EXP NO: 07

DATA PATH CIRCUITS

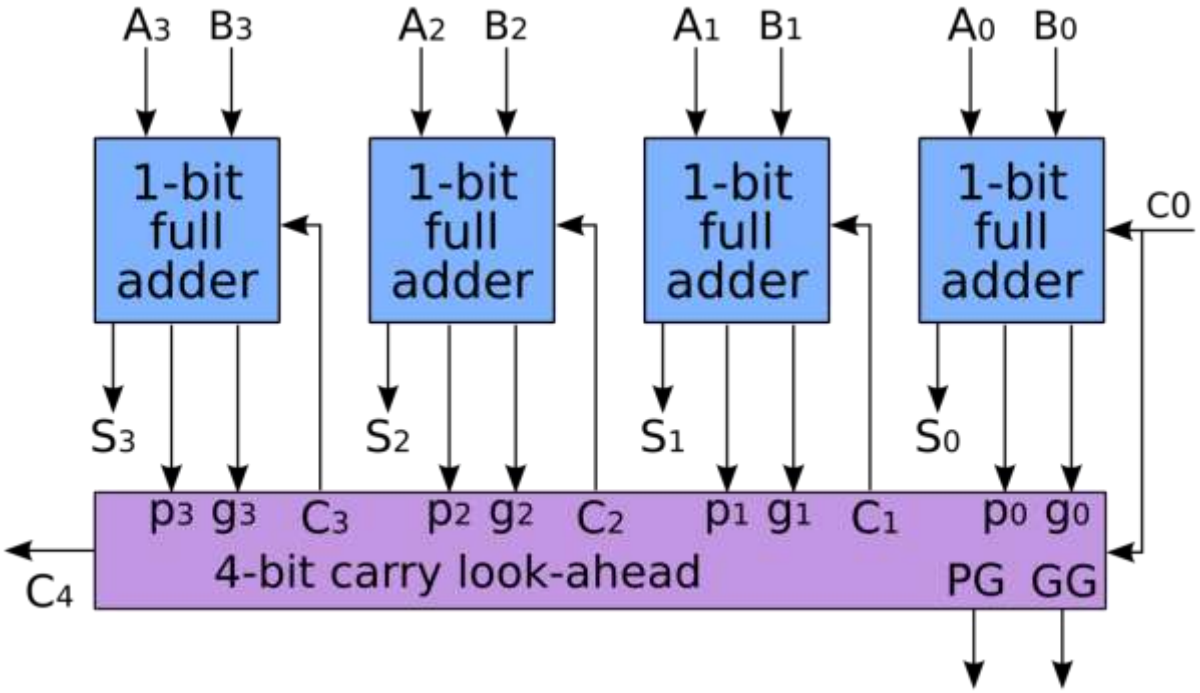
AIM:

Develop the Verilog source code for a **Carry Look-Ahead Adder (CLA), and Magnitude Comparator,** and verify their functionality through simulation.

ALGORITHM:

Step1: Define the specifications and initialize the design.
Step2: Write the source code in VERILOG.
Step3: Check the syntax and debug the errors if found, after compilation.
Step4: Verify the output by simulating the source code for various cases.

Carry Look -Ahead Adder

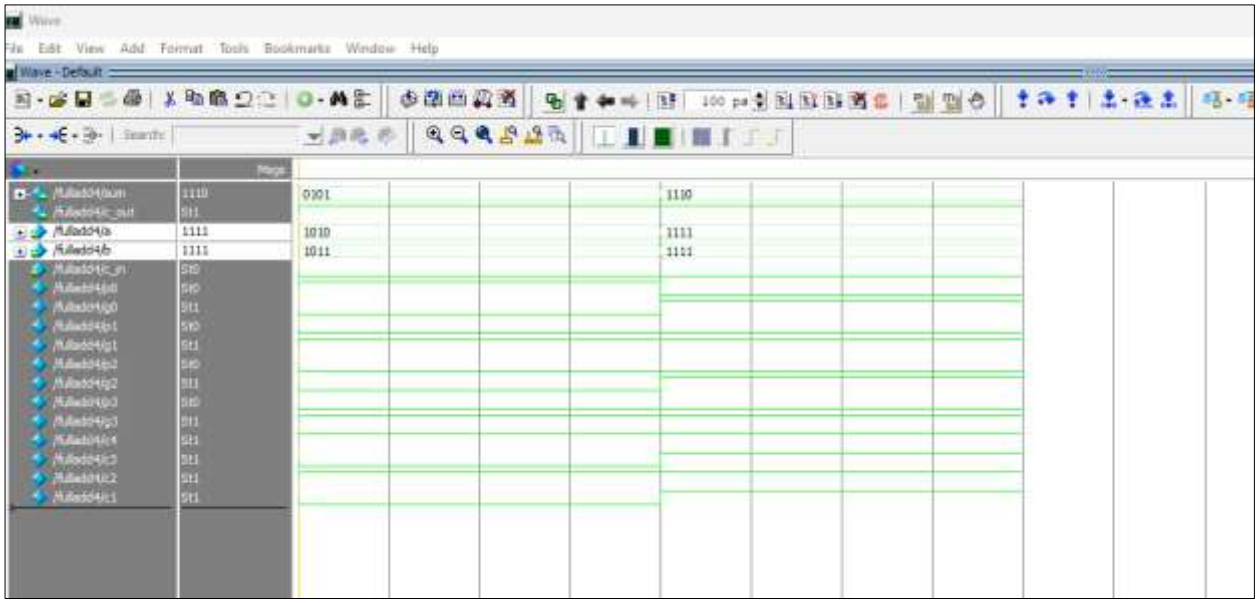


VERILOG SOURCE CODE:

```
module fulladd4(sum, c_out, a, b, c_in);
// Inputs and outputs
output [3:0] sum;
output c_out;
input [3:0] a,b;
input c_in;
// Internal wires
wire p0,g0, p1,g1, p2,g2, p3,g3;
wire c4, c3, c2, c1;
// compute the p for each stage
assign p0 = a[0] ^ b[0],
p1 = a[1] ^ b[1],
```

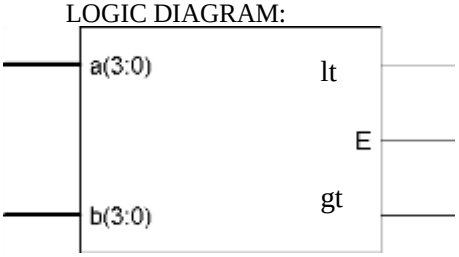
```
p2 = a[2] ^ b[2],
p3 = a[3] ^ b[3];
// compute the g for each stage
assign g0 = a[0] & b[0],
g1 = a[1] & b[1],
g2 = a[2] & b[2],
g3 = a[3] & b[3];

// compute the carry for each stage
// Note that c_in is equivalent c0 in the arithmetic equation for
// carry lookahead computation
assign c1 = g0 | (p0 & c_in),
c2 = g1 | (p1 & g0) | (p1 & p0 & c_in),
c3 = g2 | (p2 & g1) | (p2 & p1 & g0) | (p2 & p1 & p0 & c_in),
c4 = g3 | (p3 & g2) | (p3 & p2 & g1) | (p3 & p2 & p1 & g0) |
(p3 & p2 & p1 & p0 & c_in);
// Compute Sum
assign sum[0] = p0 ^ c_in,
sum[1] = p1 ^ c1,
sum[2] = p2 ^ c2,
sum[3] = p3 ^ c3;
// Assign carry output
assign c_out = c4;
endmodule
```

Simulation output :



MAGNITUDE COMPARATOR:

LOGIC DIAGRAM:

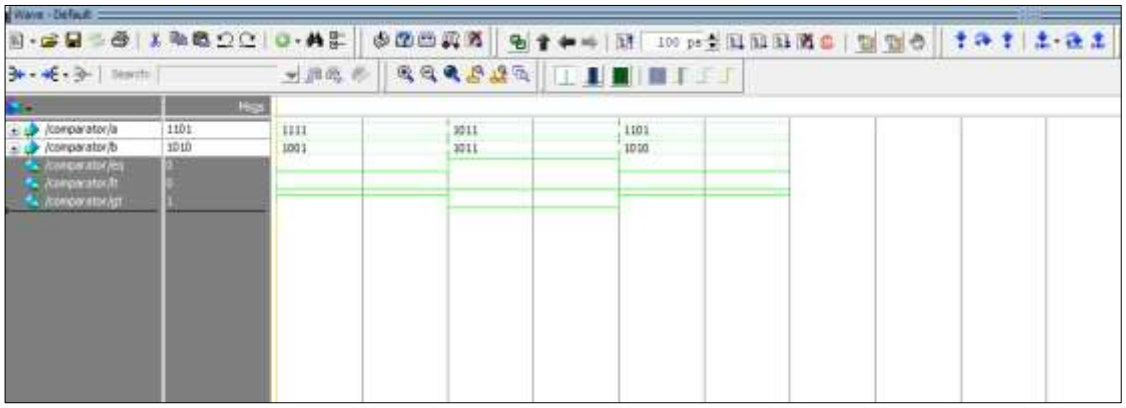


VERILOG SOURCE CODE:

Behavioral Modeling:

```
module comparator(a,b,eq,lt,gt);
input [3:0] a,b;
output reg eq,lt,gt;
always @(a,b)
begin if (a==b)
begin eq = 1'b1;
 lt = 1'b0;
 gt = 1'b0;
end
else if (a>b)
begin eq = 1'b0;
lt = 1'b0;
gt = 1'b1;
end
else begin eq = 1'b0;
lt = 1'b1;
gt = 1'b0;
end
end
endmodule
```

Simulation output:



RESULT:

Thus the outputs of the carry look ahead adder and magnitude comparator was verified using Verilog code in Modelsim.