EE5311 Tutorial_8

Mourya Sai Sandeep

EE22B045

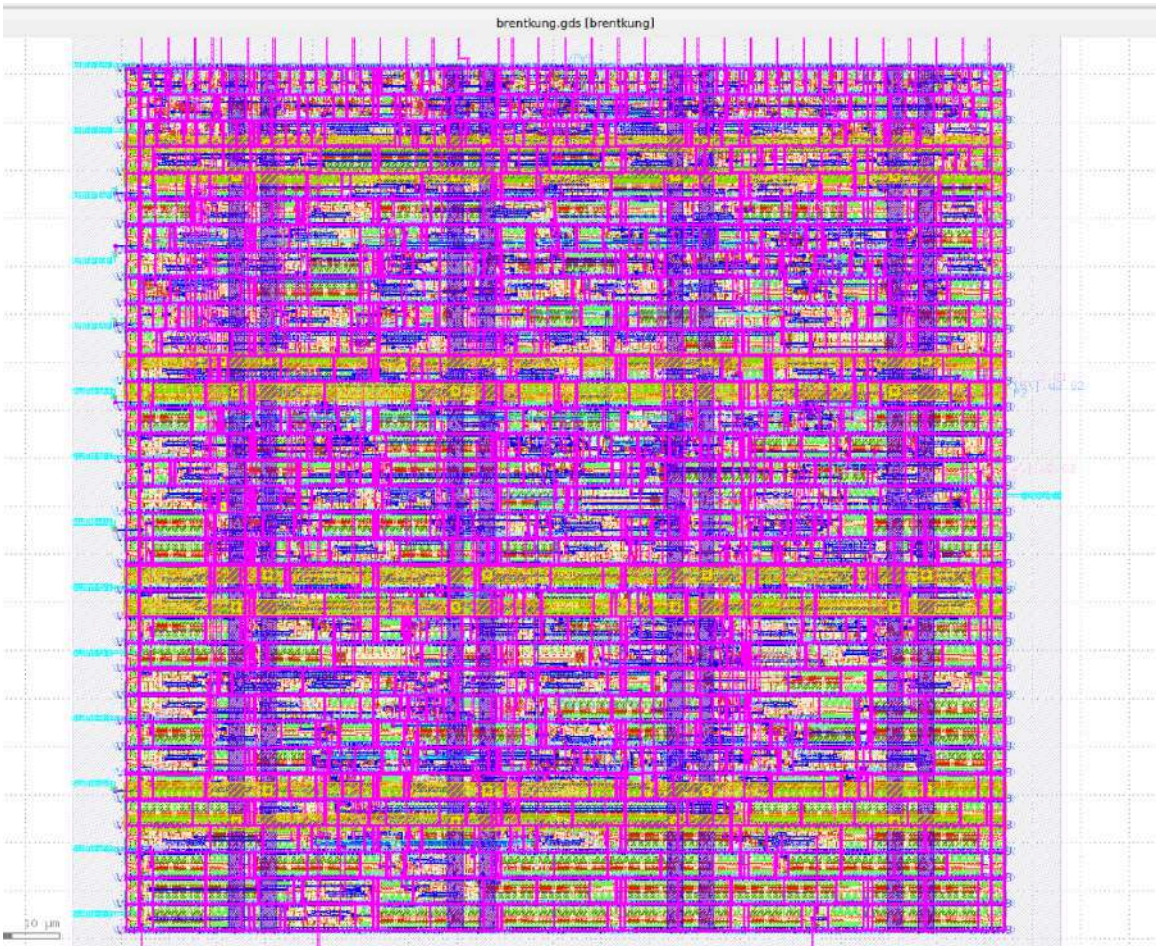
1. Add a clock input and register the existing inputs/outputs at the posedge of clock for the three 16-bit tree adders in Assignment 7.
2. Use `OpenLane` to generate layouts and obtain post-layout STA report for the three adders.
3. Find the maximum operating frequency of the input clock for each tree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

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




```
58165 ========================= 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    0.00 ^ clk (in)
58178                                          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    0.16 ^ clkbuf_0_clk/X (sky130_fd_sc_hd__clkbuf_16)
58181                                          clknet_0_clk (net)
58182                 0.10    0.00    0.16 ^ clkbuf_3_4__f_clk/A (sky130_fd_sc_hd__clkbuf_16)
58183     9    0.05    0.07    0.17    0.33 ^ clkbuf_3_4__f_clk/X (sky130_fd_sc_hd__clkbuf_16)
58184                                          clknet_3_4__leaf_clk (net)
58185                 0.07    0.00    0.33 ^ _623_/CLK (sky130_fd_sc_hd__dfxtp_2)
58186     7    0.03    0.08    0.37    0.70 v _623_/Q (sky130_fd_sc_hd__dfxtp_2)
58187                                          carry[0] (net)
58188                 0.08    0.00    0.70 v _340_/A1 (sky130_fd_sc_hd__a21o_1)
58189     4    0.01    0.08    0.21    0.90 v _340_/X (sky130_fd_sc_hd__a21o_1)
58190                                          _130_ (net)
58191                 0.08    0.00    0.90 v _355_/A2 (sky130_fd_sc_hd__a21o_1)
58192     3    0.01    0.07    0.22    1.13 v _355_/X (sky130_fd_sc_hd__a21o_1)
58193                                          _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    1.32 v _362_/X (sky130_fd_sc_hd__a21o_1)
58196                                          _146_ (net)
58197                 0.04    0.00    1.32 v _363_/B (sky130_fd_sc_hd__and2_1)
58198     3    0.01    0.06    0.18    1.50 v _363_/X (sky130_fd_sc_hd__and2_1)
58199                                          _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    1.81 v _366_/X (sky130_fd_sc_hd__or3_1)
58202                                          _149_ (net)
58203                 0.06    0.00    1.81 v _368_/B (sky130_fd_sc_hd__and3b_1)
58204     1    0.00    0.03    0.16    1.97 v _368_/X (sky130_fd_sc_hd__and3b_1)
58205                                          _151_ (net)
58206                 0.03    0.00    1.97 v _369_/A (sky130_fd_sc_hd__clkbuf_1)
58207     1    0.00    0.03    0.08    2.05 v _369_/X (sky130_fd_sc_hd__clkbuf_1)
58208                                          _015_ (net)
58209                 0.03    0.00    2.05 v _527_/D (sky130_fd_sc_hd__dfxtp_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    0.00    5.00 ^ clk (in)
58215                                          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    0.16    5.16 ^ clkbuf_0_clk/X (sky130_fd_sc_hd__clkbuf_16)
58218                                          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    0.17    5.33 ^ clkbuf_3_7__f_clk/X (sky130_fd_sc_hd__clkbuf_16)
58221                                          clknet_3_7__leaf_clk (net)
58222                 0.07    0.00    5.34 ^ _527_/CLK (sky130_fd_sc_hd__dfxtp_1)
58223                           0.00    5.34   clock reconvergence pessimism
58224                           0.10    5.44   library setup time
```

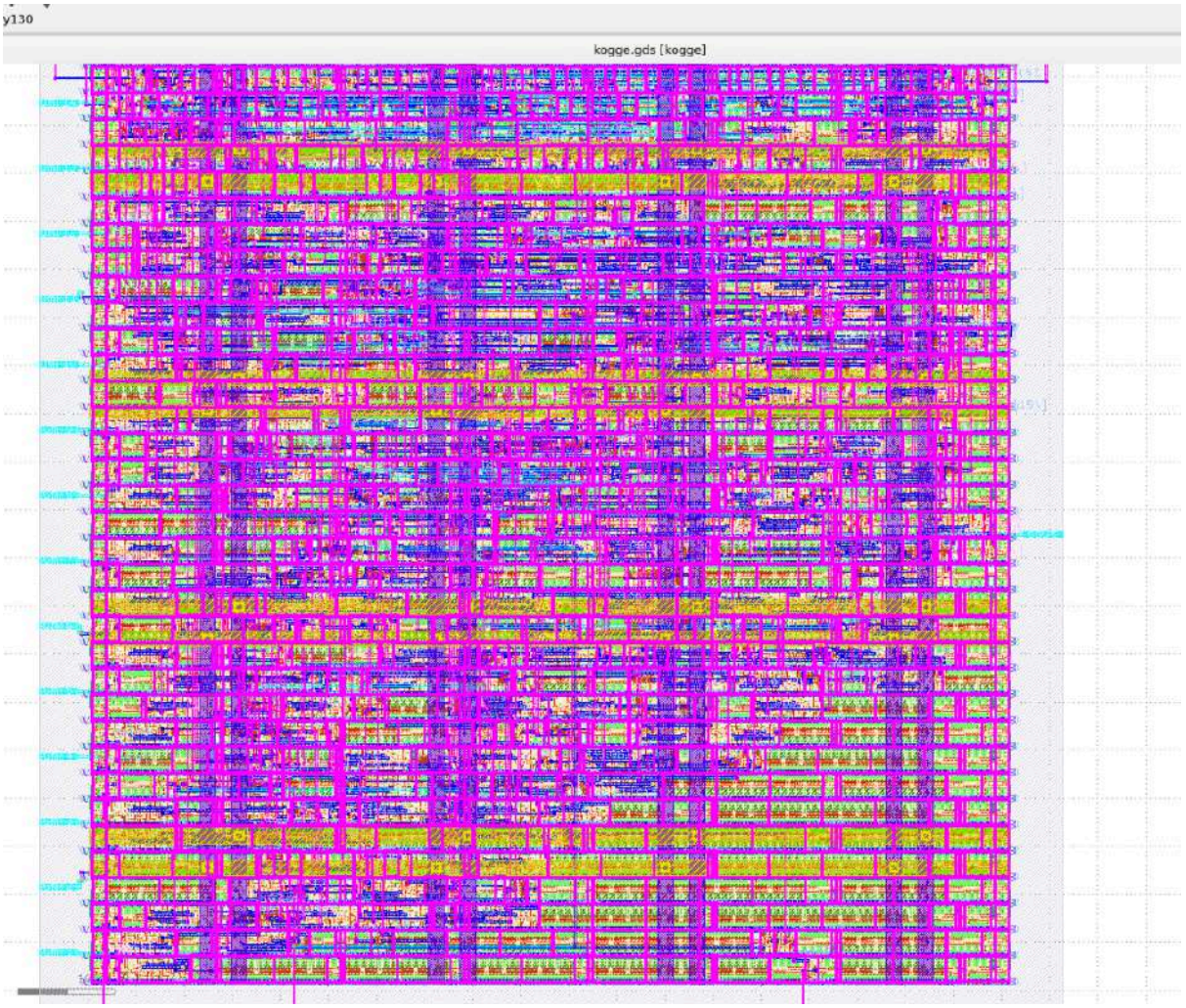
```
[INFO]: Running OpenROAD Antenna Rule Checker (log: designs/brentkung/runs/auto_pnr/logs/signoff/41-arc.log)...
[INFO]: Saving current set of views in 'designs/brentkung/runs/auto_pnr/results/final'...
[INFO]: Saving runtime environment...
[INFO]: Generating final set of reports...
[INFO]: Created 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 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__dfxtp_1)
     2    0.01    0.07    0.61    1.22 v _634_/Q (sky130_fd_sc_hd__dfxtp_1)
                                         P0_r[0] (net)
                    0.07    0.00    1.22 v _277_/B (sky130_fd_sc_hd__and2_1)
     3    0.01    0.12    0.37    1.59 v _277_/X (sky130_fd_sc_hd__and2_1)
                                         _129_ (net)
                    0.12    0.00    1.59 v _280_/A1 (sky130_fd_sc_hd__o21a_1)
     2    0.01    0.09    0.38    1.97 v _280_/X (sky130_fd_sc_hd__o21a_1)
                                         _131_ (net)
                    0.09    0.00    1.97 v _283_/A2 (sky130_fd_sc_hd__o21a_1)
     3    0.01    0.12    0.37    2.35 v _283_/X (sky130_fd_sc_hd__o21a_1)
                                         _133_ (net)
                    0.12    0.00    2.35 v _286_/A2 (sky130_fd_sc_hd__o21a_1)
     3    0.01    0.12    0.38    2.73 v _286_/X (sky130_fd_sc_hd__o21a_1)
                                         _135_ (net)
                    0.12    0.00    2.73 v _289_/A2 (sky130_fd_sc_hd__o21a_1)
     3    0.01    0.13    0.39    3.12 v _289_/X (sky130_fd_sc_hd__o21a_1)
                                         _137_ (net)
                    0.13    0.00    3.12 v _292_/A2 (sky130_fd_sc_hd__o21a_1)
     3    0.01    0.12    0.39    3.51 v _292_/X (sky130_fd_sc_hd__o21a_1)
                                         _139_ (net)
                    0.12    0.00    3.51 v _296_/A2 (sky130_fd_sc_hd__o21a_1)
     3    0.01    0.11    0.38    3.89 v _296_/X (sky130_fd_sc_hd__o21a_1)
                                         _142_ (net)
                    0.11    0.00    3.89 v _299_/A2 (sky130_fd_sc_hd__o21a_1)
     3    0.01    0.13    0.39    4.28 v _299_/X (sky130_fd_sc_hd__o21a_1)
                                         _144_ (net)
                    0.13    0.00    4.28 v _302_/A2 (sky130_fd_sc_hd__o21a_1)
     3    0.01    0.13    0.39    4.67 v _302_/X (sky130_fd_sc_hd__o21a_1)
                                         _146_ (net)
                    0.13    0.00    4.67 v _305_/A2 (sky130_fd_sc_hd__o21a_1)
     3    0.01    0.11    0.38    5.05 v _305_/X (sky130_fd_sc_hd__o21a_1)
                                         _148_ (net)
                    0.11    0.00    5.05 v _308_/A2 (sky130_fd_sc_hd__o21a_1)
     3    0.01    0.12    0.38    5.43 v _308_/X (sky130_fd_sc_hd__o21a_1)
                                         _150_ (net)
                    0.12    0.00    5.43 v _311_/A2 (sky130_fd_sc_hd__o21a_1)
     3    0.01    0.11    0.38    5.81 v _311_/X (sky130_fd_sc_hd__o21a_1)
                                         _152_ (net)
                    0.11    0.00    5.81 v _314_/A2 (sky130_fd_sc_hd__o21a_1)
     3    0.01    0.12    0.38    6.19 v _314_/X (sky130_fd_sc_hd__o21a_1)
                                         _154_ (net)
                    0.12    0.00    6.19 v _317_/A2 (sky130_fd_sc_hd__o21a_1)
     3    0.01    0.12    0.39    6.58 v _317_/X (sky130_fd_sc_hd__o21a_1)
                                         _156_ (net)
                    0.12    0.00    6.58 v _320_/A2 (sky130_fd_sc_hd__o21a_1)
     3    0.01    0.11    0.38    6.96 v _320_/X (sky130_fd_sc_hd__o21a_1)
                                         _158_ (net)
                    0.11    0.00    6.96 v _323_/C (sky130_fd_sc_hd__or3_1)
     1    0.00    0.13    0.71    7.67 v _323_/X (sky130_fd_sc_hd__or3_1)
                                         _160_ (net)
                    0.13    0.00    7.67 v _325_/B (sky130_fd_sc_hd__and3_1)
     1    0.00    0.07    0.35    8.02 v _325_/X (sky130_fd_sc_hd__and3_1)
                                         _162_ (net)
                    0.07    0.00    8.02 v _326_/A (sky130_fd_sc_hd__clkbuf_1)
     1    0.00    0.05    0.16    8.18 v _326_/X (sky130_fd_sc_hd__clkbuf_1)
                                         _039_ (net)
                    0.05    0.00    8.18 v _558_/D (sky130_fd_sc_hd__dfxtp_1)
                                    8.18   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__dfxtp_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

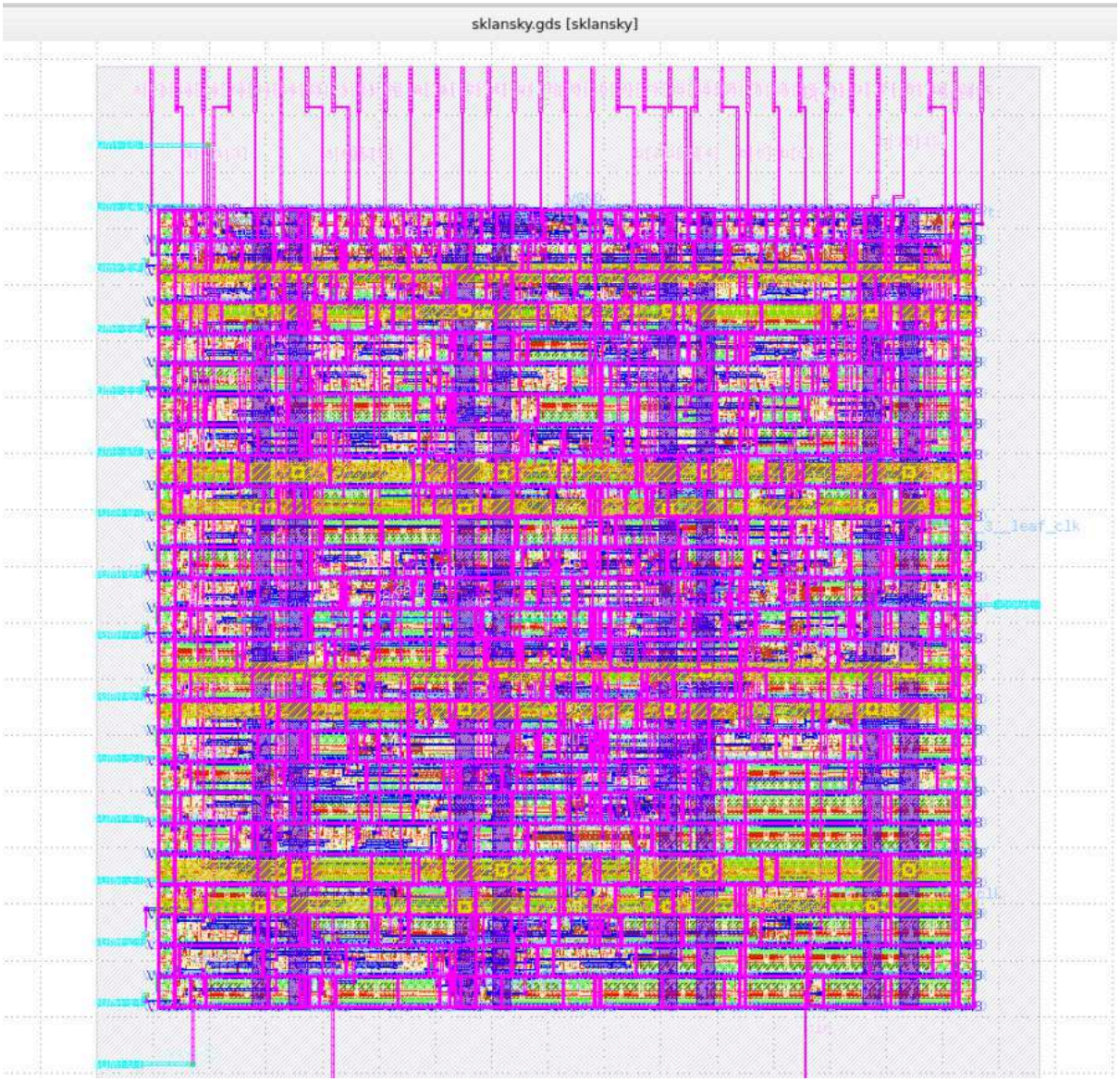
```
//------------------------------------------------------------------------------
// 16-bit Sklansky Adder (Pipelined)
//------------------------------------------------------------------------------
module sklansky #(
  parameter WIDTH = 16
)(
  input  wire              clk,
  input  wire              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
```

```
  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
```

```
9788
9789 Startpoint: _428_ (rising edge-triggered flip-flop clocked by core_clock)
9790 Endpoint: _420_ (rising edge-triggered flip-flop clocked by core_clock)
9791 Path Group: core_clock
9792 Path Type: max
9793 Corner: Typical
9794
9795 Fanout     Cap    Slew   Delay    Time   Description
9796 ---------------------------------------------------------------------------
9797                          0.00    0.00   clock core_clock (rise edge)
9798                          0.00    0.00   clock source latency
9799      1    0.02    0.00    0.00    0.00 ^ clk (in)
9800                                         clk (net)
9801                0.00    0.00    0.00 ^ clkbuf_0_clk/A (sky130_fd_sc_hd__clkbuf_16)
9802      4    0.04    0.06    0.13    0.13 ^ clkbuf_0_clk/X (sky130_fd_sc_hd__clkbuf_16)
9803                                         clknet_0_clk (net)
9804                0.06    0.00    0.13 ^ clkbuf_2_2__f_clk/A (sky130_fd_sc_hd__clkbuf_16)
9805     18    0.05    0.07    0.15    0.28 ^ clkbuf_2_2__f_clk/X (sky130_fd_sc_hd__clkbuf_16)
9806                                         clknet_2_2__leaf_clk (net)
9807                0.07    0.00    0.28 ^ _428_/CLK (sky130_fd_sc_hd__dfxtp_1)
9808      3    0.01    0.05    0.32    0.60 v _428_/Q (sky130_fd_sc_hd__dfxtp_1)
9809                                         a_r[7] (net)
9810                0.05    0.00    0.60 v _255_/B (sky130_fd_sc_hd__and2_1)
9811      2    0.01    0.05    0.16    0.76 v _255_/X (sky130_fd_sc_hd__and2_1)
9812                                         _096_ (net)
9813                0.05    0.00    0.76 v _256_/B (sky130_fd_sc_hd__or2_1)
9814      3    0.01    0.07    0.23    1.00 v _256_/X (sky130_fd_sc_hd__or2_1)
9815                                         _097_ (net)
9816                0.07    0.00    1.00 v _260_/B (sky130_fd_sc_hd__or2_1)
9817      2    0.01    0.06    0.23    1.23 v _260_/X (sky130_fd_sc_hd__or2_1)
9818                                         _100_ (net)
9819                0.06    0.00    1.23 v _261_/B (sky130_fd_sc_hd__or4bb_1)
9820      3    0.01    0.13    0.54    1.77 v _261_/X (sky130_fd_sc_hd__or4bb_1)
9821                                         _101_ (net)
9822                0.13    0.00    1.77 v _330_/A1 (sky130_fd_sc_hd__a2111o_1)
9823      1    0.00    0.06    0.38    2.15 v _330_/X (sky130_fd_sc_hd__a2111o_1)
9824                                         _162_ (net)
9825                0.06    0.00    2.15 v _334_/A1 (sky130_fd_sc_hd__a21oi_1)
9826      1    0.00    0.11    0.12    2.27 ^ _334_/Y (sky130_fd_sc_hd__a21oi_1)
9827                                         _016_ (net)
9828                0.11    0.00    2.27 ^ _420_/D (sky130_fd_sc_hd__dfxtp_1)
9829                                 2.27   data arrival time
9830
9831                          5.00    5.00   clock core_clock (rise edge)
9832                          0.00    5.00   clock source latency
9833      1    0.02    0.00    0.00    5.00 ^ clk (in)
9834                                         clk (net)
9835                0.00    0.00    5.00 ^ clkbuf_0_clk/A (sky130_fd_sc_hd__clkbuf_16)
9836      4    0.04    0.06    0.13    5.13 ^ clkbuf_0_clk/X (sky130_fd_sc_hd__clkbuf_16)
9837                                         clknet_0_clk (net)
9838                0.06    0.00    5.13 ^ clkbuf_2_1__f_clk/A (sky130_fd_sc_hd__clkbuf_16)
9839      9    0.05    0.07    0.15    5.28 ^ clkbuf_2_1__f_clk/X (sky130_fd_sc_hd__clkbuf_16)
9840                                         clknet_2_1__leaf_clk (net)
9841                0.07    0.00    5.28 ^ _420_/CLK (sky130_fd_sc_hd__dfxtp_1)
9842                          0.00    5.28   clock reconvergence pessimism
9843                         -0.06    5.22   library setup time
9844                                 5.22   data required time
9845 ---------------------------------------------------------------------------
9846                                 5.22   data required time
9847                                -2.27   data arrival time
9848 ---------------------------------------------------------------------------
9849                                 2.94   slack (MET)
```


sklansky.gds [sklansky]