Міністерство освіти і науки України

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, ч.1”

Виконав: ст. гр. КІ-32

Кононенко Р.Ю.

Прийняв:

Козак Н.Б.

Львів – 2020

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

**Завдання**: проаналізувати склад програмної моделі арифметичного конвеєра, (програма PIPE), яка виконана на мові System C.

**Хід роботи:**

1. Проаналізував перелік модулів програмної моделі арифметичного конвеєра:

Програмна модель складається з 5 модулів:

• Numgen – модуль, який генерує два числа на свої виходи.

• Stage1 – модуль, який обчислює суму та різницю вхідних значень.

• Stage2 – модуль, який обчислює добуток та частку вхідних значень.

• Stage3 – модуль, який обчислює значення a в степені b, де a та b - це вхідні сигнали.

• Display – модуль відображає на екрані значення вихідного порту.

1. Проаналізував зв’язки керування, які організовані наступним чином:

* Модуль Numgen впливає на Stage1, але не залежить від інщих модулів, оскільки генерує вхідні значення.
* Модуль Stage1 залежить від Numgen, оскільки очікує на вхідну значення, але впливає на Stage2.
* Модуль Stage2 залежить від Stage1, оскільки очікує на результи додавання та віднімання, але впливає на Stage3.
* Модуль Stage3 залежить від Stage2, оскільки очікує на результати добутку та частки, але впливає на Display.
* Модуль Display залежить від Stage3, оскільки очікує на значення вихідного порту, яке формується на етапі 3.

Крім цього, слід виділити, що інформаційний потік даної програмної моделі арифметичного конвеєра організований аналогічно до зв’язків керування, тобто:

Numgen – Stage1 – Stage2 – Stage3 – Display

1. Проаналізував особливості виконання операцій у модулях програмної моделі:

• Модуль Numgen – записує на вихід в порт out1 – значення числа a, а в порт out2 – значення числа b.

• Модуль Stage1 – зчитує значення з портів in1 та in2, які отримують їх із портів out1 та out2 відповідно, результат додавання записується в sum, а віднімання – в diff.

• Модуль Stage2 – зчитує значення з портів sum та diff, результат множення записує в prod, а ділення – в quot. Якщо значення зчитане з diff рівне 0, йому присвоюється значення 5.0.

• Модуль Stage3 – зчитує значення з портів quot та prod, результат піднесення prod до степеня quot записується в power. Крім цього, якшо якийсь з параметрів не є більшим за 0, то на вихід записується 0.

• Модуль Display – зчитує значення з порту in, який отримує його із вихідного порта модулю Stage3, і виводить його на екран.

1. Розробив структурну схему арифметичного конвеєра, що відповідає заданій програмній моделі:

Numgen

Stage1

Stage2

Stage3

Display

*Рис.1. Структурна схема арифметичного конвеєра за заданою програмною моделлю*

**Додаток**

*Лістинг програмної моделі конвеєра мовою SystemC*

main.cpp

#include "systemc.h"

#include "stage1.h"

#include "stage2.h"

#include "stage3.h"

#include "display.h"

#include "numgen.h"

#define NS \* 1e-9

int sc\_main(int ac, char \*av[])

{

//Signals

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;

numgen N("numgen"); //instance of `numgen' module

N(in1, in2, clk ); //Positional port binding

stage1 S1("stage1"); //instance of `stage1' module

//Named port binding

S1.in1(in1);

S1.in2(in2);

S1.sum(sum);

S1.diff(diff);

S1.clk(clk);

stage2 S2("stage2"); //instance of `stage2' module

S2(sum, diff, prod, quot, clk ); //Positional port binding

stage3 S3("stage3"); //instance of `stage3' module

S3( prod, quot, powr, clk); //Positional port binding

display D("display"); //instance of `display' module

D(powr,clk,diff,sum,prod,quot); //Positional port binding

sc\_initialize(); //Initialize simulation

for(int i = 0; i < 50; i++){

clk.write(1);

sc\_cycle( 10 NS );

clk.write(0);

sc\_cycle( 10 NS );

}

return 0;

}

**Висновок:**

На даній лабораторній роботі навчився здійснювати аналіз програмних моделей комп’ютерних систем, виконаних на мові System C.