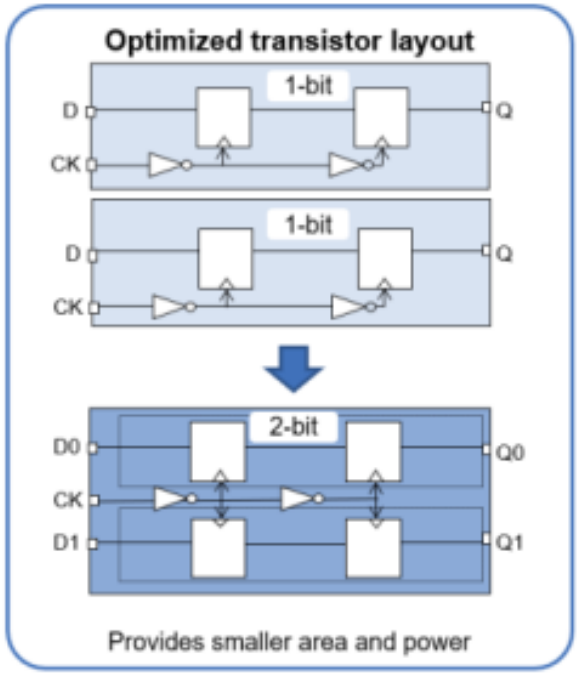
2024 CAD Contest

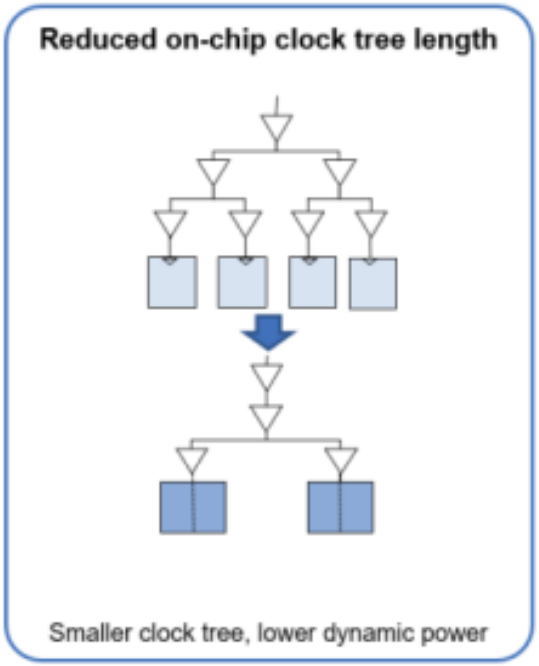
- Course contents:
  - Problem B (recommended)
  - Problem D (recommended)
  - Problem C
  - Problem E
- Reference
  - https://iccad-contest.org/2024/tw/

Problem B by Synopsys
Power and Timing Optimization Using Multibit Flip-Flop

Introduction

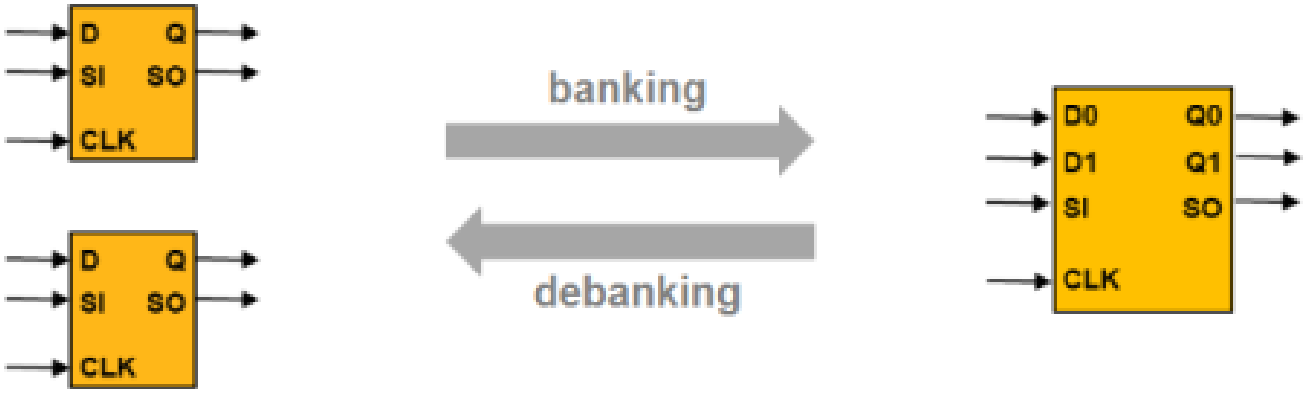
- Multibit flip-flop banking
  - Use one multibit flip-flop to replace multiple single-bit flip-flops.
  - Free up more area as one multibit flip-flop takes less area to place than the single-bit flip-flops it replaces.
  - Efficiently reduce power, ground, and clock net routing complexity.





Introduction (cont'd)

- Multibit flip-flop debanking
  - Multibit-flip-flop banking may worsen circuit timing.
  - Sometimes we have to divide a multibit flip-flop into several single-bit flip-flops to further optimize timing critical nets.



Problem Formulation

- Develop a banking & debanking algorithm to produce a placement result
  - Constraint: cell density constraints
  - Objective: optimize timing, power, and area
- Evaluation metric

$$\sum_{\forall i \in FF} \left(\alpha \cdot TNS(i) + \beta \cdot Power(i) + \gamma \cdot Area(i)\right) + \delta \cdot D$$

  - $TNS(i)$: the total negative slack of the i-th flip-flop FF_i
  - $Power(i)$: the power consumption of FF_i
  - $Area(i)$: the area cost of FF_i
  - D: the number of bins that violates the utilization rate threshold
  - α, β, γ, and δ: the weight for each cost

1	1	5	4	4	3	3	3
1	1	5	5	5	10	10	10
2	2	3	3	8	9	3	3
2	2	3	3	8	9	2	2
2	2	8	10	10	10	2	2
1	5	8	10	10	10	2	2
1	6	7	4	1	5	1	2
1	6	7	4	4	5	1	1

Input Data

Alpha 1

Beta 5

Gamma 5

Delta 1

DieSize 0 0 50 30

NumInput 2

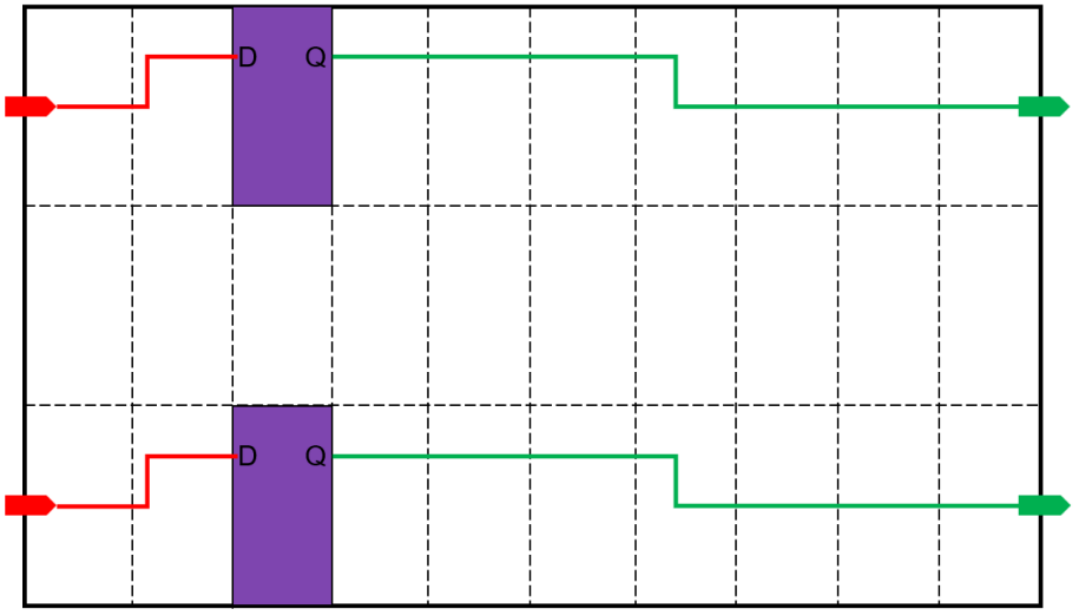
Input INPUT0 0 25

Input INPUT1 0 5

NumOutput 2

Output OUTPUT0 50 25

Output OUTPUT1 50 5



Input Data (cont'd)

FlipFlop <bits> <flipFlopLibCellName> <libCellWidth> <libCellHeight> <pinCount>

Pin <pinName> <pinLocationX> <pinLocationY>

FlipFlop 1 FF1 5 10 2

Pin D 0 8

Pin Q 5 8

FlipFlop 2 FF2 8 10 4

Pin D0 0 9

Pin D1 0 6

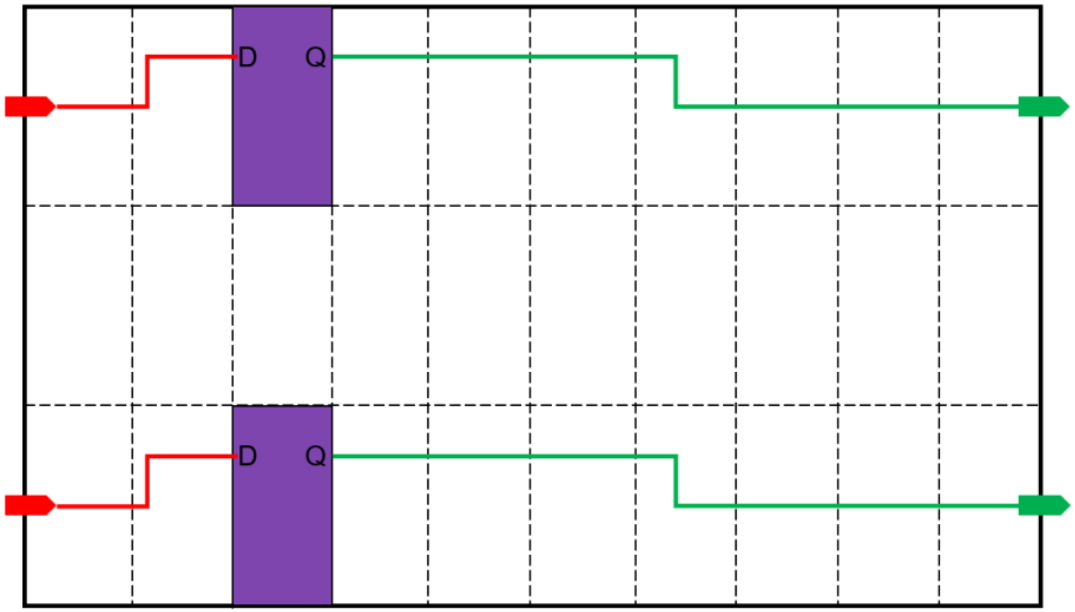
Pin Q0 8 9

Pin Q1 8 6

NumInstances 2

Inst C1 FF1 15 20

Inst C2 FF1 15 0



NumInstances <instanceCount>

Inst <instName> <libCellName> <x-coordinate> <y-coordinate>

Input Data (cont'd)

NumNets 4

Net N1 2

Pin INPUT0

Pin C1/D

Net N2 2

Pin INPUT1

Pin C2/D

Net N3 2

Pin C1/Q

Pin OUTPUT0

Net N4 2

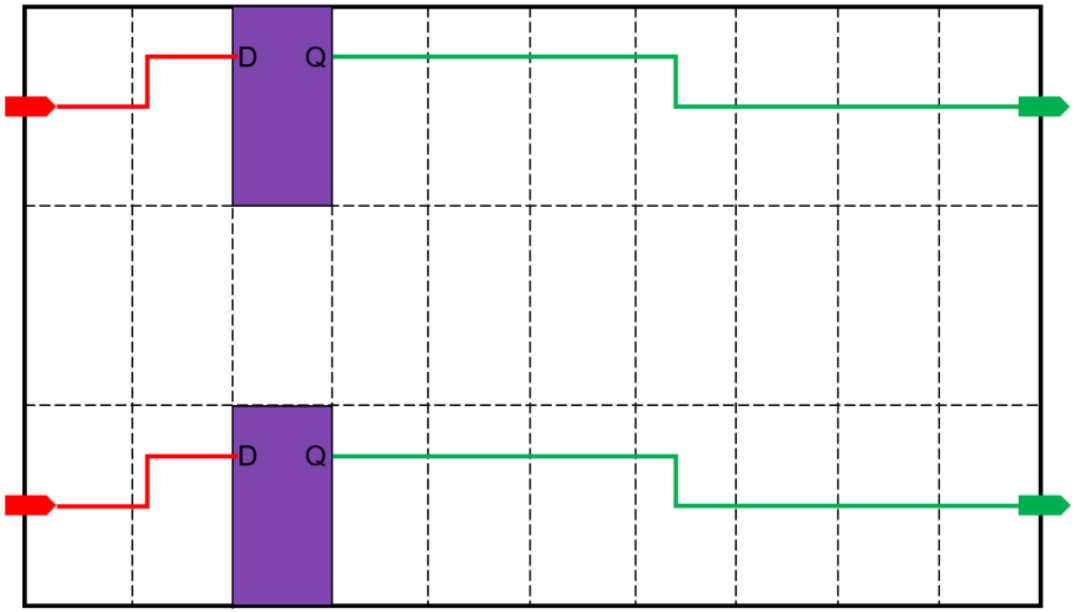
Pin C2/Q

Pin OUTPUT1

NumNets <netCount>

Net <netName> <numPins>

Pin <instName>/<libPinName>



Input Data (cont'd)

- Utilization ratio of a Bin $= \frac{\sum(area\ of\ each\ cell\ on\ the\ bin)}{area\ of\ the\ bin}$

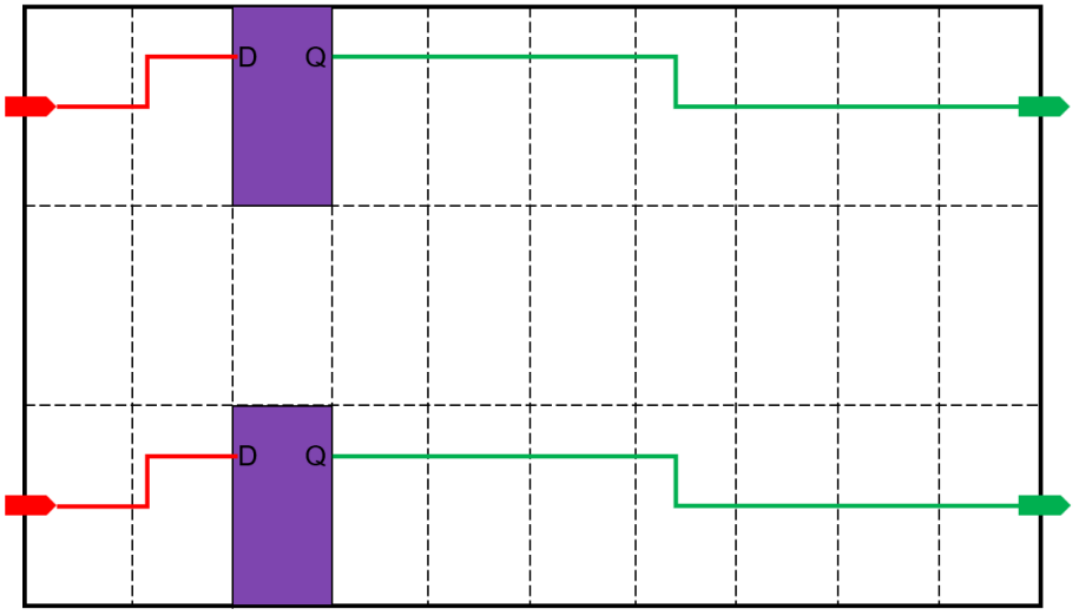
BinWidth 10

BinHeight 10

BinMaxUtil 79

PlacementRows 0 0 2 10

DisplacementDelay 0.01



Input Data (cont'd)

DisplacementDelay <coefficient>

QpinDelay <libCellName> <delay>

TimingSlack <instanceCellName> <PinName> <slack>

DisplacementDelay 0.01

QpinDelay FF1 1

QpinDelay FF2 2

TimingSlack C1 D 1

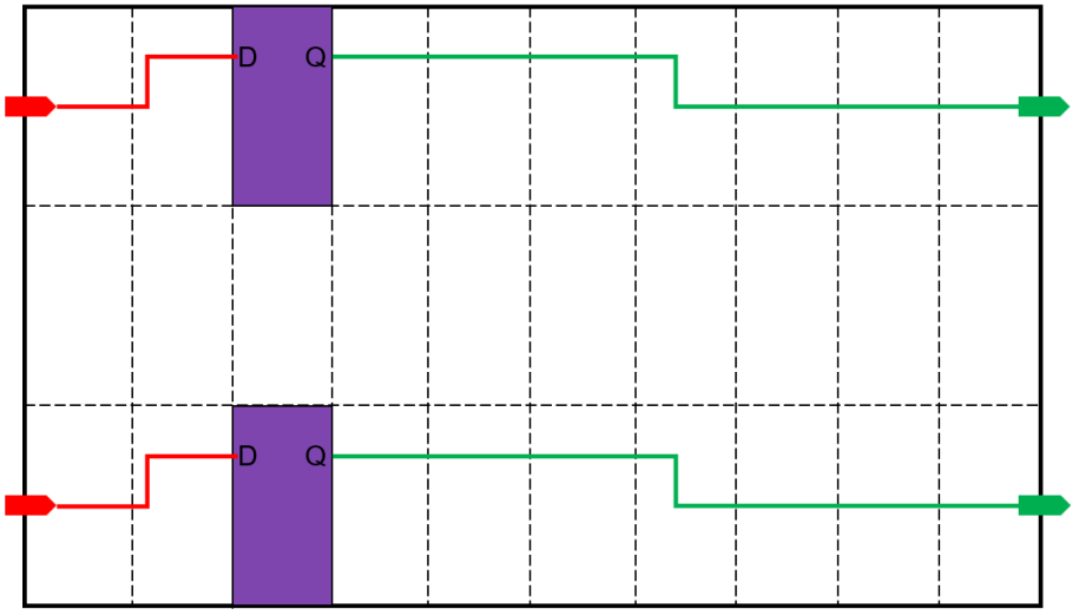
TimingSlack C1 Q 0

TimingSlack C2 D 1

TimingSlack C2 Q 0

GatePower FF1 10

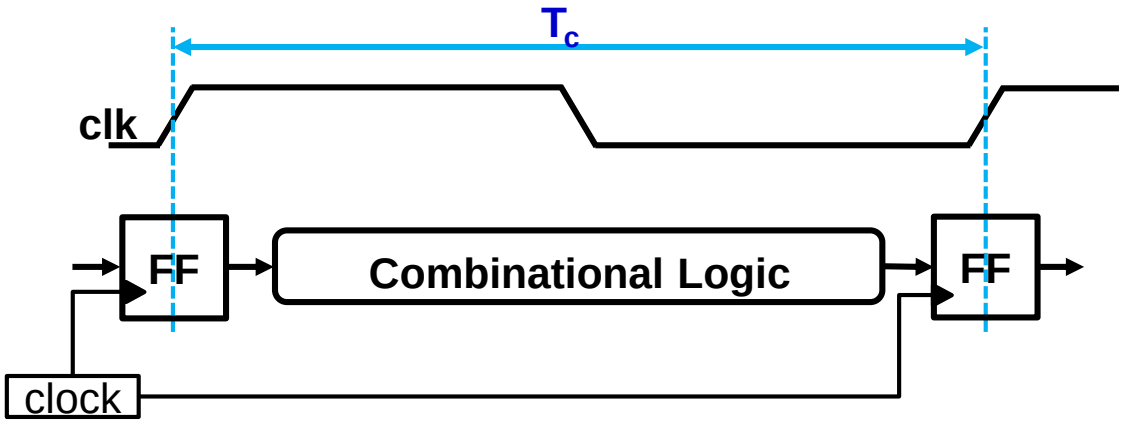
GatePower FF2 17



- $displacement\ delay = Manhatann\ displacement \times coefficient$

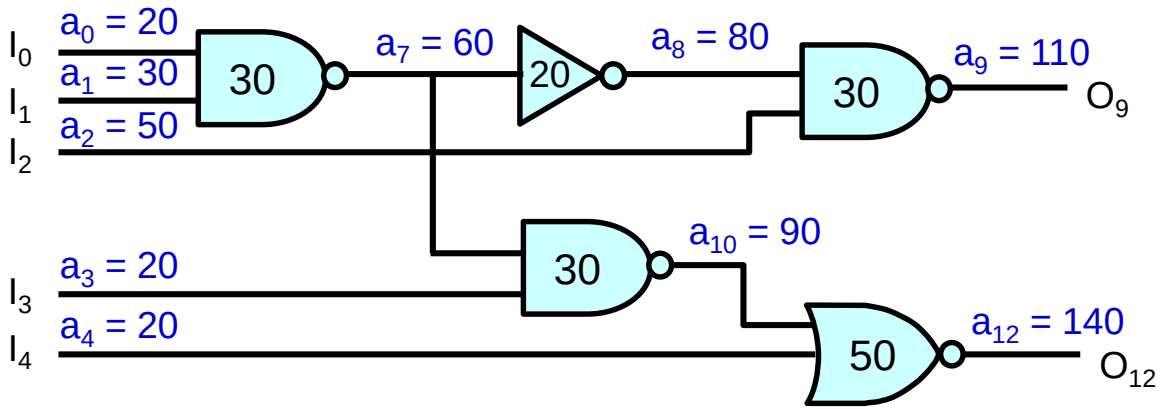
Timing Slack

- Sequential circuit



- *Slack*: the difference between the required and arrival times
  - The slack $slack(v)$ of a node v

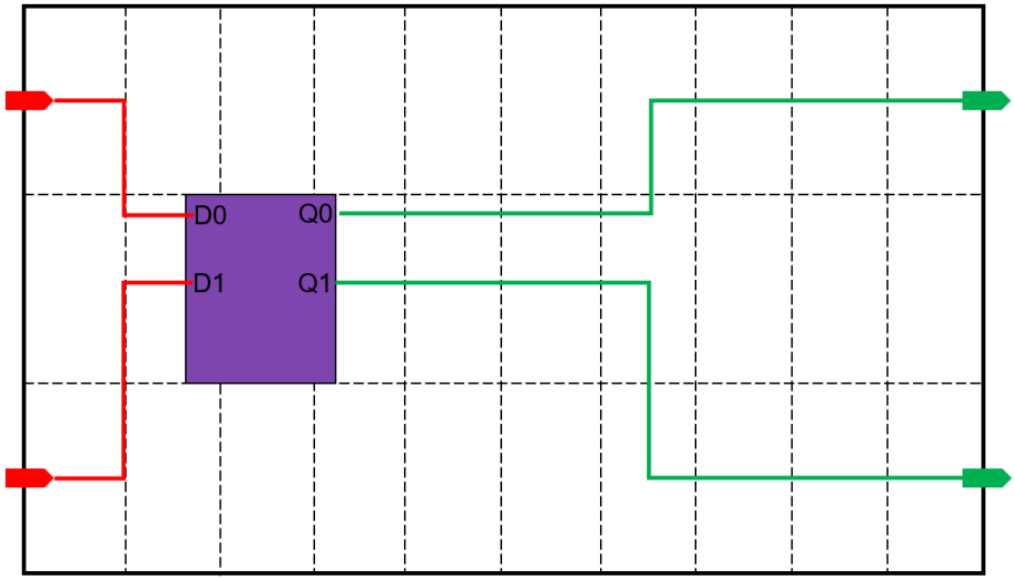
$$slack(v) = RT(v) - AT(v)$$



Output Data

```
CellInst <InstCount>
Inst <instName> <locationX> <locationY> <orientation>
NumNets <netCount>
Net <netName> <numPins>
Pin <instName>/<libPinName>
```

```
CellInst 1
Inst C1 FF2 8 10
NumNets 4
Net N1 2
Pin INPUT0
Pin C1/D0
Net N2 2
Pin INPUT1
Pin C1/D1
Net N3 2
Pin C1/Q0
Pin OUTPUT0
Net N4 2
Pin C1/Q1
Pin OUTPUT1
```



Evaluation and Execution

- Evaluation metric

$$\sum_{\forall i \in FF} \left(\alpha \cdot TNS(i) + \beta \cdot Power(i) + \gamma \cdot Area(i)\right) + \delta \cdot D$$

  - $TNS(i)$: the total negative slack of the i-th flip-flop FF_i
  - $Power(i)$: the power consumption of FF_i
  - $Area(i)$: the area cost of FF_i
  - D: the number of bins that violates the utilization rate threshold
  - α, β, γ, and δ: the weight for each cost
- Runtime limit: 60 minutes
- Command-line parameter:

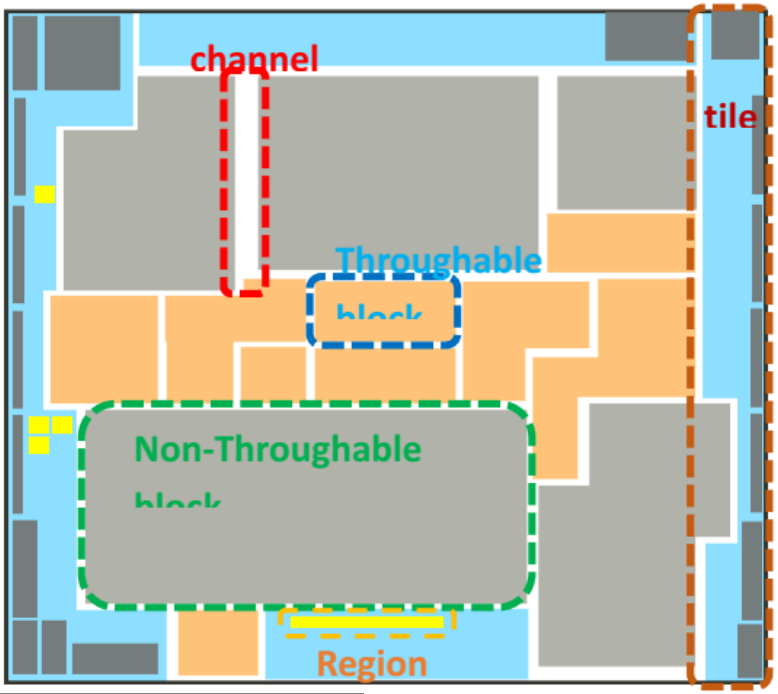
./$binary_name <input.txt> <output.txt>

  - The rule for binary name will be given out in the contest website
  - Upload your binary to the machine provided by ICCAD Contest
  - Upload every modules required to run your binary

Problem D by Mediatek
Chip Level Global Router

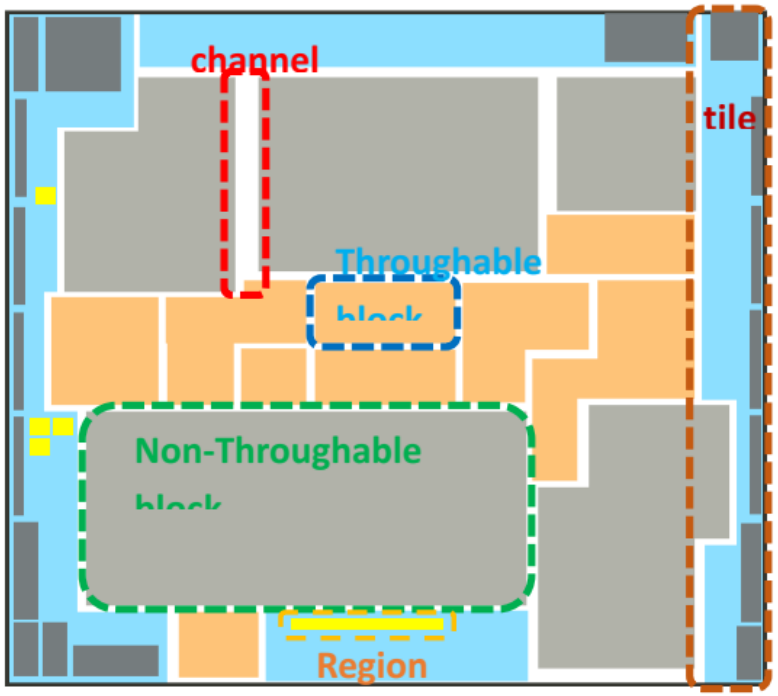
Introduction

- 當平面規劃(floorplanning)完成後，晶片將會被區分成 chip level 和 block level。
- Chip level 的繞線只能使用 block 佔據後剩餘的面積進行繞線，繞線區域可能會由許多不規則形狀的 channel 繞線區域所構成，這使得 chip level 的繞線相對於 block 具有更大的挑戰性。



Introduction (cont'd)

- Channel: 整個 chip 扣除 block 和 tile 剩下的區域即為 channel。
- Feedthroughable block: 允許 net 穿過的 block。
- Non-feedthroughable block: 除了 hard macro feedthrough (HMFT)外，不允許一般 net穿過此 block。
- Region: 散落的 standard cell會依據function 被分配到不同的 region。
- Tile:整個 die 扣除 block 和 nonthroughable block 後剩下的部分，會再進一步切出一些 tile。



Costs

- Channel overflow

$$(\#occupied\ routing\ tracks)/(\#available\ routing\ tracks)$$

  - 20 tracks/um 代表每一 um 中包含 20 條 routing tracks，本競賽將直接定義在每一組測試資料的輸入中
  - 每一種型態(type)的繞線會使用不同數量的 routing track
- Wirelength
  - 一條 net 中包含的所有 segment 長度的總和。
- Edge Pin Density

$$(net\ demand\ of\ an\ edge\ of\ a\ block)/(net\ capacity\ of\ an\ edge\ of\ a\ block\ )$$

  - 一個block 有四個邊(edge)，每一個邊上的引腳 (pin)所佔據的密度
  - Net capacity of an edge: 一個edge上最大可以容納的繞線數
    - Block A (x0,y0)->(x1,y1) 1000 (定義在輸入檔裡)
  - Net demand of an edge:一個edge上實際的繞線數
- Turn Cost
  - 每一條 net 轉折的數量

Input Files

- DEF
  - 描述電路實體佈局的狀態
  - 包含四種敘述 (VERSION、DESIGN、UNITS、及 DIEAREA)與一個 section (COMPONENTS)
  - chip_top.def: 給定所有 block 合法擺置
  - block.def: 描述在 chip_top.def 中使用的 block 形狀

A sample DEF of another design

```
VERSION 5.6 ;
DIVIDERCHAR "/" ;
BUSBITCHARS "[]" ;
DESIGN CS_GEN ;

UNITS DISTANCE MICRONS 1000 ;

PROPERTYDEFINITIONS
  COMPONENTPIN text STRING ;
END PROPERTYDEFINITIONS

DIEAREA ( 0 0 ) ( 1535670 576790 ) ;


COMPONENTS 10496 ;
- via5922 Via34
  + PLACED ( 345480 -370 ) N ;
- via5923 Via34
  + PLACED ( 405480 -370 ) N ;
- via5926 Via34
  + PLACED ( 1101480 -370 ) N ;
- via5927 Via34
  + PLACED ( 1185480 -370 ) N ;
```

Input Files (cont'd)

- CFG (副檔名為.json)
  - 描述每一個 block 的五個狀態
  - Block_name: block 的名字
  - Through_block_net_num: 當扣除從此 block 出發或接收的繞線後，可以 feedthrough 此 block 的繞線數量上限。
  - Through_block_edge_net_num: 定義在一個 block 從(X0,Y0)到(X1,Y1)這個edge出發、接收、以及 feedthrough 的繞線數量上限。
  - Block_port_region:定義在一個 block 邊上可以擺放引腳的矩形區域。
  - Is_feedthroughable:表示是否可以允許 net feedthrough 這個 block。

"block_name": AAA,

"through_block_net_num": 1000,

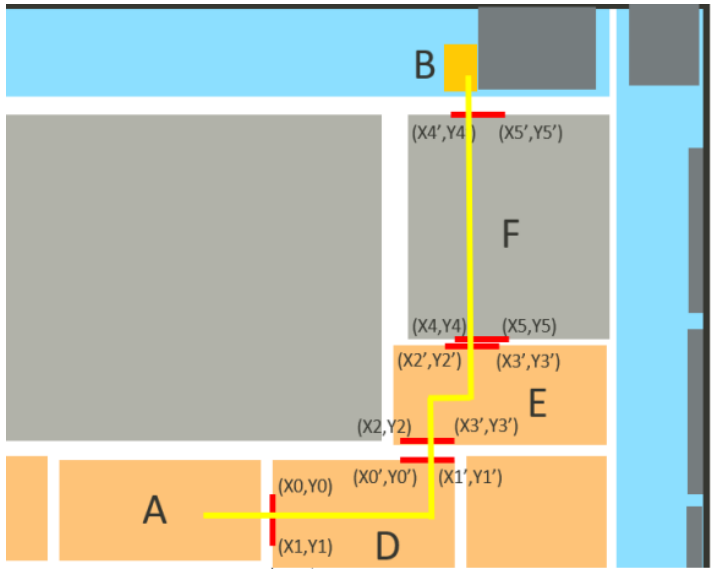
"through_block_edge_net_num": [(X0, Y0),(X1,Y1),100],

"block_port_region":[(X0,Y0),(X1,Y1)],

"is_feedthroughable:True"

Input Files (cont'd)

- Connection matrix (副檔名.json)
  - 描述每一條 net 的八種屬性
  - ID: net id。
  - TX: 定義繞線從哪一個 block 出發。
  - RX: 定義繞線的終點是哪一個 block。
  - NUM: 定義這條線需要使用的 routing tracks。



"ID": 0,

"TX": A,

"RX": [B,C],

"NUM": 10,

"MUST_THROUGH": [[D,(X0,Y0,X1,Y1),(X0',Y0',X1',Y1')],[E,(X2,Y2,X3,Y3),(X2',Y2',X3',Y3')]],

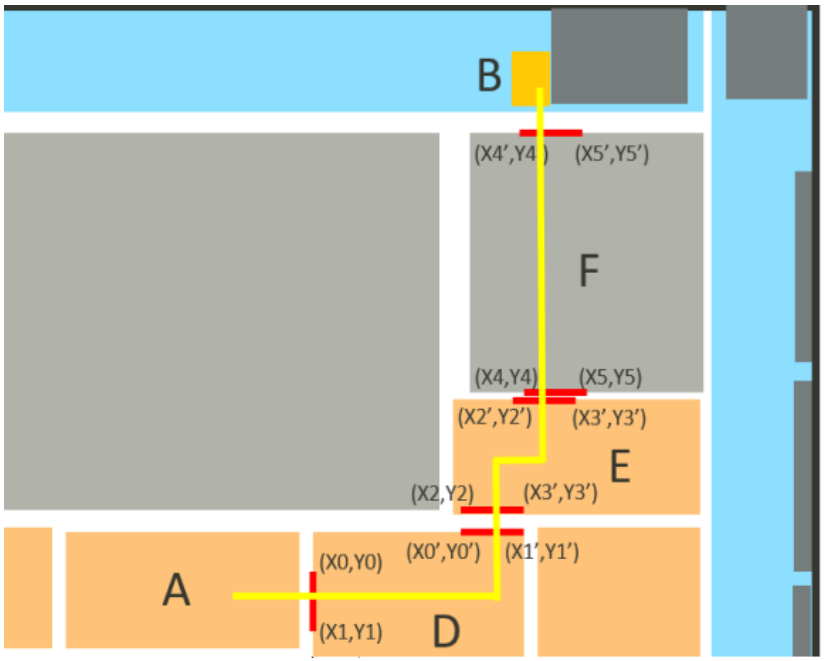
"HMFT_MUST_THROUGH":[[F,(X4,Y4,X5,Y5),(X4',Y4',X5',Y5')]],

"TX_COORD": [xa,ya],

"RX_COORD": [(xb,yb),(xc,yc)]

Input Files (cont'd)

- Connection matrix (副檔名.json)
  - MUST_THROUGH: 定義這條線必須經過的區域。
  - HMFT_MUST_THROUGH: 定義此繞線必須經過non-feedthroughable。
  - TX_COORD: 定義繞線的起始點相對於 TX block 左下角座標的相對位置。
  - RX_COORD:定義繞線終點相對於 RX block 左下角座標的相對位置。



"ID": 0,

"TX": A,

"RX": [B,C],

"NUM": 10,

"MUST_THROUGH": [[D,(X0,Y0,X1,Y1),(X0',Y0',X1',Y1')],[E,(X2,Y2,X3,Y3),(X2',Y2',X3',Y3')]],

"HMFT_MUST_THROUGH":[[F,(X4,Y4,X5,Y5),(X4',Y4',X5',Y5')]],

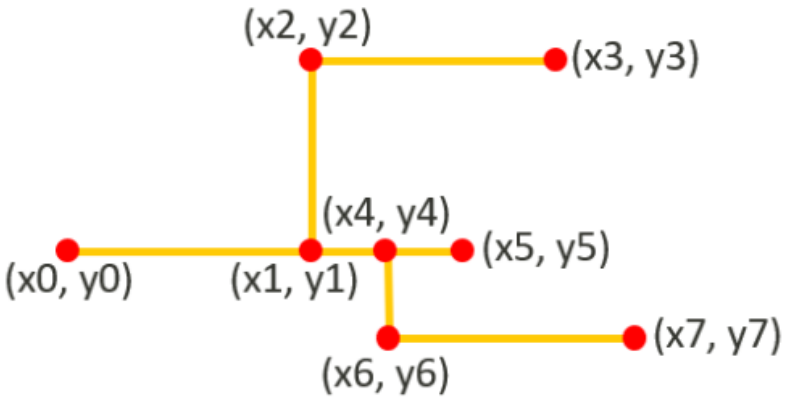
"TX_COORD": [xa,ya],

"RX_COORD": [(xb,yb),(xc,yc)]

Output File

- 每組測試資料個別輸出一個檔案caseOO_net.rpt (OO 為測試檔案編號)
  - 每一條 net 將有許多的水平或垂直 segment 構成，
  - [ID]為某一條 net 的 id
  - 一行輸出一個包含兩個端點的segment

[ID]
(x0,y0),(x1,y1)
(x1,y1),(x2,y2)
(x2,y2),(x3,y3)
(x1,y1),(x4,y4)
(x4,y4),(x5,y5)
(x4,y4),(x6,y6)
(x6,y6),(x7,y7)
[ID 2]
(x0,y0),(x1,y1)
(x1,y1),(x2,y2)
.....



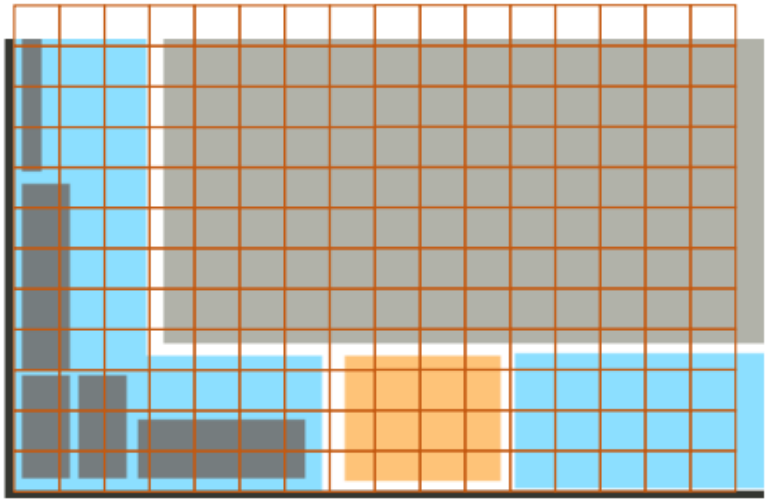
Overflow評估

- 將整個 chip 從左下角原點(0,0)開始，將整個區域切割成數個相同大小的正方形框，這些框稱為 gcell
- Evaluator 會將每一個 segment 的座標平移到gcell 的中心

$$x_{modified} = \left(\frac{x}{width_{gcell}} + 0.5\right) * width_{gcell}$$

$$y_{modified} = \left(\frac{y}{width_{gcell}} + 0.5\right) * width_{gcell}$$

  - $width_{gcell}$: gcell的寬度； 某一個測試資料中單位面積可以繞 20線 (i.e., 20 tracks/um), 而 所有net中最多需要使用的 routing track 數量為 100, 則 gcell 的寬度則定義為 5um。



執行與評分

- Command-line parameter:

  ./CGR XX(tracks/um) caseOO.def caseOO.cfg.json caseOO.connection_matrix.json

  - 執行檔取名為 “CGR”
  - OO 為測試檔案編號, XX 為整數

- 一定要滿足：
  - 每組測試資料的程式執行時間(包含讀寫檔案過程)必須在 2 小時內完成。
  - 每組測試資料產生的檔案格式必須符合 output format 定義。
  - 每條繞線的起始和終點位置必須正確。
  - 每條繞線必須符合MUST_THROUGH和HMFT_MUST_THROUGH的規定。
  - 繞線不能經過 is_throughable 定義為 False 的 block。
  - 繞線必須在 chip_top 內。
- 盡量滿足
  - 若超過through_block_net_num和through_block_edge_net_num的限制， 會對其結果酌量扣分。

執行與評分 (cont'd)

$$score = 0.55 * cost_{overflowLength} + 0.35 * cost_{edgePinDensity} + 0.1 * e^{\frac{time}{2*60*60}} + 0.3 * penalty_{\#pin\_constraint} + 0.01 * penalty_{\#net\_turn}$$

其中

$$cost_{overflowLength} = \sum_{net}^{\#all\ nets} \left(\sum_{segment}^{\frac{netlength}{gcell_{width}}} \left(1 + \left(\left[\frac{\#used\ track}{capacity_{gcell_{edge}}} \right] > 0.7\ else\ 0 \right) \right) * gcell_{width} \right) / HPWL_{bbox\_net}$$

$$HPWL_{bbox\_net} = hpwl\ of\ bounded\ box\ of\ the\ net$$

$$cost_{edgePinDensity} = \sum_{block}^{\#all\ blocks} \sum_{edge}^{\#block\ edges} \sum_{segment}^{\frac{edgelength}{gcell\_width}} \left(\left[\frac{\#used\ track}{capacity_{gcell\_edge}} \right] > 0.6\ else\ 0 \right)$$

$$penalty_{\#pin\_constraints} = \sum_{each\ constraint}^{all\ constraint} e^{\left(\left[\frac{\#used\ track}{\#pin\_constrains} \right] > 1\ else\ -inf. \right)}$$

$$penalty_{\#net\_turn} = \sum_{net}^{all\ net} e^{([\#net_{turn}] > 1 else -inf.)}$$