Міністерство освіти і науки України

Національний університет "Львівська політехніка"

Кафедра ЕОМ

****

**Звіт**

з лабораторної роботи № 2

з дисципліни: “ Комп’ютерні системи”

на тему: “ Ознайомлення з основними конструкціями мови моделювання System C ”

Варіант 6

Виконав: студент .гр. КІ-32

Данилів М.М.

Прийняв: асистент каф. ЕОМ

Козак Н.Б.

Львів 2020

**Мета:** Ознайомлення з основними конструкціями мови моделювання System C.

**ПОРЯДОК ВИКОНАННЯ РОБОТИ**

***Вхідні дані:***

Х = 6

Y = 75 (‘K’) + 86 (‘V’) = 161

Реалізувати модулі S1 та S2, разом з логікою їх функціонування згідно варіанту, провести послідовне з’єднання S1 та S2 ініціалізувати необхідні порти на S1. На вхід S2 подати вихідні порти модуля S1. До кожного модуля заводиться зовнішній вхідний для всіх сигнал синхронізації CLK. Результати подати на модуль Display для відображення.

**Завдання по реалізації модуля «S1».** Номер варіанту (1 - 5) визначається як остача від ділення на 5 номера в списку журналу:

1) 2 вихідних порта o1, o2

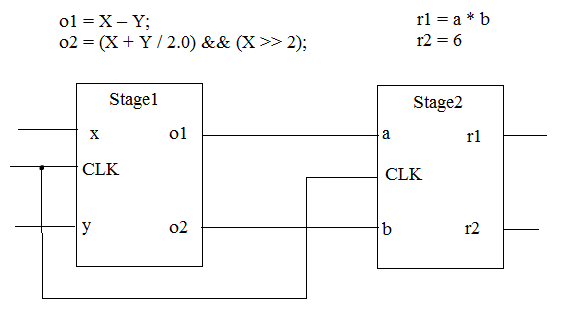
o1 = X – Y; // 6-161 = -155

o2 = (X + Y / 2.0) && (X >> 2); // (6+161/2.0) && (X>>2) = 0

2 вихідні порти з модуля S1 довільно під’єднуються до вхідних портів модуля S2: **a** та **b.**

Варіанти для реалізації модуля S2:

|  |  |
| --- | --- |
| Варіант | Завдання |
| 6, 16, 26 | Обчислити r1 = a \* b , r2 = Номер варіанту; |



**Stage1.h**

#ifndef STAGE1\_H

#define STAGE1\_H

struct stage1 : sc\_module {

sc\_in<int> x; //input 1

sc\_in<int> y; //input 2

sc\_out<int> o1; //output 1

sc\_out<int> o2; //output 2

sc\_in<bool> clk; //clock

void sub();

//Counstructor

SC\_CTOR( stage1 ) {

SC\_METHOD( sub );

sensitive<<x<<y;

sensitive\_pos << clk; }

};

#endif

**Stage1.cpp**

#include "systemc.h"

#include "stage1.h"

void stage1::sub()

{

int a;

int b;

a = x.read();

b = y.read();

o1.write(a-b);

o2.write((int)(a+b/2.0)&(a>>2));

}

**Stage2.h**

#ifndef STAGE2\_H

#define STAGE2\_H

struct stage2 : sc\_module {

sc\_in<int> a; //input port 1

sc\_in<int> b; //input port 2

sc\_out<int> r1; //output portik 1

sc\_out<int> r2; //output portik 2

sc\_in<bool> clk; //clock

void mult();

//Constructor

SC\_CTOR( stage2 ) {

SC\_METHOD( mult );

sensitive<<a<<b;

sensitive\_pos << clk; }

};

#endif

**Stage2.cpp**

#include "systemc.h"

#include "stage2.h"

void stage2::mult()

{

int \_a;

int \_b;

\_a = a.read();

\_b = b.read();

r1.write(\_a\*\_b);

r2.write(6)

}

**main.cpp**

#include "systemc.h"

#include "stage1.h"

#include "stage2.h"

#define NS \* 1e-9

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

{

//Signals

sc\_signal<int>x;

sc\_signal<int>y;

sc\_signal<int>o1;

sc\_signal<int>o2;

sc\_signal<int> r1;

sc\_signal<int> r2;

//Clock

sc\_clock clk ("clk",20 );

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

//Named port binding

S1.x(x=6);

S1.y(y=161);

S1.o1(o1);

S1.o2(o2);

S1.clk(clk);

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

S2(o1, o2, r1, r2, clk );

sc\_initialize(); //Initialize simulation

sc\_start(1000);

std::cout << "x: " << x.read() << std::endl;

std::cout << "y: " << y.read() << std::endl;

std::cout << "r1: " << r1.read() << std::endl;

std::cout << "r2: " << r2.read() << std::endl;

return 0;

}

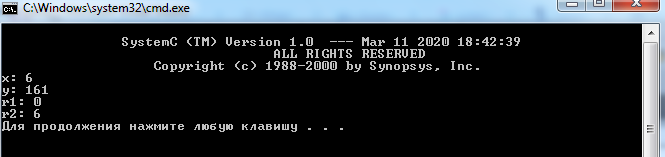


Рис.1 Результат виконання програми

**Висновок:** На цій лабораторній роботі я ознайомився з основними конструкціями мови моделювання System C.