A white background with black text

Description automatically generated

1:

2:

3:

4:

5:

6: Solution =

A math equation on a white background

Description automatically generated

1:

2:

3:

4:

5:

6:

7:

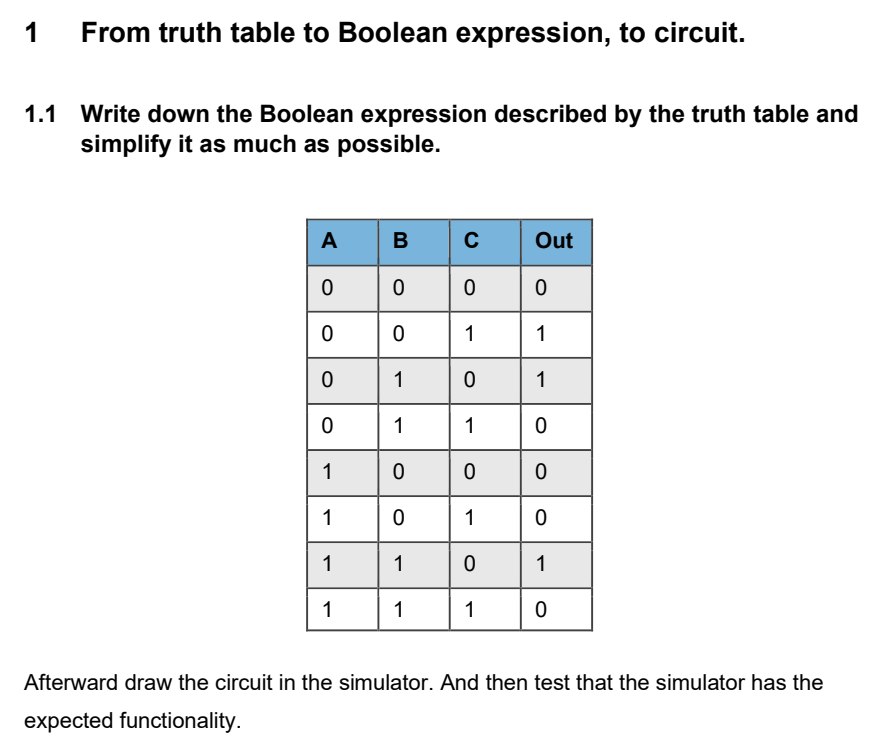
A screenshot of a computer

Description automatically generated

I can be reduced to:

A diagram of a circuit

Description automatically generated



Extracted expression:

Simplification process:

1:

2:

3:

4:

**Logisim solution:**

A diagram of a block diagram

Description automatically generated

A diagram of a diagram

Description automatically generated

|  |  |  |  |
| --- | --- | --- | --- |
| **A** | **B** | **CARRY** | **SUM** |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 |

A white background with black text

Description automatically generated

A diagram of a circuit

Description automatically generated

**Et billede, der indeholder tekst, Font/skrifttype, hvid, kvittering

Automatisk genereret beskrivelse**

**Simulation with chips:**

A diagram of a toilet

Description automatically generated

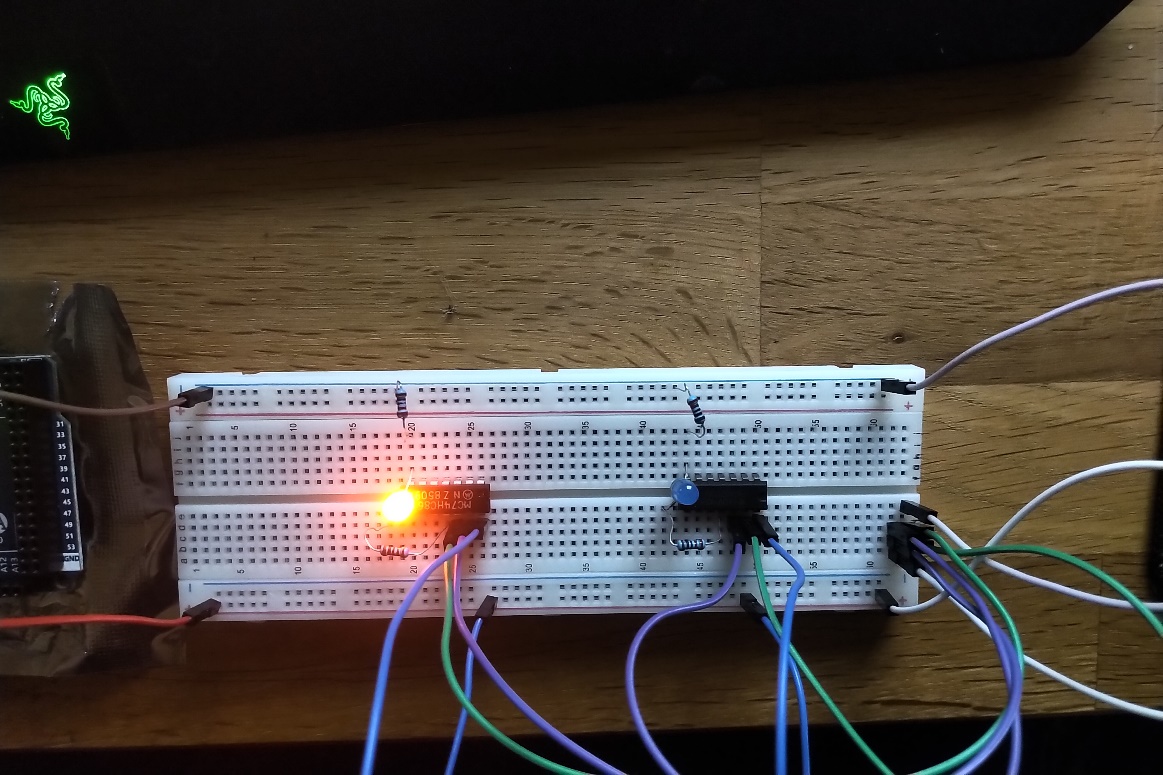
**Implementation on Breadboard:**

**1st Setup: Result, no lights – as expected**Input:  
- A is Negative (Green wires)  
- B is Negative (Purple wires)

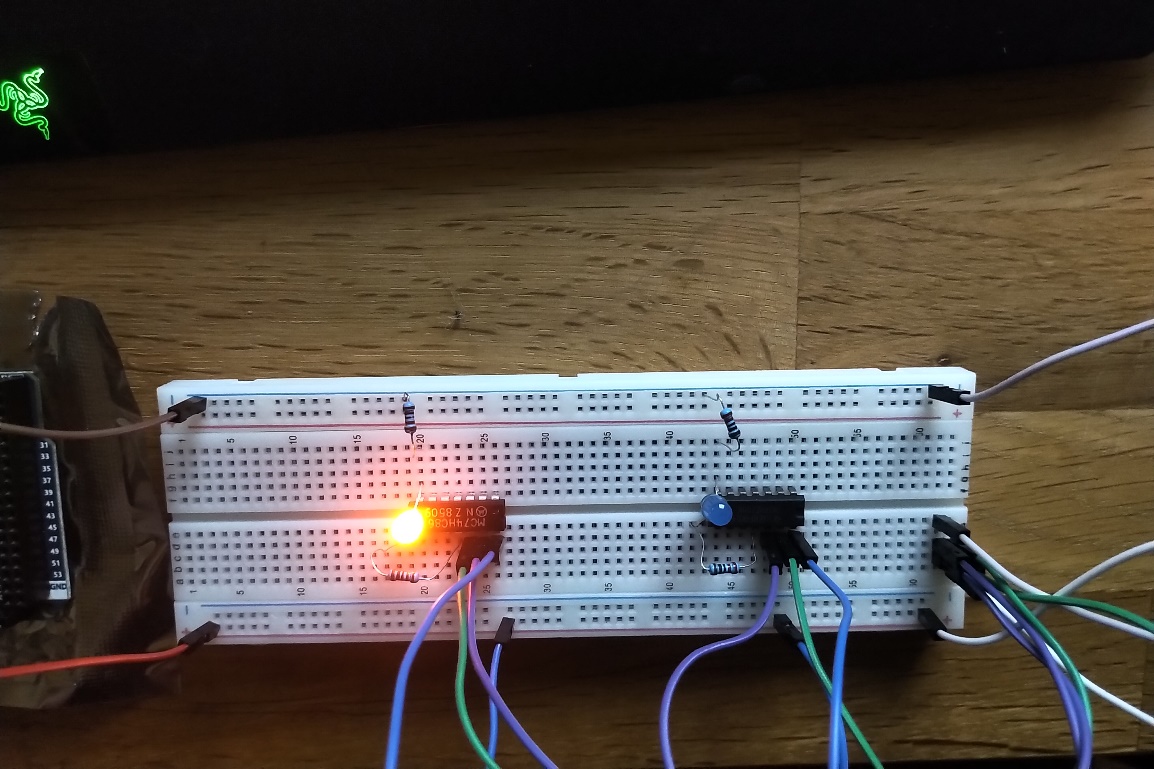
**Et billede, der indeholder elektronik, Elektroteknik, Elarbejde, kable

Automatisk genereret beskrivelse**

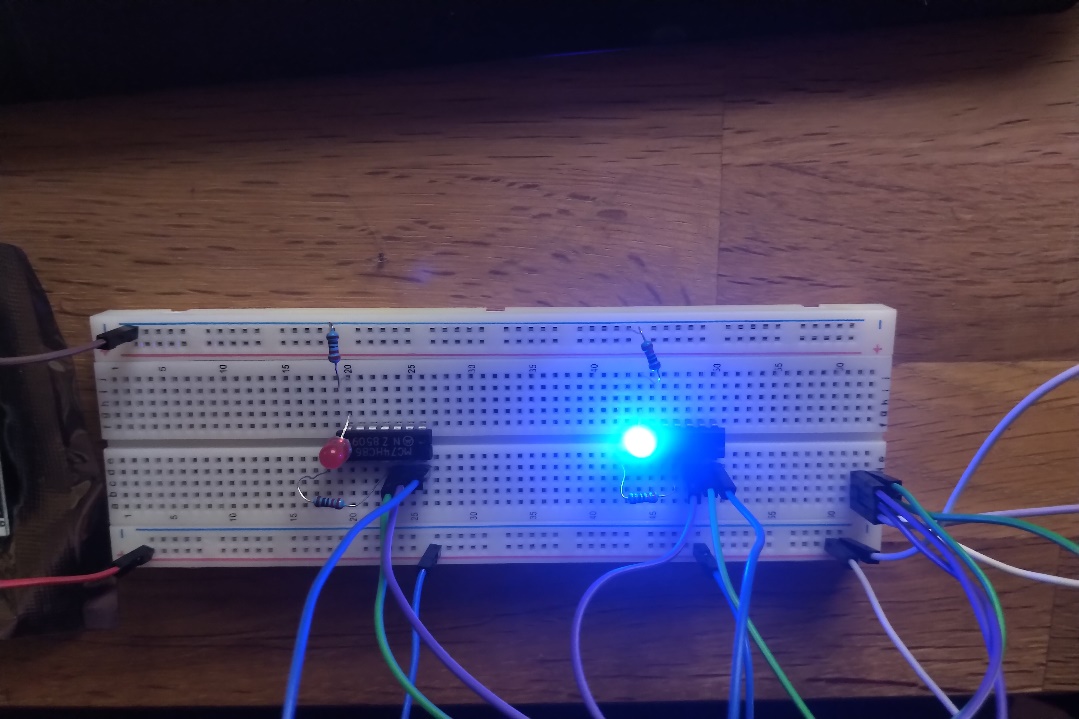
**2nd Setup: Result, Sum light shines – as expected**Input:  
- A is Positive (Green wires)  
- B is Negative (Purple wires)

****

**3rd Setup: Result, Sum light shines – as expected**Input:  
- A is Negative (Green wires)  
- B is Positive (Purple wires)

****

**4th Setup: Result, CarryOver light shines – as expected**Input:  
- A is Positive (Green wires)  
- B is Positive (Purple wires)

****

1. **Full-adder gate**
   1. **Theory**

Draw the truth-table describing the logic from a full adder.**Et billede, der indeholder diagram, skitse, tegning, Stregtegning

Automatisk genereret beskrivelse**

(5)

(4)

(3)

(2)

(1)

**Truth-table:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **A** | **B** | **Cin** | (1) | **AB** (2) | (3) | (4) | (5) |
| A exclusive OR B | A and B | **Sum** | - | **Cout** |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |

**Exclusive or:** False when both inputs are the same, True when both inputs are different.

* 1. **Simulation**

Do it in Logisim, using both ideal and ’real’ components.

**Ideal components:**

Et billede, der indeholder diagram, linje/række, Plan, Kurve

Automatisk genereret beskrivelse

**Real components:**

Et billede, der indeholder diagram, linje/række, Plan, Teknisk tegning

Automatisk genereret beskrivelse

* 1. **Realization**

To realize the full adder an OR gate is also needwed. This can be done with SN74HCT32N.

* Create a circuit on the breadboard.
* Test that it works as described in the truth table in section 3.1

**NOTE: UNABLE TO COMPLETE DUE TO LACK OF CABLES!**

1. **2’s Complement Exercise**

Et billede, der indeholder tekst, nummer/tal, skærmbillede, Parallel

Automatisk genereret beskrivelse

* All positive numbers begin with a 0
* All negative numbers begin with a 1
* To convert x to –x: flip all bits of x, add 1 and throw away the carry (if any)
* To calculate Y – X (both X and Y >= 0):
  + Calculate 2’s complement of X
  + Add that to Y
  1. **8-bit processor**

The following binary pattern is stored in the RAM: 0b11100110

**In case this should be interpreted as an unsigned integer. What is the value?**

Since unsigned integers can only be positive, we can simply convert the binary pattern to its corresponding decimal value:

Convert to decimal:

**In case this should be interpreted as a signed integer. What is the value?**

1. Find 2’s Complement by inverting all the bits:  
   0b11100110 -> 0b00011001 (*it is now a positive number*)
2. Add 1 to the inverted bits, and throw away any carry:

00011001 + 00000001 -> 00011010

Convert to decimal:

1. Invert the decimal value, because the initial binary pattern was negative.

Solution = -26

* 1. **The following numbers are saved as a 8-bit signed integer type. What is the saved binary pattern?**Note: Remember, negative signed integers begin with 1. Positives begin with 0.

1. **-4**
2. Binary Conversion Algorithm is utilized on the non-negative value 4:

|  |  |  |
| --- | --- | --- |
|  |  |  |
| 4 | 2 | 0 |
| 1 | 1 | 0 |
| 0 | 0 | 1 |
| Binary value is: 100 -> 0b00000100 | | |

1. Since initial value was negative, we invert all the bits:  
   0b00000100 -> 0b11111011
2. Then we add 1 to the binary:

0b11111011 + 0b00000001 = 0b11111100

1. -4 is represented by the signed binary 0b11111100
2. **54**
3. Binary Conversion Algorithm is utilized on the non-negative value 54:

|  |  |  |
| --- | --- | --- |
|  |  |  |
| 54 | 27 | 0 |
| 27 | 13 | 1 |
| 13 | 6 | 1 |
| 6 | 3 | 0 |
| 3 | 1 | 1 |
| 1 | 0 | 1 |
| Binary value is: 110110 -> 0b00110110 | | |

No need for further actions, since the number is positive.

1. 54 is represented by the signed binary 0b00110110
2. **-54**
3. Binary Conversion Algorithm is utilized on the non-negative value 54:

|  |  |  |
| --- | --- | --- |
|  |  |  |
| 54 | 27 | 0 |
| 27 | 13 | 1 |
| 13 | 6 | 1 |
| 6 | 3 | 0 |
| 3 | 1 | 1 |
| 1 | 0 | 1 |
| Binary value is: 110110 -> 0b00110110 | | |

1. Since initial value was negative, we invert all the bits:  
   0b00110110 -> 0b11001001
2. Then we add 1 to the binary:

0b11001001 + 0b00000001 = 0b11001010

1. -54 is represented by the signed binary 0b11001010
2. **-128**  
   Note: 8-bit binary can express values from -128 to 127!
3. Binary Conversion Algorithm is utilized on the non-negative value 128:

|  |  |  |
| --- | --- | --- |
|  |  |  |
| 128 | 64 | 0 |
| 64 | 32 | 0 |
| 32 | 16 | 0 |
| 16 | 8 | 0 |
| 8 | 4 | 0 |
| 4 | 2 | 0 |
| 2 | 1 | 0 |
| 1 | 0 | 1 |
| Binary value is: 10000000 -> 0b10000000 | | |

1. Since initial value was negative, we invert all the bits:  
   0b10000000 -> 0b01111111
2. Then we add 1 to the binary:

0b01111111+ 0b00000001 = 0b10000000

1. -54 is represented by the signed binary 0b10000000
2. **127**
3. Binary Conversion Algorithm is utilized on the non-negative value 127:

|  |  |  |
| --- | --- | --- |
|  |  |  |
| 127 | 63 | 1 |
| 63 | 31 | 1 |
| 31 | 15 | 1 |
| 15 | 7 | 1 |
| 7 | 3 | 1 |
| 3 | 1 | 1 |
| 1 | 0 | 1 |
| Binary value is: 1111111 -> 0b01111111 | | |

No need for further actions, since the number is positive.

1. 54 is represented by the signed binary 0b01111111
   1. **Adding**

Add the following numbers together after converting them to binary:

1. **4+7**

Binary Conversion Algorithm is utilized on the non-negative value 4:

|  |  |  |
| --- | --- | --- |
|  |  |  |
| 4 | 2 | 0 |
| 2 | 1 | 0 |
| 1 | 0 | 1 |
| Binary value is: 100 -> 0b00000100 | | |

Binary Conversion Algorithm is utilized on the non-negative value 7:

|  |  |  |
| --- | --- | --- |
|  |  |  |
| 7 | 3 | 1 |
| 3 | 1 | 1 |
| 1 | 0 | 1 |
| Binary value is: 111 -> 0b00000111 | | |

Add the numbers together:

0b00000100 + 0b00000111 = 0b00001011

1. **7-4**

Binary Conversion Algorithm is utilized on the non-negative value 7:

|  |  |  |
| --- | --- | --- |
|  |  |  |
| 7 | 3 | 1 |
| 3 | 1 | 1 |
| 1 | 0 | 1 |
| Binary value is: 111 -> 0b00000111 | | |

Binary Conversion Algorithm is utilized on the non-negative value 4:

|  |  |  |
| --- | --- | --- |
|  |  |  |
| 4 | 2 | 0 |
| 2 | 1 | 0 |
| 1 | 0 | 1 |
| Binary value is: 100 -> 0b00000100 | | |

Add the numbers together:

0b00000111- 0b00000100= 0b00000011

1. **4-7**

Here we need to calculate two’s complement of -7, in order to add that to 4.

Binary Conversion Algorithm is utilized on the non-negative value 7:

|  |  |  |
| --- | --- | --- |
|  |  |  |
| 7 | 3 | 1 |
| 3 | 1 | 1 |
| 1 | 0 | 1 |
| Binary value is: 111 -> 0b00000111 | | |

Binary Conversion Algorithm is utilized on the non-negative value 4:

|  |  |  |
| --- | --- | --- |
|  |  |  |
| 4 | 2 | 0 |
| 2 | 1 | 0 |
| 1 | 0 | 1 |
| Binary value is: 100 -> 0b00000100 | | |

Invert the bits in 7 and add 1, so we get two’s complement:

0b00000111-> 0b11111000

0b11111000+ 0b00000001 = 0b11111001

Now add these two values:

Solution is:

1. **-4-7**

Here we need to calculate two’s complement of -7 and -4, in order to add that to -4.

Binary Conversion Algorithm is utilized on the non-negative value 7:

|  |  |  |
| --- | --- | --- |
|  |  |  |
| 7 | 3 | 1 |
| 3 | 1 | 1 |
| 1 | 0 | 1 |
| Binary value is: 111 -> 0b00000111 | | |

Binary Conversion Algorithm is utilized on the non-negative value 4:

|  |  |  |
| --- | --- | --- |
|  |  |  |
| 4 | 2 | 0 |
| 2 | 1 | 0 |
| 1 | 0 | 1 |
| Binary value is: 100 -> 0b00000100 | | |

Invert the bits in 7 and add 1, so we get two’s complement:

0b00000111-> 0b11111000

-7 = 0b11111000+ 0b00000001 = 0b11111001

Invert the bits in 4 and add 1, so we get two’s complement:

0b00000100-> 0b11111011

-4 = 0b11111011+ 0b00000001 = 0b11111100

Now add these two values (throw any carryover away):

Solution is: