Міністерство освіти і науки України

Національний університет „Львівська політехніка”

Кафедра ЕОМ



**Звіт**

з лабораторної роботи №4

з дисципліни: “Комп’ютерні системи”

на тему: “Аналіз програмної моделі процесу роботи

арифметичного конвеєра, ч.2”

Виконав: ст. гр. КІ-32

Кононенко Р.Ю.

Прийняв:

Козак Н.Б.

Львів – 2020

**Мета роботи:** навчитись здійснювати аналіз програмних моделей комп’ютерних систем, виконаних на мові SystemC.

**Індивідуальне завдання:** здійснити модернізацію функцій або параметрів арифметичного конвеєра (див. лабораторну роботу № 3), шляхом під’єднання розроблених модулів S1 та S2 (див. лабораторну роботу № 2). Порядок та тип з’єднання мають бути обгрунтовані, можливо розробка буферних або додаткових модулів з метою надавання нових властивостей тестувальній моделі.

**Хід роботи:**

1. Модифікував файл *main.cpp* програмної моделі арифметичного конвеєра згідно індивідуального завдання:

**Лістинг модифікованої програмної моделі**

#include "systemc.h"

#include "stage1.h"

#include "stage2.h"

#include "stage3.h"

#include "S1.h"

#include "S2.h"

#include "display.h"

#include "numgen.h"

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

sc\_core::sc\_report\_handler::set\_actions("/IEEE\_Std\_1666/deprecated",

sc\_core::SC\_DO\_NOTHING);

//Signals

sc\_signal<double> in1;

sc\_signal<double> in2;

sc\_signal<double> sum;

sc\_signal<double> diff;

sc\_signal<double> prod;

sc\_signal<double> f1;

sc\_signal<double> r1;

sc\_signal<double> r2;

sc\_signal<double> f2;

sc\_signal<double> quot;

sc\_signal<double> powr;

//Clock

sc\_signal<bool> clk;

numgen N("numgen"); //instance of `numgen' module

N(in1, in2, clk); //Positional port binding

S1 s1("s1");

S2 s2("s2");

s1.clk(clk);

s2.clk(clk);

s1.x(in1);

s1.y(in2);

s1.f2(f2);

s1.f1(f1);

s2.a(f1);

s2.b(f2);

s2.r1(r1);

s2.r2(r2);

stage1 S\_1("stage1"); //instance of `stage1' module

S\_1(r1, r2, sum, diff, clk); //Positional port binding

stage2 S\_2("stage2"); //instance of `stage2' module

S\_2(sum, diff, prod, quot, clk); //Positional port binding

stage3 S\_3("stage3"); //instance of `stage3' module

S\_3(prod, quot, powr, clk); //Positional port binding

display D("display"); //instance of `display' module

D(powr, clk, diff); //Positional port binding

sc\_start(0, SC\_NS); //Initialize simulation

for (int i = 0; i < 50; i++) {

clk.write(1);

sc\_start(10, SC\_NS);

clk.write(0);

sc\_start(10, SC\_NS);

}

return 0;

}

*S1.h*

#pragma once

#ifndef TEST\_SYSTEMC\_S1\_H

#define TEST\_SYSTEMC\_S1\_H

#include "systemc.h"

#include "cmath"

#include "S2.h"

SC\_MODULE(S1) {

sc\_in<double> x;

sc\_in<double> y;

sc\_out<double> f1;

sc\_out<double> f2;

sc\_in<bool> clk;

SC\_CTOR(S1) {

SC\_METHOD(calc\_f2);

sensitive << x << y;

sensitive << clk.pos();

SC\_METHOD(calc\_f1);

sensitive << x << y;

sensitive << clk.pos();

}

void calc\_f2() {

int x\_val = x.read();

int y\_val = y.read();

double acc = 0;

if (y\_val != 0) {

acc += x\_val / y\_val;

}

f2.write(acc);

}

void calc\_f1() {

int x\_val = x.read();

int y\_val = y.read();

int acc = 0;

int factorialX = 1;

for (int i = 1; i <= x\_val; ++i) {

factorialX \*= i;

}

acc = y\_val + factorialX;

f1.write(acc);

}

};

#endif TEST\_SYSTEMC\_S1\_H

*S2.h*

#pragma once

#ifndef TEST\_SYSTEMC\_S2\_H

#define TEST\_SYSTEMC\_S2\_H

#include "systemc.h"

SC\_MODULE(S2) {

sc\_in<double> a;

sc\_in<double> b;

sc\_out<double> r1;

sc\_out<double> r2;

sc\_in<bool> clk;

SC\_CTOR(S2) {

SC\_METHOD(calc\_r1);

sensitive << a << b;

sensitive << clk.pos();

SC\_METHOD(calc\_r2)

sensitive << a << b;

sensitive << clk.pos();

}

void calc\_r1() {

int a\_val = a.read();

int b\_val = b.read();

r1.write(a\_val & b\_val);

}

void calc\_r2() {

int a\_val = a.read();

int b\_val = b.read();

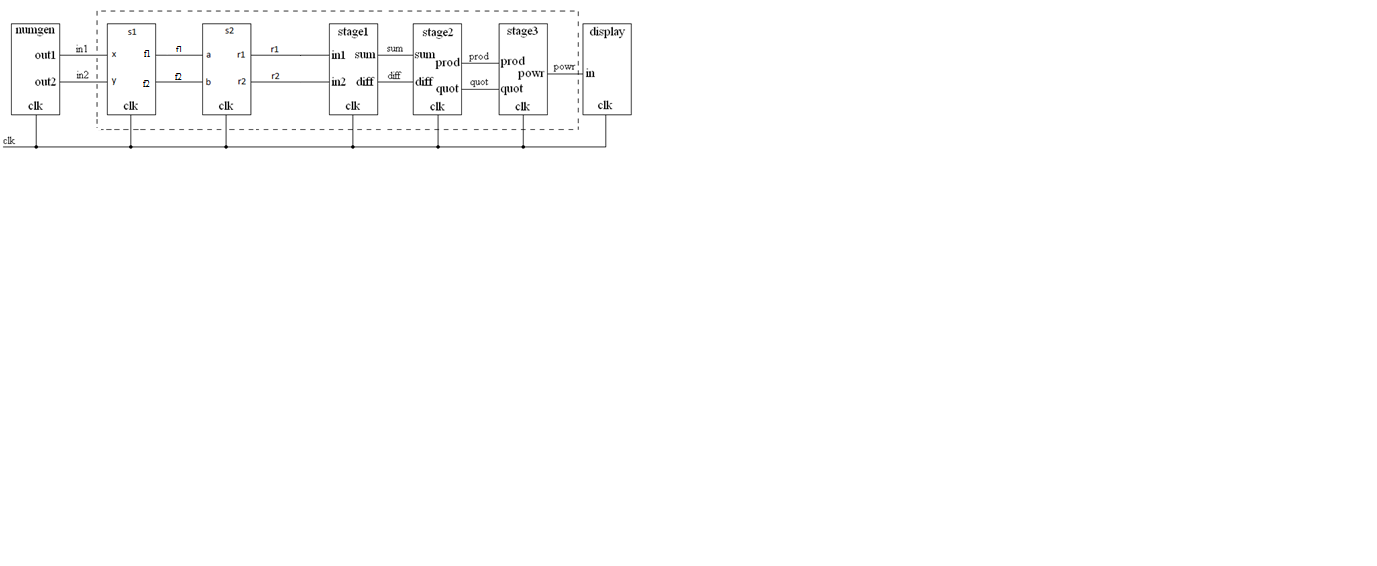
r2.write(8);

}

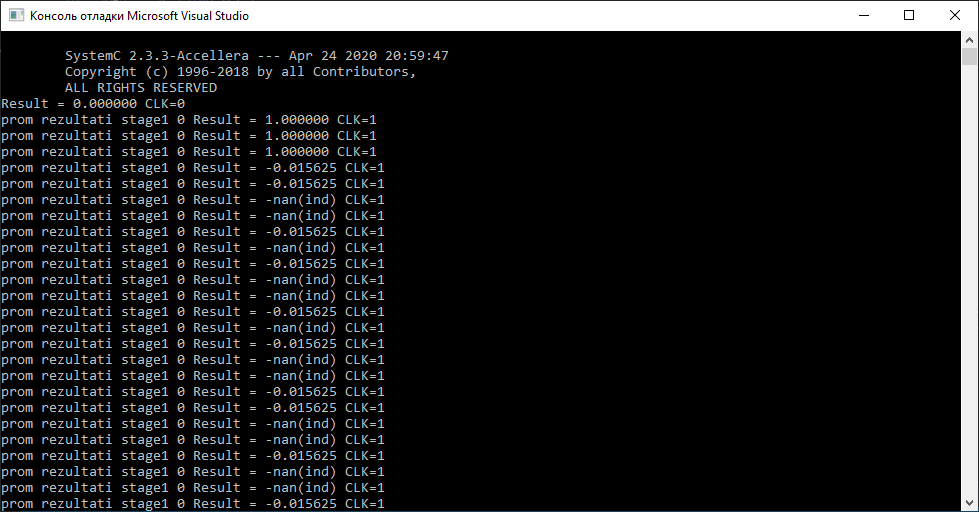
};

#endif TEST\_SYSTEMC\_S2\_H

1. Розробив кінцеву структурну схему отриманої модифікованої програмної моделі:



*Рис.1. Структурна схема отриманої модифікованої програмної моделі*



*Рис.1 Результат роботи*

**Висновок:**

На даній лабораторній роботі навчився здійснювати аналіз програмних моделей комп’ютерних систем, виконаних на мові SystemC.