Міністерство освіти і науки України

Національний університет “Львівська політехніка”



**Звіт**

**До** **Лабораторної роботи №4**

З дисципліни: «Комп'ютерні системи»

на тему: «**Аналіз програмної моделі процесу роботи арифметичного конвеєра, ч.2»**

**Підготував:** ст.гр. КІ-33

Теслюк Н.А

**Прийняв:**

Козак Н.Б.

Львів 2020

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

**Хід роботи**

**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.

**Лістинг змінених файлів тестової програми**

#include "systemc.h"

SC\_MODULE(stage\_1) {

sc\_in<bool> clk;

sc\_in<double> in1;

sc\_in<double> in2;

sc\_out<double> sum;

sc\_out<double> diff;

void addsub() {

double a, b;

a = in1.read();

b = in2.read();

sum.write(a + b);

diff.write(a - b);

};

SC\_CTOR(stage\_1) {

SC\_METHOD(addsub);

sensitive << clk.pos();

}

};

SC\_MODULE(stage\_2) {

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\_2) {

SC\_METHOD(multdiv);

sensitive << clk.pos();

}

};

SC\_MODULE(stage\_3) {

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\_3) {

SC\_METHOD(power);

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;

void print() {

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(print);

sensitive << clk.pos();

}

};

//////////////////

SC\_MODULE(Module\_S1) {

sc\_in<bool> CLK;

sc\_in<double> X, Y;

sc\_out<double> f1, f2;

void S1\_mod() {

f1.write(Y.read() + !X.read());

if (Y.read() != 0) f2.write((X.read() / Y.read()));

else f2.write(0);

cout << "X in Module S1 is " << X.read() << endl;

cout << "Y in Module S1 is " << Y.read() << endl;

cout << "f1 in Module S1 is " << f1.read() << endl;

cout << "f2 in Module S1 is " << f2.read() << endl << endl;

}

// 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<double> A, B;

sc\_out<double> Out1, Out2;

void S2\_mod() {

Out1.write(A.read());

Out2.write((double)B.read() + (double)(A.read() \* B.read()));

cout << "A in Module S1 is " << A << endl;

cout << "B in Module S1 is " << B << endl;

cout << "Out1 in Module S1 is " << Out1 << endl;

cout << "Out2 in Module S1 is " << Out2 << endl << endl;

}

// constructor for module

SC\_CTOR(Module\_S2) {

SC\_METHOD(S2\_mod);

//sensitive << A << B ;

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

}

};

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

sc\_signal<double> in1;

sc\_signal<double> in2;

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;

sc\_signal<double> Variant, ASCIICode;

sc\_signal<double> f1, f2;

Variant = 1;

ASCIICode = 162; // A + O; A = 83, O = 79

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

moduleS1(clk, Variant, ASCIICode, f1, f2);

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

moduleS2(clk, f1, f2, in1, in2);

stage\_1 S1("stage1");

S1(clk, in1, in2, sum, diff);

stage\_2 S2("stage2");

S2(clk, sum, diff, prod, quot);

stage\_3 S3("stage3");

S3(clk, prod, quot, powr);

display D("display");

D(clk, sum, diff, prod, quot, powr);

sc\_initialize();

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

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

clk.write(1);

sc\_start(25, SC\_NS);

clk.write(0);

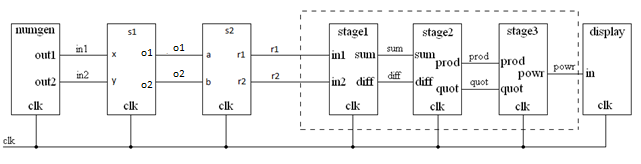
sc\_start(25, SC\_NS);

}

return 0;

}

**Модель арифметичного конвеєра з внесеними змінами**



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