МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

кафедра «Вычислительная техника»

**Лабораторная работа №2**

по дисциплине: «Теория автоматов».

Вариант №12

Выполнил:

студент гр. ИВТАПбд-21

Кондратьев П.С.

Проверил:

преподаватель кафедры ВТ

Лылова А.В.

г. Ульяновск, 2018

**Техническое Задание:**

Требуется выполнить программную реализацию автомата указанной модели, выполняющего микропрограмму операций умножения и деления. Входные данные – операнды программа считывает из файла, выполнение осуществляется пошагово, на каждом шаге отображается текущее состояние автомата, значения регистров и счетчиков в двоичном виде.

**Исходный код программы:**

#include <fstream>

#include <iostream>

#include <string>

#include <bitset>

using namespace std;

int main() {

string buff, str1, str2, str3;

string S = "S0 => ";

string Step = "Tr1 = 00000000 Tr2 = 00000000 => ";

string error = "";

ifstream fin("import.txt");

ofstream fout("output.txt");

getline(fin, buff);

if (buff.length() > 18) {

error += "Error:many simbols\n";

}

for (int i = 0; i < buff.length() - 1; i++) {

if (buff[i] != '0' && buff[i] != '1') {

error += "Error:not bin\n";

break;

}

}

str1 = buff.substr(2, 8);

str2 = buff.substr(10, 8);

S += "S1 =>";

Step += "Tr1 = ";

Step += str1;

Step += " Tr2 = ";

Step += str2;

Step += " => ";

if (str2 == "00000000") {

error += "Error:dividing by zero\n";

}

if ((buff[0] == '0' && buff[1] == '1') || (buff[0] == '1' && buff[1] == '0')) {

}

else {

error += "Error:don't know operation ";

error += buff[0];

error += buff[1];

}

if (error == "") {

if (buff[0] == '0' && buff[1] == '1') {

fout << bitset<16>(strtol(str1.c\_str(), NULL, 2) \* strtol(str2.c\_str(), NULL, 2));

str3 = bitset<16>(strtol(str1.c\_str(), NULL, 2) \* strtol(str2.c\_str(), NULL, 2)).to\_string();

Step += "Tr1 \* Tr2 = ";

}

else if (buff[0] == '1' && buff[1] == '0') {

fout << bitset<16>(strtol(str1.c\_str(), NULL, 2) / strtol(str2.c\_str(), NULL, 2));

str3 = bitset<16>(strtol(str1.c\_str(), NULL, 2) / strtol(str2.c\_str(), NULL, 2)).to\_string();

Step += "Tr1 / Tr2 = ";

}

} else {

cout << error;

}

S += "S0\n";

cout << S;

cout << Step << "Tr3 = " << str3;

}

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **Разрядность операндов** | **Код операции деления (g1g2)** | **Код операции умножения (g1g2)** | **Модель автомата** |
| 12 | 8 | 10 | 01 | Мили |

|  |  |
| --- | --- |
| **Код ошибки** | **Ошибка** |
| **#**042 | Error:many simbols |
| #420 | Error:not bin |
| #024 | Error:dividing by zero |
| #402 | Error:don't know operation |

**Результат работы программы:**

1. 101111111111111111 => 0000000000000001
2. 011111111111111111 => 1111111000000001

**Блок-схема и граф:**

****

**Кодирование:**

|  |  |
| --- | --- |
| Y | Соответствующие функции |
| Y1 | string buff, str1, str2, str3;  string S = "S0 => ";  string Step = "Tr1 = 00000000 Tr2 = 00000000 => ";  string error = "";  ifstream fin("import.txt");  ofstream fout("output.txt");  getline(fin, buff); |
| Y2 | error += "Error:many simbols\n"; |
| Y3 | error += "Error:not bin\n";  break; |
| Y4 | i++ |
| Y5 | str1 = buff.substr(2, 8);  str2 = buff.substr(10, 8);  S += "S1 =>";  Step += "Tr1 = ";  Step += str1;  Step += " Tr2 = ";  Step += str2;  Step += " => "; |
| Y6 | error += "Error:dividing by zero\n"; |
| Y7 | error += "Error:don't know operation ";  error += buff[0];  error += buff[1]; |
| Y8 | cout << error; |
| Y9 | fout << bitset<16>(strtol(str1.c\_str(), NULL, 2) \* strtol(str2.c\_str(), NULL, 2));  str3 = bitset<16>(strtol(str1.c\_str(), NULL, 2) \* strtol(str2.c\_str(), NULL, 2)).to\_string();  Step += "Tr1 \* Tr2 = "; |
| Y10 | fout << bitset<16>(strtol(str1.c\_str(), NULL, 2) / strtol(str2.c\_str(), NULL, 2));  str3 = bitset<16>(strtol(str1.c\_str(), NULL, 2) / strtol(str2.c\_str(), NULL, 2)).to\_string();  Step += "Tr1 / Tr2 = "; |
| F1 | if (buff.length() > 18) { |
| F2 | i < buff.length() - 1; |
| F3 | if (buff[i] != '0' && buff[i] != '1') { |
| F4 | if (str2 == "00000000") { |
| F5 | if ((buff[0] == '0' && buff[1] == '1') || (buff[0] == '1' && buff[1] == '0')) { |
| F6 | if (error == "") { |
| F7 | if (buff[0] == '0' && buff[1] == '1') { |
| F8 | if (buff[0] == '1' && buff[1] == '0') { |