

EE5311 Tutorial_8

L.Pradeep EE22B074

1. Add a clock input and register the existing inputs/outputs at the posedge of clock for the three 16-bit tree adders in Assignment 7.
2. Use **OpenLane** to generate layouts and obtain post-layout STA report for the three adders.
3. Find the maximum operating frequency of the input clock for each tree adder.

BRENTKUNG

```
//-----
// 16-bit Brent- Kung Adder (5-stage pipeline – no multi-output gates)
//-----
module brentkung #(
    parameter WIDTH = 16
)(
    input wire      clk,
    input wire      rst, // synchronous reset, active-high
    input wire [WIDTH-1:0] a,
    input wire [WIDTH-1:0] b,
    input wire      cin,
    output reg [WIDTH-1:0] sum,
    output reg      cout
);

// -----
// Stage 0: Register Inputs
// -----
reg [WIDTH-1:0] a_r, b_r;
reg      cin_r;
always @(posedge clk) begin
    if (rst) begin
        a_r <= 0;
        b_r <= 0;
        cin_r <= 0;
    end else begin
        a_r <= a;
        b_r <= b;
        cin_r <= cin;
    end
end

// -----
// Stage 1: Compute bitwise P/G
// -----
wire [WIDTH-1:0] P1, G1;
genvar i;
generate
    for (i = 0; i < WIDTH; i = i + 1) begin : PG1
        assign P1[i] = a_r[i] ^ b_r[i];
        assign G1[i] = a_r[i] & b_r[i];
    end
endgenerate

reg [WIDTH-1:0] P1_r, G1_r;
always @(posedge clk) begin
    if (rst) {P1_r, G1_r} <= 0;
    else {P1_r, G1_r} <= {P1, G1 };
end
```

```

end

// -----
// Stage 2: Level-2 prefix (pairs)
// -----
wire [WIDTH/2-1:0] P2, G2;
generate
  for (i = 0; i < WIDTH/2; i = i + 1) begin : LVL2
    pg_dot u2(
      .P1 (P1_r[2*i]), .G1(G1_r[2*i]),
      .P2 (P1_r[2*i+1]), .G2(G1_r[2*i+1]),
      .Pout(P2[i]), .Gout(G2[i])
    );
  end
endgenerate

reg [WIDTH/2-1:0] P2_r, G2_r;
always @(posedge clk) begin
  if (rst) {P2_r, G2_r} <= 0;
  else {P2_r, G2_r} <= {P2, G2};
end

// -----
// Stage 3: Level-3 prefix (groups of 4)
// -----
wire [WIDTH/4-1:0] P3, G3;
generate
  for (i = 0; i < WIDTH/4; i = i + 1) begin : LVL3
    pg_dot u3(
      .P1 (P2_r[2*i]), .G1(G2_r[2*i]),
      .P2 (P2_r[2*i+1]), .G2(G2_r[2*i+1]),
      .Pout(P3[i]), .Gout(G3[i])
    );
  end
endgenerate

reg [WIDTH/4-1:0] P3_r, G3_r;
always @(posedge clk) begin
  if (rst) {P3_r, G3_r} <= 0;
  else {P3_r, G3_r} <= {P3, G3};
end

// -----
// Stage 4: Level-4 prefix (groups of 8)
// -----
wire [WIDTH/8-1:0] P4, G4;
generate
  for (i = 0; i < WIDTH/8; i = i + 1) begin : LVL4
    pg_dot u4(
      .P1 (P3_r[2*i]), .G1(G3_r[2*i]),
      .P2 (P3_r[2*i+1]), .G2(G3_r[2*i+1]),
      .Pout(P4[i]), .Gout(G4[i])
    );
  end
endgenerate

reg [WIDTH/8-1:0] P4_r, G4_r;
always @(posedge clk) begin
  if (rst) {P4_r, G4_r} <= 0;
  else {P4_r, G4_r} <= {P4, G4};
end

// -----
// Stage 5: Level-5 prefix (entire 16)
// -----
wire P5, G5;

```

```

pg_dot u5(
    .P1 (P4_r[0]), .G1(G4_r[0]),
    .P2 (P4_r[1]), .G2(G4_r[1]),
    .Pout(P5),    .Gout(G5)
);

reg P5_r, G5_r;
always @(posedge clk) begin
    if (rst) {P5_r, G5_r} <= 0;
    else {P5_r, G5_r} <= {P5, G5};
end

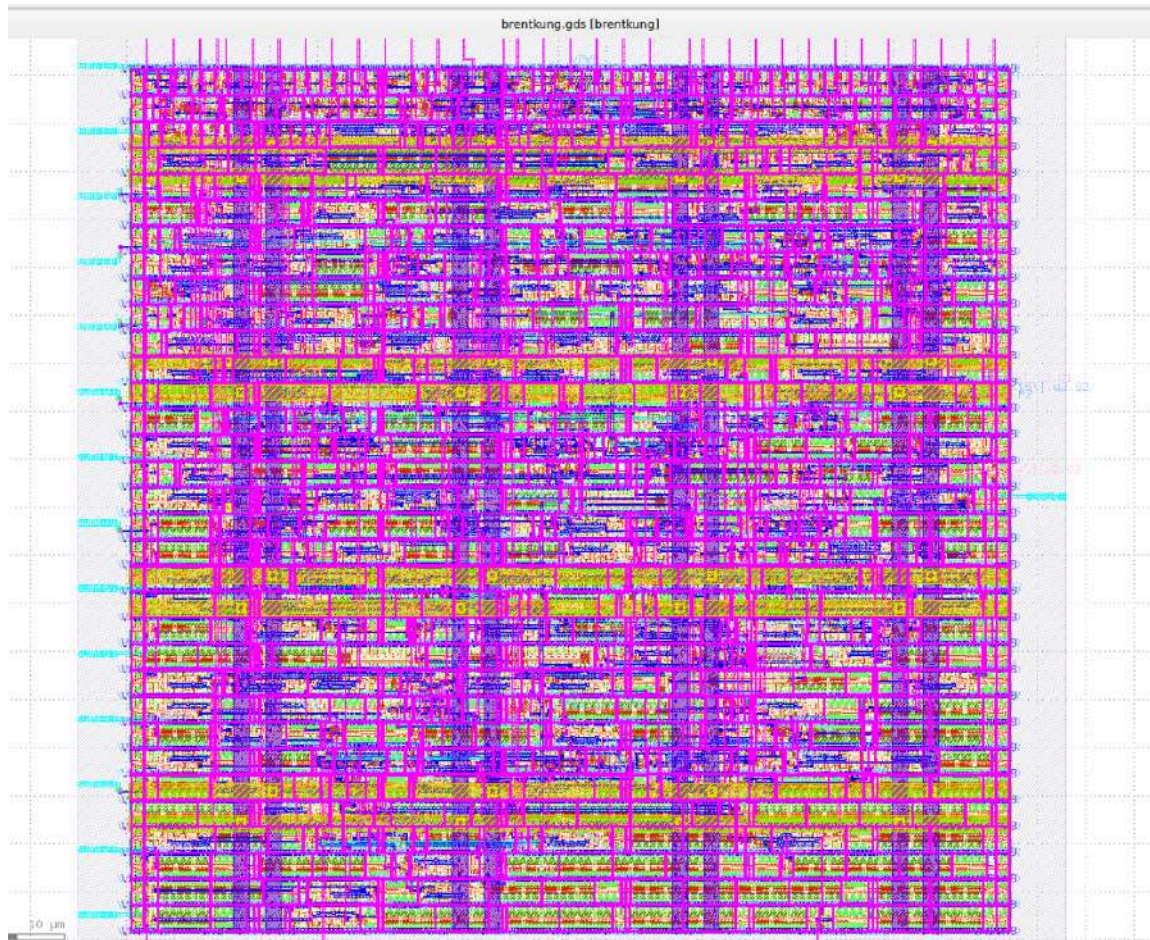
// -----
// Stage 6: Compute final carry chain and register sum & cout
// -----
wire [WIDTH:0] carry;
assign carry[0] = cin_r;
assign carry[1] = G1_r[0] | (P1_r[0] & carry[0]);
assign carry[2] = G2_r[0] | (P2_r[0] & carry[0]);
assign carry[3] = G2_r[1] | (P2_r[1] & carry[1]);
assign carry[4] = G3_r[0] | (P3_r[0] & carry[0]);
assign carry[5] = G3_r[1] | (P3_r[1] & carry[1]);
assign carry[6] = G3_r[2] | (P3_r[2] & carry[2]);
assign carry[7] = G3_r[3] | (P3_r[3] & carry[3]);
assign carry[8] = G4_r[0] | (P4_r[0] & carry[0]);
assign carry[9] = G1_r[8] | (P1_r[8] & carry[8]);
assign carry[10] = G2_r[4] | (P2_r[4] & carry[8]);
assign carry[11] = G1_r[10] | (P1_r[10] & carry[10]);
assign carry[12] = G3_r[2] | (P3_r[2] & carry[8]);
assign carry[13] = G1_r[12] | (P1_r[12] & carry[12]);
assign carry[14] = G2_r[6] | (P2_r[6] & carry[12]);
assign carry[15] = G1_r[14] | (P1_r[14] & carry[14]);
assign carry[16] = G5_r | (P5_r & carry[0]);

always @(posedge clk) begin
    if (rst) begin
        sum <= 0;
        cout <= 0;
    end else begin
        sum <= P1_r ^ carry[WIDTH-1:0];
        cout <= carry[WIDTH];
    end
end

endmodule

//-----
// pg_dot: 2-input prefix operator
//-----
module pg_dot (
    input wire P1, G1,
    input wire P2, G2,
    output wire Pout, Gout
);
    assign Gout = G2 | (P2 & G1);
    assign Pout = P2 & P1;
endmodule

```



typical corner

58166

58167 Startpoint: 623 (rising edge-triggered flip-flop clocked by clock)

58168 Endpoint: 527 (rising edge-triggered flip-flop clocked by clock)

58169 Path Group: clock

58170 Path Type: max

58171 Corner: Typical

58172

58173 Fanout	Cap	Slew	Delay	Time	Description
58174					
58175			0.00	0.00	clock clock (rise edge)
58176			0.00	0.00	clock source latency
58177	1	0.02	0.00	0.00	clk (in)
58178			0.00	0.00	clk (net)
58179		0.00	0.00	0.00	clkbuf_0 clk/A (sky130_fd_sc_hd_clkbuf_16)
58180	8	0.08	0.10	0.16	clkbuf_0 clk/X (sky130_fd_sc_hd_clkbuf_16)
58181			0.00	0.16	clknet_0 clk (net)
58182		0.07	0.00	0.33	clkbuf_3_4 f clk/A (sky130_fd_sc_hd_clkbuf_16)
58183	9	0.05	0.07	0.17	clkbuf_3_4 f clk/X (sky130_fd_sc_hd_clkbuf_16)
58184			0.00	0.33	clknet_3_4 leaf clk (net)
58185		0.07	0.00	0.33	623 /CLK (sky130_fd_sc_hd_dfxtpt_2)
58186	7	0.03	0.00	0.37	623 /0 (sky130_fd_sc_hd_dfxtpt_2)
58187			0.00	0.70	carry[0] (net)
58188		0.00	0.00	0.70	v_340 /A1 (sky130_fd_sc_hd_a21o_1)
58189	4	0.01	0.00	0.21	v_340 /X (sky130_fd_sc_hd_a21o_1)
58190			0.00	0.90	_130 (net)
58191		0.00	0.00	0.90	v_355 /A2 (sky130_fd_sc_hd_a21o_1)
58192	3	0.01	0.07	0.22	v_355 /X (sky130_fd_sc_hd_a21o_1)
58193			0.00	1.13	_141 (net)
58194		0.07	0.00	1.13	v_362 /A2 (sky130_fd_sc_hd_a21o_1)
58195	2	0.01	0.04	0.19	v_362 /X (sky130_fd_sc_hd_a21o_1)
58196			0.00	1.32	_146 (net)
58197		0.04	0.00	1.32	v_363 /0 (sky130_fd_sc_hd_and2_1)
58198	3	0.01	0.06	0.18	v_363 /X (sky130_fd_sc_hd_and2_1)
58199			0.00	1.51	_147 (net)
58200		0.06	0.00	1.51	v_366 /C (sky130_fd_sc_hd_or3_1)
58201	1	0.00	0.06	0.30	v_366 /X (sky130_fd_sc_hd_or3_1)
58202			0.00	1.81	_149 (net)
58203		0.06	0.00	1.81	v_368 /0 (sky130_fd_sc_hd_and3b_1)
58204	1	0.00	0.03	0.16	v_368 /X (sky130_fd_sc_hd_and3b_1)
58205			0.00	1.97	_151 (net)
58206		0.03	0.00	1.97	v_369 /A (sky130_fd_sc_hd_clkbuf_1)
58207	1	0.00	0.03	2.05	v_369 /X (sky130_fd_sc_hd_clkbuf_1)
58208			0.00	2.05	_015 (net)
58209		0.03	0.00	2.05	v_527 /D (sky130_fd_sc_hd_dfxtpt_1)
58210				2.05	data arrival time
58211					
58212			5.00	5.00	clock clock (rise edge)
58213			0.00	5.00	clock source latency
58214	1	0.02	0.00	5.00	clk (in)
58215			0.00	5.00	clk (net)
58216		0.00	0.00	5.00	clkbuf_0 clk/A (sky130_fd_sc_hd_clkbuf_16)
58217	8	0.08	0.10	5.16	clkbuf_0 clk/X (sky130_fd_sc_hd_clkbuf_16)
58218			0.00	5.16	clknet_0 clk (net)
58219		0.10	0.00	5.16	clkbuf_3_7 f clk/A (sky130_fd_sc_hd_clkbuf_16)
58220	16	0.05	0.07	5.33	clkbuf_3_7 f clk/X (sky130_fd_sc_hd_clkbuf_16)
58221			0.00	5.34	clknet_3_7 leaf clk (net)
58222		0.07	0.00	5.34	527 /CLK (sky130_fd_sc_hd_dfxtpt_1)
58223			0.00	5.34	clock reconvergence pessimism
58224			0.10	5.34	clock setup time

```

[INFO]: Running OpenROAD Antenna Rule Checker (log: designs/brentkung/runs/auto_pnr/logs/signoff/41-arc.log)...
[INFO]: Saving current set of views in 'designs/brentkung/runs/auto_pnr/results/final'...
[INFO]: Saving runtime environment...
[INFO]: Generating final set of reports...
[INFO]: Created manufacturability report at 'designs/brentkung/runs/auto_pnr/reports/manufacturability.rpt'.
[INFO]: Created metrics report at 'designs/brentkung/runs/auto_pnr/reports/metrics.csv'.
[INFO]: There are no max slew, max fanout or max capacitance violations in the design at the Typical corner.
[INFO]: There are no hold violations in the design at the Typical corner.
[INFO]: There are no setup violations in the design at the Typical corner.
[SUCCESS]: Flow complete.
[INFO]: Note that the following warnings have been generated:
[WARNING]: 1 warnings found by linter
[WARNING]: Current core area is too small for the power grid settings chosen. The power grid was scaled down to an
offset of 1/8 the core width and height and a pitch of 1/4 the core width and height.
[WARNING]: VSRC_LOC_FILES was not given a value, which may make the results of IR drop analysis inaccurate. If you
are not integrating a top-level chip for manufacture, you may ignore this warning, otherwise, see the documentation
for VSRC_LOC_FILES.

Final layout path      : /home/ee22b074/ee5311/designs/brentkung/runs/auto_pnr/results/signoff/brentkung.gds
Post-layout STA report : /home/ee22b074/ee5311/designs/brentkung/runs/auto_pnr/logs/signoff/30-rcx_mcsta.nom.log

```

KOGGESTONE

```

// -----
// 16-bit Pipelined Kogge-Stone Adder
// -----
module kogge #(parameter WIDTH = 16)(
    input wire      clk,
    input wire      rst,
    input wire [WIDTH-1:0] a,
    input wire [WIDTH-1:0] b,
    input wire      cin,
    output reg [WIDTH-1:0] sum,
    output reg      cout
);

    // Stage 0: Generate P and G (initial)
    wire [WIDTH-1:0] P0, G0;
    assign P0 = a ^ b;
    assign G0 = a & b;

    reg [WIDTH-1:0] P0_r, G0_r;
    reg      cin_r;
    always @(posedge clk) begin
        if (rst) begin
            P0_r <= 0; G0_r <= 0; cin_r <= 0;
        end else begin
            P0_r <= P0; G0_r <= G0; cin_r <= cin;
        end
    end

    // Stage 1: level 1 prefix (distance 1)
    wire [WIDTH-1:0] G1, P1;
    genvar i;
    generate
        for (i = 0; i < WIDTH; i = i + 1) begin: LEVEL1
            if (i == 0) begin
                assign G1[i] = G0_r[i];
                assign P1[i] = P0_r[i];
            end
        end
    endgenerate

```

```

    end else begin
        assign G1[i] = G0_r[i] | (P0_r[i] & G0_r[i-1]);
        assign P1[i] = P0_r[i] & P0_r[i-1];
    end
end
endgenerate

```

```

reg [WIDTH-1:0] G1_r, P1_r;
always @(posedge clk) begin
    if (rst) {G1_r, P1_r} <= 0;
    else    {G1_r, P1_r} <= {G1, P1};
end

```

```

// Stage 2: level 2 prefix (distance 2)
wire [WIDTH-1:0] G2, P2;
generate
    for (i = 0; i < WIDTH; i = i + 1) begin: LEVEL2
        if (i < 2) begin
            assign G2[i] = G1_r[i];
            assign P2[i] = P1_r[i];
        end else begin
            assign G2[i] = G1_r[i] | (P1_r[i] & G1_r[i-2]);
            assign P2[i] = P1_r[i] & P1_r[i-2];
        end
    end
endgenerate

```

```

reg [WIDTH-1:0] G2_r, P2_r;
always @(posedge clk) begin
    if (rst) {G2_r, P2_r} <= 0;
    else    {G2_r, P2_r} <= {G2, P2};
end

```

```

// Stage 3: level 3 prefix (distance 4)
wire [WIDTH-1:0] G3, P3;
generate
    for (i = 0; i < WIDTH; i = i + 1) begin: LEVEL3
        if (i < 4) begin
            assign G3[i] = G2_r[i];
            assign P3[i] = P2_r[i];
        end else begin
            assign G3[i] = G2_r[i] | (P2_r[i] & G2_r[i-4]);
            assign P3[i] = P2_r[i] & P2_r[i-4];
        end
    end
endgenerate

```

```

reg [WIDTH-1:0] G3_r;
always @(posedge clk) begin
    if (rst) G3_r <= 0;
    else    G3_r <= G3;
end

```



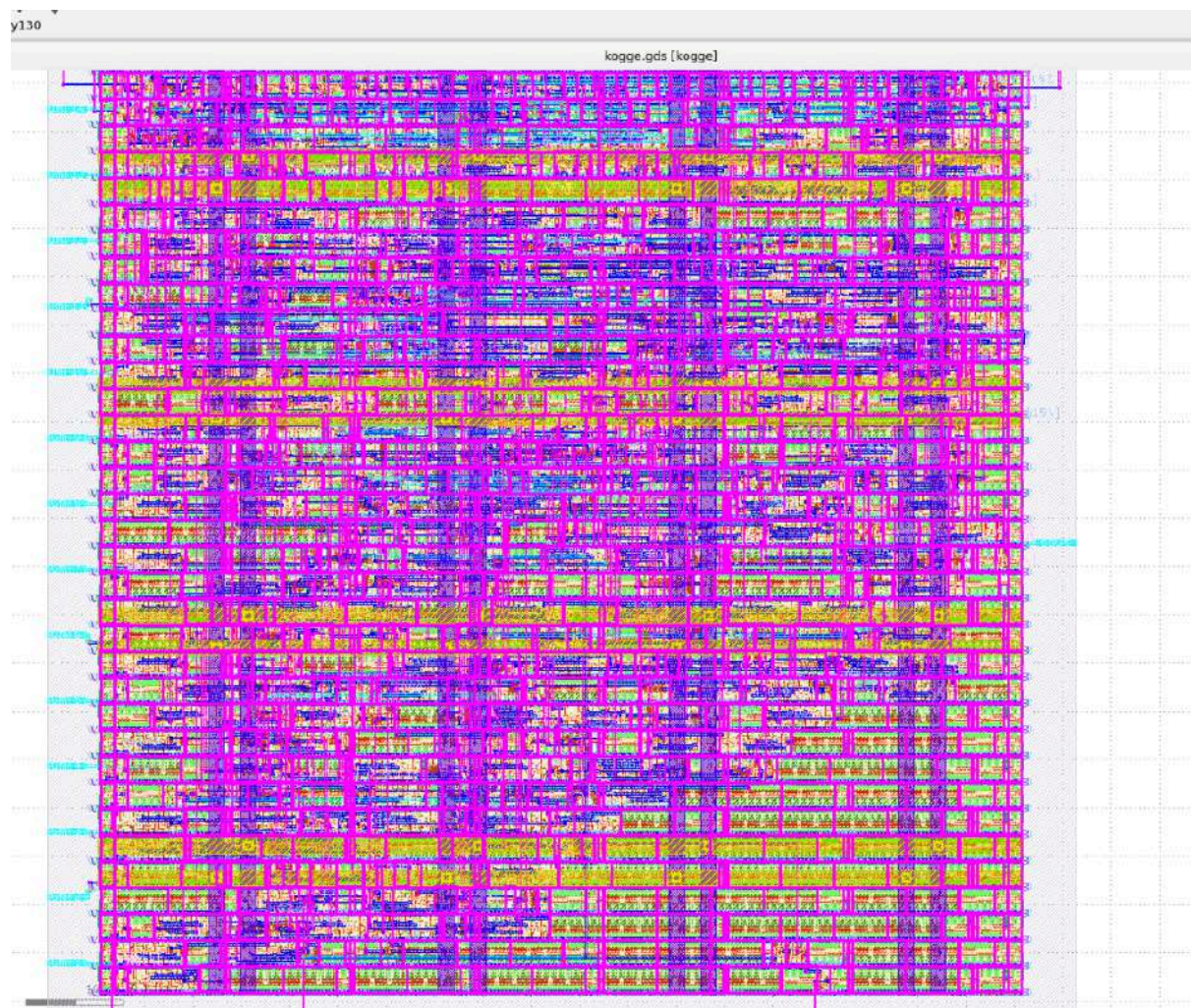
```

// Stage 4: Final carry and sum
wire [WIDTH:0] carry;
assign carry[0] = cin_r;
generate
  for (i = 0; i < WIDTH; i = i + 1) begin: CARRY_GEN
    assign carry[i+1] = G3_r[i] | (P0_r[i] & carry[i]);
  end
endgenerate

always @(posedge clk) begin
  if (rst) begin
    sum <= 0;
    cout <= 0;
  end else begin
    sum <= P0_r ^ carry[WIDTH-1:0];
    cout <= carry[WIDTH];
  end
end

endmodule

```



Corner: Slowest					
Fanout	Cap	Slew	Delay	Time	Description
			0.00	0.00	clock core_clock (rise edge)
			0.00	0.00	clock source latency
1	0.02	0.00	0.00	0.00	clk (in)
					clk (net)
		0.00	0.00	0.00	clkbuf 0 clk/A (sky130_fd_sc_hd_clkbuf_16)
16	0.11	0.19	0.29	0.29	clkbuf 0 clk/X (sky130_fd_sc_hd_clkbuf_16)
					clknet 0 clk (net)
		0.19	0.00	0.30	clkbuf 4 0 0 clk/A (sky130_fd_sc_hd_clkbuf_8)
12	0.03	0.11	0.31	0.61	clkbuf 4 0 0 clk/X (sky130_fd_sc_hd_clkbuf_8)
					clknet 4 0 0 clk (net)
		0.11	0.00	0.61	634 /CLK (sky130_fd_sc_hd_dfxtpt_1)
2	0.01	0.07	0.61	1.22	634 /0 (sky130_fd_sc_hd_dfxtpt_1)
					P0_r[0] (net)
		0.07	0.00	1.22	277 /B (sky130_fd_sc_hd_and2_1)
3	0.01	0.12	0.37	1.59	277 /X (sky130_fd_sc_hd_and2_1)
					129 (net)
		0.12	0.00	1.59	280 /A1 (sky130_fd_sc_hd_o21a_1)
2	0.01	0.09	0.38	1.97	280 /X (sky130_fd_sc_hd_o21a_1)
					131 (net)
		0.09	0.00	1.97	283 /A2 (sky130_fd_sc_hd_o21a_1)
3	0.01	0.12	0.37	2.35	283 /X (sky130_fd_sc_hd_o21a_1)
					133 (net)
		0.12	0.00	2.35	286 /A2 (sky130_fd_sc_hd_o21a_1)
3	0.01	0.12	0.38	2.73	286 /X (sky130_fd_sc_hd_o21a_1)
					135 (net)
		0.12	0.00	2.73	289 /A2 (sky130_fd_sc_hd_o21a_1)
3	0.01	0.13	0.39	3.12	289 /X (sky130_fd_sc_hd_o21a_1)
					137 (net)
		0.13	0.00	3.12	292 /A2 (sky130_fd_sc_hd_o21a_1)
3	0.01	0.12	0.39	3.51	292 /X (sky130_fd_sc_hd_o21a_1)
					139 (net)
		0.12	0.00	3.51	296 /A2 (sky130_fd_sc_hd_o21a_1)
3	0.01	0.11	0.38	3.89	296 /X (sky130_fd_sc_hd_o21a_1)
					142 (net)
		0.11	0.00	3.89	299 /A2 (sky130_fd_sc_hd_o21a_1)
3	0.01	0.13	0.39	4.28	299 /X (sky130_fd_sc_hd_o21a_1)
					144 (net)
		0.13	0.00	4.28	302 /A2 (sky130_fd_sc_hd_o21a_1)
3	0.01	0.13	0.39	4.67	302 /X (sky130_fd_sc_hd_o21a_1)
					146 (net)
		0.13	0.00	4.67	305 /A2 (sky130_fd_sc_hd_o21a_1)
3	0.01	0.11	0.38	5.05	305 /X (sky130_fd_sc_hd_o21a_1)
					148 (net)
		0.11	0.00	5.05	308 /A2 (sky130_fd_sc_hd_o21a_1)
3	0.01	0.12	0.38	5.43	308 /X (sky130_fd_sc_hd_o21a_1)
					150 (net)
		0.12	0.00	5.43	311 /A2 (sky130_fd_sc_hd_o21a_1)
3	0.01	0.11	0.38	5.81	311 /X (sky130_fd_sc_hd_o21a_1)
					152 (net)
		0.11	0.00	5.81	314 /A2 (sky130_fd_sc_hd_o21a_1)
3	0.01	0.12	0.38	6.19	314 /X (sky130_fd_sc_hd_o21a_1)
					154 (net)
		0.12	0.00	6.19	317 /A2 (sky130_fd_sc_hd_o21a_1)
3	0.01	0.12	0.39	6.58	317 /X (sky130_fd_sc_hd_o21a_1)
					156 (net)
		0.12	0.00	3.51	296 /A2 (sky130_fd_sc_hd_o21a_1)
3	0.01	0.11	0.38	3.89	296 /X (sky130_fd_sc_hd_o21a_1)
					142 (net)
		0.11	0.00	3.89	299 /A2 (sky130_fd_sc_hd_o21a_1)
3	0.01	0.13	0.39	4.28	299 /X (sky130_fd_sc_hd_o21a_1)
					144 (net)
		0.13	0.00	4.28	302 /A2 (sky130_fd_sc_hd_o21a_1)
3	0.01	0.13	0.39	4.67	302 /X (sky130_fd_sc_hd_o21a_1)
					146 (net)
		0.13	0.00	4.67	305 /A2 (sky130_fd_sc_hd_o21a_1)
3	0.01	0.11	0.38	5.05	305 /X (sky130_fd_sc_hd_o21a_1)
					148 (net)
		0.11	0.00	5.05	308 /A2 (sky130_fd_sc_hd_o21a_1)
3	0.01	0.12	0.38	5.43	308 /X (sky130_fd_sc_hd_o21a_1)
					150 (net)
		0.12	0.00	5.43	311 /A2 (sky130_fd_sc_hd_o21a_1)
3	0.01	0.11	0.38	5.81	311 /X (sky130_fd_sc_hd_o21a_1)
					152 (net)
		0.11	0.00	5.81	314 /A2 (sky130_fd_sc_hd_o21a_1)
3	0.01	0.12	0.38	6.19	314 /X (sky130_fd_sc_hd_o21a_1)
					154 (net)
		0.12	0.00	6.19	317 /A2 (sky130_fd_sc_hd_o21a_1)
3	0.01	0.12	0.39	6.58	317 /X (sky130_fd_sc_hd_o21a_1)
					156 (net)
		0.12	0.00	6.58	320 /A2 (sky130_fd_sc_hd_o21a_1)
3	0.01	0.11	0.38	6.96	320 /X (sky130_fd_sc_hd_o21a_1)
					158 (net)
		0.11	0.00	6.96	323 /C (sky130_fd_sc_hd_or3_1)
1	0.00	0.13	0.71	7.67	323 /X (sky130_fd_sc_hd_or3_1)
					160 (net)
		0.13	0.00	7.67	325 /B (sky130_fd_sc_hd_and3_1)
1	0.00	0.07	0.35	8.02	325 /X (sky130_fd_sc_hd_and3_1)
					162 (net)
		0.07	0.00	8.02	326 /A (sky130_fd_sc_hd_clkbuf_1)
1	0.00	0.05	0.16	8.18	326 /X (sky130_fd_sc_hd_clkbuf_1)
					039 (net)
		0.05	0.00	8.18	558 /D (sky130_fd_sc_hd_dfxtpt_1)
					data arrival time
			5.00	5.00	clock core_clock (rise edge)
			0.00	5.00	clock source latency
1	0.02	0.00	0.00	5.00	clk (in)
					clk (net)
		0.00	0.00	5.00	clkbuf 0 clk/A (sky130_fd_sc_hd_clkbuf_16)
16	0.11	0.19	0.29	5.29	clkbuf 0 clk/X (sky130_fd_sc_hd_clkbuf_16)
					clknet 0 clk (net)
		0.19	0.00	5.30	clkbuf 4 14 0 clk/A (sky130_fd_sc_hd_clkbuf_8)
8	0.03	0.11	0.31	5.61	clkbuf 4 14 0 clk/X (sky130_fd_sc_hd_clkbuf_8)
					clknet 4 14 0 clk (net)
		0.11	0.00	5.61	558 /CLK (sky130_fd_sc_hd_dfxtpt_1)
			0.00	5.61	clock reconvergence pessimism
			-0.25	5.36	library setup time
				5.36	data required time
				5.36	data required time
				-8.18	data arrival time
				-2.82	slack (VIOLATED)


```
[STEP 41]
[INFO]: Running OpenROAD Antenna Rule Checker (log: designs/kogge/runs/auto_pnr/logs/signoff/41-arc.log)...
[INFO]: Saving current set of views in 'designs/kogge/runs/auto_pnr/results/final'...
[INFO]: Saving runtime environment...
[INFO]: Generating final set of reports...
[INFO]: Created manufacturability report at 'designs/kogge/runs/auto_pnr/reports/manufacturability.rpt'.
[INFO]: Created metrics report at 'designs/kogge/runs/auto_pnr/reports/metrics.csv'.
[INFO]: There are no max slew, max fanout or max capacitance violations in the design at the Typical corner.
[INFO]: There are no hold violations in the design at the Typical corner.
[INFO]: There are no setup violations in the design at the Typical corner.
[SUCCESS]: Flow complete.
[INFO]: Note that the following warnings have been generated:
[WARNING]: 8 warnings found by linter
[WARNING]: Current core area is too small for the power grid settings chosen. The power grid was scaled down to an
offset of 1/8 the core width and height and a pitch of 1/4 the core width and height.
[WARNING]: VSRC_LOC_FILES was not given a value, which may make the results of IR drop analysis inaccurate. If you
are not integrating a top-level chip for manufacture, you may ignore this warning, otherwise, see the documentation
for VSRC_LOC_FILES.

Final layout path      : /home/ee22b074/ee5311/designs/kogge/runs/auto_pnr/results/signoff/kogge.gds
Post-layout STA report : /home/ee22b074/ee5311/designs/kogge/runs/auto_pnr/logs/signoff/30-rcx_mcsta.nom.log
```

SKLANKSKY

```
//-----
// 16-bit Sklansky Adder (Pipelined)
//-----
module sklansky #(
    parameter WIDTH = 16
)(
    input wire      clk,
    input wire      rst, // synchronous reset, active-high
    input wire [WIDTH-1:0] a,
    input wire [WIDTH-1:0] b,
    input wire      cin,
    output reg [WIDTH-1:0] sum,
    output reg      cout
);

// -----
// Stage 0: Register Inputs
// -----
reg [WIDTH-1:0] a_r, b_r;
reg      cin_r;
always @(posedge clk) begin
    if (rst) begin
        a_r <= 0;
        b_r <= 0;
        cin_r <= 0;
    end else begin
        a_r <= a;
        b_r <= b;
        cin_r <= cin;
    end
end

// -----
// Stage 1: Compute bitwise P/G
// -----
```

```

wire [WIDTH-1:0] P0, G0;
assign P0 = a_r ^ b_r;
assign G0 = a_r & b_r;

// -----
// Sklansky Tree Generation
// -----

wire [WIDTH-1:0] G1, P1;
wire [WIDTH-1:0] G2, P2;
wire [WIDTH-1:0] G3, P3;
wire [WIDTH-1:0] G4, P4;

genvar i;
generate
  for (i = 0; i < WIDTH; i = i + 1) begin : STAGE1
    if (i == 0) begin
      assign G1[i] = G0[i];
      assign P1[i] = P0[i];
    end else if (i % 2 == 1) begin
      pg_dot u1 (.P1(P0[i-1]), .G1(G0[i-1]), .P2(P0[i]), .G2(G0[i]), .Pout(P1[i]), .Gout(G1[i]));
      assign G1[i-1] = G0[i-1];
      assign P1[i-1] = P0[i-1];
    end
  end
endgenerate

generate
  for (i = 0; i < WIDTH; i = i + 1) begin : STAGE2
    if (i < 2) begin
      assign G2[i] = G1[i];
      assign P2[i] = P1[i];
    end else if (i % 4 >= 2) begin
      pg_dot u2 (.P1(P1[i-2]), .G1(G1[i-2]), .P2(P1[i]), .G2(G1[i]), .Pout(P2[i]), .Gout(G2[i]));
    end else begin
      assign G2[i] = G1[i];
      assign P2[i] = P1[i];
    end
  end
endgenerate

generate
  for (i = 0; i < WIDTH; i = i + 1) begin : STAGE3
    if (i < 4) begin
      assign G3[i] = G2[i];
      assign P3[i] = P2[i];
    end else if (i % 8 >= 4) begin
      pg_dot u3 (.P1(P2[i-4]), .G1(G2[i-4]), .P2(P2[i]), .G2(G2[i]), .Pout(P3[i]), .Gout(G3[i]));
    end else begin
      assign G3[i] = G2[i];
      assign P3[i] = P2[i];
    end
  end
end

```

```
endgenerate
```

```
generate
```

```
for (i = 0; i < WIDTH; i = i + 1) begin : STAGE4
```

```
    if (i < 8) begin
```

```
        assign G4[i] = G3[i];
```

```
        assign P4[i] = P3[i];
```

```
    end else if (i % 16 >= 8) begin
```

```
        pg_dot u4 (.P1(P3[i-8]), .G1(G3[i-8]), .P2(P3[i]), .G2(G3[i]), .Pout(P4[i]), .Gout(G4[i]));
```

```
    end else begin
```

```
        assign G4[i] = G3[i];
```

```
        assign P4[i] = P3[i];
```

```
    end
```

```
end
```

```
endgenerate
```

```
// -----
```

```
// Final Carry Generation and Sum Calculation
```

```
// -----
```

```
wire [WIDTH:0] carry;
```

```
assign carry[0] = cin_r;
```

```
generate
```

```
for (i = 0; i < WIDTH; i = i + 1) begin : FINAL_CARRY
```

```
    assign carry[i+1] = G4[i] | (P4[i] & carry[0]);
```

```
end
```

```
endgenerate
```

```
always @(posedge clk) begin
```

```
    if (rst) begin
```

```
        sum <= 0;
```

```
        cout <= 0;
```

```
    end else begin
```

```
        sum <= P0 ^ carry[WIDTH-1:0];
```

```
        cout <= carry[WIDTH];
```

```
    end
```

```
end
```

```
endmodule
```

```
//-----
```

```
// pg_dot: 2-input prefix operator
```

```
//-----
```

```
module pg_dot (
```

```
    input wire P1, G1,
```

```
    input wire P2, G2,
```

```
    output wire Pout, Gout
```

```
);
```

```
    assign Gout = G2 | (P2 & G1);
```

```
    assign Pout = P2 & P1;
```

```
endmodule
```

```

[INFO]: Running OpenROAD Antenna Rule Checker (log: designs/sklansky/runs/auto_pnr/logs/signoff/41-arc.log)...
[INFO]: Saving current set of views in 'designs/sklansky/runs/auto_pnr/results/final'...
[INFO]: Saving runtime environment...
[INFO]: Generating final set of reports...
[INFO]: Created manufacturability report at 'designs/sklansky/runs/auto_pnr/reports/manufacturability.rpt'.
[INFO]: Created metrics report at 'designs/sklansky/runs/auto_pnr/reports/metrics.csv'.
[INFO]: There are no max slew, max fanout or max capacitance violations in the design at the Typical corner.
[INFO]: There are no hold violations in the design at the Typical corner.
[INFO]: There are no setup violations in the design at the Typical corner.
[SUCCESS]: Flow complete.
[INFO]: Note that the following warnings have been generated:
[WARNING]: 14 warnings found by linter
[WARNING]: Current core area is too small for the power grid settings chosen. The power grid was scaled down to an
offset of 1/8 the core width and height and a pitch of 1/4 the core width and height.
[WARNING]: VSRC_LOC_FILES was not given a value, which may make the results of IR drop analysis inaccurate. If you
are not integrating a top-level chip for manufacture, you may ignore this warning, otherwise, see the documentation
for VSRC_LOC_FILES.

```

```

Final layout path      : /home/ee22b074/ee5311/designs/sklansky/runs/auto_pnr/results/signoff/sklansky.gds
Post-layout STA report : /home/ee22b074/ee5311/designs/sklansky/runs/auto_pnr/logs/signoff/30-rcx_mcsta.nom.log

```

```

#788
#789 Startpoint: _428_ (rising edge-triggered flip-flop clocked by core_clock)
#790 Endpoint: _420_ (rising edge-triggered flip-flop clocked by core_clock)
#791 Path Group: core_clock
#792 Path Type: max
#793 Corner: Typical
#794
#795 Fanout    Cap    Slew    Delay    Time    Description
#796 -----
#797           0.00    0.00    0.00    0.00    clock core_clock (rise edge)
#798           0.00    0.00    0.00    0.00    clock source latency
#799     1     0.02    0.00    0.00    0.00    ^ clk (in)
#800           clk (net)
#801           0.00    0.00    0.00    0.00    ^ clkbuf_0_clk/A (sky130_fd_sc_hd_clkbuf_16)
#802     4     0.04    0.06    0.13    0.13    ^ clkbuf_0_clk/X (sky130_fd_sc_hd_clkbuf_16)
#803           clknet_0_clk (net)
#804           0.06    0.00    0.13    0.13    ^ clkbuf_2_2_f clk/A (sky130_fd_sc_hd_clkbuf_16)
#805    18     0.05    0.07    0.15    0.28    ^ clkbuf_2_2_f clk/X (sky130_fd_sc_hd_clkbuf_16)
#806           clknet_2_2_leaf clk (net)
#807           0.07    0.00    0.28    0.28    ^ _428_/CLK (sky130_fd_sc_hd_dfxt_1)
#808     3     0.01    0.05    0.32    0.60    v _428_/Q (sky130_fd_sc_hd_dfxt_1)
#809           a_r[7] (net)
#810           0.05    0.00    0.60    0.60    v _255_/B (sky130_fd_sc_hd_and2_1)
#811     2     0.01    0.05    0.16    0.76    v _255_/X (sky130_fd_sc_hd_and2_1)
#812           _096_ (net)
#813           0.05    0.00    0.76    0.76    v _256_/B (sky130_fd_sc_hd_or2_1)
#814     3     0.01    0.07    0.23    1.00    v _256_/X (sky130_fd_sc_hd_or2_1)
#815           _097_ (net)
#816           0.07    0.00    1.00    1.00    v _260_/B (sky130_fd_sc_hd_or2_1)
#817     2     0.01    0.06    0.23    1.23    v _260_/X (sky130_fd_sc_hd_or2_1)
#818           _100_ (net)
#819           0.06    0.00    1.23    1.23    v _261_/B (sky130_fd_sc_hd_or4bb_1)
#820     3     0.01    0.13    0.54    1.77    v _261_/X (sky130_fd_sc_hd_or4bb_1)
#821           _101_ (net)
#822           0.13    0.00    1.77    1.77    v _330_/A1 (sky130_fd_sc_hd_a211lo_1)
#823     1     0.00    0.06    0.38    2.15    v _330_/X (sky130_fd_sc_hd_a211lo_1)
#824           _162_ (net)
#825           0.06    0.00    2.15    2.15    v _334_/A1 (sky130_fd_sc_hd_a21oi_1)
#826     1     0.00    0.11    0.12    2.27    ^ _334_/Y (sky130_fd_sc_hd_a21oi_1)
#827           _016_ (net)
#828           0.11    0.00    2.27    2.27    ^ _420_/D (sky130_fd_sc_hd_dfxt_1)
#829           data arrival time
#830
#831           5.00    5.00    5.00    5.00    clock core_clock (rise edge)
#832           0.00    5.00    5.00    5.00    clock source latency
#833     1     0.02    0.00    0.00    5.00    ^ clk (in)
#834           clk (net)
#835           0.00    0.00    5.00    5.00    ^ clkbuf_0_clk/A (sky130_fd_sc_hd_clkbuf_16)
#836     4     0.04    0.06    0.13    5.13    ^ clkbuf_0_clk/X (sky130_fd_sc_hd_clkbuf_16)
#837           clknet_0_clk (net)
#838           0.06    0.00    5.13    5.13    ^ clkbuf_2_1_f clk/A (sky130_fd_sc_hd_clkbuf_16)
#839     9     0.05    0.07    0.15    5.28    ^ clkbuf_2_1_f clk/X (sky130_fd_sc_hd_clkbuf_16)
#840           clknet_2_1_leaf clk (net)
#841           0.07    0.00    5.28    5.28    ^ _420_/CLK (sky130_fd_sc_hd_dfxt_1)
#842           0.00    5.28    5.28    5.28    clock reconvergence pessimism
#843           -0.06    5.22    5.22    5.22    library setup time
#844           data required time
#845 -----
#846           5.22    data required time
#847           -2.27    data arrival time
#848 -----
#849           2.04    clock (MET)

```