**Міністерство освіти і науки України**

**Національний університет "Львівська політехніка"**

****

**Звіт до лабораторної роботи №2**

**з дисципліни “Комп’ютерні системи”**

**Ознайомлення з основними конструкціями мови моделювання System C**

Виконав ст. гр. КІ-33:

Теслюк Н.А.

Прийняв: Козак Н.Б.

**Львів 2020 р.**

**Мета роботи:** Ознайомлення з основними конструкціями мови моделювання System C.

**Вхідні дані**

Х = № варіанту за списком в журналі.

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

Х = 1

Y = 144 // А + О; А = 65, О = 79

**Завдання**

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.

**Завдання по реалізації модуля S1**

2 вихідних порта o1, o2

o1 = X – Y;

o2 = (X + Y / 2.0) && (X >> 2);

**Завдання по реалізації модуля S2**

2 вихідні порти з модуля S1 довільно під’єднуються до вхідних портів модуля S2: **a** та **b.**

2 вихідних порта r1, r2



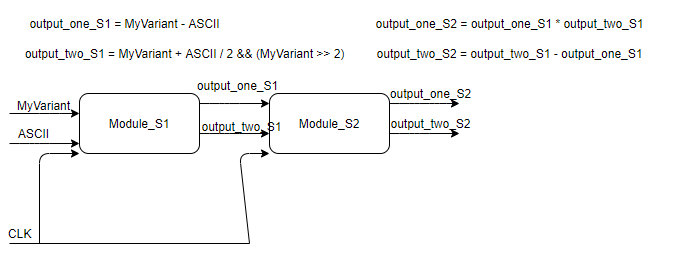


Рис.1 Схема підключення модулів

**Код програми**

#include "stdafx.h"

SC\_MODULE(Module\_S1) {

sc\_in<bool> CLK;

sc\_in<int> X, Y;

sc\_out<int> output\_one, output\_two;

void S1\_mod() {

output\_one.write(X.read() - Y.read());

output\_two.write((X.read() + Y.read() / 2) && X.read() >> 2);

}

// constructor for module

SC\_CTOR(Module\_S1) {

SC\_METHOD(S1\_mod);

// sensitive << X << Y;

sensitive << X << Y << CLK.pos();

}

};

SC\_MODULE(Module\_S2) {

sc\_in<bool> CLK;

sc\_in<int> A, B;

sc\_out<int> output\_one, output\_two;

void S2\_mod() {

output\_one.write(A.read() \* B.read());

output\_two.write(B.read() - A.read());

}

// constructor for module

SC\_CTOR(Module\_S2) {

SC\_METHOD(S2\_mod);

//sensitive << A << B ;

sensitive << A << B << CLK.pos();

}

};

SC\_MODULE(display) {

sc\_in<bool> CLK;

sc\_in<int> output\_one\_S1;

sc\_in<int> output\_two\_S1;

sc\_in<int> output\_one\_S2;

sc\_in<int> output\_two\_S2;

void print()

{

cout << "Module\_S1 output\_one is " << output\_one\_S1.read() << endl;

cout << "Module\_S1 output\_two is " << output\_two\_S1.read() << endl;

cout << "Module\_S2 output\_one is " << output\_one\_S2.read() << endl;

cout << "Module\_S2 output\_two is " << output\_two\_S2.read() << endl;

}

SC\_CTOR(display) {

SC\_METHOD(print);

sensitive << CLK.pos();

}

public:

};

int sc\_main(int argc, char\*\* argv) {

sc\_clock CLK{ "clk", 1, SC\_NS };

sc\_signal<int> myVariant, myASCIICode;

myVariant = 1;

myASCIICode = 144; // A + O; A = 65, O = 79

sc\_signal<int> output\_one\_S1, output\_two\_S1;

Module\_S1 moduleS1("Module\_S1");

moduleS1.CLK(CLK);

moduleS1.X(myVariant);

moduleS1.Y(myASCIICode);

moduleS1.output\_one(output\_one\_S1);

moduleS1.output\_two(output\_two\_S1);

sc\_signal<int> output\_one\_S2, output\_two\_S2;

Module\_S2 moduleS2("Module\_S2");

moduleS2.CLK(CLK);

moduleS2.A(output\_one\_S1);

moduleS2.B(output\_two\_S1);

moduleS2.output\_one(output\_one\_S2);

moduleS2.output\_two(output\_two\_S2);

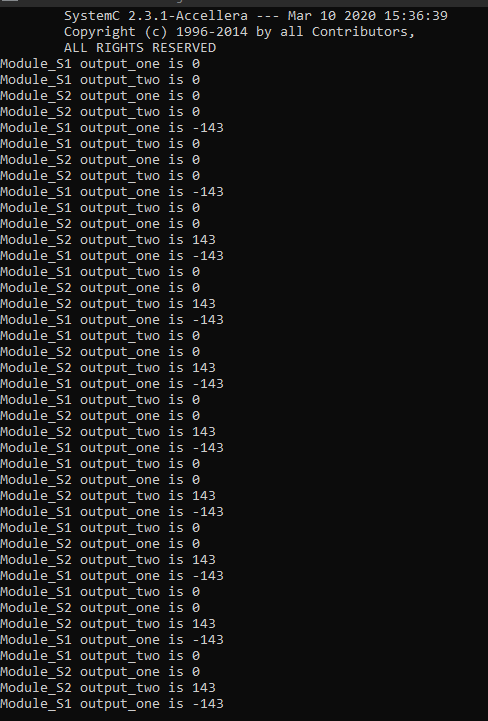
display disp("disp");

disp(CLK, output\_one\_S1, output\_two\_S1, output\_one\_S2, output\_two\_S2);

sc\_start(50, SC\_NS);

return 0;

}



**Висновок**

На даній лабораторній роботі я ознайомився з основними конструкціями мови system C.