**Ex No:**

**Date: 18.10.2024**

**4-BIT CARRY LOOK AHEAD ADDER**

**Aim:**

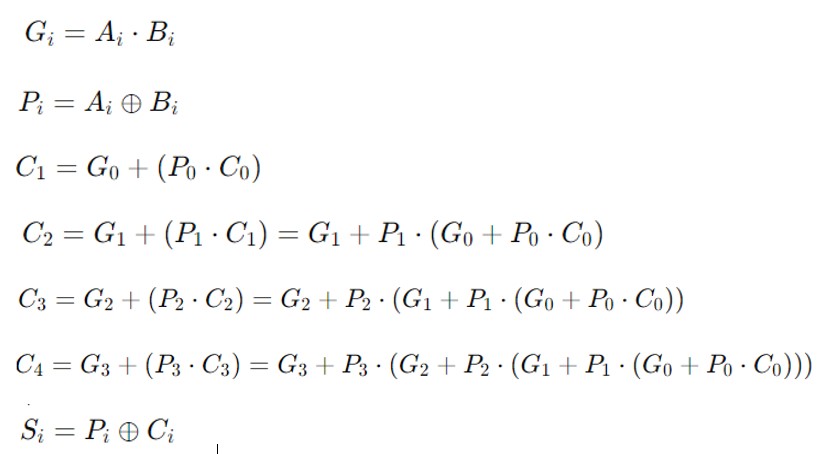
To study the Xilinx with FPGA trainer kit and to implement the full adder Verilog code with FPGA trainer kit.

**Hardware & Software:**

FPGA Trainer Kit & Xilinx ISE Design Suite

**Theory:**

A 4-bit Carry Look-Ahead Adder (CLA) is a digital circuit designed to add two 4-bit binary numbers efficiently by reducing the delay caused by the carry propagation in traditional adders. The CLA adder achieves this by calculating the carry signals in advance using the concepts of generate and propagate, enabling faster addition compared to ripple-carry adders. For each bit, the sum and carry-out are calculated. The carry-out depends on the previous carry, which traditionally creates a delay as each bit must wait for the previous one. The general equations are as follows:



A diagram of a circuit

Description automatically generated

**Source Code:**

module carry\_lookahead\_adder\_4bit (

input [3:0] A, // 4-bit input A

input [3:0] B, // 4-bit input B

input Cin, // Carry input

output [3:0] Sum, // 4-bit Sum output

output Cout // Carry output

);

wire [3:0] P; // Propagate signals

wire [3:0] G; // Generate signals

wire [3:0] C; // Carry signals

// Propagate and Generate signals

assign P = A ^ B; // P[i] = A[i] ^ B[i]

assign G = A & B; // G[i] = A[i] & B[i]

// Carry signals

assign C[0] = Cin;

assign C[1] = G[0] | (P[0] & C[0]);

assign C[2] = G[1] | (P[1] & C[1]);

assign C[3] = G[2] | (P[2] & C[2]);

assign Cout = G[3] | (P[3] & C[3]);

// Sum calculation

assign Sum = P ^ C;

endmodule

**Stimulus Code:**

module tb\_carry\_lookahead\_adder\_4bit;

// Inputs

reg [3:0] A;

reg [3:0] B;

reg Cin;

// Outputs

wire [3:0] Sum;

wire Cout;

// Instantiate the 4-bit Carry Look-Ahead Adder

carry\_lookahead\_adder\_4bit uut (

.A(A),

.B(B),

.Cin(Cin),

.Sum(Sum),

.Cout(Cout)

);

// Testbench procedure

initial begin

// Initialize inputs

A = 4'b0000; B = 4'b0000; Cin = 1'b0;

// Monitor outputs

$monitor("A = %b, B = %b, Cin = %b : Sum = %b, Cout = %b", A, B, Cin, Sum, Cout);

#10 A = 4'b0001; B = 4'b0010; Cin = 1'b0;

#10 A = 4'b0101; B = 4'b0011; Cin = 1'b0;

#10 A = 4'b1111; B = 4'b0001; Cin = 1'b1;

#10 A = 4'b1010; B = 4'b0101; Cin = 1'b0;

#10 A = 4'b0111; B = 4'b0111; Cin = 1'b1;

// Stop the simulation

#10 $stop;

end

endmodule

**Simulated Output:**

A screenshot of a computer

Description automatically generated

**Synthesis Report:**

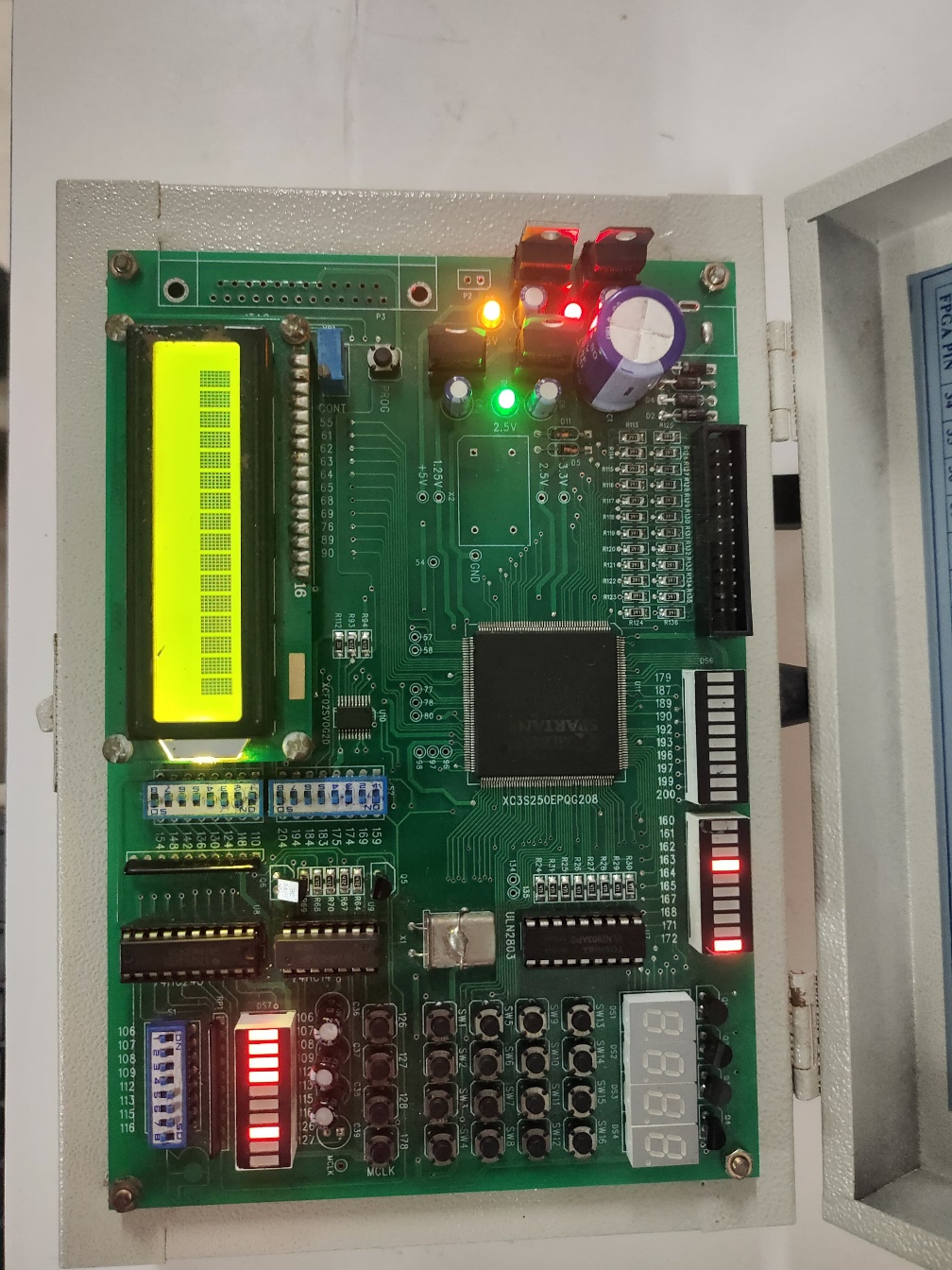
A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**FPGA TRAINER KIT Photos:**



**Result:**

Thus, the carry look ahead adder is simulated using Xilinx ISE Tool and has been verified by implementing in FPGA Trainer Kit.