# COMSM1302 Lab Sheet 1 (Solutions)

## Simplification

- $0 \wedge 0 \equiv 0$
- $0 \wedge 1 \equiv 0$
- $1 \wedge 0 \equiv 0$
- $1 \wedge 1 \equiv 1$
- $0 \vee 0 \equiv 0$
- $0 \vee 1 \equiv 1$
- $1 \vee 0 \equiv 1$
- $1 \vee 1 \equiv 1$
- $0 \wedge A \equiv 0$
- $1 \wedge A \equiv A$
- $A \wedge A \equiv A$
- $\neg A \wedge A \equiv 0$
- $0 \vee A \equiv A$
- $1 \vee A \equiv 1$
- $A \vee A \equiv A$
- $\neg A \vee A \equiv 1$

The first column is just the values in the AND truth table, while the second column is the values in the OR truth table. You can prove the other simplifications by drawing out the truth tables e.g:

| $A$ | $0 \wedge A$ |
|---|---|
| 0 | 0 |
| 1 | 0 |

As $0 \wedge A$ outputs 0, no matter the value of $A$, it is equivalent to 0 i.e. always false.

| $A$ | $A \vee A$ |
|---|---|
| 0 | 0 |
| 1 | 1 |

As $A \vee A$ outputs the same value as $A$, it is equivalent to $A$.

## Circuit Assembly

Direct circuit implementations are shown in the Logisim solutions (subcircuit names detailed below), as well as the simplified "challenge" version. Note that these solutions use the least number of NOT, AND, and OR gates; there may be solutions that use fewer gates and use other 2-input gates such as NAND.

$((A \wedge C) \vee (B \wedge C)) \wedge C$ — For implementations see subcircuit *circuit1*
$((A \vee B) \wedge C) \wedge C$
⇒ Distributivity, taking out the common variable "$\wedge C$"
$(A \vee B) \wedge C$
⇒ $((A \vee B) \wedge C) \wedge C \equiv (A \vee B) \wedge (C \wedge C)$ and $C \wedge C \equiv C$

$\neg(A \vee \neg(B \wedge C))$ — For implementations see subcircuit *circuit2*
$\neg A \wedge (B \wedge C)$
⇒ De Morgan's Laws and double negation

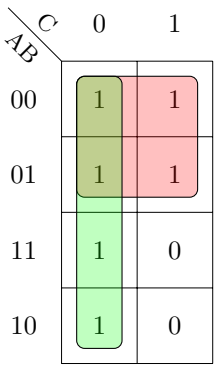
$((A \vee B) \wedge (A \vee \neg B)) \vee C$ — For implementations see subcircuit *circuit3*
$(A \vee (B \wedge \neg B)) \vee C$
⇒ Distributivity, taking out the common variable "$A \vee$"
$A \vee C$
⇒ $B \vee \neg B \equiv 0$ and $A \vee 0 \equiv A$

$(A \wedge B) \vee \neg(C \vee \neg A)$ — For implementations see subcircuit *circuit4*
$(A \wedge B) \vee (\neg C \wedge A)$
⇒ De Morgan's Laws and double negation
$A \wedge (B \vee \neg C)$
⇒ Distributivity, taking out the common variable "$A \wedge$"

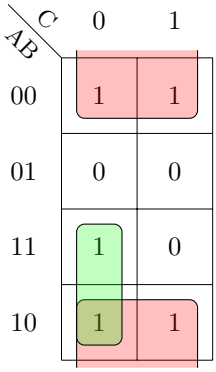
$A \vee (B \wedge (C \vee A))$ — For implementations see subcircuit *circuit5*
$(A \vee B) \wedge (A \vee (C \vee A))$
⇒ Distributivity, adding in the common variable "$A \vee$"
$(A \vee B) \wedge (A \vee C)$
⇒ $A \vee (C \vee A) \equiv (A \vee A) \vee C$ and $A \vee A \equiv A$
$A \vee (B \wedge C)$
⇒ Distributivity, taking out the common variable "$A \vee$"

$\neg(\neg(A \wedge B \wedge C) \wedge C)$ — For implementations see subcircuit *circuit6*
$(A \wedge B \wedge C) \vee \neg C$
⇒ De Morgan's Laws and double negation
$(A \vee \neg C) \wedge (B \vee \neg C) \wedge (C \vee \neg C)$
⇒ Distributivity, adding in the common variable "$\vee \neg C$"
$(A \vee \neg C) \wedge (B \vee \neg C)$
⇒ $C \vee \neg C \equiv 1$ and $(A \vee \neg C) \wedge (B \vee \neg C) \wedge 1 \equiv (A \vee \neg C) \wedge (B \vee \neg C)$
$(A \wedge B) \vee \neg C$
⇒ Distributivity, taking out the common variable "$\vee \neg C$"

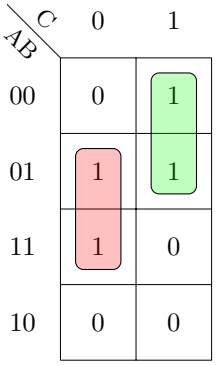
## Karnaugh maps




$\neg A \vee \neg C$




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




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

## NAND gates

When trying to "find NANDs" in Boolean expressions, you're looking to manipluate your expression into the form $\neg(X \wedge Y)$, where $X$ and $Y$ may contain further NAND expressions. The final result may also include extra $\neg$'s, like in the case of AND, as this can also be implemented using a singular NAND gate with copied inputs. A common trick to achieve this is to first add a double negation, which creates a logically equivalent expression that, if needed, can then have a $\neg$ pushed through (using De Morgan's Laws) to remove any $\vee$'s from the expression.

| | |
|---|---|
| $\neg A$ | For NAND implementation see subcircuit *not_nand* |
| $\neg(A \wedge A)$ | $\Rightarrow A \equiv A \wedge A$ |
| | |
| $A \wedge B$ | For NAND implementation see subcircuit *and_nand* |
| $\neg\neg(A \wedge B)$ | $\Rightarrow$ Double negation |
| | |
| $A \vee B$ | For NAND implementation see subcircuit *or_nand* |
| $\neg\neg(A \vee B)$ | $\Rightarrow$ Double negation |
| $\neg(\neg A \wedge \neg B)$ | $\Rightarrow$ De Morgan's Laws |
| | |
| $\neg(A \vee B)$ | For NAND implementation see subcircuit *or_nor* |
| $\neg\neg\neg(A \vee B)$ | $\Rightarrow$ Double negation |
| $\neg\neg(\neg A \wedge \neg B)$ | $\Rightarrow$ De Morgan's Laws |
| | |
| $(\neg A \wedge B) \vee (A \wedge \neg B)$ | For NAND implementation see subcircuit *xor_nand5* |
| $\neg(\neg(\neg A \wedge B) \wedge \neg(A \wedge \neg B))$ | $\Rightarrow$ Double negation and De Morgan's Laws |
| | |
| $(A \vee B) \wedge (\neg A \vee \neg B)$ | For NAND implementation see subcircuit *xor_nand4* |
| $(A \vee B) \wedge \neg(A \wedge B)$ | $\Rightarrow$ De Morgan's Laws |
| $(A \wedge \neg(A \wedge B)) \vee (B \wedge \neg(A \wedge B))$ | $\Rightarrow$ Distributivity, adding in the common variable "$\wedge\ \neg(A \wedge B)$" |
| $\neg(\neg(A \wedge \neg(A \wedge B)) \wedge \neg(B \wedge \neg(A \wedge B)))$ | $\Rightarrow$ Double negation and De Morgan's Laws |

## 5-variable logic

A = 0

| BC \ DE | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 | 0 | 1 | 1 |
| 01 | 1 | 0 | 1 | 1 |
| 11 | 1 | 0 | 1 | 1 |
| 10 | 0 | 0 | 1 | 1 |

A = 1

| BC \ DE | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 | 0 | 0 | 0 |
| 01 | 1 | 0 | 0 | 0 |
| 11 | 1 | 0 | 0 | 0 |
| 10 | 0 | 0 | 1 | 1 |

$$(\neg A \wedge D) \vee (C \wedge \neg D \wedge \neg E) \vee (B \wedge \neg C \wedge D)$$

For an implementation of this expression, see subcircuit *5vars*

| DE \ ABC | 000 | 001 | 011 | 010 | 110 | 111 | 101 | 100 |
|---|---|---|---|---|---|---|---|---|
| 00 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 |
| 01 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 11 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| 10 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |

The key takeaway from the 5-variable Karnaugh Map is the mirror line between 010 and 110 for boxes that cover 1s on both sides of it. In particular, the yellow box doesn't directly connect but can be grouped because of the symmetry and the green box cannot be extended as symmetry must be maintained. The red box does not require this symmetry as it cover 1s only on one side of the mirror line.

## NAND implementation

A solution is valid as long as it produces the correct outputs and only uses 2-input NAND gates. For a NAND implementation only using 14 NAND gates, see subcircuit *5vars_nand*.

$(\neg A \wedge D) \vee (C \wedge \neg D \wedge \neg E) \vee (B \wedge \neg C \wedge D)$
$\Downarrow$ Double negation and De Morgan's Laws

$\neg(\neg(\neg A \wedge D) \wedge \neg(C \wedge \neg D \wedge \neg E) \wedge \neg(B \wedge \neg C \wedge D))$
$\Downarrow$ Add brackets and double negation

$\neg(\neg\neg(\neg(\neg A \wedge D) \wedge \neg(C \wedge \neg D \wedge \neg E)) \wedge \neg(B \wedge \neg C \wedge D))$
$\Downarrow$ Add brackets and double negation

$\neg(\neg\neg(\neg(\neg A \wedge D) \wedge \neg(\neg\neg(C \wedge \neg D) \wedge \neg E)) \wedge \neg(B \wedge \neg C \wedge D))$
$\Downarrow$ Add brackets and double negation

$\neg(\neg\neg(\neg(\neg A \wedge D) \wedge \neg(\neg\neg(C \wedge \neg D) \wedge \neg E)) \wedge \neg(\neg\neg(B \wedge \neg C) \wedge D))$

When attempting a physical implementation on NAND boards, you should be able to identify which NAND on your NAND board represents each NAND in your Logisim design. For the fifth variable, a wire can be connected to a constant pin for 1 or disconnected for 0.