

In this lab you will learn:

1. Given a truth table, how to make a K-Map
2. Given a K-Map how to solve and build a circuit

## Karnaugh Map

The Karnaugh map (KM or K-map) is a method of simplifying Boolean algebra expressions. Maurice Karnaugh introduced it in 1953. The Karnaugh map reduces the need for extensive calculations by taking advantage of humans' pattern-recognition capability. It also permits the rapid identification and elimination of potential race conditions. The required Boolean results are transferred from a truth table onto a two-dimensional grid where, in Karnaugh maps, the cells are ordered in Gray code, and each cell position represents one combination of input conditions, while each cell value represents the corresponding output value. Optimal groups of 1s or 0s are identified, which represent the terms of a canonical form of the logic in the original truth table. These terms can be used to write a minimal Boolean expression representing the required logic.

Karnaugh maps are used to simplify real-world logic requirements so that they can be implemented using a minimum number of physical logic gates. A sum-of-products expression can always be implemented using AND gates feeding into an OR gate, and a product-of-sums expression leads to OR gates feeding an AND gate. Karnaugh maps can also be used to simplify logic expressions in software design. Boolean conditions, as used for example in conditional statements, can get very complicated, which makes the code difficult to read and to maintain. Once minimized, canonical sum-of-products and product-of-sums expressions can be implemented directly using AND and OR logic operators. Diagrammatic and mechanical methods for minimizing simple logic expressions have existed since at least the medieval times. More systematic methods for minimizing complex expressions began to be developed in the early 1950s, but until the mid to late 1980's the Karnaugh map was the most common used in practice.

## Activities

### Activity 1: Making a K-Map from Truth Table

From your lectures, now you already know what Min-terms and Max-terms are. Here, we will use the same techniques to make a K-Map from a given truth table with Min-terms. We will start with a small 3 variable table and make our way through to more complex problems.

Min-terms	A	B	C	Out (F)
M0	0	0	0	0
M1	0	0	1	0
M2	0	1	0	1
M3	0	1	1	0
M4	1	0	0	1
M5	1	0	1	1
M6	1	1	0	1
M7	1	1	1	1

Given the above table, **construct the K-Map for the same and simplify this as minimal SOP**. If you are stuck somewhere, feel free to ask the instructors for directions.

## Activity 2: Validating your K-Map

Once you are done with the previous activity, using only NOR, NOT, and NAND gates and **build the equivalent circuit on bread board**. Give all valid combinations as input and validate the output against the truth table.

Fill in the following table with your observations from the circuit.

A	B	C	F (expect)	F (actual)
0	0	0	0	0
0	0	1	0	0
0	1	0	1	1
0	1	1	0	0
1	0	0	1	1
1	0	1	1	1
1	1	0	1	1
1	1	1	1	1

## Activity 3: Solving a 4 variable K-Map

Using **only NOR and NOT gates**, build a circuit for the following function using the minimal SOP expression:

given a function  $f(a_1, a_2, a_3, a_4)$ :

$a_1a_2 \backslash a_3a_4$	00	01	11	10
00	0	1	1	0
01	1	0	0	1
11	1	0	0	1
10	0	1	1	0

**Finish the truth table below:**

$a_1$	$a_2$	$a_3$	$a_4$	f		$a_1$	$a_2$	$a_3$	$a_4$	f
0	0	0	0	0		1	0	0	0	0
0	0	0	1	1		1	0	0	1	1
0	0	1	0	0		1	0	1	0	0
0	0	1	1	1		1	0	1	1	1
0	1	0	0	1		1	1	0	0	1
0	1	0	1	0		1	1	0	1	0
0	1	1	0	1		1	1	1	0	1
0	1	1	1	0		1	1	1	1	0

### Submission Requirements

Activity 1 - One filled out k-map

Activity 2 - Screenshot of the circuit and filled out truth table

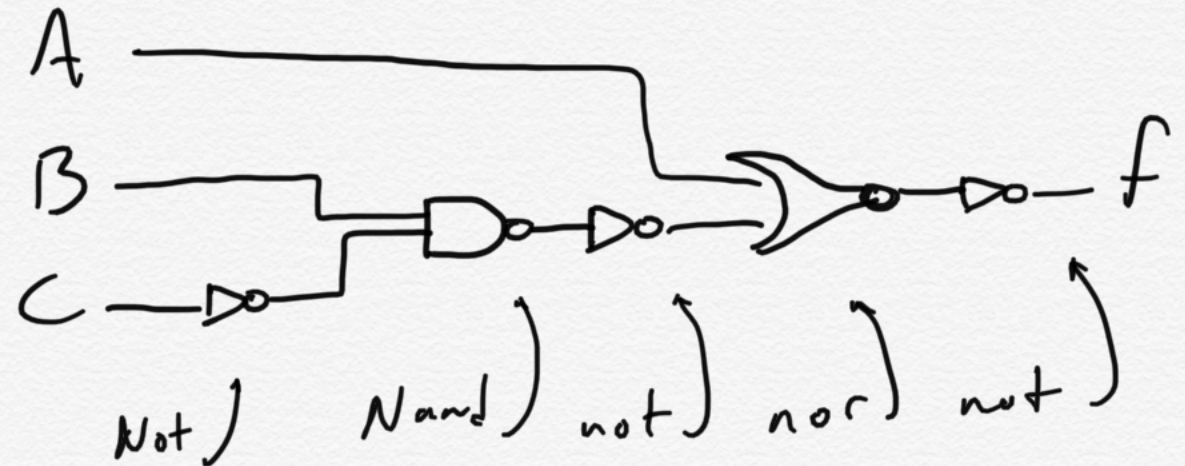
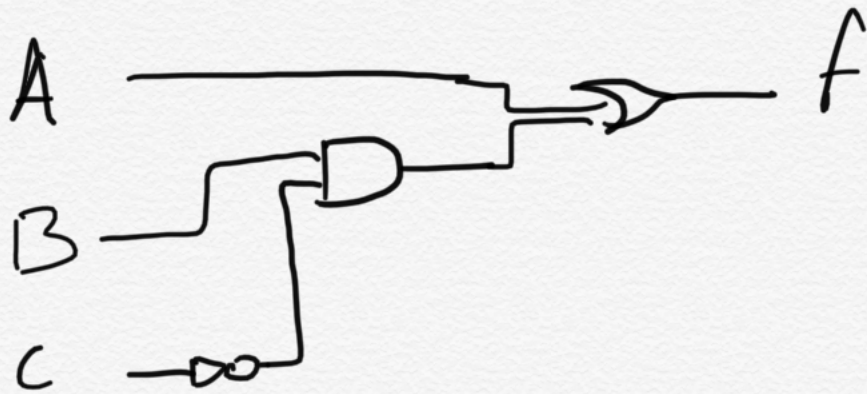
Activity 3 - Screenshot of the circuit and filled out truth table

**Make sure to attach all of these things to 1 file and submit that 1 file with everything on it.**

AB					
C		00	01	11	10
	0		1	1	1
	1			1	1

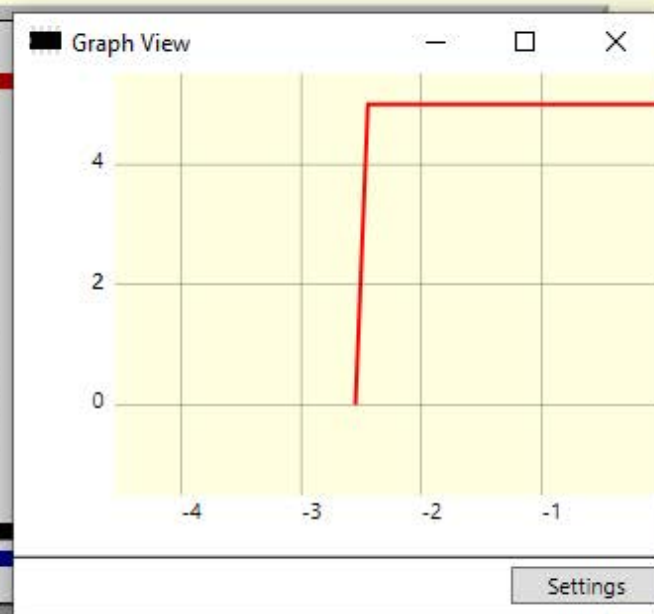
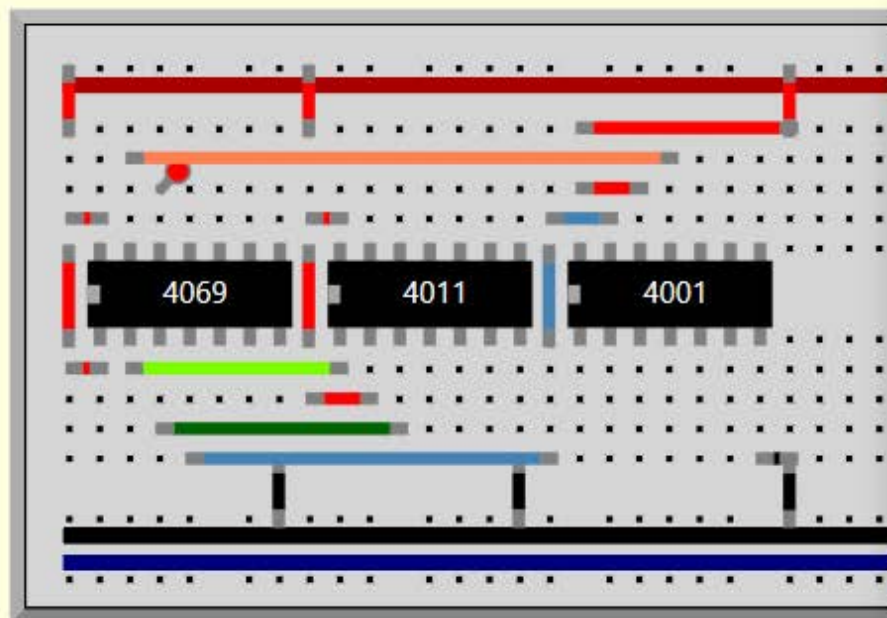
0	2	6	7
1	3	7	5

$$f = B\bar{C} + A \quad \underline{\text{SOP}}$$





- ▷ Resistors
- ▷ Capacitors
- ▷ Electrolytic Capacitors
- ▷ Input Devices
- ▷ Diodes
- ▷ NPN Transistors
- ▷ PNP Transistors
- ▷ N-channel MOSFETs
- ▲ Digital ICs
  - 4001 NOR
  - 4011 NAND
  - 4069 NOT
  - 4070 XOR
  - 4071 OR
  - 4081 AND
  - 4013 D Flip Flop
  - 4017 Decade Counter
  - 4024 Binary Counter
  - 4511 Display Decoder
- ▷ Analog ICs
- ▷ Output Devices
- Oscilloscope Probe

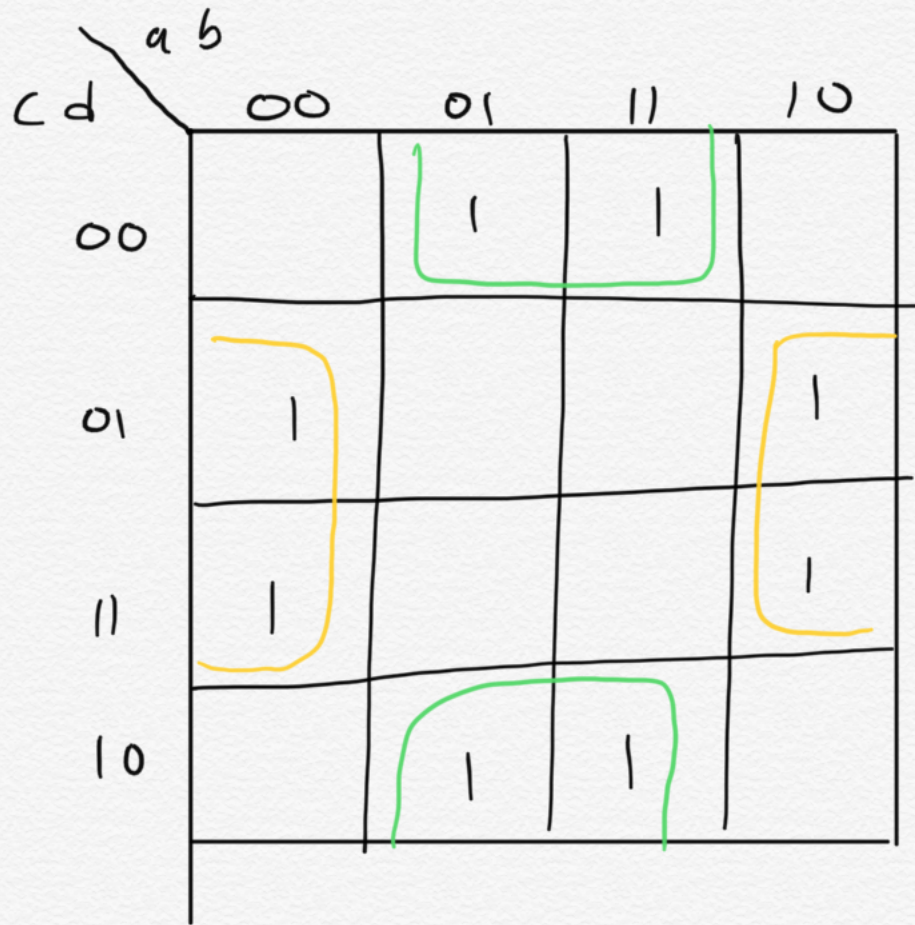




③

$$f(a,b,c,d) = \text{S.o.P}$$

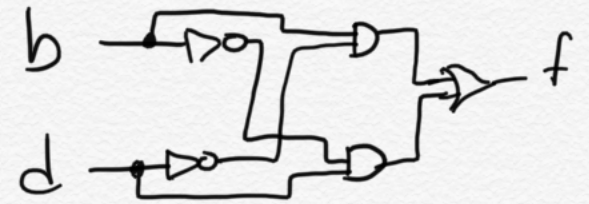
Nor & Not



$$\begin{matrix} 0 & + & 1 & \rightarrow & 1 \\ \overline{1} & 0 & + & 1 & \overline{0} \end{matrix}$$

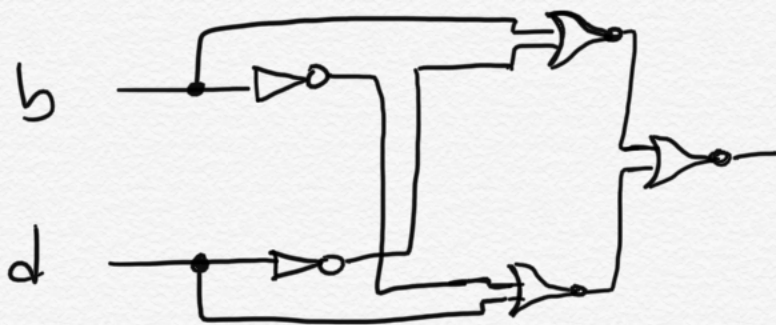
$$f = \underline{\overline{b}d} + \underline{b\overline{d}}$$

0	4	12	8
1	5	13	9
3	7	15	11
2	6	14	10



$$\overline{ab} \rightarrow \overline{a} + \overline{b}$$

\* demorgan's \*



build this  
Nor & Not  
only

$$\begin{aligned} &\overline{b}d + b\overline{d} \\ &\overline{\overline{b}d} + \overline{b\overline{d}} \\ &\overline{b}d + \overline{b}d \\ &(b + \overline{d}) \{ \overline{b} + d \} \\ &f = (\overline{b + d}) + (\overline{\overline{b} + d}) \end{aligned}$$

$$\begin{aligned} &(\overline{b + d}) + (\overline{\overline{b} + d}) \\ &\overline{0 + 1} + \overline{1 + 0} = 0 \\ &\overline{0 + 0} + \overline{1 + 1} = 1 \\ &\overline{1 + 1} + \overline{0 + 0} = 1 \\ &\overline{1 + 0} + \overline{0 + 1} = 0 \end{aligned}$$

$$\begin{aligned} &b\overline{d} \\ &00 \\ &01 \\ &10 \\ &11 \\ &f = b\overline{d} + \overline{b}d \\ &01 + 10 = 0 \\ &00 + 11 = 1 \\ &11 + 00 = 1 \\ &10 + 01 = 0 \end{aligned}$$



- Resistors
- Capacitors
- Electrolytic Capacitors
- Input Devices
- Diodes
- NPN Transistors
- PNP Transistors
- N-channel MOSFETs
- ▲ Digital ICs
  - 4001 NOR
  - 4011 NAND
  - 4069 NOT
  - 4070 XOR
  - 4071 OR
  - 4081 AND
  - 4013 D Flip Flop
  - 4017 Decade Counter
  - 4024 Binary Counter
  - 4511 Display Decoder
- Analog ICs
- Output Devices
- Oscilloscope Probe

