**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО**

**ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ  
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В.Г.ШУХОВА»  
(БГТУ им. В.Г.Шухова)**

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

Лабораторная работа №7

Дисциплина: Теория цифровых автоматов

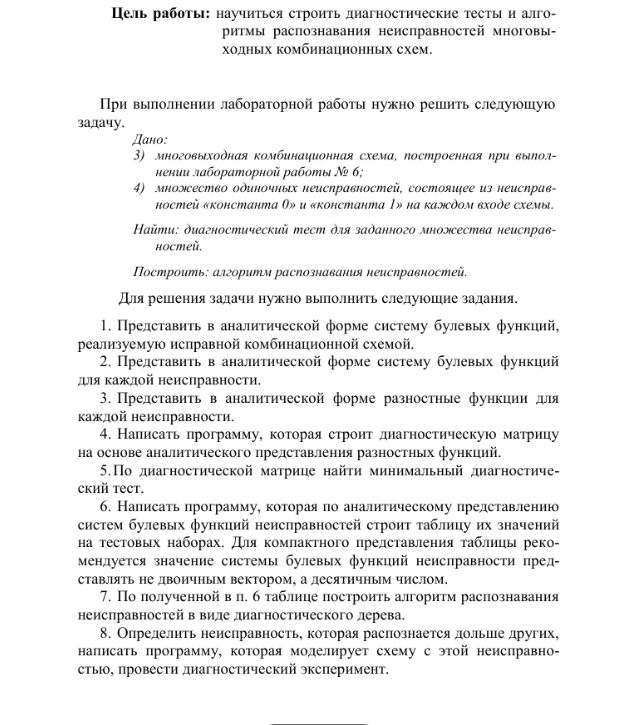
по теме Диагностика неисправностей многовыходных

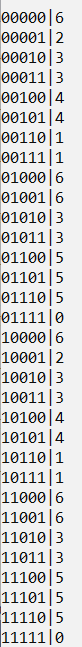
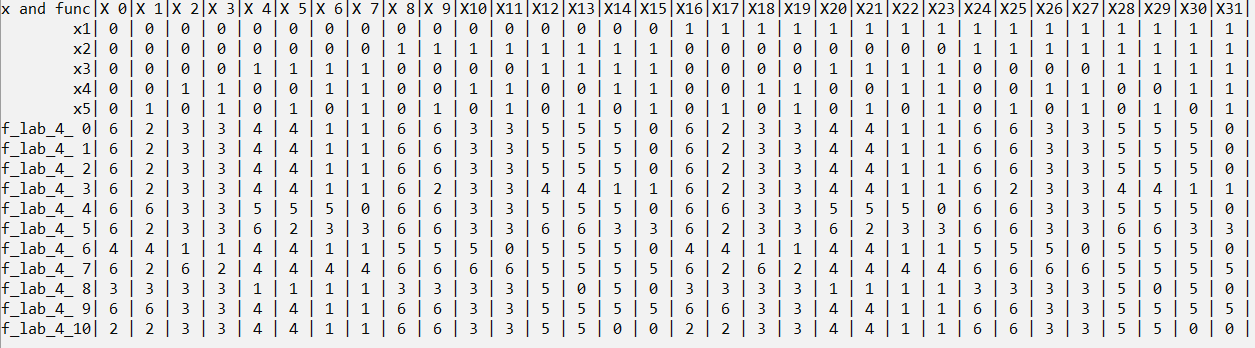
комбинационных схем

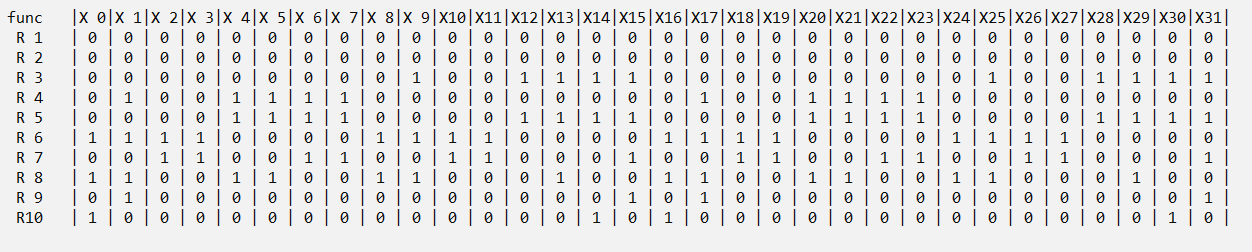
Выполнил: ст. группы ВТ-31  
Новожен Н.В

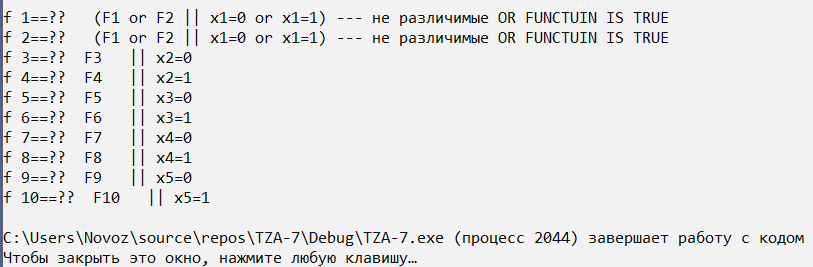
Проверил: Рязанов.Ю.Д

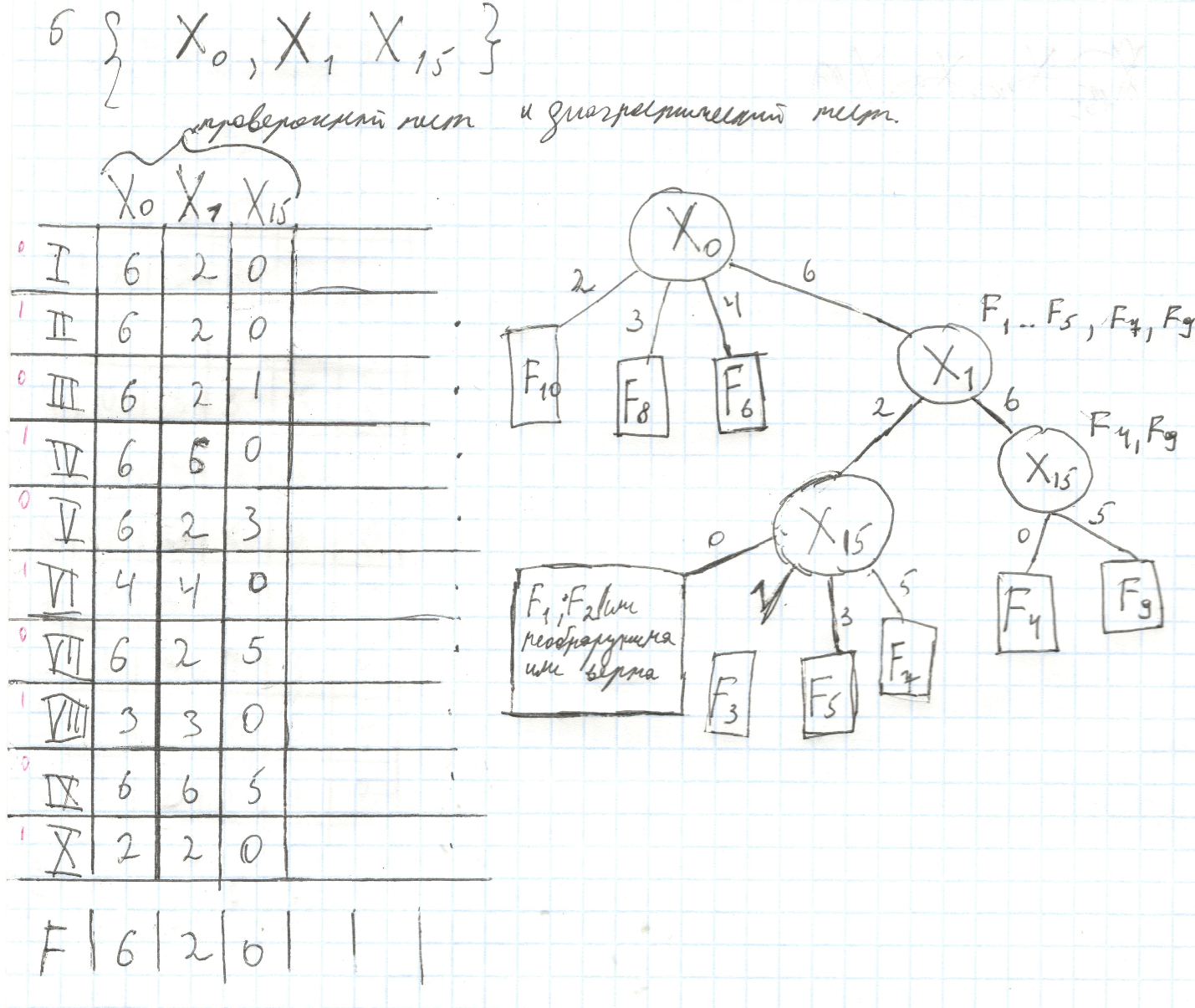
**Белгород 2019**

****

**** ****

****

****



#pragma optimize( "g", on )

#include <iostream>

#include <string>

#include <set>

#include <map>

#include <stack>

#include <queue>

#include <cassert>

#include <cstdlib>

#include <vector>

#include <array>

void dl(int\*\* a, int m)

{

for (size\_t i = 0; i < m; i++)

{

delete[] a[i];

}

delete[] a;

a = nullptr;

}

void GetVector(int\* vect, int pos, int\*\* matr, int m, int n)

{

static int s = 0;

for (size\_t i = 0; i < 2; i++)

{

vect[pos] = i;

if (pos == n - 1)

{

for (int j = 0; j < n; ++j)

{

matr[s][j] = vect[j];

//std::cout << vect[j];

}

//std::cout << '\n';

++s;

}

else GetVector(vect, pos + 1, matr, m, n);

}

}

int\*\* GetTableTrue(int m, int n)

{

int\*\* matr = new int\* [m];

for (int i = 0; i < m; ++i) {

matr[i] = new int[n /\*+1+ 1 + 1 + 1 + 1\*/];

}

auto vect = new int[n];

GetVector(vect, 0, matr, m, n);

delete[] vect;

return matr;

}

struct my { int f1, f2, f3; };

my f\_var1(int x1, int x2, int x3, int x4, int x5)

{

my ff;

int z1 = x2 && x3;

int z2 = x2 && !x3;

int u1 = !x4 && z1;

int u2 = !x5 && z1;

int u3 = !x3 && x4;

int u4 = x4 && !x2;

int u5 = x3;

int u6 = !x4 && z2;

int u7 = !x4 && !x5;

int u8 = x3 && !x4;

int v1 = u1 || u2;

int v2 = u3 || u4;

int v3 = !u5 || u3;

int v4 = u6 || u7;

int w = u8 || v1;

int f1 = v2 || v1;

int f2 = u6 || v3;

int f3 = w || v4;

ff.f1 = f1;

ff.f2 = f2;

ff.f3 = f3;

return ff;

}

my f\_lab\_4(int x1, int x2, int x3, int x4, int x5)

{

return f\_var1(x1, x2, x3, x4, x5);

}

my f\_lab\_4\_10(int x1, int x2, int x3, int x4, int x5) { x1 = 0; return f\_lab\_4(x1, x2, x3, x4, x5); }

my f\_lab\_4\_11(int x1, int x2, int x3, int x4, int x5) { x1 = 1; return f\_lab\_4(x1, x2, x3, x4, x5); }

my f\_lab\_4\_20(int x1, int x2, int x3, int x4, int x5) { x2 = 0; return f\_lab\_4(x1, x2, x3, x4, x5); }

my f\_lab\_4\_21(int x1, int x2, int x3, int x4, int x5) { x2 = 1; return f\_lab\_4(x1, x2, x3, x4, x5); }

my f\_lab\_4\_30(int x1, int x2, int x3, int x4, int x5) { x3 = 0; return f\_lab\_4(x1, x2, x3, x4, x5); }

my f\_lab\_4\_31(int x1, int x2, int x3, int x4, int x5) { x3 = 1; return f\_lab\_4(x1, x2, x3, x4, x5); }

my f\_lab\_4\_40(int x1, int x2, int x3, int x4, int x5) { x4 = 0; return f\_lab\_4(x1, x2, x3, x4, x5); }

my f\_lab\_4\_41(int x1, int x2, int x3, int x4, int x5) { x4 = 1; return f\_lab\_4(x1, x2, x3, x4, x5); }

my f\_lab\_4\_50(int x1, int x2, int x3, int x4, int x5) { x5 = 0; return f\_lab\_4(x1, x2, x3, x4, x5); }

my f\_lab\_4\_51(int x1, int x2, int x3, int x4, int x5) { x5 = 1; return f\_lab\_4(x1, x2, x3, x4, x5); }

typedef my (\*rrr)(int, int, int, int, int);

rrr\* arr\_fun = nullptr;

void show\_step3\_lab4(int\*\* a, int m, int n)

{

std::cout << " func |";

for (int i = 0; i < n; ++i)

{

std::cout << "X";

std::cout.width(2);

std::cout << i << "|";

}

std::cout << "\n";

int t = 0;

for (int i = 0; i < m; ++i)

{

std::cout << " R"; std::cout.width(2); std::cout << i + 1;

std::cout << " | ";

for (int j = 0; j < n; ++j)

{

std::cout << (a[i][j] == 1 ? 1 : 0) << " | ";

}

std::cout << "\n";

}

std::cout << "\n\n\n";

}

void show\_step2\_lab4(int\*\* a, int m, int n, int count\_x)

{

std::cout << "x and func|";

for (int i = 0; i < n; ++i)

{

std::cout << "X";

std::cout.width(2);

std::cout << i + 1 - 1 << "|";

}

std::cout << "\n";

int t = 0;

for (int i = 0; i < m; ++i)

{

if (i < count\_x) {

std::cout.width(9); std::cout << "x" << i + 1;

}

else { std::cout << "f\_lab\_4\_"; std::cout.width(2); std::cout << t++; }

std::cout << "| ";

for (int j = 0; j < n; ++j)

{

std::cout << a[i][j] << " | ";

}

std::cout << "\n";

}

std::cout << "\n\n\n";

}

void FillArrFun(rrr\* arr\_fun, int coun)

{

arr\_fun[0] = &f\_lab\_4;

arr\_fun[1] = &f\_lab\_4\_10;

arr\_fun[2] = &f\_lab\_4\_11;

arr\_fun[3] = &f\_lab\_4\_20;

arr\_fun[4] = &f\_lab\_4\_21;

arr\_fun[5] = &f\_lab\_4\_30;

arr\_fun[6] = &f\_lab\_4\_31;

arr\_fun[7] = &f\_lab\_4\_40;

arr\_fun[8] = &f\_lab\_4\_41;

arr\_fun[9] = &f\_lab\_4\_50;

arr\_fun[10] = &f\_lab\_4\_51;

}

int\*\* viev\_lab4\_tabl1(int\*\* a, int m, int n, rrr\* arr\_fun, int coun)

{

int\*\* matr = new int\* [n + coun];

for (int i = 0; i < n + coun; ++i) {

matr[i] = new int[m];

}

for (int i = 0; i < m; ++i) {

for (int j = 0; j < n; ++j) {

matr[j][i] = a[i][j];

}

}

for (int i = 0; i < coun; ++i) {

for (int j = 0; j < m; ++j) {

auto tmp =arr\_fun[i](matr[0][j], matr[1][j], matr[2][j], matr[3][j], matr[4][j]);

matr[i + n][j] = tmp.f1 + tmp.f2\*2 + tmp.f3\*4;

}

}

return matr;

}

int XR(int i, int j){ return (i && !j) || (!i && j); }

int Xor(int i, int j) { return XR (i & 1,j & 1) + XR(i & 2, j & 2)\*2 + XR(i & 4, j & 4)\*4 ; }

int\*\* step3\_lab3(int\*\* a, int m, int n, int coun)//return R size =(coun-1,m)

{

int\*\* matr = new int\* [coun - 1];

for (int i = 0; i < coun - 1; ++i) {

matr[i] = new int[n];

}

int t = m - coun;//f

for (int i = 0; i < coun - 1; ++i) {

for (int j = 0; j < n; ++j) {

matr[i][j] = (Xor(a[t][j], a[t + i + 1][j])>0?1:0);

}

}

return matr;

}

int\*\* get4\_lab4(int\*\* a, int m, int n, int k, int x1, int x2, int x3, int x4, int x5, int x6, int x7, int x8, int x9, int x10)

{

int\*\* matr = new int\* [m];

for (int i = 0; i < m; ++i) {

matr[i] = new int[k];

}

int kk = 0;

for (int i = 0; i < n; ++i) {

if ((kk < k) && (x1 == i || x2 == i || x3 == i || x4 == i || x5 == i || x6 == i || x7 == i || x8 == i || x9 == i || x10 == i))

{

for (int j = 0; j < m; ++j) {

matr[j][i] = a[j][i];

}

kk++;

}

}

return matr;

}

void GetVector4(int\* vect, int pos, int\*\* matr, int m, int n)

{

static int s = 0;

for (size\_t i = 0; i < 2; i++)

{

vect[pos] = i;

if (pos == n - 1)

{

for (int j = 0; j < n; ++j)

{

matr[s][j] = vect[j];

}

++s;

}

else GetVector4(vect, pos + 1, matr, m, n);

}

}

int\*\* GetTableTrue\_lab4(int m, int n)

{

int\*\* matr = new int\* [m];

for (int i = 0; i < m; ++i) {

matr[i] = new int[n + 1 ];

}

auto vect = new int[n];

GetVector4(vect, 0, matr, m, n);

delete[] vect;

return matr;

}

void fun\_lab4(int\*\* a, int m, int n, rrr pf) {

for (size\_t i = 0; i < m; i++)

{

auto tmp= pf(a[i][0], a[i][1], a[i][2], a[i][3], a[i][4]);//это f1

a[i][n]= tmp.f1 + tmp.f2 \* 2 + tmp.f3 \* 4;

}

}

std::string Is\_Error\_Lab4(int\*\* a, int m, int n)

{

if (a[0][n] == 2) return "F10 || x5=1";

if (a[0][n] == 3) return "F8 || x4=1";

if (a[0][n] == 4) return "F6 || x3=1";

if (a[1][n] == 6)

{

if(a[15][n] == 0)return "F4 || x2=1";

if(a[15][n] == 5)return "F9 || x5=0";

}

if (a[15][n] == 1)return "F3 || x2=0";

if (a[15][n] == 3)return "F5 || x3=0";

if (a[15][n] == 5)return "F7 || x4=0";

return " (F1 or F2 || x1=0 or x1=1) --- не различимые OR FUNCTUIN IS TRUE";

}

void proverka(int m, int n)

{

auto mt = GetTableTrue\_lab4(m, n);

for (size\_t i = 0; i < 10; i++)

{

fun\_lab4(mt, m, n, arr\_fun[i + 1]);

std::cout << "f " << i + 1 << "==?? " << Is\_Error\_Lab4(mt, m, n) << std::endl;

}

dl(mt, m);

}

void viev\_lab4(int\*\* a, int m, int n)

{

for (size\_t i = 0; i < m; i++)

{

//a[i][n] = f\_lab\_4(a[i][0], a[i][1], a[i][2], a[i][3], a[i][4]);//это f1

for (int j = 0; j < n; ++j)

{

std::cout << (a[i][j] ? 1 : 0);

}

auto tmm = (f\_lab\_4(a[i][0], a[i][1], a[i][2], a[i][3], a[i][4]) );

std::cout << "|" << (tmm.f1 + tmm.f2\*2 + tmm.f3\*4) << '\n';

}

int k = 11;

arr\_fun = new rrr[k];

FillArrFun(arr\_fun, k);

auto matr = viev\_lab4\_tabl1(a, m, n, arr\_fun, k);

dl(a, m);

int n1 = m;

int m1 = n + k;

show\_step2\_lab4(matr, m1, n1, n);

auto Arr = step3\_lab3(matr, m1, n1, k);

show\_step3\_lab4(Arr, k - 1, n1);

proverka(m, n);

dl(matr, m1);

dl(Arr, k - 1);

delete[] arr\_fun;

}

int main() {

setlocale(LC\_ALL, "Russian");

int n = 5;

int m = 32;

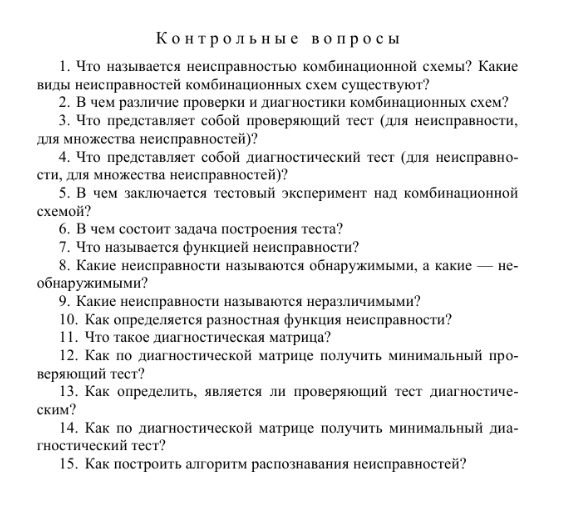
auto a = GetTableTrue(m, n);

std::cout << '\n';

viev\_lab4(a, m, n);//f верна

return 0;

}

****