

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-38
Хомин В.Б
Перевірив:
Козак Н.Б

Львів 2022 р.

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

Вхідні дані

X = № варіанту за списком в журналі.

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

X = 24

Y = 156

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

4) 2 вихідних порта power, f1

power = Pow(X, Y) ;

f1 = $X * 1 + X * 2 + X * 3 + \dots + X * Y$;

Завдання по реалізації модуля S2

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

4, 14, 24	Обчислити $r1 = a * a$, $r2 = b * b$;
-----------	---

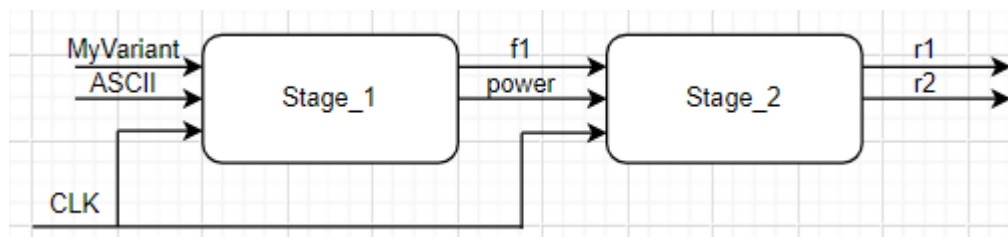


Рис.1 Схема підключення модулів

Код програми

```
// All systemc modules should include systemc.h header file
#include "systemc.h"
#include<iomanip>
#include <iostream>
#include "math.h"
```

```
double calc (double x, double y)
{
    double c = 0;
    for (int i = 1; i <= y; i++)
    {
```

```

        c += x * i;
    }
    return c;
}

SC_MODULE(Module_S1) {
    sc_in<bool> CLK;
    sc_in<double> X, Y;
    sc_out<double> f1, power;
    void S1_mod() {
        f1.write(calc(X.read(), Y.read()));
        power.write(pow(X.read(), Y.read()));
    }

    // constructor for module
    SC_CTOR(Module_S1) {
        SC_METHOD(S1_mod);
        // sensitive << X << Y;
        sensitive << X << Y << CLK.pos();
    }
};

//r1 = a * a, r2 = b * b;
SC_MODULE(Module_S2) {
    sc_in<bool> CLK;
    sc_in<double> A, B;
    sc_out<double> r1, r2;

    void S2_mod() {
        r1.write(A.read() * A.read());
        r2.write(B.read() * B.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<double> f1_S1;
    sc_in<double> power_S1;
    sc_in<double> r1_S2;
    sc_in<double> r2_S2;
    void print()
    {
        cout << "Module_S1 f1 is " << f1_S1.read() << endl;
        cout << "Module_S1 power is " << power_S1.read() << endl;
        cout << "Module_S2 r1 is " << r1_S2.read() << endl;
        cout << "Module_S2 r2 is " << r2_S2.read() << endl;
    }

    SC_CTOR(display) {
        SC_METHOD(print);
        sensitive << CLK.pos();
    }
public:
};

int sc_main(int argc, char** argv) {
    sc_core::sc_report_handler::set_actions("/IEEE Std 1666/deprecated",
    sc_core::SC_DO_NOTHING);
    sc_clock CLK{ "clk", 1, SC_NS };

```

```

sc_signal<double> myVariant, myASCIICode;

myVariant = 24.0;

myASCIICode = 156.0; // S + I (83+73 = 156)
while (myASCIICode.get_new_value() == 0)
{
    std::cout << "Reenter a myASCIICode: ";
    int a = 0;
    std::cin >> a;
    myASCIICode = a;
}
sc_signal<double> f1_S1, power_S1;
Module_S1 moduleS1("Module_S1");
moduleS1.CLK(CLK);
moduleS1.X(myVariant);
moduleS1.Y(myASCIICode);
moduleS1.f1(f1_S1);
moduleS1.power(power_S1);
sc_signal<double> r1_S2, r2_S2;
Module_S2 moduleS2("Module_S2");
moduleS2.CLK(CLK);
moduleS2.A(f1_S1);
moduleS2.B(power_S1);
moduleS2.r1(r1_S2);
moduleS2.r2(r2_S2);

display disp("disp");
disp(CLK, f1_S1, power_S1, r1_S2, r2_S2);
sc_start(50, SC_NS);

return 0;
}

```

