Міністерство освіти і науки України

Національний університет „Львівська політехніка”

Кафедра ЕОМ



**Звіт**

**з ЛабораторнОЇ роботИ №4**

З дисципліни: “Комп’ютерні системи”

Тема «Аналіз програмної моделі процесу роботи арифметичного конвеєра, ч.2»

Виконав: ст. гр. КІ-33

Фещенко З.-А.С.

Прийняв: асистент каф. ЕОМ

Козак Н.Б.

**Львів 2020**

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

**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.

**Код модифікованої програмної моделі:**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include "systemc.h"

SC\_MODULE(stage\_1) {

sc\_in<bool> clk;

sc\_in<double> in1;

sc\_in<double> in2;

sc\_out<double> power;

sc\_out<double> f1;

void first() {

double x = in1.read();

double y = in2.read();

//power

double resP = pow(x, y);

power.write(resP);

//f1

double resF1 = 0;

double temp = 0;

for (int i = 1; i < y + 1; i++)

{

temp = x \* i;

resF1 = resF1 + temp;

}

f1.write(resF1);

}

SC\_CTOR(stage\_1) {

SC\_METHOD(first);

sensitive << clk.pos();

}

};

SC\_MODULE(stage\_2) {

sc\_in<bool> clk;

sc\_in<double> power;

sc\_in<double> f1;

sc\_out<double> r1;

sc\_out<double> r2;

void setOutputVars() {

r1.write(power.read() || power.read());

r2.write(~(f1.read() || f1.read()));

}

SC\_CTOR(stage\_2) {

SC\_METHOD(setOutputVars);

sensitive << clk.pos();

}

};

SC\_MODULE(stage\_3) {

sc\_in<bool> clk;

sc\_in<double> r1;

sc\_in<double> r2;

sc\_out<double> sum;

sc\_out<double> diff;

void addsub() {

double a, b;

a = r1.read();

b = r2.read();

sum.write(a + b);

diff.write(a - b);

};

SC\_CTOR(stage\_3) {

SC\_METHOD(addsub);

sensitive << clk.pos();

}

};

SC\_MODULE(stage\_4) {

sc\_in<bool> clk;

sc\_in<double> sum;

sc\_in<double> diff;

sc\_out<double> prod;

sc\_out<double> quot;

void multdiv() {

double a, b;

a = sum.read();

b = diff.read();

if (b == 0) {

b = 5.0;

}

prod.write(a \* b);

quot.write(a / b);

}

SC\_CTOR(stage\_4) {

SC\_METHOD(multdiv);

sensitive << clk.pos();

}

};

SC\_MODULE(stage\_5) {

sc\_in<bool> clk;

sc\_in<double> prod;

sc\_in<double> quot;

sc\_out<double> powr;

void power() {

double a;

double b;

double c;

a = prod.read();

b = quot.read();

c = pow(a, b);

powr.write(c);

}

SC\_CTOR(stage\_5) {

SC\_METHOD(power);

sensitive << clk.pos();

}

};

SC\_MODULE(numgen) {

sc\_in<bool> clk;

sc\_out<double> in1;

sc\_out<double> in2;

void generate() {

static double x = 15;

static double y = 15;

x -= 1;

y -= 1;

cout << "x is " << x << endl;

cout << "y is " << y << endl;

in1.write(x);

in2.write(y);

}

SC\_CTOR(numgen) {

SC\_METHOD(generate);

sensitive << clk.pos();

}

};

SC\_MODULE(display) {

sc\_in<bool> clk;

sc\_in<double> sum;

sc\_in<double> diff;

sc\_in<double> prod;

sc\_in<double> quot;

sc\_in<double> powr;

sc\_in<double> power;

sc\_in<double> f1;

sc\_in<double> r1;

sc\_in<double> r2;

void prdouble() {

cout << "power is " << power << endl;

cout << "f1 is " << f1 << endl;

cout << "r1 is " << r1 << endl;

cout << "r2 is " << r2 << endl;

cout << "clk is " << clk << endl;

cout << "sum is " << sum << endl;

cout << "diff is " << diff << endl;

cout << "prod is " << prod << endl;

cout << "quot is " << quot << endl;

cout << "powr is " << powr << endl;

}

SC\_CTOR(display) {

SC\_METHOD(prdouble);

sensitive << clk.pos();

}

};

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

sc\_signal<double> in1;

sc\_signal<double> in2;

sc\_signal<double> power;

sc\_signal<double> f1;

sc\_signal<double> r1;

sc\_signal<double> r2;

sc\_signal<double> sum;

sc\_signal<double> diff;

sc\_signal<double> prod;

sc\_signal<double> quot;

sc\_signal<double> powr;

//Clock

sc\_signal<bool> clk;

clk = true;

numgen N("numgen");

N(clk, in1, in2);

stage\_1 S1("stage1");

S1(clk, in1, in2, power, f1);

stage\_2 S2("stage2");

S2(clk, power, f1, r1, r2);

stage\_3 S3("stage3");

S3(clk, r1, r2, sum, diff);

stage\_4 S4("stage4");

S4(clk, sum, diff, prod, quot);

stage\_5 S5("stage5");

S5(clk, prod, quot, powr);

display D("display");

D(clk, sum, diff, prod, quot, powr, power, f1, r1, r2);

sc\_initialize();

for (double i = 0; i < 250; i+= 25) {

cout << "Time is: " << sc\_time\_stamp() << endl;

clk.write(1);

sc\_start(25, SC\_NS);

clk.write(0);

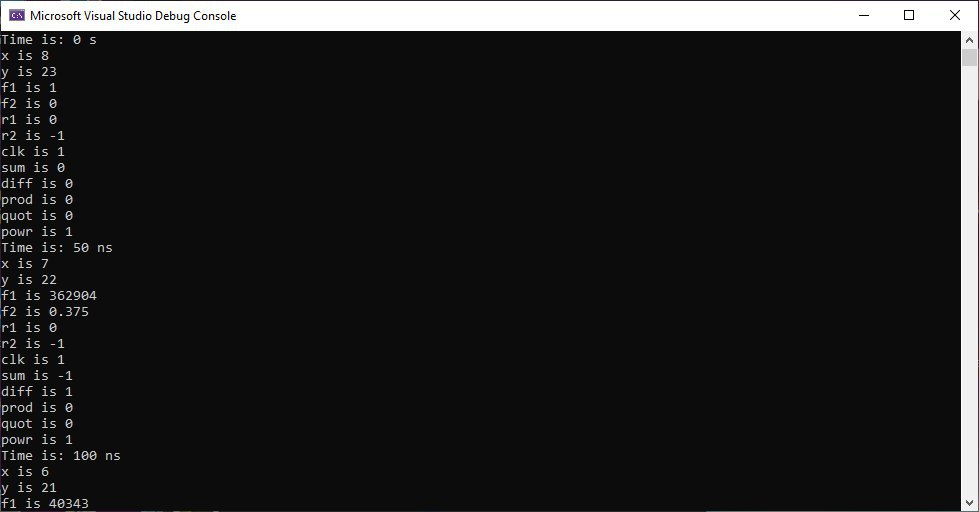
sc\_start(25, SC\_NS);

}

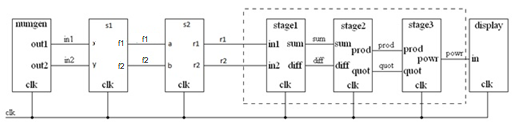
return 0;

}

**Результат виконання програми:**



**Модель арифметичного конвеєра із внесеними змінами:**



**Висновок:** на цій лабораторній роботі янавчився здійснювати аналіз програмних моделей комп’ютерних систем, виконаних на мові System C.

.