

再構成型アーキテクチャ特論 (9)

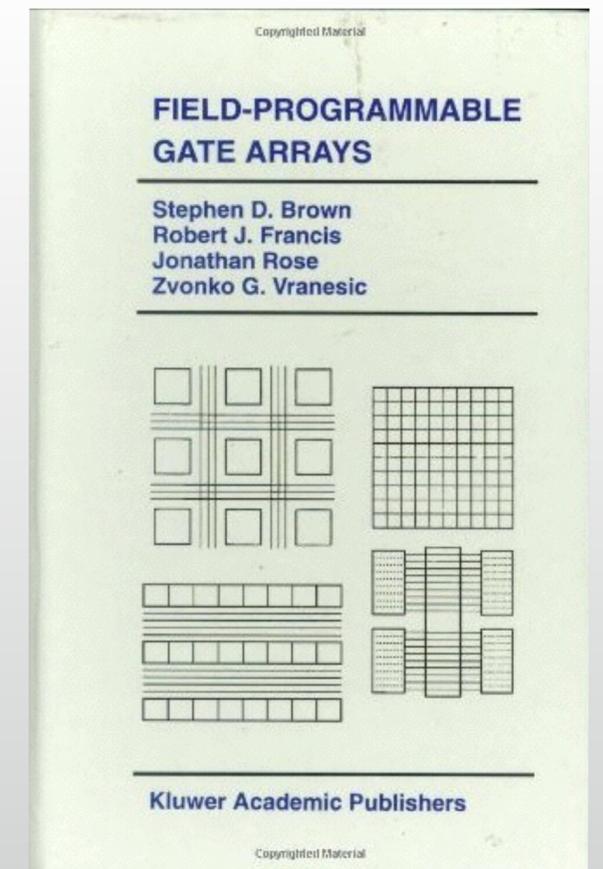
2016年度後学期 長名

osana@eee.u-ryukyu.ac.jp

先週のまとめ

- * FPGA の基本構成: LB, CB, SB
 - * ビルディングブロックを効率よく作る: 加算器、SR、メモリ
 - * クラスタリングによる配線遅延の低減
- * その他の構成要素: 入出力, クロック

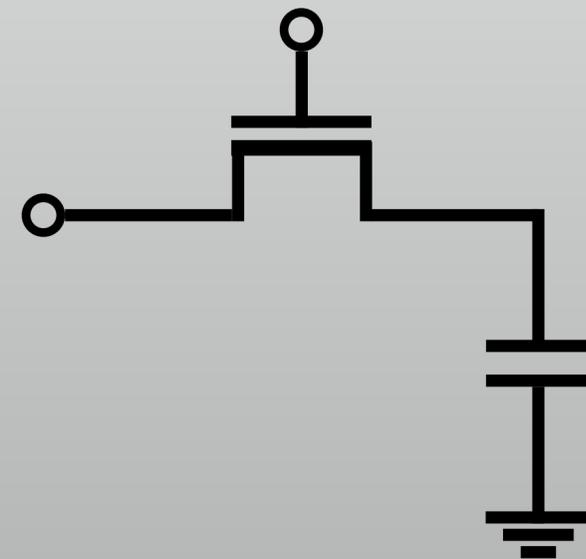
先週の積み残し



- * LUTは何本入力があるのがいいのか？
- * CBやSBは完全結合じゃないよ: どれくらいの接続ポイント数が必要？
- * SBとSBをつなぐトラックの長さや本数はどうなっているの？

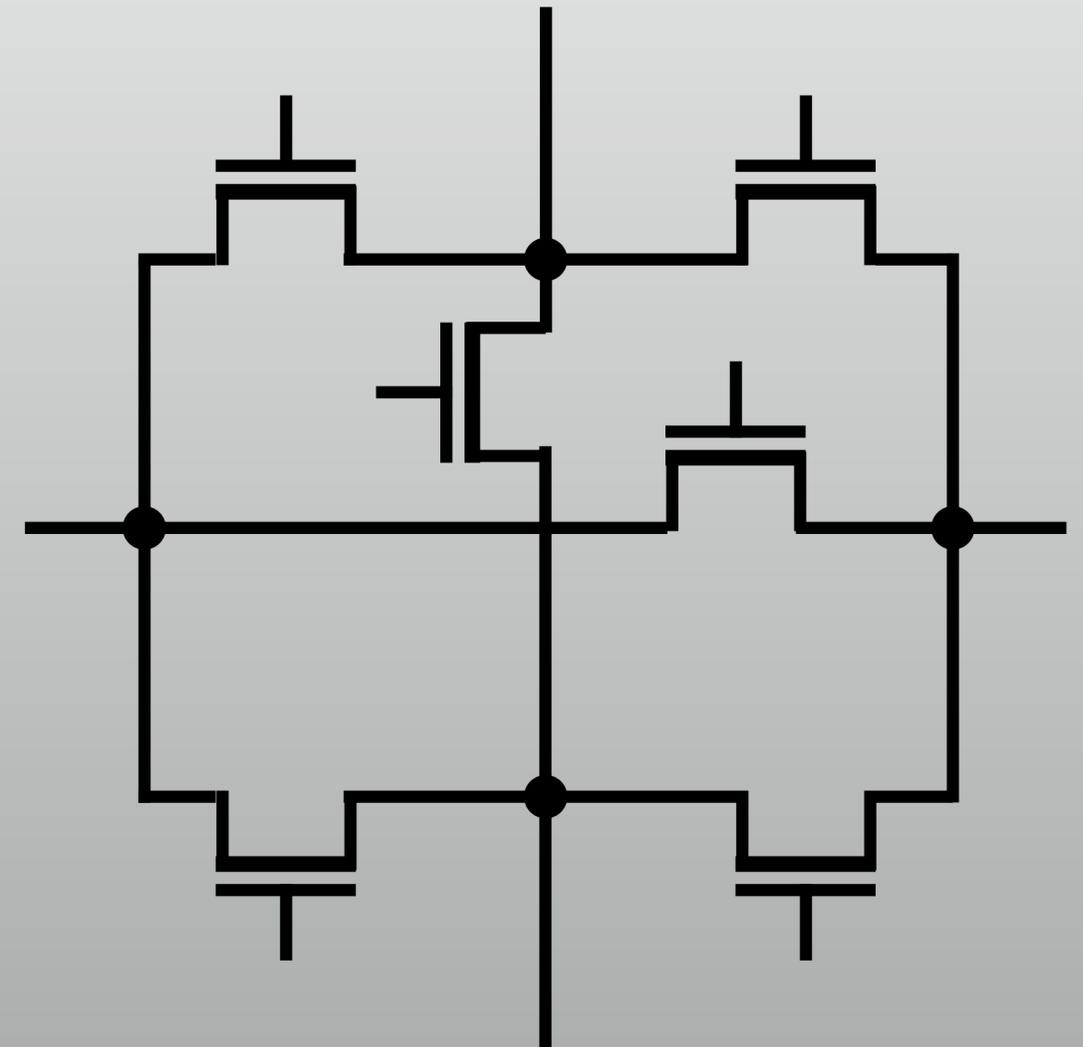
スイッチはどう作る？

- * スイッチは on なら短絡、off なら開放
 - * 方向はどちらでも
- * 普通のトランジスタは片方向だけ
 - * DRAM のセル (1C1T) がそうですね
 - * 読み出しでも書き込みで電流の方向が違う



パストランジスタによる配線スイッチ

- * 縦横4方向からの配線
- * パストランジスタ6つで自由に接続可能
- * ゲートは FF を接続してコントロール



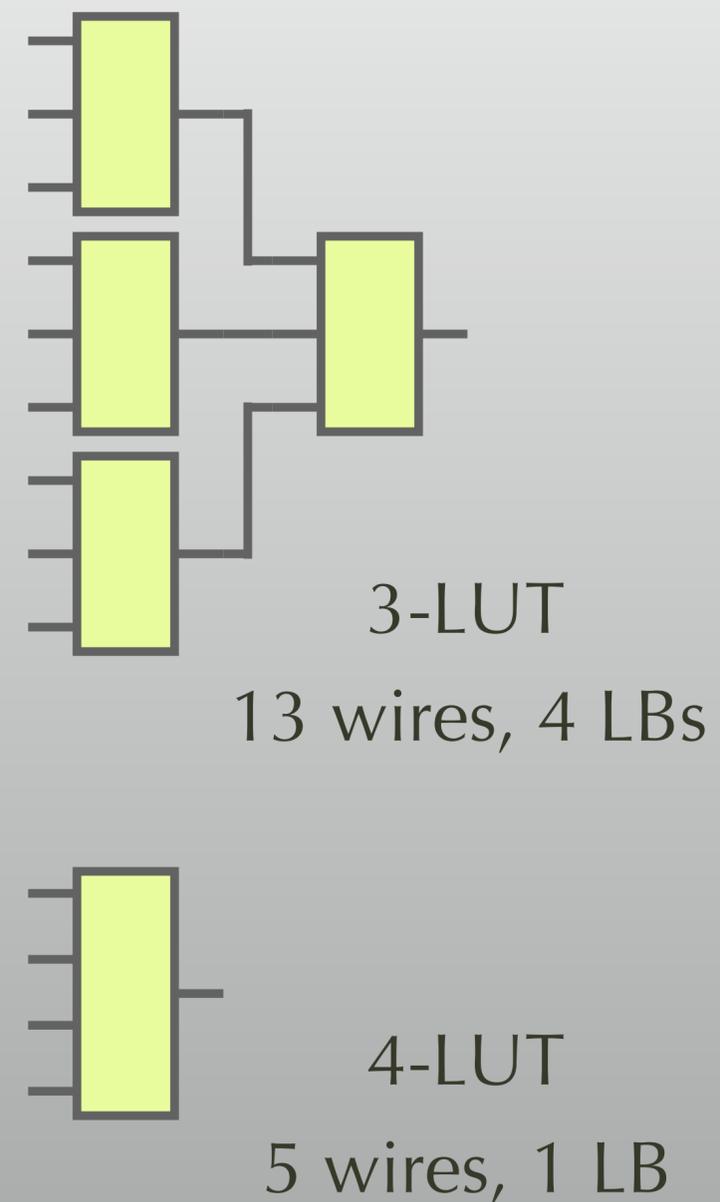
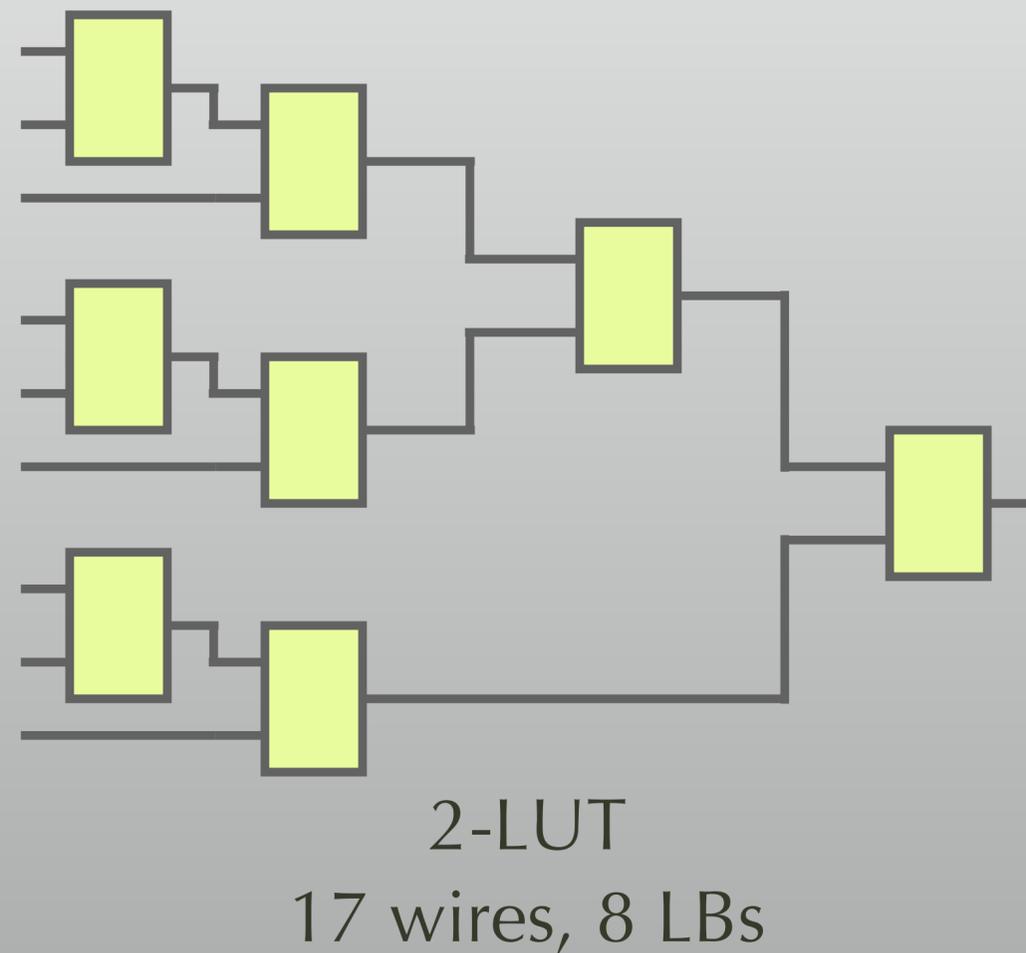
Logic Block

LBの機能性

- * LBをどう構成するかは非常に重要
 - * FPGAのチップに何がどれくらい載るか
 - * ロジックはどれくらい載せられるか
 - * 配線はどれくらい必要で載せられるか

エリア効率

* $f = abd + bcd + \bar{a} \bar{b} \bar{c}$

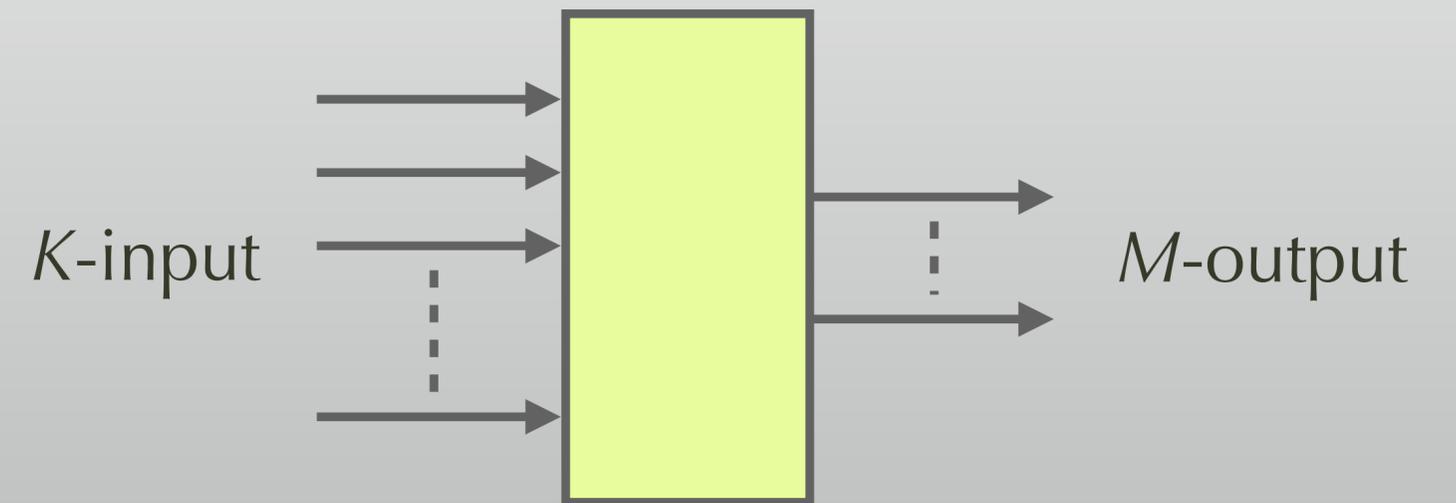


LBの構成

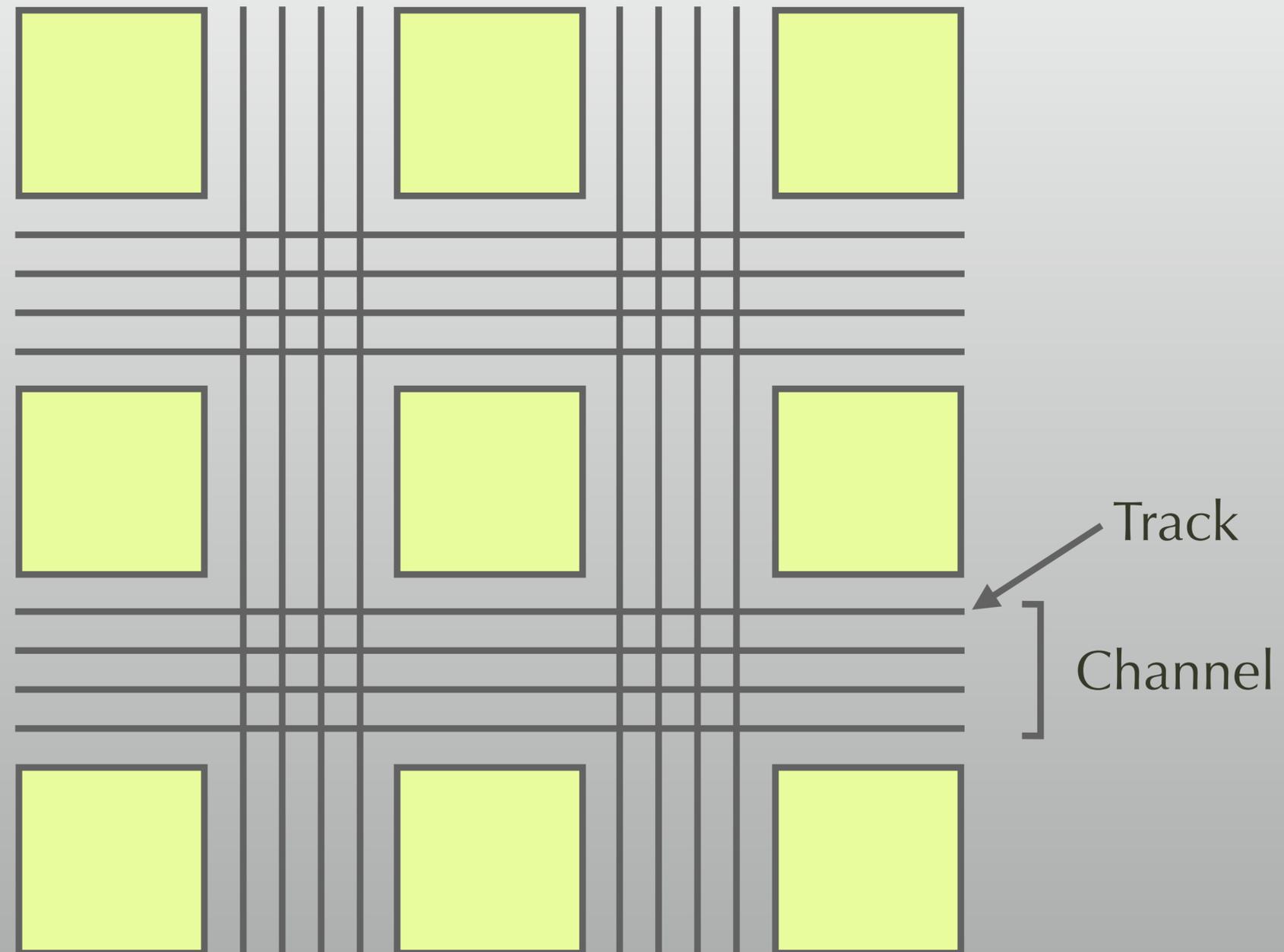
- * K-input, 1-output のLUT
- * K-input, M-output の LUT
- * 分解可能なLUT
 - * いずれにしてもKやMがあまり大きいと必要な配線数が問題に

分解できないLUT

- * 一般的に使われているもの



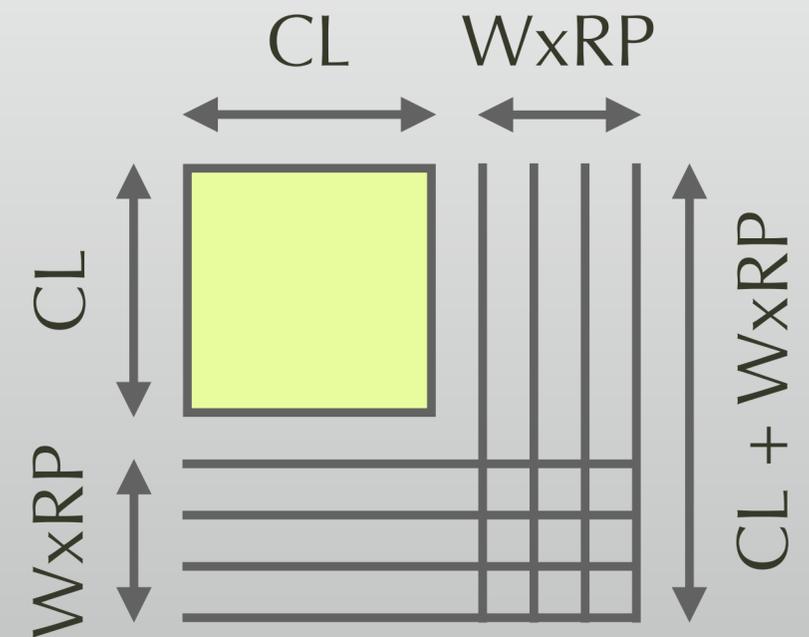
Routing Model



面積モデル

論理ブロックの基本モデル

- * $K =$ 入力数、 $M =$ 出力数
 $BA =$ bitあたり面積
 $FA =$ LUTのアクセスに必要な(固定の)面積
- * Logic Block Area = $FA + (M \times BA \times 2^K)$
- * $CL = \sqrt{\text{Logic Block Area}}$

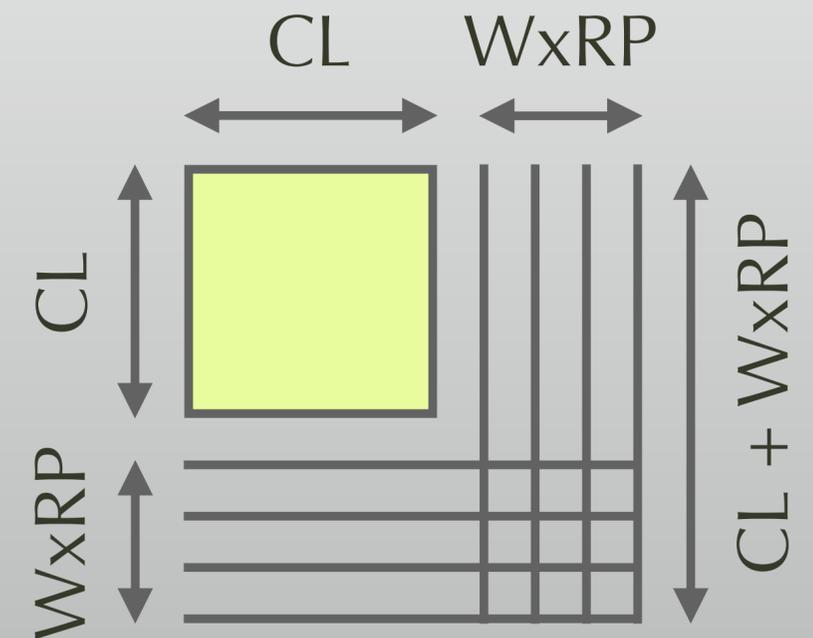


面積モデル

配線部分の基本モデル

- * $W = \text{Channel} / \text{Track}$
 $RP = \text{Routing Pitch}$

- * $\text{Routing Area} / \text{Block} = 2(\text{CL} \times \text{W} \times \text{RP}) + (\text{W} \times \text{RP})^2$



比較する

どの構成がよいか？

- * 実験方法
 - * いくつかの論理回路をテクノロジマッピングして LB へ割り当てる
 - * 配置(Placement) を行って LB の場所を決定
 - * Global routing を行い、どのチャンネルで何本トラックが必要かを決定

K-input LUT の面積

- * FPGA の論理ブロックとしてひろく使われている
- * もっとも基本的な構成

LBの面積と数

- * 論理ブロックの数は4入力まで一気に落ちる
- * ブロックの面積は5入力から急に大きくなる

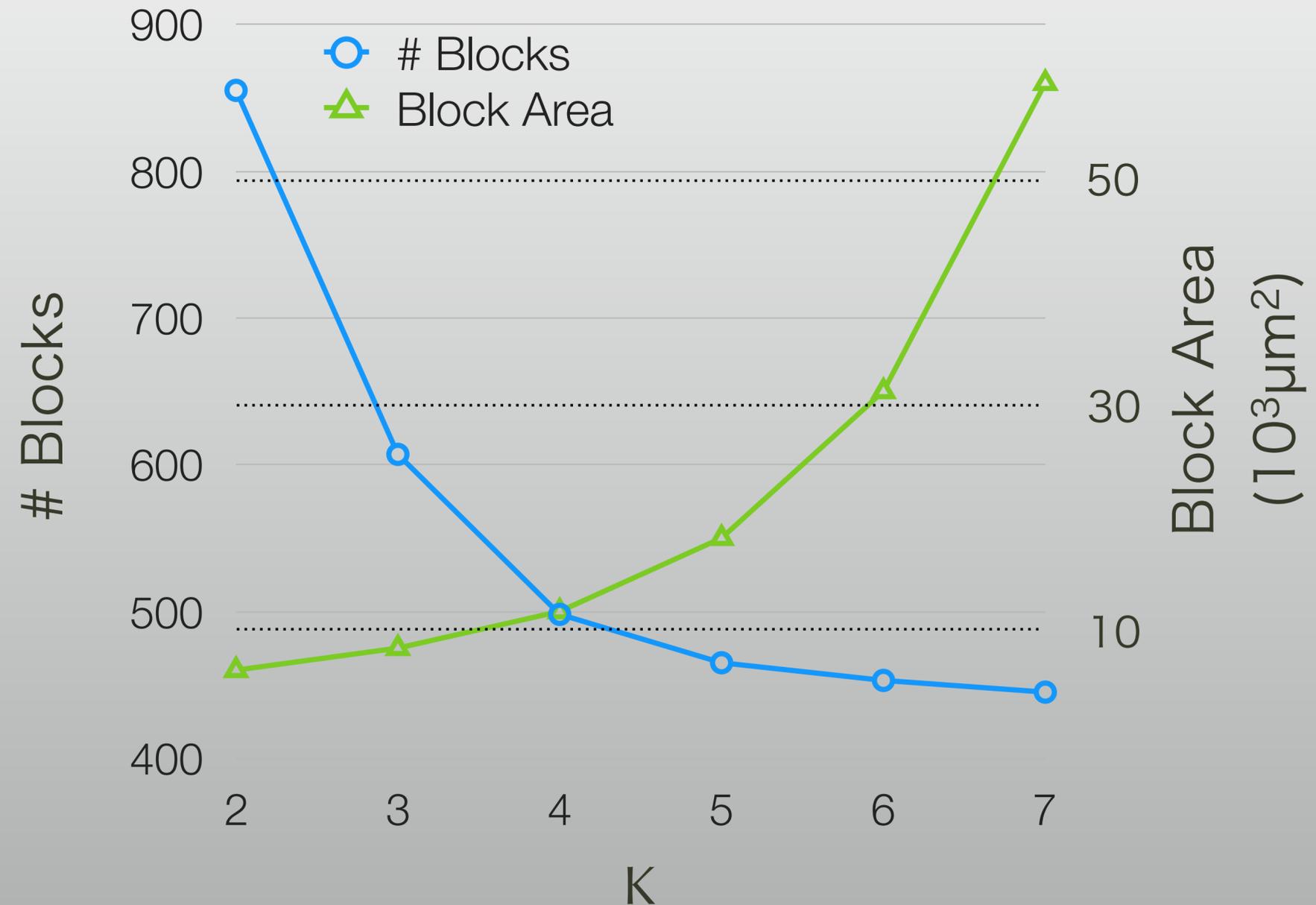


Fig 4.6 No. of Blocks and Block Area, for one Circuit

配線の面積とLBの数

- * LB の入力数を変えた場合
 - * ブロックあたりの配線面積(緑): ずっと増える
 - * 必要なブロック数 (青) は 4 入力でがくっと減る
 - * 青はさっきと同じ結果

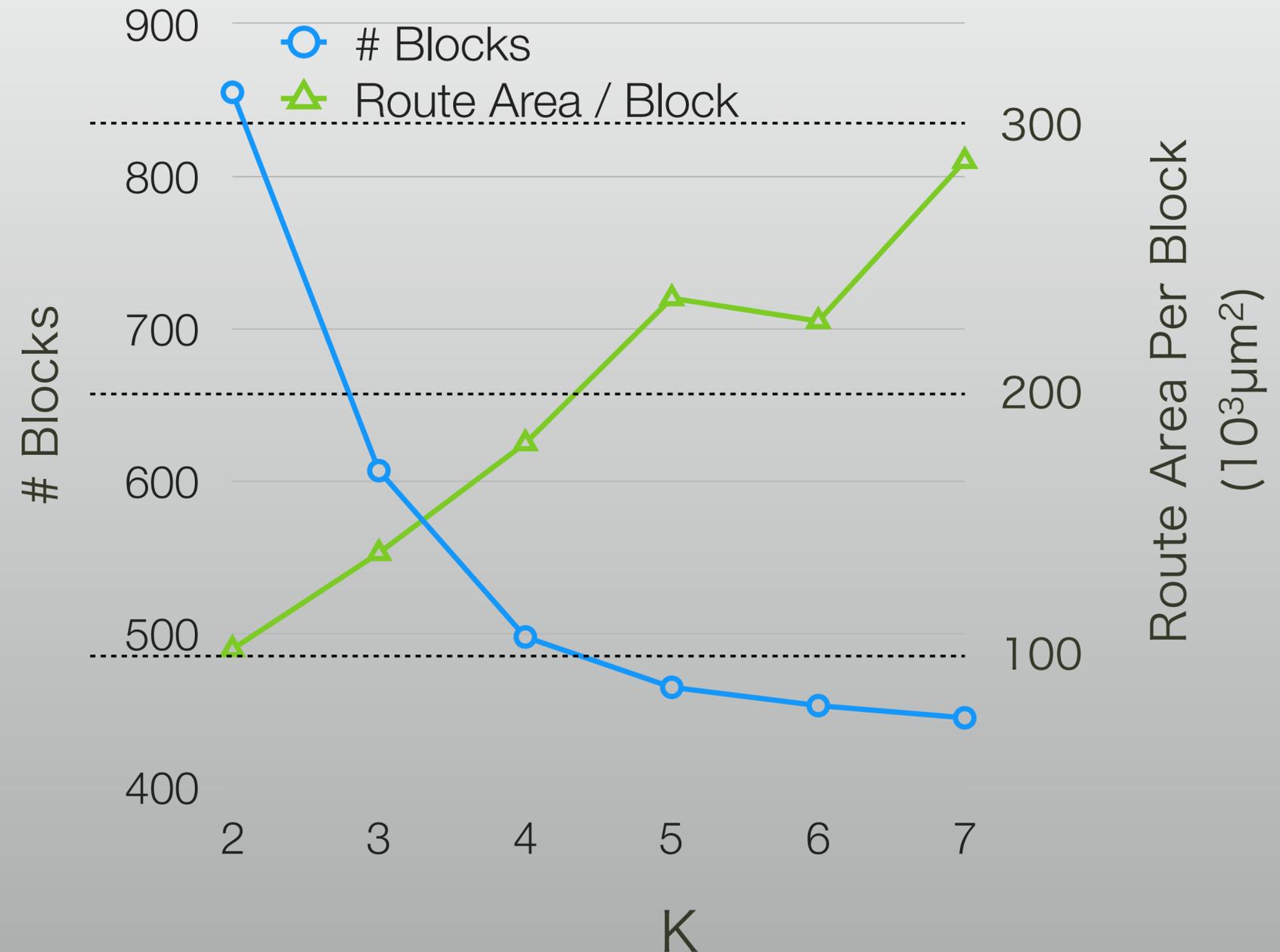


Fig 4.6 No. of Blocks and Routing Area / Block, for one Circuit

回路全体の面積

- * LB の入力数を変えた場合
- * 4-LUT の結果が、Bit Areaに関係なく良好

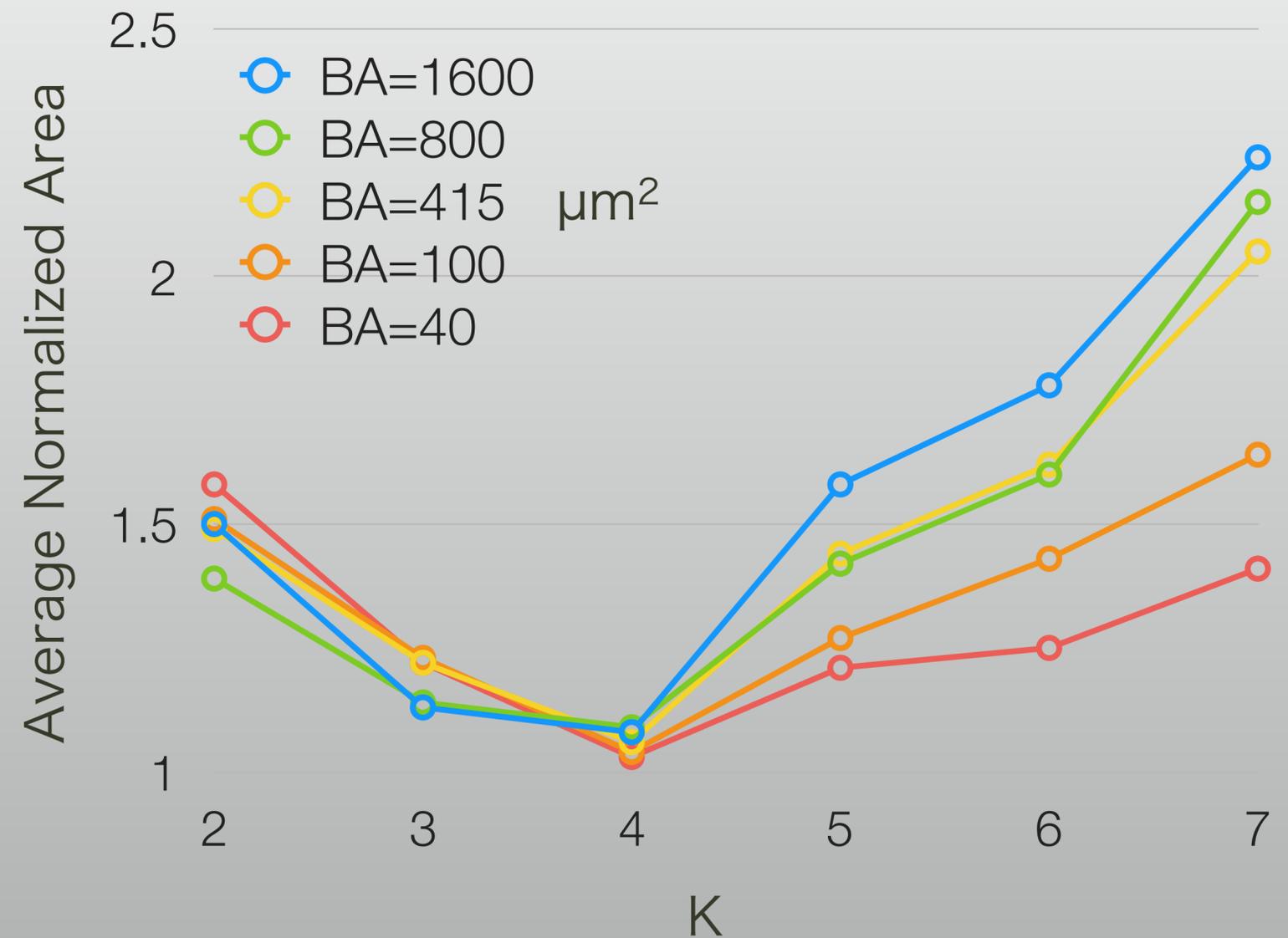


Fig 4.8 Average Normalized Total Area for Single Output K-LUT

K-input LUT

- * 入力数が増えると、だいたい...
 - * 論理ブロックの必要な数は反比例な感じで減少
 - * 論理ブロックあたりの面積は 2^n で増える
 - * 配線の面積は比例で増える
- * トータルでは bit area に関わらず 4-LUT が有利

多出力LUT・分割可能LUT

- * K-input LUT より高度な論理ブロック
 - * Altera の ALM (Adaptive Logic Module) とか
 - * 基本的に6-LUTだが2つの5-LUT としても使える
(つまりそのための余分の入力を持っている)

K-input M-output

- * 入力と出力の数を変更
- * ここでも4入力がベスト

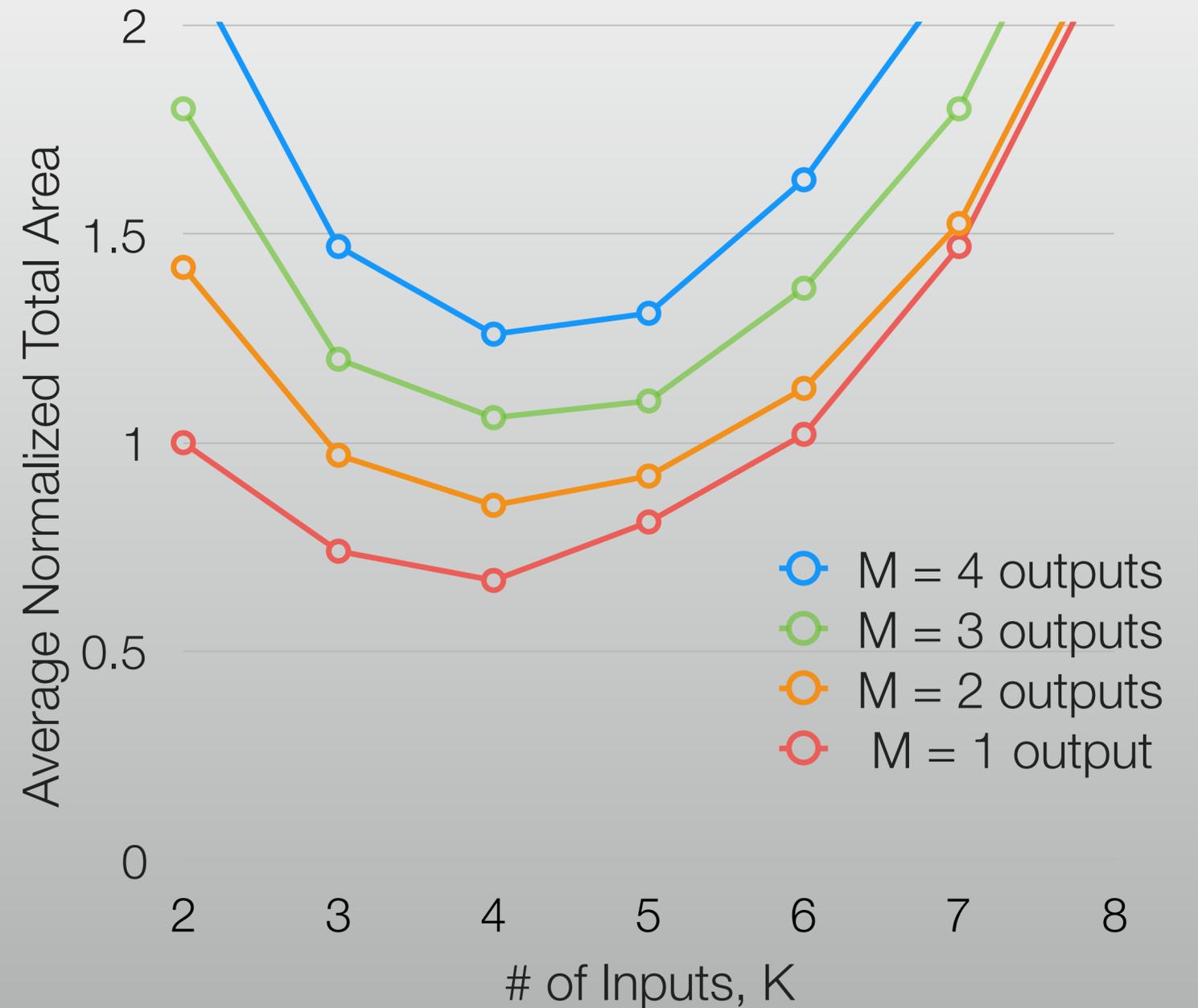
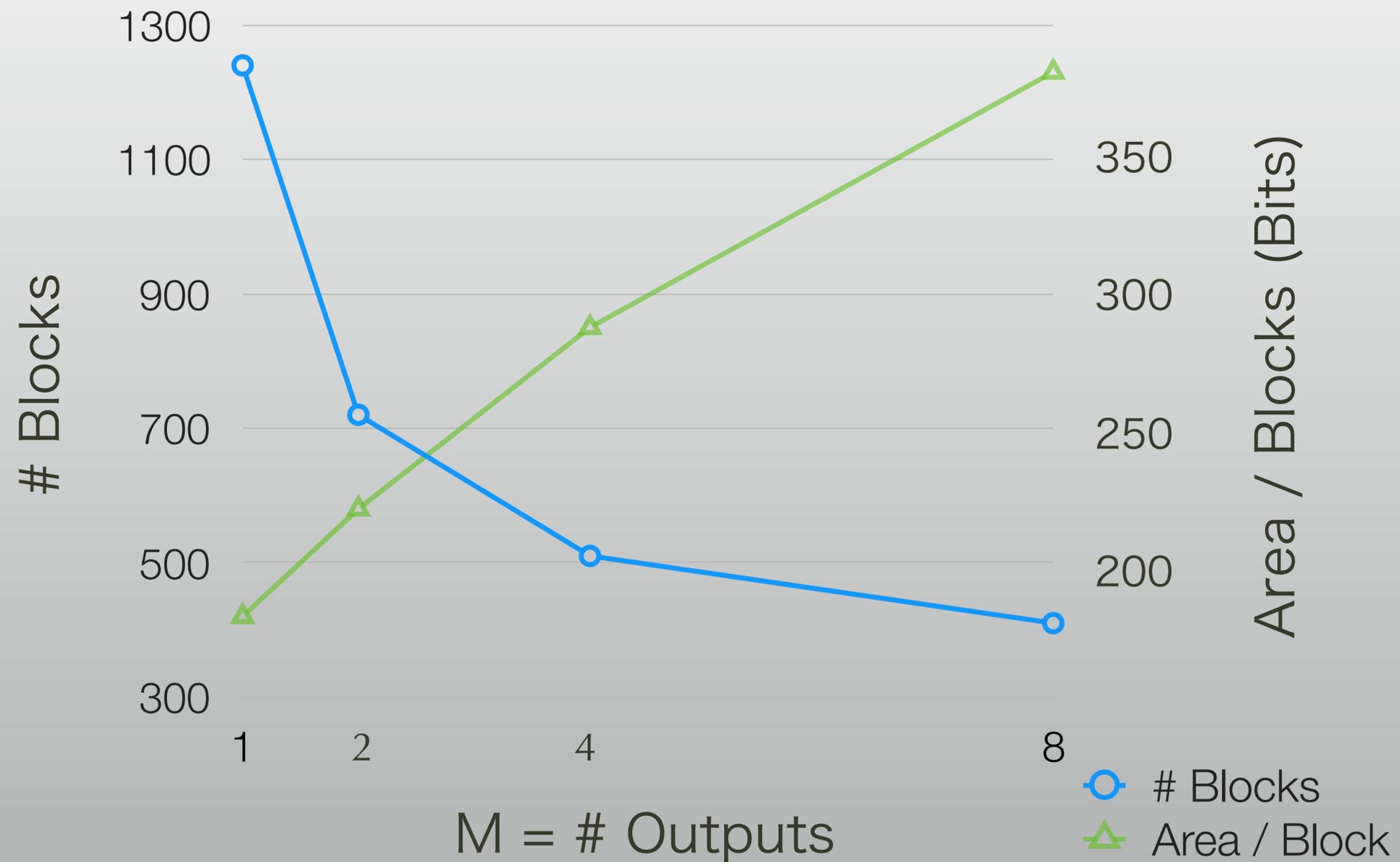


Fig 4.9 - Average Normalized Total Area [Koul92a]

分解可能型LUT

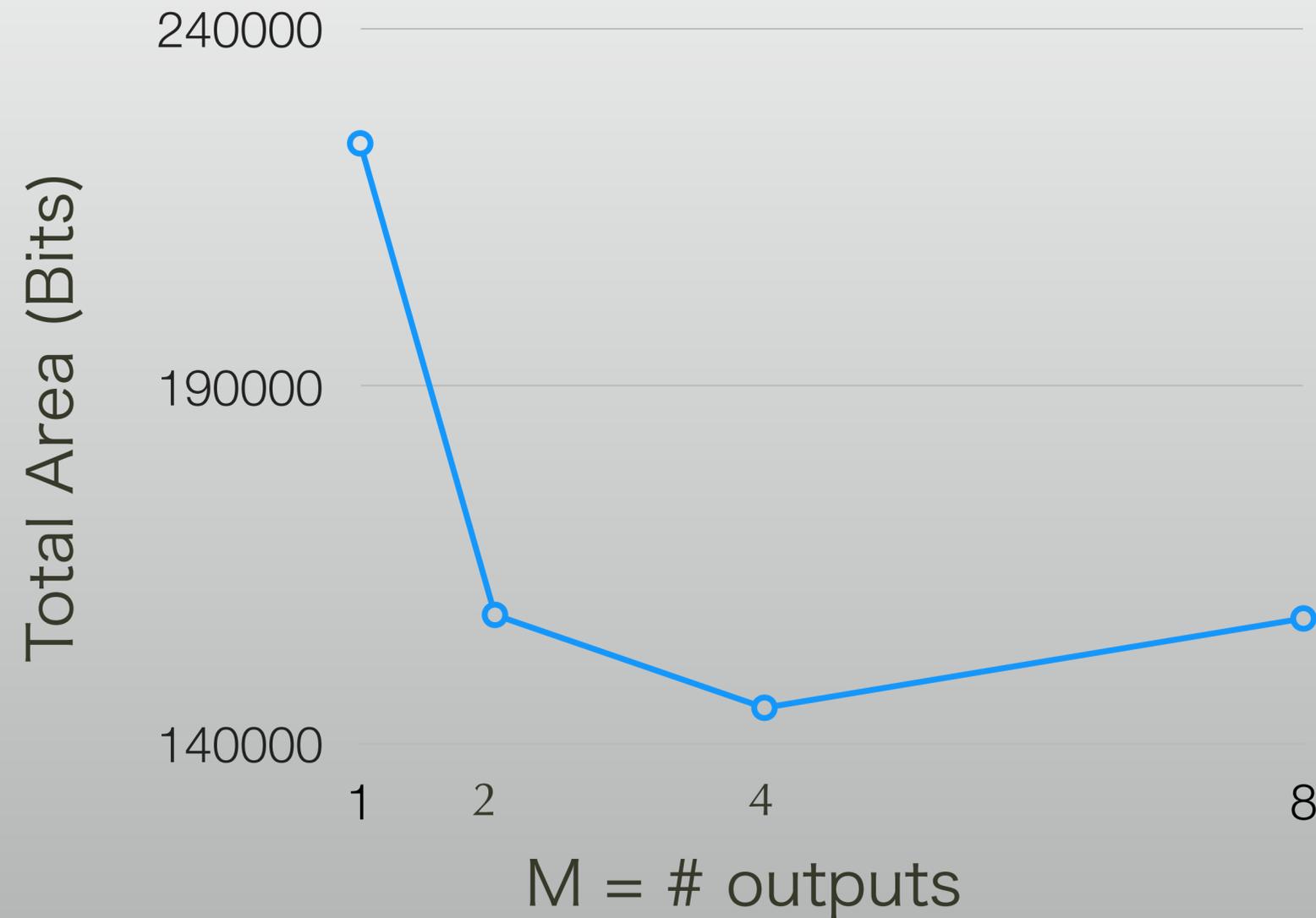
- * 出力数 $M=1, 2, 4, 8$ な構成を想定
 - * $2^5 = 32\text{bit}$ で構成される
 - * $M=1$: 5-LUT, $M=2$: 2x4-LUT, $M=4$: 4x3-LUT, $M=8$: 8x2-LUT

分解可能型LUT



出力が増えると配線に必要な面積(ビット数)が増加する

分解可能型LUT



トータルビット数は 3-LUT を4つ、が最小

まとめ

- * 多出力でも 4入力
- * 分解を許すと小さなLB数で収まる
 - * その一方で配線がどんどん増える...
- * K-LUT (1出力) の入力数はどうする？
 - * 遅延の面からの検討も必要

LUTの入力数と遅延

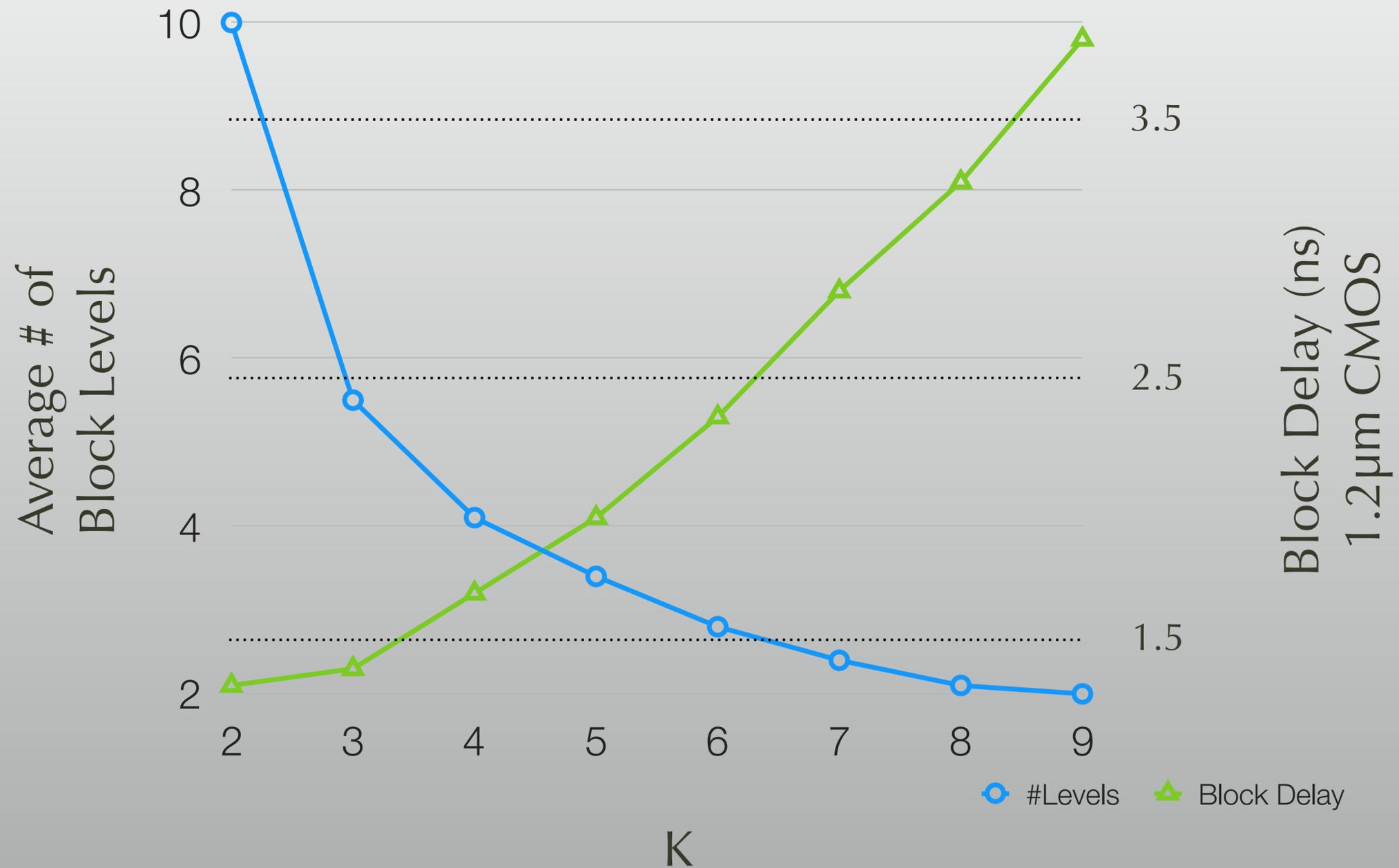


Fig 4.14 Avg. No. of Logic Block Levels and Block Delay for K-LUTs

配線遅延とLUTの入力数

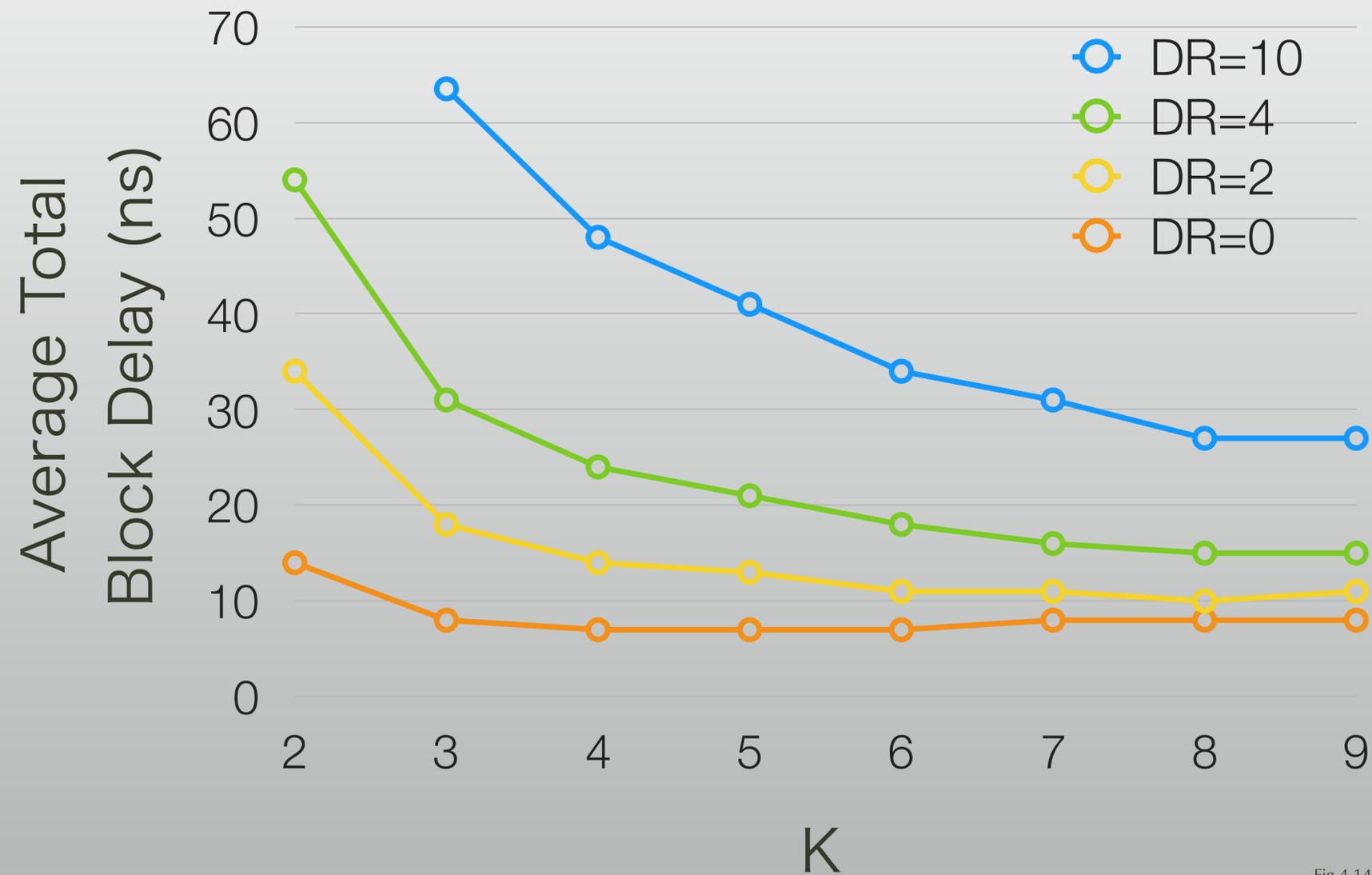


Fig 4.14 Average Total Delay for K-LUTs

D_R (配線遅延)は2~4くらいが現実的
 D_R も K も両方大きいというのは現実的でない

論理ブロックに関するまとめ

- * 配線遅延のことを考えなければ、4-LUT が最良のバランス
- * これは Bit area に関係ないのでプロセス非依存
- * 「たいていの論理回路」の性質による？
- * 配線遅延が大きいと5-LUTとか6-LUTも有利

Connection Block / Switch Block

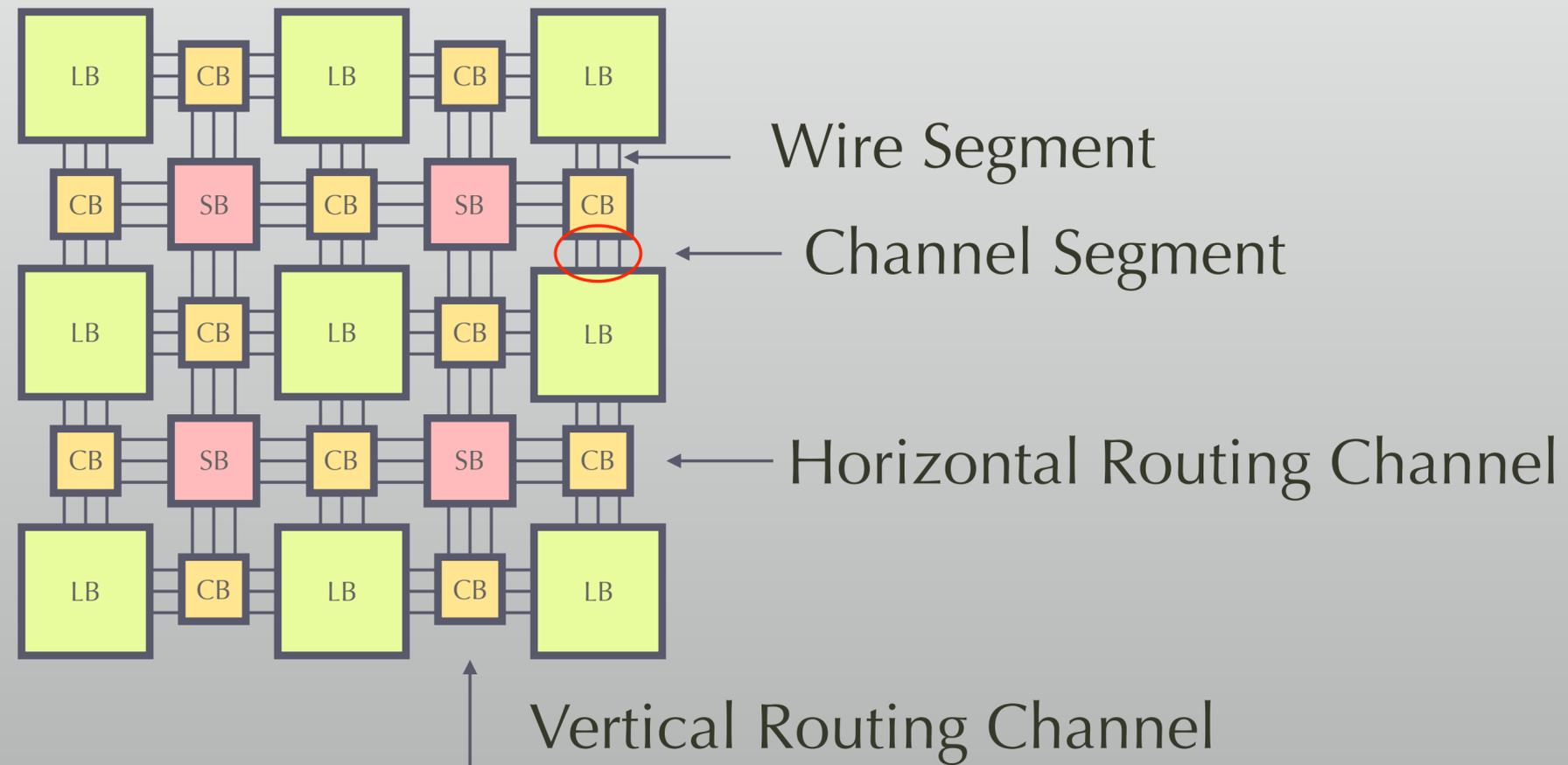
用語

- * pin: LB の入力・出力
- * connection: 互いに接続されているひと組のpin
- * net: 電氣的に接続されているpinの集合。
 - * 複数のconnectionで構成される
- * wire segment: connectionの一部をなす、直線の配線ひとつ

用語

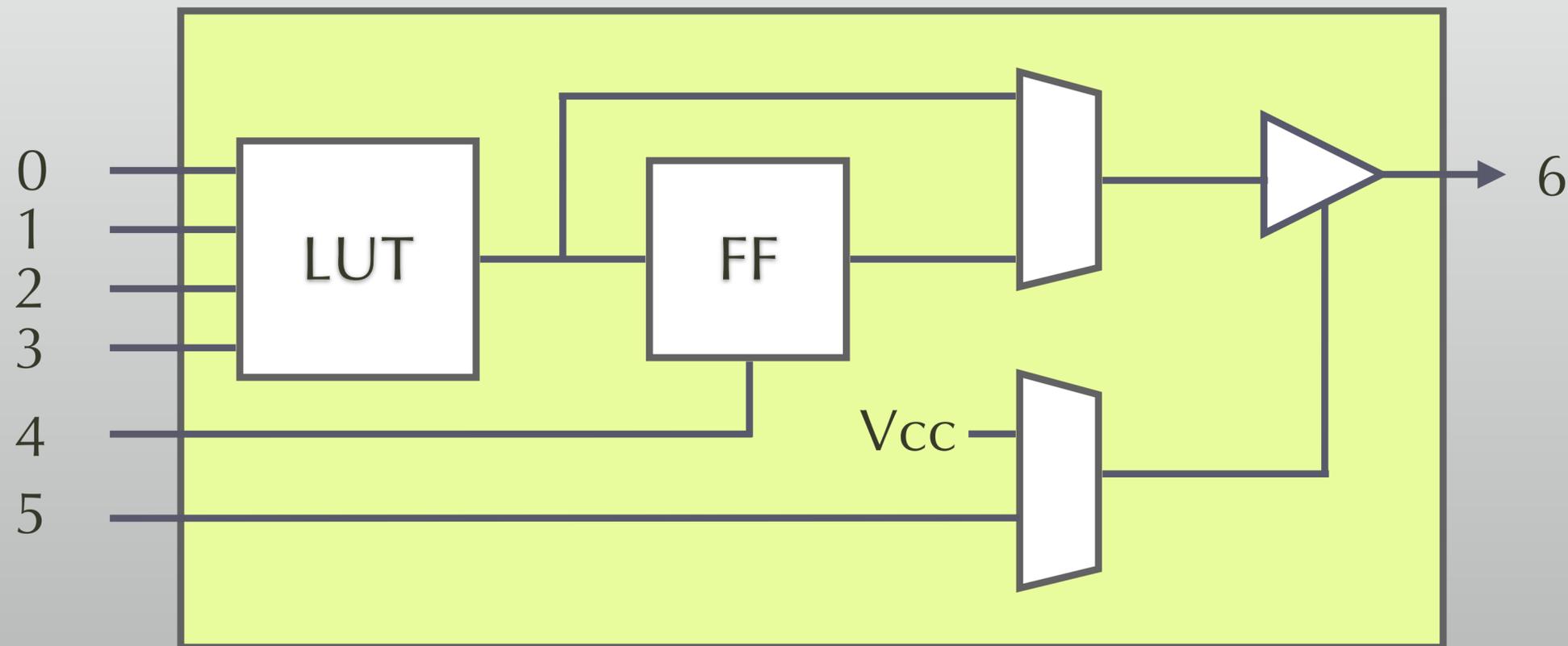
- * routing switch: 2つのwire segmentを電氣的に接続することのできるプログラマブルスイッチ
- * track: routing channelの長さ(あるいは幅)と同じ長さの配線。
- * routing channel: ふたつの行または列の間にまたがる配線領域

用語

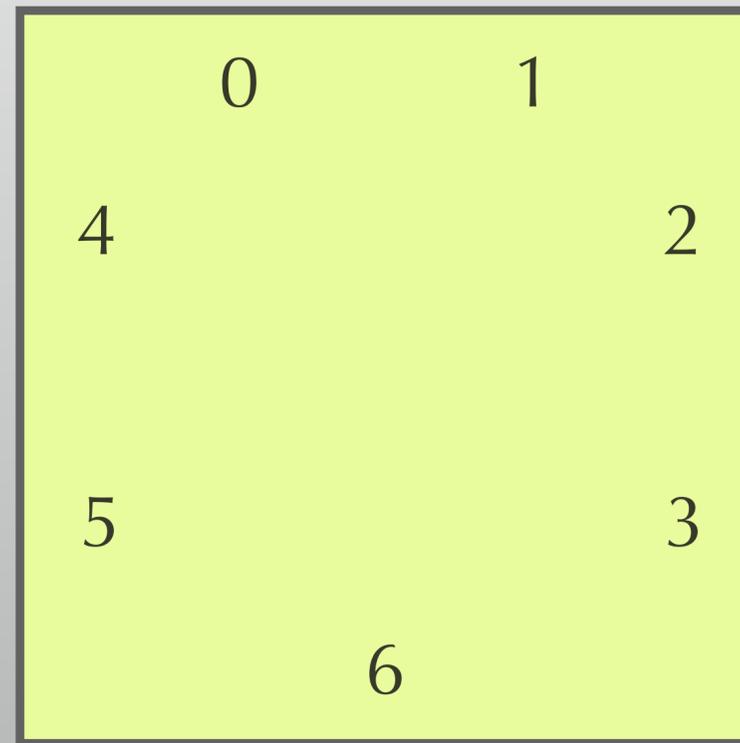


LBの構成と配線アーキテクチャ

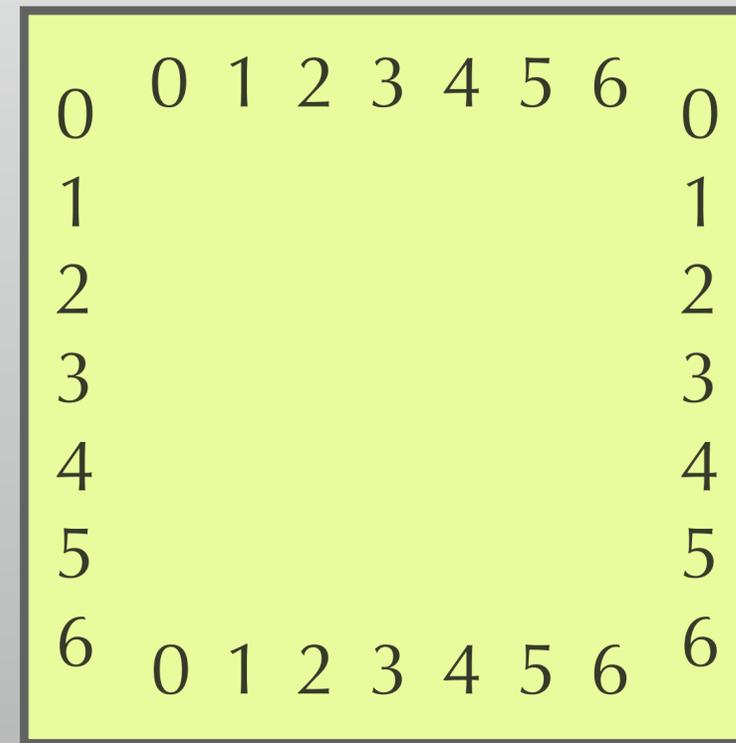
- * LB にはピンがある
- * 配線と密接な関係
- * なので、まずLBについて



LBのピン配置

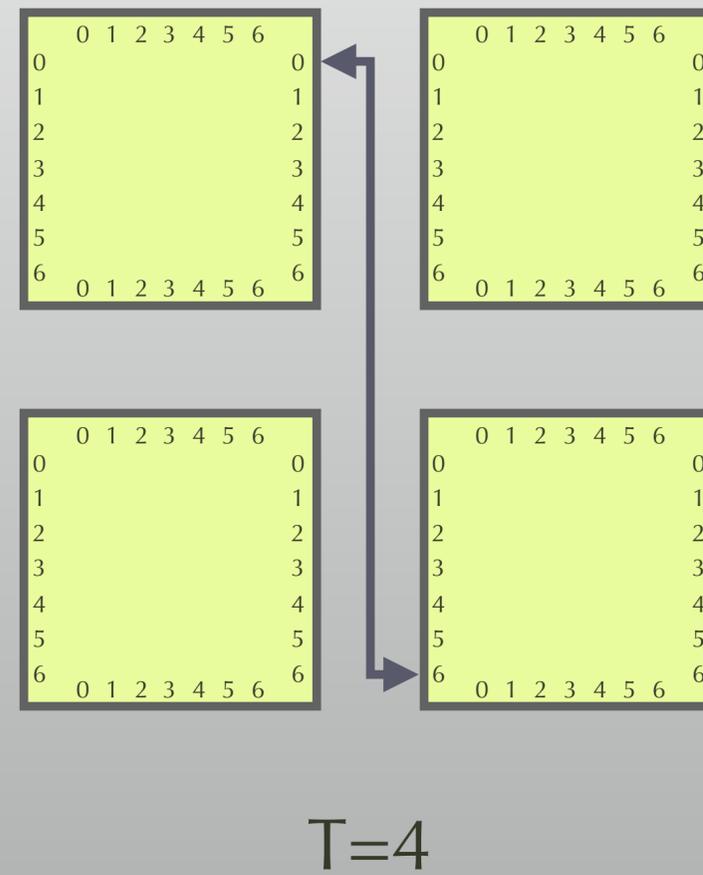
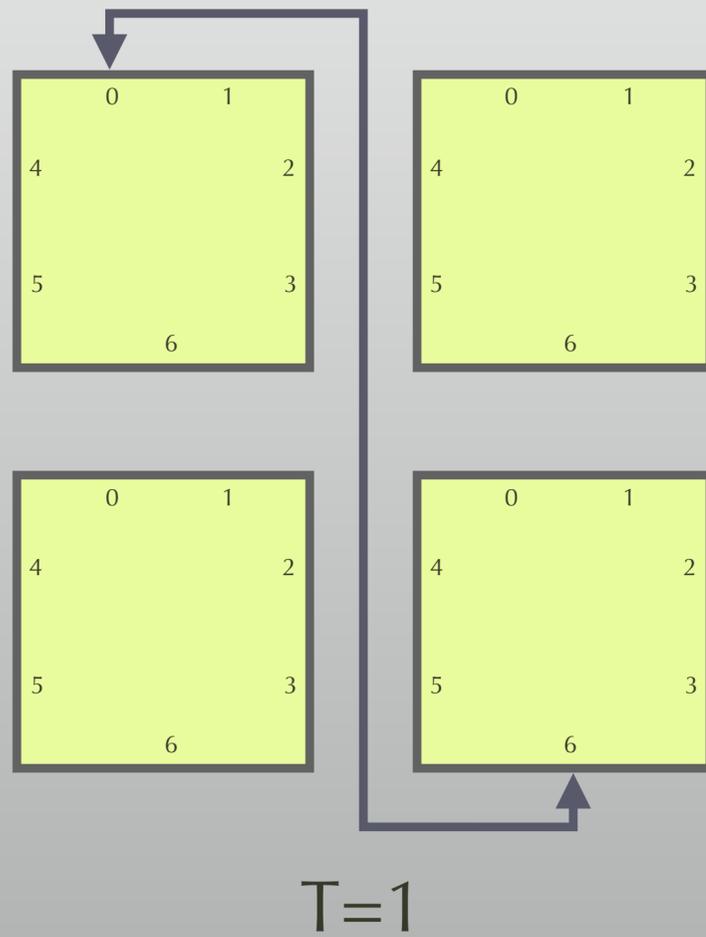


T=1



T=4

Tと配線長



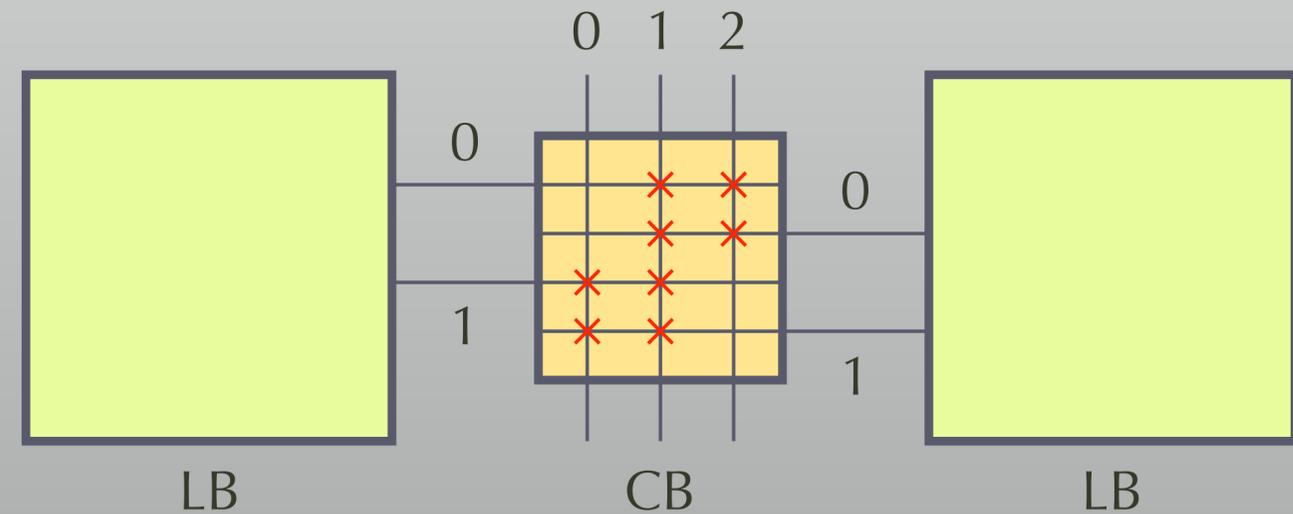
Tの最適点

- * Tが少ないと配線密度(Channel Density)が上がる
- * Tが大きいと無駄になるスイッチが増える
- * 2くらいでよい？

| T | 平均Channel Density |
|---|-------------------|
| 1 | 15 |
| 2 | 12 |
| 3 | 11 |
| 4 | 11 |

Connection Block

- * チャンネルはCBを通り抜ける
- * LBのピンはCBの向こうへは行かない (ことにしておく)
- * 1つのピンがつながるチャンネルワイヤの数: F_c

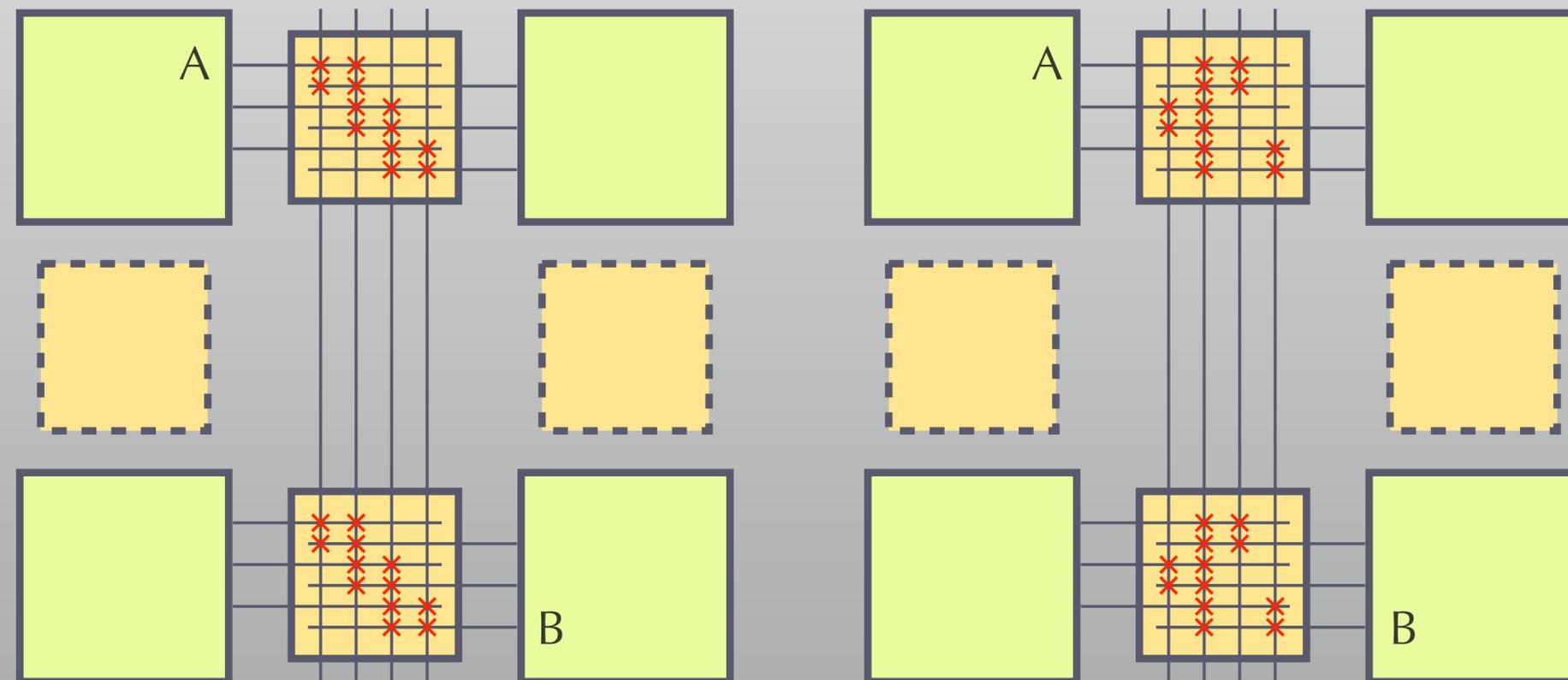


この場合の $F_c = 2$

CB Topology

スイッチの配置

- * ピンA からピンB に配線できるか？
- * 左のスイッチ配置では共通の配線がない...



よいトポロジ

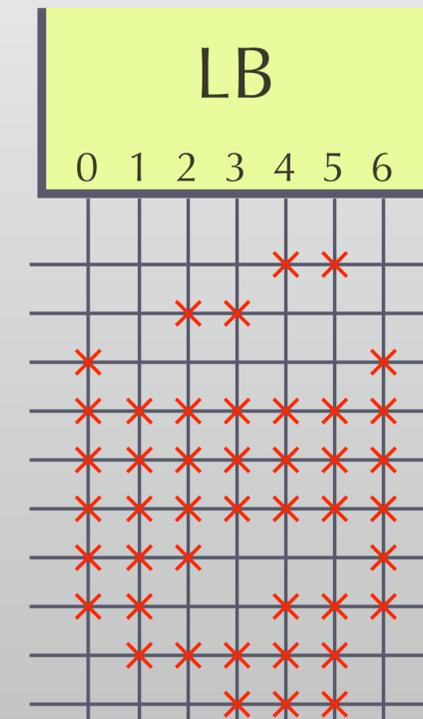
- * ピンからピンへ行ける自由度が必要
 - * 共通のtrackが見つかるチャンスを多く
- * スイッチが分散していることも重要
 - * 使用できないtrackが発生するケースを少なく

トレードオフ

- * スイッチをどう分散させるか が鍵
 - * F_c がトラック数 W に近いなら設計は楽
 - * $F_c < 0.5W$ くらいになると難しい
- * 統計的にどのピンとどのピンの接続が重要か、
を調べて反映する (回路を載せないとわからない...)

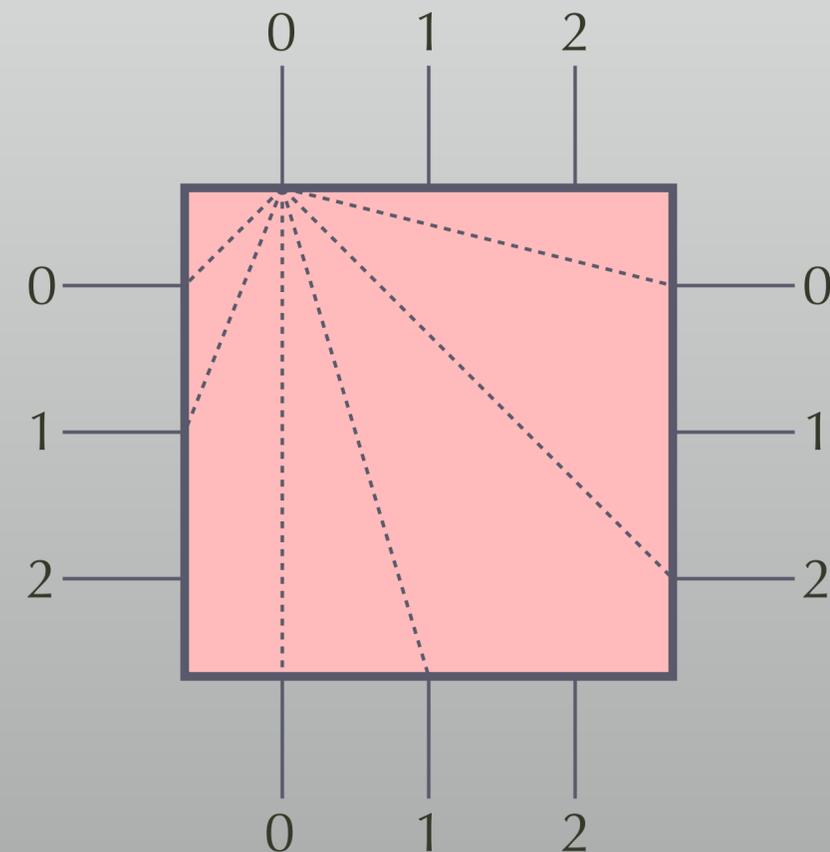
実例

- * $W=10, Fc =6$
- * 0 (入力) と 6 (出力) はよくつながる
- * 6本のtrackを共有
- * 5 (入力) とはあまりつながらない: 3本を共有



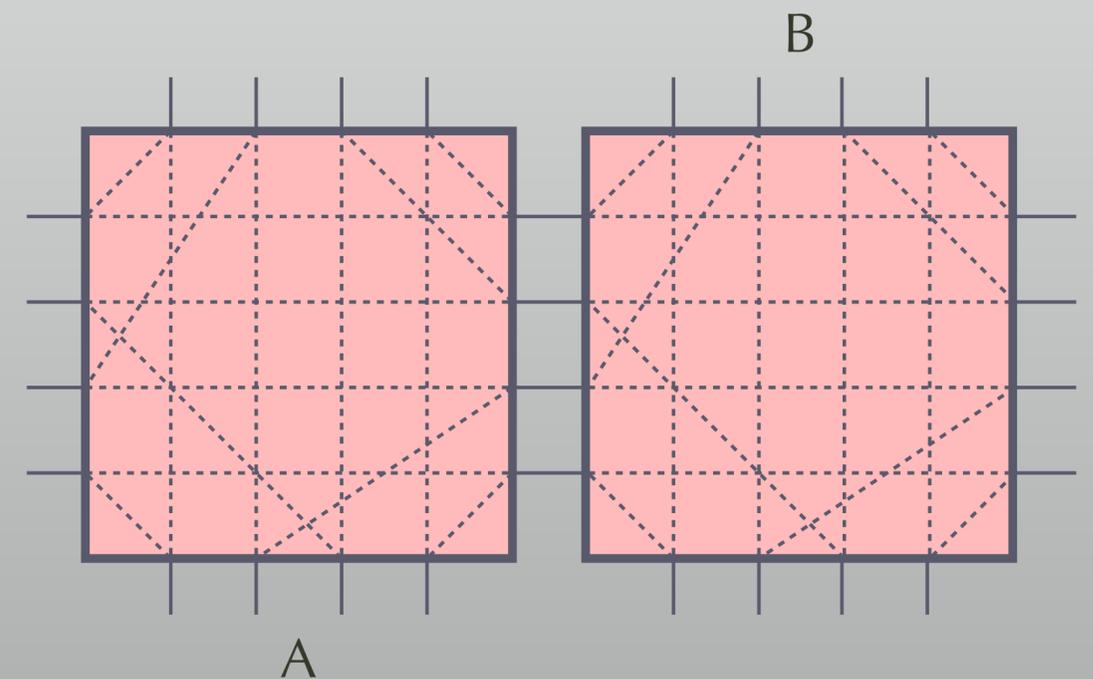
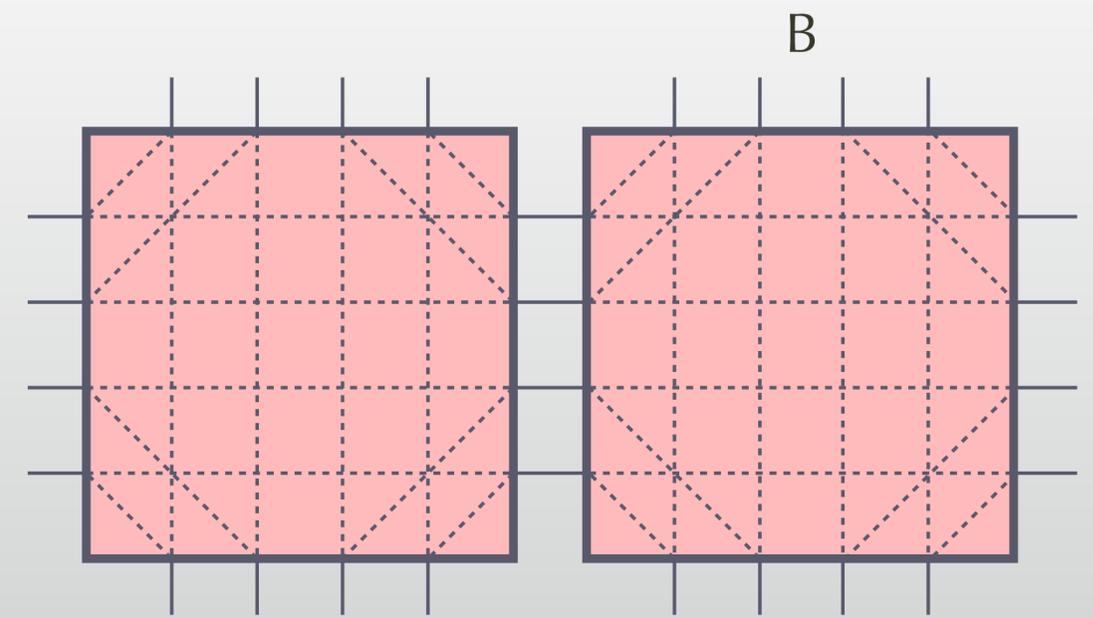
Switch Block

- * 4方向からくる配線を互いに接続
 - * 一本の信号が何カ所につながるか = F_s
 - * 右の例では $F_s = 6$



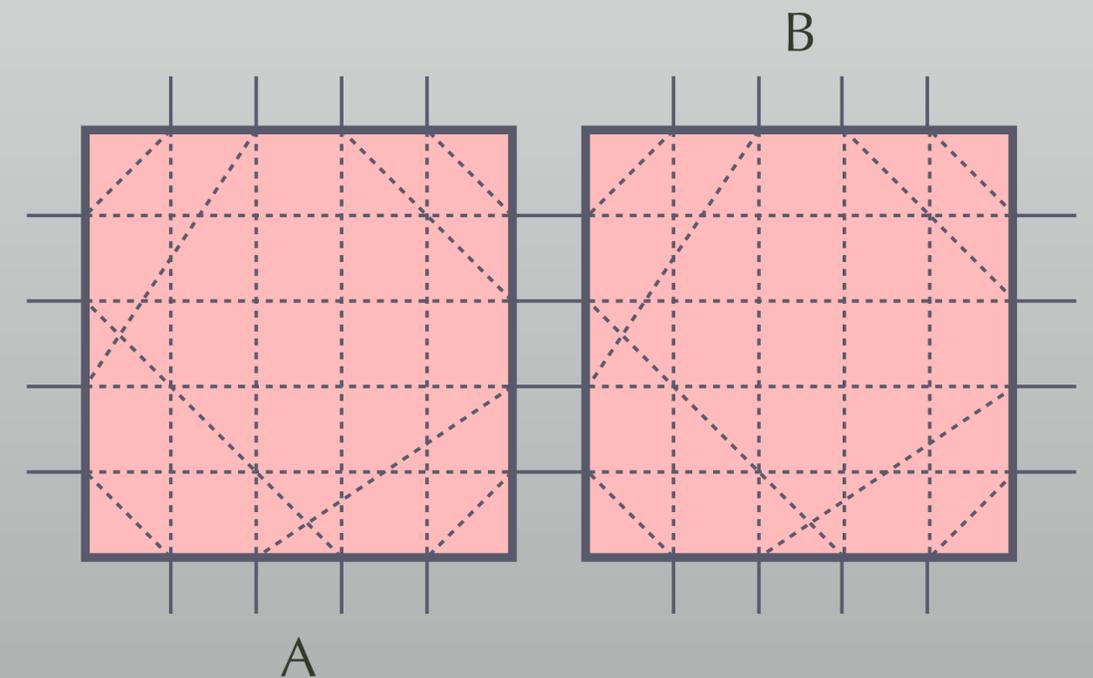
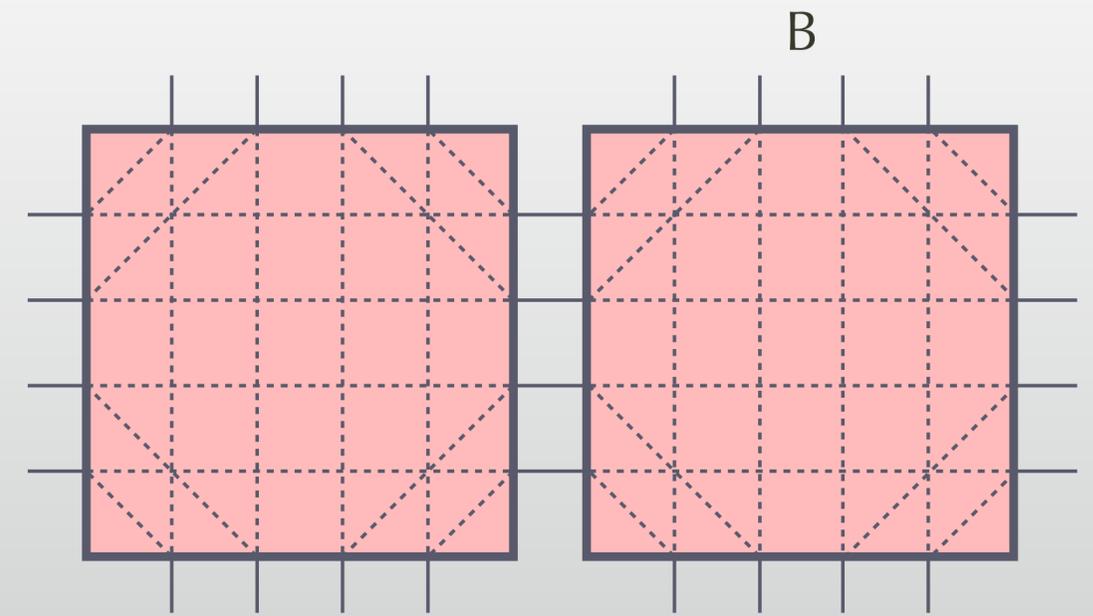
SB Topology

- * どこからどこへ行けるようになるか
(実際にはLBと同じでスイッチの置き方)
- * 下の例はどちらもFs=2だが、
AからBへ接続できるのは下だけ



ターンの制約

- * あるスイッチにおける特定のターンが次のスイッチにおけるターンへの制約にはならないほうがよい



検証

- * 以下は5つの回路を配置配線してみた結果
 - * W, Fs, Fc などを変更
 - * 100%配線可能かどうかを調べる

Fc, Fs と配線可能性

- * Fsがある程度あればFcはそれほど高くなくても配線できる

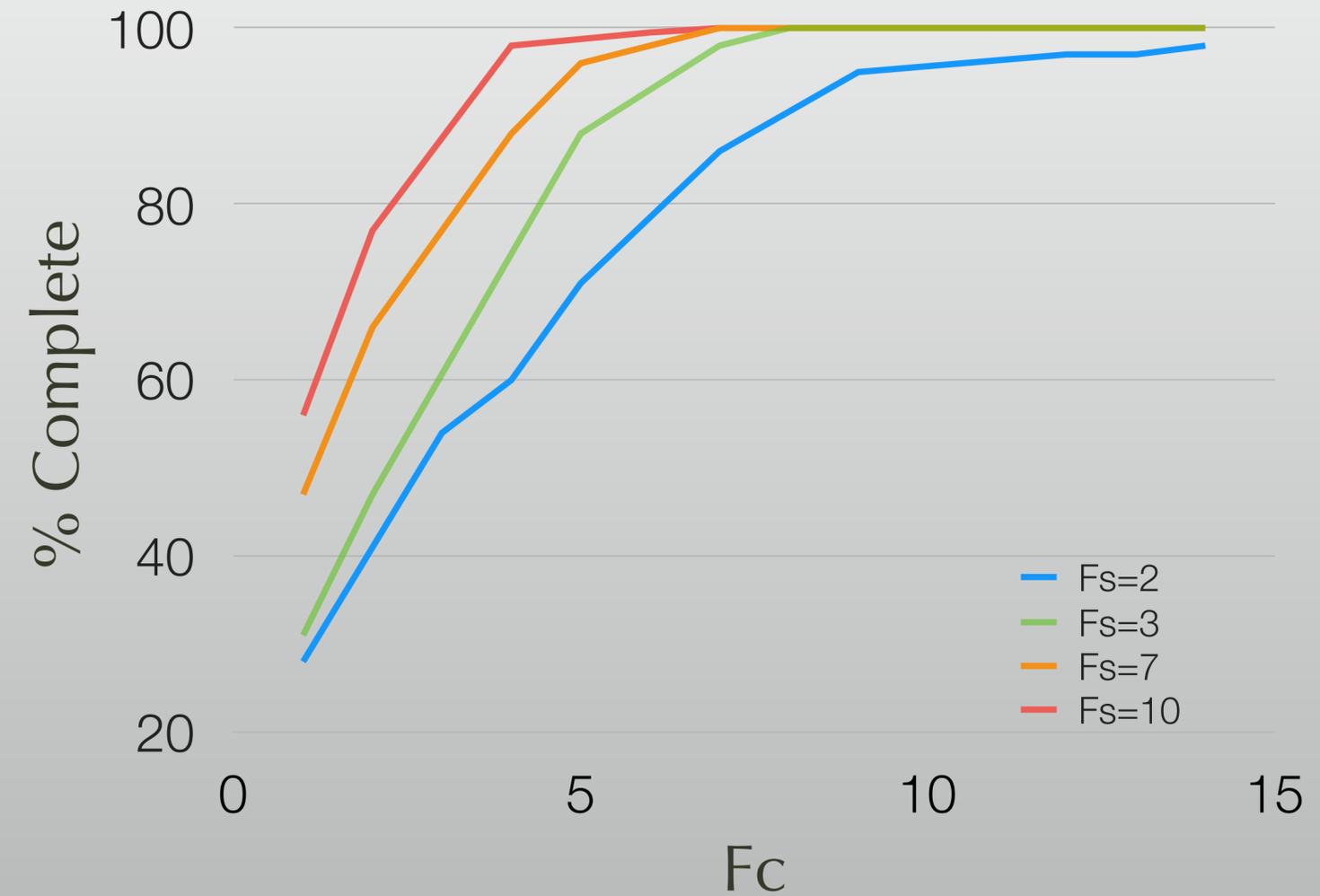


Fig 6.9 Percent Routing vs Fc, Circuit BNRE

Fs, Fcと配線可能性

- * Fcがある程度あればFsはそれほど高くなくても配線できる... あれ？

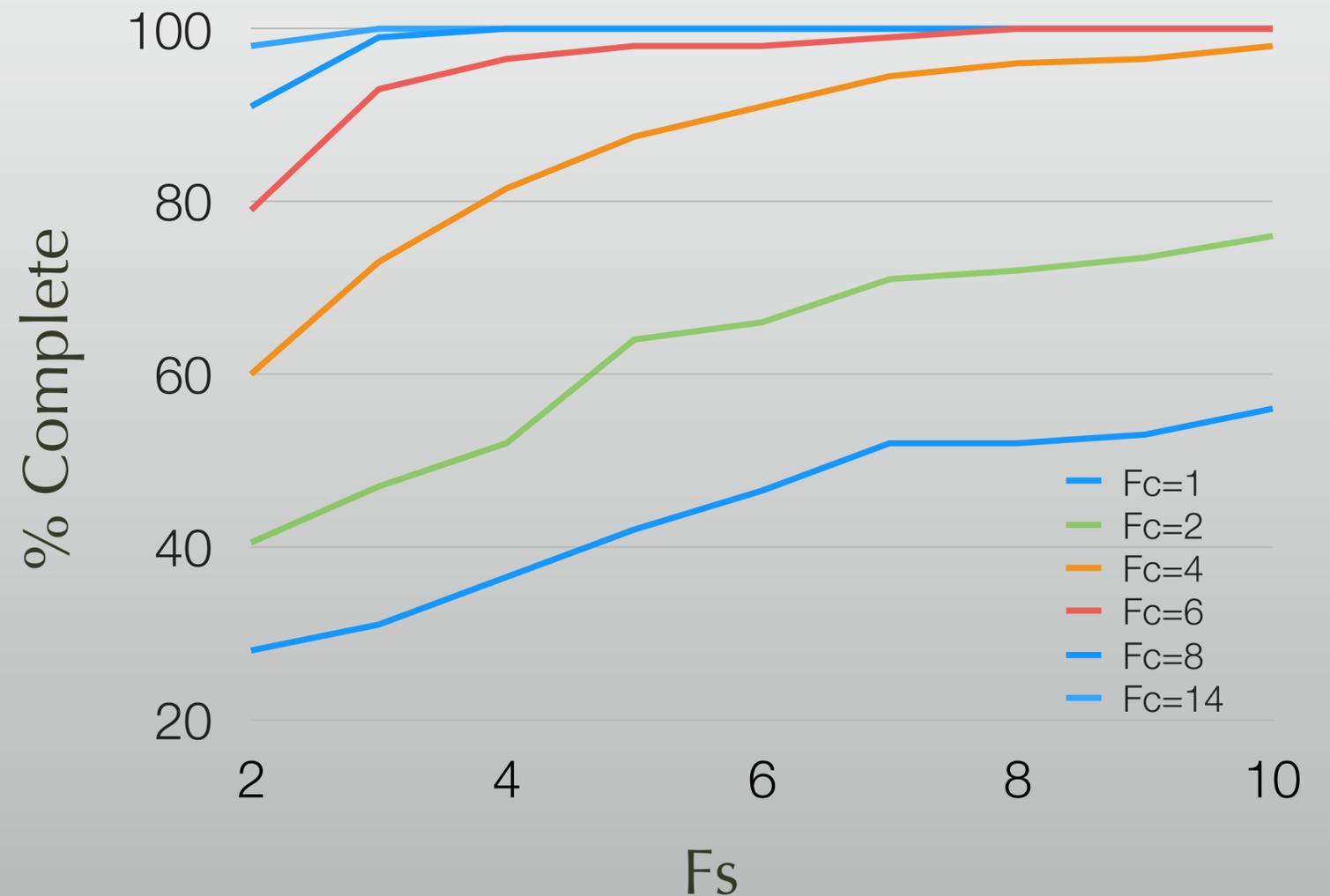
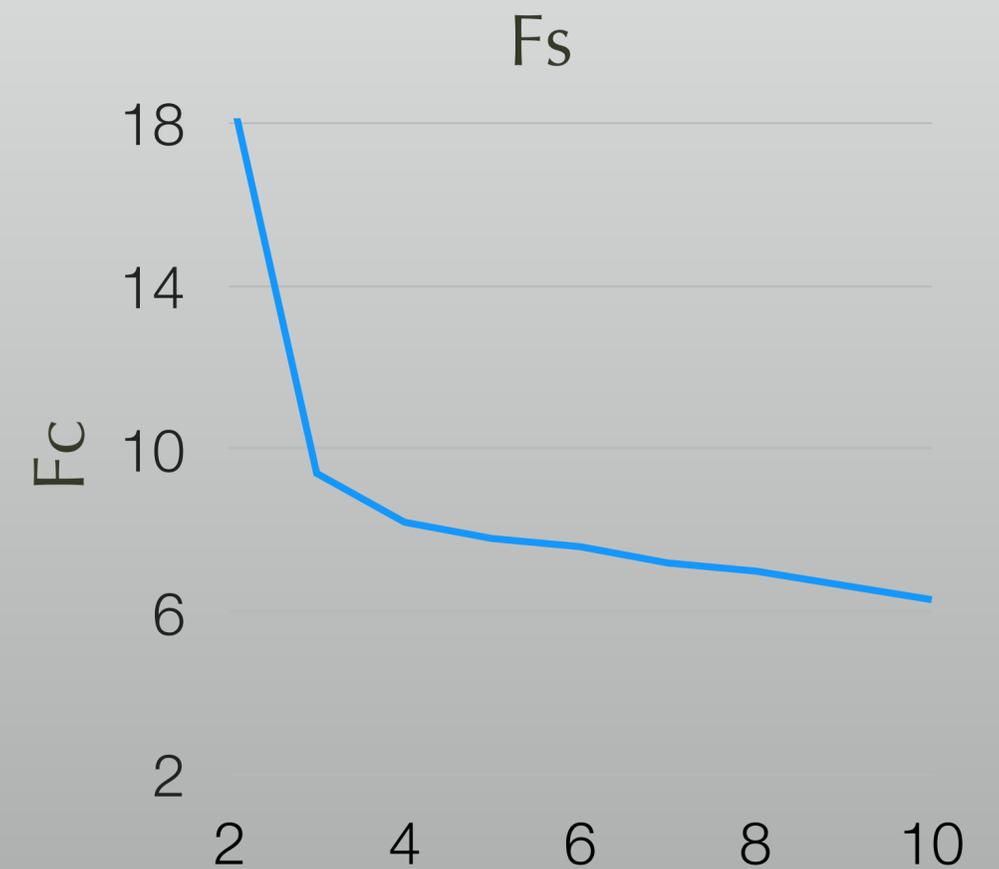
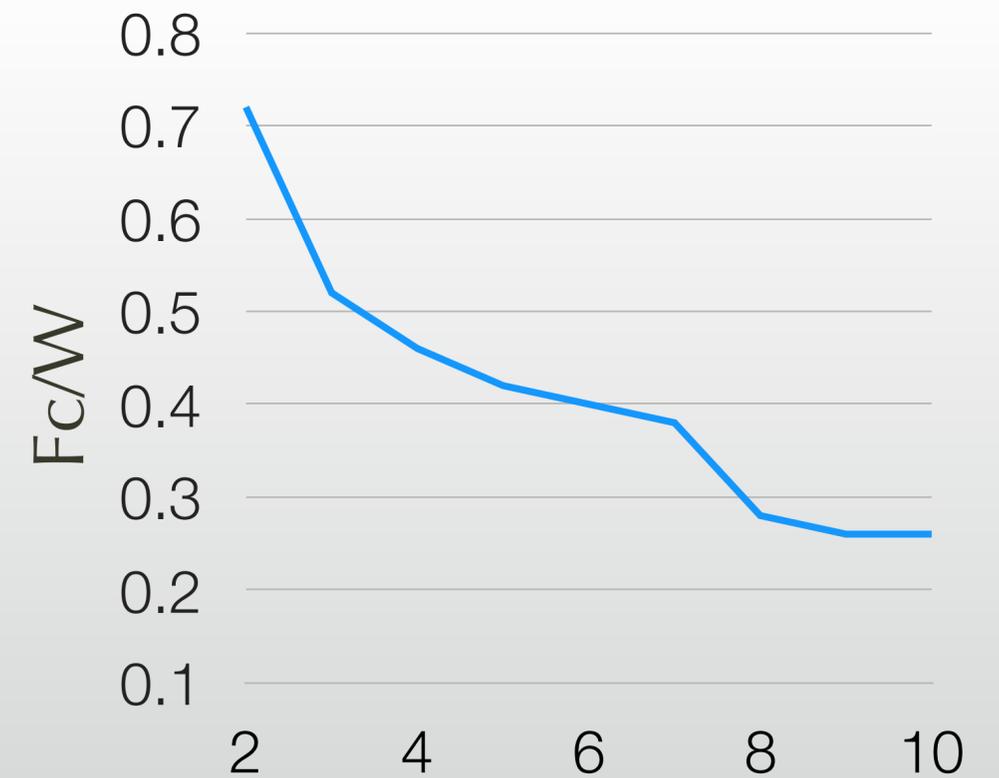


Fig 6.12 % Routing Completion vs Fs, Circuit BNRE

必要なFcとFs

- * FcとFsはトレードオフの関係にあるが、どちらも限界がある
- * F_c/W も大事



FcとFsのバランス

CBにはいくつスイッチが必要？

- * # Switches in CB = $T \cdot P \cdot Fc / 2$
- * # Switches in SB = $2 \cdot Fs \cdot W$

デザイン空間探索

Fc/W

| | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 1 |
|----|-----|-----|-----|-----|------------|-----|-----|-----|
| 2 | nr | nr | nr | nr | nr | 349 | 381 | 306 |
| 3 | nr | nr | nr | 259 | 221 | 223 | 241 | 260 |
| 4 | nr | nr | 270 | 229 | 231 | 249 | 267 | 286 |
| 5 | nr | nr | 306 | 257 | 257 | 254 | 271 | 288 |
| 6 | nr | 369 | 304 | 285 | 305 | 278 | 319 | 338 |
| 7 | nr | 372 | 357 | 313 | 285 | 302 | 319 | 336 |
| 8 | nr | 410 | 345 | 317 | 309 | 326 | 343 | 360 |
| 9 | 510 | 401 | 325 | 343 | 333 | 350 | 367 | 384 |
| 10 | 459 | 409 | 351 | 369 | 357 | 374 | 391 | 408 |

Fs

(検証に使った回路が)
配置配線可能で、なおかつ
一番スイッチの少ない構成

Table 6.5 Average # of Switches per tile for each architecture

Interconnect

配線アーキテクチャ

- * 長さ1だけのwire segmentでは性能が上がらない
 - * どれだけ多くのLBを低遅延で接続できるか
 - * 長さの違うwire segmentを使うのがポイント

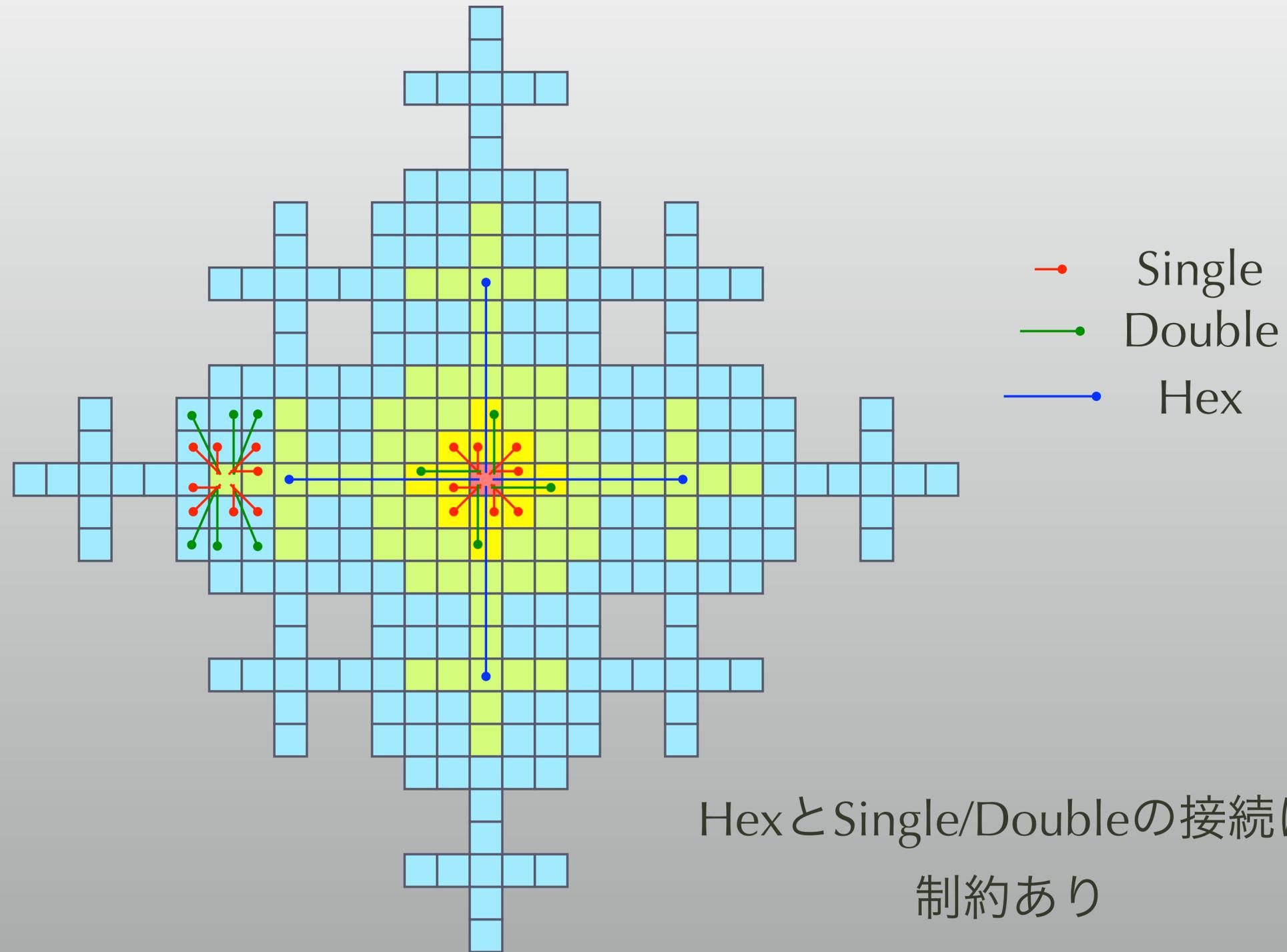
Virtex-4 & 5

- * 正確なことは公式な資料だけではわかりづらい
- * 以下は公開資料からの推測で、かつ説明用
 - * いろいろ省略してあるので注意
- * 自信のない数字も出てきますので「なんとなくこういうもの」程度に

Xilinx Virtex-4

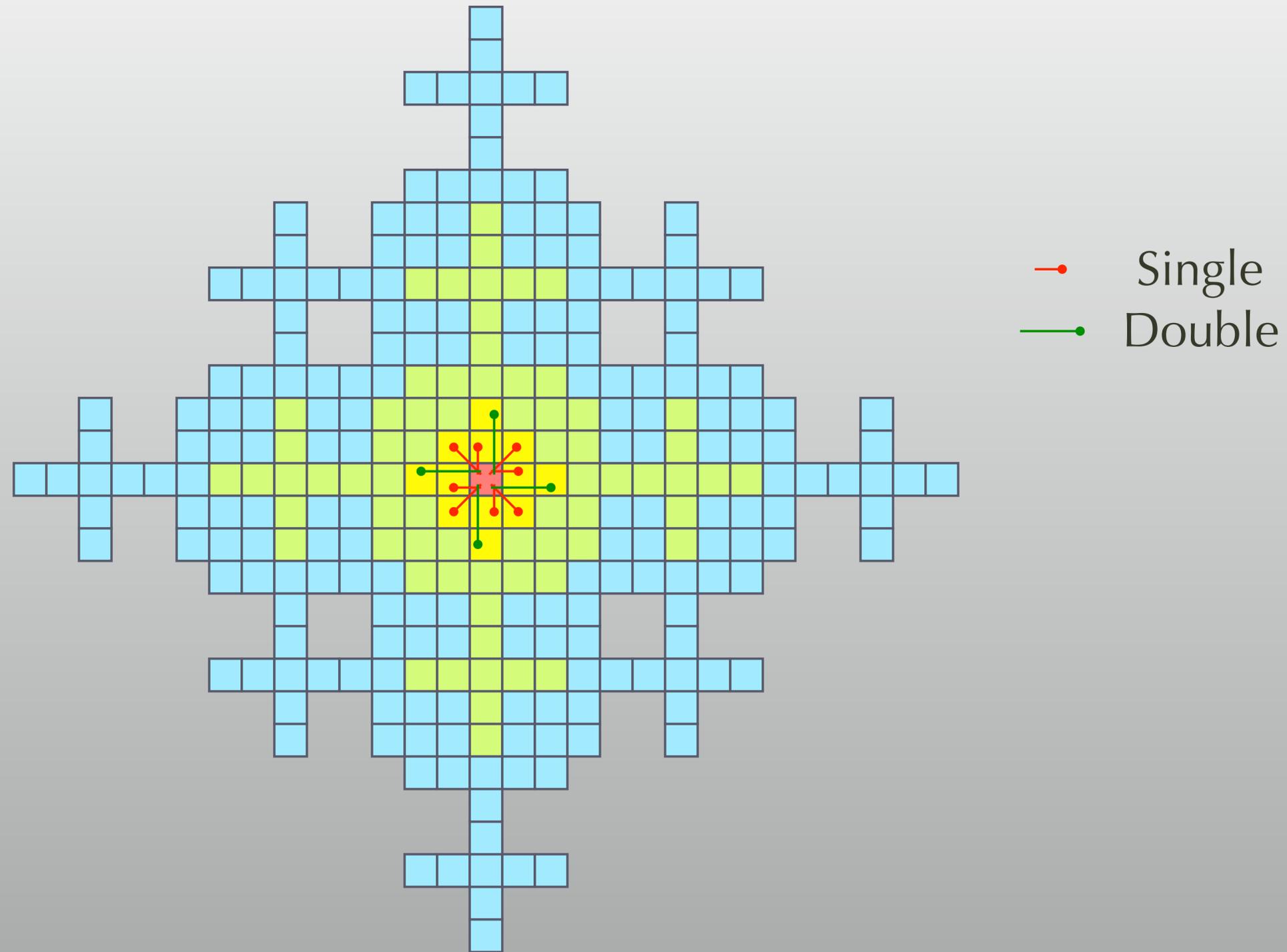
- * Double (2), Hex (6), Long (端から端まで) の3種類
- * DoubleとHexの併用で近隣のLBを低遅延で接続
- * 配線構造としては古典的な手法の延長

Virtex-4 Interconnect



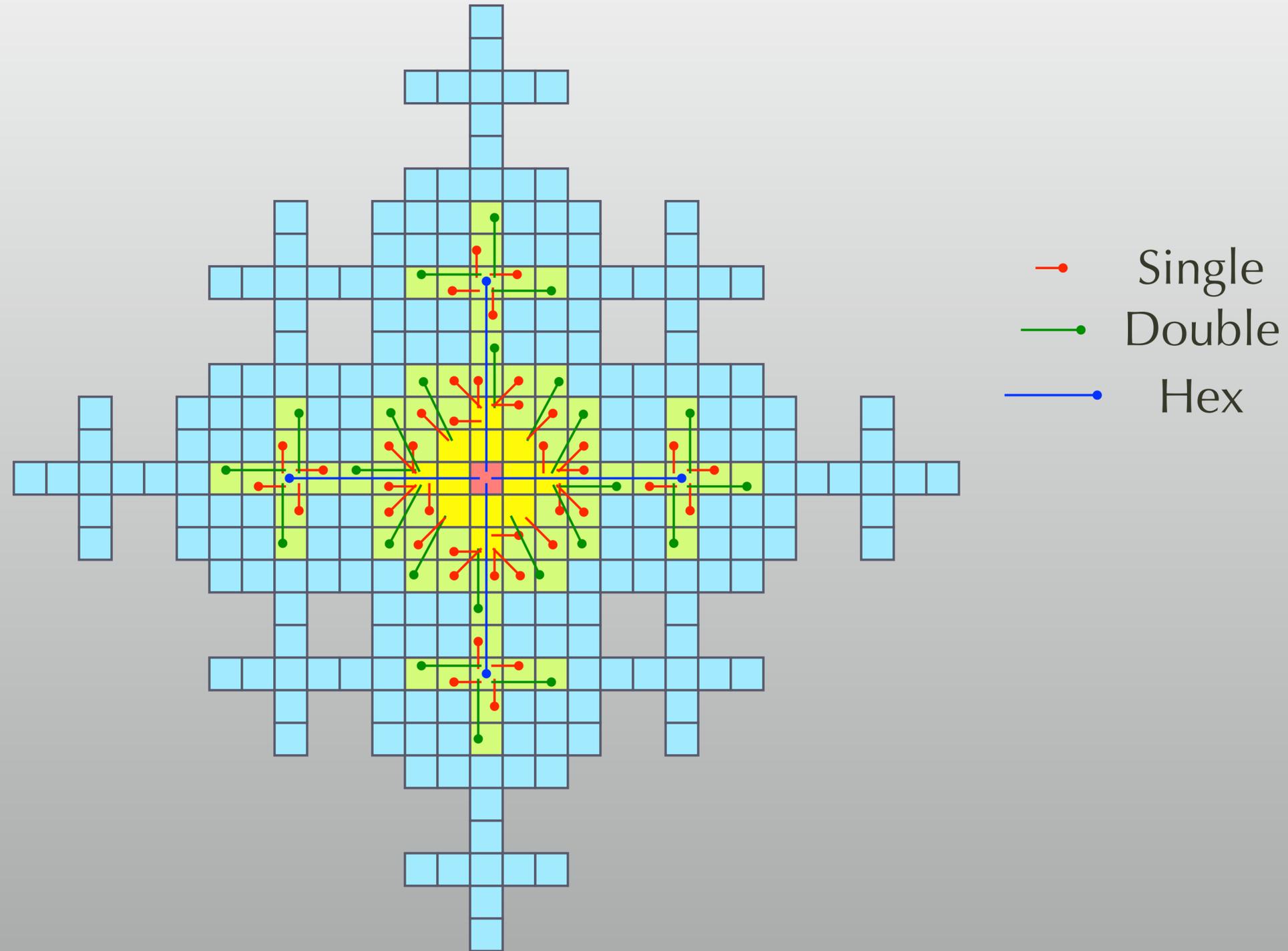
Virtex-4 Interconnect

1 Hop



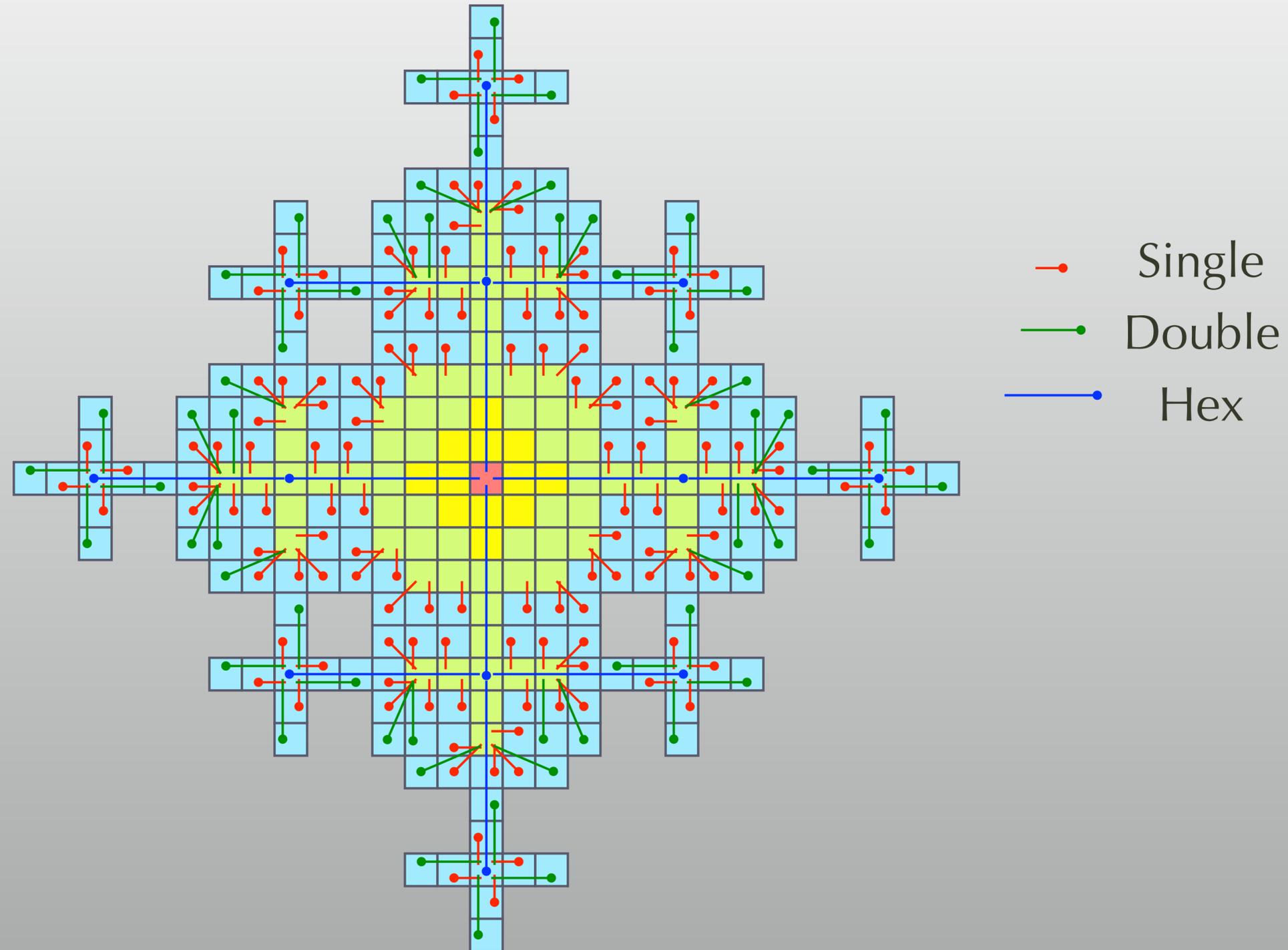
Virtex-4 Interconnect

2 Hops



Virtex-4 Interconnect

3 Hops

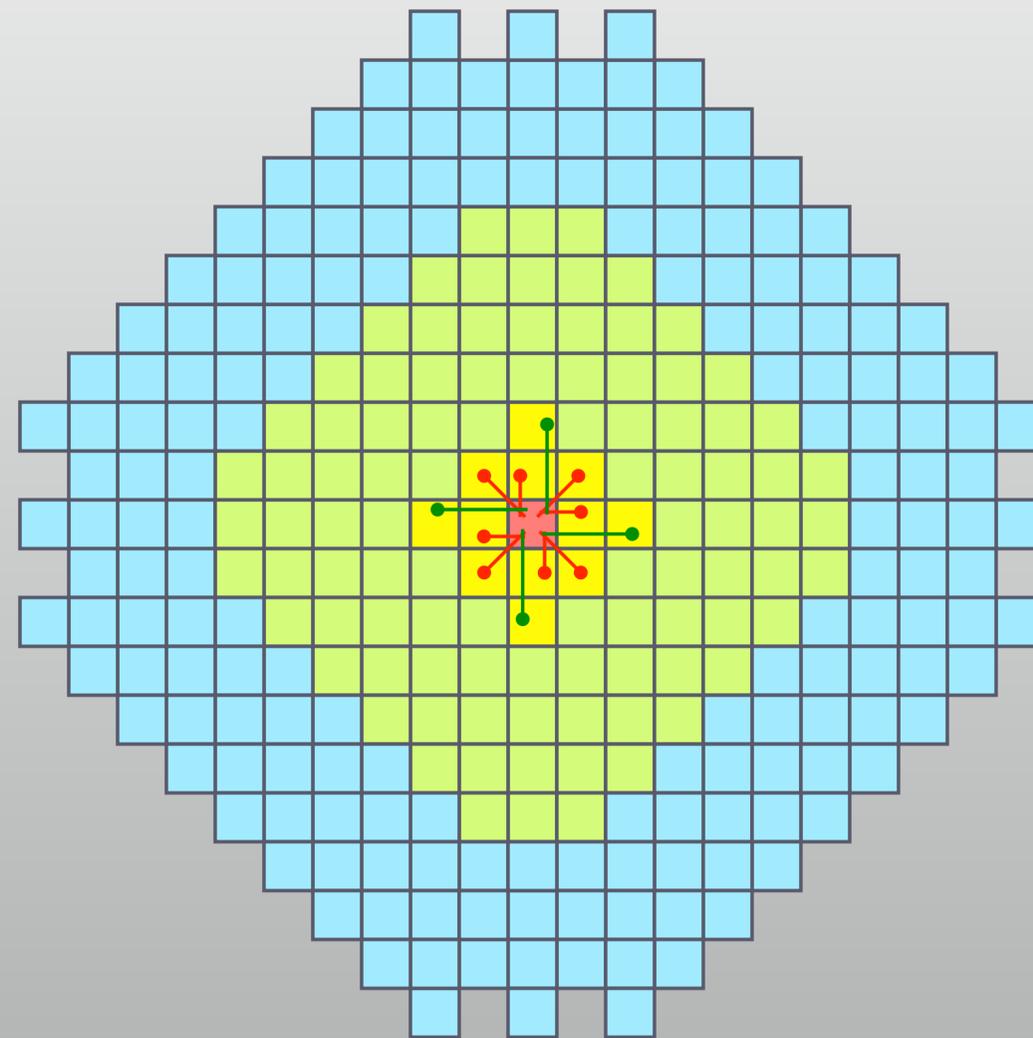


Xilinx Virtex-5

- * 6-LUT を採用
 - * 単純に考えて 50% 多くの配線が必要
 - * L字型の配線 (Pent) を導入、Hex はなくなった
 - * より多くの近隣のLBを接続

Virtex-5 Interconnect

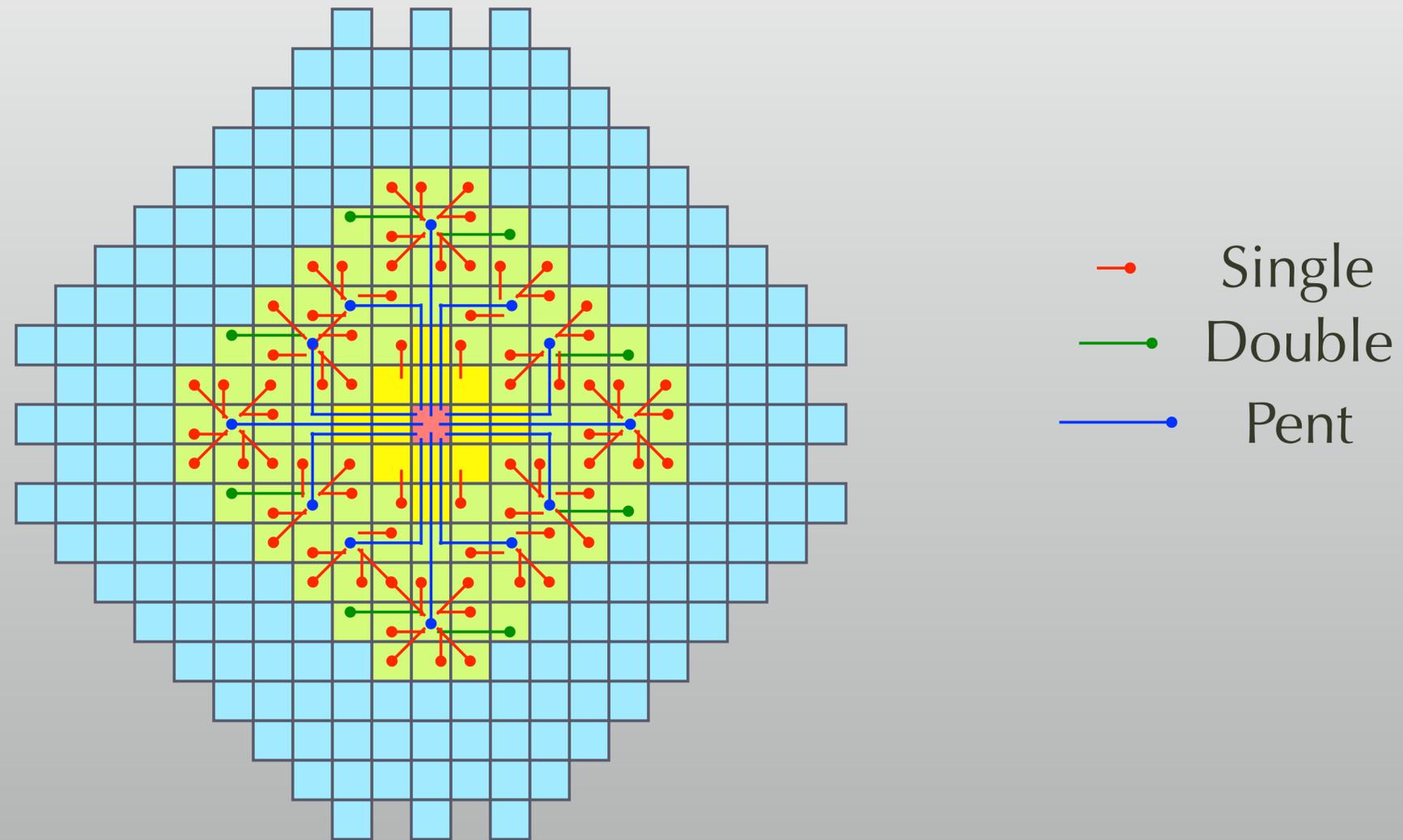
1 Hop



- Single
- Double
- Pent

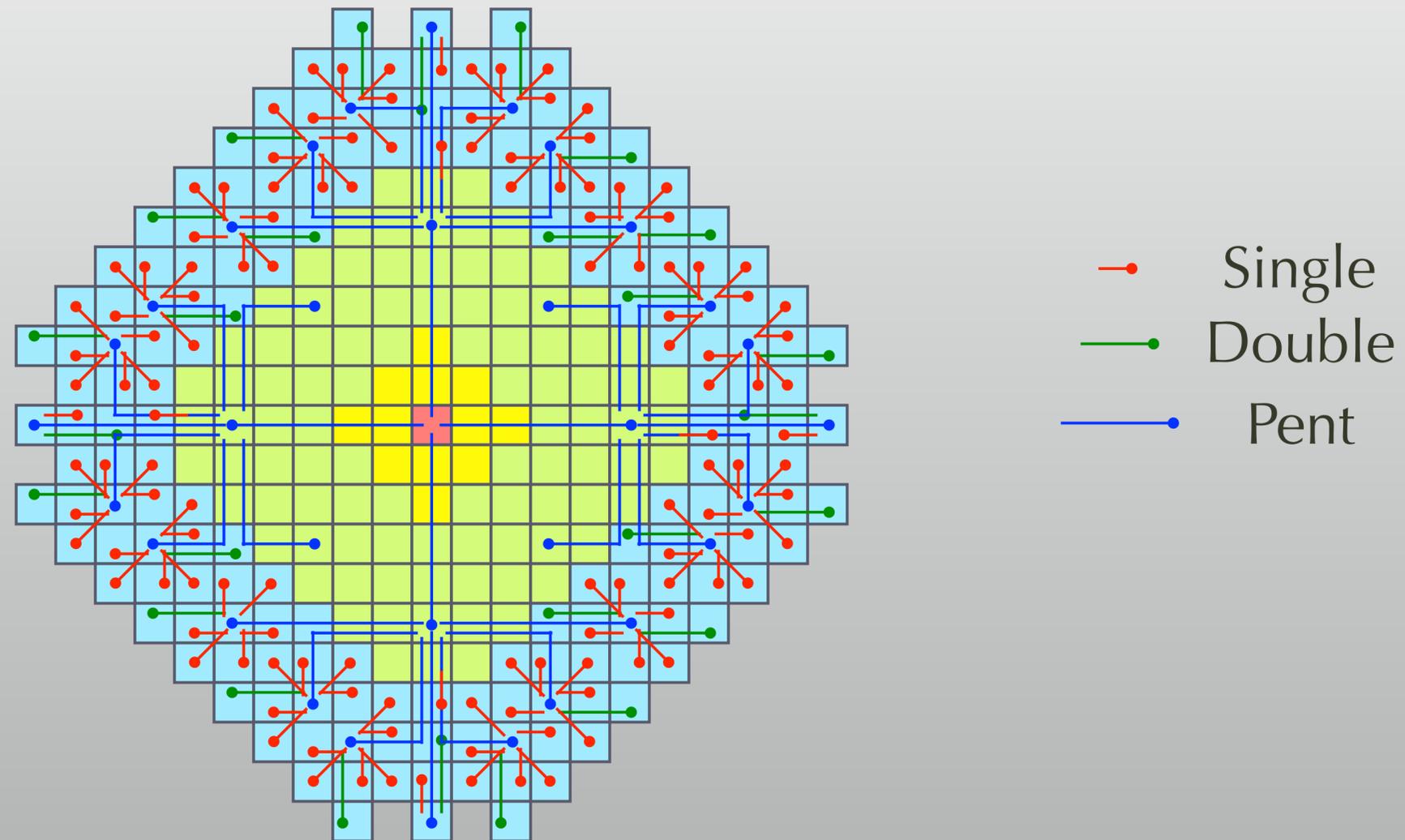
Virtex-5 Interconnect

2 Hops



Virtex-5 Interconnect

3 Hops



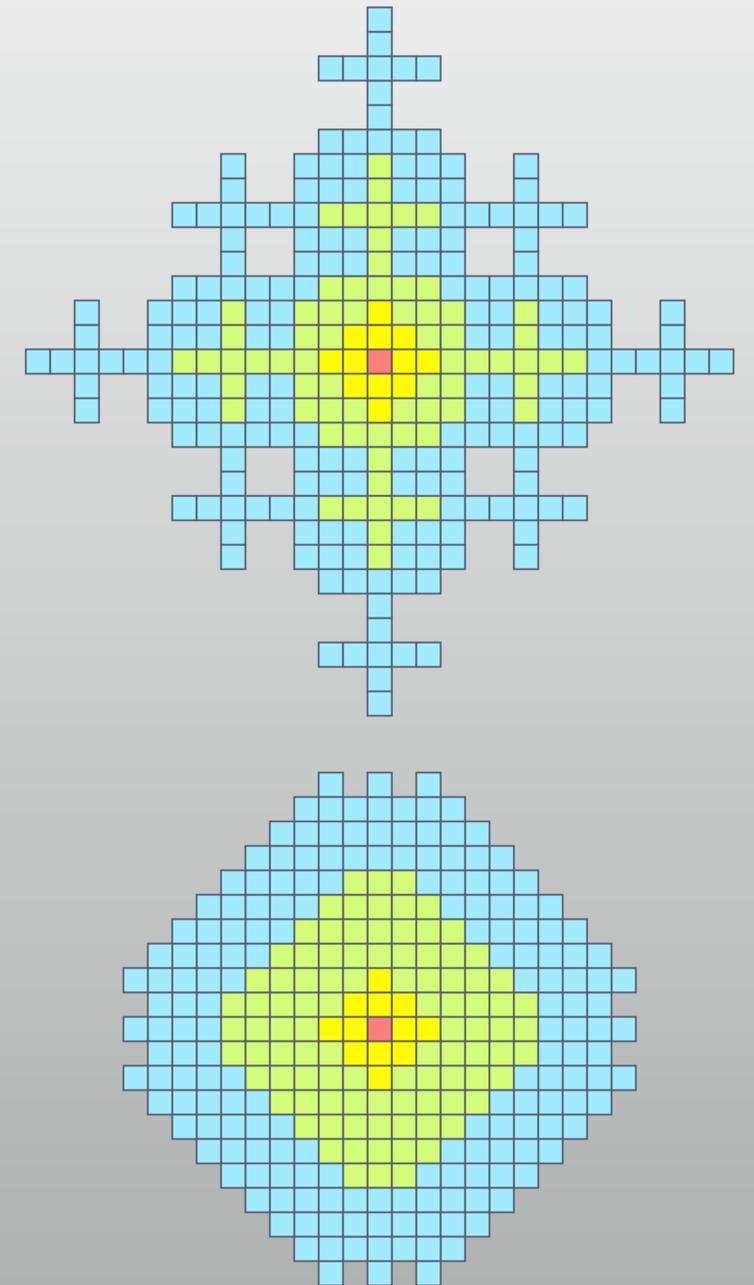
比較

Reachable CLBs

| | Virtex-4 | Virtex-5 |
|--------|----------|----------|
| 1 Hop | 12 | 12 |
| 2 Hops | 68 | 96 |
| 3 Hops | 200 | 180 |
| Total | 280 | 288 |

Wire segments

| | Virtex-4 | Virtex-5 |
|--------|----------|----------|
| Double | 40 | 42 |
| Hex | 120 | - |
| Pent | - | 120 |
| Long | 24 | 18 |
| Total | 184 | 180 |



配線は増やせる？

- * 配線技術の進歩によって配線層が増加、配線の自由度が向上
 - * Virtex-II 150nm 6層メタル (2001)
 - * Virtex-II Pro 130nm 7層メタル (2002)
 - * Virtex-4 90nm 10層メタル (2004)
 - * Virtex-5 65nm 11層メタル (2008)
 - * Virtex-6 40nm 12層メタル (2009)
 - * Virtex-7 28nm (2011)