Міністерство освіти і науки України
Національний університет „Львівська політехніка”
Кафедра ЕОМ

Звіт

З ЛАБОРАТОРНОЇ РОБОТИ №3

З дисципліни: “Комп’ютерні системи”

Тема «Аналіз програмної моделі процесу роботи арифметичного конвеєра, ч.1.»

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:
1. Проаналізувати склад програмної моделі арифметичного конвеєра, (програма PIPE), яка виконана на мові System C.
2. Визначити інформаційні потоки у моделі арифметичного конвеєра.
3. Визначити зв'язки керування.
4. Накреслити блоки, з яких складається арифметичний конвеєр згідно поданої моделі.

Хід роботи:

Код програмної моделі:

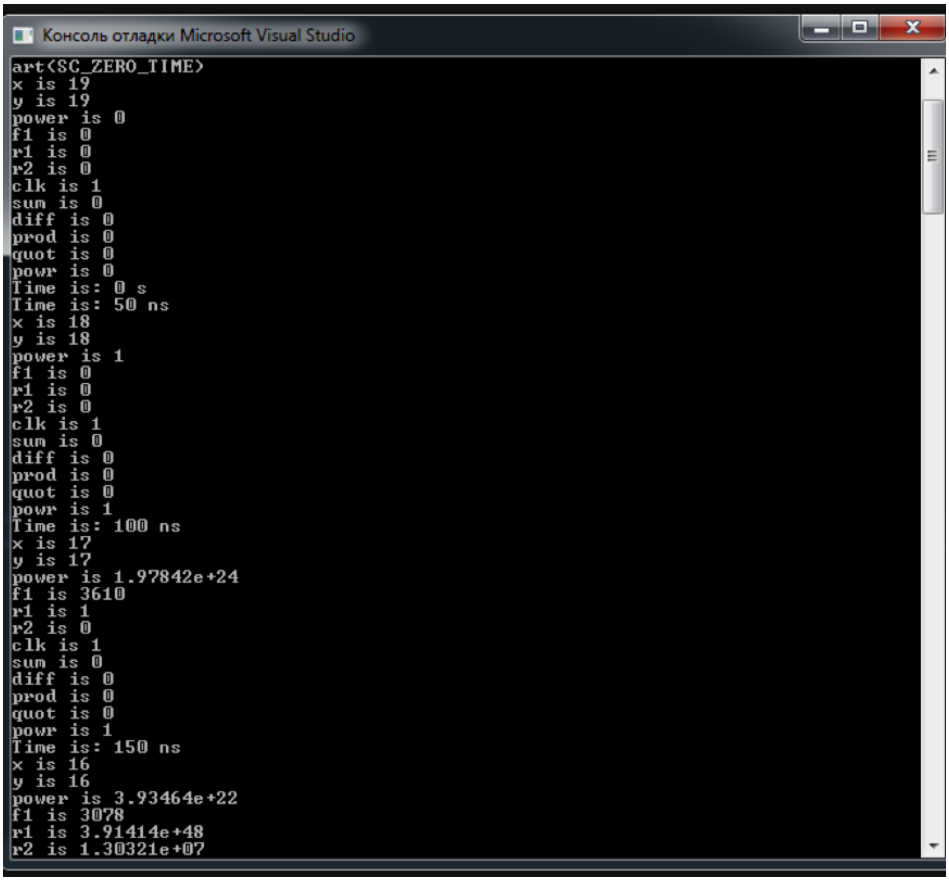
```
#include "systemc.h"

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(numgen) {
    sc_in<bool> clk;
    sc_out<double> out1;
    sc_out<double> out2;

    void generate() {
        static double a = 134.56;
        static double b = 98.24;
        a -= 1.5;
        b -= 2.8;
        cout << "a is " << a << endl;
        cout << "b is " << b << endl;
        out1.write(a);
        out2.write(b);
    }

    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;

    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();
    }
};


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;


    numgen N("numgen");
    N(clk, 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;
        clk.write(1);
        sc_start(25, SC_NS);
        clk.write(0);
```

```
        sc_start(25, SC_NS);
    }

    return 0;
}
```

Результат виконання програми:



1. Перелік і призначення блоків арифметичного конвеєра:

1. Numgen – модуль який генерує на свої 2 виходи числа.
2. Stage1 – модуль обчислює суму та різницю вхідних значень.
3. Stage2 – модуль обчислює добуток та частку вхідних значень.
4. Stage3 – модуль обчислює значення a в степені b (a та b вхідні сигнали).
5. Display – модуль відображає на екрані значення з вхідного порту.

2. Потік даних починається з Numgen далі йде до Stage1, потім з Stage1 йдуть до Stage2, потім Stage2 з йдуть до Stage3, потім дані з Stage3 йдуть до Display.

3. Зв’язки керування виглядають наступним чином:

- Модуль Numgen впливає на Stage1.
- Модуль Stage1 залежить від Numgen, та впливає на Stage2.
- Модуль Stage2 залежить від Stage1, та впливає на Stage3.
- Модуль Stage3 залежить від Stage2, та впливає на Display.
- Модуль Display залежить від Stage3.

4. Блоки, з яких складається арифметичний конвеєр згідно поданої моделі.

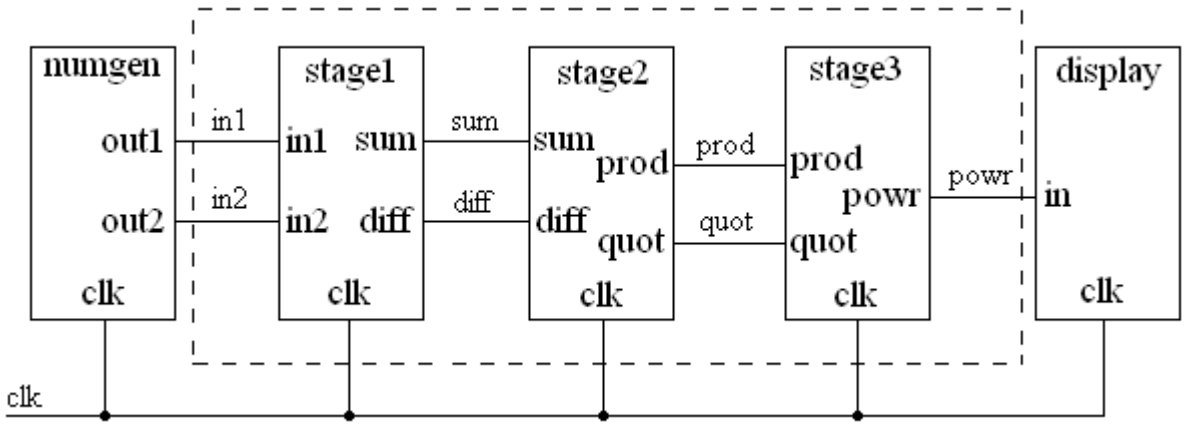


Рис. 1 Блоки арифметичного конвеєра

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