МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

КАФЕДРА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

Лабороторна робота 5

з дисципліни «Технології програмування на ПЛІС(FPGA)»

Виконав:

студент гр. ІП-11

Прищепа Владислав

Номер залікової книжки – 1125

Перевірив:

Гайдай А.

Київ 2024

**Виконання роботи**

1125 **=** 10001100101

0101 - варіант по h2 h1 h4 h3

**Мета роботи**

Реалізувати чотирьохрозрядний демультиплексор 1-в-5 (кількість вихідних сигналів – 5, розрядність – 4 біт) на мові Verilog.

Створити тестовий модуль (testbench) для перевірки його роботи.

Виконати моделювання в середовищі ModelSim і проаналізувати результати.

**Теоретичні відомості**

Демультиплексор — це комбінаційний логічний пристрій, який використовується для управління передачею даних від одного вхідного каналу на кілька вихідних. Демультиплексор має один інформаційний вхід, n адресних входів і 2^n виходів.

Для чотирьохрозрядного демультиплексора 1-в-5 структура включає наступні компоненти:

- 1 інформаційний вхід (4 біти)

- 3 адресні входи (A0, A1, A2)

- 5 вихідних сигналів (по 4 біти кожен)

Логічні рівняння для кожного вихідного сигналу:

- Q0 = D \* A2' \* A1' \* A0'

- Q1 = D \* A2' \* A1' \* A0

- Q2 = D \* A2' \* A1 \* A0'

- Q3 = D \* A2' \* A1 \* A0

- Q4 = D \* A2 \* A1' \* A0'

Де:

- D — вхідний інформаційний сигнал (4 біти)

- A0, A1, A2 — адресні сигнали

- Q0, Q1, Q2, Q3, Q4 — вихідні сигнали (по 4 біти кожен)

**Реалізація чотирьохрозрядного демультиплексора 1-в-5 мовою Verilog**

module lr5\_Prishchepa(

input [3:0] D, // 4-бітний вхідний сигнал

input A0, // адресний вхід 0

input A1, // адресний вхід 1

input A2, // адресний вхід 2

output [3:0] Q0, // 4-бітний вихідний сигнал 0

output [3:0] Q1, // 4-бітний вихідний сигнал 1

output [3:0] Q2, // 4-бітний вихідний сигнал 2

output [3:0] Q3, // 4-бітний вихідний сигнал 3

output [3:0] Q4 // 4-бітний вихідний сигнал 4

);

assign Q0 = (~A2 & ~A1 & ~A0) ? D : 4'b0000;

assign Q1 = (~A2 & ~A1 & A0) ? D : 4'b0000;

assign Q2 = (~A2 & A1 & ~A0) ? D : 4'b0000;

assign Q3 = (~A2 & A1 & A0) ? D : 4'b0000;

assign Q4 = ( A2 & ~A1 & ~A0) ? D : 4'b0000;

endmodule

**Тестовий модуль для перевірки роботи (Testbench)**

`timescale 1ns / 1ps

module testbench\_lr5;

reg [3:0] D;

reg A0;

reg A1;

reg A2;

wire [3:0] Q0;

wire [3:0] Q1;

wire [3:0] Q2;

wire [3:0] Q3;

wire [3:0] Q4;

demux1to5\_4bit uut (

.D(D),

.A0(A0),

.A1(A1),

.A2(A2),

.Q0(Q0),

.Q1(Q1),

.Q2(Q2),

.Q3(Q3),

.Q4(Q4)

);

initial begin

$monitor("Time = %0d A2 = %b A1 = %b A0 = %b D = %b Q0 = %b Q1 = %b Q2 = %b Q3 = %b Q4 = %b",

$time, A2, A1, A0, D, Q0, Q1, Q2, Q3, Q4);

// Випадок 1

A2 = 0; A1 = 0; A0 = 0; D = 4'b1010; #10;

// Випадок 2

A2 = 0; A1 = 0; A0 = 1; D = 4'b1100; #10;

// Випадок 3

A2 = 0; A1 = 1; A0 = 0; D = 4'b1110; #10;

// Випадок 4

A2 = 0; A1 = 1; A0 = 1; D = 4'b0001; #10;

// Випадок 5

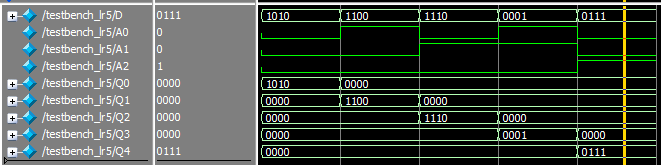
A2 = 1; A1 = 0; A0 = 0; D = 4'b0111; #10;

$finish;

end

endmodule

**Результати симуляції та висновки**



Ми реалізували чотирьохрозрядний демультиплексор 1-в-5 мовою Verilog і створили тестбенч для перевірки його роботи в середовищі ModelSim. Демультиплексор приймає 4-бітний вхідний сигнал D та адресні входи A0, A1 і A2, спрямовуючи вхідний сигнал на один із п'яти 4-бітних виходів залежно від значень адресних входів.

Результати симуляції підтверджують коректну роботу чотирьохрозрядного демультиплексора 1-в-5:

- Демультиплексор правильно спрямовує 4-бітний вхідний сигнал на відповідний вихід залежно від значень адресних входів A0, A1 і A2.

- Всі інші виходи залишаються нульовими, що відповідає логічним рівнянням і таблиці істинності, розробленим на основі теоретичних відомостей.

Таким чином, можна зробити висновок, що демультиплексор працює коректно, і наша реалізація та тестування пройшли успішно.