LOOK A -HEAD CARRY ADDER

Order for carry lookahead adders to function, two bits termed Carry Propagate and Carry Generate, denoted by Cp and Cg, are generated. The Cg bit is used to generate the output carry bit, which is independent of the input carry bit, once the Cp bit has been propagated….

The drawback of Ripple carry adder' is that it has a carry propagation delay that introduces slow computation. Since adlers are used in desims like multipliers and divisions, it causes slowness in their computation. To tackle this issue, a carry look- abeud adder (CLA) can be used that reduces propagation delay with additional hardware complexity.

CLA has introduced some functions like "carry generate (G)" and "carry propagate (P) to boost the speed.

Carry Generate (G): This function denotes how the carry is generated for single-bit two inputs regardless of any input carry.

As we have seen in the full adder, carry is generated using the equation as A.B Hence C=A\cdot R (similar to how carry is generated by full adder)

Carry Propagate (P): This function denotes when the carry is propagated to the next stage with an addition whenever there is an input carry.

RTL CODE:

```
module carry(A,B,cin,sum,cout);
  input [3:0]A,B;
  input cin;
  output [3:0]sum;
  output cout;
  wire [3:0]ci;
  assign ci[0]=cin;
  assign ci[1]= A[0]&B[0]| (A[0]^B[0])&ci[0];
  assign ci[2]= A[1]&B[1]| (A[1]^B[1])&ci[1];
  assign ci[3]= A[2]&B[2]| (A[2]^B[2])&ci[2];
```

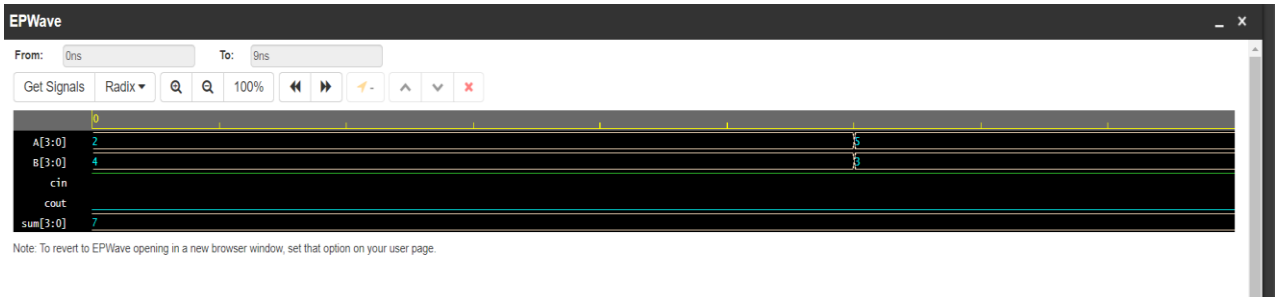
```
  assign cout = A[3]&B[3]| (A[3]^B[3])&ci[3];
  assign sum=A^B^cin;
endmodule
```

TEST BENCH:

```
module testbench;
  reg [3:0] A,B;
  reg cin;
  wire [3:0] sum;
  wire cout;
  carry  lca(A,B,cin,sum,cout);

  initial
    begin
      $dumpfile("dump.vcd");
      $dumpvars(1);
    end
  initial
    begin

      $monitor("A=%b, B=%b, cin=%b --> sum=%b, cout=%b", A, B, cin, sum,
cout);
      A = 1; B = 0; cin = 0;
      A = 2; B = 4; cin = 1; #6;
      A = 5; B = 3; cin = 1; #3;
    end
endmodule
```


EPWave
From:
0ns
To:
9ns
Get Signals
Radix
100%
0
A[3:0]
2
5
B[3:0]
4
3
cin
cout
sum[3:0]
7
Note: To revert to EPWave opening in a new browser window, set that option on your user page.