­­­4.

(1).

module cla\_gl(

    output C3,       // carry output

    output[2:0] S,   // sum

    input[2:0] A, B, // operands

    input C0         // carry input

    );

    // TODO:: Implement gate-level CLA

    wire [2:0] p, g;

    //0: first bit

    AND g\_0(g[0], A[0], B[0]);

    OR p\_0(p[0], A[0], B[0]);

    AND pc\_0(pc0, p[0], C0);

    OR c\_1(C1, g[0], pc0);

    FA fa\_0(temp0, S[0], A[0], B[0], C0);

    //1: second bit

    AND g\_1(g[1], A[1], B[1]);

    OR p\_1(p[1], A[1], B[1]);

    AND4 c\_0p\_0p\_1(c0p0p1, C0, p[0], p[1], 1'b1);

    AND g\_0p\_1(g0p1, g[0], p[1]);

    OR4 c\_2(C2, g[1], g0p1, c0p0p1, 1'b0);

    FA fa\_1(temp1, S[1], A[1], B[1], C1);

    //2: third bit

    AND g\_2(g[2], A[2], B[2]);

    OR p\_2(p[2], A[2], B[2]);

    AND4 c\_0p\_0p\_1p\_2(c0p0p1p2, C0, p[0], p[1], p[2]);

    AND4 g\_0p\_1p\_2(g0p1p2, g[0], p[1], p[2], 1'b1);

    AND g\_1p\_2(g1p2, g[1], p[2]);

    OR4 c\_3(C3, g[2], g1p2, g0p1p2, c0p0p1p2);

    FA fa\_2(temp2, S[2], A[2], B[2], C2);

endmodule

(2).

module rca\_gl(

    output C3,       // carry output

    output[2:0] S,   // sum

    input[2:0] A, B, // operands

    input C0         // carry input

    );

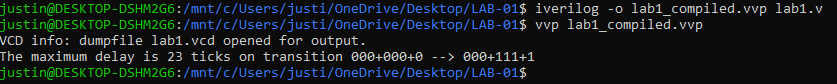
    // TODO:: Implement gate-level RCA

    FA fa\_0(C1, S[0], A[0], B[0], C0);

    FA fa\_1(C2, S[1], A[1], B[1], C1);

    FA fa\_2(C3, S[2], A[2], B[2], C2);

endmodule

­­­6. (a)Maximum propogation delay for rca\_gl (ripple carry adder)

(b) Maximum propogation delay for cla\_gl (carry lookahead adder)

