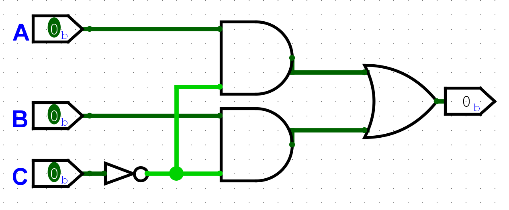
**DLD STEM ICE – Week 1 Tuesday PM**

***Complete these exercises on a separate piece of paper.***

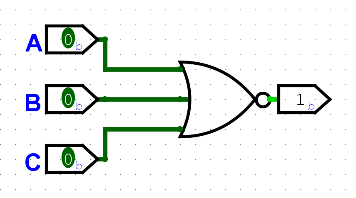
1. Simplify the following expressions using Boolean algebra laws and rules.
   1. **= 0**
      1. B(C+!C+A)
      2. B(1+A)
      3. B(1) = **B**
   2. = **C+AB**
   3. = **D+!C**
   4. 1. A+!A+B+!C+D
      2. 1+B+!C+D
      3. 1+!C+D (etc..)
      4. **= 1**
   5. 1. A+B+!A!BC+!A!B!CD+!A!B!C!DE
      2. A+B+C+!A!B!CD+!A!B!C!DE
      3. A+B+C+D+!A!B!C!DE
      4. = **A+B+C+D+E**
2. For the following logic gate circuit diagrams, first write the Boolean expression for the output in terms of the inputs, then simplify using Boolean algebra, then redraw the simplified circuit diagrams:
   1. 

AB!C + B!C + !BA

AB!C +!BA +B!C

A!C+!BA+B!C

**A!C+B!C**

* 1. A close up of a logo

     Description automatically generated

!(AB+AC)

!A+!B+!A+!C

**!A+!B+!C or !(A+B+C)**

A picture containing clock

Description automatically generated

1. Use DeMorgan’s Rules to simplify the following Boolean expressions:
   1. = !(!A(!B+!C)) -> **A + BC**
   2. = **A!D + !BC + C!D**
2. Using Boolean algebra laws and rules, write the following Boolean expressions in Sum-of-Products form:
   1. = **AB + !B!A + !BC**
   2. (( = A + !AA+BA+!A!B+B!B -> A + !A!B + BA -> A + !B + BA -> A + !B + A -> **A + !B**
   3. = C + (A+B)\*(!A+C) -> C + !AA + !AB + AC + BC -> !AB + C + AC + BC -> !AB + C(1+A+B) -> **!AB + C**
3. 4-variable K-Map

Consider the following Boolean expression:

* 1. What size K-Map will this expression need?
     1. 2^4 = 16 cells
  2. Fill out a truth table for the expression.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A | B | C | D | !A!D | !B!C+!BC | !B!D | !(!A+!B+!C+!D) | X |
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 |

* 1. Use a K-Map to find the simplified Boolean expression.

!A!B!C!D + !A!BC!D

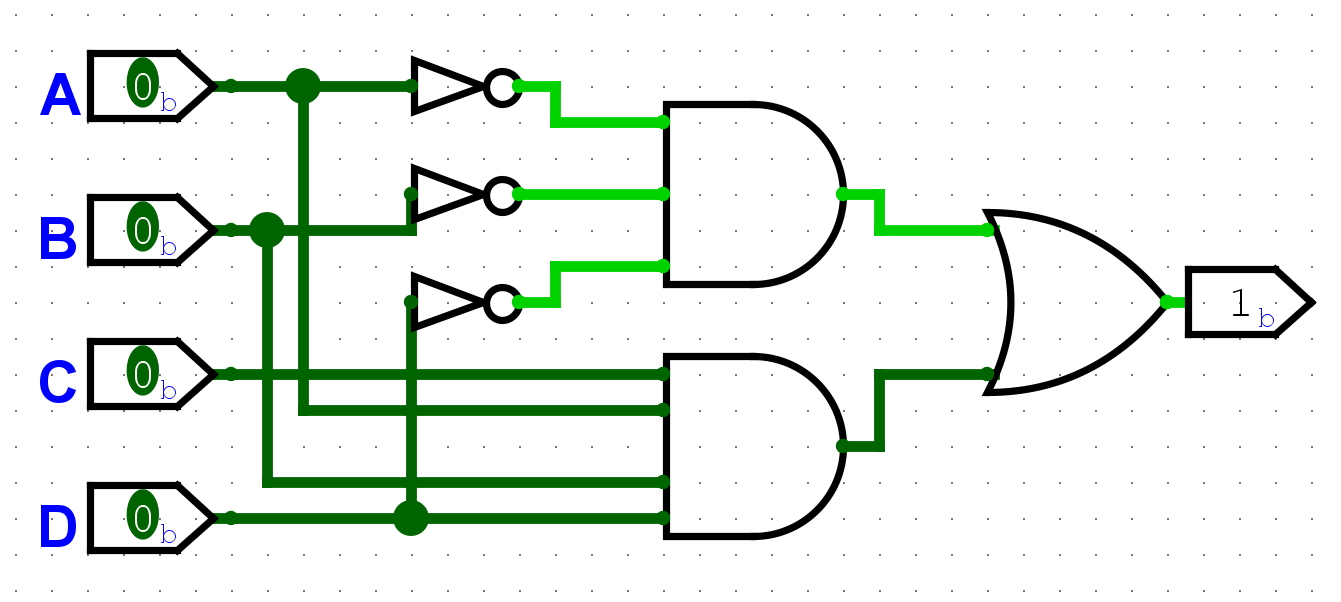
= !A!B!D

ABCD

**!A!B!D + ABCD**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| AB\CD | 00 | 01 | 11 | 10 |
| 00 | 1 | 0 | 0 | 1 |
| 01 | 0 | 0 | 0 | 0 |
| 11 | 0 | 0 | 1 | 0 |
| 10 | 0 | 0 | 0 | 0 |

* 1. Sketch the simplified circuit diagram.



1. Another 4-variable K-Map

Consider the following Boolean expression:

* 1. What size K-Map will this expression need?

2^4 = 16

* 1. Fill out a truth table for the expression.

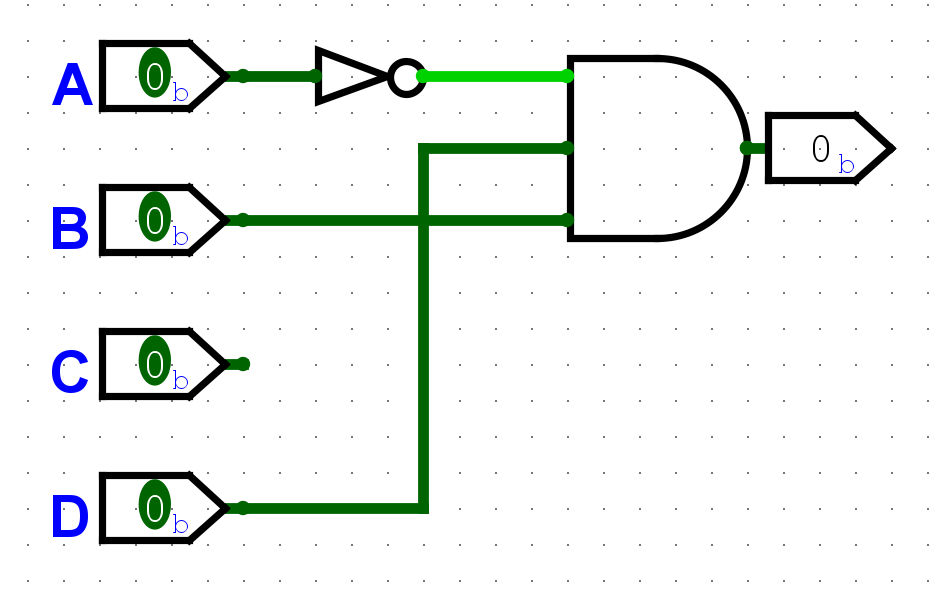
|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A | B | C | D | !A!C | AC | BCD | !AC | A!C | !ABD | X |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |

* 1. Use a K-Map to find the simplified Boolean expression.

!AB!CD + !ABCD = **!ABD**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| AB\CD | 00 | 01 | 11 | 10 |
| 00 | 0 | 0 | 0 | 0 |
| 01 | 0 | 1 | 1 | 0 |
| 11 | 0 | 0 | 0 | 0 |
| 10 | 0 | 0 | 0 | 0 |

* 1. Sketch the simplified circuit diagram.



1. 4-variable K-Map with Don’t Cares -- transition the information in this truth table to a K-Map and determine the reduced expression from the K-Map​

(May be continued on next page!)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Inputs​ | | | | Output​ |
| **A​** | **B​** | **C​** | **D​** | **Z​** |
| 0​ | 0​ | 0​ | 0​ | 0​ |
| 0​ | 0​ | 0​ | 1​ | **1​** |
| 0​ | 0​ | 1​ | 0​ | **1​** |
| 0​ | 0​ | 1​ | 1​ | **1​** |
| 0​ | 1​ | 0​ | 0​ | 0​ |
| 0​ | 1​ | 0​ | 1​ | X​ |
| 0​ | 1​ | 1​ | 0​ | 0​ |
| 0​ | 1​ | 1​ | 1​ | X​ |
| 1​ | 0​ | 0​ | 0​ | 0​ |
| 1​ | 0​ | 0​ | 1​ | 0​ |
| 1​ | 0​ | 1​ | 0​ | 0​ |
| 1​ | 0​ | 1​ | 1​ | **1​** |
| 1​ | 1​ | 0​ | 0​ | 0​ |
| 1​ | 1​ | 0​ | 1​ | **1​** |
| 1​ | 1​ | 1​ | 0​ | X​ |
| 1​ | 1​ | 1​ | 1​ | 0​ |

!A!BCD - !A!BC!D = !A!B!D

!A!BCD – A!BCD = !BCD

AB!CD

**!A!B!D + !BCD + AB!CD**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| AB\CD | 00 | 01 | 11 | 10 |
| 00 | 0 | 0 | 1 | 1 |
| 01 | 0 | X | X | 0 |
| 11 | 0 | 1 | 0 | X |
| 10 | 0 | 0 | 1 | 0 |

1. Design Problem: A car has a fuel-level detector that outputs the current fuel-level as a 4-bit binary number, with 0000 meaning empty and 1111 meaning full.​ In this problem, you will create a circuit that illuminates a “low fuel” indicator light (by setting an output L to 1) when the fuel level drops below ¼ full.​
   1. How many possible input fuel levels are there? (How many rows will there be on the truth table?) How many of those will result in an output of “low fuel”?

16 rows. 4 will result in low fuel

* 1. Fill in the truth table​.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | B | C | D | L |
| 0​ | 0​ | 0​ | 0​ | 1 |
| 0​ | 0​ | 0​ | 1​ | 1 |
| 0​ | 0​ | 1​ | 0​ | 1 |
| 0​ | 0​ | 1​ | 1​ | 1 |
| 0​ | 1​ | 0​ | 0​ | 0 |
| 0​ | 1​ | 0​ | 1​ | 0 |
| 0​ | 1​ | 1​ | 0​ | 0 |
| 0​ | 1​ | 1​ | 1​ | 0 |
| 1​ | 0​ | 0​ | 0​ | 0 |
| 1​ | 0​ | 0​ | 1​ | 0 |
| 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​ | 0 |

* 1. What size K-map will this truth-table convert to?

2^4 = 16

* 1. Convert the truth table to a K-map and obtain the simplified Boolean expression.

!A!B!C!D + !A!B!CD + !A!BCD +!A!BC!D

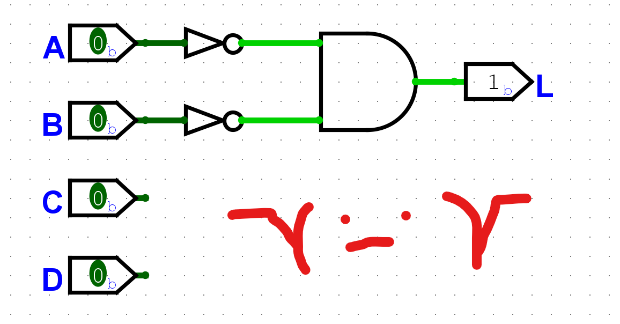
!A!B!C + !A!BCD + !A!BC!D

!A!B!C + !A!BC

**= !A!B**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| AB\CD | 00 | 01 | 11 | 10 |
| 00 | 1 | 1 | 1 | 1 |
| 01 | 0 | 0 | 0 | 0 |
| 11 | 0 | 0 | 0 | 0 |
| 10 | 0 | 0 | 0 | 0 |

* 1. Draw the circuit implementation of your minimized solution.



1. A close up of a sign

   Description automatically generatedMultiple-Output Combination Network - BCD to Seven-Segment Display: Below is a block diagram for a BCD to common-anode seven segment display decoder.

A common-anode display is that where all the anodes of the LEDs are connected to a +5V, and for the segment to produce light we apply a logic “0” to the semgent. To display a decimal “0” we must apply logic “0” to segments a, b, c, d, e, and f and logic “1” to segment g. The truth table below defines the relationships between the inputs and outputs of the decoder.

A close up of a door

Description automatically generated

* 1. Obtain the K-map for each output. (How many K-Maps will you have?)
  2. From the map, determine the minimized function for each output.
  3. From the expressions you obtained, use AND, OR, and NOT gates to construct the circuit network for the display.

1. In yesterday afternoon’s ICE exercise, you constructed a truth table for a car buzzer that sounds in response to feedback from sensors that detect the following binary variables: door open (D = 1 when door is open), key in ignition (K = 1 when key inserted), seat belt fastened (S = 1 à belt fastened), headlights on (H = 1 when lights on).
   1. Now, determine the SOP expression from the truth table you constructed.
   2. Convert your truth table to a K-map and attempt to simplify the Boolean expression.
   3. Draw the circuit of logic gates from the simplified Boolean expression.
   4. How many gates does your design use?