**Міністерство освіти І науки України**

**національний університет “Львівська політехніка”**

****

**Кафедра ЕОМ**

лабораторна робота № 3

“ Аналіз програмної моделі процесу роботи

арифметичного конвеєра, ч.1”

з дисципліни

**"** **Комп’ютерні системи "**

Студента групи КІ-32

Кузишина Назара

Прийняв: Козак Н. Б.

**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. Визначити інформаційні потоки у моделі арифметичного конвеєра.
3. Визначити зв’язки керування.
4. Накреслити блоки, з яких складається арифметичний конвеєр згідно поданої моделі.
5. Оформити звіт.

**Хід роботи:**

Завданням даної лабораторної роботи є аналіз програмної моделі арифметичного конвеєра. В складі даної моделі є 5 модулів: numgen, state1, state2, state3, display. Модуль numgen відповідає за генерацію двох вхідних значень, модуль stage1 відповідає за додавання та віднімання вхідних значень, модуль stage2 за множення та ділення, stage3 за піднесення одного значення до степеня іншого, а модуль display виводить значення вихідного сигналу та значення різниці з модуля stage1.

Інформаційний потік у моделі арифметичного конвеєра виглядає наступним чином:

Numgen -------> stage1 -------> stage2 ------>stage3 ------> display

Зв’язки керування виглядають наступним чином:

• Модуль numgen незалежний, впливає на stage1

• Модуль stage1 залежить від Numgen оскільки очікує на вхідне значення, але впливає на Stage2.

• Модуль Stage2 залежить від Stage1, оскільки очікує на результати 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.

Структурна схема нашої програмної моделі конвеєра буде мати наступний вигляд:

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, якщо 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.

**текст програми**:

Лістинг програмної моделі конвеєра мовою SystemC

main.cp

#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;

}