**Міністерство освіти і науки України**

**Національний університет «Львівська політехніка»**

**Кафедра ЕОМ**



Звіт

до лабораторної роботи № 2

з дисципліни «Комп’ютерні мережі»

на тему: «**Ознайомлення з основними конструкціями мови моделювання System C**»

Варіант 29

Виконав:

ст.гр. КІ-38

Яців О. А.

Прийняв:

Козак Н. Б.

**Львів 2022**

**МЕТА РОБОТИ:**

Ознайомлення з основними конструкціями мови моделювання System C.

**Завдання**

***Вхідні дані:***

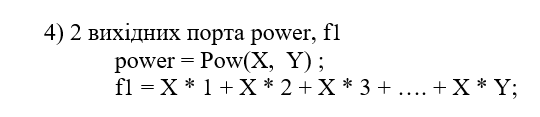
Х = № варіанту за списком в журналі.

Y = сума ASCII code першої літери прізвища + першої літери імені.

X = 29

Y= Y + O = 89 + 79 = 168

Реалізувати модулі S1 та S2, разом з логікою їх функціонування згідно варіанту, провести послідовне з’єднання S1 та S2 ініціалізувати необхідні порти на S1. На вхід S2 подати вихідні порти модуля S1. До кожного модуля заводиться зовнішній вхідний для всіх сигнал синхронізації CLK. Результати подати на модуль Display для відображення.



f1 можна розписати як x\*(1+2+3+…+y), a (1+2+3+…+y)=(y\*(y+1))/2, тому f1 = x\*((y\*(y+1))/2)

|  |  |  |
| --- | --- | --- |
| Варіант | | Завдання |
| 9, 19, 29 | Обчислити r1 = OR (a, b) , r2 = NOT( OR (a, b)); операції NOT & OR побітово. | | |

**Хід роботи**

X f1 r1

S2

r1 = OR(a + b ),

r2 = NOT(OR(a,b))

S1

power = POW(X, Y) ;

f1 = X\*1+X\*2+X\*3+….+X\*Y;

Y rep r2

CLK

Рис.1. Створена схема відповідно до варіанту

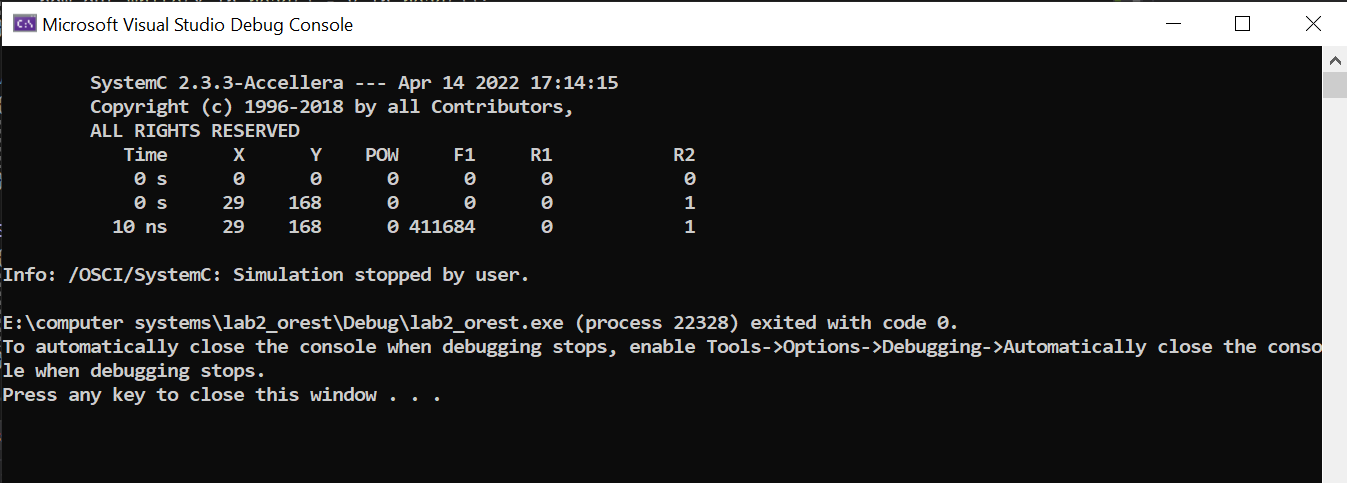


Рис.2. Результат виконання програми

**Код програми**

s1\_unit.h

#pragma once

#include "systemc.h"

SC\_MODULE(s1\_unit)

{

sc\_in<int> x\_in, y\_in;

sc\_out<int> pow\_out, f1\_out;

void outPOW()

{

pow\_out.write(x\_in.read() - y\_in.read());

}

void outF1()

{

f1\_out.write(x\_in.read()\*((y\_in.read()\*(y\_in.read() + 1))/2) );

}

SC\_CTOR(s1\_unit)

{

SC\_METHOD(outF1);

sensitive << x\_in << y\_in;

}

};

s2\_unit.h

#pragma once

#include "systemc.h"

#include "s1\_unit.h"

SC\_MODULE(s2\_unit)

{

sc\_in<int> pow\_in, f1\_in;

sc\_out<int> r1, r2;

void OutR1()

{

int result = pow\_in.read() & f1\_in.read();

r1.write(result);

}

void OutR2()

{

OutR1();

r2.write( ( ! (pow\_in.read() & f1\_in.read() ) ) );

}

SC\_CTOR(s2\_unit)

{

SC\_METHOD(OutR2);

sensitive << pow\_in << f1\_in;

}

};

Monitor.h

#pragma once

#include "systemc.h"

#include <iomanip>

SC\_MODULE(Monitor)

{

sc\_in<int> X, Y;

sc\_in<int> POW, F1;

sc\_in<int> R1, R2;

sc\_in<bool> CLK;

void monitor()

{

cout << std::setw(15) << "Time";

cout << std::setw(7) << "X";

cout << std::setw(7) << "Y";

cout << std::setw(7) << "POW";

cout << std::setw(7) << "F1";

cout << std::setw(7) << "R1";

cout << "\t" << std::setw(7) << "R2" << endl;

for (int i = 0; i < 3; i++)

{

cout << std::setw(15) << sc\_time\_stamp();

cout << std::setw(7) << X.read();

cout << std::setw(7) << Y.read();

cout << std::setw(7) << POW.read();

cout << std::setw(7) << F1.read();

cout << std::setw(7) << R1.read();

cout << "\t" << std::setw(7) << R2.read() << endl;

wait();

}

sc\_stop();

}

SC\_CTOR(Monitor)

{

SC\_THREAD(monitor);

sensitive << CLK.pos();

}

};

Stimulator.h

#pragma once

#include "systemc.h"

SC\_MODULE(Stimulator)

{

sc\_out<int> X, Y;

sc\_in<bool> CLK;

void Generator()

{

X.write(29);

Y.write(89 + 79);

wait();

}

SC\_CTOR(Stimulator)

{

SC\_THREAD(Generator);

sensitive << CLK.pos();

}

};

lab2.cpp

#include "systemc.h"

#include "Stimulator.h"

#include "Monitor.h"

#include "s1\_unit.h"

#include "s2\_unit.h"

int sc\_main(int argc, char\* argv[])

{

sc\_signal<int> XSig, YSig;

sc\_signal<int> POWSig, F1Sig;

sc\_signal<int> R1Sig, R2Sig;

sc\_clock TestClk("TestClock", 10, SC\_NS, 0.5);

Stimulator Stim1("stim");

Stim1.X(XSig);

Stim1.Y(YSig);

Stim1.CLK(TestClk);

s1\_unit first\_part("s1\_unit");

first\_part.x\_in(XSig);

first\_part.y\_in(YSig);

first\_part.pow\_out(POWSig);

first\_part.f1\_out(F1Sig);

s2\_unit second\_part("s2\_unit");

second\_part.pow\_in(POWSig);

second\_part.f1\_in(F1Sig);

second\_part.r1(R1Sig);

second\_part.r2(R2Sig);

Monitor Monitor1("monitor");

Monitor1.X(XSig);

Monitor1.Y(YSig);

Monitor1.POW(POWSig);

Monitor1.F1(F1Sig);

Monitor1.R1(R1Sig);

Monitor1.R2(R2Sig);

Monitor1.CLK(TestClk);

sc\_start();

return 0;

}

**Висновок:** Я ознайомився з термінологією, специфікацією та іншими основними поняттями мови моделювання SystemC, вивчив особливості основної мови моделювання процесів SystemC, які можуть бути реалізовані як апаратно (переважно), так і програмним шляхом. А також дав відповіді на контрольні запитання.