

# FPGAの構造

KEK IPNS E-sys  
本多良太郎

## Xilinx Kintex-7 FPGA (XC7K160T-2FFG676)



私が一番詳しいのがKintex-7なので  
この講義ではKintex-7を中心に進めます

**構造や性能はFPGAファミリーごとに異なるので実際の設計の際にはかららずDSとUGを参照してください！**

### FPGAのバンク構造と信号規格

- 利用可能なIO規格と信号スピードの違い
  - High performance bank (Spartan7, Artix7には存在しない)
  - High range bank
  - Multi-gigabit transceiver (MGT) bank
- こんな時どうする？よくある事例と解決策の例

FPGAをやる人は回路図も最低限読めてほしい

### FPGAの内部構造

- どういう専用タイルが存在するのか
  - IO tile
  - CM tile
  - DSP tile
  - RAMB tile
  - CLB tile (汎用ロジックスライス)
- クロックツリー

### FPGAの動作速度とは何か？

- BUFGの動作周波数
- IOピンの最大データレート
- トランシーバの最大データレート

#### 略称

CM	Clock Management
CLB	Configurable Logic Block
RAMB	Block RAM
DSP	Digital Signal Processor
GT	Gigabit Transceiver

# FPGAのバンク構造と 信号規格

# FPGAのバンク構造

XC7K325T, XC7K410T  
FF/FFG900のケース  
(UG475, Fig.3-118)

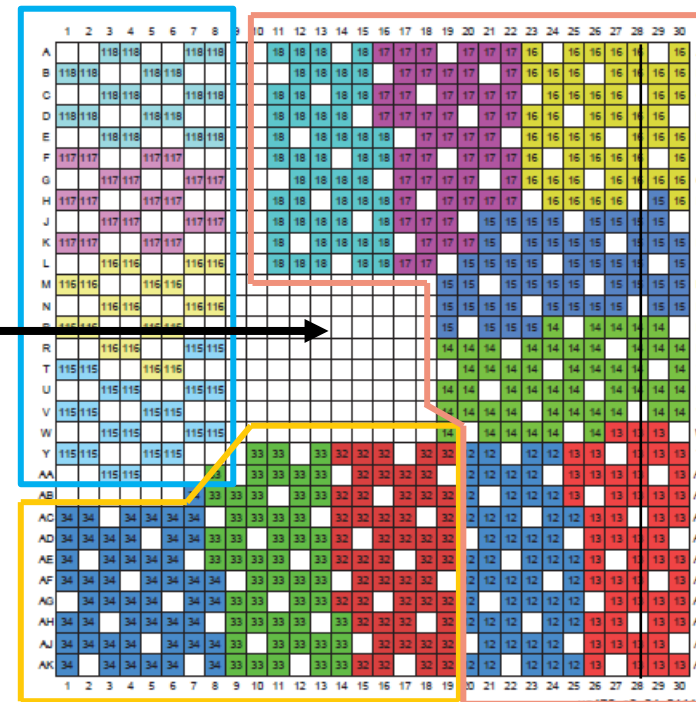
## MGT bank

- 高速シリアルトランシーバチャンネルを利用するための専用バンク

## Configuration bank (bank0)

- JTAGなどでFPGAを初期化するための専用ピンが集まっている
- SPIフラッシュ用の専用ピンは一部 bank14, 15にも存在

Bank 0の駆動電位を決めると自動的に bank14, 15のIO電位が決まる  
(正確にはコンフィギュレーション中の)



## High performance bank

- 高速、高クオリティの信号をやりとり
- Kintex-7とVirtex-7に存在

## この情報はどこにある？

UG475: 7 シリーズ FPGAパッケージおよびピン配置  
DS182: DC 特性および AC スイッチ特性

## High range bank

- 3.3Vまでのスイングの大きい信号規格をサポート

## 電源

- |                |                             |
|----------------|-----------------------------|
| • VCCINT       | コア電源                        |
| • VCCBRAM      | ブロックRAM電源                   |
| • VCCAUX       | 補助電源                        |
| • VCCAUX_IO    | 補助電源                        |
| • VCCO         | IOバンク電圧                     |
| • VREF         | 入力基準電圧                      |
| } バンクごとに設定     |                             |
| • MGTAVCC      | } GTトランシーバ用電源<br>値はほぼ決まっている |
| • MGTAVTT      |                             |
| • MGTVCCAUX    |                             |
| • MGTAVTTTRCAL |                             |
| • MGTTRREF     |                             |

## High performance (HP) bank

- 入出力が可能なデータレート(繰り返し速度)が高い
- VCCO=2.0Vまで(推奨1.8Vまで)
  - 差動信号の入出力に使う事が多い
  - メモリ (DDR3 SDRAMなど) アクセスはHP bankで行う事が普通

## High range (HR) bank

- HP bankよりも許容データレートが低い
- VCCO=3.6Vまで(推奨3.3Vまで)
  - 3.3Vや2.5V系のICの制御はこのバンクでやる事が多い
  - LVDS出力を行う場合VCCOは**2.5V**である必要がある。
  - 差動信号を受けるだけならHR bankでもよい(後述)

## 良く使う単一端信号 (DS182, 表9)

IO std	VIL		VIH		VOL	VOH	Drive current
	Min	Max	Min	Max	Max	Min	
LVCMS18	-0.3	35%Vcco	65%Vcco	Vcco+0.3	0.45	Vcco-0.45	Selectable
LVCMS33	-0.3	0.8	2.0	3.45	0.4	Vcco-0.4	Selectable
LVTTL	-0.3	0.8	2.0	3.45	0.4	2.4	Selectable

DS182: DC 特性および AC スイッチ特性

信号スイング幅やHIGH, LOWと認識する電位がVCCOに依存して決まっている。

LVCMS25,33はHR bankでしか使えない

HSTL, SSTLはメモリ (SDRAM等) 用。ここでは省略

ドライブカレントは変更可能

## LVDS, LVDS\_25規格 (同表12,13)

LVDS: HP bankでのみ利用可能  
LVDS\_25: HR bankでのみ利用可能

電気的には同一

Symbol	DC parameter	Condition	Min	Typ	Max	Unit
VODIFF	Output (Q-/Q)	100Ω termination	247	350	600	mV
VOCM	Common mode output voltage	100Ω termination	1.000	1.250	1.425	V
VIDIFF	Input (Q-/Q)		100	350	600	mV
VICM	Common mode input voltage		0.300	1.200	1.500	V

LVDS: VCCO=1.8V  
LVDS\_25: VCCO=2.5V

他にも...

BLVDS:

- 複数のエンドポイントを持てる差動信号規格。
- ドライブカレントが大きく(10 mA), 振幅も大きい。

## 出力

- LVDS, LVDS\_25の指定電圧 (1.8V, 2.5V) でないと利用不可

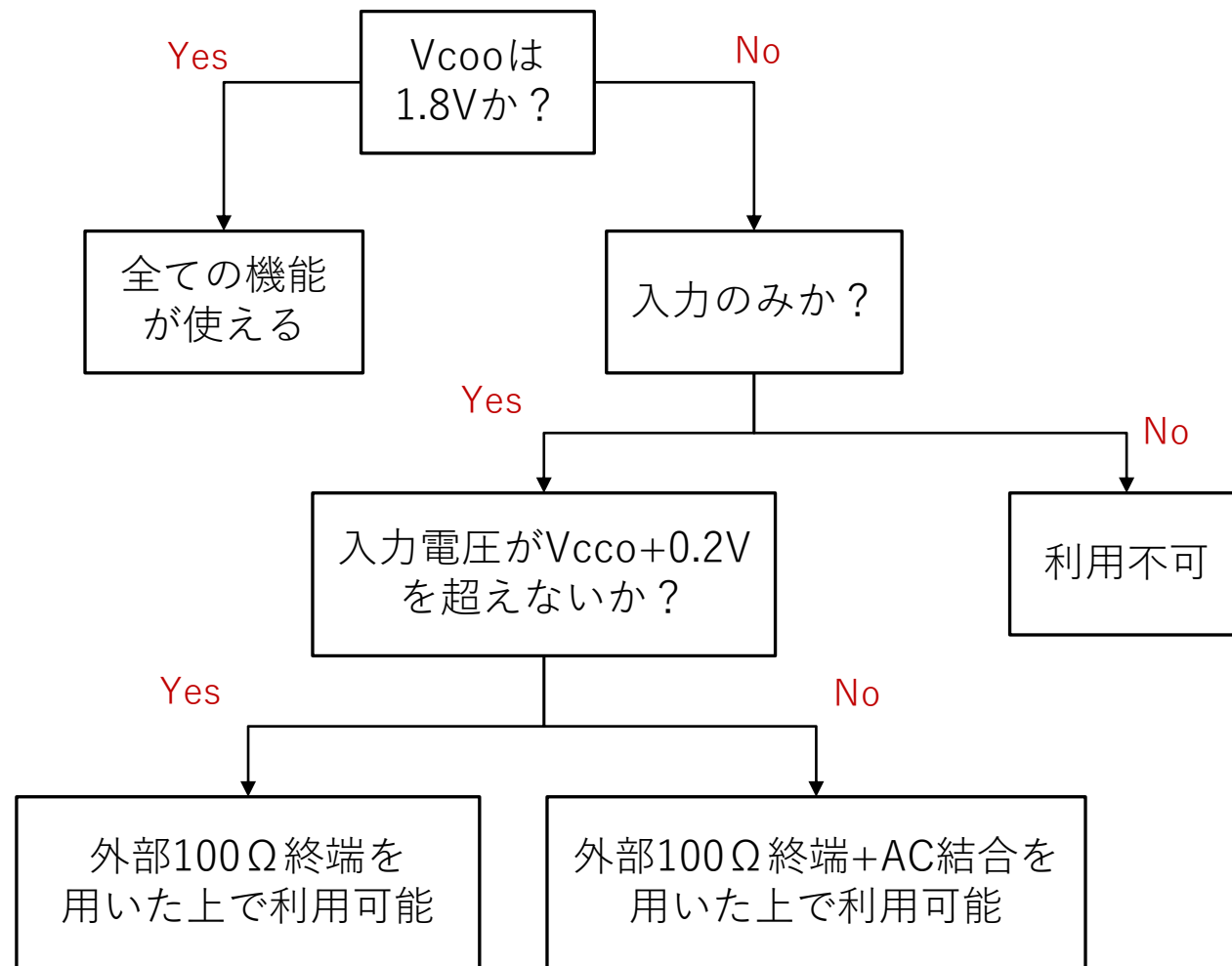
## 入力

- 指定電圧だとDIFF\_TERMが利用可能
- 指定電圧外だと外部終端が必要
- 同相電圧によってはAC結合が必要

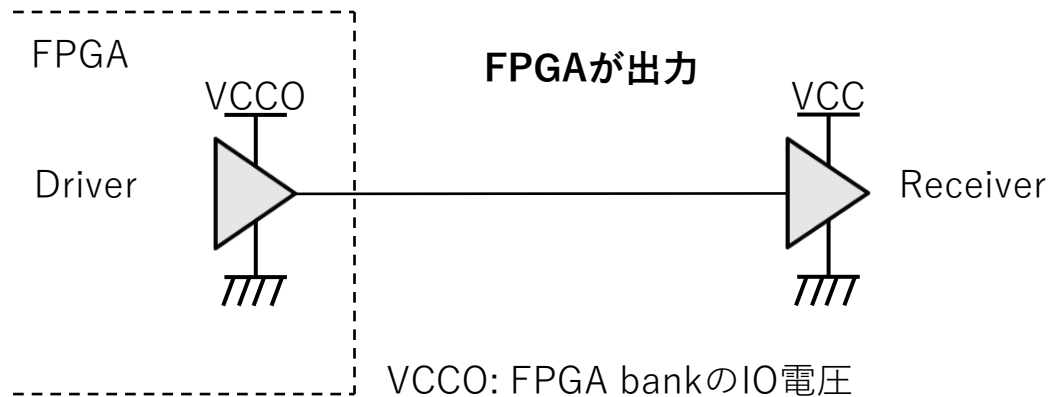
### DIFF\_TERM:

- FPGAにビルトインされた100Ω並列終端抵抗を利用する方法。外部抵抗が不要なくスタブが最小となるため高速信号では利用が推奨される。

HP bankの場合



Xilinx AR (Answer Record) 43989

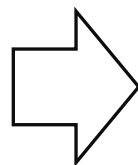


## 基本的なポイント

- 信号規格を入出力で揃える
  - VCCOとVCCが同じになることが殆ど
- DC結合にする

## もう少し進むために必要な知識

- 入力電圧はIC毎に絶対定格が決まっており電源電圧に依存する。
- 閾値電圧は信号規格によって決まっている。



## Single-end (単一端)なら

- 入力側で $V_{OH}$ がHIGH状態の閾値電圧からVCCの間に入っていればよい。

## Differential (差動)なら

- $V_{OD}$ がと入力側で認識可能な範囲に入っていればよい
- $V_{CM}$ が入力側で受け入れ可能な範囲に入っていればよい

\*\*\*Drive currentを今の議論では無視している。

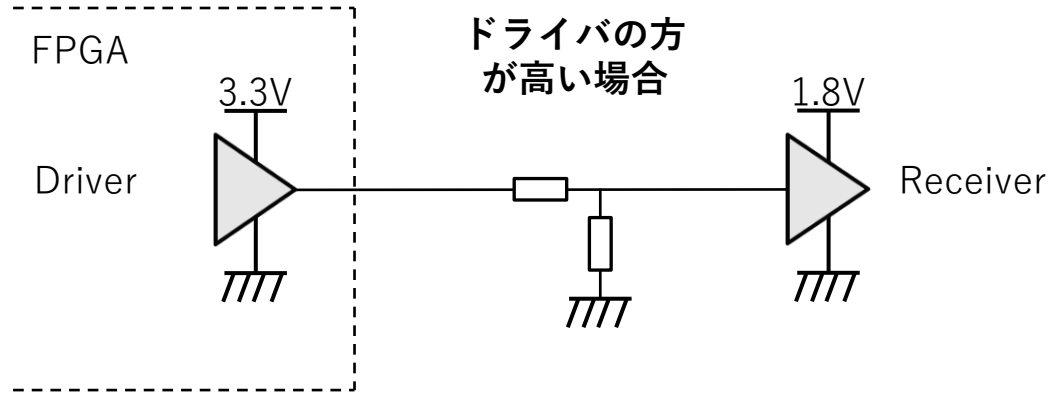
$V_{OH}$ : HIGH状態の出力電位

$V_{OD}$ : 差動信号のp/n間の電位差

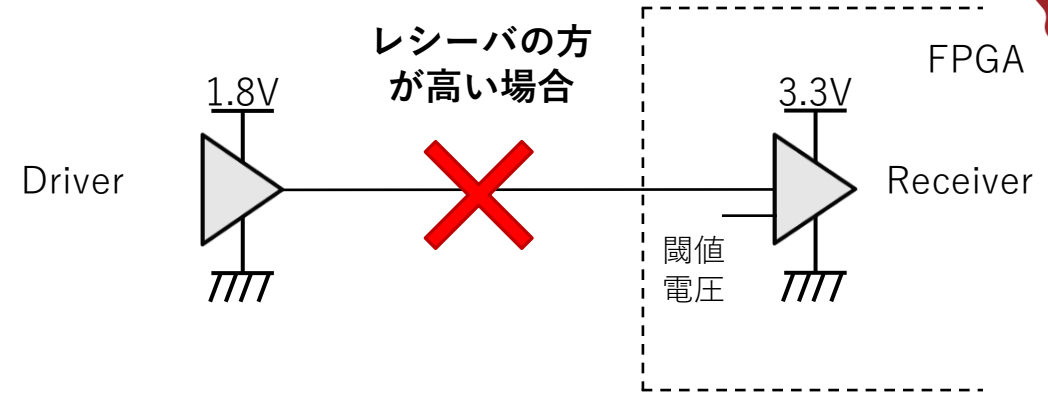
$V_{CM}$ : 差動信号のp/n間の平均電圧 (common-mode)



# 信号規格が合わない場合（単一端）



抵抗分圧で降圧させる方法がある  
(レベルシフタICで変換するのもOK)

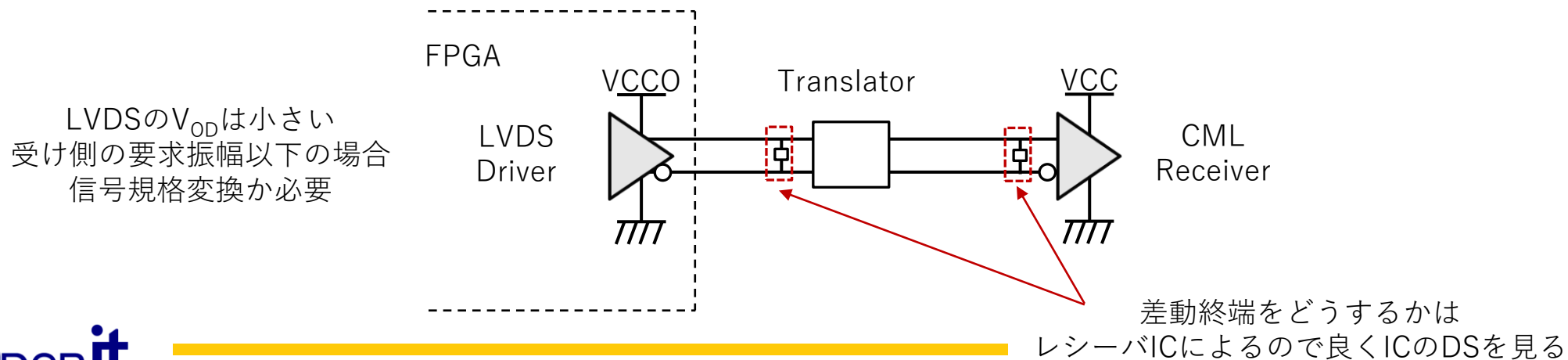
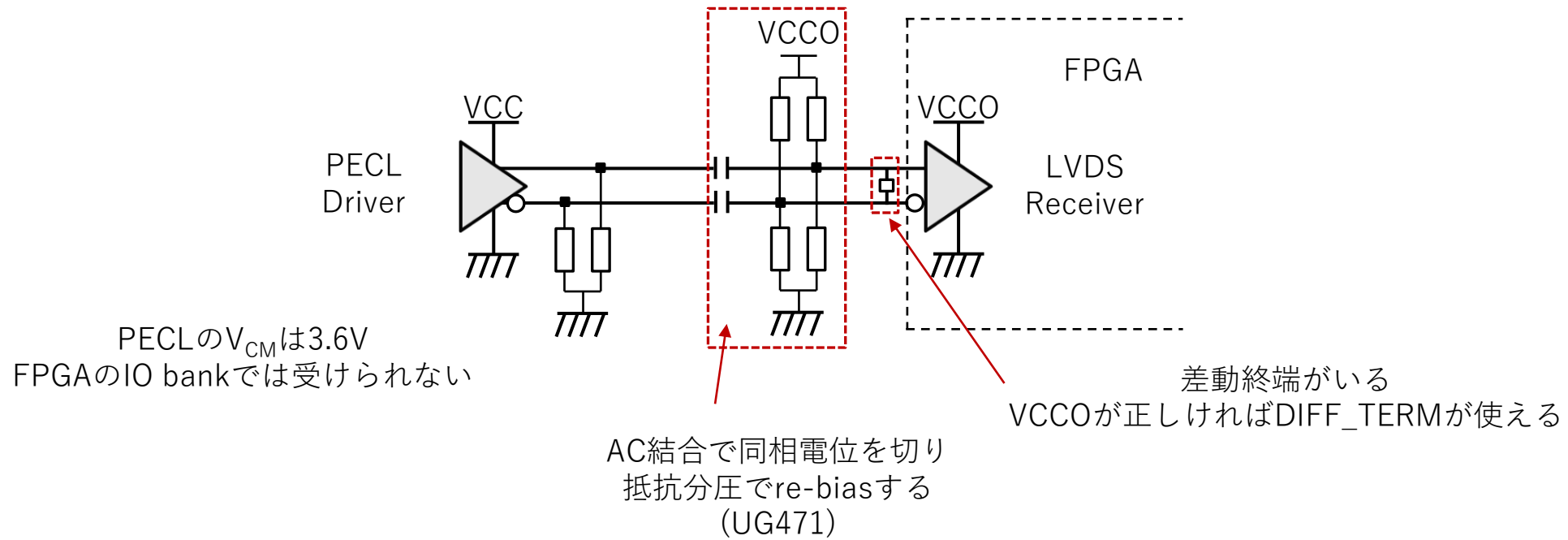


レシーバがHIGHを認識できない  
受動素子で昇圧は不可能

レベルシフタICやトランジスタで昇圧

FPGAのIOバンクを柔軟に利用可能になり配線がやりやすくなる

# 信号規格が合わない場合（差動信号）

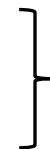


## 高速シリアル通信専用のバンク

- 送信 (TX)/ 受信 (RX) の差動ペアで1セット
- **1つの転送ラインで数Gbpsのラインレート**
  - FPGAファミリ, スピードグレード, パッケージに強く依存！
- 使い方のルールが決まっており変な使い方は出来ない。
  - DS182に色々書いてあるが推奨回路通りつなぐしかない

素核で良く使うのは…

- 1000BASE-SX/LX, 10GBASE-SR/LR (Ethernet)
  - 光ファイバーモジュール
- PCI express
- 最近の高速ADCはシリアル通信でデータを出す事もある



どちらも物理層はCML

自作の回路をトランシーバに接続するのは中級編の範疇を大きく超える

# FPGAの内部構造 (各種タイルの概要)

## リソース, ブロック

- 具体的な機能が割り当てられていない物理的資源
- XilinxのUGではリソース付きで呼ばれるか、IO blockなどと書かれる。

## TILE, SITE, BEL

- 物理的構造を表す言葉 (XilinxではこれらをCLASSと呼んでいる)
- TILEが一番大きく右に行くほど内部構造のCLASS

## SLICE

- 論理ブロックの単位
- リソースの物理的大きさで見るとSITEと殆ど同じ

## Primitive

- 専用リソースに特定の機能を実体化 (instantiate) させた状態
- 例: ILOGICはIDDRプリミティブにもなれるしISERDESプリミティブにもなれる

## Cell

- ある機能 (ロジック) を実現するためのインスタンス
- 通常HDLでつけたラベル名で識別される
- ある機能を実現するインスタンスが複数リソースにまたがって実装される場合全部まとめてCell

## Leaf Cell

- それ以上下位のインスタンスを生成しないセル。機能ツリーの端点、葉っぱ。

## Net

- 配線名

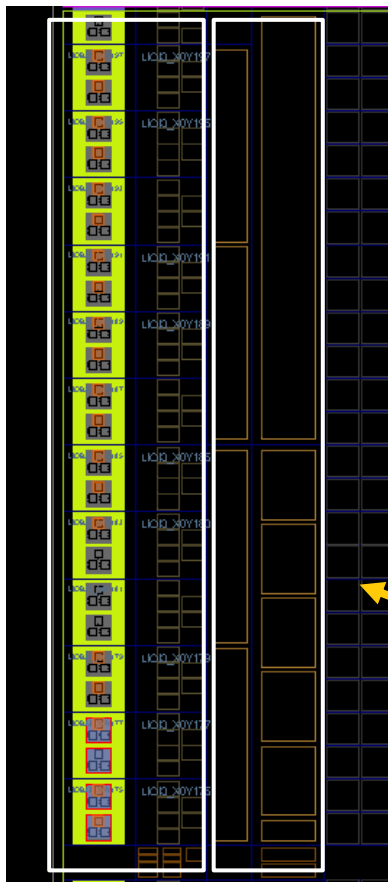
UG912により詳しい説明が載っています。

# Xilinx 7シリーズ FPGAのフロアレイアウト

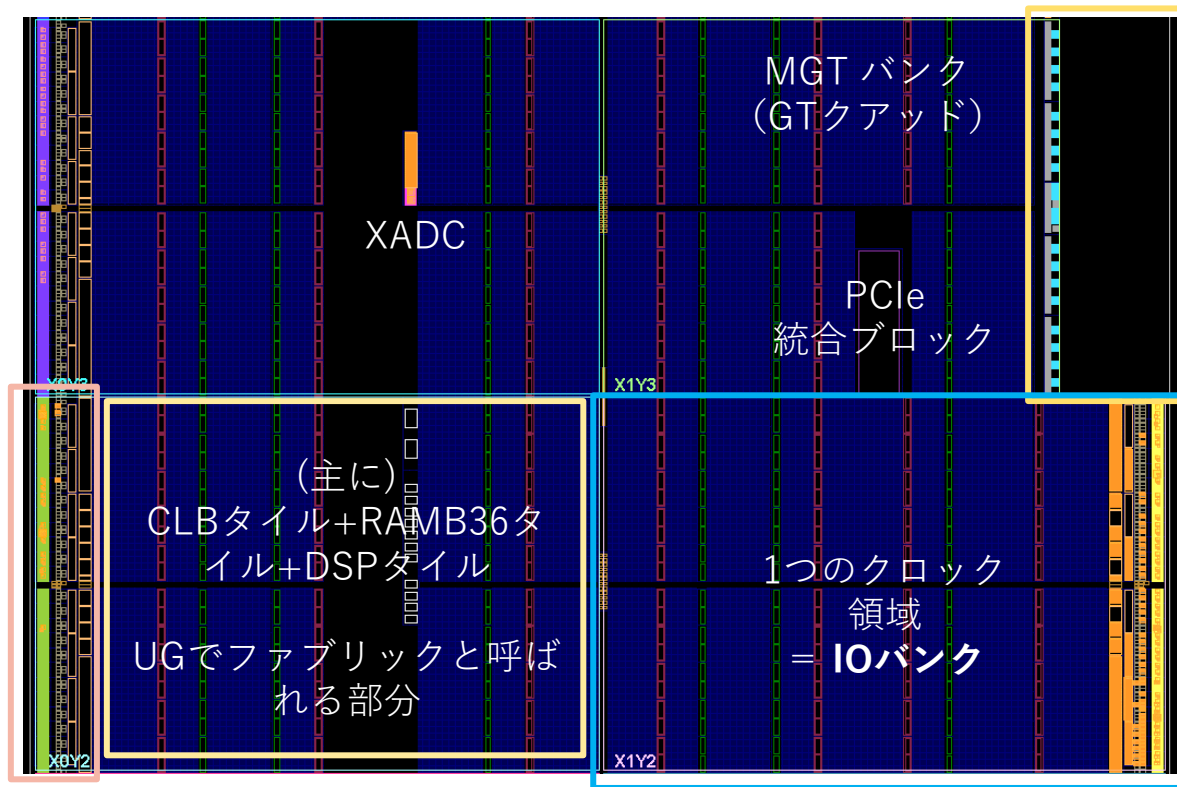
Xilinx 7シリーズはタイルベースのアーキテクチャ

Vivadoで表示したXC7K160T-2FFG676の  
フロアレイアウトの一部

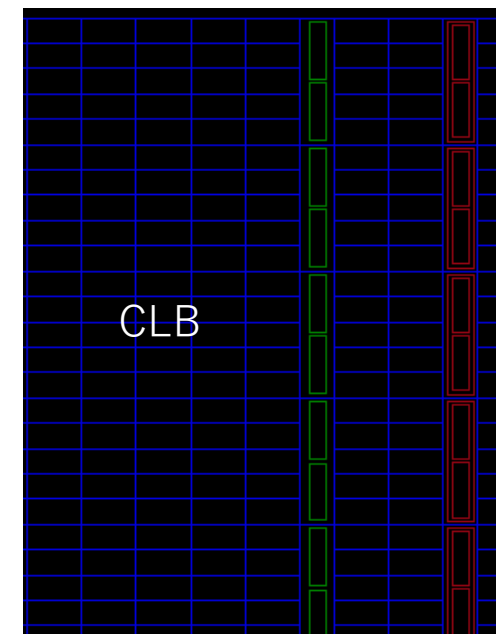
IOタイル



CMタイル  
(CMT)



ファブリックの構造



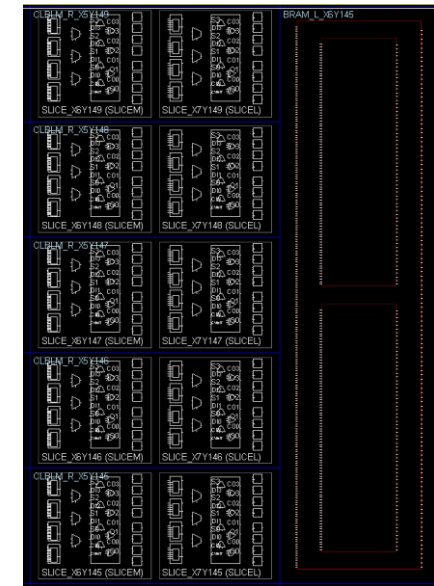
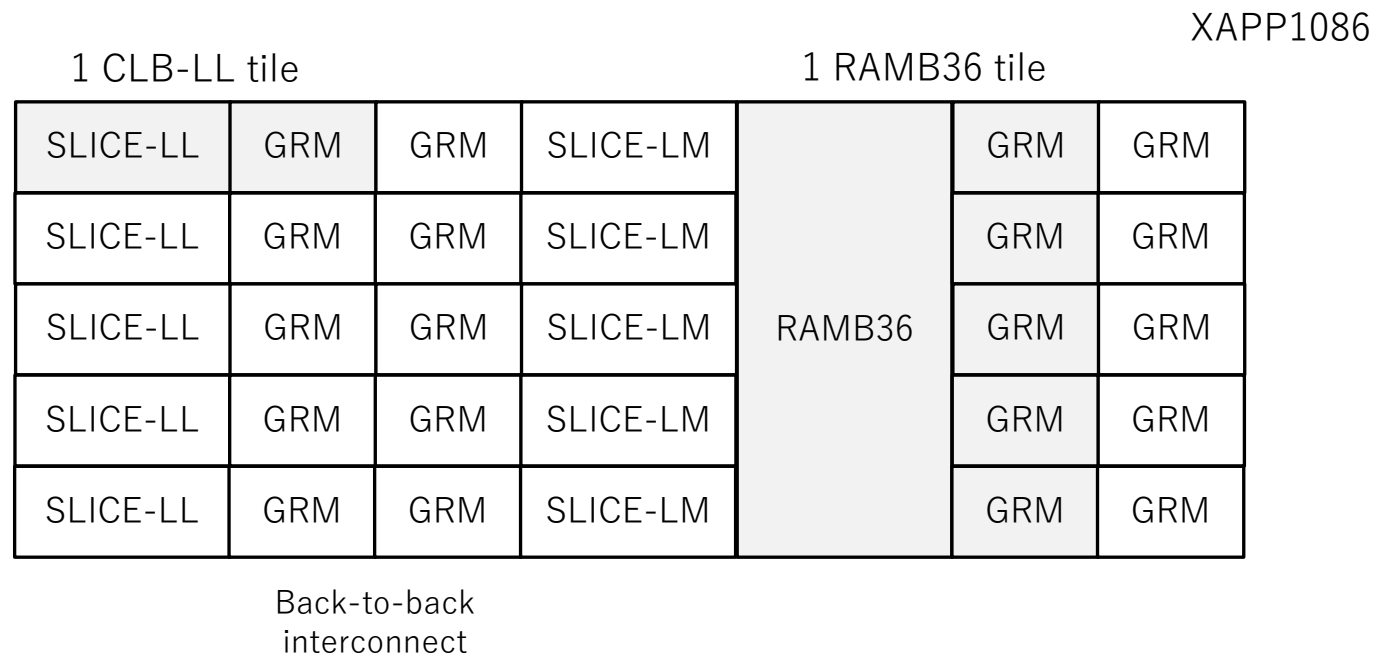
DSP  
tile      RAMB36  
tile

## Tile

- 機能ブロックと配線リソースを合わせた1式がタイルになる

## Site

- tileの1つ下位の構成要素。DSPやRAMBなどまとまった機能を提供するブロックを指す。



Global Routing Matrix (GRM):配線リソース

GRMを実際の設計で意識することは殆どない（Vivadoでも表示されない）  
が、配線リソースも有限であるという事は頭に入れておく必要がある。

## SLICE: CLB tileのsite

### 構成要素

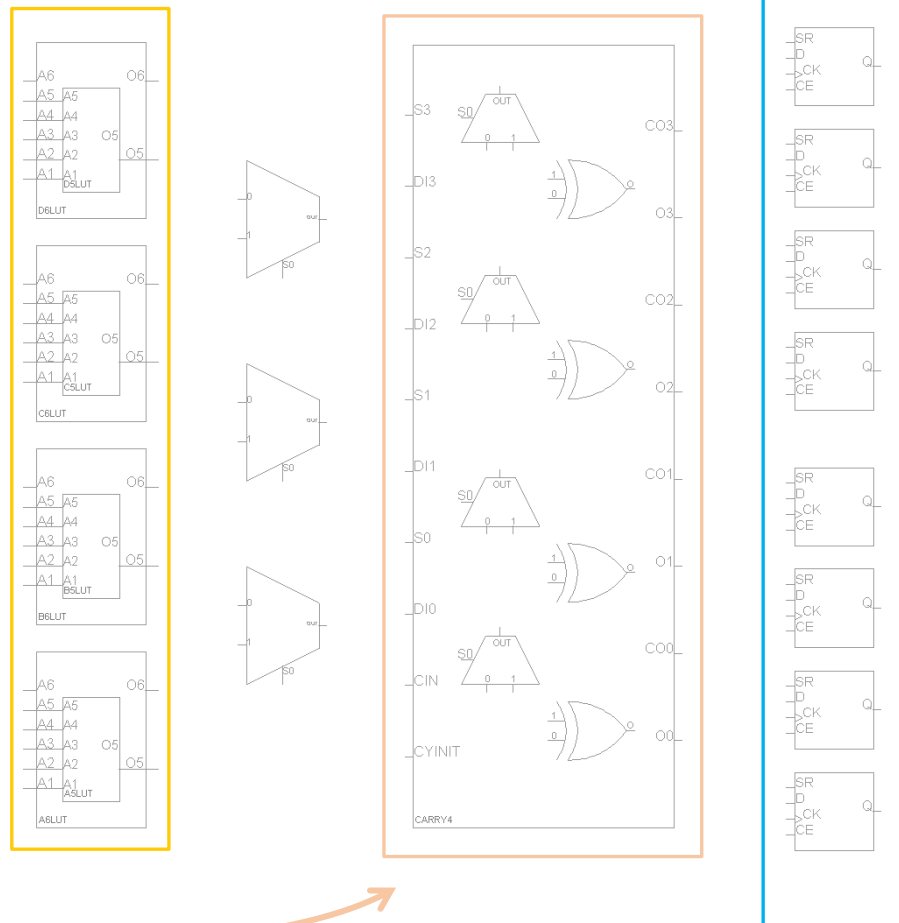
- 6入力, 2出力のLUT x4
- 4-bitのCARRY素子
- 8つのストレージエレメント

### Look Up Table (LUT)

- 組み合わせ回路のもと
- 真理値表を埋め込んで任意の組み合わせを実現

### CARRY4

- 桁上りを生成伝搬させる専用ライン
- 低遅延になるようにGRMとは独立



UG474

### ストレージエレメント

- DFFとLATCHになれる素子が4つ
- DFFにのみなれる素子が4つ

ゲートレベルの回路は論理合成によりCLBのレベルへ翻訳される

ゲートレベルの回路ブロック  
≠ CLBスライス単位  
1つの機能を実現するのに複数の  
CLBスライスが使用される



## 構成要素

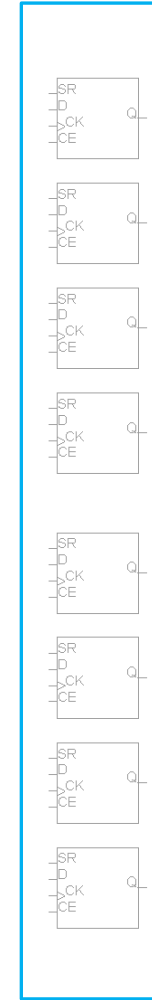
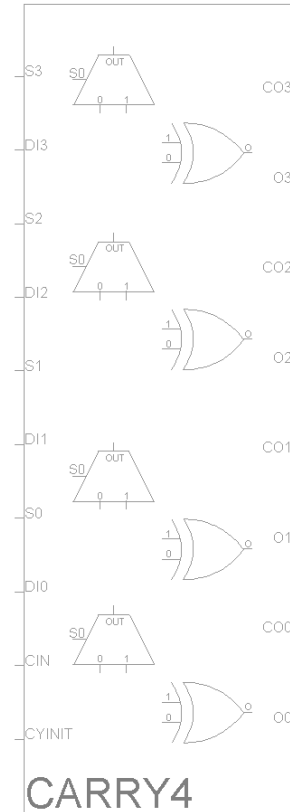
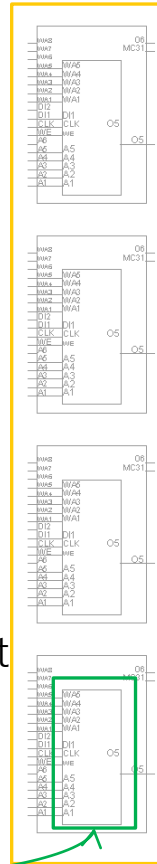
- 6入力, 2出力のLUT x4
- 4-bitのCARRY素子
- 8つのストレージエレメント

## 拡張機能の付いたLUT

- Function generator
- (分散)RAMとシフトレジスタを生成するために使う

## LUT1つで

- 6-bit address (64-depth), 1-bitデータのシングルポートRAM
- 32遅延のシフトレジスタ



SLICEMはCLBベースのシフトレジスタか分散RAMを生成する際に利用

## シフトレジスタの場合

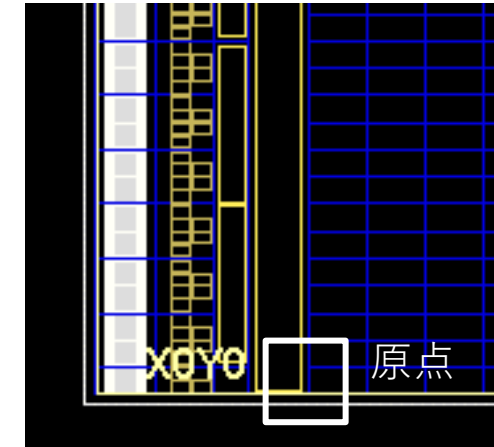
- 遅延はLUTだけで生成できる
  - 128遅延/SLICE
- 途中経過の出力にはD/DQピンが必要

## 分散RAMの場合

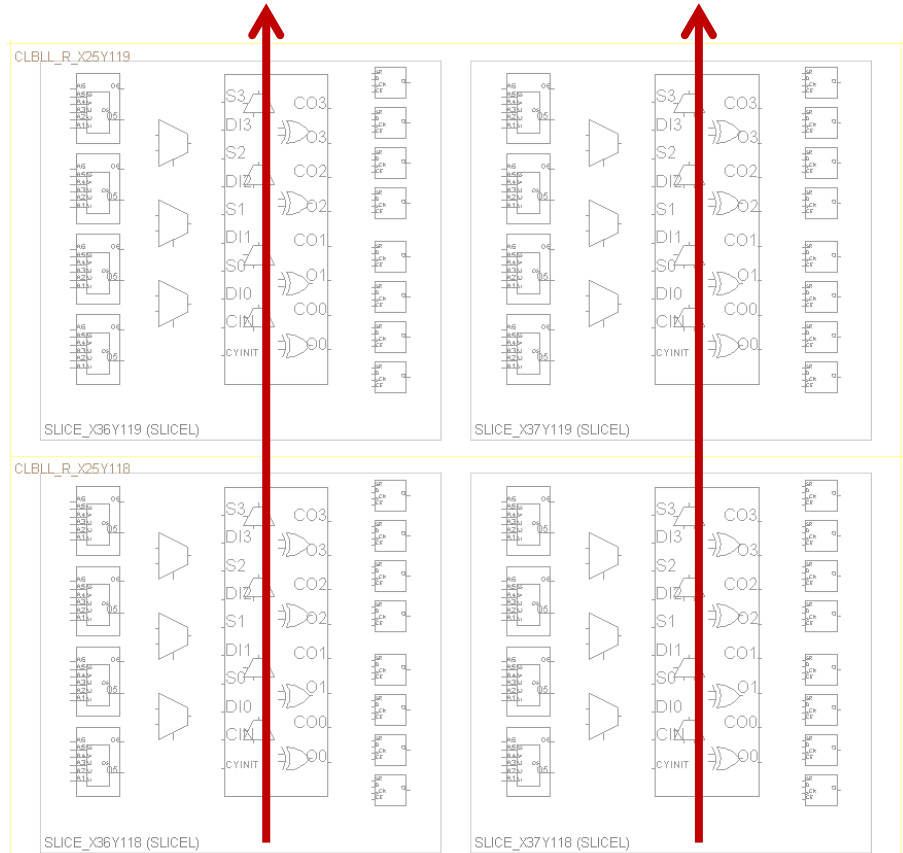
- 分散RAMは同期書き込み・非同期読み出し
- 同期読み出しするにはストレージエレメントの実体化が必要
- 4-bitデータ/SLICE。大データの場合RAMBタイルの利用が推奨。

## CLBタイル (ファブリック)の構造

- スライス 2つで1つのタイル。
- 高速キャリーラインを形成するためにスライスは列を形成している
  - 列番号がX。行番号がY。（クロック領域のXYとは別！）
  - 原点は左下。



縦方向の接続はCARRY4専用  
桁数の大きい加算・乗算器を  
CLBで作る際に使用



2つのスライスがMとLの場合

- CLBLM

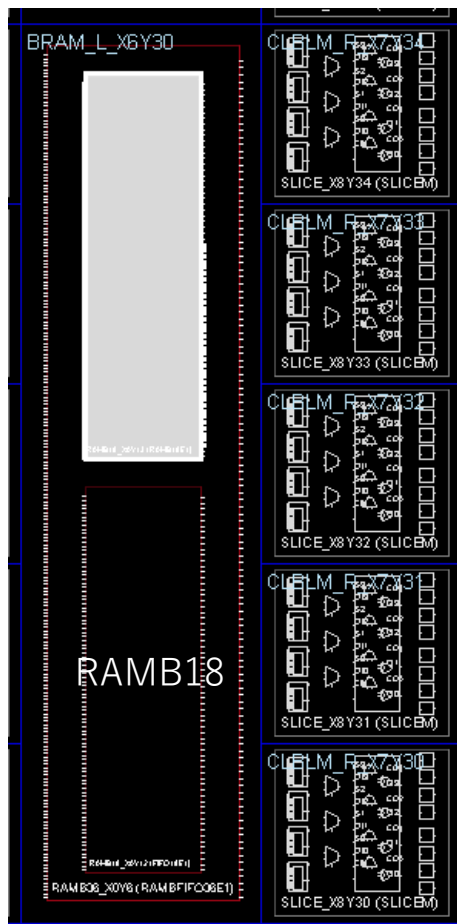
2つのスライスが両方ともLの場合

- CLBLL

CLBLMとLLの列は交互に並んでいる

分散RAMになれるスライスは全体の1/4しかない

18が2つ合わさって  
RAMB36



## Block RAM

- 分散RAMに対してBlock RAM (BRAM)と呼ばれる
- ブロックRAMカラムに配置されている
- 18 KbのRAMB18が2つ合わさってRAMB36 (36 Kb)

## 36 Kbの記憶領域をどのように使うかはユーザー次第

例: 1-bit幅ならアドレス範囲は36,000 (16-bit) => Shift register

例: 36-bitデータ幅で使うのならアドレス範囲は1024 (10-bit)

## 1つのRAMB36タイルでは心もとない…

- 複数のBRAMを接続して巨大なRAMが利用可能

ブロックRAMカラム内の専用配線を利用して最小限の配線遅延で  
接続：**ファブリックのリソースを消費しない**

- IPでFIFOを生成するときはこの限りではないので注意

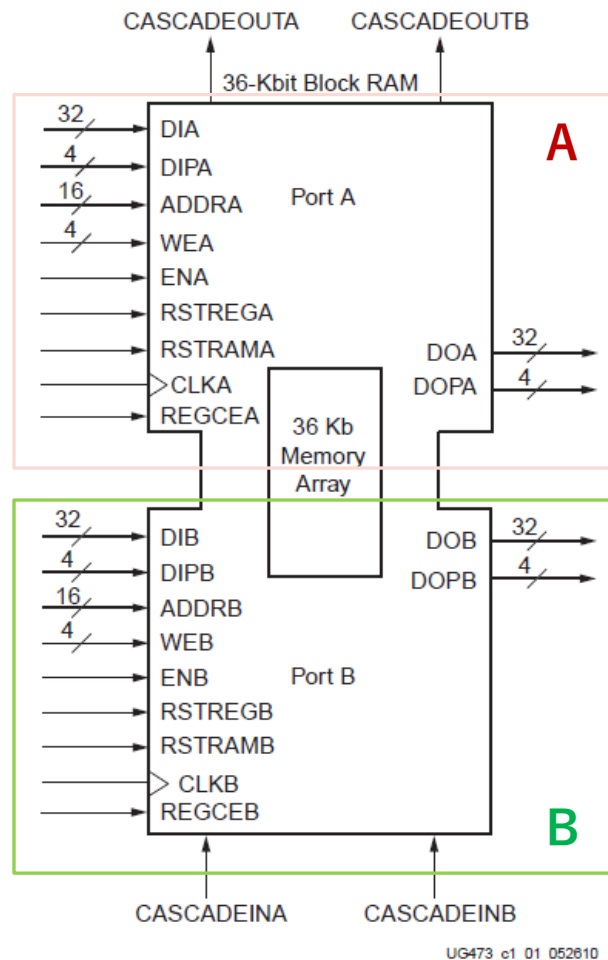


図 1-1 : RAMB36 の TDP データ フロー

## ポイント

### True Dual Port (TDP)

- 2つの独立ポートから1つのメモリ領域の任意の場所にアクセス可能なモード
- 独立の読み書きが可能

### Simple Dual Port (SDP)

- ポートAを読み出し、ポートBを書き込みとするモード
- AとBのポートをまとめてポート幅を倍に出せる

- DI: データ入力
- DIP: パリティビット (拡張データとしても利用可)
- ADDR: アドレス

メモリ領域は36Kbなので(DI+DIP)のデータ幅と16-bitアドレス空間で表現できる範囲でしか利用できない

すなわち

32Kx1, 16Kx2, 8Kx4, 4Kx9, 2Kx36, 512x72 (SDPモードのみ)

アドレス範囲 512 x 72 データ幅 (DI+ DIP)

余談: パリティビット

- データ誤り検出用のビット
- データ列に存在する1の数が偶数なら0、奇数なら1
- パリティビットを足した1の数は必ず偶数

余談: なぜ72-bit単位?

- 64-bitデータに8-bit Error Correction Code (ECC: 誤り訂正符号) がついて72-bit単位が1まとまりだから。
- DIMM規格のDDR3/4-SDRAMのバス幅と同様

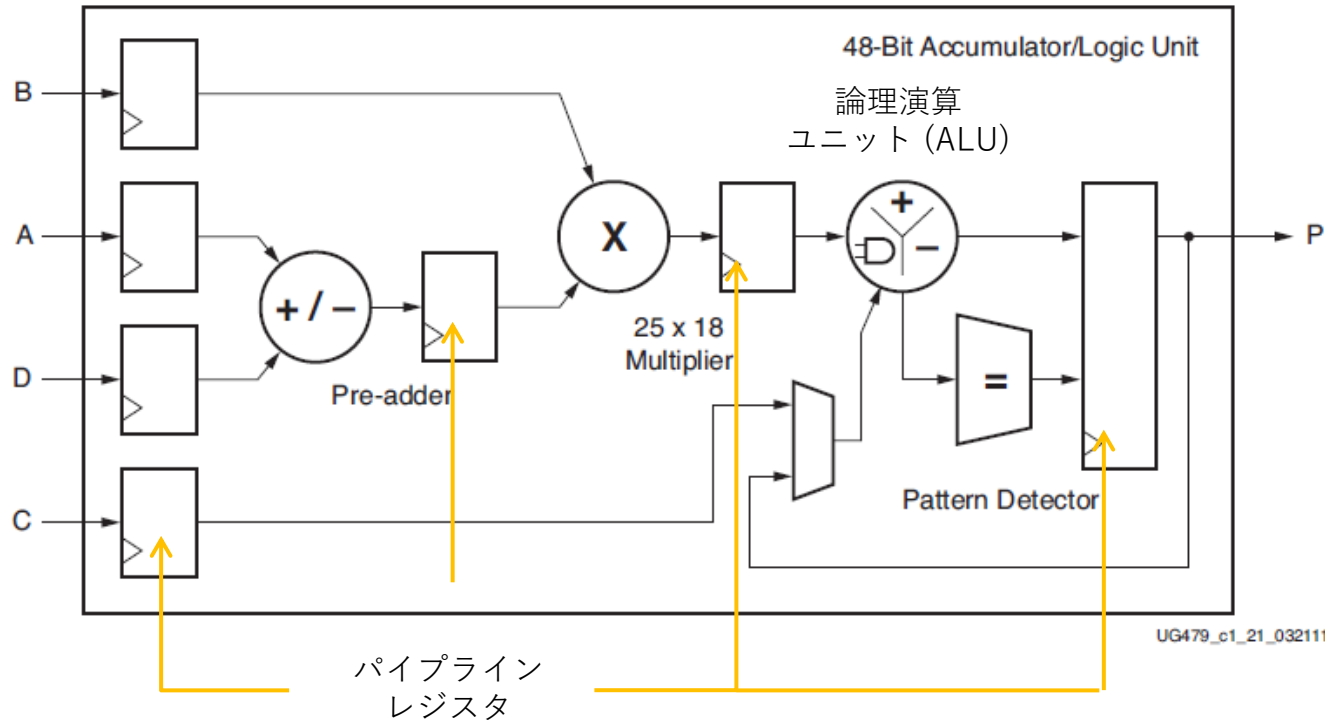
詳しくはメモリリソースの章で

FPGAではCLBを利用して加算器・乗算器が実装可能であるが大規模な演算には専用タイルの**digital signal processor (DSP)** を利用する  
ブロックRAMを同じくDSPカラム内に複数配置されている

**Xilinx 7シリーズFPGAのDSP48では**  
**4つの入力に対し加算, 累算, 乗算などが演算可能**  
詳しくは四則演算で

## 実装できる演算の例

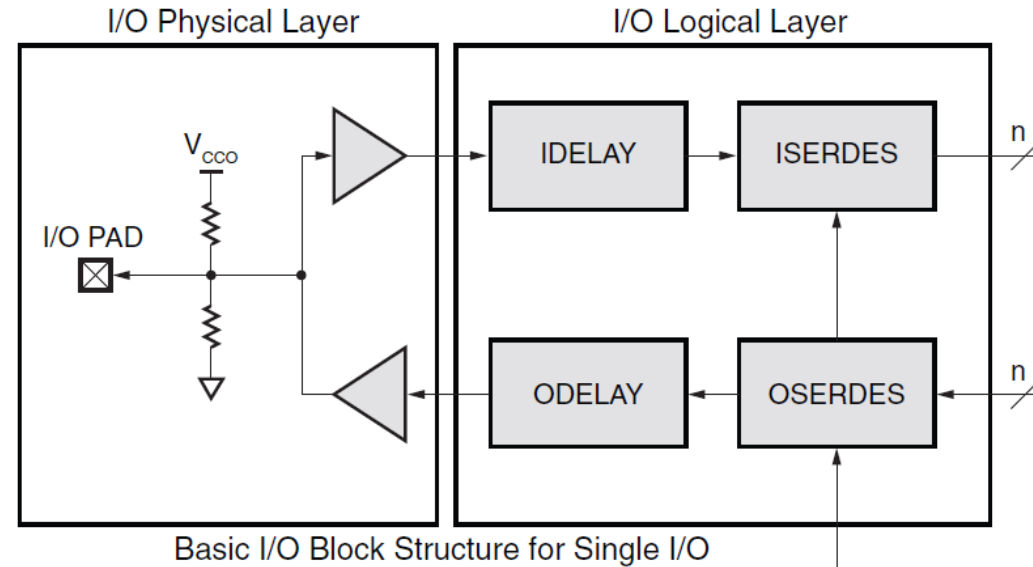
- $(A+D)*B+C$
- $A+D+C$
- $A+P$  (累算)



## ILOGIC (リソース)

### De-serializer, ISERDES (Primitive)

- 1本の信号線から複数ビットのデータを取り出す (Data In)



Logical layerを使用しない場合IOBから直接ファブリックへ信号線が繋がる

\* ODELAYはHPバンクにのみ存在

## Physical layer

- 入出力の物理的な性質を決定する
  - 終端抵抗 (入力)
  - 入力同相電位
  - 出力電位
  - etc...

## Single data rate (SDR)

- クロックエッジの立ち上がり (立下り)だけを利用する

## Double data rate (DDR)

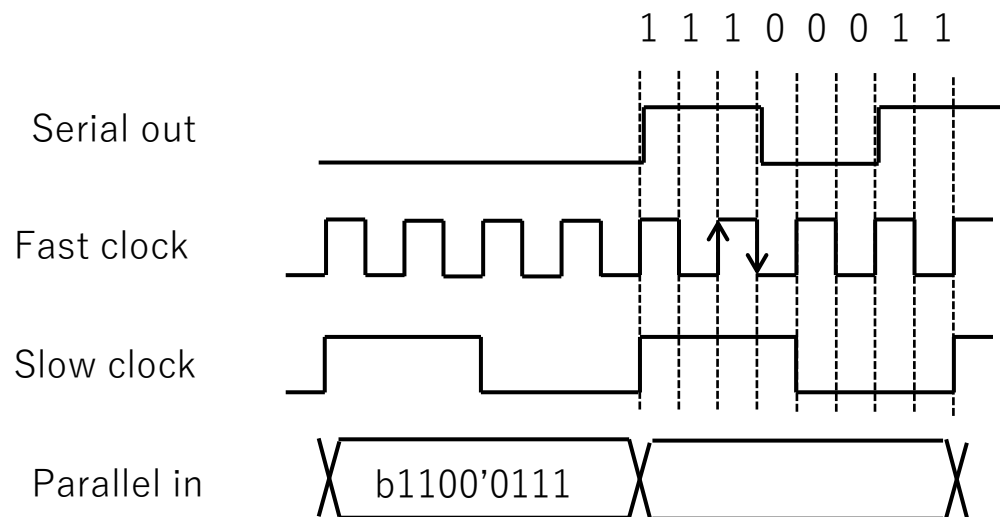
- クロックの両エッジを使う

## OLOGIC (リソース)

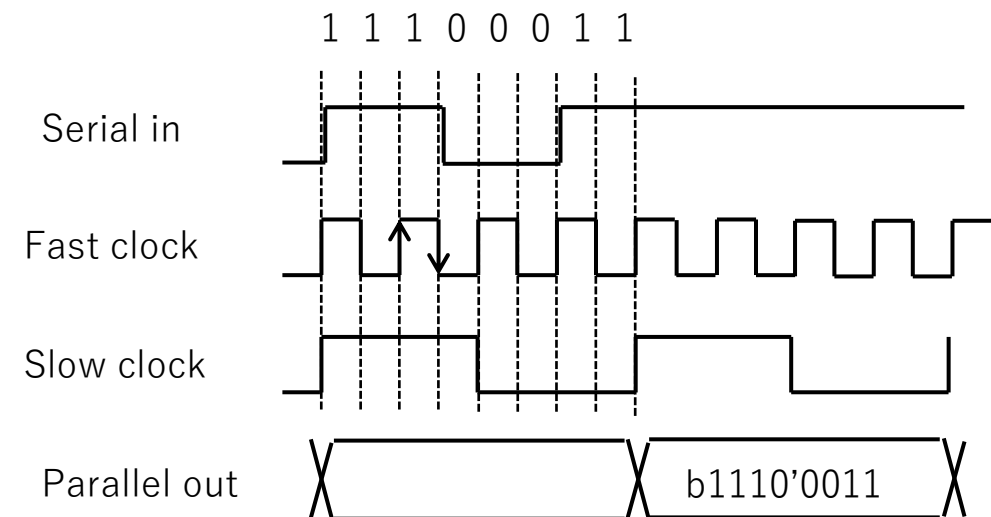
### Serializer, OSERDES (Primitive)

- 複数ビットのデータを1ビット列の信号に変換する (Data Out)

## Parallel-to-Serial (OSERDES)



## Serial-to-Parallel (ISERDES)



7シリーズFPGAには2つのクロック生成器が存在

## Mixed Mode Clock Manager (MMCM)

- **入力クロック**に同期した**異なった周波数・位相**のクロック生成を行う専用ブロック
- コア技術はPhase Lock Loop (PLL) (余談: PLLは一般的な技術なのでググると出てきます)
- どんな周波数でも生成できるわけではない

$$F_{OUT} = F_{CLKIN} \times \frac{M}{D \times O}$$

- MとDは整数であり1つのMMCMからの出力クロックすべてに対して共通
- Oは基本は整数 (整数分周)だが特定の分数 (分数分周)も取ることが出来る  
必要なクロックが生成できるか？

=> 入力クロックを素因数分解してみよう。大体のクロックは2,3,5,11の積で書けるようになっている。

## PLL

- MMCMを基にしたクロック生成器で性能はMMCMの方が高い

他にもPhaserやIO\_FIFOと言ったブロックがCMTには存在するが  
主にメモリコントローラ用なので中級トレーニングコースでは割愛



## Multi-gigabit-transceiver (MGT): 高速シリアル通信のトランシーバ

- SERDESと同じ？

### TX/RX線, つまりデータ線だけで通信を行う ⇒ クロックは？

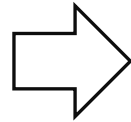
- クロックはデータ列の中に埋め込まれており受信側はクロックの復元が必要
  - クロックは周期的に0と1が繰り返される信号
  - 任意のデータに対してはそんな都合のいいビットパターンにならない

### 8b10b, 64b66b, 65b67b変換を利用する

例: 8b10b変換

8-bitバイナリ

b0000'0000



シンボル

b100111 0100  
b011000 1011

これを送信する

シンボルはディスパリティという  
極性を持っている  
0と1の積算数がバランスするように  
エンコーダは極性を切り替える

8b10変換では0/1が連続する期間が4クロック以下になるようシンボルが決まっている

### ラインレートと実際に送れるデータ量

- ラインレートは物理層が送受信しているビットレートを指す。データレートよりも少し高めに設定してある。
  - 例: 1GbpsのGigabit Ethernet, ラインレート 1.25 Gbps.
  - 変換するとユーザーデータでないビットがシンボルにつく(8b10では2ビット)

**GT channel** ⇔ 高速シリアル通信のPHY (物理層を駆動するデバイス)

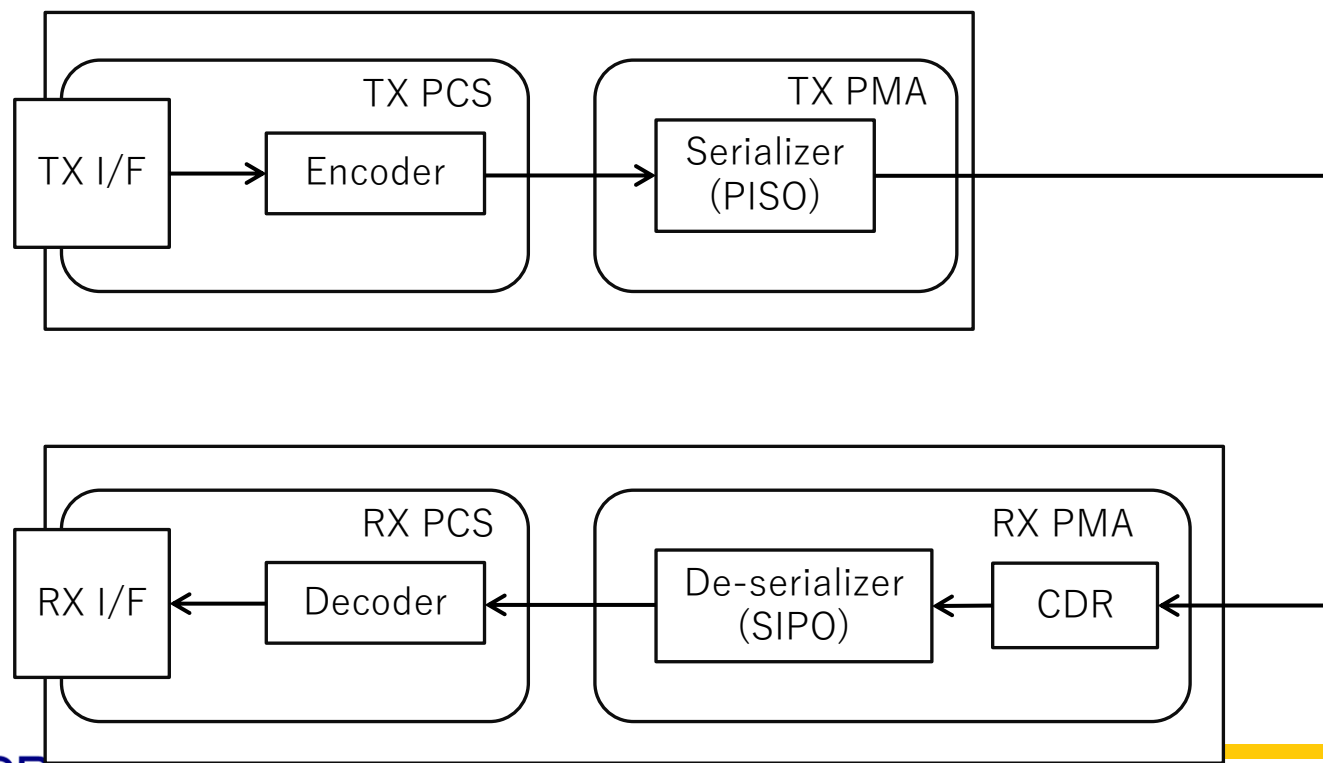
**PCS** (Physical Coding Sublayer) 物理符号化副層

- 主にバイナリ/シンボル変換を行う (8b10b encode/decodeなど)

**PMA** (Physical Medium Attachment) sublayer 物理媒体接続副層

- 主にシリアル/パラレル変換を行う

GTチャンネルの大まかなブロック図  
(多くの機能を省略して書いてます)



\*GTREFCLKはTXのラインレートを決定するために使う

\***SIPO**: Serial In Parallel Out

\***PISO**: Parallel In Serial Out

\***CDR**: Clock Data Recovery

データのビット列からクロックを取り出す(復元)するための回路

GTチャンネルはPHYであるため  
通信の最下層の機能しか提供しない。

実際の通信には  
EthernetならEthernetのルールに従ったフレームを  
AuroraならAuroraのルールに従ったフレームを  
組んで流す必要がある。  
(ふつうはIPを使う)

ここまでの説明ではリソースとプリミティブが混ざった状態で進めました。  
(完全に分離して説明すると説明がむしろ難解となる)  
1つリソースは複数種のプリミティブに変化できる、ということは覚えておいてください。

## いくつかの事例

- RAMB36E1はプリミティブでありメモリリソースではないですが、TDPのRAMB18かRAMB36がメモリリソースの基本要素と言って良いでしょう。
- CLBのレジスタリソースに構成されるFFの種類は複数ある (例: FDCE, FDPEなど) がどういうプリミティブを使用するかはこの演習では殆ど指示しません。(通常Vivadoに推定させるため)
  - 明示的な利用は制約の所でだけ少し扱います。
- IOSERDESはILOGIC, OLOGICに構成されるプリミティブの一種です。
  - IO blockの先頭に存在するリソースはILOGICおよびOLOGICであり、これが複数のプリミティブに変化できます。

## 余談

どういうプリミティブがXilinx 7-series FPGAに存在するか  
UG953を参照

# FPGAの内部構造 (クロック配線)

クロック配線のアーキテクチャは初心者が中級レベルに到達する登竜門だと思います。  
それだけに鬼門です。

## グローバルクロック

- Xilinx FPGAを取り扱っているとまず出会うクロック種別。
- FPGA内の全てのクロックポイントに到達できる。

これだけなら簡単なのだが…

## リージョナルクロック

- 特定のクロック領域の
  - ロジックリソースを駆動可能
  - IOリソース（IOタイルのリソースの事: 主にSERDES関連）駆動可能

## IOクロック

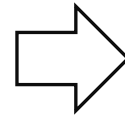
- 特定のクロック領域のIOリソースを駆動可能。

BUFMRが絡むことによって  
隣接した複数のクロック領域の  
リソースを駆動可能になる。

Xilinx FPGAでクロックを使いこなすには  
各タイルとクロック配線ネットワークの関係を理解する必要があります

**IOクロック:** BUFIOが出力するクロック

- UG472によると...ILOGIC/OLOGICを駆動可能...?
  - ILOGIC/OLOGIC: **ISERDES/OSERDES**



**BUFIOが存在する⇒IOSERDESを利用している  
(しかし逆は真でない!)**

余談

SERDESのCLKとCLKDIVの駆動元は  
BUFIOとBUFRだけではないので  
UG471を読んでよく理解してください

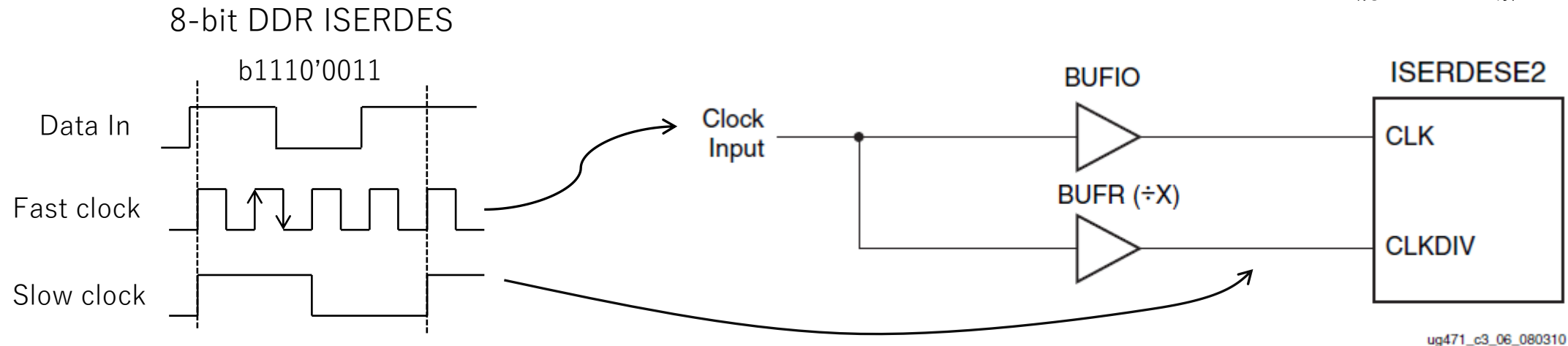


図 3-6 : BUFIO および BUFR を使用したクロック配置

ADC (例: AD9637)からのデータを受信する際の典型的な配線例

# フロアレイアウトとクロック配線の関係

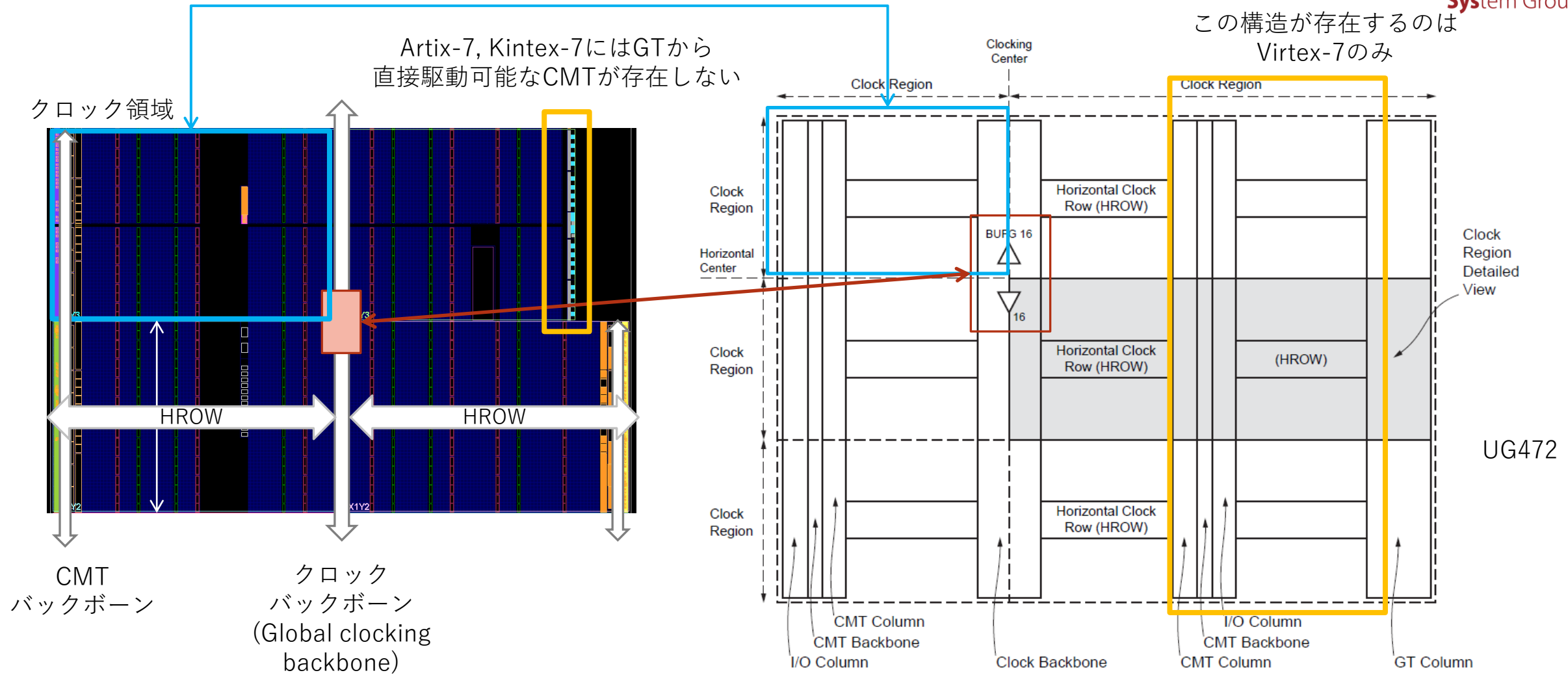


図 1-1 : 7 シリーズ FPGA のクロック アーキテクチャの概略図

UG472\_e1\_30\_020712

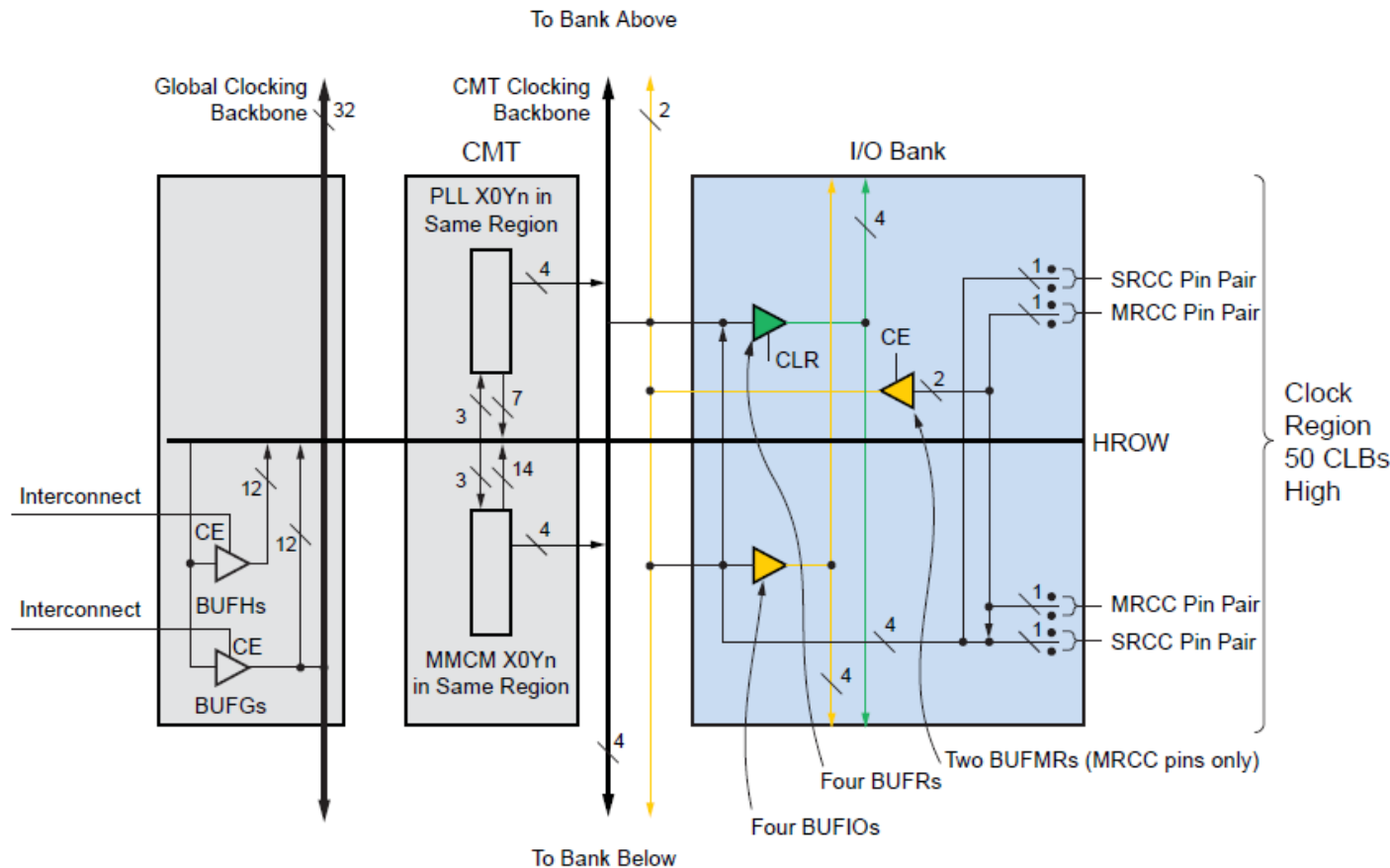


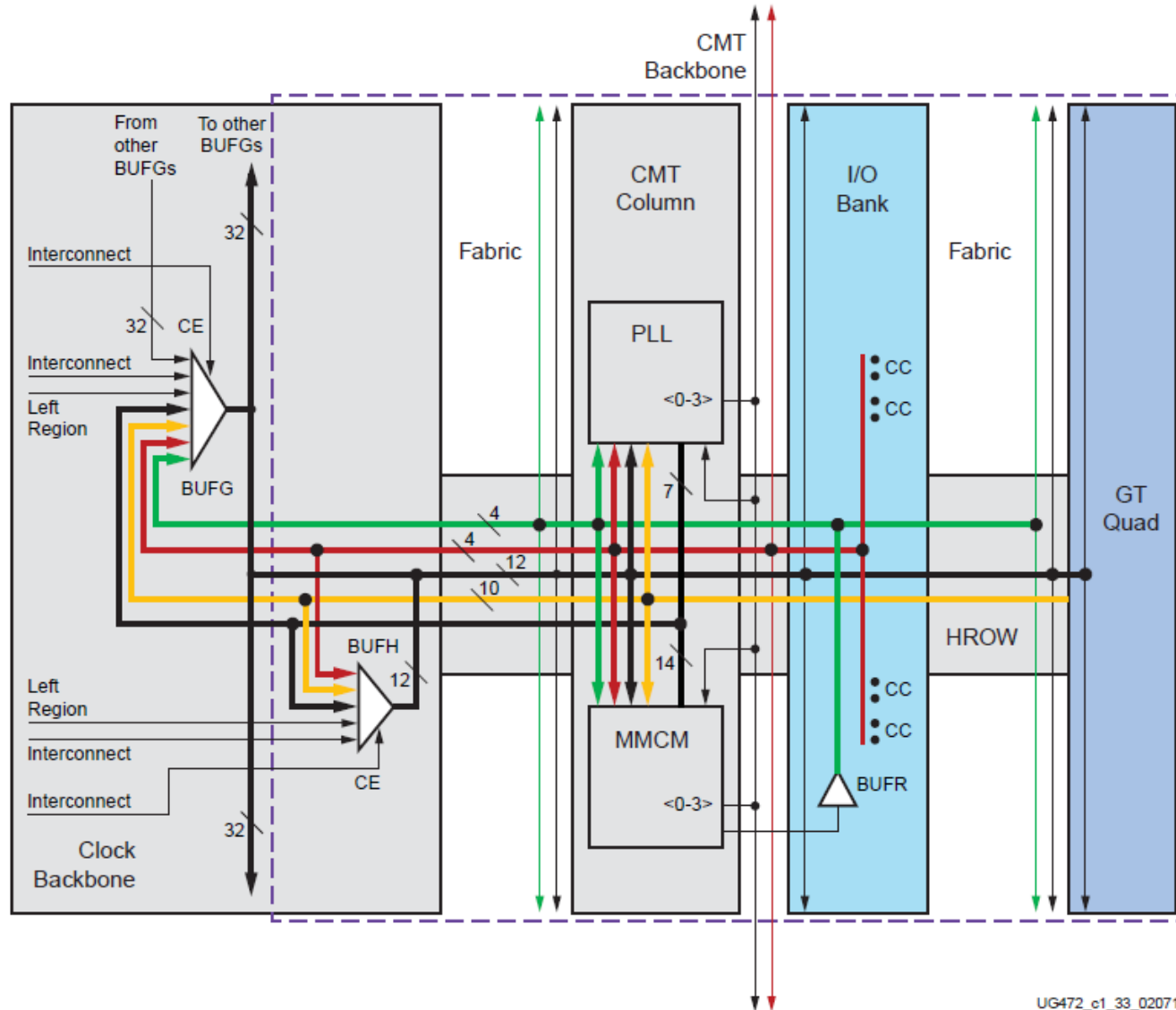
図 1-3 : 1つのクロック領域 (デバイスの右側)

## 要点

- クロックバックボーンを駆動可能なのはBUFGのみ
- 水平ライン (HROW) は複数のドライバが共有している (後述)
- CMTバックボーンにはCMTの相互接続用とIOリソース駆動用の2つのラインが存在する
- **Clock Capable (CC) pin**にはSingle Region (SR) CCとMulti Region (MR) CCの2種類が存在し、各IOバンクに2ペアずつ存在。**BUFMR**を駆動可能なのはMRCCのみ。
- BUFMR: **上下に隣接した**クロック領域のBUFIOとBUFRを駆動可能。
  - BUFRが駆動可能なので隣接したクロック領域のファブリックも駆動可能。

UG472\_c1\_32\_011713





UG472\_c1\_33\_020712

## 多分一番わかりやすい図

- IOタイル内のリソースが省略されている。
- BUFRがCMTでしか駆動できないように見えるがそんなことは無い。

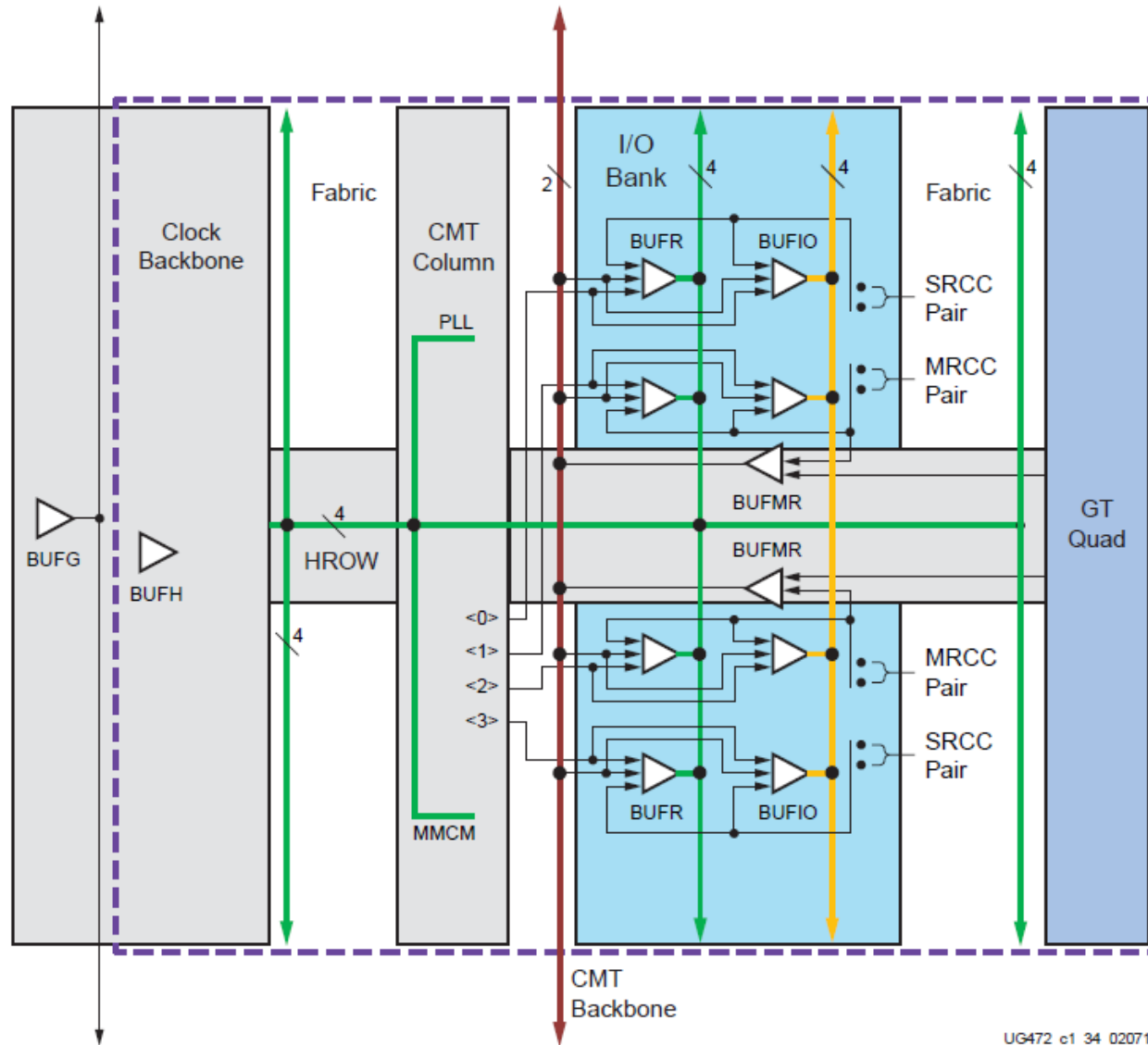
## BUFHはどんな時に使うか？

- 例：**水平方向に隣接**するクロック領域のCMTを駆動する。(Kintex-7でGTXトランシーバからのクロックでMMCMを駆動するなど)

## 問題

- CCピンから入力した発振器のクロックをMMCMへ入力し、生成したクロックをグローバルクロックにする経路を描いてみよう。
- あるクロック領域のファブリックを駆動可能なクロックソースはBUFG以外だと何？

図 1-4 : BUFG/BUFH/CMT クロック領域の詳細



UG472\_c1\_34\_020712

## IOリソースの駆動関係が最も正しく書かれた図

- BUFRとBUFIOはCCピン, CMTバックボーン及びMMCMの特定の出力で駆動可能。
- BUFIOはIOリソース駆動専用でありHROWを駆動できない。
- BUFRはHROWを駆動可能
- CMTバックボーンを駆動可能なのはBUFMのみ  
⇨ MRCCピンからの入力かGTからのクロック

図 1-5 : BUFR/BUFM/BUFIO クロック領域の詳細

FPGAのタイル構成を理解するとDS180に書かれている事がわかるようになる

CMTにつきPLLとMMCMが  
1つずつなので  
クロック生成器の数が分かる

表 6: Kintex-7 FPGA の機能一覧

デバイス	ロジックセル	コンフィギュラブル ロジックブロック (CLB)		DSP スライス <sup>(2)</sup>	ブロック RAM ブロック <sup>(3)</sup>			CMT <sup>(4)</sup>	PCIe <sup>(5)</sup>	GTX	XADC ブロック	総 I/O パ ンク <sup>(6)</sup>	最大ユー ザー I/O <sup>(7)</sup>
		スライス 数 <sup>(1)</sup>	最大分散 RAM (Kb)		18Kb	36Kb	最大 (Kb)						
XC7K70T	65,600	10,250	838	240	270	135	4,860	6	1	8	1	6	300
XC7K160T	162,240	25,350	2,188	600	650	325	11,700	8	1	8	1	8	400
XC7K325T	326,080	50,950	4,000	840	890	445	16,020	10	1	16	1	10	500
XC7K355T	356,160	55,650	5,088	1,440	1,430	715	25,740	6	1	24	1	6	300
XC7K410T	406,720	63,550	5,663	1,540	1,590	795	28,620	10	1	16	1	10	500
XC7K420T	416,960	65,150	5,938	1,680	1,670	835	30,060	8	1	32	1	8	400
XC7K480T	477,760	74,650	6,788	1,920	1,910	955	34,380	8	1	32	1	8	400

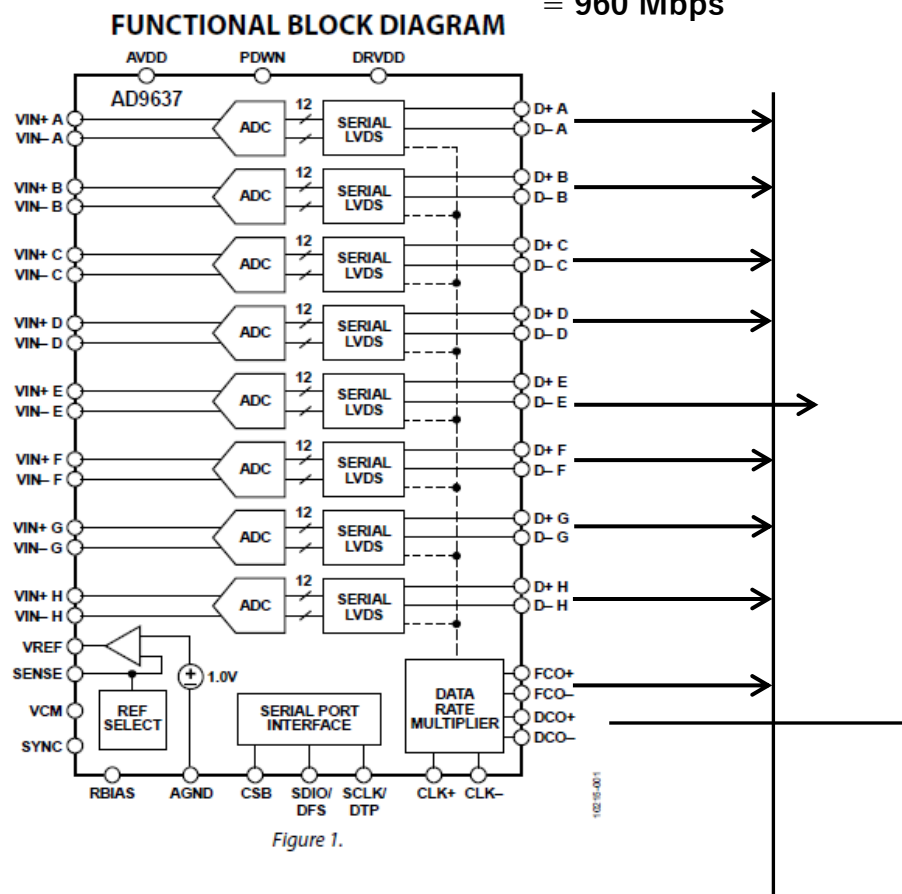
ファブリックに存在する  
CLBのリソース量  
分散RAM≡スライス

RAMBの数  
RAMB18・RAMB36の列が  
存在するが独立ではない

高速シリアルトランシーバの  
PHYの数

# これまでの知見を まとめる

データ線のデータレート  
12-bit x 80 MSPS  
= 960 Mbps



AD9637  
12-bit, 80 MSPS ADC

## 次のようなファームウェアを考える

- 8ch, 12-bit, 80 MSPSのADC, AD9637のデータを受信。
- ベースラインを求め引き算するブロックを有する。
- トリガー決定を待つためのデータ遅延機構を持つ。
- トリガー入力があったらイベントビルドを行う。
- データをSiTCPを利用しTCP通信で転送する。
- 通信のインターフェースは光モジュール (SFP) である。

Timing Diagrams

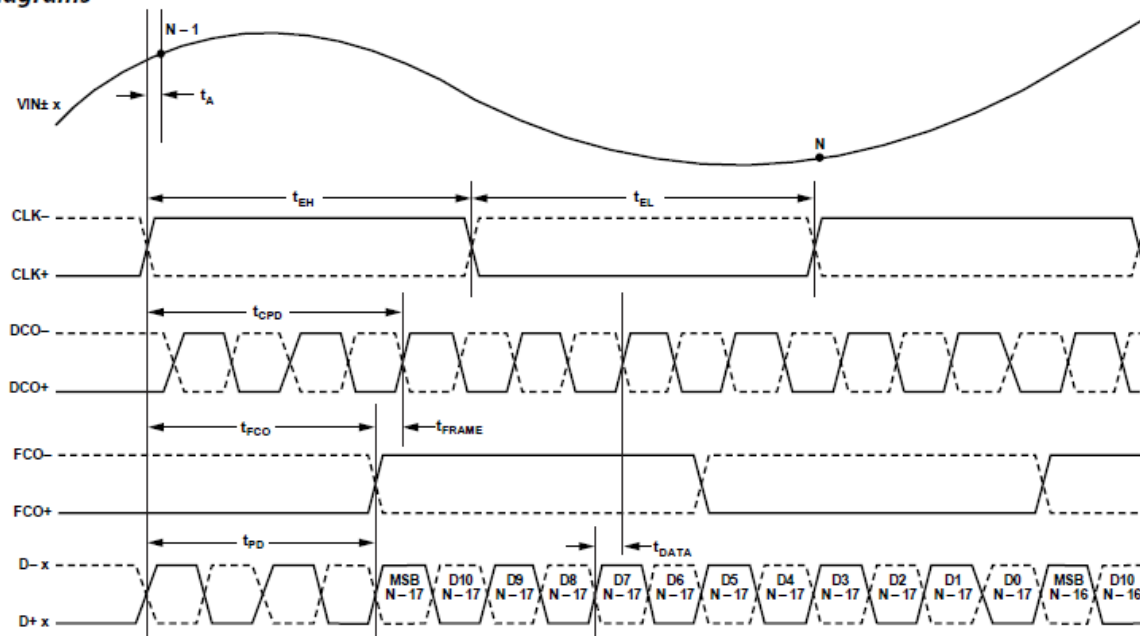


Figure 2. Word-Wise DDR, 1x Frame, 12-Bit Output Mode (Default)

# FPGA回路の構成ブロックの例

データ線のデータレート  
12-bit x 80 MSPS  
= 960 Mbps

Kintex-7であれば  
HRバンクでも  
HPバンクでも受けられる

## FPGA内部のブロック図

(ADCの制御やトリガー管理などのブロックは省略)

### FUNCTIONAL BLOCK DIAGRAM

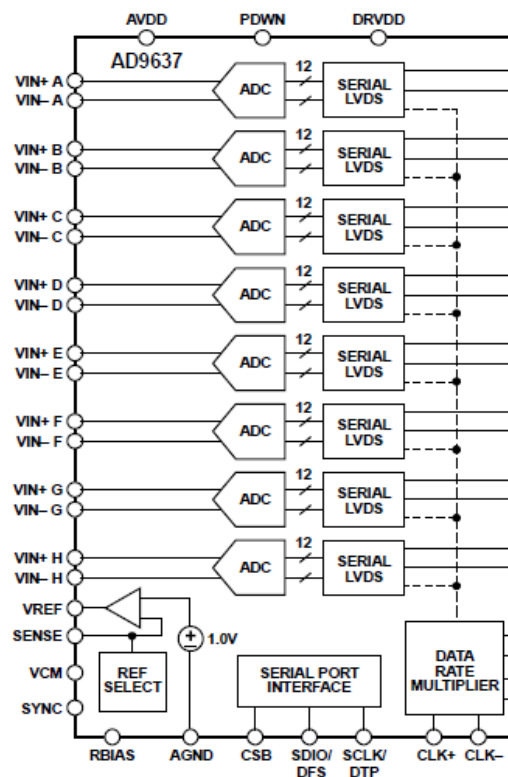


Figure 1.

AD9637  
12-bit, 80 MSPS ADC

MRCCかSRCCピンへ  
SRCCだとデータ線は  
バンク跨ぎできない。

IOタイル

ベースライン導出  
引き算

省略されている事  
リージョナルクロックと  
グローバルクロックの乗り換え

### GTX-channel

