Homework 5

Fall 2014 DSD

Name ______Andrew Quick________ abc123 ___qxt050__________ Date ___10/2/2024_

1. Design a 4 bit ripple carry adder and use gate delay of 5 ns calculate the delay for a 4 bit add

```
module FullAdder (A, B, Ci, Co, S);

    input A, B, Ci;     // Inputs: A, B, and Carry In (Ci)
    output S, Co;       // Outputs: Sum (S) and Carry Out (Co)

    assign #5 S = A ^ B ^ Ci;      // Sum calculation
    assign #5 Co = (A & B) | (B & Ci) | (A & Ci);  // Carry Out
calculation

endmodule


module Adder4 (S, Co, A, B, Ci);

    input [3:0] A, B;  // 4-bit inputs
    input Ci;          // Carry input
    output [3:0] S;    // 4-bit sum output
    output Co;         // Carry out
    wire [3:1] C;      // Internal carry signals

    // Instantiate 4 Full Adders
    FullAdder FA0 (A[0], B[0], Ci, C[1], S[0]);
    FullAdder FA1 (A[1], B[1], C[1], C[2], S[1]);
    FullAdder FA2 (A[2], B[2], C[2], C[3], S[2]);
    FullAdder FA3 (A[3], B[3], C[3], Co, S[3]);

endmodule
```

Total delay for the 4-bit adder is 20 ns.

2. Design a 4 bit carry look ahead adder and use gate delay of 5ns and calculate the delay for a 4 bit add

```
module cla_4bit (
    input [3:0] A, B,
    input Cin,
    output [3:0] Sum,
    output Cout
);
    wire [3:0] G, P;
    wire C1, C2, C3;

    assign G = A & B;  // Generate Signal
    assign P = A | B;  // Propagate SIgnal

    // Carry look-ahead logic
    assign #5 C1 = G[0] | (P[0] & Cin);
    assign #5 C2 = G[1] | (P[1] & G[0]) | (P[1] & P[0] & Cin);
    assign #5 C3 = G[2] | (P[2] & G[1]) | (P[2] & P[1] & G[0]) | (P[2] &
P[1] & P[0] & Cin);
    assign #5 Cout = G[3] | (P[3] & G[2]) | (P[3] & P[2] & G[1]) | (P[3] &
P[2] & P[1] & G[0]) | (P[3] & P[2] & P[1] & P[0] & Cin);

    // Sum calculation
    assign #5 Sum[0] = A[0] ^ B[0] ^ Cin;
    assign #5 Sum[1] = A[1] ^ B[1] ^ C1;
    assign #5 Sum[2] = A[2] ^ B[2] ^ C2;
    assign #5 Sum[3] = A[3] ^ B[3] ^ C3;

endmodule
```

Sum is calculated in parallel therefore total delay is 5 * 4 = 20ns.

3. Does the delay difference between the designs increase linearly or not as the number of bits increases?

In a ripple carry adder, the delay grows linearly with the number of bits, as each bit requires the carry to propagate through all previous stages, resulting in a delay of O(N). In contrast, a carry look-ahead adder computes the carry signals in parallel, leading to a logarithmic delay growth, O(logN). This means that as the number of bits increases, the CLA becomes significantly faster than the RCA, with the delay difference expanding non-linearly. For large bit widths, the CLA's delay advantage is much more pronounced.