Assignment 10

1A) ABC + !A!B!C

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A | B | C | !A | !B | !C | ABC | !ABC | OUT |
| 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |

1B) A(B!C + !BC)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A | B | C | !A | !B | !C | AB!C | A!BC | OUT |
| 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

2A) ST + RST

* ST(1 + R) = ST(1) = ST

2B) (A + B (A + C) (!A + !B)

* (A^2 + AC + AB + BC) (!A + !B)
* A^2!A + A^2!B + AC!A + AC!B + AB!B + BC!A + BC!B
* A\* !A + A!B + AC!A + AB!A + AB!B + BC!A + BC!B
* 0 + A!B + 0 \* C + AC!B + 0 \* B + 0 \* A + 0\* C + BC!A
* A!B + AC!B + BC!A
* !B (A + AC) + BC!A

2C) !!D!DA = DD!A

* D + A

2D AB + BA = A!B

* AB + A!B = A(B + !A) = A

3) /\*

1. Write a void C++ function named TruthTable() that takes two int parameters and a bool parameter named op1, op2 and opr respectively. If the absolute values of op1 and op2 are either 1 or 2, the function displays a truth table of the encoded expression of the parameters; otherwise, it displays nothing. For the truth table, true should be represented by 1 and false should be represented by 0. The header of the truth table must be A |B ||R and the columns must be divided as the header. The expression decoding is 1 equals A, -1 equals A, 2 equals B, -2 equals B, true equals AND and false equals OR. For instances, to write the truth table of the expression B + A, the parameters of the function would be -2 , 1 and false respectively; and the display would be

\*/

#include<iostream>

#include<string>

// #include<stdafx.h>

using namespace std;

int main(){

bool Value;bool TRUE;bool FALSE;

cout << " OR Truth Table\n\n";

cout << " 0 + 0 = " << (Value = 0 + 0) << endl;

cout << " 1 + 0 = " << (Value = 1 + 0) << endl;

cout << " 0 + 1 = " << (Value = 0 + 1) << endl;

cout << " 1 + 1 = " << (Value = 1 + 1) << endl << endl;

cout << " FALSE or FALSE = " << (Value = FALSE || FALSE) << endl;

cout << " TRUE or FALSE = " << (Value = TRUE || FALSE) << endl;

cout << " FALSE or TRUE = " << (Value = FALSE || TRUE) << endl;

cout << " TRUE or TRUE = " << (Value = TRUE || TRUE) << endl<< endl;

cout << " AND Truth Table\n\n";

cout << " 0 \* 0 = " << ( Value = 0 \* 0 ) << endl;

cout << " 1 \* 0 = " << ( Value = 1 \* 0 ) << endl;

cout << " 0 \* 1 = " << ( Value = 0 \* 1 ) << endl;

cout << " 1 \* 1 = " << ( Value = 1 \* 1 ) << endl;

cout << "False And False = " << (Value = false && false) << endl;

cout << "True And False = " << (Value = true && false) << endl;

cout << " False And True = " << (Value = false && true) << endl;

cout << " True And True = " << (Value = true && true) << endl<< endl;

cout << " False And False= " << (Value = false && false) << endl;

cout << " True And False= " << (Value = true && false) << endl;

cout << " False And True= " << (Value = false && true) << endl;

cout << " True And True= " << (Value = true && true) << endl;

system ("PAUSE");

return 0;

}

system ("PAUSE");

return 0;

}