**Міністерство освіти і науки України**

**Національний університет "Львівська політехніка"**

**Кафедра ЕОМ**



**Звіт з лабораторної роботи №4**

**з дисципліни “** **Комп’ютерні системи ”**

**на тему: ” Аналіз програмної моделі процесу роботи арифметичного конвеєра, ч.2”**

**Виконав: студент .гр. КІ-33**

**Кіндій В. А.**

**Прийняв: викладач**

**Козак Н. Б.**

**Львів 2020 р.**

**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.

**Хід роботи:**

1. Аналізую склад програмної моделі арифметичного конвеєра, (програма PIPE), яка виконана на мові System C.
2. Здійснюю модернізацію функцій або параметрів арифметичного, шляхом під’єднання розроблених модулів S1 та S2.
3. Креслю кінцеву структурну схему отриманої програмної моделі.

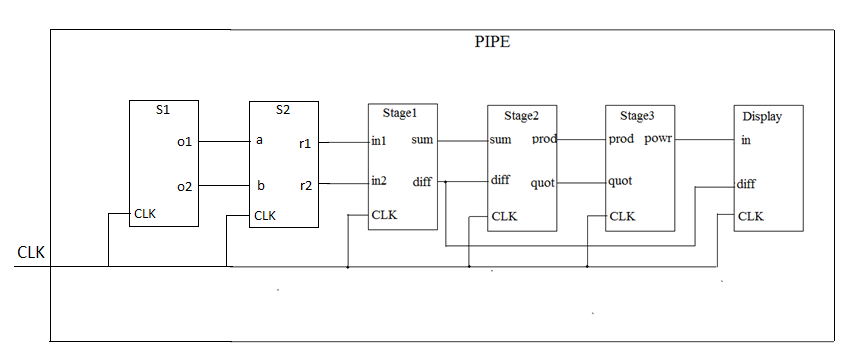


Рис 1. Схема арифметичного блоку згідно поданої моделі

4. Наводжу стисло код та внесені нові зміни.

|  |  |  |
| --- | --- | --- |
| S1  const int X = 5;  const int Y = 161;  SC\_MODULE(S1) {  sc\_out<double> o1, o2;  sc\_in<bool> clk;  void mainFunc() {  o1.write(X - Y);  o2.write((X + Y / 2) && (X >> 2));  cout << "clk read: " << clk.read() << endl;  }  SC\_CTOR(S1) {  SC\_METHOD(mainFunc);  sensitive\_pos << clk;  }  }; | S2  SC\_MODULE(S2) {  sc\_out<double> r1, r2;  sc\_in<double> a, b;  sc\_in<bool> clk;  void mainFunc() {  double temp1 = a.read();  double temp2 = b.read();  r1.write(temp1 + temp2);  r2.write((double)variant);  }  SC\_CTOR(S2) {  SC\_METHOD(mainFunc);  sensitive\_pos << clk;  }  }; | Main  //Clock  sc\_clock clock("Mod", 2.0, 0.5, 0.2, false);    // instance of S1 module  S1 myS1("S1");  myS1.clk(clock);  myS1.o1(s1\_out1);  myS1.o2(s1\_out2);  // instance of S2 module  S2 myS2("S2");  myS2.clk(clock);  myS2.a(s1\_out1);  myS2.b(s1\_out2);  myS2.r1(s2\_out1);  myS2.r2(s2\_out2);    //instance of `stage1' module  stage1 S1("stage1");  //Named port binding  S1.in1(s2\_out1);  S1.in2(s2\_out2);  S1.sum(sum);  S1.diff(diff);  S1.clk(clock);  //instance of `stage2' module  stage2 S2("stage2");  S2(sum, diff, prod, quot, clock);  //instance of `stage3' module  stage3 S3("stage3");  S3( prod, quot, powr, clock); |

5. Наводжу результати тестування та використання програмної моделі.

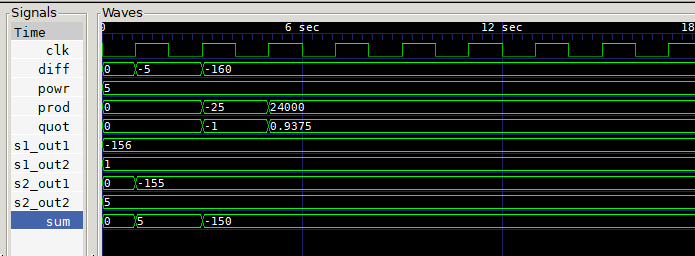


Рис 2. WaveForm виконання арифметичного блоку.

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