**Міністерство освіти і науки України**

**Національний університет "Львівська політехніка"**

**Кафедра ЕОМ**



**Звіт з лабораторної роботи №2**

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

**на тему: ” Ознайомлення з основними поняттями мови моделювання SystemC”**

**Виконав: студент .гр. КІ-33**

**Барсієнко В. О.**

**Прийняв: викладач**

**Козак Н. Б.**

**Львів 2020 р.**

**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.

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.

1) 2 вихідних порта o1, o2

o1 = X – Y;

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

X = 5, Y = K + V = 75 + 86 = 161

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

Варіанти для реалізації модуля S2:

|  |  |
| --- | --- |
| Варіант | Завдання |
| 1, 11, 21 | Обчислити r1 = a \* b, r2 = b – a; |

**Хід роботи:**

1. Реалізовую модуль S1

#include <systemc.h>

// X = 1 - номер варіанту

// Y = B + V = 66 + 86 = 152 - сума ASCII першої літери Пр і Імя

// out o1 = X - Y, o2 = (X + Y/2.0) && (x >> 2)

const int X = 1;

const int Y = 152;

SC\_MODULE(S1) {

sc\_out<int> o1, o2;

sc\_in<bool> clk;

void mainFunc() {

if (clk.read() == 0) {

o1.write(X - Y);

o2.write((X + Y / 2) && (X >> 2));

cout << "clk read: " << clk.read() << endl;

cout << "o1 val: " << o1.read() << endl;

cout << "o2 val: " << o2.read() << endl;

}

else

{

o1.write(0);

o2.write(0);

cout << "clk read: " << clk.read() << endl;

cout << "o1 val: " << o1.read() << endl;

cout << "o2 val: " << o2.read() << endl;

}

}

SC\_CTOR(S1) {

SC\_METHOD(mainFunc);

sensitive << clk.pos() << clk.neg();

}

};

1. Реалізовую модуль S2

#include <systemc.h>

// r1 = a \* b

// r2 = b - a

SC\_MODULE(S2) {

sc\_out<int> r1, r2;

sc\_in<int> a, b;

sc\_in<bool> clk;

void mainFunc() {

if (clk.read() == 1) {

r1.write(a.read() \* b.read());

r2.write(b.read() – a.read());

cout << "a val: " << a.read() << endl;

cout << "b val: " << b.read() << endl;

cout << "r1 val: " << r1.read() << " abs: " << a.read() \* b.read() << endl << endl;

cout << "r2 val: " << r2.read() << " abs: " << b.read() - a.read() << endl << endl;

}

else

{

r1.write(0);

r2.write(0);

cout << "a val: " << a.read() << endl;

cout << "b val: " << b.read() << endl;

cout << "r1 val: " << r1.read() << " abs: " << a.read() \* b.read() << endl << endl;

cout << "r2 val: " << r2.read() << " abs: " << b.read() - a.read() << endl << endl;

}

}

SC\_CTOR(S2) {

SC\_METHOD(mainFunc);

sensitive << clk.pos() << clk.neg();

}

};

1. Реалізовую спільний модуль (S1 + s2)

#include <systemc.h>

#include "S1.h"

#include "S2.h"

#include <iostream>

// S1 + s2

SC\_MODULE(TestModule) {

sc\_out<int> r1, r2;

sc\_in<bool> clk;

S1 s1;

S2 s2;

sc\_signal<int> toO1, toO2;

sc\_signal<bool> innerClk;

SC\_CTOR(TestModule) : s1("S1"), s2("S2") {

s1.o1(toO1);

s1.o2(toO2);

s2.a(toO1);

s2.b(toO2);

s2.clk(clk);

s1.clk(clk);

s2.r1(r1);

s2.r2(r2);

}

};

Реалізовую функцію для симуляції:

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

sc\_signal<int> out1, out2;

sc\_clock clock("clk", 10, SC\_NS, 0.5, 10, SC\_NS, false);

TestModule test("Test module");

test.clk(clock);

test.r1(out1);

test.r2(out2);

// Open VCD file

sc\_trace\_file\* wf = sc\_create\_vcd\_trace\_file("TestModule");

// Dump the desired signals

sc\_trace(wf, out1, "out1");

sc\_trace(wf, out2, "out2");

sc\_trace(wf, clock, "clk");

// clock = 0;

sc\_start(20, SC\_NS);

sc\_start(20, SC\_NS);

sc\_start(20, SC\_NS);

sc\_close\_vcd\_trace\_file(wf);

return 0;// Terminate simulation

}

Запускаю симуляцію, відкриваю waveform

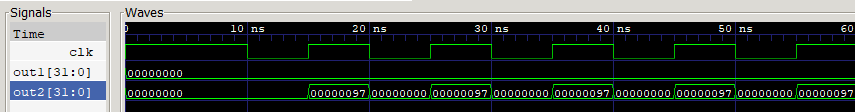


Рис 1. Форма симуляції

Висновок: Ознайомився з основними конструкціями мови SystemC.