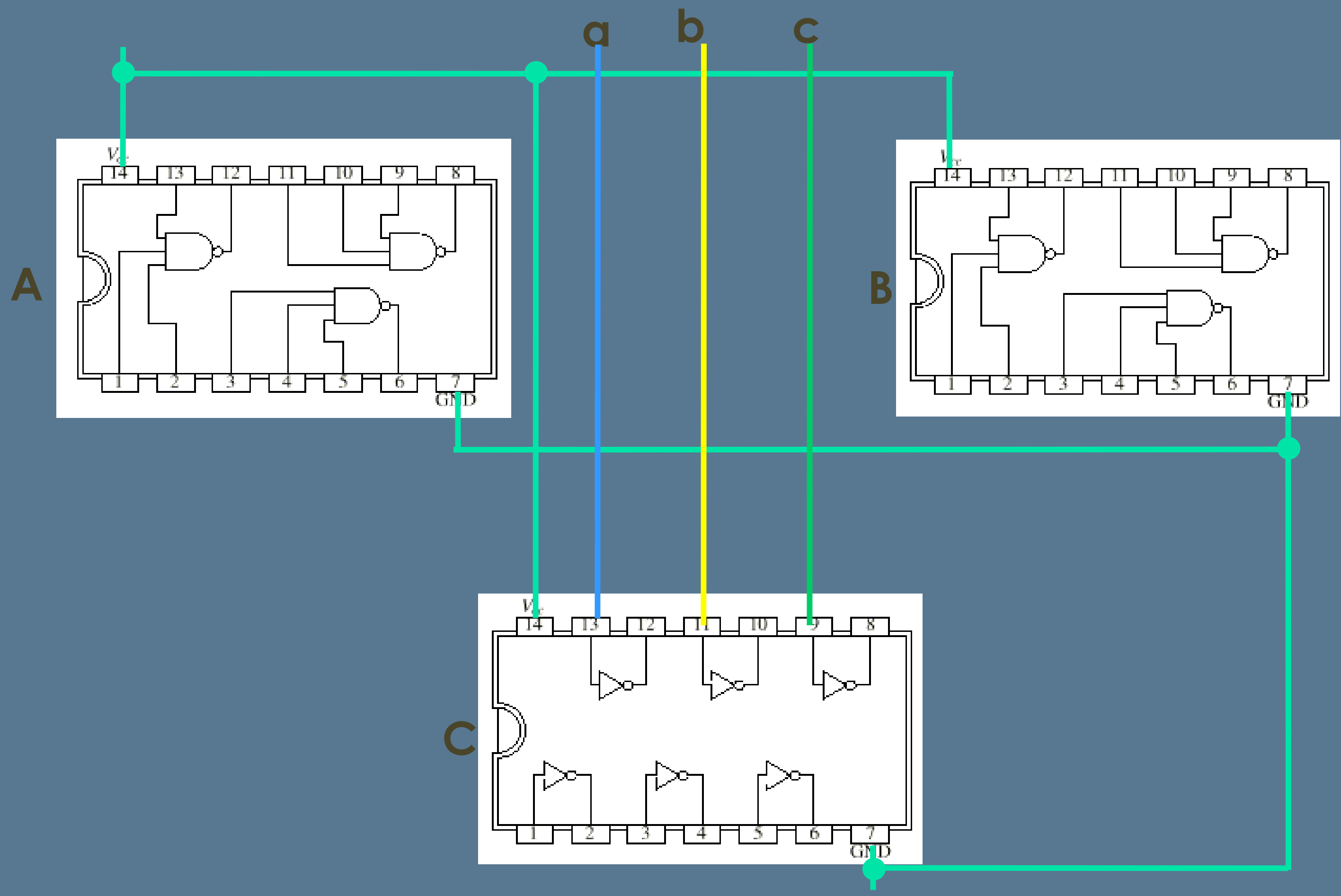
Prototype

$F = a'b'c' + ab + ac$



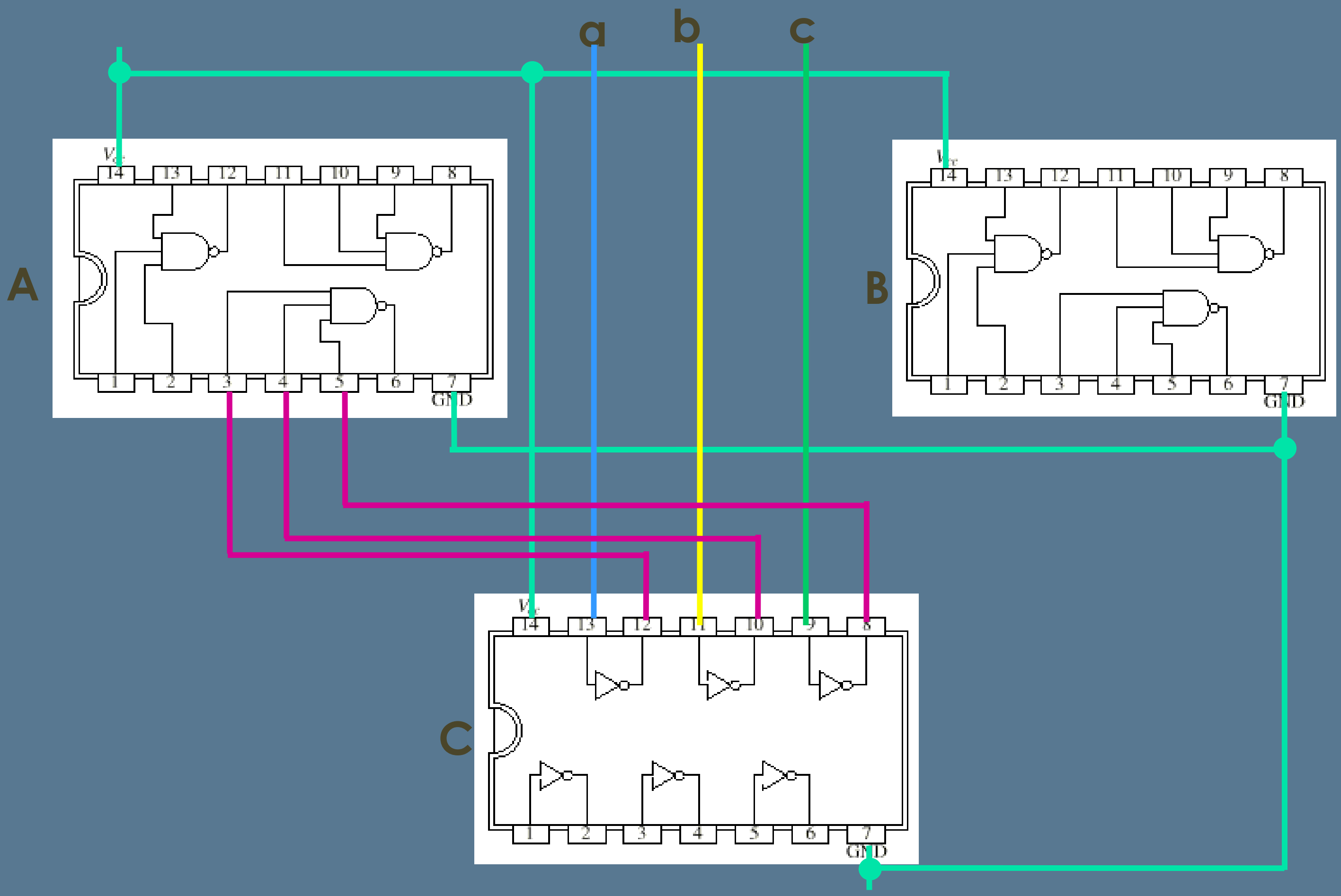
Prototype

$F = a'b'c' + ab + ac$



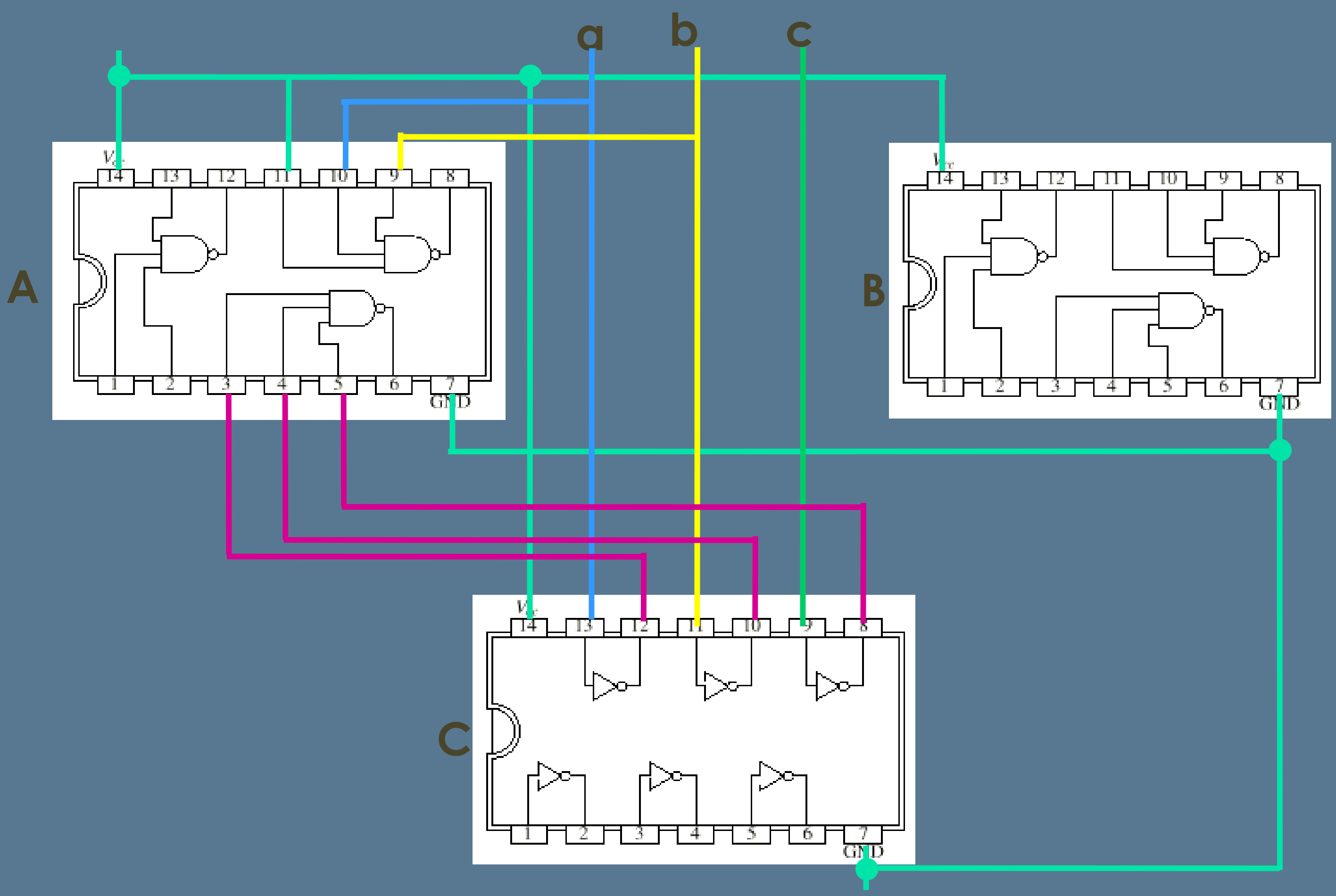
Prototype

$F = a'b'c' + ab + ac$



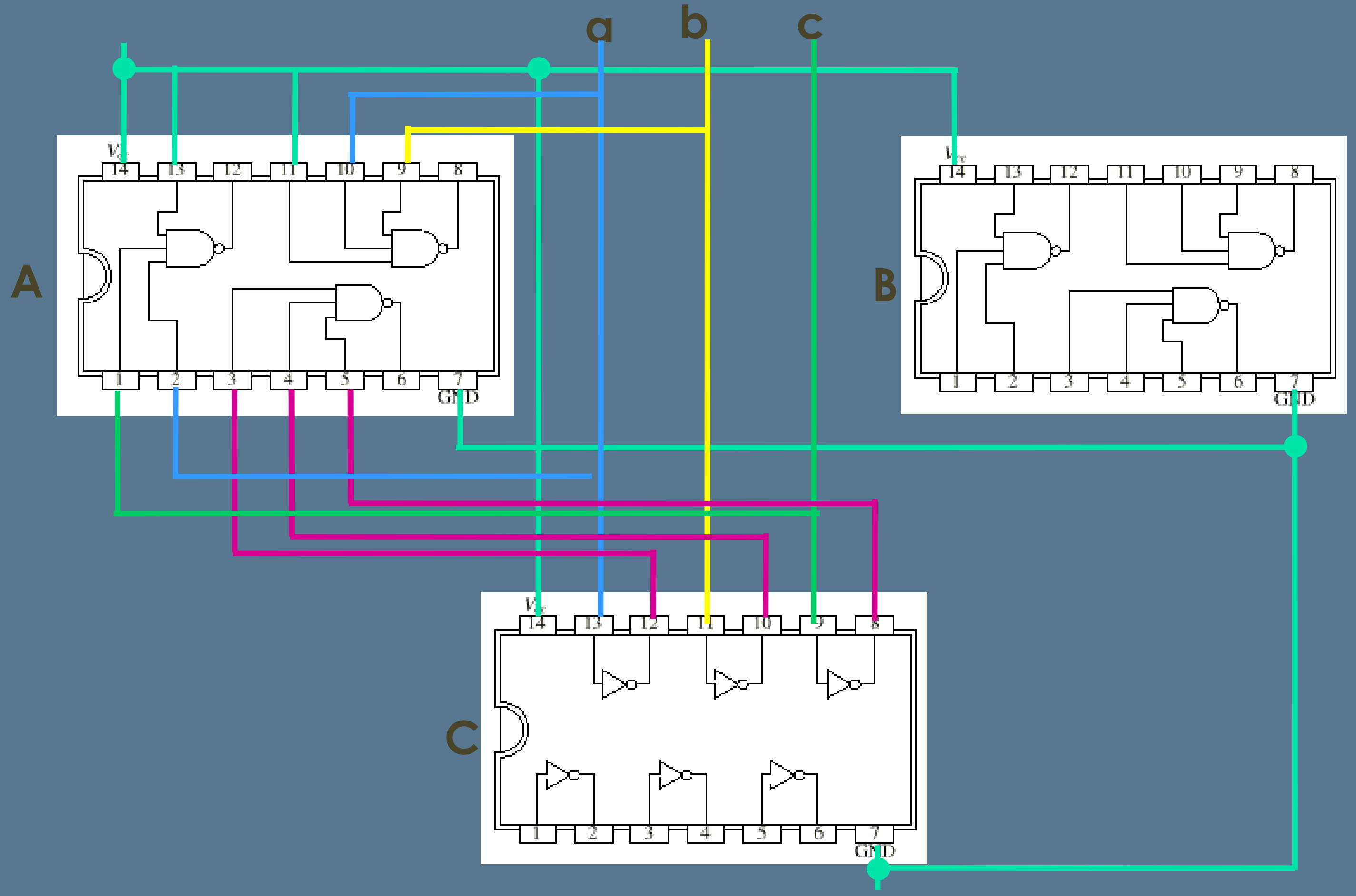
Prototype

$F = a'b'c' + ab + ac$



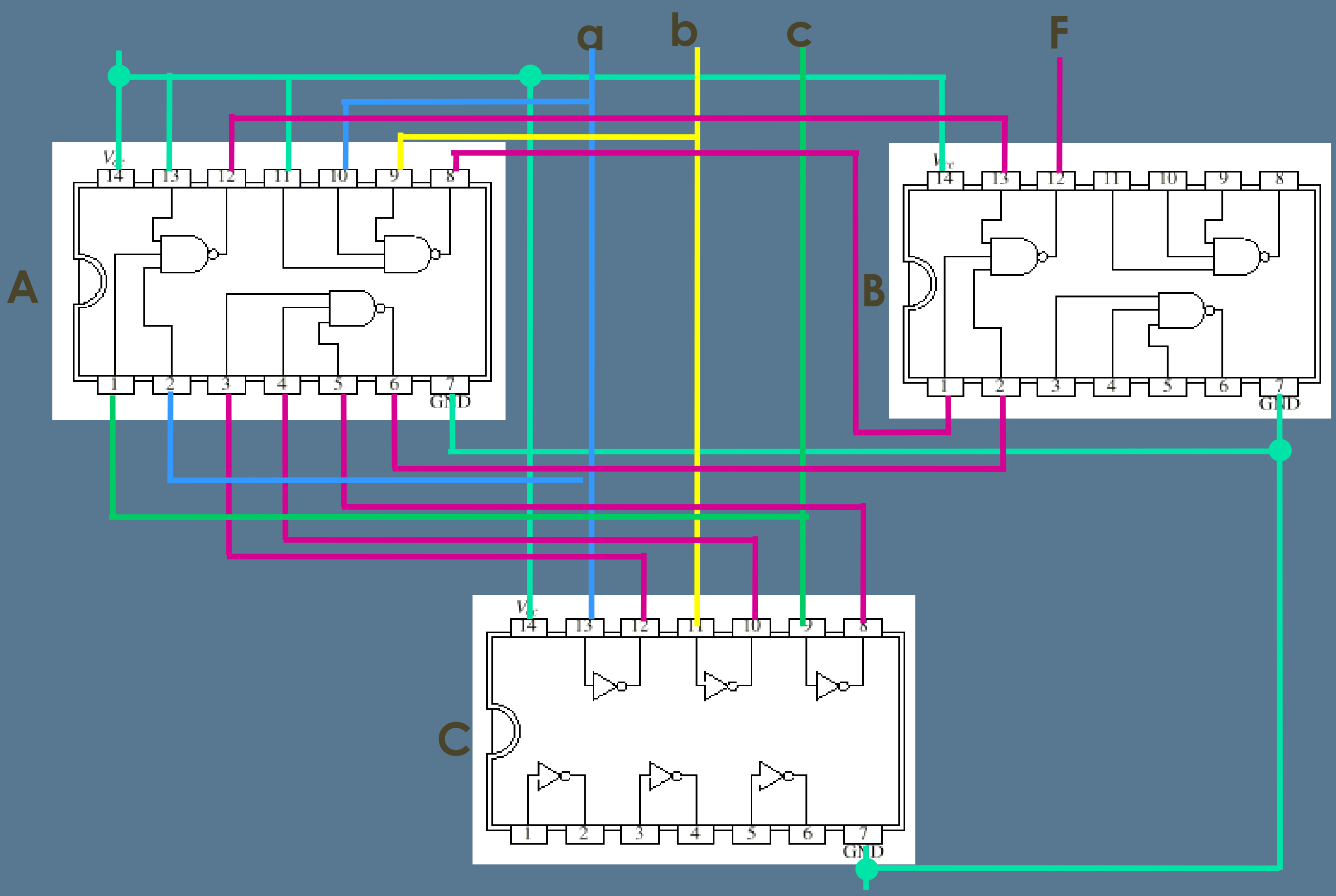
Prototype

$F = a'b'c' + ab + ac$

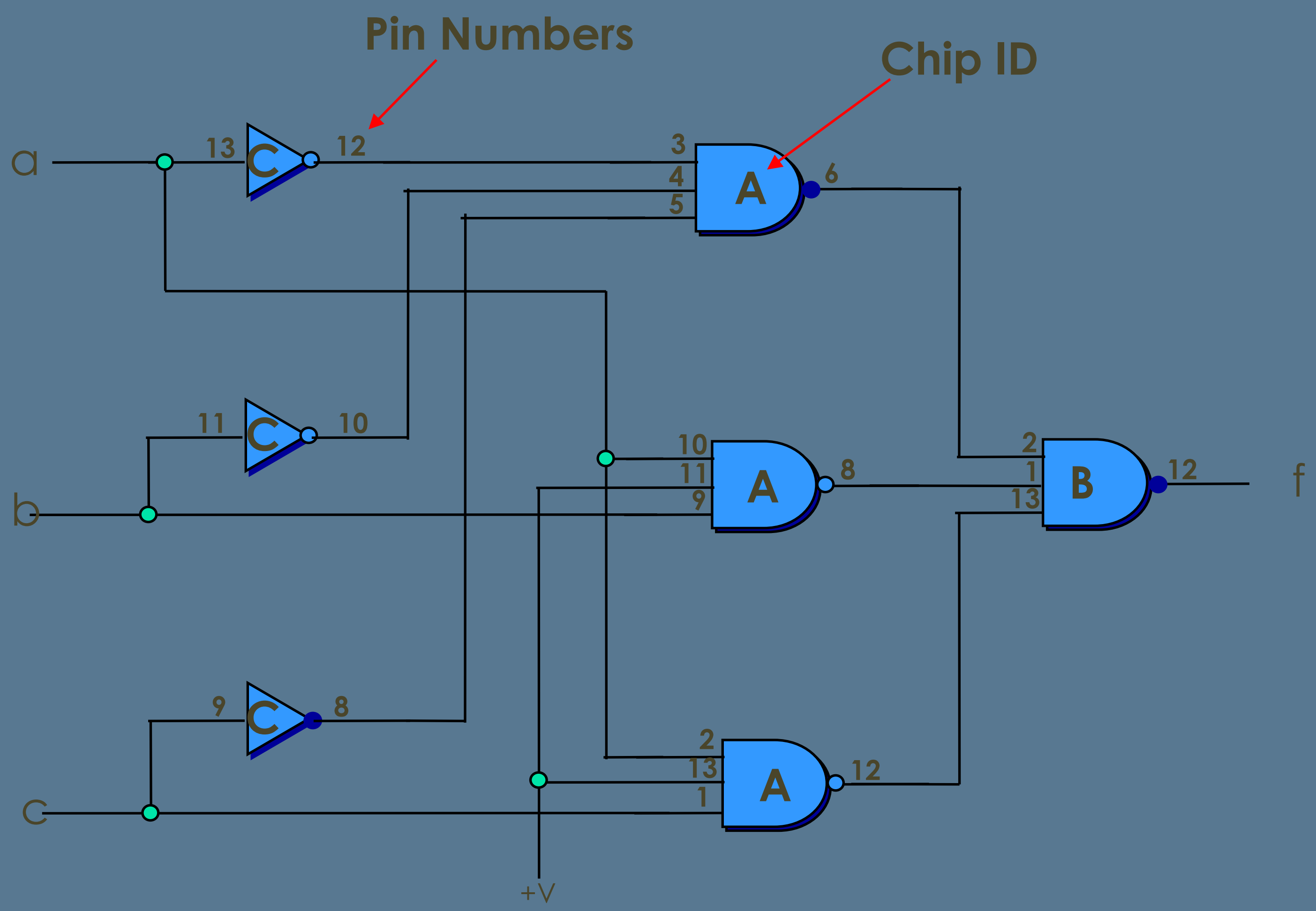


Prototype

$F = a'b'c' + ab + ac$

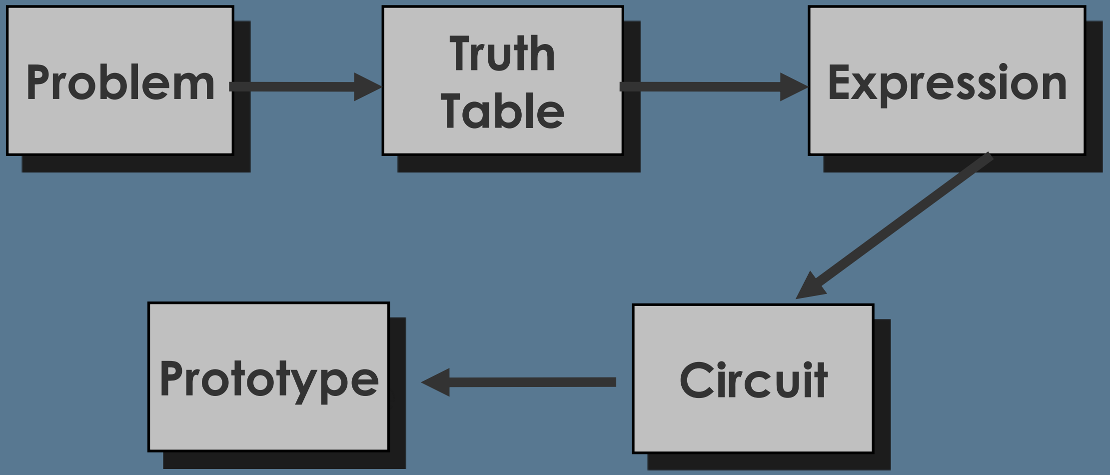


Schematic Annotation



Design Summary

- Design involves translating a problem from one representation to another



More Design Examples

1. A four-input function Z has the value 1 if and only if either of the following two conditions holds:
   a. All the inputs have the same value
   b. Half the inputs are 0s and half are 1s

1. Truth Table

A B C D	Z	A B C D	Z
0 0 0 0		1 0 0 0	
0 0 0 1		1 0 0 1	
0 0 1 0		1 0 1 0	
0 0 1 1		1 0 1 1	
0 1 0 0		1 1 0 0	
0 1 0 1		1 1 0 1	
0 1 1 0		1 1 1 0	
0 1 1 1		1 1 1 1	

1. Truth Table

A B C D	Z
0 0 0 0	1
0 0 0 1	0
0 0 1 0	0
0 0 1 1	1
0 1 0 0	0
0 1 0 1	1
0 1 1 0	1
0 1 1 1	0

A B C D	Z
1 0 0 0	0
1 0 0 1	1
1 0 1 0	1
1 0 1 1	0
1 1 0 0	1
1 1 0 1	0
1 1 1 0	0
1 1 1 1	1

More Design Examples

1. A four-input function Z has the value 1 if and only if either of the following two conditions holds:
   a. All the inputs have the same value
   b. Half the inputs are 0s and half are 1s

2. An alarm circuit contains 4 switches. The alarm A is sounded if at least two of the switches are on.

2. Truth Table

w x y z	A
0 0 0 0	
0 0 0 1	
0 0 1 0	
0 0 1 1	
0 1 0 0	
0 1 0 1	
0 1 1 0	
0 1 1 1	

w x y z	A
1 0 0 0	
1 0 0 1	
1 0 1 0	
1 0 1 1	
1 1 0 0	
1 1 0 1	
1 1 1 0	
1 1 1 1	

2. Truth Table

w x y z	A
0 0 0 0	0
0 0 0 1	0
0 0 1 0	0
0 0 1 1	1
0 1 0 0	0
0 1 0 1	1
0 1 1 0	1
0 1 1 1	1

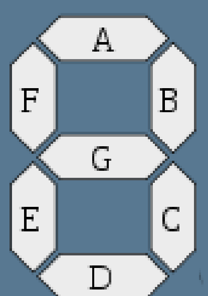
w x y z	A
1 0 0 0	0
1 0 0 1	1
1 0 1 0	1
1 0 1 1	1
1 1 0 0	1
1 1 0 1	1
1 1 1 0	1
1 1 1 1	1

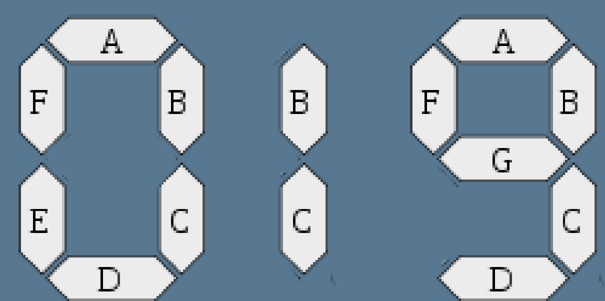
More Design Examples

3. A BCD-to-seven-segment converter is a combinational circuit that accepts a decimal digit in BCD and generates the appropriate outputs for selection of segments in a display indicator used for displaying the decimal digit. The outputs of the converter select the corresponding segments in the display as shown in the figure below. No segments should be selected for invalid inputs. Design the BCD-to-seven-segment circuit.

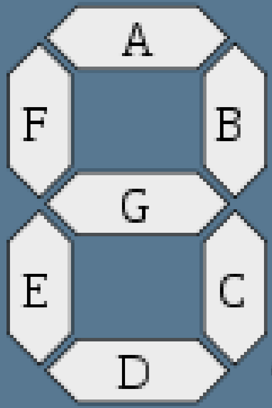
More Design Examples

3. A BCD-to-seven-segment converter is a combinational circuit that accepts a decimal digit in BCD and generates the appropriate outputs for selection of segments in a display indicator used for displaying the decimal digit. The outputs of the converter select the corresponding segments in the display as shown in the figure below. No segments should be selected for invalid inputs. Design the BCD-to-seven-segment circuit.



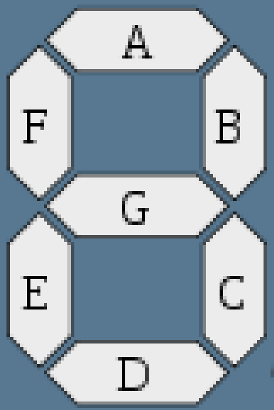


3. Truth Table



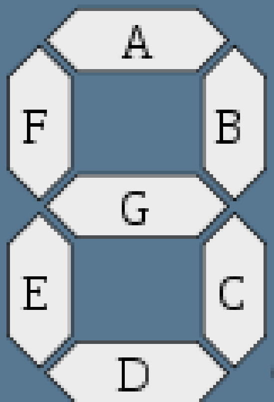
w	x	y	z	a	b	c	d	e	f	g
0	0	0	0							
0	0	0	1							
0	0	1	0							
0	0	1	1							
0	1	0	0							
0	1	0	1							
0	1	1	0							
0	1	1	1							

3. Truth Table



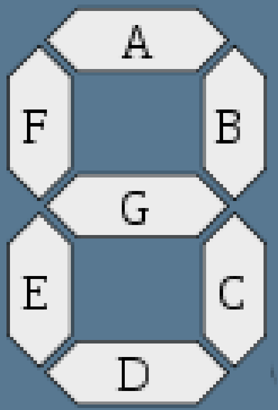
w x y z	a b c d e f g
0 0 0 0	1 1 1 1 1 1 0
0 0 0 1	
0 0 1 0	
0 0 1 1	
0 1 0 0	
0 1 0 1	
0 1 1 0	
0 1 1 1	

3. Truth Table



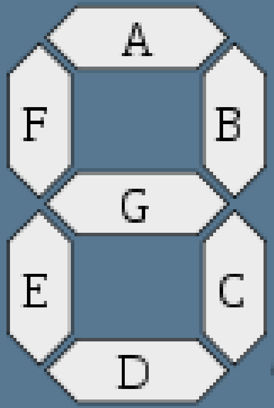
w	x	y	z	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0							
0	0	1	1							
0	1	0	0							
0	1	0	1							
0	1	1	0							
0	1	1	1							

3. Truth Table



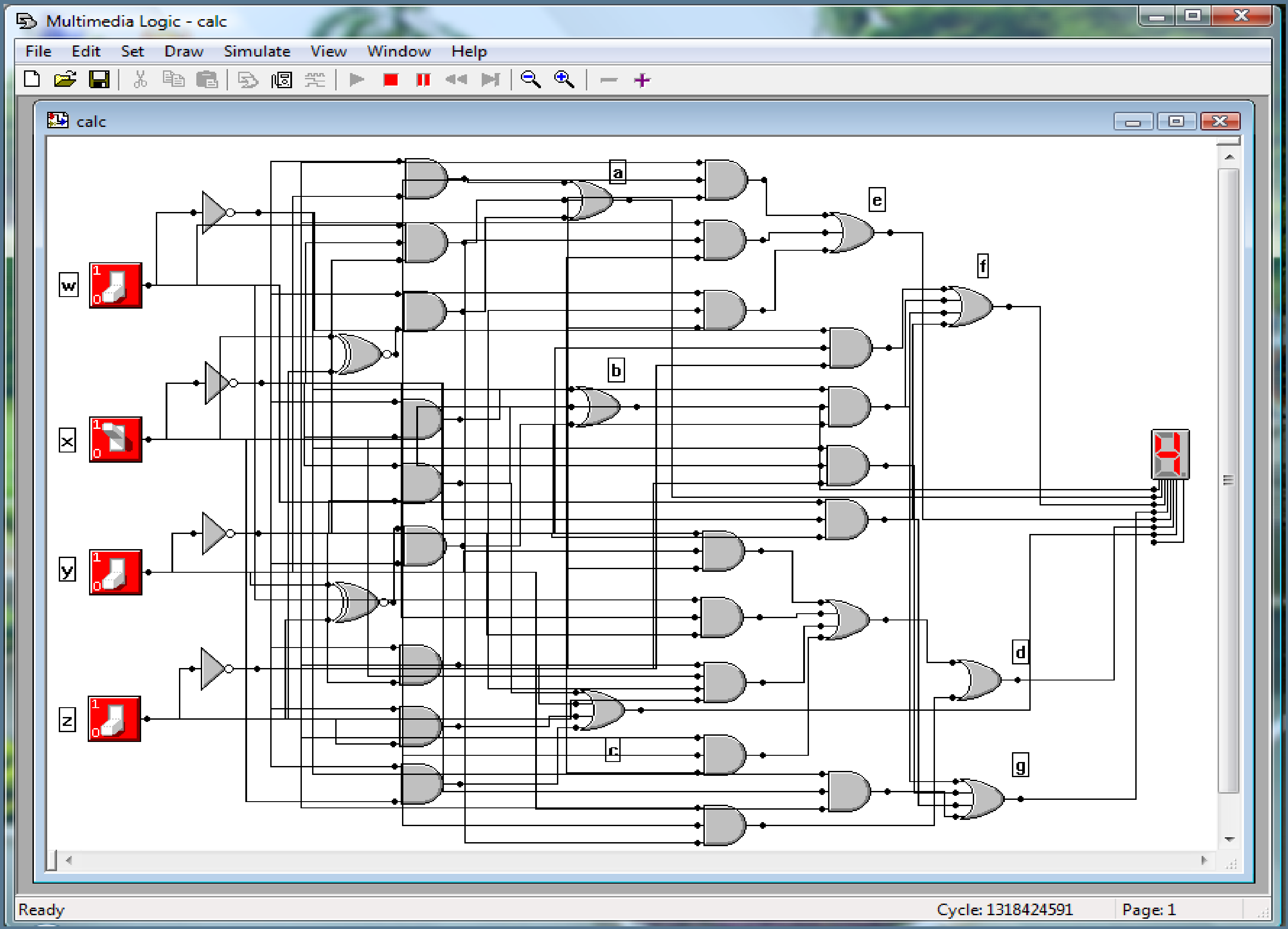
w	x	y	z	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0

3. Truth Table



w	x	y	z	a	b	c	d	e	f	g
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1
1	0	1	0	0	0	0	0	0	0	0
1	0	1	1	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0
1	1	0	1	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0

BCD-to-seven-segment converter



More Design Examples

4. Design a parity checker logical circuit with three inputs A, B, and C. The output should be high only when there is an even number of 1's in the input.

More Design Examples

4. Design a parity checker logical circuit with three inputs A, B, and C. The output should be high only when there is an even number of 1's in the input.

a b c	F
0 0 0	
0 0 1	
0 1 0	
0 1 1	
1 0 0	
1 0 1	
1 1 0	
1 1 1	

More Design Examples

4. Design a parity checker logical circuit with three inputs A, B, and C. The output should be high only when there is an even number of 1's in the input.

a b c	F
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	0

More Design Examples

4. Design a parity checker logical circuit with three inputs A, B, and C. The output should be high only when there is an even number of 1's in the input.

a b c	F
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	0

$$F = a'bc + ab'c + abc'$$

$$= (a'b + ab')c + abc'$$

More Design Examples

5. Design a circuit which adds two one-bit binary number.

More Design Examples

5. Design a circuit which adds two one-bit binary number.

A	B	
0	0	
0	1	
1	0	
1	1	

More Design Examples

5. Design a circuit which adds two one-bit binary number.

A	B	C_{out}	Sum
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

More Design Examples

5. Design a circuit which adds two one-bit binary number.

A	B	C_{out}	Sum
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

$$Sum = A \oplus B$$

$$C_{out} = AB$$