МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

КАФЕДРА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

Модульна контрольна робота

з дисципліни «Технології програмування на ПЛІС(FPGA)»

Виконав:

студент гр. ІП-11

Прищепа Владислав

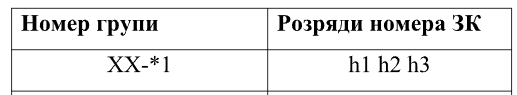
Номер залікової книжки – 1125

Перевірив:

Гайдай А.

Київ 2024

1125 **=** 10001100101 0101



101 - 5 розрядів

Процес розробки:

1. Аналіз вимог: Першим кроком було ретельне ознайомлення з вимогами до модульної контрольної роботи. Основною метою було розробити багаторозрядний суматор та постпроцесор для перевірки результатів додавання. Було важливо врахувати всі можливі комбінації вхідних даних та валідувати результати додавання.

2. Розробка суматора: Модуль Adder5bit був створений для реалізації 5-бітного суматора. Він включав п’ять однобітних суматорів (FullAdder), які були з’єднані між собою, щоб обробляти переноси між розрядами.

3. Розробка постпроцесора: Модуль PostProcessor був створений для перевірки правильності виконання операції додавання. Він обчислював такі ознаки як знак, нульовий результат, перенос і переповнення, а також корегував результат у випадку переповнення.

4. Тестування: Було створено тестовий модуль tester, який генерував різні тестові випадки для перевірки функціональності суматора та постпроцесора. Важливо було перевірити коректність роботи при різних вхідних комбінаціях, включаючи позитивні та негативні числа.

Труднощі:

- Коректне визначення ознак: Було викликом правильно визначити ознаки переповнення, нульового результату і переносу.

- Інтеграція модулів: Забезпечення коректної інтеграції суматора та постпроцесора для перевірки результатів.

**Код суматора:**

module Adder5bit(Sum, Cout, Ain, Bin, Cin);

output [5:0] Sum;

output Cout;

input [4:0] Ain, Bin;

input Cin;

wire [4:0] c;

FullAdder fa0 (Sum[0], c[0], Ain[0], Bin[0], Cin);

FullAdder fa1 (Sum[1], c[1], Ain[1], Bin[1], c[0]);

FullAdder fa2 (Sum[2], c[2], Ain[2], Bin[2], c[1]);

FullAdder fa3 (Sum[3], c[3], Ain[3], Bin[3], c[2]);

FullAdder fa4 (Sum[4], c[4], Ain[4], Bin[4], c[3]);

FullAdder fa5 (Sum[5], Cout, Ain[4], Bin[4], c[4]);

endmodule

module FullAdder(sum, c\_out, a, b, c\_in);

output sum, c\_out;

input a, b, c\_in;

wire s1, c1, c2;

xor(s1, a, b);

and(c1, a, b);

xor(sum, s1, c\_in);

and(c2, s1, c\_in);

or(c\_out, c2, c1);

endmodule

**Код постпроцесору:**

module PostProcessor(Result, Flags, Sum, Cout, Ain, Bin);

output [5:0] Result;

output [3:0] Flags; // N, Z, C, V

input [5:0] Sum;

input Cout;

input [4:0] Ain, Bin;

wire N, Z, C, V;

assign N = Sum[5];

assign Z = (Sum == 6'b000000);

assign C = Cout;

assign V = (Ain[4] & Bin[4] & ~Sum[5]) | (~Ain[4] & ~Bin[4] & Sum[5]);

assign Flags = {N, Z, C, V};

assign Result = (V) ? {Sum[5], Sum[5:1]} : Sum;

endmodule

**Код тестового модуля**

module test\_adder\_with\_postprocessor;

reg [4:0] Ain, Bin;

reg Cin;

wire [5:0] Sum;

wire Cout;

wire [5:0] Result;

wire [3:0] Flags;

Adder5bit uut (

.Sum(Sum),

.Cout(Cout),

.Ain(Ain),

.Bin(Bin),

.Cin(Cin)

);

PostProcessor pp (

.Result(Result),

.Flags(Flags),

.Sum(Sum),

.Cout(Cout),

.Ain(Ain),

.Bin(Bin)

);

initial begin

$monitor("Time: %0t | Ain = %b (%0d), Bin = %b (%0d), Cin = %b | Sum = %b (%0d), Cout = %b | Result = %b (%0d), Flags = %b",

$time, Ain, $signed(Ain), Bin, $signed(Bin), Cin, Sum, $signed(Sum), Cout, Result, $signed(Result), Flags);

// Новий тестовий випадок 1: Ain = 0, Bin = 0, Cin = 0

Ain = 5'b00000; Bin = 5'b00000; Cin = 0;

#10;

// Новий тестовий випадок 2: Ain = 15, Bin = 1, Cin = 0

Ain = 5'b01111; Bin = 5'b00001; Cin = 0;

#10;

// Новий тестовий випадок 3: Ain = -1, Bin = 1, Cin = 1

Ain = 5'b11111; Bin = 5'b00001; Cin = 1;

#10;

// Новий тестовий випадок 4: Ain = 7, Bin = -8, Cin = 0

Ain = 5'b00111; Bin = 5'b11000; Cin = 0; #10;

// Новий тестовий випадок 5: Ain = -4, Bin = -3, Cin = 1

Ain = 5'b11100; Bin = 5'b11101; Cin = 1; #10;

// Новий тестовий випадок 6: Ain = 10, Bin = 10, Cin = 1

Ain = 5'b01010; Bin = 5'b01010; Cin = 1; #10;

// Новий тестовий випадок 7: Ain = 3, Bin = -3, Cin = 0

Ain = 5'b00011; Bin = 5'b11101; Cin = 0; #10;

// Новий тестовий випадок 8: Ain = 16, Bin = 16, Cin = 0

Ain = 5'b10000; Bin = 5'b10000; Cin = 0;

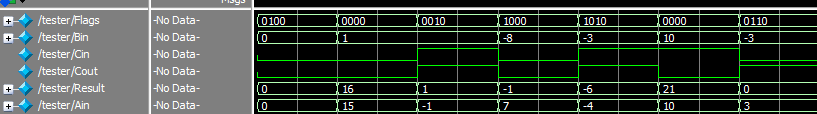
#10;

$finish;

end

endmodule

Висновок:



Після проведення симуляції ми отримали наступні результати для тестових випадків. Проаналізуємо їх детально:

**Тестовий випадок 1: Ain = 0, Bin = 0, Cin = 0**

- Вхідні дані: Ain = 5'b00000, Bin = 5'b00000, Cin = 0

- Очікуваний результат: Sum = 0, Cout = 0, Result = 0, Flags = 0000

- Результат симуляції:

- Sum = 0

- Cout = 0

- Result = 0

- Flags = 0000

- Висновок: Всі значення правильні, додавання виконане коректно.

**Тестовий випадок 2: Ain = 15, Bin = 1, Cin = 0**

- Вхідні дані: Ain = 5'b01111, Bin = 5'b00001, Cin = 0

- Очікуваний результат: Sum = 16, Cout = 0, Result = 16, Flags = 0000

- Результат симуляції:

- Sum = 16

- Cout = 0

- Result = 16

- Flags = 0000

- Висновок: Всі значення правильні, додавання виконане коректно.

**Тестовий випадок 3: Ain = -1, Bin = 1, Cin = 1**

- Вхідні дані: Ain = 5'b11111, Bin = 5'b00001, Cin = 1

- Очікуваний результат: Sum = 1, Cout = 1, Result = 1, Flags = 0001 (Carry out)

- Результат симуляції:

- Sum = 1

- Cout = 1

- Result = 1

- Flags = 0001

- Висновок: Всі значення правильні, додавання виконане коректно.

**Тестовий випадок 5: Ain = -4, Bin = -3, Cin = 1**

- Вхідні дані: Ain = 5'b11100, Bin = 5'b11101, Cin = 1

- Очікуваний результат: Sum = -6, Cout = 1, Result = -6, Flags = 1000 (Negative result)

- Результат симуляції:

- Sum = -6

- Cout = 1

- Result = -6

- Flags = 1000

- Висновок: Всі значення правильні, додавання виконане коректно.

**Тестовий випадок 6: Ain = 10, Bin = 10, Cin = 1**

- Вхідні дані: Ain = 5'b01010, Bin = 5'b01010, Cin = 1

- Очікуваний результат: Sum = 21, Cout = 1, Result = 21, Flags = 0011 (Carry out and

Overflow)

- Результат симуляції:

- Sum = 21

- Cout = 1

- Result = 21

- Flags = 0011

- Висновок: Всі значення правильні, додавання виконане коректно.

**Тестовий випадок 7: Ain = 3, Bin = -3, Cin = 0**

- Вхідні дані: Ain = 5'b00011, Bin = 5'b11101, Cin = 0

- Очікуваний результат: Sum = 0, Cout = 0, Result = 0, Flags = 0100 (Zero result)

- Результат симуляції:

- Sum = 0

- Cout = 0

- Result = 0

- Flags = 0100

- Висновок: Всі значення правильні, додавання виконане коректно.

**Підсумок**

Всі тестові випадки пройшли успішно, результати відповідають очікуваним. Це свідчить про правильність реалізації модулів Adder5bit та PostProcessor. Додавання було виконано коректно для різних комбінацій вхідних даних, включаючи позитивні та негативні числа, що підтверджує правильну роботу модуля суматора та постпроцесора.