

## Лабораторна робота 2

### Проектування АЛП структурним стилем.

#### 1 Мета лабораторної роботи:

оволодіти знаннями і навичками по проектуванню арифметико-логічних пристроїв (АЛП) для сучасних комп'ютерів. Навчитись вводити логічні схеми АЛП на VHDL структурним стилем.

#### Теоретичні відомості

В лабораторній роботі вивчається проектування обчислювальних засобів за допомогою VHDL структурним стилем. Структурний опис системи задає компоненти проекту і їх міжз'єднання. Алгоритм поведінки такого проекту не заданий явним чином, але він виражений в структурному вигляді. Нижче приведений приклад опису двохрозрядного суматора, структура якого показана на рис.

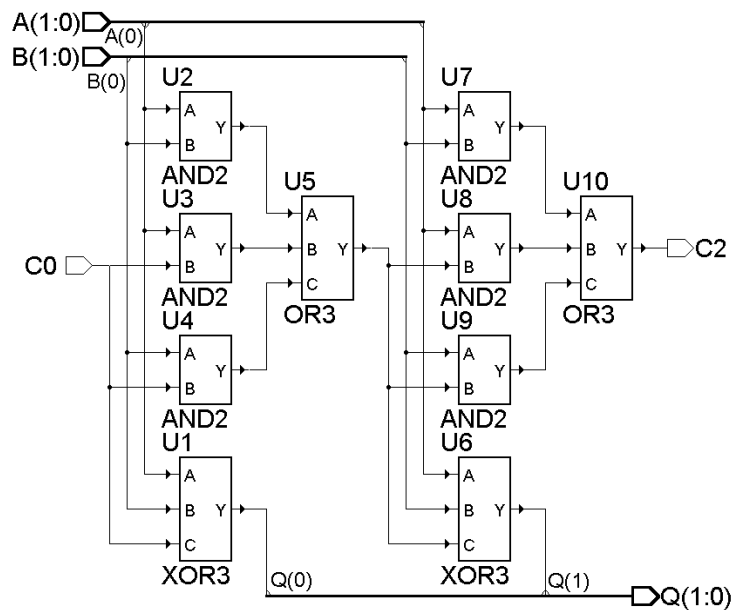
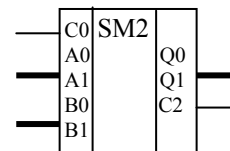


Рис. Структура двохрозрядного суматора

```

library IEEE;
use IEEE.STD_LOGIC_1164.all;
library Unisim ;
use Unisim.all ;
entity SM2 is
port (C0:in STD_LOGIC; -- сигнал переносу
      A:in STD_LOGIC_vector(1 downto 0); --вхідне дане
      B:in STD_LOGIC_vector(1 downto 0); --вхідне дане
      C2: out STD_LOGIC; -- вихід переносу
      Q:out STD_LOGIC_vector(1 downto 0));--вихідне дане
end SM2;
architecture STRUCT of SM2 is
component AND2 port( O : out std_ulogic; I0 : in std_ulogic; I1 : in std_ulogic);
end component;
component OR3 port( O : out std_ulogic; I0, I1, I2: in std_ulogic);
end component;
end architecture;

```



```

signal C1: bit; --біт переносу з розряду 0 в розряд1
signal t1,t2,t3,t4,t5,t6: bit; --проміжні біти
begin
  U1: entity XOR3 port map(Q(0),A(0),B(0),C0);
  U2:AND2 port map(t1,A(0),B(0));
  U3:AND2 port map(t2,A(0),C0);
  U4:AND2 port map(t3,B(0),C0);
  U5:OR3 port map(C1,t1,t2,t3);
  U6: entity XOR3 port map(Q(1),A(1),B(1),C1);
  U7:AND2 port map(t4,A(1),B(1));
  U8:AND2 port map(t5,A(1),C1);
  U9:AND2 port map(t6,B(1),C1,);
  U10:OR3 port map(C2,t4,t5,t6);
end STRUCT;

```

В декларативній частині архітектури об'явлені компоненти, які використовуються в структурі. Ці компоненти можуть бути бібліотечними компонентами або описані в іншому місці і навіть на початку проектування незкомпільовані. В даному випадку вони описані в бібліотеці UNISIM (UNified SIMulation), що поставляється з усіма САПР. Можна компонент не об'являти в декларативній частині, але тоді він повинен бути вже зкомпільованим і при його вставці додається ключове слово **entity**, як це зроблено у прикладі вставки компонента U1. При цьому часто необхідно компонент вказувати як пару назв об'єкт – архітектура, вказуючи конкретну реалізацію компонента, наприклад,

```

entity XOR3(XOR3_V) port map(Q(0),A(0),B(0),C0);

```

Оператори, що відмічені обов'язковими мітками U1,...,U10 є операторами вставки компонента. Так як це паралельні оператори, то їхній порядок в програмі довільний. В фразі зв'язування портів і сигналів **port map** виконується підключення сигналів і портів компонентів у відповідності з графом структури. Тут виконано асоціативне зв'язування, коли сигнал відповідає порту в тому самому порядку, який заданий в об'яві компонента. Але частіше використовується поіменоване зв'язування, коли сигнал відноситься до порту явним чином і порядок зв'язування портів довільний, як наприклад:

```

U2:AND2 port map(A=>A(0),B=>B(0), Y=>t1);

```

## 2. Завдання для лабораторної роботи:

- розробити функціональну схему АЛП, що виконує задані функції;
- змодельовати роботу АЛП.

**Результати виконання** оформлюються у вигляді звіту (протоколу). Звіт повинен вміщувати:

- опис і рисунок заданого варіанта АЛП,
- хід проектування і схему АЛП,
- графіки сигналів, знятих при іспитах АЛП,
- висновки.

Варіанти завдань брати з лабораторної роботи 1

## 3. Виконання роботи

За основу беруть результати лабораторної роботи 1, а саме одержану функціональну схему АЛП. На основі цієї схеми складають VHDL-програму, використовуючи структурний стиль опису і бібліотеку елементів Unisim, вбудовану в САПР ActiveHDL. В табл. 1 представлена частина цієї бібліотеки

Табл.1. Бібліотека символів логічних елементів

Символ	Функція	Символ	Функція
AND2	2-входова &	OR2	2-входова ∨
AND2B1	2-входова & з 1 інверсним входом	OR2B1	2-входова ∨ з 1 інверсним входом
AND2B2	2-входова & з 2 інверсними входами	OR2B2	2-входова ∨ з 2 інверс. входами
AND3	3-входова &	OR3	3-входова ∨
AND3B1	3-входова & з 1 інверсним входом	OR3B1	3-входова ∨ з 1 інверсним входом
AND3B2	3-входова & з 2 інверсними входами	OR3B2	3-входова ∨ з 2 інверс. входами
AND4	4-входова &	OR4	4-входова ∨
NAND2	2-входова &-not	XOR2	2-входова ⊕
NAND2B1	2-входова &-not з 1 інверс. входом	XOR3	3-входова ⊕
NAND2B2	2-входова &-not з 2 інверс. входами	XNOR2	2-входова ⊕-not
NAND3	3-входова &-not	XNOR3	3-входова ⊕-not
NAND3B1	3-входова &-not з 1 інверсним входом	BUF	Буфер - повторювач
NAND3B2	3-входова &-not з 2 інверс. входами	BUFT	Тристабільний буфер
NAND4	4-входова &-not	INV	Буфер - інвертор

Типовий опис бібліотечного компонента виглядає наступним чином



```
entity AND2B1 is
  port( O : out std_ulogic;
        I0 : in std_ulogic;
        I1 : in std_ulogic );
  architecture AND2B1_V of AND2B1 is
  begin
    O <= (not I0) and I1 after 0 ps;
  end AND2B1_V;
```


Слід відмітити, що в списку зв'язування портів вихід компонента стоїть першим.

Запускають САПР ActiveHDL, відкривають власний робочий простір Workspace, вводять назву проекту, наприклад, ALU\_lab2.

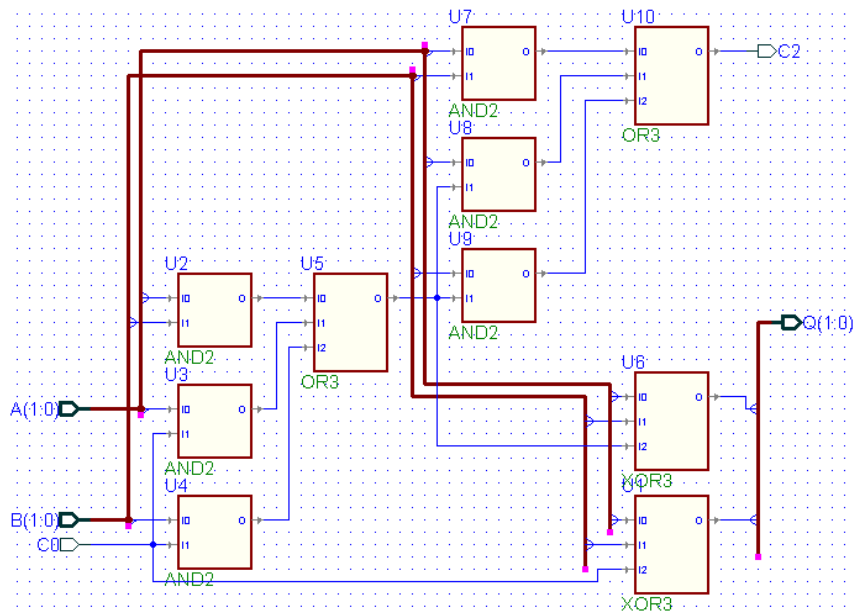
Натиснувши праву кнопку миші на строці Add\_New\_File вибирають опцію New -> VHDL\_Source.



В запропонованому діалозі вводять назву проекту і назви портів Entity.





Потім у вікні редактора вводять VHDL-опис проекту АЛП у відповідності з функціональною схемою, одержаною в попередній лабораторній роботі. Для доступу до бібліотеки компонентів натискають піктограму  і вибирають у вікні Libraries бібліотеку . У вікні каталогу бібліотеки знаходять і відмічають необхідні компоненти. При необхідності посилання на інтерфейс компонента його знаходять в текстовому описі бібліотеки, вибравши правою кнопкою миші опцію View source – text, або зкопіювавши опис Entity в «карман» опцією Copy Declaration, щоб потім його вставити в текст програми опцією Paste.

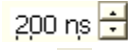


Після закінчення вводу програми її трансльють, натиснувши піктограму . Якщо в програмі є помилки, про що буде повідомлення у вікні консолі, ці помилки необхідно усунути.

Якщо в декларативній частині задекларовано описи всіх компонентів, то можна побудувати графічне представлення моделі АЛП, застосовуючи засіб Code2Graphics, що вибирається в меню Tools. Для вищеприведеного прикладу (при додаванні до нього опису компонента XOR3 ) одержимо наступний рисунок, представлений графічним файлом з розширенням .BDE.



Після успішної компіляції одержану модель АЛП моделюють після виклику вікна графіків сигналів . В цьому вікні встановлюють імена сигналів, які необхідно відобразити, в тому числі всі вхідні сигнали викликавши вікно . Можна встановити сигнали, перетаскуючи їх з вікна Design Browser – Structure.

Всі вхідні сигнали необхідно підключити до стимуляторів. Для цього викликають вікно стимуляторів . Серед стимуляторів можна вибрати генератор синхросигналів з заданим періодом , лічильник , натискання вибраної кнопки клавіатури , або задану послідовність зміни (графік) сигналу (Formula).

Моделюють проект, задавши інтервал часу моделювання  200 ns і натиснувши кнопку . Симулятор встановлюють в початковий стан, натиснувши .

Одержані графіки сигналів заносять в протокол лабораторної роботи. При таких самих стимулюючих сигналах, які застосовувались в попередній лабораторній роботі, повинні бути одержані такі самі графіки сигналів.