step 1: investigate $A \equiv \neg B$

A	B	$\neg B$	$A \equiv \neg B$	$\neg (A \equiv \neg B)$
0	0	1	0	1
0	1	0	1	0
1	0	1	1	0
1	1	0	0	1

$\neg (A \equiv \neg B) \;=\; \neg A \wedge \neg B \;\vee\; A \wedge B$
$A \equiv \neg B \;=\; \neg (\neg A \wedge \neg B \;\vee\; A \wedge B)$

De Morgan's law (1): $\neg (X \vee Y)$ is the same as $(\neg X) \wedge (\neg Y)$

$A \equiv \neg B \;=\; (\neg(\neg A \wedge \neg B)) \;\wedge\; \neg(A \wedge B)$

De Morgan's law (2): $\neg (X \wedge Y)$ is the same as $(\neg X) \vee (\neg Y)$
applied to the first term

$A \equiv \neg B \;=\; (\neg\neg A \vee \neg\neg B) \;\wedge\; \neg(A \wedge B)$
$A \equiv \neg B \;=\; (A \vee B) \;\wedge\; \neg(A \wedge B)$

applied to the second term

$$A \equiv \neg B \;=\; (A \vee B) \;\wedge\; (\neg A \vee \neg B)$$

step 2: investigate $A \equiv B \wedge C$

A	B	C	$B\wedge C$	$A \equiv B\wedge C$	$\neg (A \equiv B\wedge C)$
0	0	0	0	1	0
0	0	1	0	1	0
0	1	0	0	1	0
0	1	1	1	0	1
1	0	0	0	0	1
1	0	1	0	0	1
1	1	0	0	0	1
1	1	1	1	1	0

$\neg (A \equiv B\wedge C) = \neg A\wedge B\wedge C \vee A\wedge\neg B\wedge\neg C \vee A\wedge\neg B\wedge C \vee A\wedge B\wedge\neg C$
$A \equiv B\wedge C = \neg (\neg A\wedge B\wedge C \vee A\wedge\neg B\wedge\neg C \vee A\wedge\neg B\wedge C \vee A\wedge B\wedge\neg C)$

De Morgan's law (1): $\neg (X \vee Y)$ is the same as $(\neg X) \wedge (\neg Y)$

$A \equiv B\wedge C = (\neg(\neg A\wedge B\wedge C)) \wedge (\neg(A\wedge\neg B\wedge\neg C)) \wedge (\neg(A\wedge\neg B\wedge C)) \wedge (\neg(A\wedge B\wedge\neg C))$

De Morgan's law (2): $\neg (X \wedge Y)$ is the same as $(\neg X) \vee (\neg Y)$
applied to all three terms

$A \equiv B\wedge C = (\neg\neg A\wedge\neg B\wedge\neg C) \wedge (\neg A\wedge\neg\neg B\wedge\neg\neg C) \wedge (\neg A\wedge\neg\neg B\wedge\neg C) \wedge (\neg A\wedge\neg B\wedge\neg\neg C)$

$$A \equiv B\wedge C = (A\vee\neg B\vee\neg C) \wedge (\neg A\vee B\vee C) \wedge (\neg A\vee B\vee\neg C) \wedge (\neg A\vee\neg B\vee C)$$

step 3: investigate $A \equiv B \vee C$

A	B	C	$B \vee C$	$A \equiv B \vee C$	$\neg (A \equiv B \vee C)$
0	0	0	0	1	0
0	0	1	1	0	1
0	1	0	1	0	1
0	1	1	1	0	1
1	0	0	0	0	1
1	0	1	1	1	0
1	1	0	1	1	0
1	1	1	1	1	0

$\neg (A \equiv B \vee C) = \neg A \wedge \neg B \wedge C \vee \neg A \wedge B \wedge \neg C \vee \neg A \wedge B \wedge C \vee A \wedge \neg B \wedge \neg C$
$A \equiv B \vee C = \neg (\neg A \wedge \neg B \wedge C \vee \neg A \wedge B \wedge \neg C \vee \neg A \wedge B \wedge C \vee A \wedge \neg B \wedge \neg C)$

De Morgan's law (1): $\neg (X \vee Y)$ is the same as $(\neg X) \wedge (\neg Y)$

$A \equiv B \vee C = (\neg(\neg A \wedge \neg B \wedge C)) \wedge (\neg(\neg A \wedge B \wedge \neg C)) \wedge (\neg(\neg A \wedge B \wedge C)) \wedge (\neg(A \wedge \neg B \wedge \neg C))$

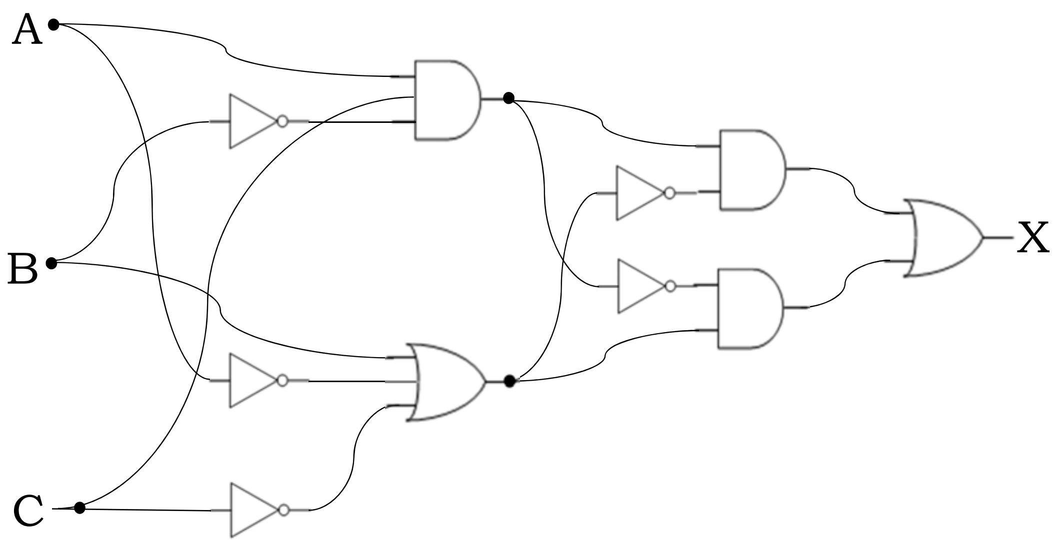
De Morgan's law (2): $\neg (X \wedge Y)$ is the same as $(\neg X) \vee (\neg Y)$
applied to all three terms

$A \equiv B \vee C = (\neg\neg A \wedge \neg\neg B \wedge \neg C) \wedge (\neg\neg A \wedge \neg B \wedge \neg\neg C) \wedge (\neg\neg A \wedge \neg B \wedge \neg C) \wedge (\neg A \wedge \neg\neg B \wedge \neg\neg C)$

$$A \equiv B \vee C = (A \vee B \vee \neg C) \wedge (A \vee \neg B \vee C) \wedge (A \vee \neg B \vee \neg C) \wedge (\neg A \vee B \vee C)$$

step 4: take a BOOL-SAT problem

Any one at all

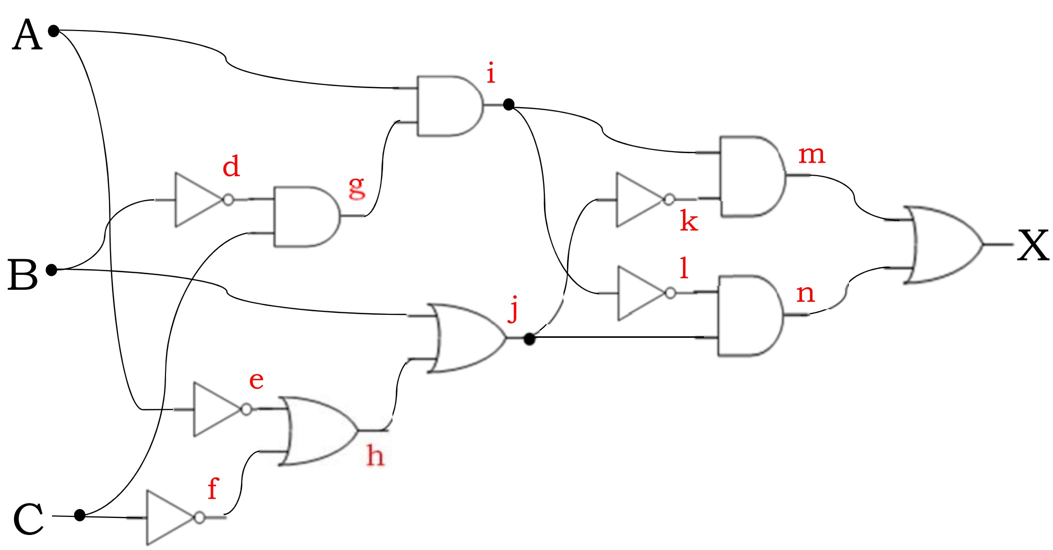


(This one doesn't look very good)

Replace all gates that have more than two inputs with the equivalent combination of two input gates, and

Label every gate's output.

step 5: Turn it into a lot of equivalences



$d \equiv \neg B$

$e \equiv \neg A$

$f \equiv \neg C$

$g \equiv d \wedge C$

$h \equiv e \vee f$

$i \equiv A \wedge g$

$j \equiv B \vee h$

$k \equiv \neg j$

$l \equiv \neg i$

$m \equiv i \wedge k$

$n \equiv j \wedge l$

$X \equiv m \vee n$

step 6: Combine them

To describe the circuit in a single formula, we need to insist that all those equivalencies are true.

To express the boolean satisfiability problem, we need to insist that the output is true too.

$$
\begin{aligned}
&(d \equiv \neg B) \\
&\wedge (e \equiv \neg A) \\
&\wedge (f \equiv \neg C) \\
&\wedge (g \equiv d \wedge C) \\
&\wedge (h \equiv e \vee f) \\
&\wedge (i \equiv A \wedge g) \\
&\wedge (j \equiv B \vee h) \\
&\wedge (k \equiv \neg j) \\
&\wedge (l \equiv \neg i) \\
&\wedge (m \equiv i \wedge k) \\
&\wedge (n \equiv j \wedge l) \\
&\wedge (X \equiv m \vee n) \\
&\wedge X
\end{aligned}
$$

step 7: Abracadabra!

Replace the equivalencies with their alternate forms discovered in steps 1 to 3.

$A \equiv \neg B \;=\; (A \vee B) \wedge (\neg A \vee \neg B)$

$A \equiv B \wedge C \;=\; (A \vee \neg B \vee \neg C) \wedge (\neg A \vee B \vee C) \wedge (\neg A \vee B \vee \neg C) \wedge (\neg A \vee \neg B \vee C)$

$A \equiv B \vee C \;=\; (A \vee B \vee \neg C) \wedge (A \vee \neg B \vee C) \wedge (A \vee \neg B \vee \neg C) \wedge (\neg A \vee B \vee C)$

$$
\begin{array}{ll}
(d \equiv \neg B) & (d \vee B) \wedge (\neg d \vee \neg B) \\
\wedge (e \equiv \neg A) & \wedge (e \vee A) \wedge (\neg e \vee \neg A) \\
\wedge (f \equiv \neg C) & \wedge (f \vee C) \wedge (\neg f \vee \neg C) \\
\wedge (g \equiv d \wedge C) & \wedge (g \vee \neg d \vee \neg C) \wedge (\neg g \vee d \vee C) \wedge (\neg g \vee d \vee \neg C) \wedge (\neg g \vee \neg d \vee C) \\
\wedge (h \equiv e \vee f) & \wedge (h \vee e \vee \neg f) \wedge (h \vee \neg e \vee f) \wedge (h \vee \neg e \vee \neg f) \wedge (\neg h \vee e \vee f) \\
\wedge (i \equiv A \wedge g) & \wedge (i \vee \neg A \vee \neg g) \wedge (\neg i \vee A \vee g) \wedge (\neg i \vee A \vee \neg g) \wedge (\neg i \vee \neg A \vee g) \\
\wedge (j \equiv B \vee h) \quad \longrightarrow & \wedge (j \vee B \vee \neg h) \wedge (j \vee \neg B \vee h) \wedge (j \vee \neg B \vee \neg h) \wedge (\neg j \vee B \vee h) \\
\wedge (k \equiv \neg j) & \wedge (k \vee j) \wedge (\neg k \vee \neg j) \\
\wedge (l \equiv \neg i) & \wedge (l \vee i) \wedge (\neg l \vee \neg i) \\
\wedge (m \equiv i \wedge k) & \wedge (m \vee \neg i \vee \neg k) \wedge (\neg m \vee i \vee k) \wedge (\neg m \vee i \vee \neg k) \wedge (\neg m \vee \neg i \vee k) \\
\wedge (n \equiv j \wedge l) & \wedge (n \vee \neg j \vee \neg C) \wedge (\neg n \vee j \vee C) \wedge (\neg n \vee j \vee \neg C) \wedge (\neg n \vee \neg j \vee C) \\
\wedge (X \equiv m \vee n) & \wedge (X \vee m \vee \neg n) \wedge (X \vee \neg m \vee n) \wedge (X \vee \neg m \vee \neg n) \wedge (\neg X \vee m \vee n) \\
\wedge X & \wedge X
\end{array}
$$

We've got one big formula in 3CNF.

If you really want to be truly in 3CNF, we can simply duplicate literals in terms that have only got one or two.

$(d \vee B)$	becomes	$(d \vee B \vee B)$
X	becomes	$(X \vee X \vee X)$

This formula in pure 3CNF

$$
\begin{aligned}
&(d \vee B \vee B) \wedge (\neg d \vee \neg B \vee \neg B) \\
&\wedge (e \vee A \vee A) \wedge (\neg e \vee \neg A \vee \neg A) \\
&\wedge (f \vee C \vee C) \wedge (\neg f \vee \neg C \vee \neg C) \\
&\wedge (g \vee \neg d \vee \neg C) \wedge (\neg g \vee d \vee C) \wedge (\neg g \vee d \vee \neg C) \wedge (\neg g \vee \neg d \vee C) \\
&\wedge (h \vee e \vee \neg f) \wedge (h \vee \neg e \vee f) \wedge (h \vee \neg e \vee \neg f) \wedge (\neg h \vee e \vee f) \\
&\wedge (i \vee \neg A \vee \neg g) \wedge (\neg i \vee A \vee g) \wedge (\neg i \vee A \vee \neg g) \wedge (\neg i \vee \neg A \vee g) \\
&\wedge (j \vee B \vee \neg h) \wedge (j \vee \neg B \vee h) \wedge (j \vee \neg B \vee \neg h) \wedge (\neg j \vee B \vee h) \\
&\wedge (k \vee j \vee j) \wedge (\neg k \vee \neg j \vee \neg j) \\
&\wedge (l \vee i \vee i) \wedge (\neg l \vee \neg i \vee \neg i) \\
&\wedge (m \vee \neg i \vee \neg k) \wedge (\neg m \vee i \vee k) \wedge (\neg m \vee i \vee \neg k) \wedge (\neg m \vee \neg i \vee k) \\
&\wedge (n \vee \neg j \vee \neg C) \wedge (\neg n \vee j \vee C) \wedge (\neg n \vee j \vee \neg C) \wedge (\neg n \vee \neg j \vee C) \\
&\wedge (X \vee m \vee \neg n) \wedge (X \vee \neg m \vee n) \wedge (X \vee \neg m \vee \neg n) \wedge (\neg X \vee m \vee n) \\
&\wedge (X \vee X \vee X)
\end{aligned}
$$

is exactly the same as the boolean satisfiability problem for the logic circuit.

The transformation was easy (once the method was worked out), and the number of terms in the 3CNF formula is no more than 4 times the number of gates in the logic circuit plus 1 (once the multi-input gates have been converted)

Therefore, if we could solve 3CNF-SAT efficiently, then we could solve BOOL-SAT efficiently.

3CNF-SAT solves BOOL-SAT.

A formula in 3CNF is just a boolean formula with restrictions on what you can say. Every 3CNF formula *is* a BOOL formula, so

BOOL-SAT solves 3CNF-SAT.

They are equally hard.

Restrict boolean logic so far that all you can have is a conjunction of disjunctions of three literals, and it remains just as difficult to solve. 3CNF-SAT is NP-complete.

Restrict the other way, so that all you can have is a disjunction of conjunctions of three literals, and it (3DNF-SAT) becomes trivial, just like DNF-SAT.

Take the restriction just one step further to 2CNF, and it also becomes trivial.