Міністерство освіти і науки

Національний університет “Львівська політехніка”

Кафедра ЕОМ



**Звіт**

з лабораторної роботи №3

з дисципліни: “ Комп’ютерні системи ”

на тему: “ Аналіз програмної моделі процесу роботи арифметичного конвеєра ”

Виконала: ст. гр. КІ-32

Фоменко Г. Ю.

Перевірив: Козак Н.Б.

Львів 2020

**МЕТА РОБОТИ:** 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. 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.

3. Накреслити кінцеву структурну схему отриманої програмної моделі.

4. Навести стисло код та внесені нові зміни.

5. Навести результати тестування та використання програмної моделі.

**Послідовність виконання роботи**

1.Програмна модель конвеєра складається із трьох функціональних блоків, результат виконання операцій попереднього подається на вхід наступному як операнд.

2.Перший функціональний блок приймає на вхід операнди a та b. Далі виконує над ними перетворення і їх результат передає на вхід наступному блоку, який виконує над ними операцій і тд. Всього у моделі можна виділити два потоки: потік даних(a та b для першого блоку, результат виконання операцій попереднього блоку як вхідна дані для наступного) та потік керування(сигнал синхронізації clk).

3.Перелік і призначення блоків, з яких складається конвеєр:

1)блок генерування даних:



2)блок виконання сумування та віднімання:



3)блок виконання множення та ділення:



4)блок піднесення до степеня:



5)блок відображення результату:



4.Структурна схема:



5.Лістинги функціональних блоків:

stage1.cpp

#include "systemc.h"

#include "stage1.h"

//Definition of addsub method

void stage1::addsub()

{

double a;

double b;

a = in1.read();

b = in2.read();

sum.write(a+b);

diff.write(a-b);

} // end of addsub method

stage2.cpp

#include "systemc.h"

#include "stage2.h"

//definition of multdiv method

void stage2::multdiv()

{

double a;

double b;

a = sum.read();

b = diff.read();

if( b == 0 )

b = 5.0;

prod.write(a\*b);

quot.write(a/b);

} // end of multd

stage3.cpp

#include <math.h>

#include "systemc.h"

#include "stage3.h"

//Definition of power method

void stage3::power()

{

double a;

double b;

double c;

a = prod.read();

b = quot.read();

c = (a>0 && b>0)? pow(a, b) : 0.;

powr.write(c);

} // end of power method

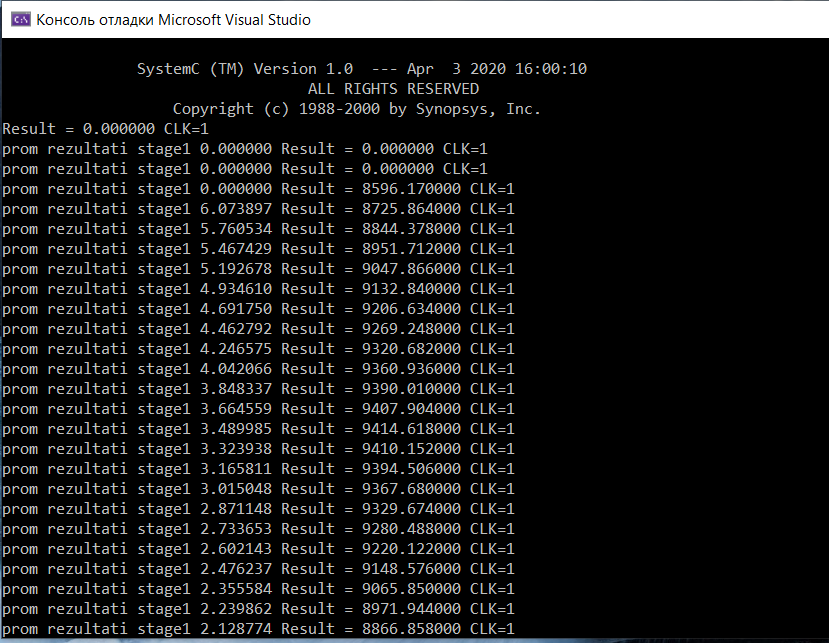


Рис.2 Результат роботи програми

**Висновок:** виконуючи дану лабораторну роботу я навчилась здійснювати аналіз програмних моделей комп’ютерних систем, виконаних на мові System C.