1. Introduction

Clocking power becomes an addressable factor as the working frequency increases. In typical cell-based synchronous designs, clocking network contributes 20~30% of the total power. Since clock nets have the highest switching activity, it is necessary to reduce the clocking power in high performance designs. **In this contest, students are required to develop a program to synthesize a low power clock tree.** The techniques include buffer insertion, buffer sizing, buffer relocation, and buffer removal under the constraint of the maximum clock skew.

Clock tree synthesis is usually performed after cell placement to get more accurate physical information. The clock latency of a flip-flop’s clock pin is the path delay starting from the root of clock tree (i.e., the first buffer), through the distribution cells, and ending at the clock pin (leaf pin). In other words, the arrival time of a flip-flop’s clock pin is its clock latency. The clock skew between two flip-flops is the difference of their clock latency. Using a clock tree shown in Figure 1 as an example, Table 1 gives the information of this clock tree.

![Figure 1](image)

<table>
<thead>
<tr>
<th>Flip-Flop</th>
<th>Clock latency</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>0.36</td>
</tr>
<tr>
<td>R2</td>
<td>0.28</td>
</tr>
<tr>
<td>Clock Skew</td>
<td>0.08</td>
</tr>
</tbody>
</table>

Table 1
2. Problem Description

Given (1) a design that has been implemented with a cell placement and an initial clock tree, (2) Synopsys liberty library that includes clock buffers and flip-flops, the developed software has to apply allowable techniques, such as buffer insertion, buffer resizing, buffer relocation, and buffer removal to reduce the dynamic power under the constraint of the maximum clock skew. Note that the initial clock tree is only for reference. Students are allowed to re-synthesize a better clock tree, except that the root of the clock tree should be the same as the root of the initial clock tree.

3. Input

The default time, capacitance, and distance (coordinate) units are in nano-second (ns), pico-fara (pf), and micron-meter (um), respectively.

(1). Design file (design.def)
The format of the .def file is as follows.

```
DIEAREA  (lower-left coordinate) (upper-right coordinate)
PINS
    Pin_name1  Direction  X1-coordinate  Y1-coordinate
    Pin_name2  Direction  X2-coordinate  Y2-coordinate
    ...
END PINS
COMPONENTS
    Instance_Name1  Cell_Name1  X3-coordinate  Y3-coordinate
    Instance_Name2  Cell_Name1  X4-coordinate  Y4-coordinate
    ...
END COMPONENTS
NET
    Net_Name1  Type  Instance_Name1.pin1  Instance_Name2.pin1 ...
    Net_Name2  Type  Instance_Name4.pin2  Instance_Name5.pin2 ...
    ...
END NET
```

*lower-left coordinate*: the coordinate at the lower-left corner of the die.
*upper-right coordinate*: the coordinate at the upper-right corner of the die
*Pin_name*: the pin name of input, output, and in/out pins
*Direction*: the direction of the corresponding pin. It can be IN, OUT, or INOUT.
**X-coordinate**: the coordinate in X-axis.

**Y-coordinate**: the coordinate in Y-axis.

**Instance_Name**: the instance name of a placed cell.

**Cell_Name**: the name of a placed cell.

**Net_Name**: the name of an interconnect.

**Type**: the type of the specified net. It can be **CLOCK** or **SIGNAL**.

**Instance_Name.pin**: the pin name of the cell connected to the net.

For example, u1/u10/F1.CK represents that the instance name is u1/u10/F1 and CK pin is connected to the net. The slash “/” stands for hierarchy divider.

### (2). Synopsys liberty library of clock buffers/ flip-flops

The given .lib file is in Synopsys liberty format library and contains timing and power information and input capacitance of the cell. It is often a two-dimension table-look-up model. The non-linear model is related to the input transition time and the output capacitance. Below is an example:

```plaintext
cell (CLKBUFX12) {
    cell_footprint : clkbuf;
    area : 86.4000;
    pin(A) {
        direction : input;
        capacitance : 0.02588;
    }
    pin(Y) {
        direction : output;
        capacitance : 0.0;
        function : "A";
        internal_power() {
            related_pin : "A";
            rise_power(energy_template_7x7) {
                index_1 ("0.0500, 0.1000, 0.4000, 0.9000, 1.4000, 2.0000, 3.0000");
                index_2 ("0.00720, 0.36000, 0.72000, 1.80000, 3.24000, 4.68000, 6.12000");
                values ( \
                    "0.3816, 0.3020, 0.2630, 0.2000, 0.1100, 0.0300, -0.0600", \ 
                    "0.4223, 0.3770, 0.3510, 0.2900, 0.2100, 0.1200, 0.0300", \ 
                    "0.5766, 0.4460, 0.4320, 0.2800, 0.1500, 0.2200, 0.1300", \ 
                    "0.8698, 0.7270, 0.6350, 0.5000, 0.4000, 0.3000, 0.2100", \ 
                    "1.1520, 0.9620, 0.8730, 0.7600, 0.6600, 0.5600, 0.4700", \ 
                    "1.4930, 1.2510, 1.1860, 1.0100, 0.9000, 0.8000, 0.7000", \ 
                    "2.0370, 1.7380, 1.6560, 1.4500, 1.3100, 1.2000, 1.1000" );
            }
            fall_power(energy_template_7x7) {
                index_1 ("0.0500, 0.1000, 0.4000, 0.9000, 1.4000, 2.0000, 3.0000");
                index_2 ("0.00720, 0.36000, 0.72000, 1.80000, 3.24000, 4.68000, 6.12000");
                values ( \
                    "0.5327, 0.5127, 0.5144, 0.5183, 0.5210, 0.5206, 0.5214", \ 
                    "0.5316, 0.5257, 0.5306, 0.5323, 0.5345, 0.5340, 0.5347", \ 
                    "0.6900, 0.6210, 0.6211, 0.6175, 0.6171, 0.6170, 0.6170", \ 
                    "0.9725, 0.8399, 0.8181, 0.8046, 0.8011, 0.8000, 0.7990", \ 
                    "1.2410, 1.0690, 1.0310, 1.0020, 0.9948, 0.9915, 0.9894" );
            }
        }
    }
}
```
"1.5680, 1.3480, 1.2940, 1.2480, 1.2290, 1.2230, 1.2200", 
"2.0890, 1.8190, 1.7440, 1.6680, 1.6380, 1.6240, 1.6170); 
}
timing() {
  related_pin : "A";
}
cell_rise(delay_template_7x7) {
  index_1 ("0.0500, 0.1000, 0.4000, 0.9000, 1.4000, 2.0000, 3.0000");
  index_2 ("0.00720, 0.36000, 0.72000, 1.80000, 3.24000, 4.68000, 6.12000");
  values ( 
    "0.0921, 0.1827, 0.2661, 0.5158, 0.8488, 1.1820, 1.5150", 
    "0.1015, 0.1918, 0.2751, 0.5248, 0.8579, 1.1910, 1.5240", 
    "0.1443, 0.2366, 0.3189, 0.5678, 0.9066, 1.2330, 1.5660", 
    "0.1786, 0.2770, 0.3588, 0.6065, 0.9385, 1.2710, 1.6040", 
    "0.1975, 0.3018, 0.3848, 0.6318, 0.9634, 1.2960, 1.6280", 
    "0.2101, 0.3205, 0.4054, 0.6543, 0.9853, 1.3170, 1.6500", 
    "0.2185, 0.3376, 0.4258, 0.6785, 1.0120, 1.3440, 1.6760");
}

rise_transition(delay_template_7x7) {
  index_1 ("0.0500, 0.1000, 0.4000, 0.9000, 1.4000, 2.0000, 3.0000");
  index_2 ("0.00720, 0.36000, 0.72000, 1.80000, 3.24000, 4.68000, 6.12000");
  values ( 
    "0.0525, 0.2205, 0.4017, 0.9556, 1.6970, 2.4390, 3.1810", 
    "0.0538, 0.2206, 0.4017, 0.9556, 1.6970, 2.4390, 3.1810", 
    "0.0680, 0.2279, 0.4051, 0.9561, 1.6970, 2.4390, 3.1810", 
    "0.0909, 0.2430, 0.4152, 0.9587, 1.6980, 2.4390, 3.1810", 
    "0.1096, 0.2589, 0.4267, 0.9643, 1.7000, 2.4410, 3.1820", 
    "0.1289, 0.2767, 0.4424, 0.9742, 1.7060, 2.4450, 3.1840", 
    "0.1599, 0.3043, 0.4688, 0.9946, 1.7210, 2.4560, 3.1940");
}
cell_fall(delay_template_7x7) {
  index_1 ("0.0500, 0.1000, 0.4000, 0.9000, 1.4000, 2.0000, 3.0000");
  index_2 ("0.00720, 0.36000, 0.72000, 1.80000, 3.24000, 4.68000, 6.12000");
  values ( 
    "0.1002, 0.1991, 0.2840, 0.5357, 0.8710, 1.2060, 1.5410", 
    "0.1121, 0.2108, 0.2957, 0.5475, 0.8827, 1.2180, 1.5530", 
    "0.1754, 0.2755, 0.3600, 0.6111, 0.9462, 1.2810, 1.6160", 
    "0.2481, 0.3524, 0.4372, 0.6878, 1.0220, 1.3570, 1.6920", 
    "0.3056, 0.4152, 0.5013, 0.7514, 1.0860, 1.4200, 1.7550", 
    "0.3641, 0.4792, 0.5686, 0.8204, 1.1540, 1.4890, 1.8230", 
    "0.4477, 0.5708, 0.6646, 0.9237, 1.2590, 1.5930, 1.9270");
}
fall_transition(delay_template_7x7) {
  index_1 ("0.0500, 0.1000, 0.4000, 0.9000, 1.4000, 2.0000, 3.0000");
  index_2 ("0.00720, 0.36000, 0.72000, 1.80000, 3.24000, 4.68000, 6.12000");
  values ( 
    "0.0459, 0.1905, 0.3397, 0.8015, 1.4230, 2.0460, 2.6680", 
    "0.0464, 0.1906, 0.3395, 0.8014, 1.4230, 2.0460, 2.6680", 
    "0.0598, 0.1983, 0.3441, 0.8023, 1.4230, 2.0460, 2.6680", 
    "0.0811, 0.2134, 0.3555, 0.8063, 1.4250, 2.0460, 2.6690", 
    "0.1002, 0.2314, 0.3682, 0.8126, 1.4270, 2.0480, 2.6690", 
    "0.1201, 0.2526, 0.3871, 0.8247, 1.4340, 2.0510, 2.6710", 
    "0.1502, 0.2841, 0.4189, 0.8497, 1.4510, 2.0640, 2.6810");
}

cell (DFFX1) {
  cell_footprint : dff;
  area : 97.9200;
  pin(CK) {
    direction : input;
    capacitance : 0.00421;
  }
}
The template of the look-up-table is often a two-dimension table, where the first index denotes the input net transition time and the second index denotes the total output net capacitance. The total output net capacitance includes the net capacitance and the summation of the input pin capacitances of driven cells. Assume that a CLKBUFX12 buffer drives two DFFX1 and its output net capacitance is 0.01pf. Then, the total output net capacitance is 0.01842pf (0.01pf + 2 * 0.00421pf). Both indexes have 7 elements.

The cell_rise and cell_fall groups define cell’s rising delay and falling delay of the output pin that related to an input pin; the rise_transition and fall_transition groups define cell’s output pin transition time; the rise_power and fall_power groups define cell’s internal power. If the input net transition and/or the output net capacitance do not exist in these indexes, the program should use the interpolation method to calculate the timing. For the details of Synopsys timing/power model format, please refer to Synopsys Library Compiler Manual.

In this contest, the inverters are also allowed to be used as clock buffers for the construction of the clock tree. However, note that all flip-flops are positive-edge-triggered flip-flops.

(3). The clock tree information

The clock tree information includes the input clock transition time and the maximum clock skew. Notice that each clock buffer in the clock tree has to satisfy the DRC constraint (i.e. the input net transition and the output net capacitance can not be out of the index range in .lib file).

(4). Net capacitance calculation

The program doesn’t need to perform real routing but it must use the given formula below to estimate the net capacitance for calculating the delay of clock buffers. The formula to estimate the net load C of a driver pin is

\[
C = \sum_{all\_fanout} \text{net\_length} \times \phi
\]

\[
\phi = 0.000076
\]

where the net_length is the net length from driving cell to the driven cell, its unit is in \(\mu\text{m}\) and
its formula is

\[ \text{net}\_\text{length} = |X_{\text{cell1}} - X_{\text{cell2}}| + |Y_{\text{cell1}} - Y_{\text{cell2}}| \]

where the (X,Y) is the coordinates of a cell in X-axis and Y-axis, respectively. \( \phi \) is a factor and its value is 0.000076 pF/um.

4. Output

Let’s use Figure 1 as an example, the program should output the files as below:

(1). Design file with optimized clock tree (design_opt.def)

The format of output design file should be identical to that of input design file. The lower-left coordinate and the upper-right coordinate should be the same as those of input design file. The output design file should contain entire design including optimized clock tree. All placed standard cells except the clock tree cannot be changed, relocated, or removed. In addition, the root of the clock tree (i.e., Cell C1 in Figure 1) also cannot be changed, relocated, or removed. The clock buffers are not allowed to overlap with flip-flops or other clock buffers. Note that the shape of each cell is assumed to be square.

(2). Gate-level verilog netlist of the clock tree (clock_tree.v)

```
module test(CLK);
    input CLK;
    CLKBUFFX12 C1 (.A(CLK),.Y(W1));
    CLKBUFFX1 C2 (.A(W1),.Y(W2));
    CLKBUFFX2 C3 (.A(W1),.Y(W3));
    DFFX1 R1 (.CK(W2));
    DFFX1 R2 (.CK(W3));
endmodule
```

(3). Set load file (net_load.rpt)

The total net capacitance file contains capacitance of all clock nets. The capacitance unit is in pF.

```
# set_load -subtract_pin_load capacitance net_name
set_load -subtract_pin_load 0.031753 W1
set_load -subtract_pin_load 0.031061 W2
set_load -subtract_pin_load 0.022285 W3
```

(4). Skew report (skew.rpt) and cell area report of the modified design (cell_area.rpt)

The clock skew of the modified design shall be within the given constraint. Clock skew
calculation is the same as the previous example. Cell area report is the summation of each cell area (shown in the .lib file).

<table>
<thead>
<tr>
<th># Flip-Flop</th>
<th>Clock latency(ns)</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>0.36</td>
</tr>
<tr>
<td>R2</td>
<td>0.28</td>
</tr>
</tbody>
</table>

The maximum clock latency is 0.36

<table>
<thead>
<tr>
<th># Maximum clock skew</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1 R2</td>
</tr>
<tr>
<td>0.08</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th># cell_names</th>
<th>cell number</th>
<th>cell area</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLKBUFX12</td>
<td>1</td>
<td>86.4</td>
</tr>
<tr>
<td>CLKBUFX1</td>
<td>1</td>
<td>23.04</td>
</tr>
<tr>
<td>CLKBUFX2</td>
<td>1</td>
<td>23.04</td>
</tr>
<tr>
<td>DFFX1</td>
<td>2</td>
<td>195.84</td>
</tr>
</tbody>
</table>

# total cell number | total cell area
5                  | 328.32

(5). Power report (power.rpt)

<table>
<thead>
<tr>
<th># Net</th>
<th>Switching power(mW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLK</td>
<td>0.0162</td>
</tr>
<tr>
<td>W1</td>
<td>0.0198</td>
</tr>
<tr>
<td>W2</td>
<td>0.0194</td>
</tr>
<tr>
<td>W3</td>
<td>0.0139</td>
</tr>
</tbody>
</table>

The total switching power is 0.0693 mW

<table>
<thead>
<tr>
<th># Cell</th>
<th>Internal power(mW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>0.0950</td>
</tr>
<tr>
<td>C2</td>
<td>0.0073</td>
</tr>
<tr>
<td>C3</td>
<td>0.0123</td>
</tr>
<tr>
<td>R1</td>
<td>0.0205</td>
</tr>
<tr>
<td>R2</td>
<td>0.0194</td>
</tr>
</tbody>
</table>

The total internal power is 0.1552 mW

The total dynamic power is 0.2245 mW

The primary goal of this program is to minimize the dynamic power. Dynamic power includes switching power and dynamic power. The switching power of net N = net load of N* (supply voltage)^2 * working frequency.
The internal power of each gate is defined in the Synopsys power model. Both rise_power and fall_power have to be considered in a clock cycle.

Figure 2

Figure 2 denotes each net capacitance and rise/fall transition time after calculation. Let the clock frequency, supply voltage, and input transition time of clock root be 100MHz, 2.5V, and 0.1ns respectively. We can get the result below:

The switching power of net “CLK”
\[ = 0.02588 \times 10^{-12} \times (2.5)^2 \times 100 \times 10^6 = 0.0162 \, \text{mW} \]

The switching power of net “W1”
\[ = 0.031753 \times 10^{-12} \times (2.5)^2 \times 100 \times 10^6 = 0.0198 \, \text{mW} \]

The constant (A,B,C,D) has to follow the equations below to do interpolation of internal power.

The internal power of cell “C1”
\[ = (\text{Rise}_\text{power} + \text{Fall}_\text{power}) \times 10^{-12} \times 100 \times 10^6 = 0.0950 \, \text{mW} \]

For rising edge:

\[ 0.3816 = A + 0.05 \times B + 0.0072 \times C + 0.05 \times 0.0072 \times D \]
\[ 0.4223 = A + 0.10 \times B + 0.0072 \times C + 0.10 \times 0.0072 \times D \]
\[ 0.3020 = A + 0.05 \times B + 0.36 \times C + 0.05 \times 0.36 \times D \]
\[ 0.3770 = A + 0.10 \times B + 0.36 \times C + 0.10 \times 0.36 \times D \]
\[ \text{Rise}_\text{power} = A + 0.10 \times B + 0.031753 \times C + 0.10 \times 0.031753 \times D \]

For falling edge:
0.5327 = A + 0.05*B + 0.0072*C + 0.05*0.0072*D  
0.5316 = A + 0.10*B + 0.0072*C + 0.10*0.0072*D  
0.5127 = A + 0.05*B + 0.36*C + 0.05*0.36*D  
0.5257 = A + 0.10*B + 0.36*C + 0.10*0.36*D  
Fall_power = A + 0.10*B + 0.031753*C + 0.10*0.031753*D

The internal power of cell “R1”

= (Rise_power + Fall_power)*10^{-12}*100*10^6 = 0.0205 mW

For rising edge:
Rise_power = 0.0875 + (0.0954 - 0.0875)*(0.37 - 0.1)/(0.4 - 0.1)

For falling edge:
Fall_power = 0.1029 + (0.1129 - 0.1029)*(0.31 - 0.1)/(0.4 - 0.1)

5. Language/Platform

- Language: C or C++
- Platform: SUN OS/Solaris

6. Evaluation

Requirements:
1. All output files are correct.
2. The clock skew cannot exceed the maximum clock skew constraint.
3. The new clock tree has no DRC violation.
4. All placed standard cells except the clock tree cannot be changed, relocated, or removed.
5. The root of the clock tree cannot be changed, relocated, or removed.
6. The clock buffers are not allowed to overlap with flip-flops or other clock buffers.

Evaluation priorities:
- Total dynamic power
- Run time
- Total cell area
- Maximum clock latency
- Practical issue

The final results will be compared with the report from Synopsys Power Compiler. The verilog netlist and set load file will be feed into Power Compiler to calculate the total power.