12.1 Two-level combinational logic simplification

Simplifying a sum-of-products expression

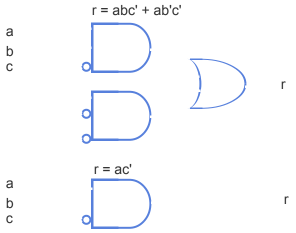
Logic simplification (also referred to as ***logic minimization***) means to simplify a Boolean expression before converting to a circuit to yield a smaller circuit.

PARTICIPATION ACTIVITY 12.1.1: Simplifying an expression before converting to a circuit.

Start □ 2x speed

A store's entry system should activate ringer (output r = 1) if motion is sensed (input a = 1) and daylight is detected (input b = 1) and clerk is not present (input c = 0), OR if motion sensed and no daylight detected and clerk not present.

r = a b c' + a b' c'
r = a c' (b + b') Distributive
r = a c' (1) Complement
r = a c' Identity



PARTICIPATION ACTIVITY 12.1.2: Simplifying a sum-of-products expression.

Consider the example above.

1) How many literals exist in the original expression?

○ 3
○ 6

2) If each AND or OR gate input requires two transistors, how many transistors does the original expression's circuit require? Ignore NOT gates.

○ 8
○ 16

3) If each AND or OR gate input requires two transistors, how many transistors does the simplified expression's circuit require? Ignore NOT gates.

○ 4
○ 16

Seeking i(j + j') opportunities

Given a sum-of-products expression, knowing what to simplify can be hard. To make simplification opportunities more obvious, a common algebraic simplification process is to:

- Convert to sum-of-minterms
- Seek i(j + j') opportunities: ij + ij' = i(j + j') = i

PARTICIPATION ACTIVITY 12.1.3: Seeking i(j + j') opportunities.

Start □ 2x speed

Original expression: ac + ab'c
ac(1) + ab'c
ac(b + b') + ab'c
abc + ab' c + ab'c
abc + ab' c
ac (b + b')
ac (1)
ac

PARTICIPATION ACTIVITY 12.1.4: Seeking i(j + j') simplification opportunities.

Only type the ? part. Type answers as: ab'

1) y = cd + cd'
y = c(?)

Check Show answer

2) y = c(d + d')
y = c(?)

Check Show answer

3) y = c(1)
y = ?

Check Show answer

4) y = efg + ef'g
y = eg(?)

Check Show answer

5) y = cd' + cd
y = c(?)

Check Show answer

6) y = dc + d'c
y = c(?)

Check Show answer

PARTICIPATION ACTIVITY 12.1.5: First translating to sum-of-minterms, then seeking simplification opportunities.

Simplify. Only type the ? part. Type answers as: ab'

1) y = cd + *c*
y = cd + *c(d + ?)*

Check Show answer

2) y = cd + c
y = cd + *c(d + d')*
y = cd + *cd + ?*

Check Show answer

3) y = *cd + cd* + cd'
y = *?* + cd'

Check Show answer

4) y = cd + cd'
y = c(?)

Check Show answer

5) y = c(d + d')
y = ?

Check Show answer

Algebraic simplification by hand can be hard

The algebraic simplification process can be hard to do by hand.

PARTICIPATION ACTIVITY 12.1.6: Algebraic simplification can be hard to do by hand.

Start □ 2x speed

Original expression: ab + a'

$$\begin{aligned}
&ab + a'\\
&ab + a'(b + b')\\
&ab + a'b + a'b'\\
&\underbrace{ab + a'b} + \underbrace{a'b + a'b'}\\
&(a + a')b + a'(b + b')\\
&(1)b + a'(1)\\
&a' + b
\end{aligned}$$

PARTICIPATION ACTIVITY 12.1.7: Simplifying algebraically can be hard.

Consider the example above.

1) What expression came after: ab + a'b + a'b'

○ (a + a')b + a'b'

○ ab + a'(b + b')

○ ab + a'b + a'b + a'b'

2) How many equations were written during the simplification process?

○ 3

○ 7

○ 21

Provide feedback on this section