

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-38

Хомин В.Б.

Прийняв: викладач каф. ЕОМ

Козак Н.Б.

**Львів 2022**

**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.

```
power = Pow(X, Y) ;  
f1 = X * 1 + X * 2 + X * 3 + .... + X * Y;  
r1 = a * a;  
r2 = b * b;
```

**Код модифікованої програмної моделі:**

```
#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 = 20;
        static double y = 20;
        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 < 200; 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;
}

```

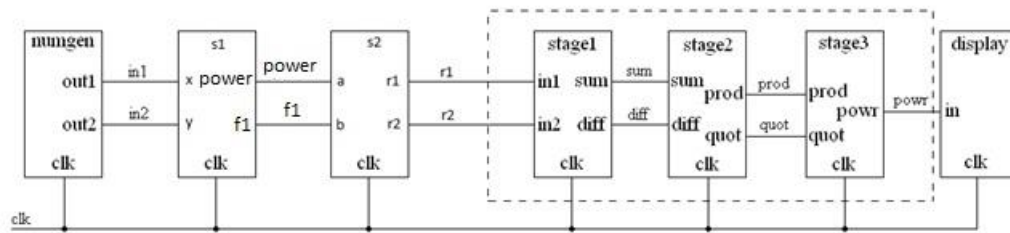
**Результат виконання програми:**

```

Консоль отладки Microsoft Visual Studio
art(SC_ZERO_TIME)
x is 19
y is 19
power is 0
f1 is 0
r1 is 0
r2 is 0
clk is 1
sum is 0
diff is 0
prod is 0
quot is 0
powr is 0
Time is: 0 s
Time is: 50 ns
x is 18
y is 18
power is 1
f1 is 0
r1 is 0
r2 is 0
clk is 1
sum is 0
diff is 0
prod is 0
quot is 0
powr is 1
Time is: 100 ns
x is 17
y is 17
power is 1.97842e+24
f1 is 3610
r1 is 1
r2 is 0
clk is 1
sum is 0
diff is 0
prod is 0
quot is 0
powr is 1
Time is: 150 ns
x is 16
y is 16
power is 3.93464e+22
f1 is 3078
r1 is 3.91414e+48
r2 is 1.30321e+07

```

## Модель арифметичного конвеєра із внесеними змінами:



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