CSCI 220 Computer Architecture I

Lab 4: Karnaugh Maps  
Name:   
Date:2/26/2025

**Introduction**

The purpose of this lab to show simplifying of a Boolean function using Karnaugh maps (K-Maps) and analyzing the impact of don’t care conditions on the minimization process. The goal is to reduce the complexity of the logic expression and implement it in a digital circuit. The report includes the construction of a truth table, K-Map analysis, identification of prime implicants, circuit design, and an evaluation of the don’t care outputs.

**Objective**

The objectives of this lab are:

1. To construct a truth table for the given Boolean function, including don’t care conditions.
2. To use a K-Map to derive a minimized sum-of-products (SOP) expression.
3. To identify essential prime implicants and analyze their role in simplification.
4. To implement the simplified expression using logic gates and integrated circuits (ICs).
5. To evaluate the outputs for don’t care cases and explain their behavior in the circuit.

**Procedure**

The experiment began by analyzing the Boolean function:

*f* (*A*,*B*,*C*,*D*)=*A*′*B*′*C*′*D*′+*A*′*B*′*CD*+*A*′*B*′*CD*+*A*′*BCD*+*ABCD*+*AB*′*CD*+*AB*′*CD*′ with don’t care terms: *AB*′*C*′*D*′, *ABC*′*D*, and *A*′*B*′*CD*′.

**1. Breadboard with +5 Vdc Power Supply**

The breadboard served as the platform for connecting the integrated circuits (ICs) and other components. It provided a convenient way to prototype the circuit without soldering.

1. The power supply was connected to the breadboard to provide +5 Vdc to the ICs.
2. The positive terminal (+5 Vdc) was connected to the Vcc pins of the ICs, and the ground (GND) terminal was connected to the GND pins of the ICs.
3. Power rails on the breadboard were used to distribute power to all components.

**2. Integrated Circuits (ICs)**

The following ICs were used to implement the logic gates required for the simplified Boolean function:

**7404 (Hex Inverter):**

 This IC contains six NOT gates (inverters). It was used to generate the complemented inputs *A*′, *B*′, and *D*′.

**Usage:**

* + - *Pin 1 (Input A) was connected to the input signal AA, and Pin 2 (Output Y1) provided A′.*
    - *Pin 3 (Input B) was connected to the input signal BB, and Pin 4 (Output Y2) provided B′.*
    - *Pin 5 (Input C) was connected to the input signal DD, and Pin 6 (Output Y3) provided D′.*
    - *Pins 7 (*GND) and 14 (Vcc) were connected to the power supply.

**7408 (Quad 2-Input AND Gate):**

This IC contains four AND gates. It was used to implement the AND operations in the simplified expression *CD*, *A*′*B*′*D*′, and *AB*′*C*.

**Usage:**

***For CD:***

* + - *Pin 1 (Input A1) was connected to C, and Pin 2 (Input B1) was connected to D.*
    - *Pin 3 (*Output Y1) provided the result of *CD*.

***For A′B′D′:***

* + - *Pin 4 (Input A2) was connected to A′, Pin 5 (Input B2) was connected to B′, and Pin 6 (Input C2) was connected to D′.*
    - *Pin 7* (Output Y2) provided the result of *A*′*B*′*D*′.

***For AB′C:***

* + - *Pin 9 (Input A3) was connected to AA, Pin 10 (Input B3) was connected to B′, and Pin 11 (Input C3) was connected to C.*
    - *Pin 8 (*Output Y3) provided the result of *AB*′*C*.

Pins 7 (GND) and 14 (Vcc) were connected to the power supply.

**7432 (Quad 2-Input OR Gate):**

This IC contains four OR gates. It was used to combine the outputs of the AND gates (*CD*, *A*′*B*′*D*′, and *AB*′*C*) into the final output *f*.

**Usage:**

* + - *Pin 1 (Input A1) was connected to the output of CD, and Pin 2 (Input B1) was connected to the output of A′B′D′.*
    - *Pin 3 (Output Y1) provided the intermediate result of CD+A′B′D′.*
    - *Pin 4 (Input A2) was connected to the intermediate result, and Pin 5 (Input B2) was connected to the output of AB′C.*
    - *Pin 6 (Output Y2) provided the final output f=CD+A′B′D′+AB′C.*
    - *Pins 7 (GND) and* 14 (Vcc) were connected to the power supply.

**3. Jumper Wires**

Jumper wires were used to make connections between the ICs, input switches, and output LEDs.

**Usage:**

Wires were used to connect the input signals (A, B, C, D) to the appropriate pins on the ICs.

Outputs from the ICs were connected to LEDs to visually verify the circuit's functionality.

Power and ground connections were made using jumper wires to ensure all components were properly powered.

**4. LEDs for Output Verification**

LEDs were used to display the output of the circuit for each input combination.

The final output *f* from the 7432 IC was connected to an LED through a current-limiting resistor (220 Ω). When the output *f* was high (1), the LED lit up, indicating a true output. When the output was low (0), the LED remained off.

**Circuit Implementation**

**Equation 1: Simplified Boolean Function**

*f*=*CD*+*A*′*B*′*D*′+*AB*′*C*

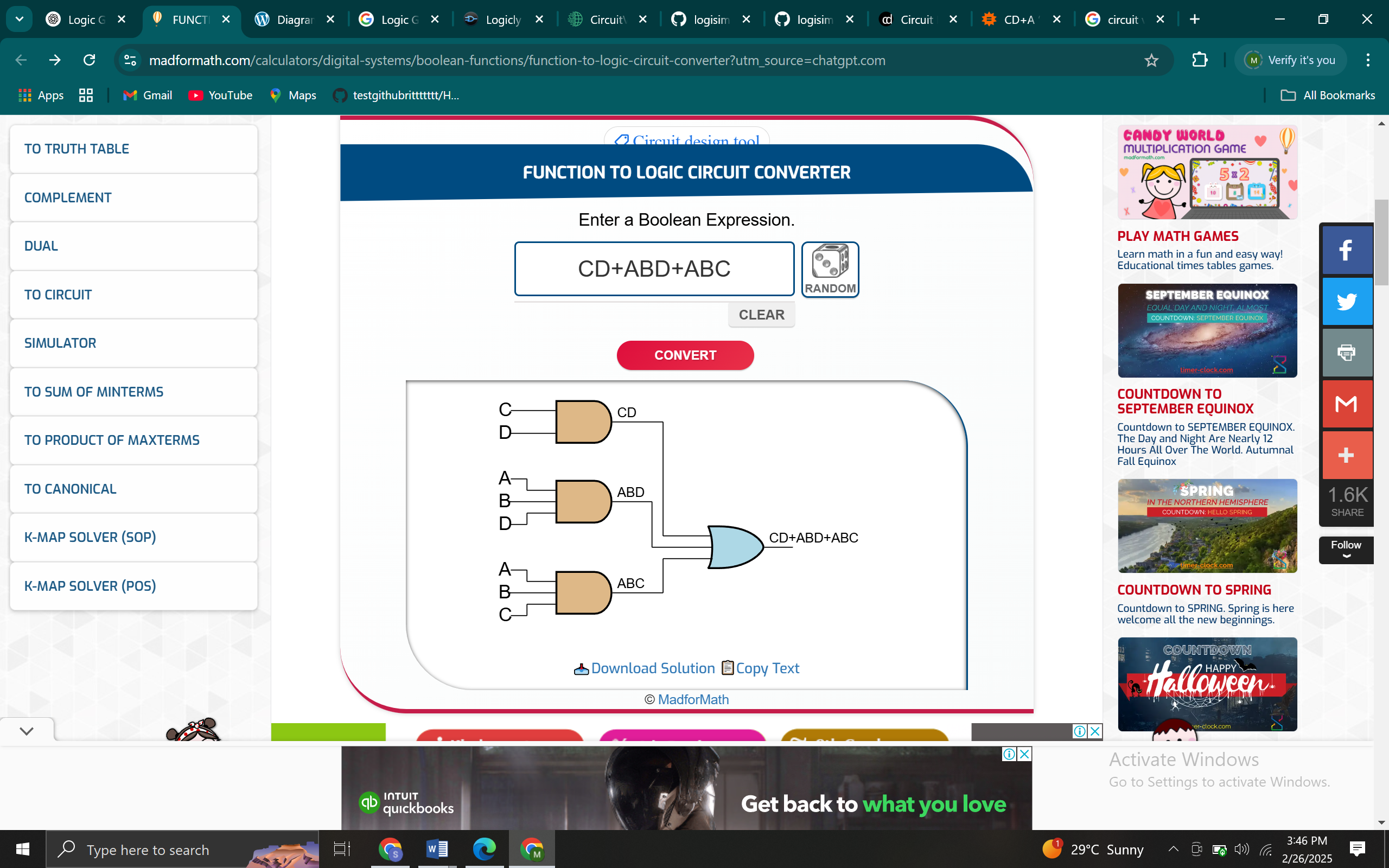


Figure 1: f=CD+A′B′D′+AB′C

**Circuit Breakdown**

**Inverters (7404):**

* + - *A′: Generated by connecting A to Pin 1 (Input A1) of the 7404 IC. The output A′ was obtained from Pin 2 (Output Y1).*
    - *B′: Generated by connecting BB to Pin 3 (Input A2) of the 7404 IC. The output B′ was obtained from Pin 4 (Output Y2).*
    - *D′: Generated by connecting D to Pin 5 (Input A3) of the 7404 IC. The output D′ was obtained from Pin 6 (Output Y3).*

**AND Gates (7408):**

* + - *CD: Implemented by connecting C to Pin 1 (Input A1) and D to Pin 2 (Input B1) of the 7408 IC. The output CD was obtained from Pin 3 (Output Y1).*
    - *A′B′D′: Implemented by connecting A′ to Pin 4 (Input A2), B′ to Pin 5 (Input B2), and D′ to Pin 6 (Input C2) of the 7408 IC. The output A′B′D′ was obtained from Pin 7 (Output Y2).*
    - *AB′C: Implemented by connecting A to Pin 9 (Input A3), B′ to Pin 10 (Input B3), and C to Pin 11 (Input C3) of the 7408 IC. The output AB′C was obtained from Pin 8 (Output Y3).*

**OR Gate (7432):**

* + - *The outputs CD, A′B′D′, and AB′C were combined using the 7432 IC.*
    - *CD and A′B′D′ were connected to Pin 1 (Input A1) and Pin 2 (Input B1) of the 7432 IC, respectively. The intermediate result CD+A′B′D′ was obtained from Pin 3 (Output Y1).*
    - *The intermediate result and AB′C were connected to Pin 4 (Input A2) and Pin 5 (Input B2) of the 7432 IC, respectively. The final output f=CD+A′B′D′+AB′C was obtained from Pin 6 (Output Y2).*

**Output Verification:**

The final output *f* was connected to an LED through a 220 Ω resistor. The LED lit up for high outputs (1) and remained off for low outputs (0).

**Steps:**

1. Truth Table Construction: All 16 input combinations (A, B, C, D) were evaluated. Outputs for don’t care terms were marked as "X."
2. K-Map Construction: A 4-variable K-Map was populated using the truth table data.
3. Prime Implicant Identification: Groups of adjacent 1s and Xs were circled to identify prime implicants.
4. Simplification: The essential prime implicants were selected to form the minimal SOP expression.
5. Circuit Implementation: The simplified expression was translated into a circuit using ICs on a breadboard.
6. Don’t Care Analysis: Outputs for don’t care cases were tested and analysed.

**Data and Discussion**

**1. Truth Table**

The truth table for the Boolean function is constructed by evaluating all possible input combinations of A, B, C, and D. The don’t care conditions are marked as "X."

**Table 1: Truth Table**

Table 1: f (A, B, C, D) =CD+A

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **D** | **f** |
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | X |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | X |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | X |
| 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 |

**2. Karnaugh Map**

The K-Map is constructed based on the truth table. The rows represent AB, and the columns represent CD. The cells are filled with the corresponding output values from the truth table.

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

Figure 2: Karnaugh Map

**3. Prime Implicants**

The prime implicants are identified by grouping adjacent 1s and Xs in the K-Map. The essential prime implicants are those that cover at least one unique minterm.

**Prime Implicants Identified:**

* + - 1. *CD* (covers minterms 3, 7, 11, 15).
      2. *A*′*B*′*D*′ (covers minterm 0 and don’t care 2).
      3. *AB*′*C* (covers minterms 10, 11).

All three are essential, as each covers at least one unique minterm.

**Simplified Expression**

The minimal SOP expression derived from the K-Map is:

*f*=*CD*+*A*′*B*′*D*′+*AB*′*C*

***(Equation 1)***

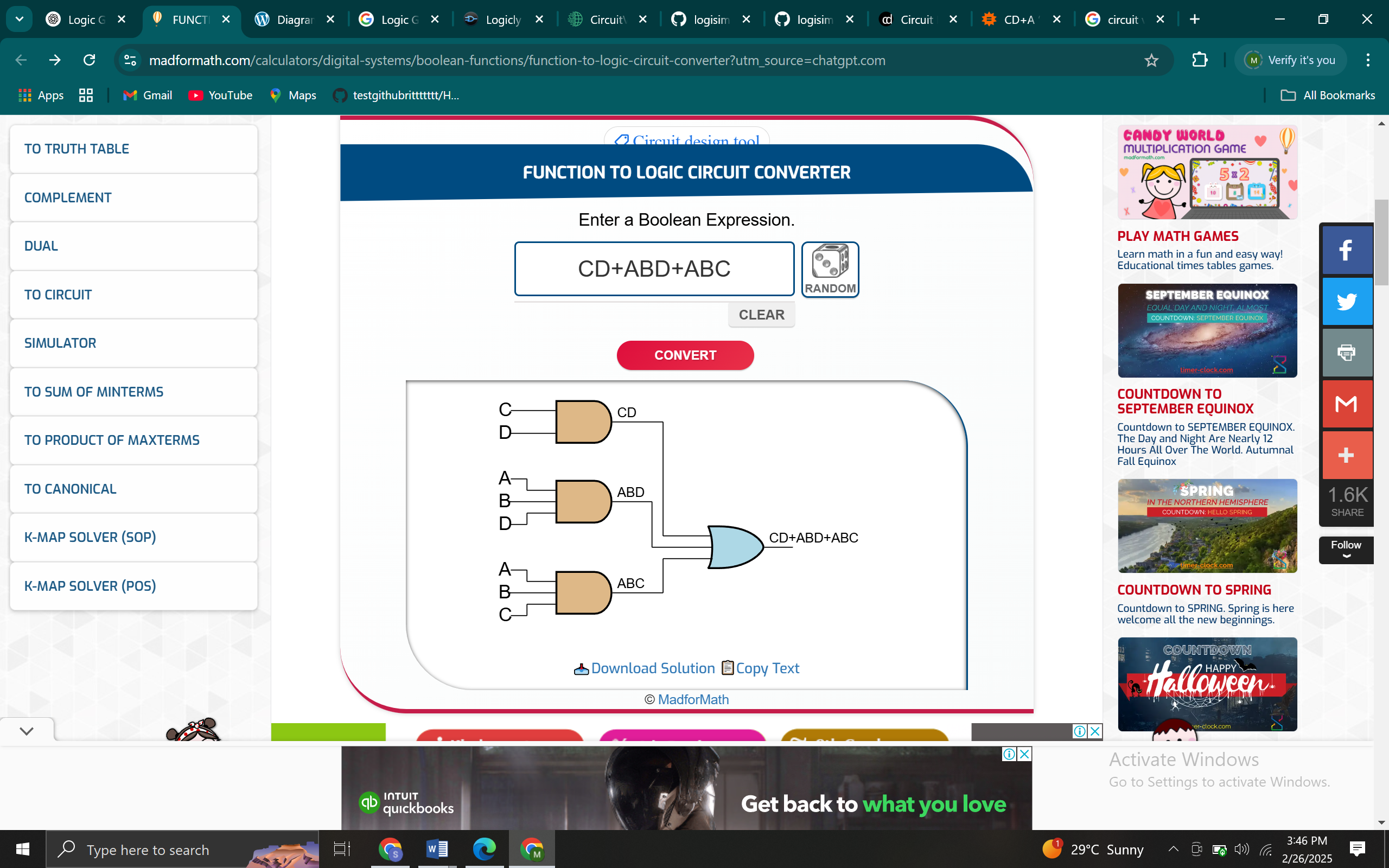


Figure 1: f=CD+A′B′D′+AB′C

**Circuit Implementation**

The circuit is implemented using the following ICs:

* + - *7404 (Inverters): For A′, B′, and D′.*
    - *7408 (AND Gates): For CD, A′B′D′, and AB′C.*
    - *7432 (OR Gate***):** *To combine the AND gate outputs.*

**Figure 2: Circuit Diagram**   
*(Note: Insert an image or diagram of the circuit here, showing the connections for CD, A′B′D′, and AB′C combined with an OR gate.)*

**6. Don’t Care Outputs**

*Minterm 2 (0010): Output = 1 (covered by A′B′D′).*

*Minterm 9 (1001): Output = 0 (not covered by any prime implicant).*

*Minterm 13 (1101): Output = 0 (not covered).*

**Discussion:**  
The don’t care at minterm 2 was included to simplify *A*′*B*′*D*′, while minterms 9 and 13 were excluded. This trade-off minimized the number of gates without affecting critical outputs.

**Conclusions**

1. Karnaugh maps effectively reduced the Boolean function to *f*=*CD*+*A*′*B*′*D*′+*AB*′*C*, eliminating redundant terms.
2. Don’t care conditions allowed flexibility in grouping, reducing circuit complexity.
3. The implemented circuit functioned as expected, validating the K-Map analysis.

**Appendix***.*

**Karnaugh Map**

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

Figure 2: Karnaugh Map

**Circuit Diagram**

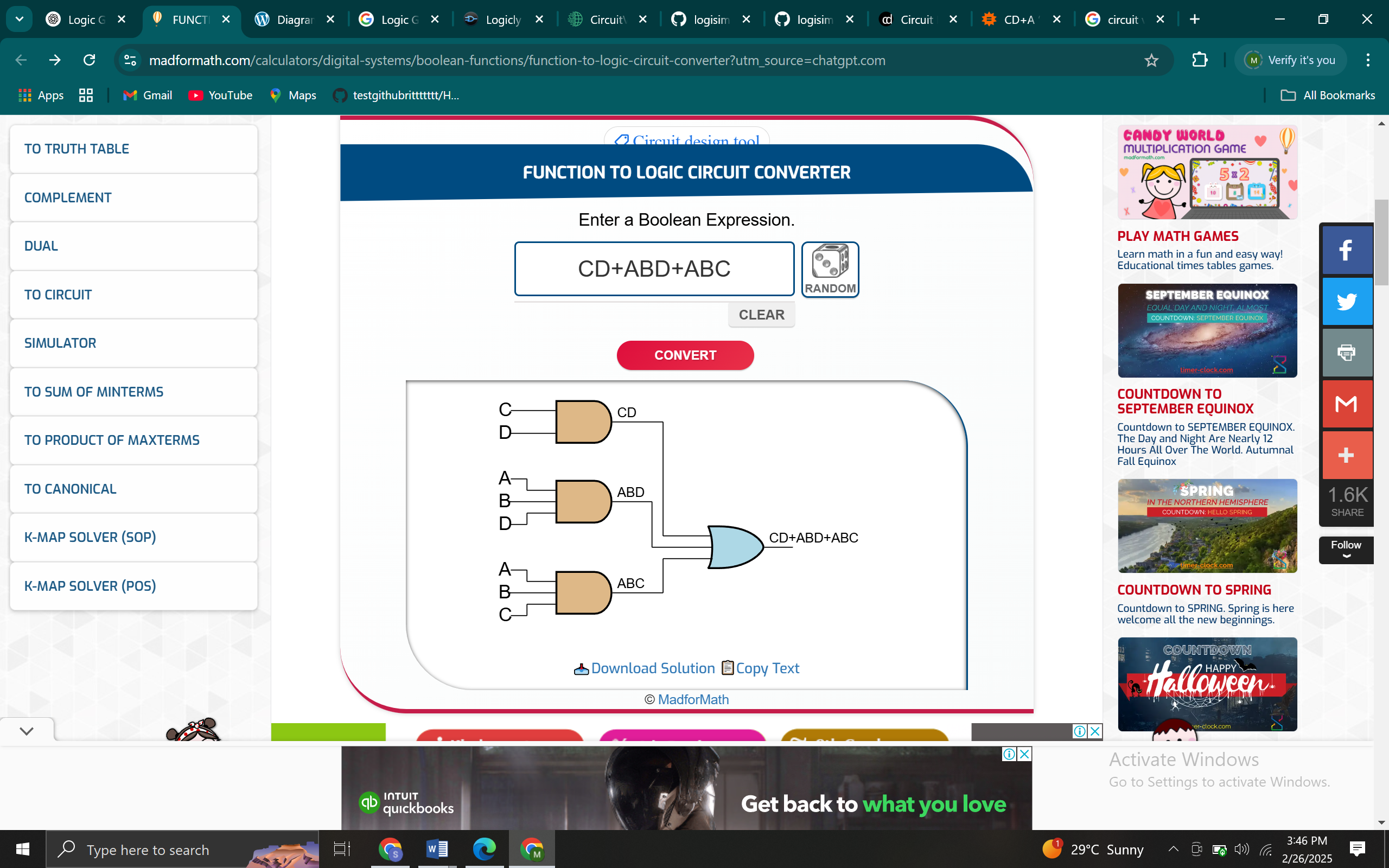


Figure 1: f=CD+A′B′D′+AB′C

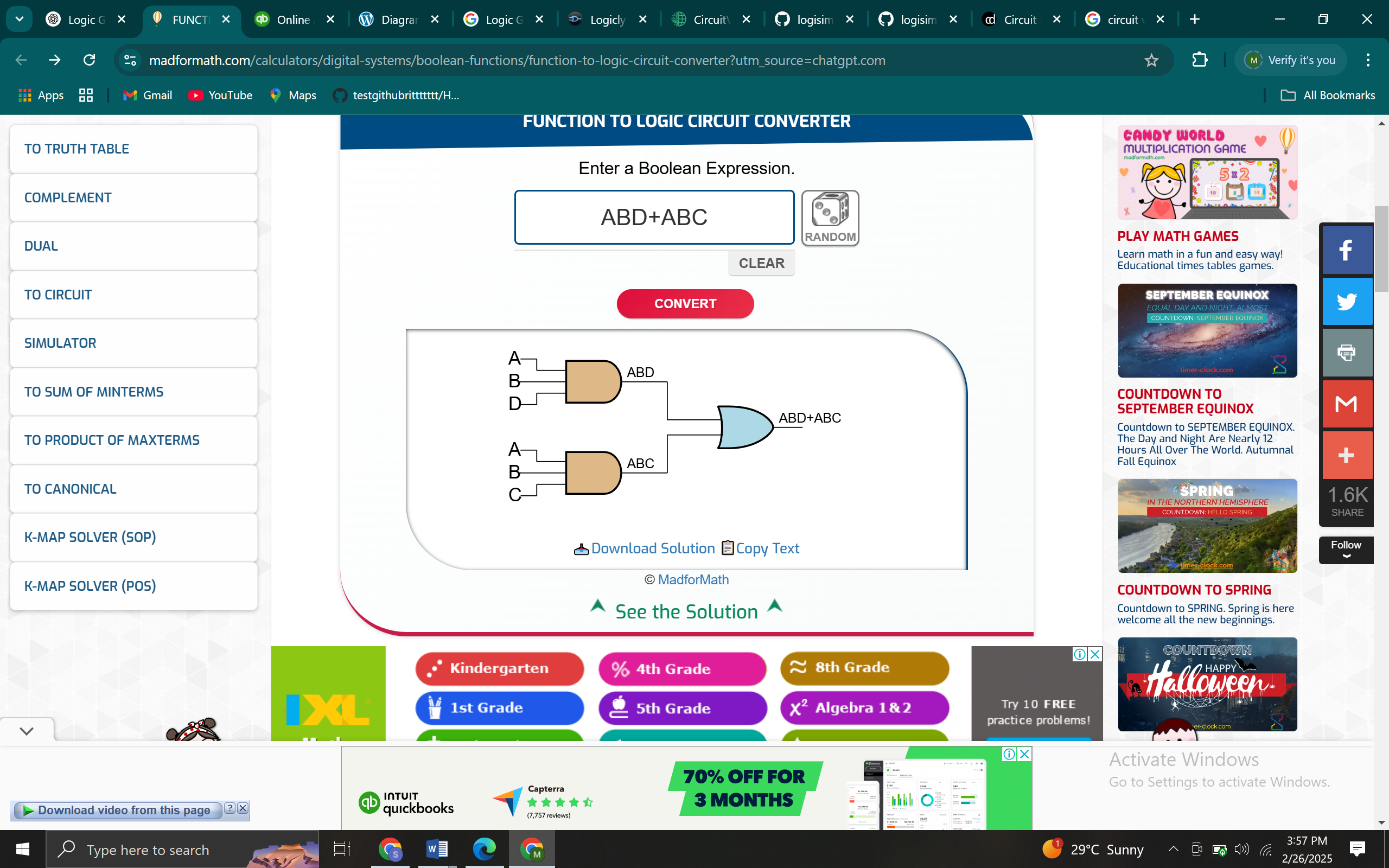


Figure 3: circuit diagram for f =CD+A′B′D′+AB′C

**Equation 1: Simplified Boolean Function**

f =*CD*+*A*′*B*′*D*′+*AB*′*C*

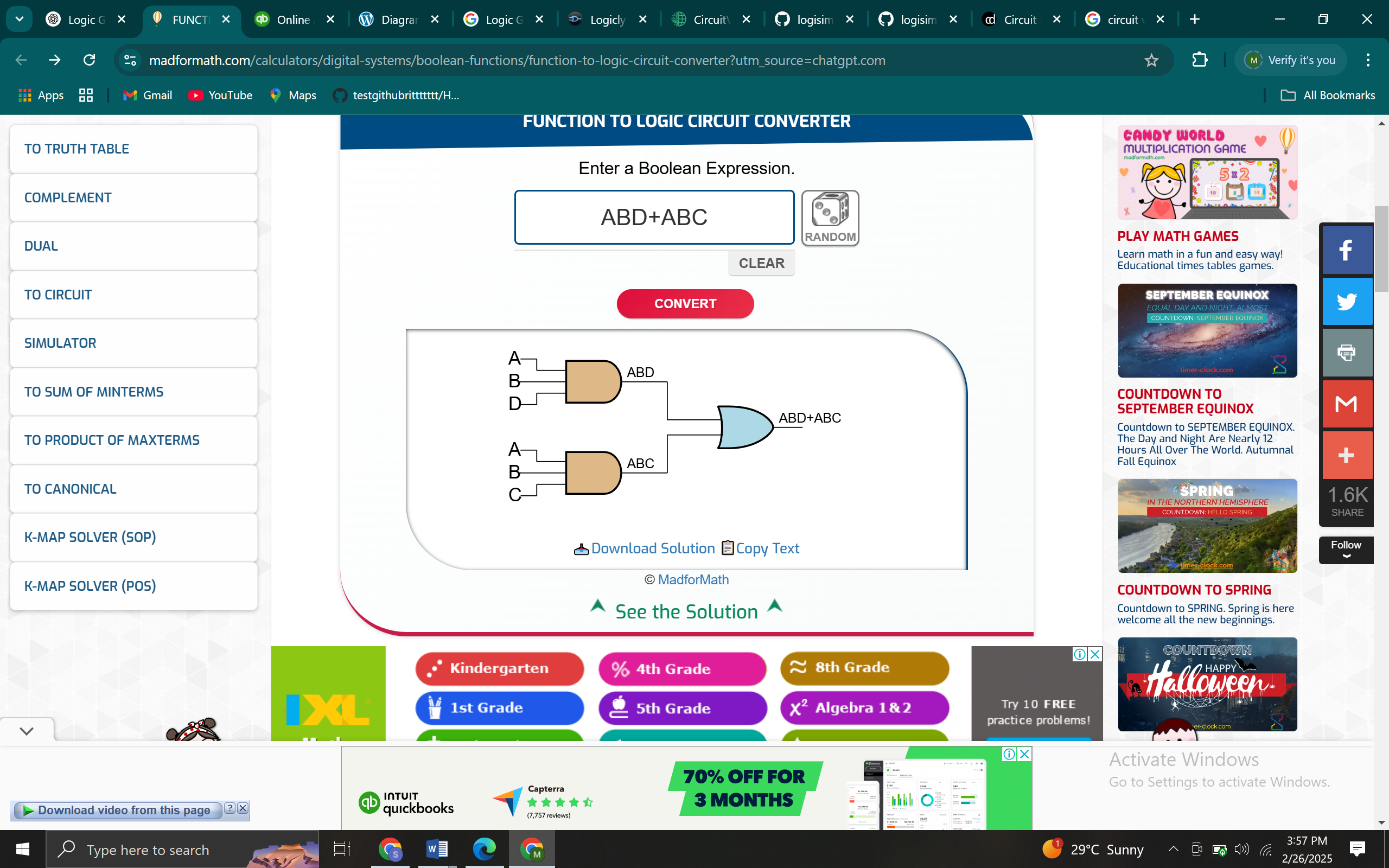


Figure 3: circuit diagram for f =CD+A′B′D′+AB′C

**Truth Table**  
Table 1: f (A, B, C, D) =CD+A

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **D** | **f** |
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | X |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | X |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | X |
| 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 |