2024集成电路EDA设计精英挑战赛

赛题指南

一、赛题名称

数字电路时钟树综合

二、命题单位

芯行纪科技有限公司

三、赛题主席

李兴权（鹏城实验室）

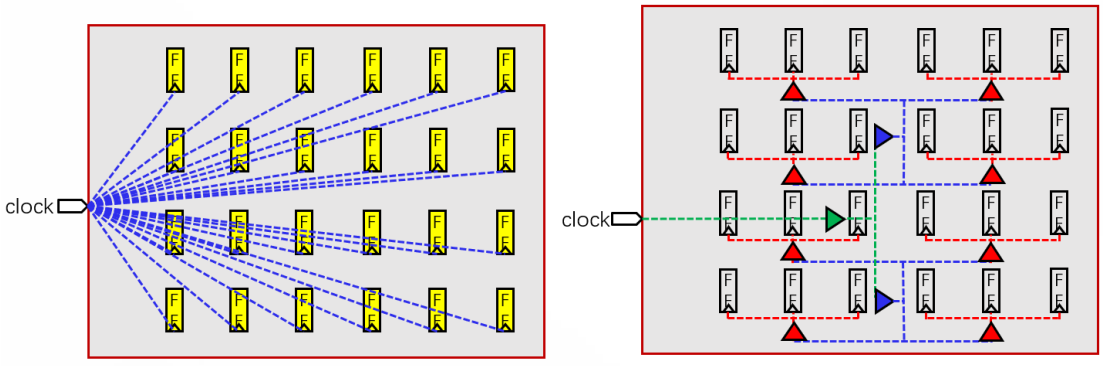
四、赛题背景

在数字集成电路中，时钟信号是数据传输的基准，它对于同步数字系统的功能、性能和稳定性起决定性作用，所以时钟信号的特性及其分配网络尤被人们关注。时钟信号通常是整个芯片中有最大扇出、通过最长距离、以最高速度运行的信号。时钟信号必须保证在最差的条件下，关键的时序要求能得到满足，否则对时钟信号任何不当的控制都可能导致紊乱情况，将错误的数据信号锁存到寄存器，从而导致系统功能的错误。

在现有基于标准单元库的ASIC流程中，通常将扇出较大的时钟网络在综合阶段设置为理想时钟，在物理设计阶段进行时钟树综合（clock tree synthesis, CTS），因此它是后

端物理设计的关键步骤之一。

时钟信号在物理设计中的实现结果被形象地称之为时钟树，时钟信号的起点叫做根节点（root pin 或 root cell），时钟信号经过一系列分布节点最终达到寄存器时钟输入端或其他时钟终点（例如，存储器时钟输入端）被称为叶节点。根节点、分布节点和叶节点都依附于的逻辑单元则分别称作根单元、分布单元和叶单元。时钟网络从根节点逐级插入驱动器（buffer、inverter），从而到达其叶节点，按照芯片时钟网络的约束要求产生时钟树的过程叫做时钟树综合。即生成从时钟源（root 或者 source）到所有寄存器（FF）的时钟端口（sink pin）的时钟树。



original　　After CTS

时钟树根据其在芯片内的分布特征，可分为多种结构，主要有 H 树（H-tree）、X 树（X-tree）、平衡树（balanced tree），以及梳状或脊椎状时钟网（clock tree mesh： comb or spine）。

H-tree 从中心点到达各个叶节点的距离是相等的，因此

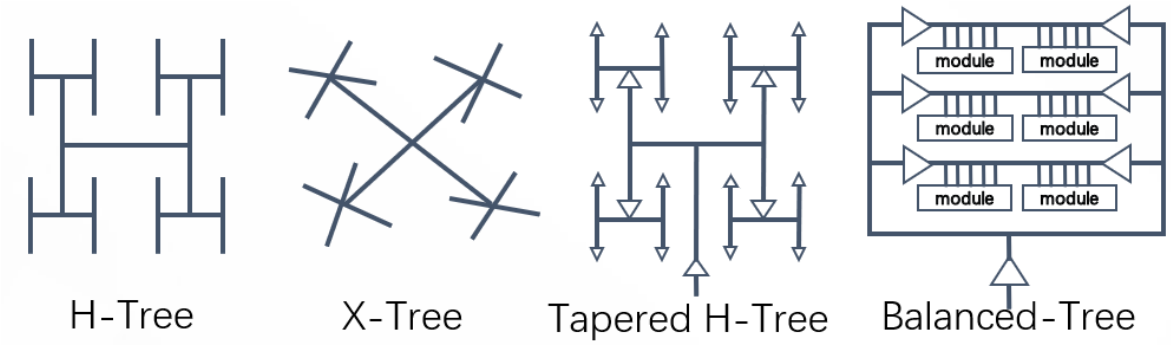
信号到达叶节点的时间理论上相等，时钟偏差理论值为 0，该结构的不足是布线比较困难，扇出难以协调一致，适用于较小的设计。

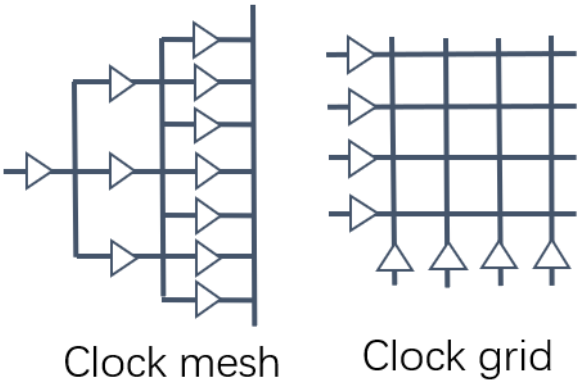
X-tree 也是一种实现等长互连线的方法，与 H-tree 相比，X-tree 采用了很多非 90°的互连线，与 H-tree 的扇出为 2 相比，X-tree 的扇出是 4。

为了减小叶节点反射电流的影响，将分支后的时钟网线的电阻提高到分支前的 2 倍，也即分支后的线宽为分支前的 1/2，得到“裁剪状的”（tapered）H-tree。

平衡树采用的是两个层次的驱动，比较适用于分层次的时钟树设计以及较大的时钟树综合。在顶层上，它采用一个层次的驱动插入以平衡时钟偏差，到模块级再采用模块级的驱动插入平衡时钟偏差。从图中可以看出从时钟的根节点通过一级驱动到达模块的时间是不同的，存在时钟偏差，但是比较小。

对于很大的时钟树，用时钟网络或时钟网格（clock mesh, clock grid）的方法是获取较小时钟偏差较好实用方案，如图所示。尽管当代的 EDA 工具能自动生成时钟网络或网格等拓扑结构，但它还是需要很丰富的经验和细致的手动方法去设计并作调整。





网状类时钟结构

五、赛题解析

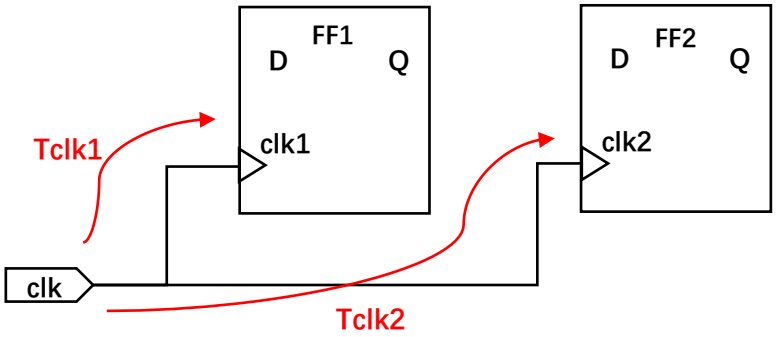
时钟树综合（CTS）主要分为两个步骤：cluster 和 balance。cluster 是构建 clock tree 的基本结构，balance 则是对 clock tree 进行进一步的优化，都通过插 buffer 来实现。优化指标主要包括时钟延迟（clock latency）、时钟偏移（clock skew）以及驱动单元数量（buffer count），这也是本赛题主要考察的三大指标。

clock latency 为 clock root 点到所有寄存器（FF）的时钟延时。clock skew 是指到两个 FF 的 clock latency 的差值。如下图所示，FF 数量为 2，到两个寄存器的 clock latency 分别

为 Tclk1 和 Tclk2。

$$T_{avg} = (T_{clk1} + T_{clk2})/2 \quad \text{(average clock latency)}$$

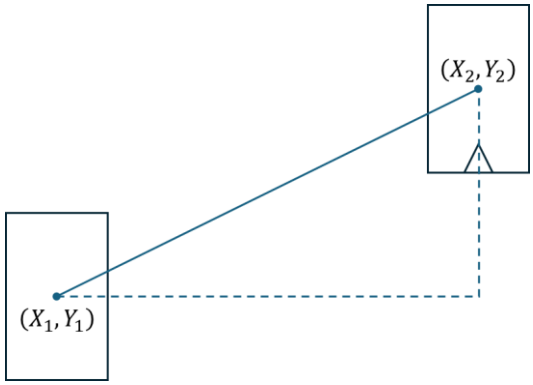
$$T_{skew} = T_{clk2} - T_{clk1} \quad \text{(clock skew)}$$



Net delay 模型：

a) Manhattan 距离：定义两个 instance 间的 Manhattan 距离为两个 instance 中心点的沿 X 和 Y 方向的距离之和，即

$$D_{1,2} = |X_1 - X_2| + |Y_1 - Y_2|$$



b) Unit rc，输入或输入文件定义。包括单位长度 net 电阻值 r 与电容值 c，其中 r 单位为 Ω/um，c 单位为pF/um

c) Fanout：对于每一条 net，从 net driver 到所有 sink 或 buffer 的数量即为这条 net 的 fanout 值，下图中从 driver instance 1

出来的 net fanout 为 2。

基于 Manhattan 距离的定义，以及 unit rc，对于每一条 net，可以定义从 net driver 到每一个 sink 的 net rc 值为：$rc_{1,2} = \frac{1}{2} \times r \times c \times D_{1,2} \times D_{1,2}$，如：

Driver Instance 1 到 sink Instance 2 的 net rc：

$$rc_{1,2} = \frac{1}{2} rcD_{1,2}^2$$

Driver Instance 1 到 sink Instance 3 的 net rc：

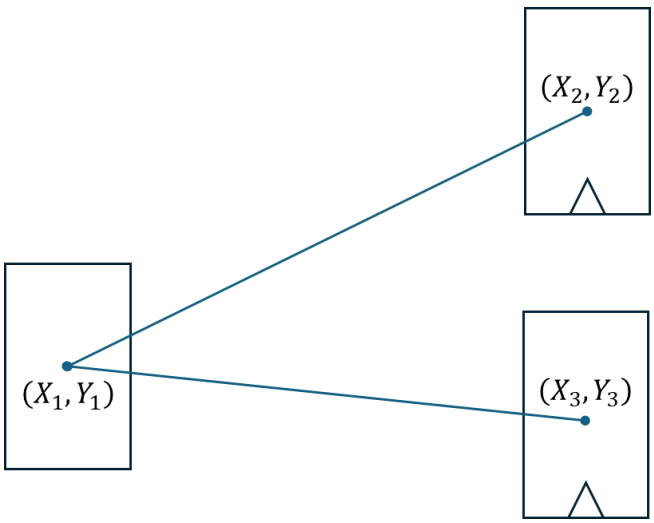
$$rc_{1,3} = \frac{1}{2} rcD_{1,3}^2$$

Driver 驱动的总 rc 为

$$\frac{1}{2} rcD_{1,2}^2 + \frac{1}{2} rcD_{1,3}^2$$

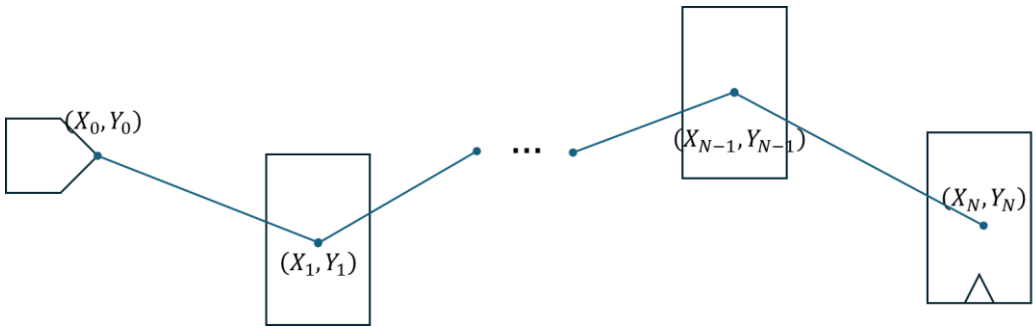
b) Net delay 和 net rc 成正比，单位为 ps：

$$net\_delay_{1,2} = 0.69 \times rc_{1,2}$$



Latency 和 skew：

1. 从 clock root 到每个 FF ck pin 的 latency（延时）可以通过公式计算：



$$latency_N = (N-1) \times buf\_delay + \sum_{k=0}^{N-1} net\_delay_{k,k+1},$$

其中buf_delay为给定的 buffer cell delay，单位为ps.

Clock local skew 定义为 clock tree 上任两个 FF ck pin 之间的 latency 之差：

$$skew_{k,l} = |latency_k - latency_l|$$

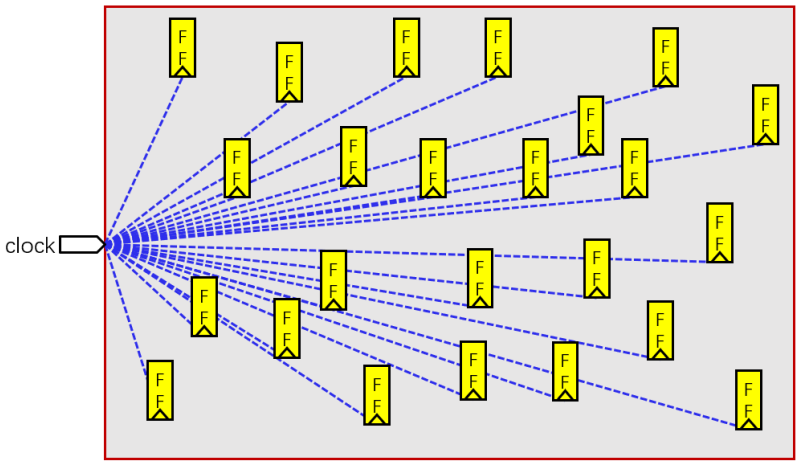
Clock global skew 定义为 clock tree 上最大的 local skew，即

$$skew = \max_{k,l} skew_{k,l}$$

六、赛题描述

赛题输入：包含数量 100K~200K 不均匀分布的 FF 的电路文本文件。

所有的 FF 均为同种类型尺寸，不均匀地摆放在 floorplan 区域内，clock root 点直连到所有的 FF，本赛题仅提供一种 buffer cell 用于时钟树综合。



文件格式：包含 unit 单位，floorplan 大小，clock root 点坐标，FF cell 和 buffer cell 尺寸，所有的 FF instance 坐标。

```
UNITS DISTANCE MICRONS 1000 ;  #文件中的数值单位为 1nm
DIEAREA ( 0 0 ) ( 0 y ) ( x y ) ( x 0 ) ;  #floorplan 区域为{0 0 x y}
FF ( a b ) ;                   #寄存器 cell FF size 为 a*b
BUF ( c d ) ;                  #buffer cell BUF size 为 c*d
CLK ( e f ) ;                  #clock root 点 CLK 坐标为（e f）
COMPONENTS N ;                 #instance 总数为 N，在输入文件中即 FF 总数
- FF1 FF ( x1 y1 ) ;           #instance FF1 cell 类型为 FF，坐标为（x1 y1）
- FF2 FF ( x2 y2 ) ;           #所有 instance 坐标均为 cell 左下角坐标
- FF3 FF ( x3 y3 ) ;           #所有 instance 方向默认一致，不考虑翻转旋转
.....
END COMPONENTS
```

其他输入：unit rc，max rc，max fanout，buffer delay，net delay 模型。

赛题输出：输出文件格式需与输入文件一致，需包含 instance 之间的完整连接关系。

输出格式：net 连接关系部分

```
NETS NUM ;                      #net 总数为 N
- net_1 ( CLK ) ( BUF1 ) ;      #net name 为 net_1, 从 CLK output 到 BUF1
- net_2 ( BUF1 ) ( BUF2 BUF3) ; #net net_2 从 BUF1 output 到 BUF2 & BUF3
- net_3 ( BUF4 ) ( BUF5 ) ;     #同一 output driver 出来的 net 为同一 net
……                             #net 和 instance 命名方式可自定义
- net_NUM ( BUFn1 ) ( FFn2 );   #不同 driver 出来的 net 命名不可重复
END NETS
```

答题要求：

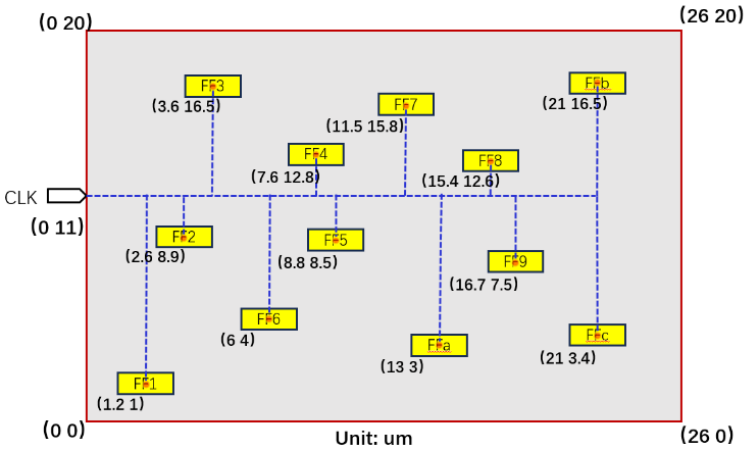
1. 输出文件格式需正确，可通过提供的 checker 程序检查；
2. Checker 程序读入正确的输出文件后输出该时钟树综合后电路的 average latency，global skew 和 buffer count；
3. 要求所有的 instance 之间不能有 overlap;
4. 要求所有的 instance 区域不能超过 floorplan 区域；
5. 走线距离计算采取 Manhattan 距离，不分 layer；
6. Buffer drive 数量不能超过 max fanout, buffer drive RC 不能超过 max RC；
7. 赛题对 runtime 有一定要求，允许使用多线程，最多可用 8 个线程。

七、赛题示例

输入文件：

```
UNITS DISTANCE MICRONS 1000 ;
DIEAREA ( 0 0 ) ( 0 20000 ) ( 26000 20000 )
( 26000 0 ) ;
FF ( 2000 1000 ) ;
BUF ( 1000 1000 ) ;
CLK ( 0 11000 ) ;
COMPONENTS 12 ;
- FF1 FF ( 1200 1000 ) ;
- FF2 FF ( 2600 8900 ) ;
- FF3 FF ( 3600 16500 ) ;
- FF4 FF ( 7600 12800 ) ;
- FF5 FF ( 8800 8500 ) ;
- FF6 FF ( 6000 4000 ) ;
- FF7 FF ( 11500 15800 ) ;
- FF8 FF ( 15400 12600 ) ;
- FF9 FF ( 16700 7500 ) ;
- FFa FF ( 13000 3000 ) ;
- FFb FF ( 21000 16500 ) ;
- FFc FF ( 21000 3400 ) ;
END COMPONENTS
```

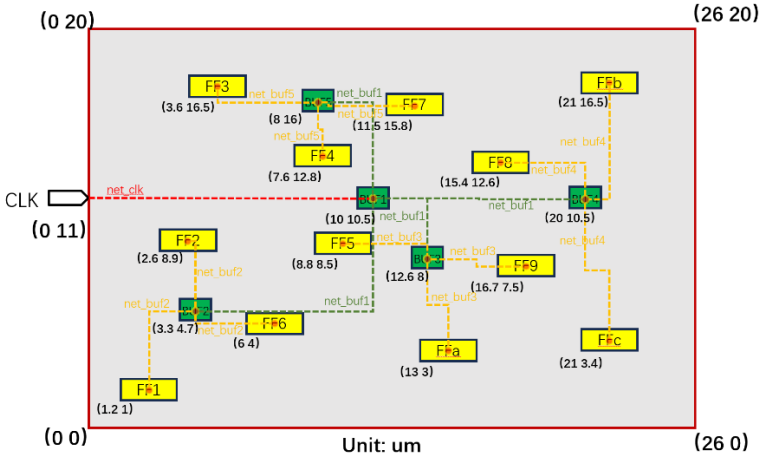
输入文件图解：



输出文件:

```
UNITS DISTANCE MICRONS 1000 ;
DIEAREA ( 0 0 ) ( 0 20000 ) ( 26000 20000 )
( 26000 0 ) ;
FF ( 2000 1000 ) ;
BUF ( 1000 1000 ) ;
CLK ( 0 11000 ) ;
COMPONENTS 17 ;
- FF1 FF ( 1200 1000 ) ;
- FF2 FF ( 2600 8900 ) ;
- FF3 FF ( 3600 16500 ) ;
- FF4 FF ( 7600 12800 ) ;
- FF5 FF ( 8800 8500 ) ;
- FF6 FF ( 6000 4000 ) ;
- FF7 FF ( 11500 15800 ) ;
- FF8 FF ( 15400 12600 ) ;
- FF9 FF ( 16700 7500 ) ;
- FFa FF ( 13000 3000 ) ;
- FFb FF ( 21000 16500 ) ;
- FFc FF ( 21000 3400 ) ;
- BUF1 BUF ( 10000 10500 ) ;
- BUF2 BUF ( 3300 4700 ) ;
- BUF3 BUF ( 12600 8000 ) ;
- BUF4 BUF ( 20000 10500) ;
- BUF5 BUF ( 8000 16000) ;
END COMPONENTS
NETS 6 ;
- net_clk ( CLK ) ( BUF1 ) ;
- net_buf1 ( BUF1 ) ( BUF2 BUF3 BUF4 BUF5) ;
- net_buf2 ( BUF2 ) ( FF1 FF2 FF6) ;
- net_buf3 ( BUF3 ) ( FF5 FF9 FFa ) ;
- net_buf4 ( BUF4 ) ( FF8 FFb FFc) ;
- net_buf5 ( BUF5 ) ( FF3 FF4 FF7) ;
END NETS
```

输出文件图解：



计算过程：

Buffer count: 17-12=5

latency 计算：例 CLK 到 FF1，因为文件坐标为 instance

左下角，Manhattan 距离需计算 instance 中心点，因此在计算距离时需要加上 1/2 cell size 的距离。

$\text{net\_delay}_1 = \text{net\_delay}_{CLK,BUF1} + \text{net\_delay}_{BUF1,BUF2} + \text{net\_delay}_{BUF2,FF1}$

$= 0.69 \times (rc_{CLK,BUF1} + rc_{BUF1,BUF2} + rc_{BUF2,FF1})$

$= 0.69 \times \frac{1}{2} rc(D^2_{CLK,BUF1} + D^2_{BUF1,BUF2} + D^2_{BUF2,FF1})$

$= 0.345rc \times [(|10 + 0.5 - 0| + |10.5 + 0.5 - 11|)^2 + (|3.3 - 10| + |4.7 - 10.5|)^2 + (|1.2 + 1 - 3.3 - 0.5| + |1 + 0.5 - 4.7 - 0.5|)^2]$

$= 0.345 \times 2 \times 12 \times 294.59$

$= 2439.2052$ ps

$\text{Latency}_1 = \text{net\_delay}_1 + 2 \times \text{buf\_delay} = 2439.2052 + 2 \times 100 = 2639.2052$ ps

同理可得其他 Latency 分别为 2566.9208(FF2), 2532.5588(FF6), 1342.6728(FF5), 1354.2428(FF9), 1616.4596(FFa), 2259.1532(FF8), 2406.62(FFb), 2553.2588(FFc), 1738.9208(FF3), 1668.7892(FF4), 1724.6792(FF7) ps

Average latency:

$\text{Latencyavg} = (\sum_{k=1}^{12} latency_k)/12 = 2033.6234$ ps

Global skew:

$$skew_{global} = latency_{max} - latency_{min}$$
$$= 2639.2052 - 1342.6728 = 1296.5324\text{ps}$$

最后计算一下每条 net drive 的 rc 有没有超过 max_rc 的要求：

$$\text{RC}_{\text{net\_clk}} = \frac{1}{2}rc \times 10.5^2 = 1323ps$$

$$\text{RC}_{\text{net\_buf1}} = \frac{1}{2}rc \times (12.5^2 + 5.1^2 + 10^2 + 7.5^2) = 4062.12ps$$

$$\text{RC}_{\text{net\_buf2}} = \frac{1}{2}rc \times (5.3^2 + 4.4^2 + 3.9^2) = 751.92ps$$

$$\text{RC}_{\text{net\_buf3}} = \frac{1}{2}rc \times (3.8^2 + 5.1^2 + 5.9^2) = 903.12ps$$

$$\text{RC}_{\text{net\_buf4}} = \frac{1}{2}rc \times (6.2^2 + 7.5^2 + 8.6^2) = 2023.8ps$$

$$\text{RC}_{\text{net\_buf5}} = \frac{1}{2}rc \times (4.4^2 + 3.3^2 + 4.2^2) = 574.68ps$$

每条 net drive rc 均未超过 max rc 5000ps 的要求。

八、评分细则

1. 基础分：

基础评分按照三项指标排名，单一指标第一名得 10 分，第二名得 9 分，第三名 8 分，第四名得 7 分，第五名得 6 分，第六到八名得 4 分，第九到十名得 3 分，其余均得 1 分。最后基础分得分为各项指标分加权总和。

若有不止一项指标排名前五，则每多一项基础分加权 1.1；若有任一指标排在十名以后，每有一项基础分加权 0.9。

三项指标为：clock global skew, average clock latency, Buffer count.

计分示例：比如队伍 A 在三项指标中排名为 1、4、11：

则队伍 A 的基础分为（10+7+1）*1.1*0.9=17.82

比如队伍 B 在三项指标中排名为 5、5、7：

则队伍 B 的基础分为（6+6+4）*1.1=17.6

2.扣分细则：

instance 区域超出 floorplan 区域扣 0.5 分。

instance 之间发生 overlap 按照 overlap 区域面积占 instance 面积百分比扣分：

0~3%(含 3%)扣 0.5 分;

3%~10%(含 10%)扣 1 分;

10%~20%(含 20%)扣 2 分;

20%~30%(含 30%)扣 3 分;

超过 30%，成绩无效。

fanout value 超出要求的 max_fanout 按照累计超出百分比扣分：

0~3%(含 3%)扣 0.5 分;

3%~10%(含 10%)扣 1 分;

10%~20%(含 20%)扣 2 分;

20%~30%(含 30%)扣 3 分;

超过 30%，成绩无效。

Runtime 超过赛题要求时长按照超出百分比扣分：

0~3%(含 3%)扣 0.5 分;

3%~10%(含 10%)扣 1 分;

10%~20%(含 20%)扣 2 分;

20%~30%(含 30%)扣 3 分;

超过 30%，成绩无效。

3.附加分：

基础分前十名按照 runtime 从小到大排序，runtime 前三名各加一分。

4.总分：

最终得分按比例 scaling 到百分制。

九、参考资料

在 [1] 中，作者考虑了基于矩形区域切分的方式进行聚类，通过聚类算法来 获得电容负载近似相等的划分方案。文献 [2] 介绍了一种功耗驱动的聚类算法，该算法相比于传统聚类方法，进一步考虑了负载约束和 RC 约束。文献 [3] 中给 出了诸多适用于寄存器的聚类方法，包括考虑扇出约束和负载电容约束的聚类方法，并提出一种基于信号转换时间约束的缓冲器分配方案。文献 [4] 提出了加权聚类方法，其中的单元重分配技术能够有效优化设计质量，并且其考虑了潜在的 扇出约束违例，该方法在功耗层面得到显著优化。在 [5] 中，作者提出广义分支 H 树，并利用动态规划寻找最优的分支情况，在考虑设计约束的情况下，利用线性规划问题进行缓冲器位置的选

取和寄存器聚类结果的优化。文献 [6] 将 CTS 建模为层次化设计问题，并利用解空间探索方法（模拟退火）进行单元划分的进一步优化。

针对该题给出的时序计算方法，时钟偏差约束仍然可以通过较为主流的 deferred-merge embedding (DME) 算法解决，可以根据其衍生的零偏差时钟树 [7,8]（ZST）和有界偏差时钟树 [9]（BST）的问题建模推导该题的时钟偏差控制模型。

在 [10] 中，作者对 DME 算法进行较为细致的阐述，并给出基本的示例和代码。 [11] 中介绍了同步数字集成电路中的时钟分配网络，而 [12] 中介绍了一种用于非对称 VLSI（超大规模集成电路）电路的时钟分配方案。[13] 介绍了通过对称时钟分配树和优化的高速互连来减少 ULSI（超超大规模集成电路）和 WSI（晶圆级集成电路）中的时钟偏差。

参考文献

[1] A. D. Mehta, Y.-P. Chen, N. Menezes, D. Wong, and L. Pilegg, “Clustering and load balancing for buffered clock tree synthesis,” in *Proc. ICCD*. IEEE, 1997, pp. 217–223. 1

[2] R. S. Shelar,“An efficient clustering algorithm for low power clock tree synthesis,”in *Proc. ISPD*, 2007, pp. 181–188.

[3] C. Deng, Y.-C. Cai, and Q. Zhou, “Register clustering methodology for low power clock tree synthesis,” *Journal of Computer Science and Technology*, vol. 30, no. 2, pp. 391–403, 2015.

[4] G. Wu, Y. Xu, D. Wu, M. Ragupathy, Y.-y. Mo, and C. Chu, “Flip-flop clustering by weighted k-means algorithm,” in *Proc. DAC*, 2016, pp. 1–6.

[5] K. Han, A. B. Kahng, and J. Li, “Optimal generalized h-tree topology and buffering for high-performance and low-power clock distribution,” *IEEE Trans. on CAD*, vol. 39, no. 2, pp. 478–491, 2018.

[6] W. Li, Z. Huang, B. Yu, W. Zhu, and X. Li, “Toward controllable hierarchical clock tree synthesis with skew-latency-load tree,” in *Proc. DAC,* 2024.

[7] K. D. Boese and A. B. Kahng, “Zero-skew clock routing trees with minimum wirelength,” *in Proc. of ASICON*, 1992, pp. 17–21.

[8] R.-S. Tsay, “An exact zero-skew clock routing algorithm,” *IEEE Trans. on CAD*, vol. 12, no. 2, pp. 242–249, 1993.

[9] J. Cong, A. B. Kahng, C.-K. Koh, and C.-W. A. Tsao, “Bounded-skew clock and steiner routing,”*ACM Trans. on DAES*, vol. 3, no. 3, pp. 341–388, 1998.

[10] Friedman, E.G. “Clock distribution networks in synchronous digital integrated circuits”, in *Proc. of the IEEE*, 2000.

[11] P Ramanathan, KG Shin, “A clock distribution scheme for nonsymmetric VLSI circuits”, *in Proc. of ICCAD*, 1989.

[12] H. Bakoglu, J. T. Walker and J. D. Meindl, "A Symmetric Clock-Distribution Tree and Optimized High-Speed Interconnections for Reduced Clock Skew in ULSI and WSI Circuits", in *Proc. of ICCD*, 1986, pp. 118-122.