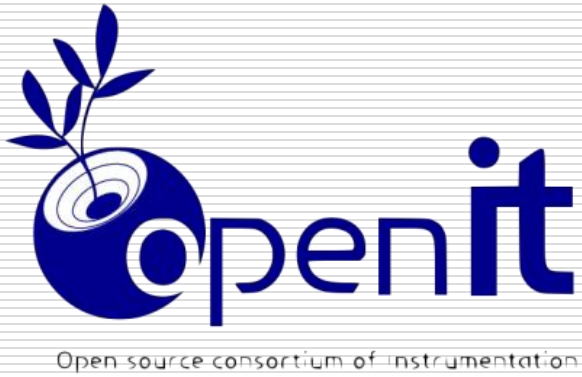
Open-It FPGAトレーニングコース（入門編）

1. デジタル回路の基礎（復習）



第3.4版

2016年07月14日

予習内容の確認

- 導入
  - FPGA
  - デジタル技術の特長
  - データ収集システムとデジタル回路（予習資料に含まれていない内容）
  - 標準IO
  - 記数法
- デジタル回路
  - 基本4要素
  - 組み合わせ回路
  - 順序回路

FPGA

Field Programmable Gate Array (FPGA)
Programmable Logic Device (PLD)の一種

デジタル回路を簡単に集積化できる！

- 集積回路
  - デジタル回路のみ実装可能
  - 例えば、2×2 cm程度の大きさ

- ユーザーが回路情報を書き込んで使用する
  - ソフトウエアの様に使用できる
  - ソフトウエアではないので注意！
    - CPUのプログラムではなく回路情報を書き込む

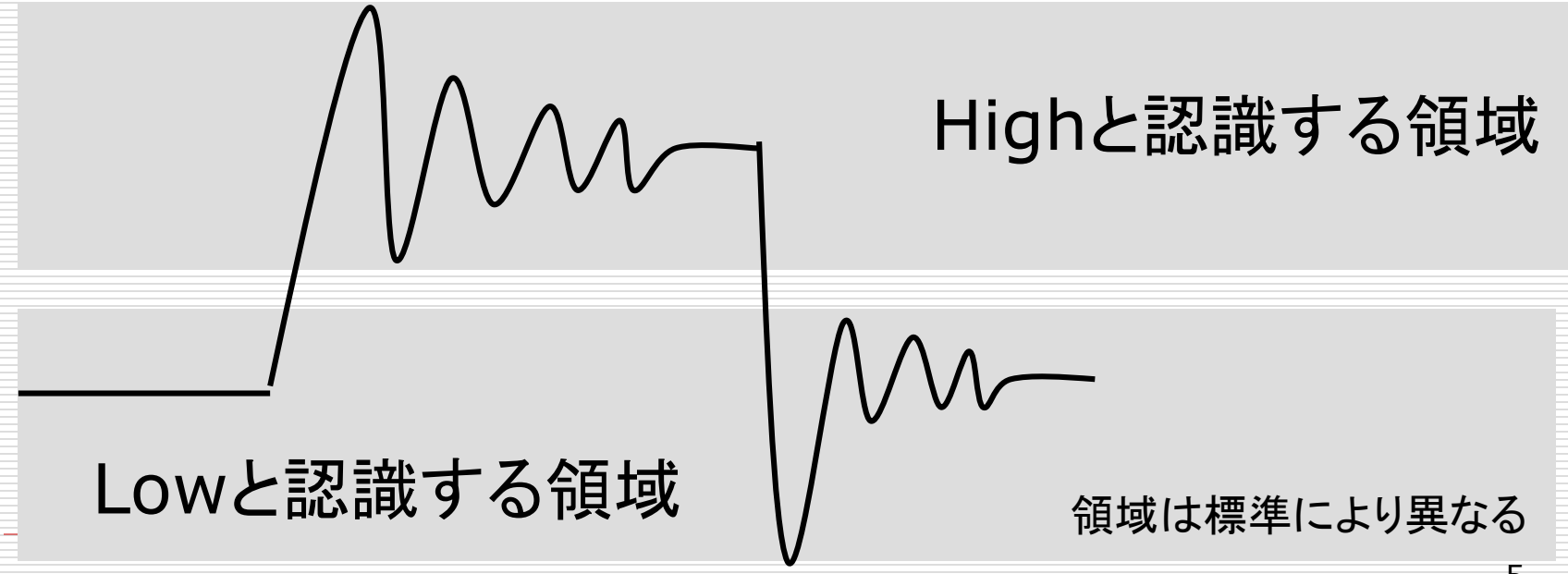
デジタル技術の特長

デジタル信号の長所

一つだけ挙げるなら

ノイズに強い！

この様なひどい信号でもOK



デジタル信号の短所

- 信号数が多くなる
  - 1本で2値(0と1)しか伝送できない
  - 複数本で数字として意味をなす
    - この本数がbit数
    - 8bitなら0から255まで表現できます
- 回路規模(トランジスタ数)が大きくなる
  - アナログ回路と比較すると圧倒的にトランジスタの数が多くなる

⇒ 近年の微細加工技術の発展により問題にならなくなった

- ノイズを発生しやすい

⇒ アナログ回路への影響について注意する

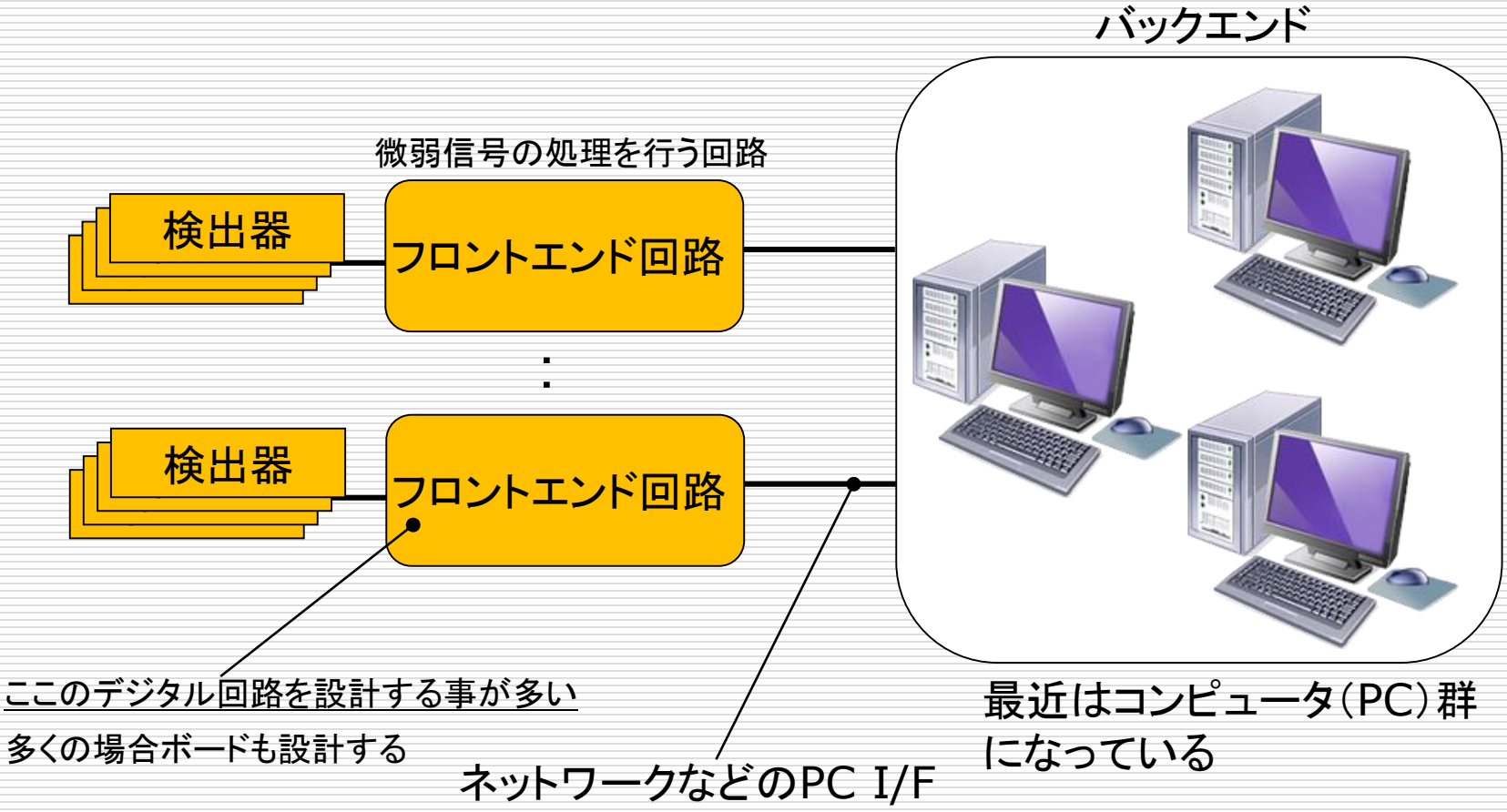
データ収集システムとデジタル回路

デジタル回路（FPGA）が使われる場所

- FPGAに実装する事ができる回路はデジタル回路です。
- DAQシステム内のどこでデジタル回路が使用されているのか説明します
- 最後にどのような機能をFPGAに実装するのか説明します

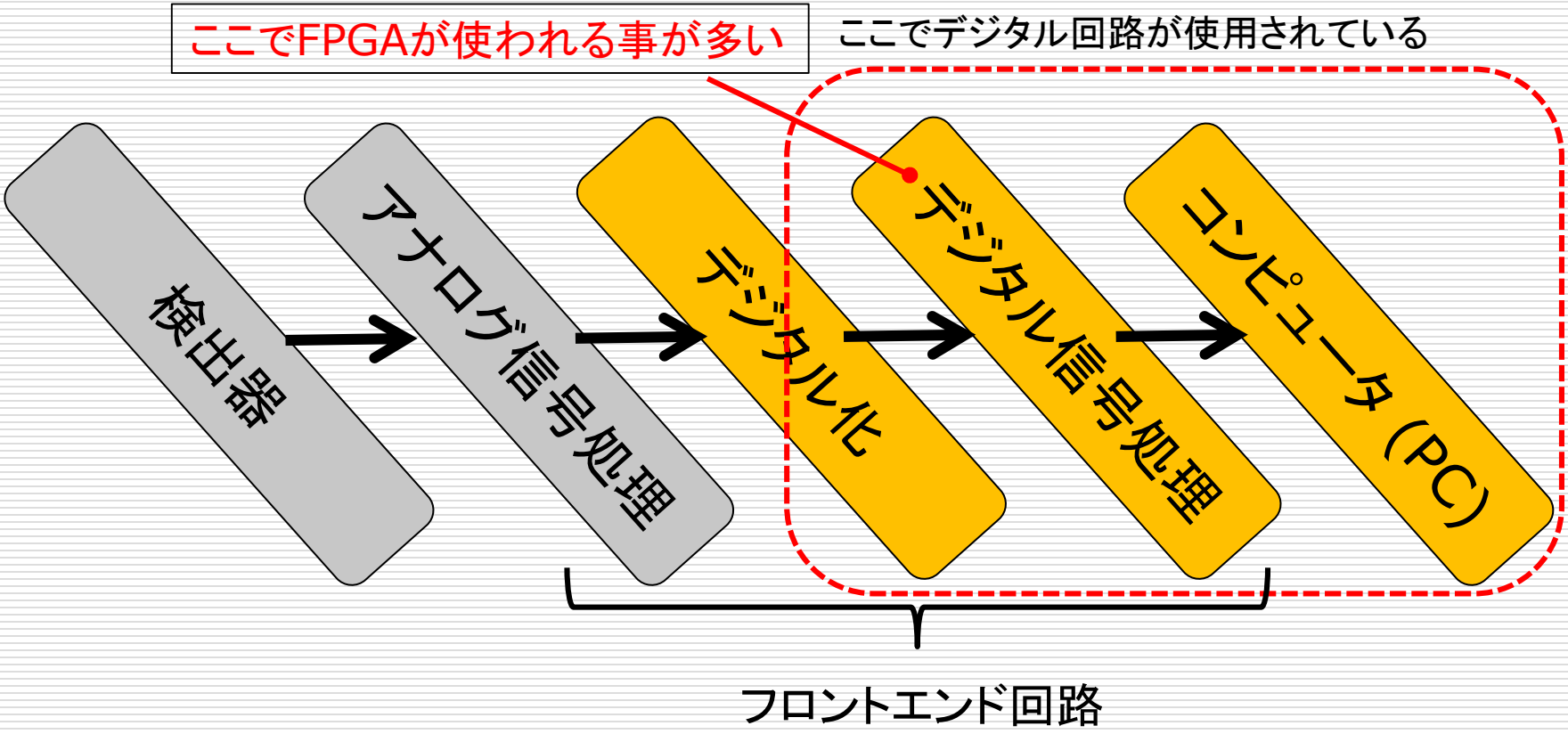
- デジタル回路が使用されている場所
  - 読み出し(DAQ)システム
  - PCへ送るべき情報
  - DAQシステム内の機能ブロック
- FPGAに実装する機能

計測システム


バックエンド
微弱信号の処理を行う回路
検出器
フロントエンド回路
検出器
フロントエンド回路
ここのデジタル回路を設計する事が多い
多くの場合ボードも設計する
ネットワークなどのPC I/F
最近はコンピュータ(PC)群
になっている

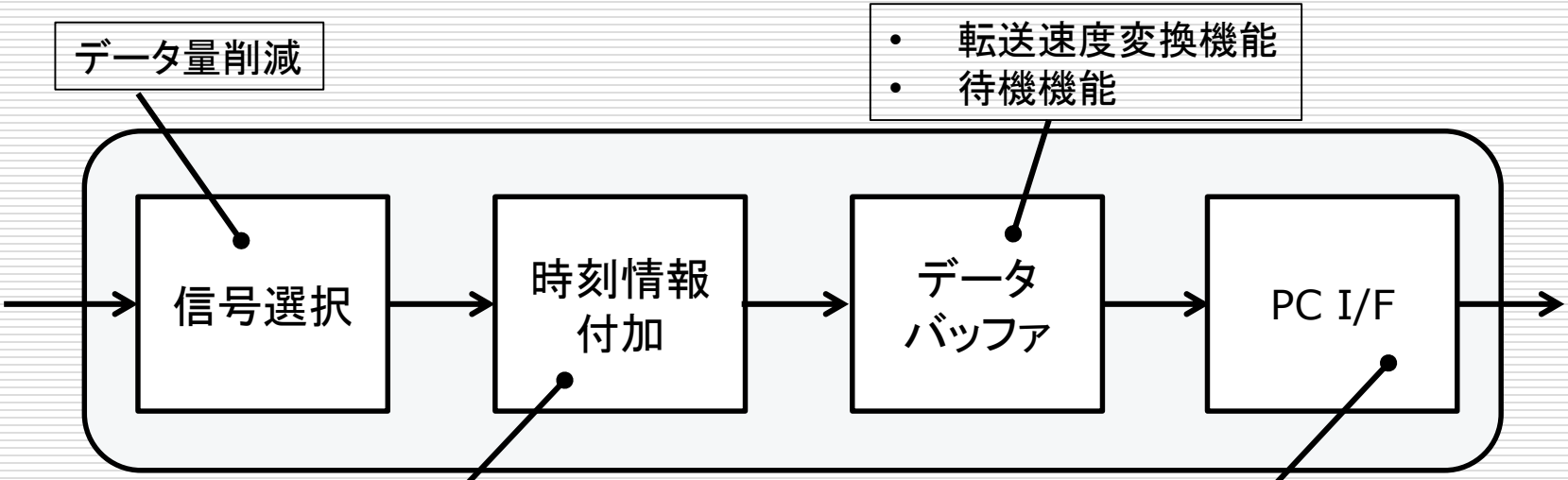
FPGAが使われる場所

デジタル回路が占める割合は大きい！



FPGAに実装する機能

多くのデジタル処理ブロックに含まれる機能



ここまでのまとめ

- FPGAはデジタル回路
- デジタル回路が使われる場所
  - フロントエンド回路に搭載されているADCからバックエンドまで
- FPGAはフロントエンドの信号処理回路として使われる

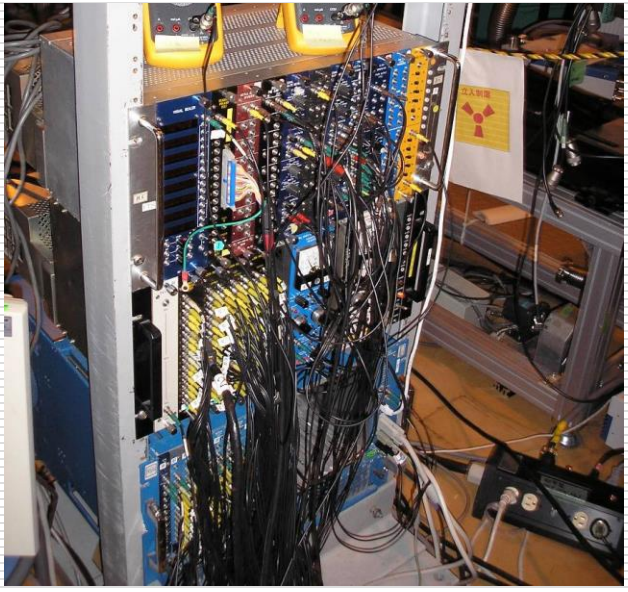
近年のフロントエンド回路
と
集積化技術

フロントエンドの変遷

先ず
過去と現在のフロントエンドの写真を
見比べます

フロントエンドの例：過去

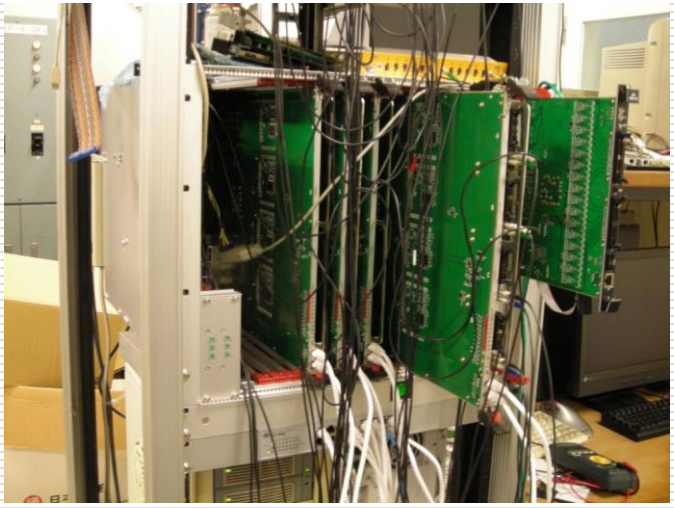
汎用モジュールを多用して組み立てた



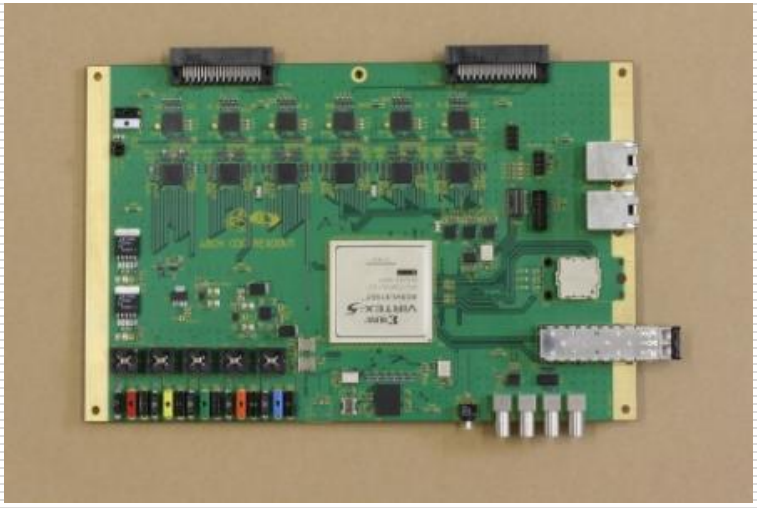
NIMモジュール、CAMACモジュール

フロントエンドの例：現在

自分たちで開発している



COPPER Lite

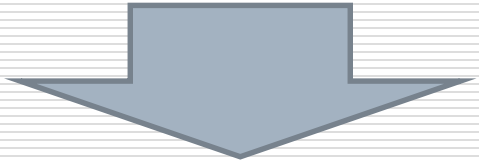


Belle II CDC readout module

集積化がキーワード

背景

- 実験からの要求
  - 高精度測定の要求
  - CH密度の増加（CH数増加）
  - データ量増加

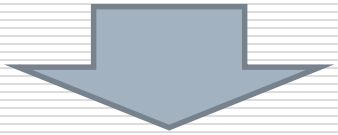


- 集積化
- 高速化

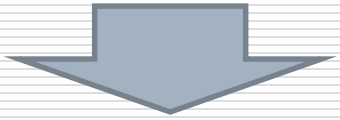
集積化による影響

集積化する為には用途に合わせて特化させる必要あり

パラメータを実験に合わせて作るので、使用用途が限定される



検出器信号を処理する回路は特殊



（実験毎に）自分たちで開発する必要がある

集積化技術

- Application Specific IC (ASIC)
  - トランジスタレベルから設計する集積回路
  - アナログ回路、デジタル回路の実装が可能

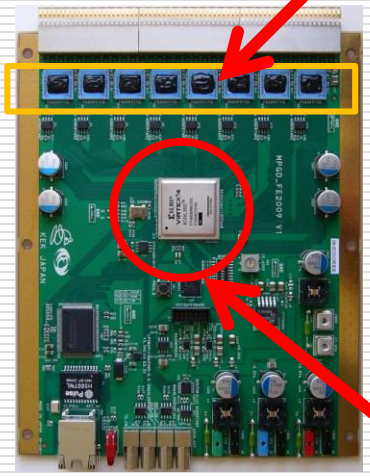
- Field Programmable Gate Array (FPGA)
  - ソフトウエアの様に書き換え可能な集積回路
  - デジタル回路のみ実装可能

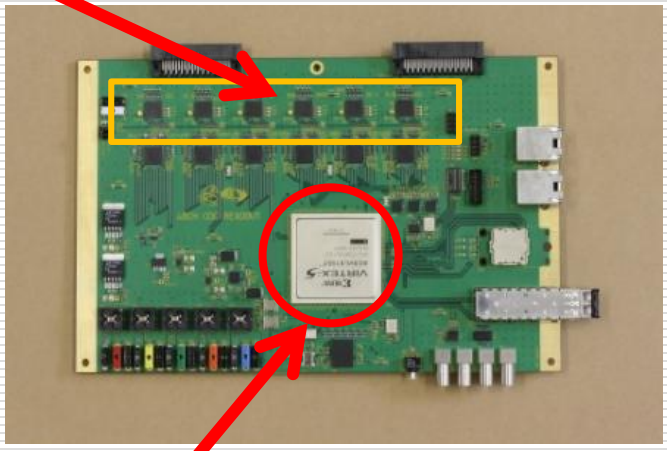
どちらもICだが開発方法が異なるので分けた

ASIC,FPGA使用例

現在のフロントエンド・ボードの主要部品はASICとFPGAのみと言っても過言ではない

ASIC



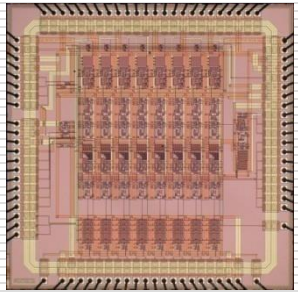


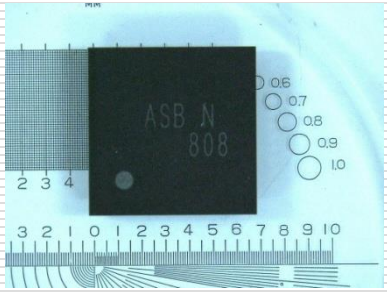
FPGA

Application Specific IC (ASIC)

私達がアナログ回路集積化できる唯一の方法

- 独自回路を集積回路化する技術
- アナログおよびデジタル回路を実装可能
- トランジスタ･レベルから設計可能
- 開発費用と開発期間が問題





ASIC実装の長所と短所

- 長所
  - 大量生産時に安価にできる
  - 技術的に高度な事が出来る
    - 高速回路、高精度回路、アナデジ混載など
- 短所
  - 開発期間が長い
    - 費用、開発効率の面で問題
  - 試作費用が高い

FPGA

デジタル回路を簡単に集積化できる！

- 集積回路
  - 主にデジタル回路
  - 例えば、2×2 cm程度の大きさ
- ユーザーが回路をダウンロードして使用
  - ソフトウエアの様に使用できる
  - ソフトウエアではないので注意！

FPGA実装の長所と短所

- 長所
  - 簡単にICを作れる
  - 回路修正が容易
- 短所
  - デジタルのみ
  - 単価が比較的高価

デジタルのみを考えるとASICと比較しても
価格性能とも見劣りしなくなっている

補足） 私たちが使用できるASICプロセスは数世代古いものになる。一方、FPGAは最先端プロセスで製造されるので私たちが使用できる技術として比較するとFPGAの性能の方が高いまたは同等になっている

FPGAの技術動向

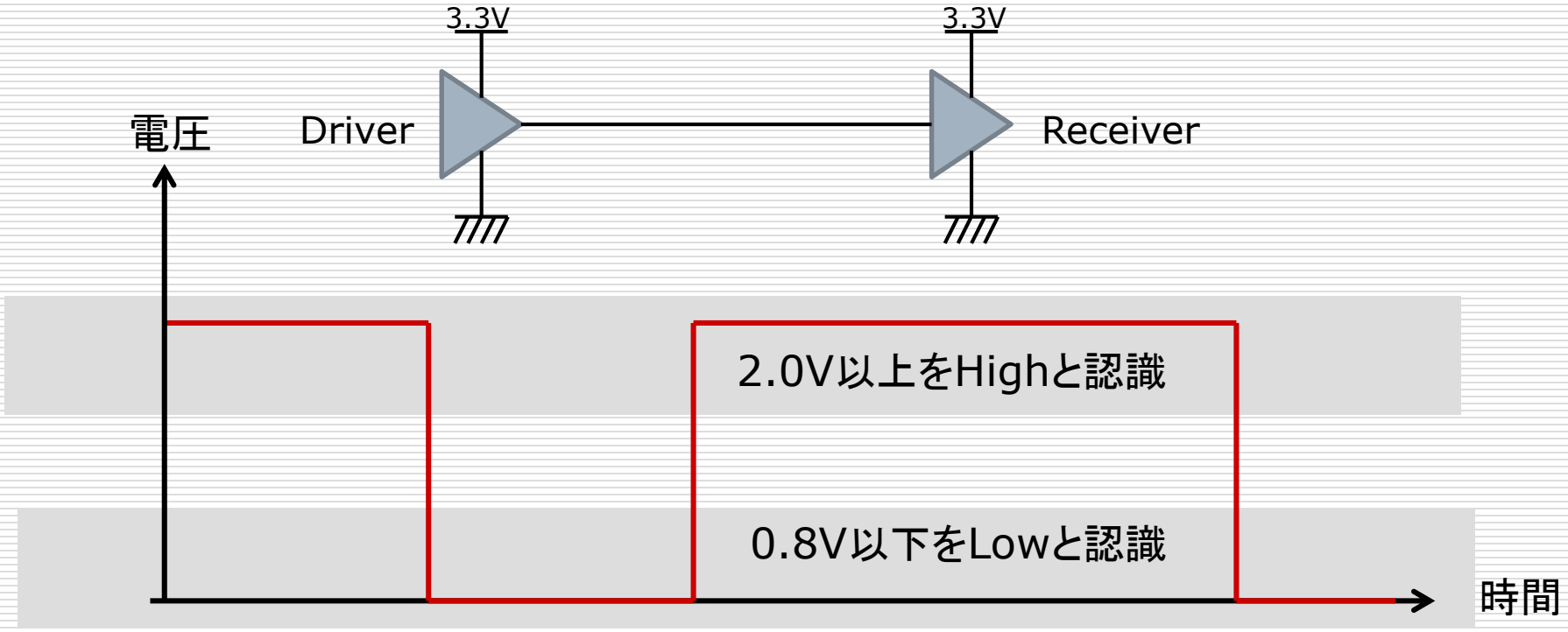
FPGA上にマイコンシステムを構築

- 様々な専用機能が搭載されている
- 組み込みCPU
  - Linuxを動かす事もできる
- Digital Signal Processing (DSP)
  - 積和演算ユニットの事
- 高速シリアル通信
  - ~10Gbps
- メモリコントローラ
  - DDR2/3 SDRAM

標準IO規格

LVCMOS

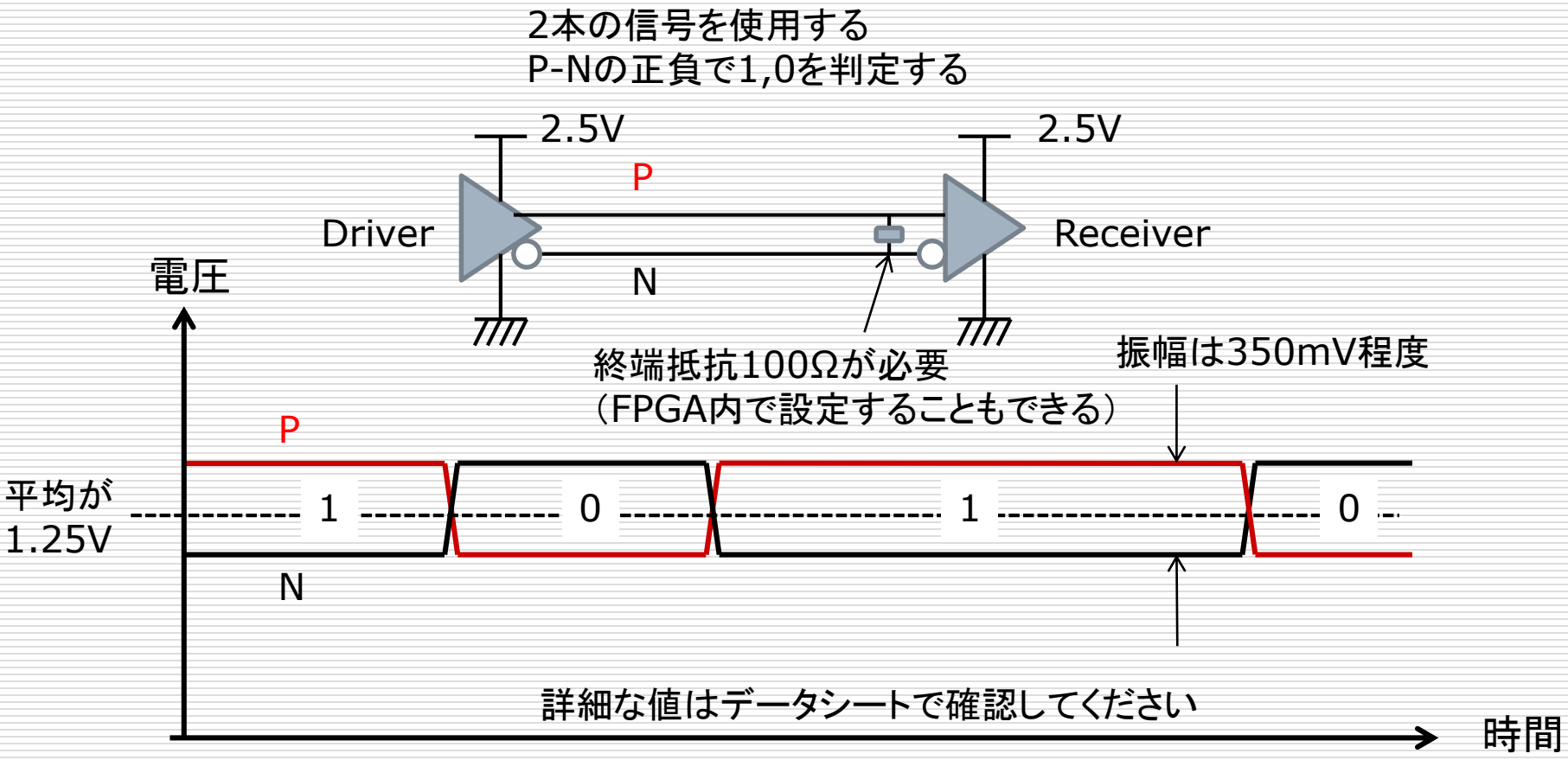
ここではLVCMOS33の例をあげます
（電源電圧により複数決められている）



LVDS

高速動作に適した信号規格

差動信号の代表格



記数法

2進数/16進数の表現

整数の表現

$N = \sum_{k=0}^{n} a_k B^k$　Bは基数, akは係数∋{0,…,B-1}

akを小さいkから右から左へ順に並べて表現する

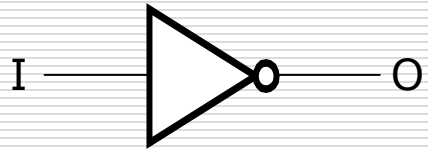
10進表現と区別するために16進表現では0x, 2進表現では0bを付けることがある

10進表現	16進表現	2進表現	10進表現	16進表現	2進表現
0	0x0	0b0	8	0x8	0b1000
1	0x1	0b01	9	0x9	0b1001
2	0x2	0b10	10	0xA	0b1010
3	0x3	0b11	11	0xB	0b1011
4	0x4	0b100	12	0xC	0b1100
5	0x5	0b101	13	0xD	0b1101
6	0x6	0b110	14	0xE	0b1110
7	0x7	0b111	15	0xF	0b1111

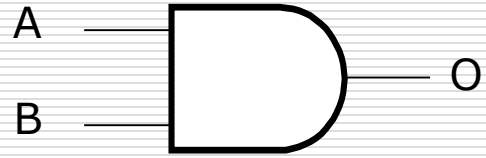
デジタル回路の基本4要素

デジタル回路の基本要素

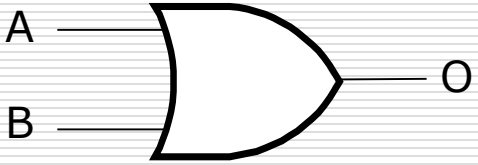
インバータ



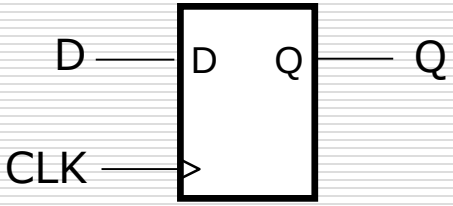
ANDゲート



ORゲート



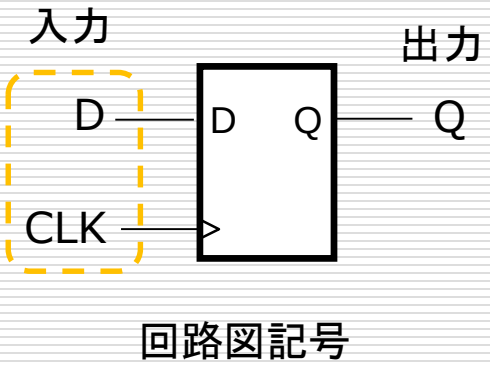
Dタイプ　フリップフロップ（DFF）



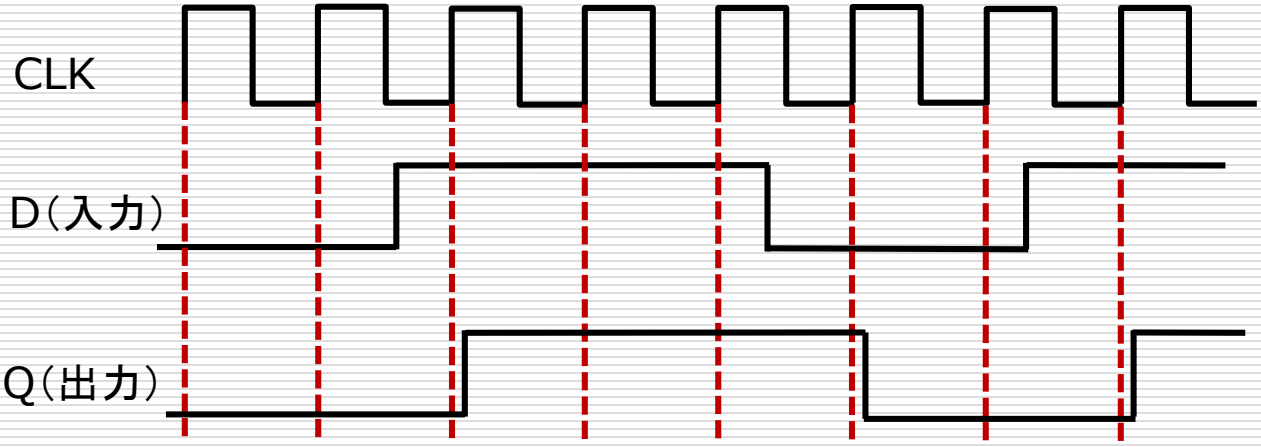
全てのデジタル回路は
この4つの要素（回路）のみで構成されている

CPUなど一見複雑な回路もこの4つの組み合わせで設計可能

Dタイプ　フリップフロップ(DFF)



回路図記号



- CLKの立ち上がり時の入力(D)が出力(Q)される
- 次の立ち上がりまで出力は保持される

デジタル回路の種類

デジタル回路の種類

- 出力が過去の履歴に依存しない回路
  - 常に入力信号の組み合わせのみで出力が決定
  - 組み合わせ回路と呼ばれる

- 出力が過去の履歴に依存する回路
  - 過去の履歴と入力信号の組み合わせにより出力が決定
  - 順序回路と呼ばれる

組み合わせ回路と順序回路

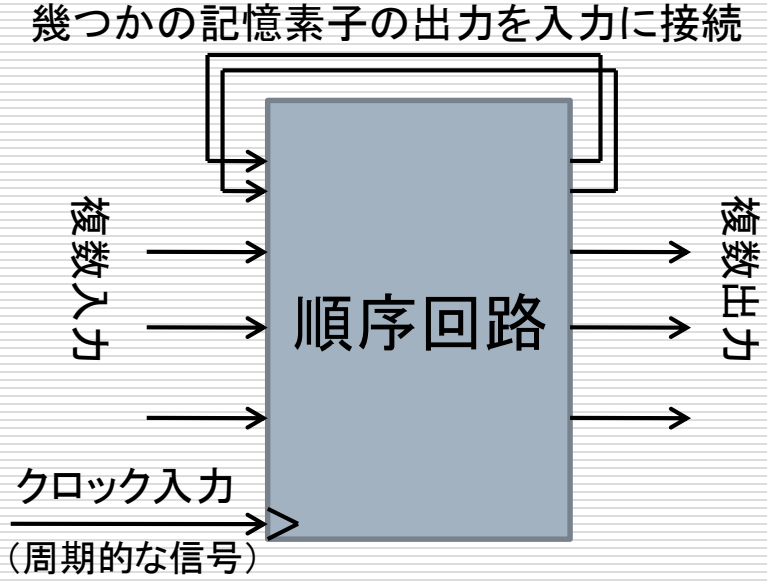
組み合わせ回路

記憶素子を含まない回路

入出力の対応が1:1で決まっている

順序回路

記憶素子を含む回路



組み合わせ回路

組み合わせ回路

- 記憶素子を使わない回路
  - AND, OR, NOTのみで構成される
  - 入力と出力の関係が一意に決定する
  - 入出力対応関係を表で表すことができる

希望する状態を検出する

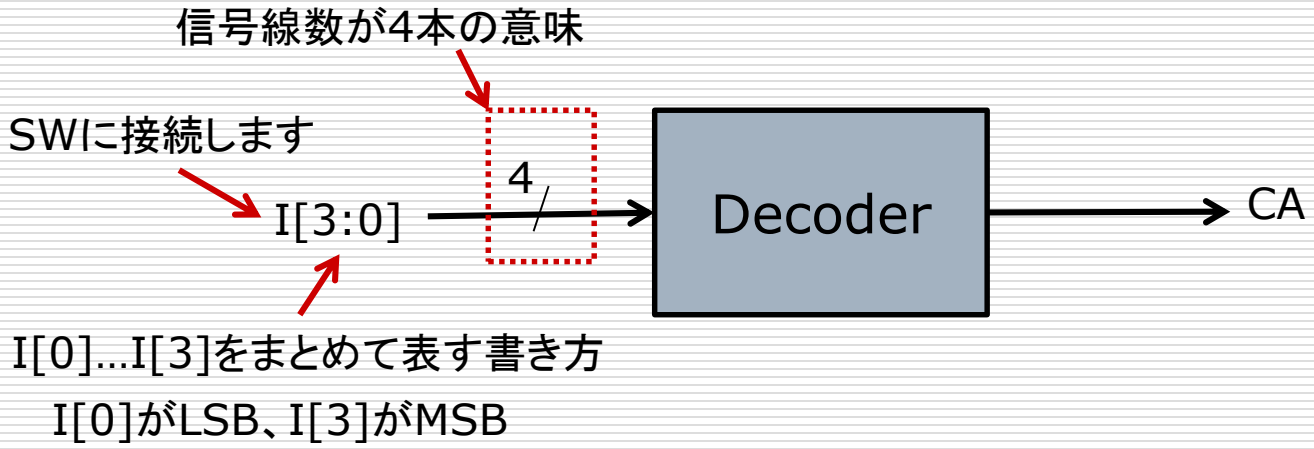
例）検出信号のコインシデンスを取る

組み合わせ回路設計方針

1. 真理値表を書く
2. 各行の動作をANDとNOTを使って条件抽出回路を設計
3. ORを使って上の各回路を足す

設計演習

I=1, I=4, I=11(0xB), I=13(0xD)の時にのみCA=1となる回路を設計してください



真理値表は次ページ

演習問題の真理値表

I[3]	I[2]	I[1]	I[0]	CA
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0

I[3]	I[2]	I[1]	I[0]	CA
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	0

補足

- この方法ではエレガントな回路を設計する事はできません。見た目はかなり悪いです。しかし、必ず設計できます
- 見た目の悪さは実用上問題になりません。
  - 開発ツールが論理回路の最小化などの最適化を行ってくれるからです。
- エレガントな回路設計に興味がある方はデジタル回路の教科書中に必ず現れるカルノー図を使った論理回路の最小化などを参考にしてください。

設計者が理解しやすい方法や記述を使うことが重要です

順序回路

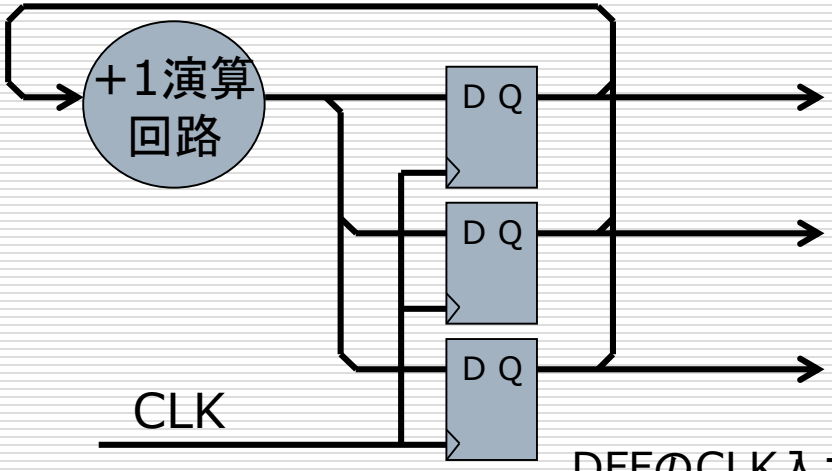
設計手法

- 同じ動きをする回路を異なる二つの方法で設計する事が出来ます。

- 異なる2つの設計手法があります
  - 非同期設計法
  - 同期設計法←FPGAではこちらが主流

カウンター回路を例に2つの設計手法の違いを見てみましょう

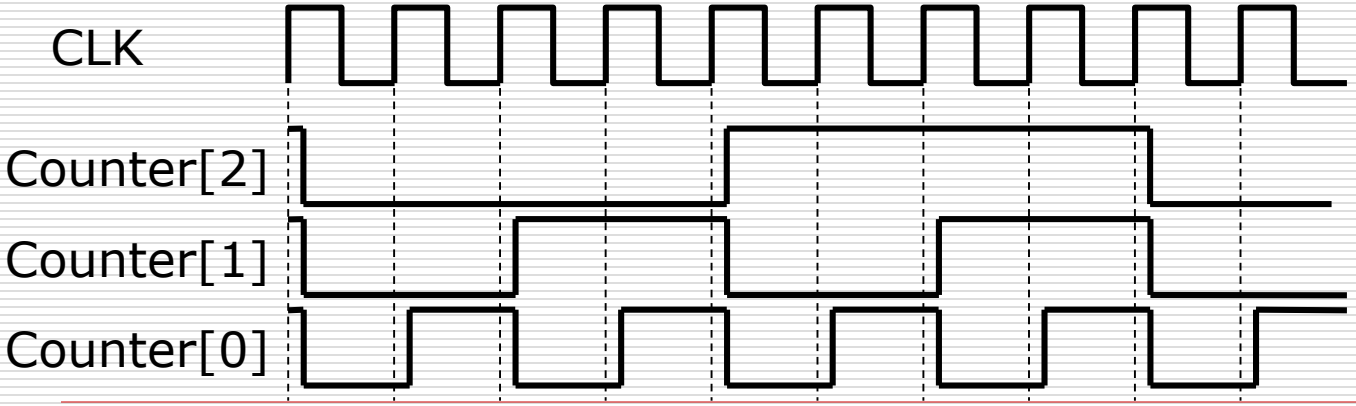
同期カウンタ



特徴
- 回路が複雑
- 遅延がある範囲内におさまる

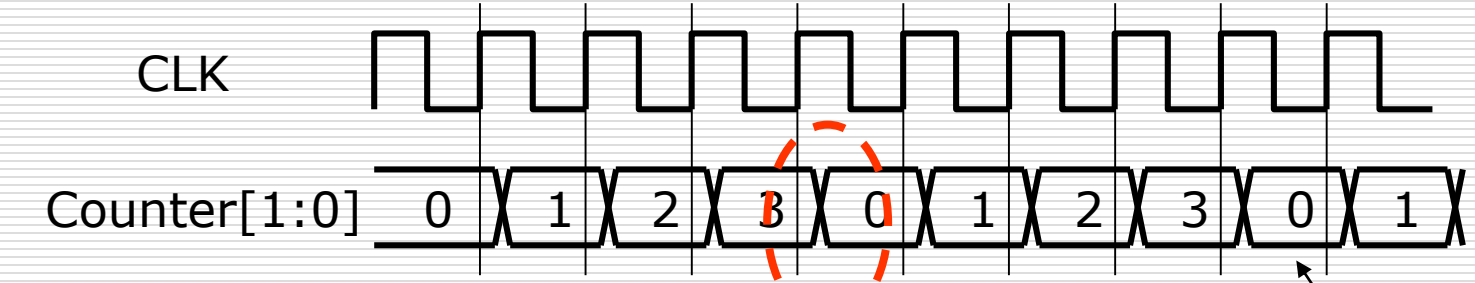
良く使用される

DFFのCLK入力にシステム共通のクロック信号が入力されている



この様な回路を
同期回路
と呼びます

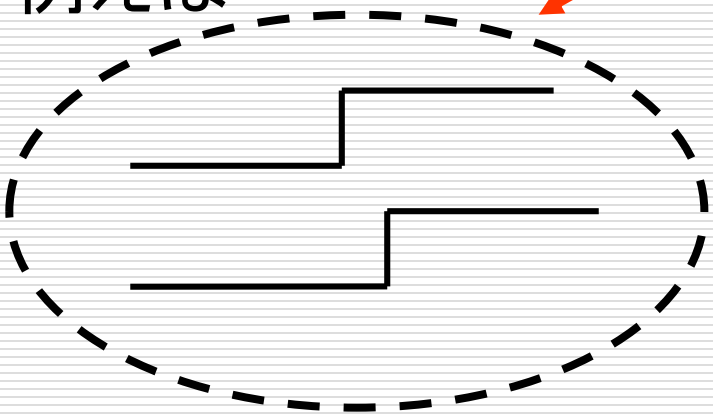
同期カウンター



DFFはクロックの立ち上がりでしか動作しない

符号（複数信号）はこのように書く

例えば



次のCLKの立ち上がりまでに決定すれば
途中経過は考慮する必要なし

設計が簡単になった！！

実際は値が決まるまでに時間が掛かっている

履歴

- 2012/5/17 第1.0版 ISE13.4対応　内田智久(Esys, KEK/総研大), 林達也（大阪大学）
- 2014/8/7　第2.0版 章構成変更　内田智久(Esys, KEK/総研大)
- 2015/7/31　第3.0版 章構成変更　内田智久(Esys, KEK/総研大)
- 2015/8/5　第3.1版「データ集システムとデジタル回路」追加　内田智久(Esys, KEK/総研大)
- 2015/12/03　第3.2版 ページ構成変更　内田智久(Esys, KEK/総研大)
- 2015/12/04 第3.3版 組み合わせ回路演習問題追加　内田智久(Esys, KEK/総研大)
- 2016/01/27 第3.4版 補足説明追加　内田智久(Esys, KEK/総研大)
- 2016/07/14 第3.5版 予習資料との重複内容を削除(Esys, KEK/総研大)