

デジタル回路入門 Part 3

2010年7月28日

内田智久 (e-sys, IPNS, KEK)

先端エレクトロニクスDAQセミナー'10

順序回路

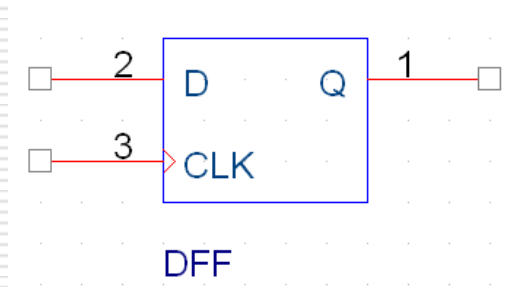
- 記憶素子を使う回路
 - 組み合わせ回路+DFF
- DFFのクロック立ち上がりで出力が変化する
 - 入力と出力で次の出力が決定する

複数の状態を持ち、入力により状態間を遷移する

まずは記憶素子単体を学習します

D-type Flip Flop

記憶素子

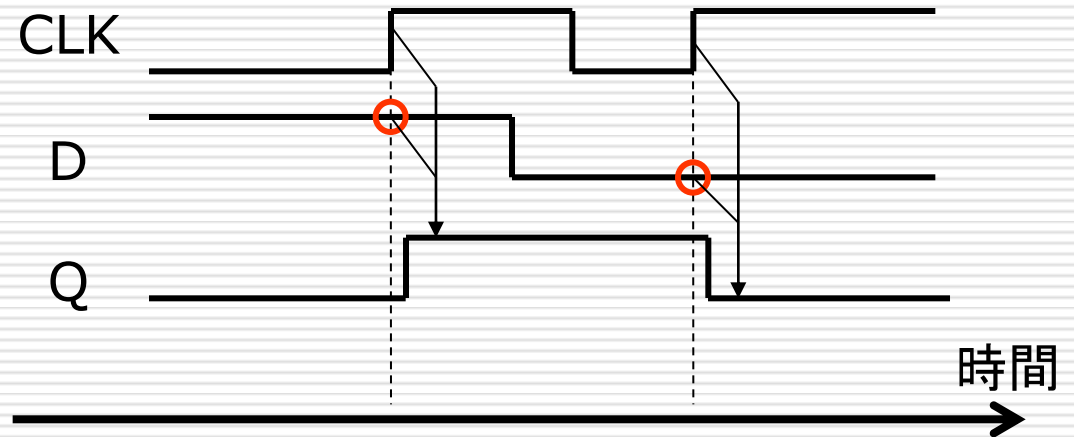


D:入力

Q:出力

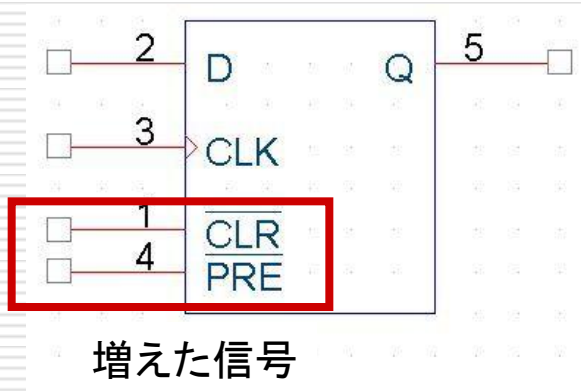
CLK:クロック入力

このような図をタイミングチャートと呼んでいます



CLKの立ち上がりでDを取り込む
次の立ち上がりまで保持する

実際のD-type Flip Flop



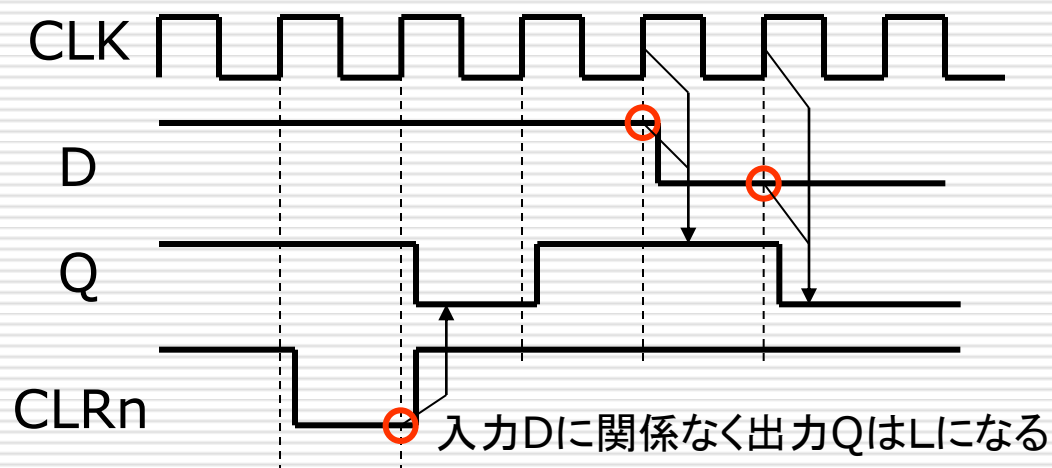
CLRn: クリア

PREn: プリセット

片方のみ使用する事が多い

両方まとめてリセットと読んだりもする
リセット時に1になる、
リセット時に0になる、など

CLRnの動作



CLKに合わせてクリアされるので
同期クリアと呼ばれる
非同期クリアもある

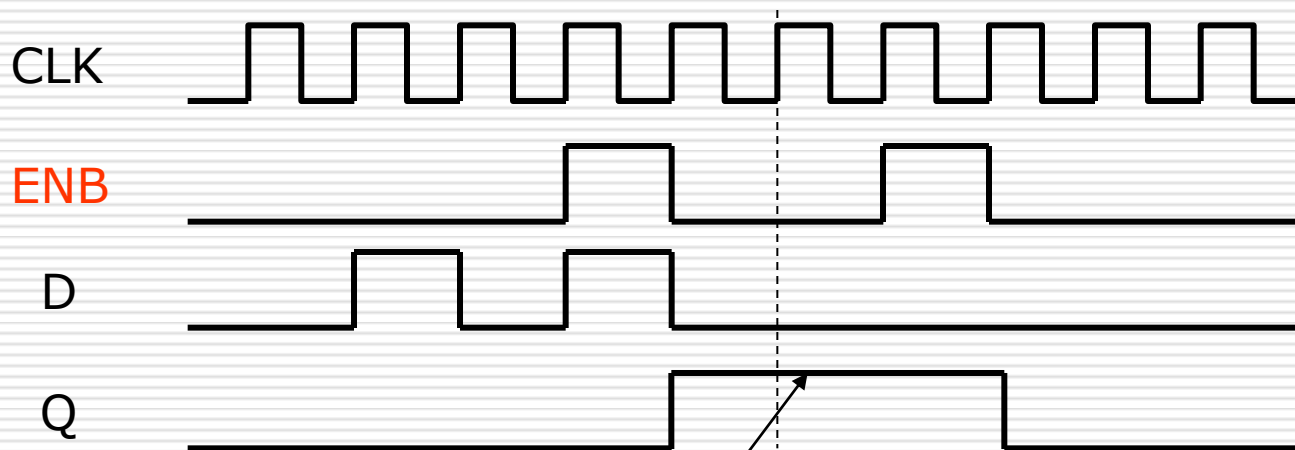
電源投入時の初期値を決める
リセットとして多用される

クロック・イネーブル信号

□ クロック・イネーブル

■ DFFの動作、停止を切り替える為の信号

クロック・イネーブルがHの時のみ
のクロック立ち上がりが有効



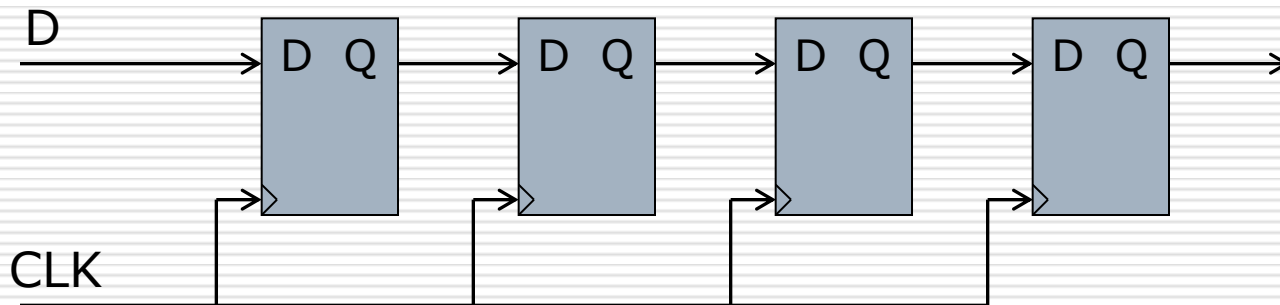
イネーブルがLなので保持されたまま

記憶素子を使ったブロック

- Shift register
 - シリアル/パラレル変換に良く使われる
 - 2^n 計算にも使われる
- Sample and hold
- Counter
 - 多ビット出力
 - クロック毎に出力が1ずつ増加する回路
 - クロックの数を数える回路
 - 同期/非同期型
- Memory
 - RAM
 - 読み書き可能な記憶素子の集まり
 - ROM
 - 読み込み専用の記憶素子の集まり
 - FIFO
 - 書き込んだ順番に読み出す事ができる記憶素子の集まり

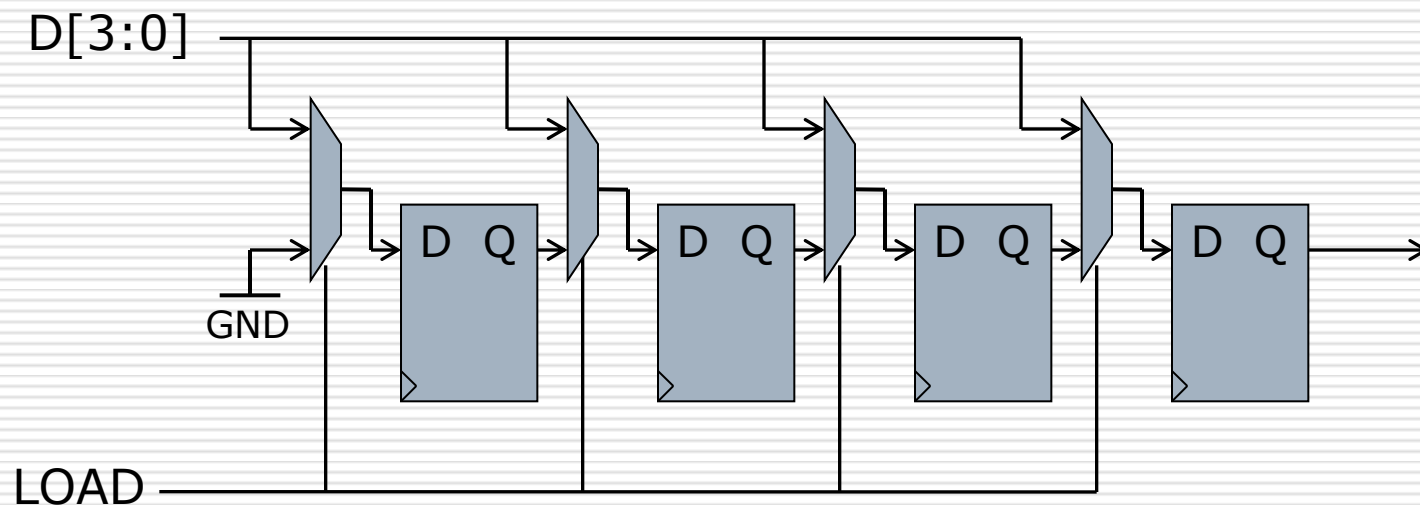
Shift register

DFFを直列接続した物



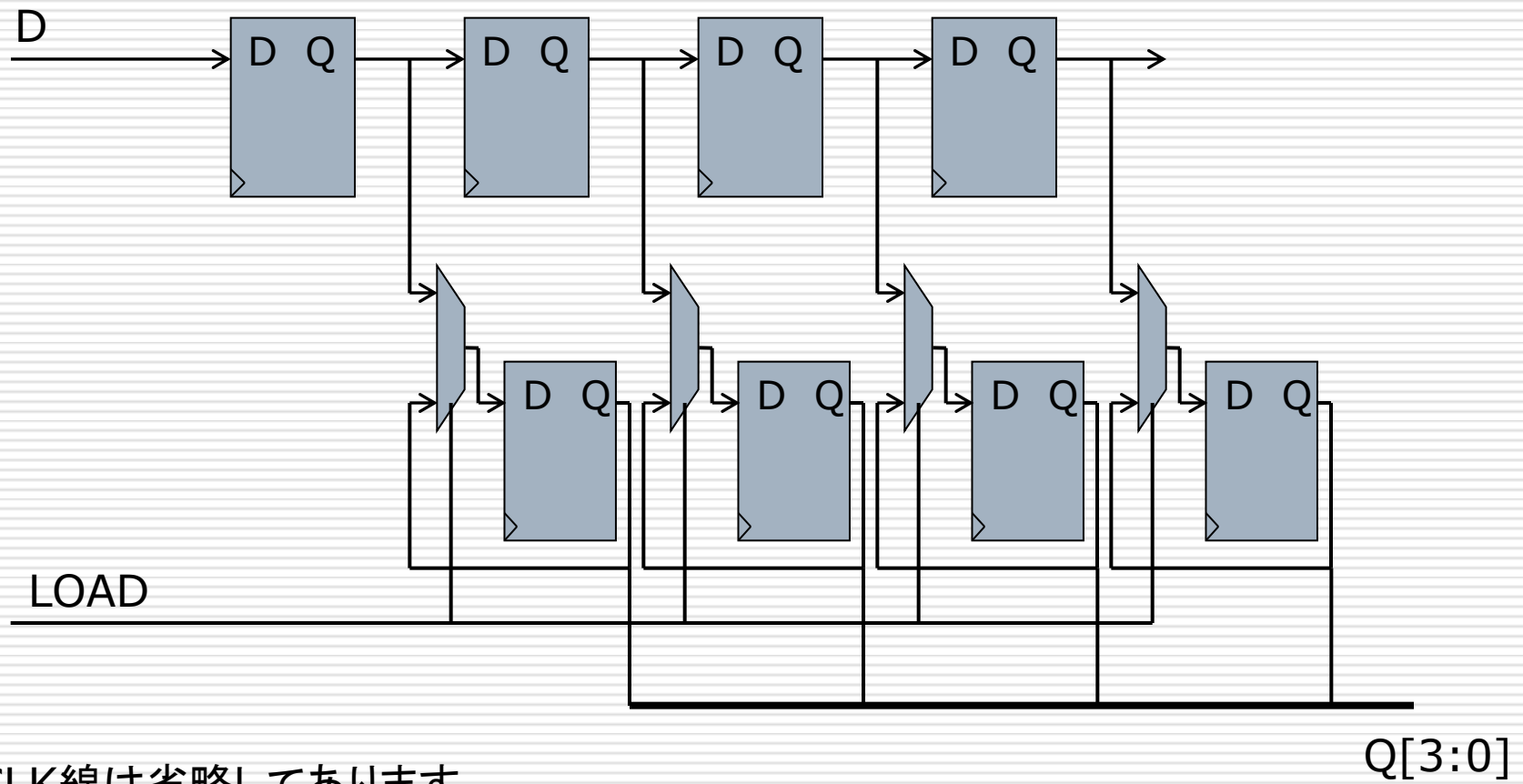
クロック単位の遅延生成(時間調整)
シリアル変換
カウンタと見なす事もできる

Parallel to Serial変換



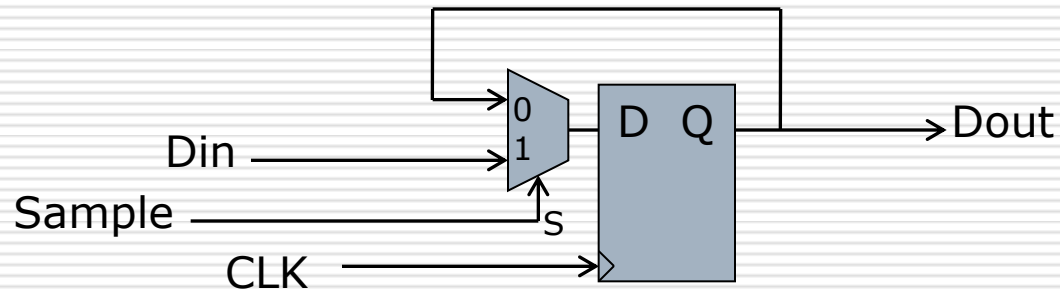
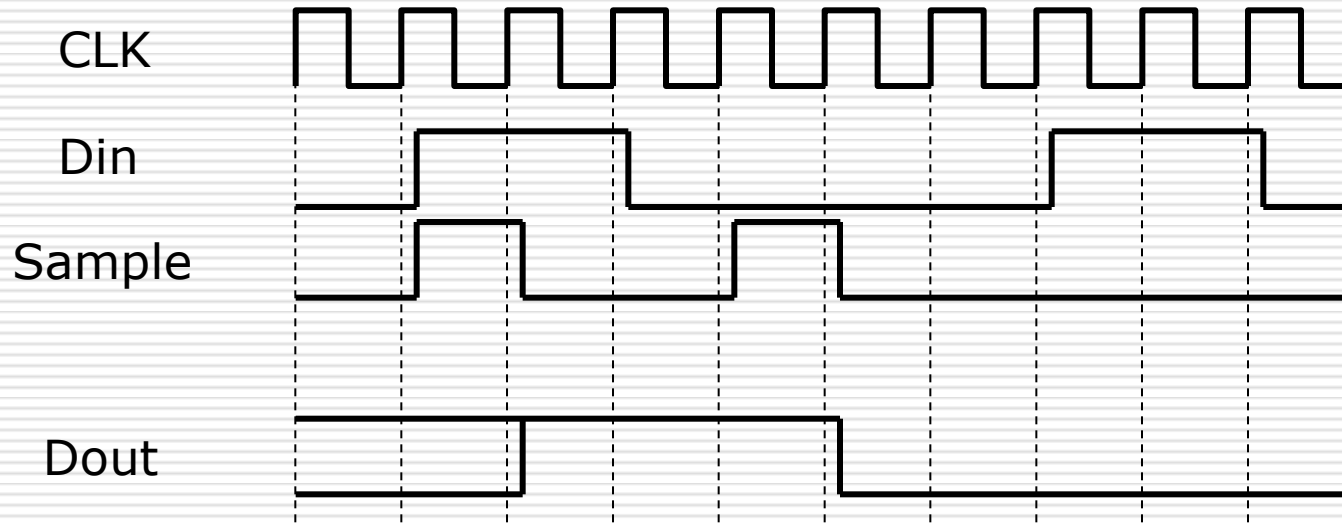
CLK線は省略してあります

Serial to parallel 変換



CLK線は省略してあります

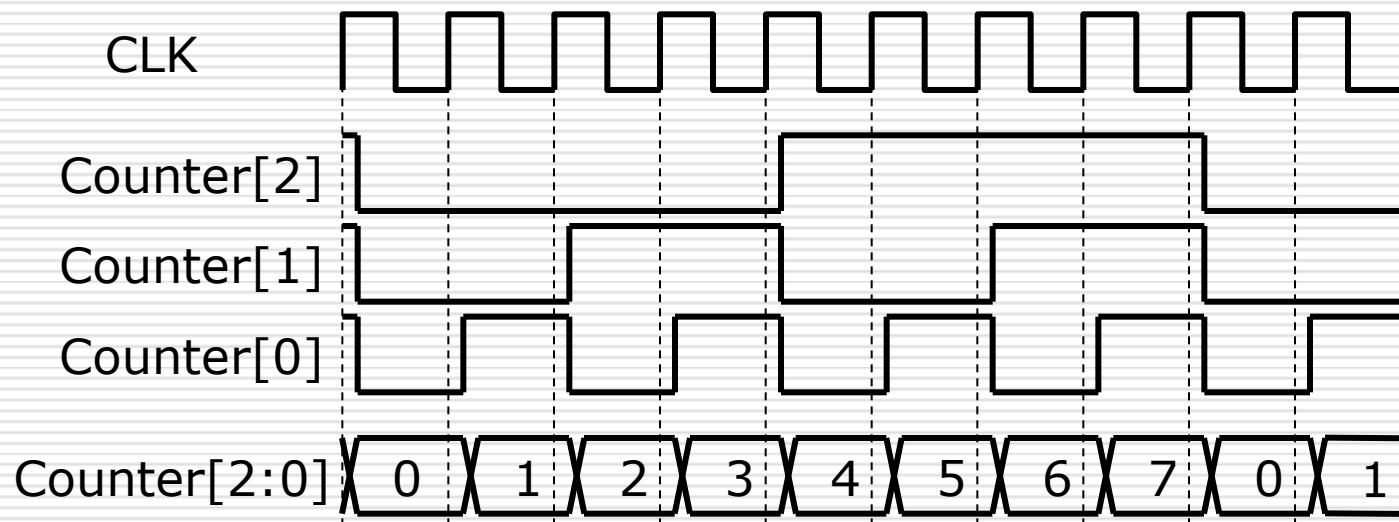
Sample and hold



Counter

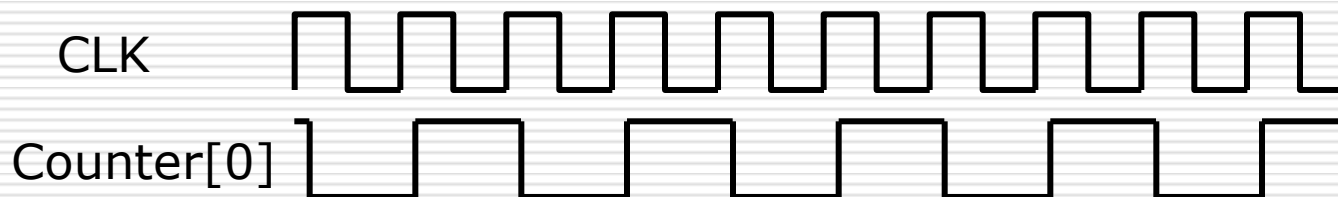
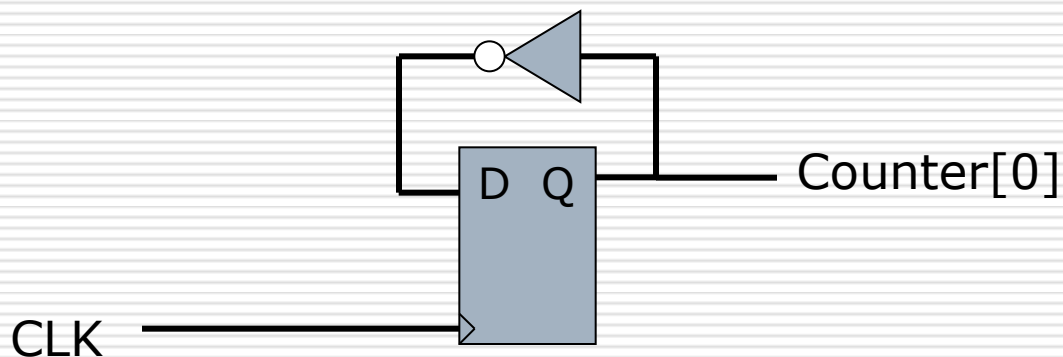
クロック数を数える回路

イベントの回数を数えたり、
時間を計測する時、
あるクロックを分周する時に使用します



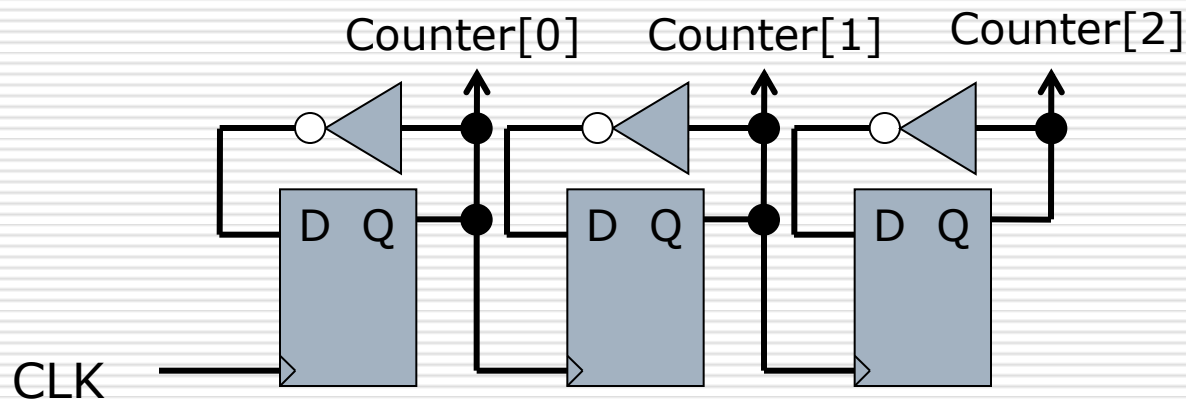
DFFを使った2分周器

DFFを1個使うと2分周出来る！



上位ビット(4分周)はどうする？

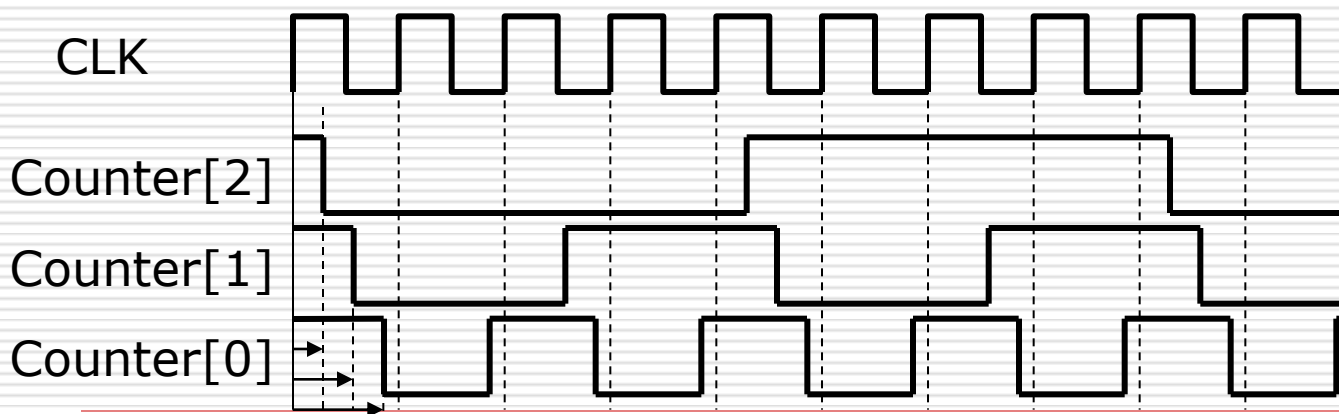
非同期カウンタ



特徴

- 回路が簡単
- しかし欠点がある...

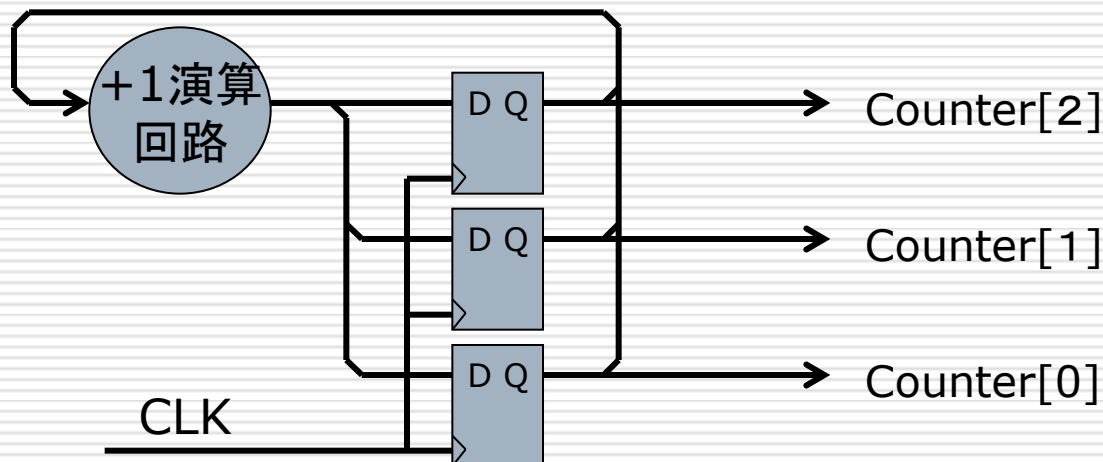
最近は使用しない。



このような回路を
非同期式回路
と呼びます

上位ほど遅延する

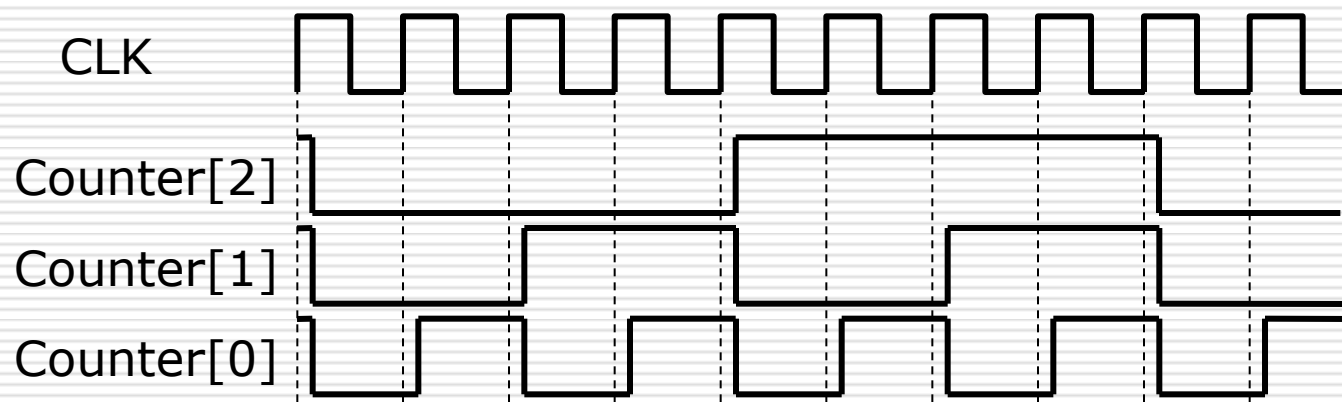
同期カウンタ



特徴

- 回路が複雑
- 遅延がある案以内におさまる

良く使用される

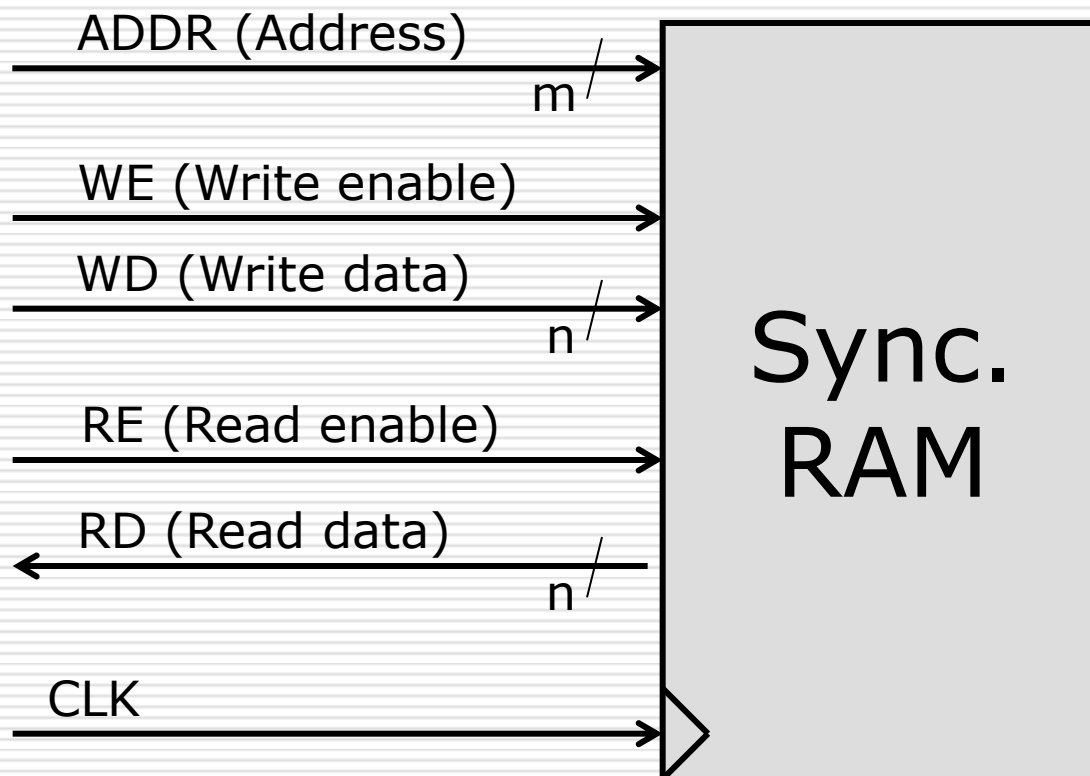


このような回路を
同期式回路
と呼びます

Memory

- 記憶方式とアクセス方式により様々な種類がある
- 記憶素子の集合体
 - アドレス信号で素子を指定して読み書きする
- Random Access Memory (RAM)
 - 揮発性メモリ
 - SRAM (Static)
 - DRAM (Dynamic)
- Read Only Memory (ROM)
 - 不揮発性メモリ
- ここではFPGA内の同期RAM, ROMのみ紹介します

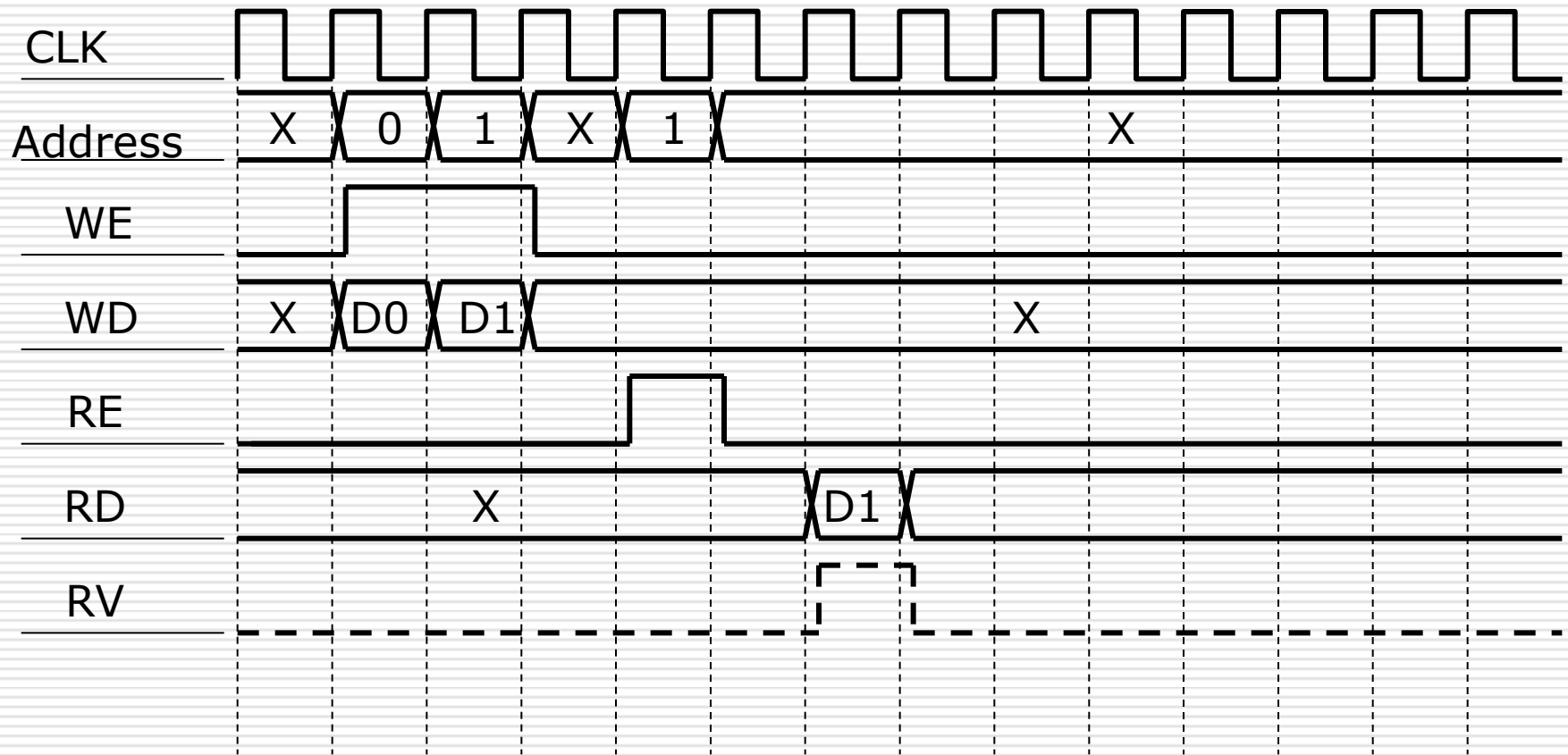
RAM (1)



記憶素子の数でアドレス信号の本数が決まる

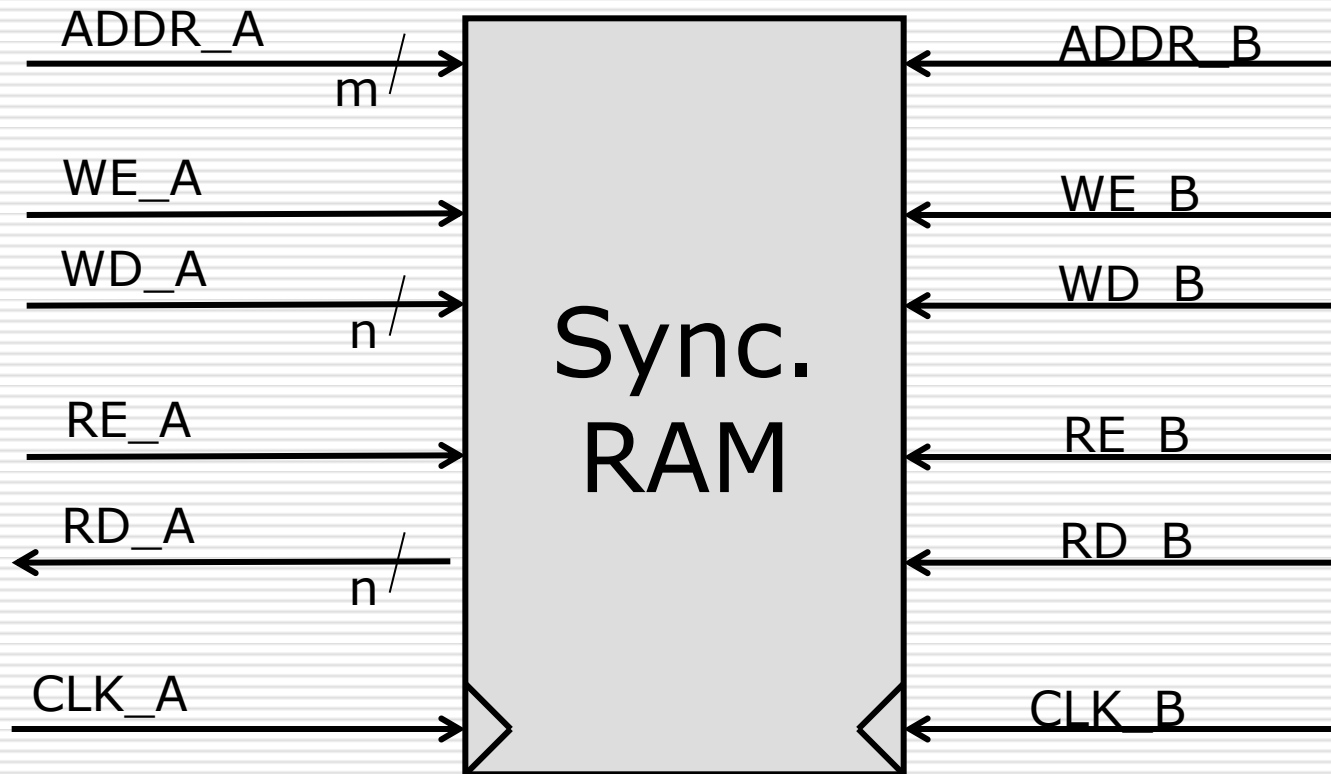
一度にアクセスできる記憶素子の数でデータ信号の本数が決まる

RAM (2)

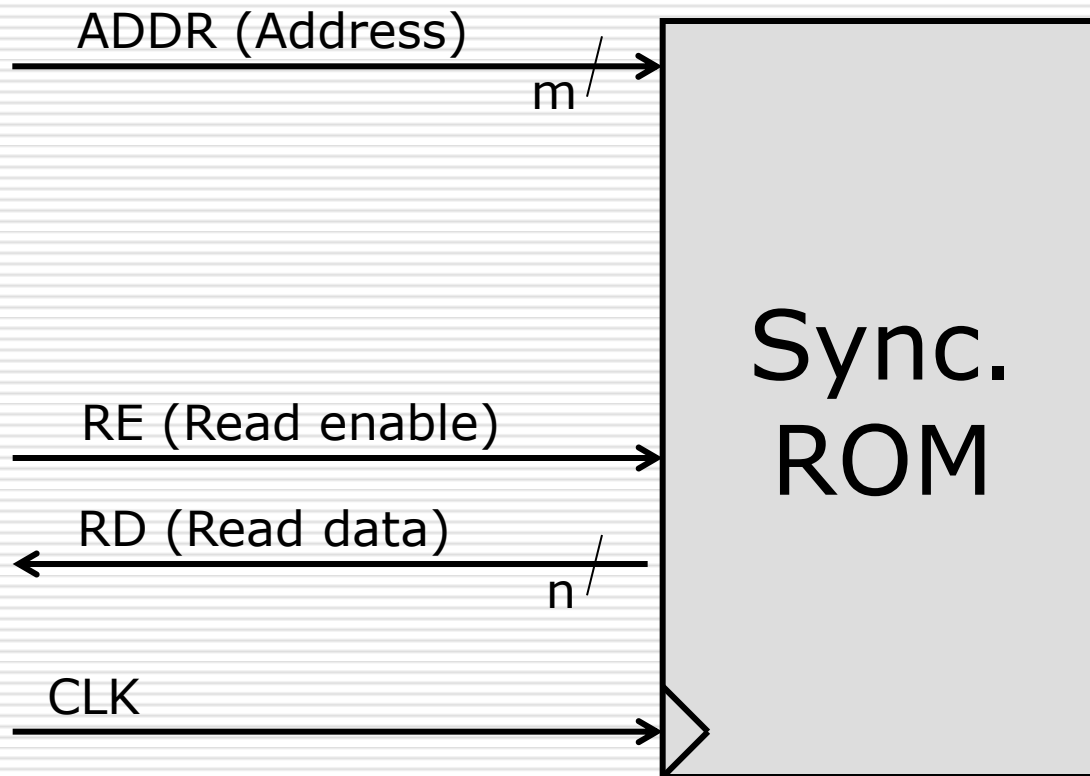


Dual-port RAM

同じ記憶素子を独立した2ポートからアクセスできる



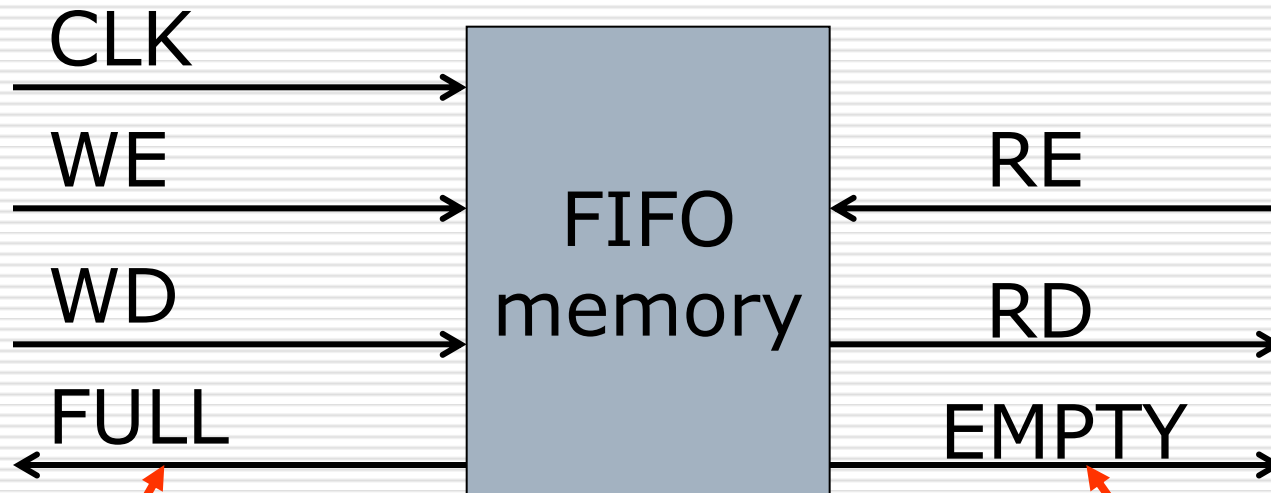
ROM



FIFO Memory (1)

書いた順番に読み出すことが出来るメモリ

順番にしか読めないなのでアドレスが無い

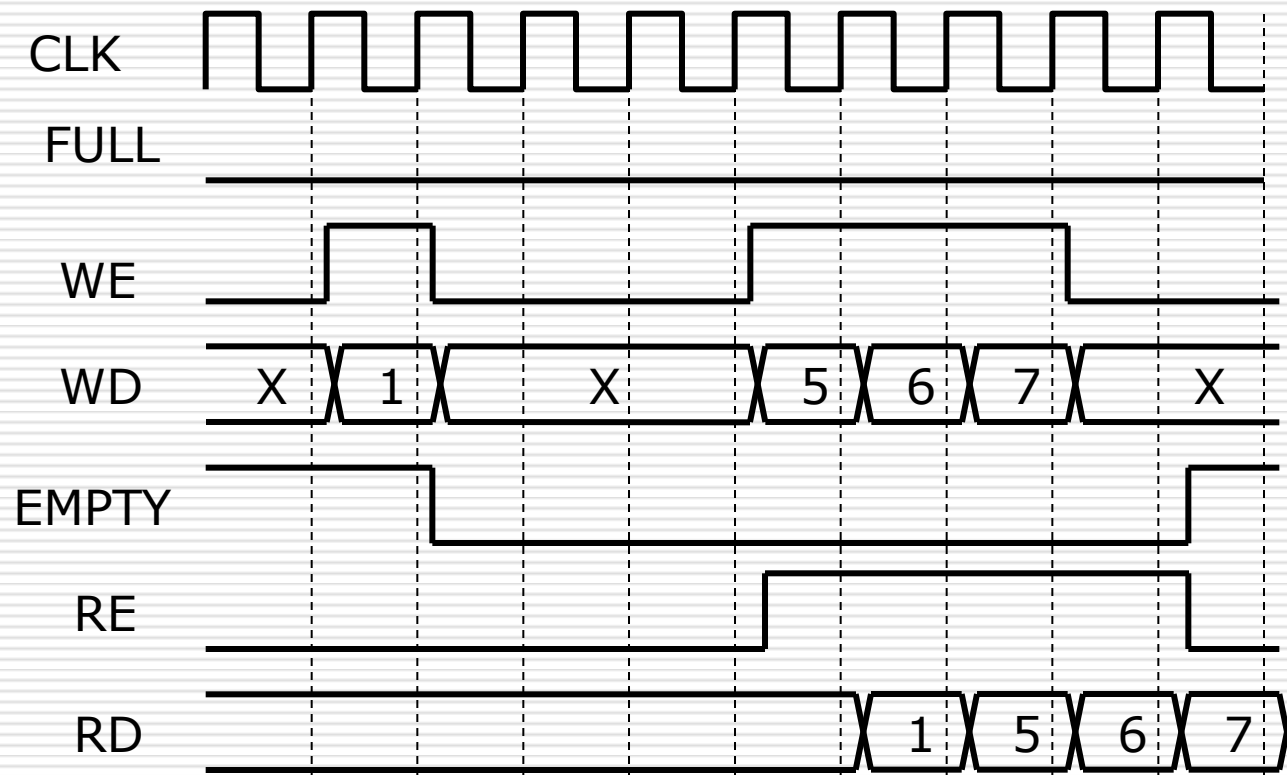


書く場所がなくなるとHになる

空の時Hになる

メモリ容量は有限なので書き込みのみしていると一杯になる

FIFO Memory (2)



書いた順番に読める(遅れて出てくる)

このセクションのまとめ

代表的なデジタル回路

- デジタル回路は4要素で構成されている
- 組み合わせ回路を使った回路ブロック
- 記憶素子を使った回路ブロック

同期式回路

目的: 同期式回路設計の特徴を理解する

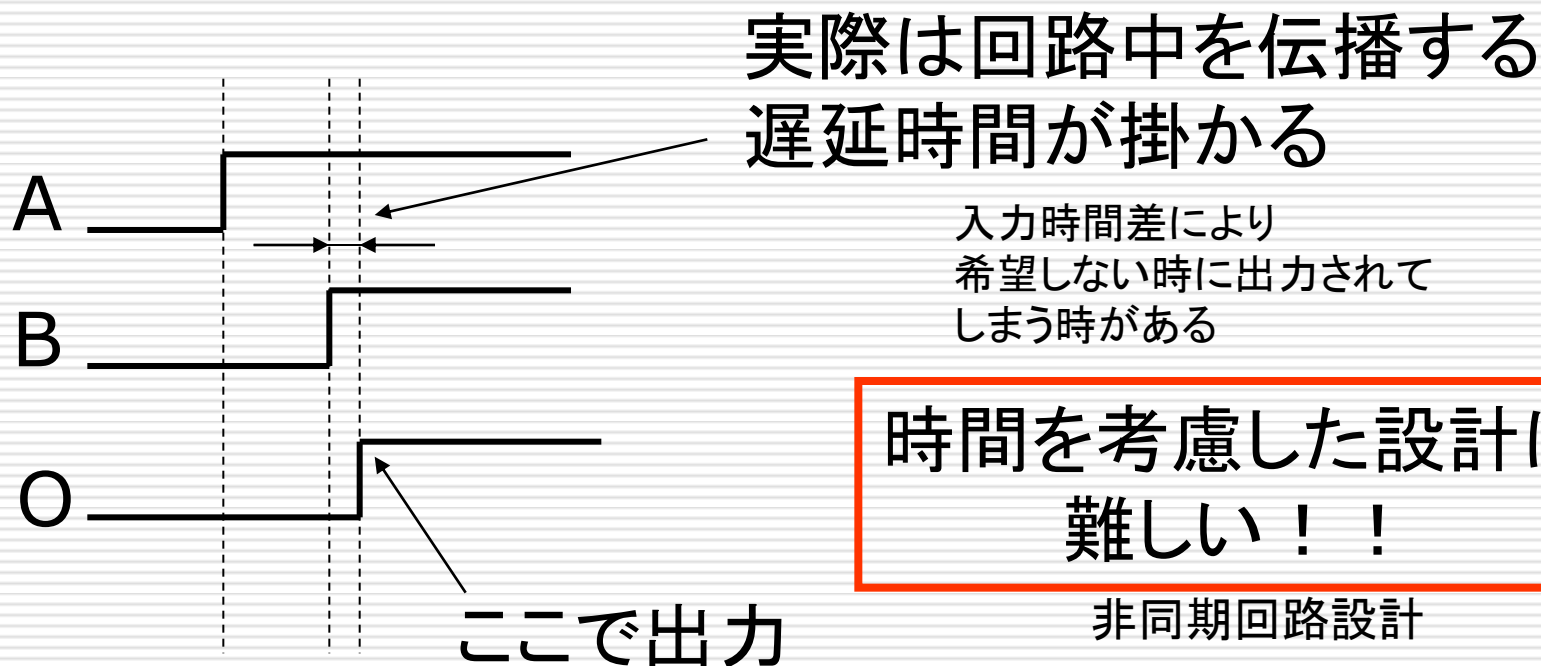
同期式回路

- このセミナーでは同期式回路設計を採用します
 - 同期式回路については後で説明します
- 最初に採用理由を説明します
 - 組み合わせ回路の注意点(問題点)
 - その解決方法として同期回路を提示します

では、
組み合わせ回路設計の注意点
から見てみましょう

組み合わせ回路設計の注意点

- 入力の時間差に注意
- 例えば2入力AND



時間を考慮した設計は
難しい！！

非同期回路設計

解決案

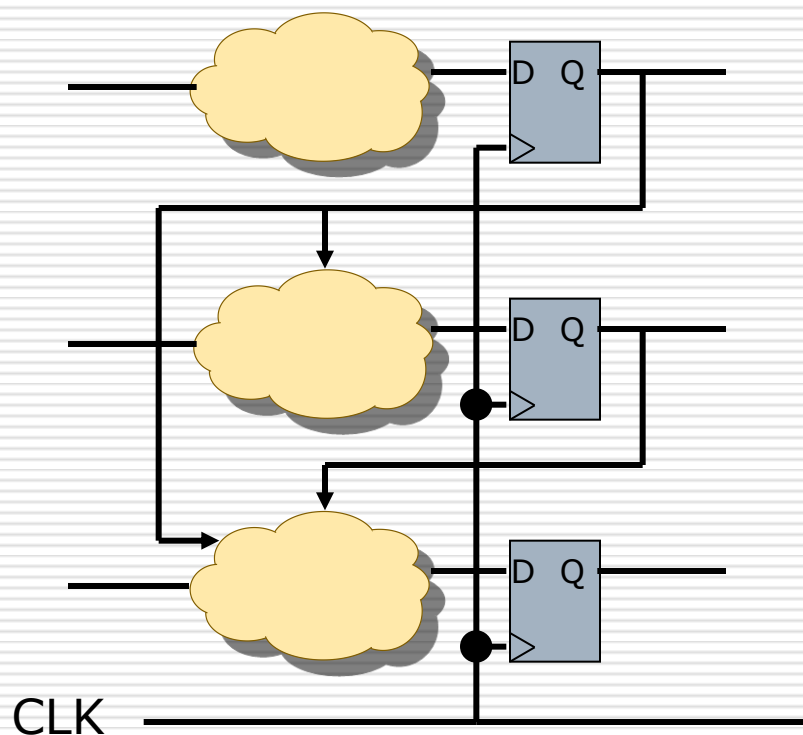
□ 問題点

- 出力タイミングは様々な理由で決定する
 - 正確なタイミングを知るのが難しい
 - デバイス個性、温度依存など
- 入力信号の時間差は様々な理由で発生
 - 配線遅延
 - 素子の伝播遅延時間
 - 温度依存性は？
 - 考える事は沢山ある

□ 解決案

- 全ての信号にDFFを挿入
- 全てのDFFに共通の周期信号を入力する

同期回路



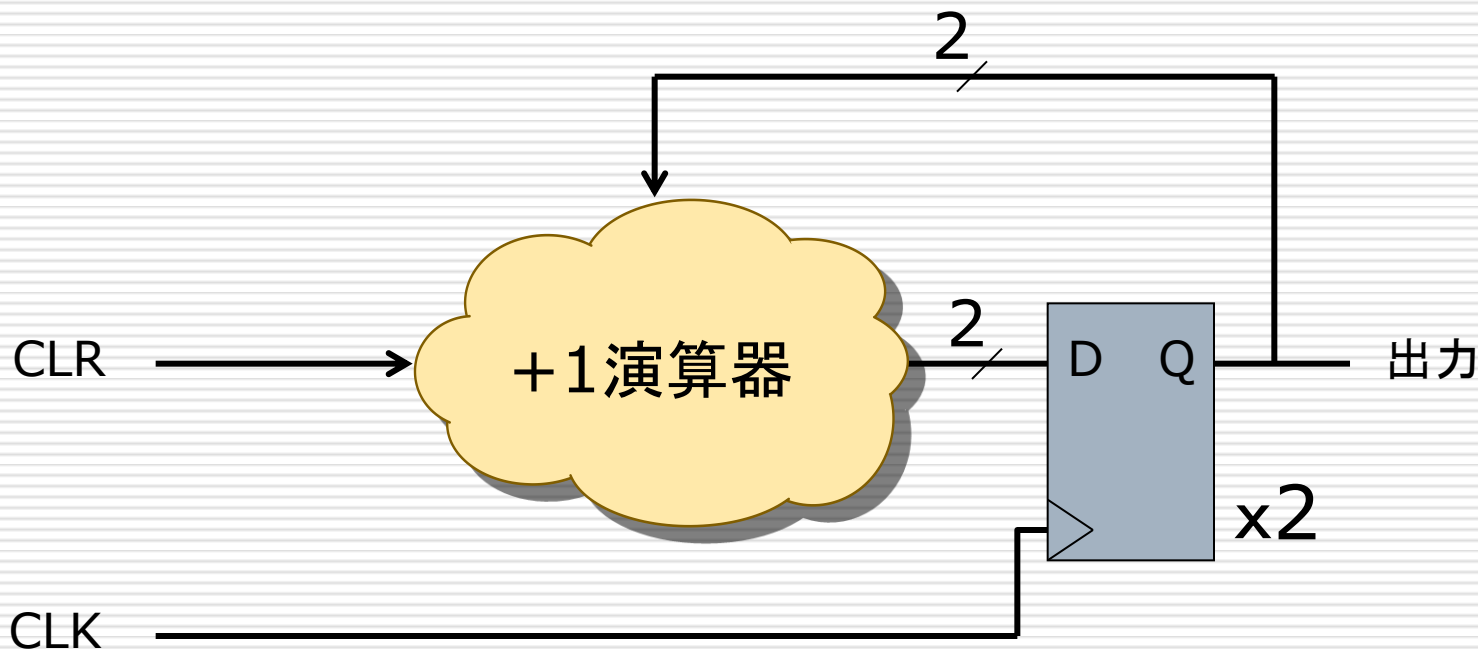
クロック単位で動作を考えれば良いので
タイミング設計が簡単

どんな風に設計しようが出力は
クロックの立ち上がりで変化すると考えて
設計できる

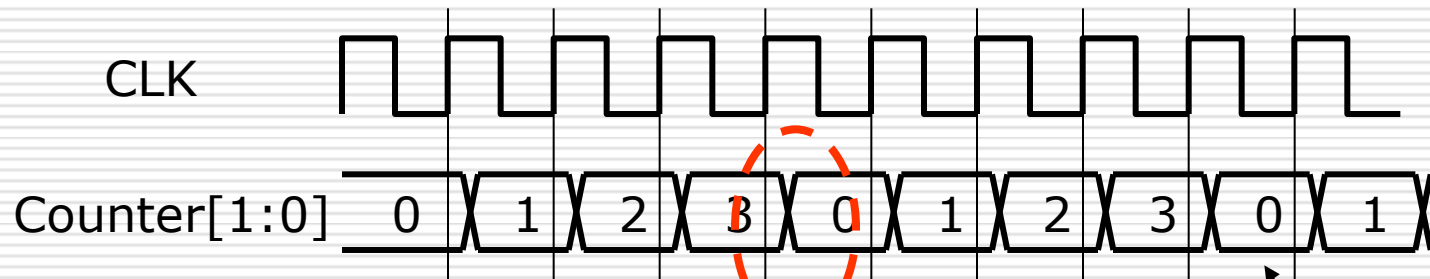
CLKが全てのDFFに入力されている

実際に設計してみよう

- 2bit幅の+1演算器と2つのDFFを使用してカウンタを設計してください

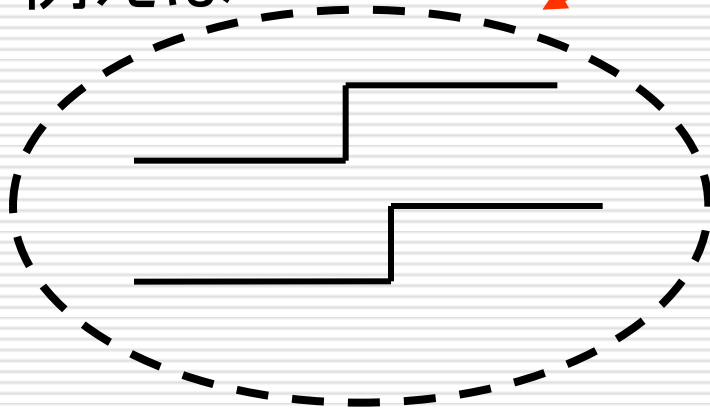


Counter



符号(複数信号)はこのように書く

例えば



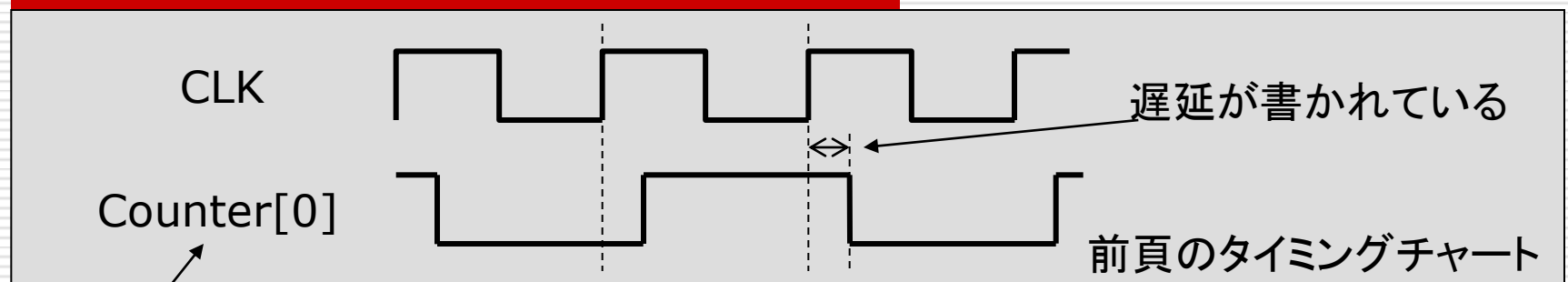
次のCLKの立ち上がりまでに決定すれば
途中経過は考慮する必要なし



設計が簡単になった！！

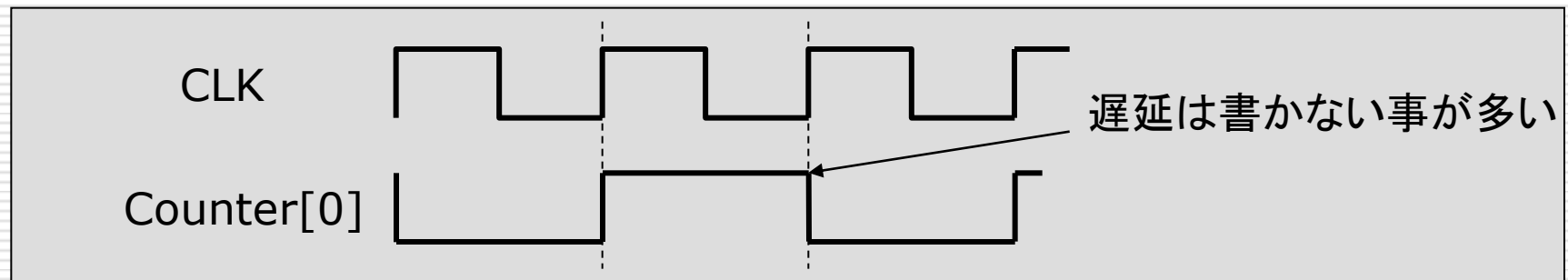
実際は値が決まるまでに時間が掛かっている

同期回路のタイミングチャート



0ビット目のみ
と言う意味

同期回路の長所は
遅延を考えなくて良い事である

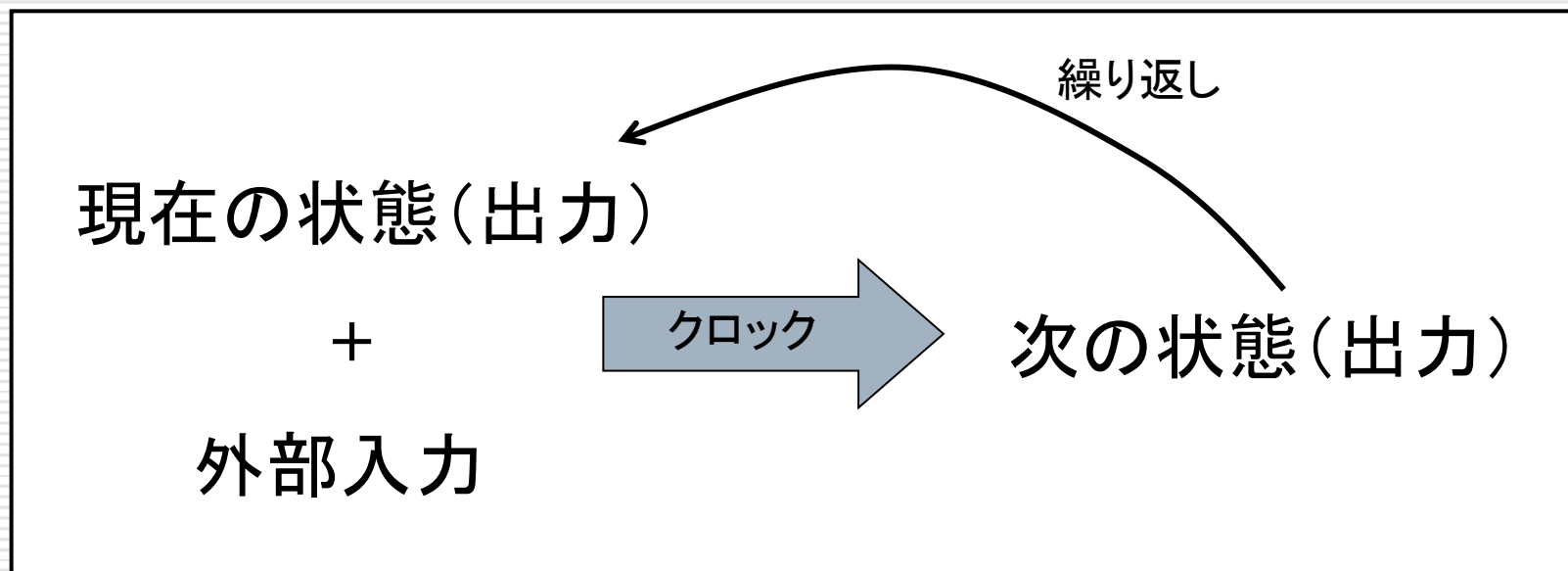


データシートなどを見るときに気をつけてください

カウンターの状態遷移

カウンターの動作は

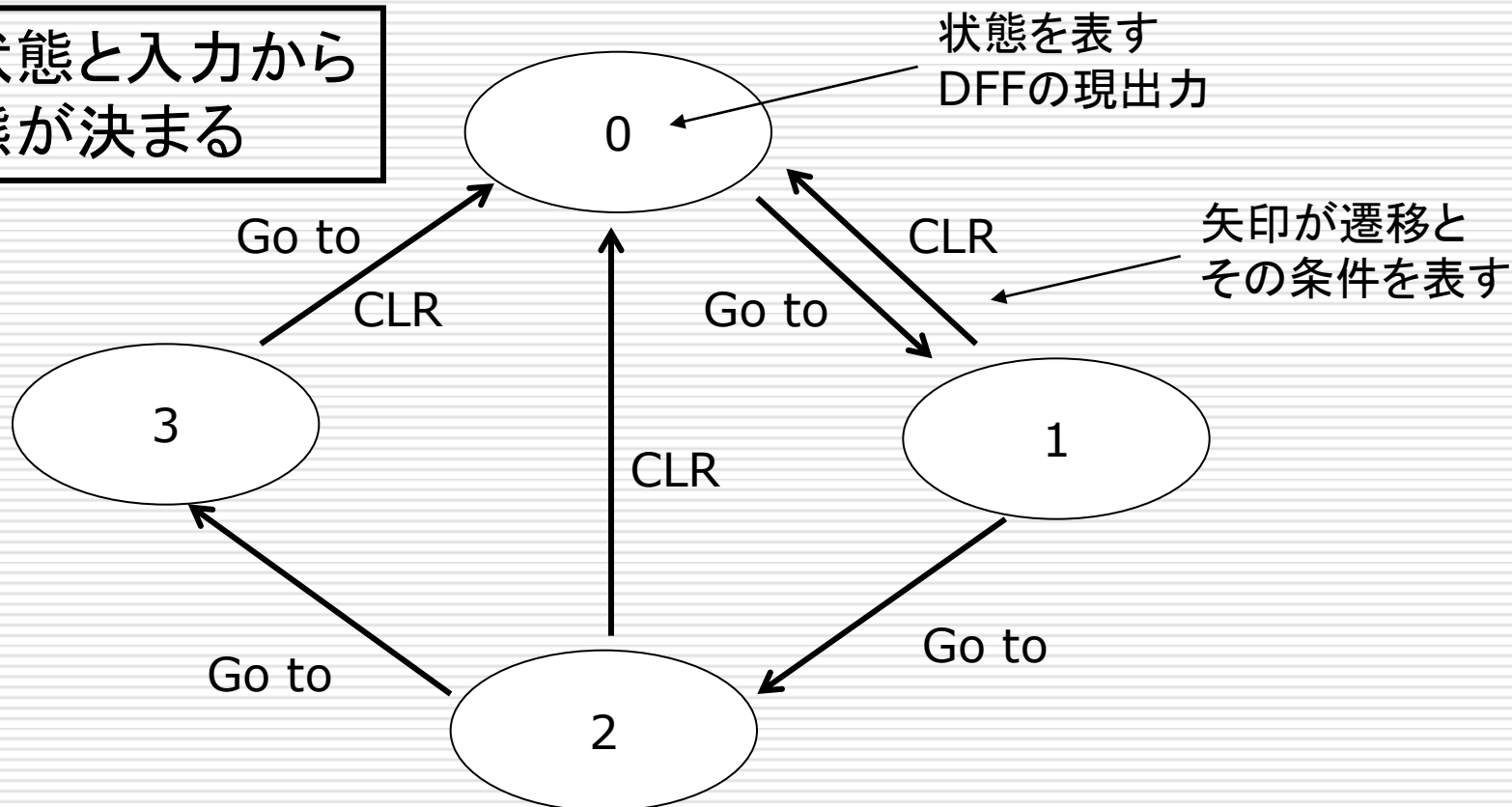
現在の状態(出力)に+1した値を次の状態にする



これを分かりやすく図にした物が状態遷移図

状態遷移図

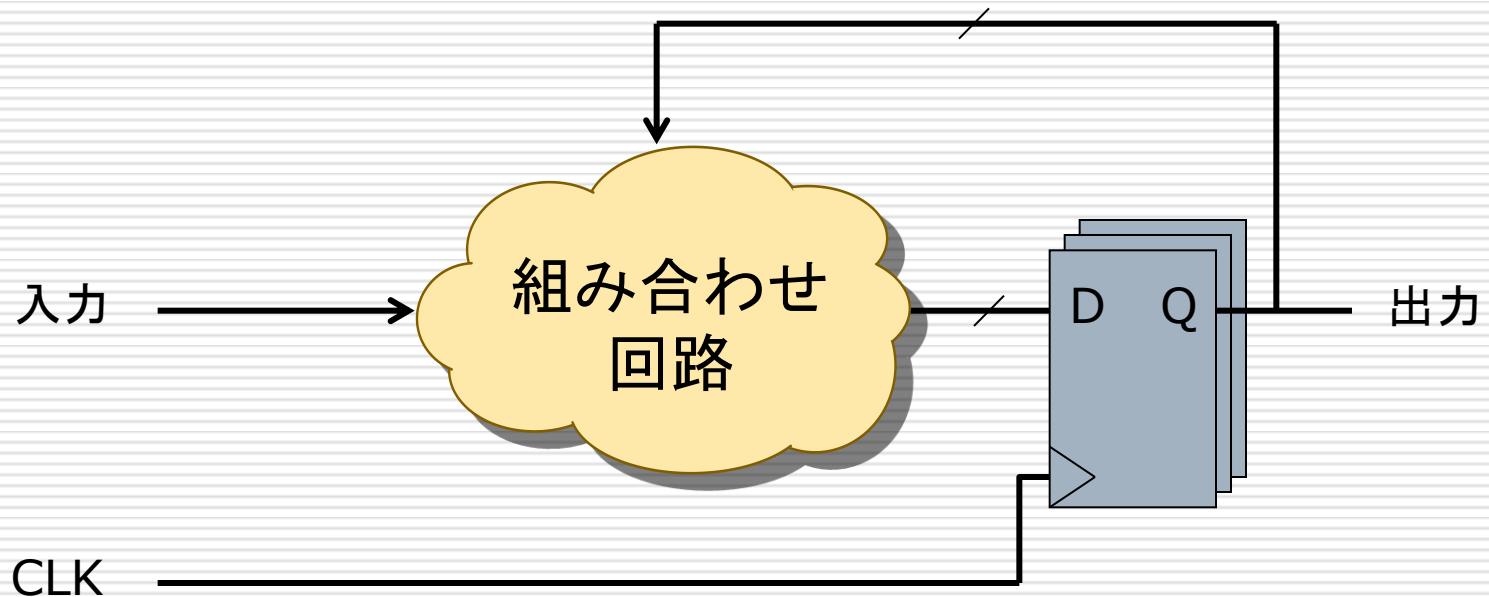
現在の状態と入力から
次の状態が決まる



これは便利なツールなのですが、ここでは書き方だけ覚えてもらえれば良いです

ステートマシン

状態遷移図を回路にすると……



状態遷移図が書ければ回路にする事ができる

FPGAと同期回路

- 現在の設計は同期回路が一般的です
- FPGAは同期回路で設計する事を前提に設計されています
 - 同期回路を上手に設計する事でFPGAの性能を十分に発揮させることが出来る
- 非同期回路設計は一般に難しいです
- 慣れるまでは同期回路で設計しましょう

まとめ： 同期回路

- 標準的設計手法
- 動作をクロック基準で考える
 - ゲート遅延を考慮せずに設計できる
 - 設計効率の向上、バグ発生率の低下
- FPGAは同期式回路設計を仮定して設計されている

FPGA

実験でデジタル回路を設計すると言えばFPGAを設計する、
と言っても良いくらい使われている技術

FPGAの構造

目的:FPGAの動作原理を理解する

回路なのに何故書き換え可能なのか？

Field Programmable Gate Array (FPGA)とは

□ 柔軟かいハードウェア

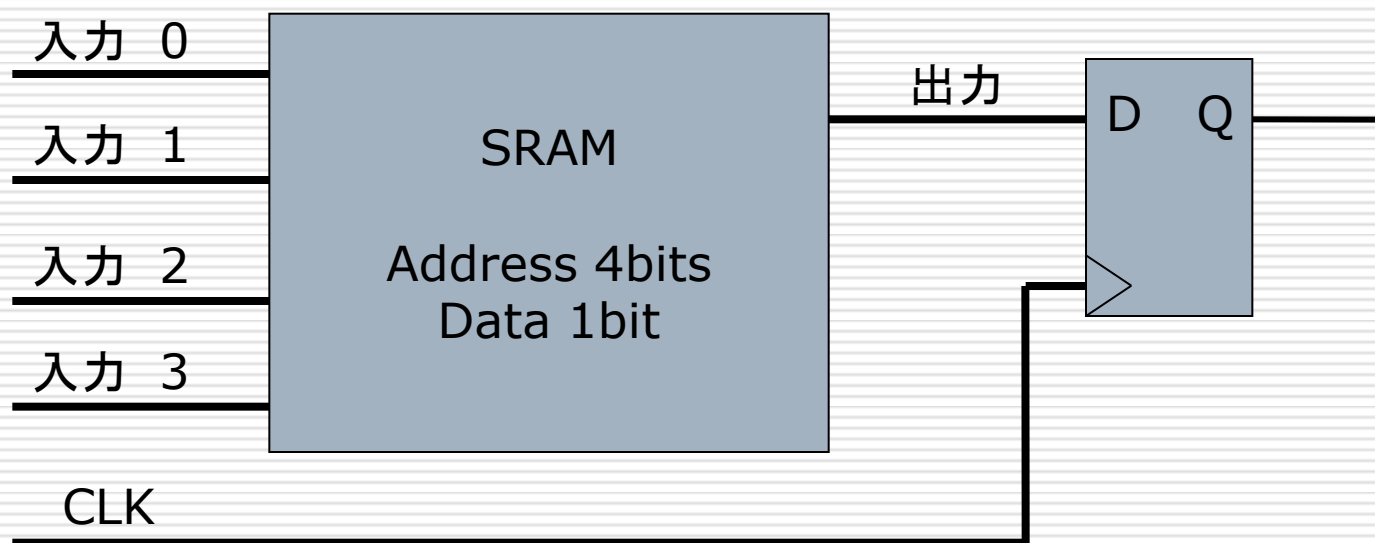
- 主としてデジタル(論理)回路
- 最近はアナログ回路もできるようになっている

□ 論理回路

- ソフトウェアの様にダウンロードして使用する
- 通常回路情報を格納するROMが必要
 - 電源投入時に自動的に動作開始させる為
 - 毎回手動でダウンロードする時は必要ない

動作原理

基本はSRAM+DFF



RAMなので入力の全ての組み合わせの出力をテーブルを持たせる事ができる。

4入力ANDの時

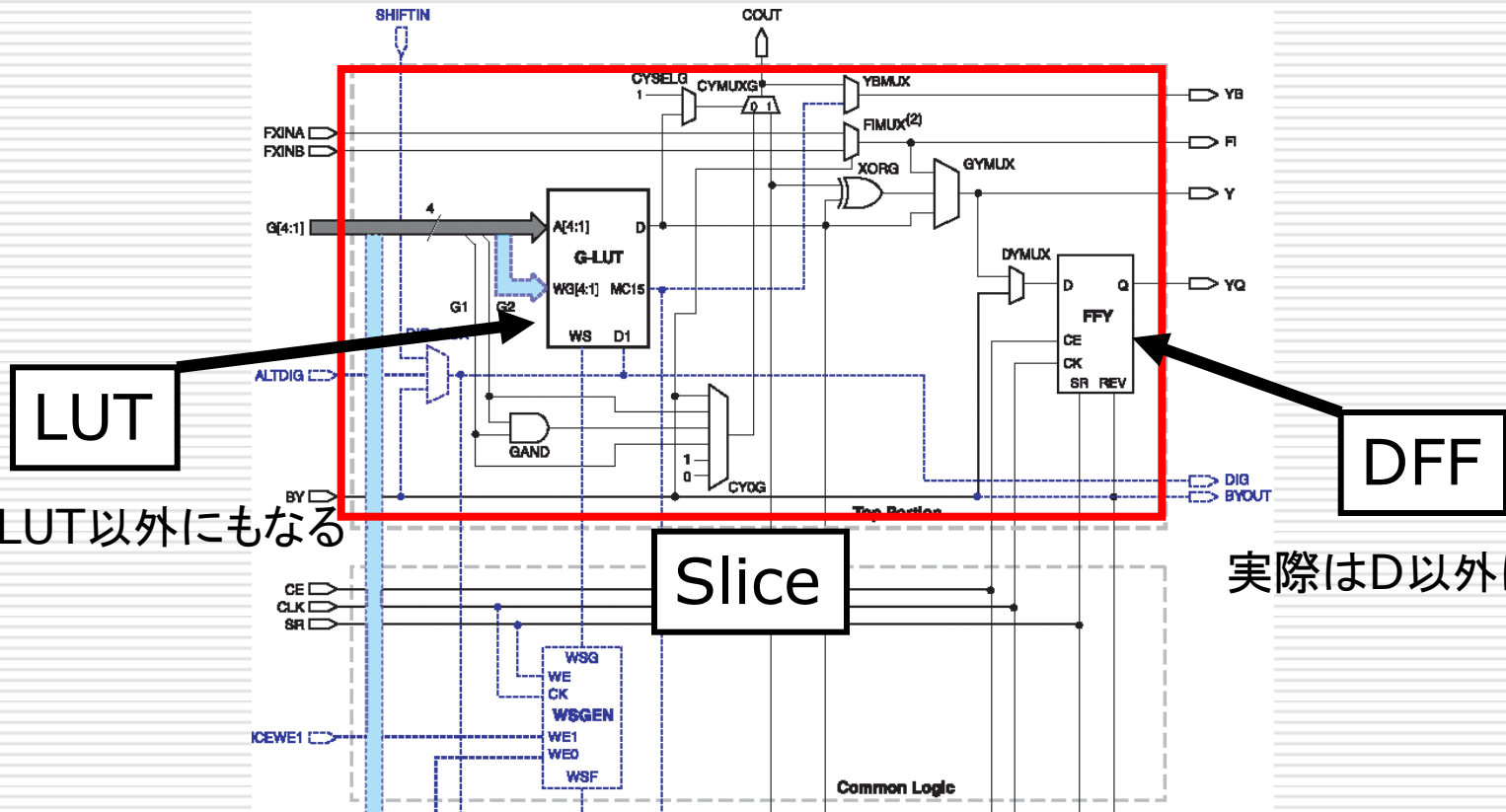
アドレス	データ	アドレス	データ
0	0	8	0
1	0	9	0
2	0	A	0
3	0	B	0
4	0	C	0
5	0	D	0
6	0	E	0
7	0	F	1

この様なデータをSRAMに書いておけばよい

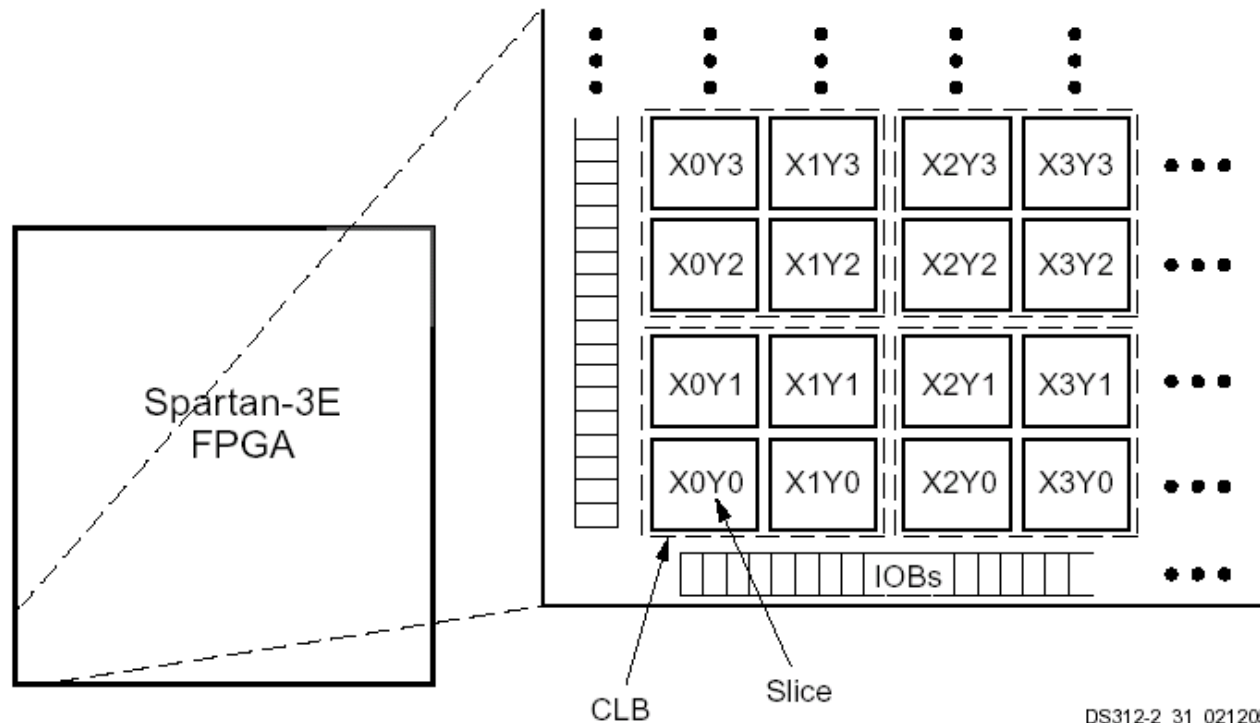
Slice

- SRAMはLUT (Look Up Table)と言います
- この組み合わせ + aをメーカー毎に名前を付けています。
 - XilinxではSliceと言います
 - Sliceが4つでCLBと言います
- データをダウンロードするとは
 - このLUTのデータを書き込むことです。

SLICE



CLBとSLICE



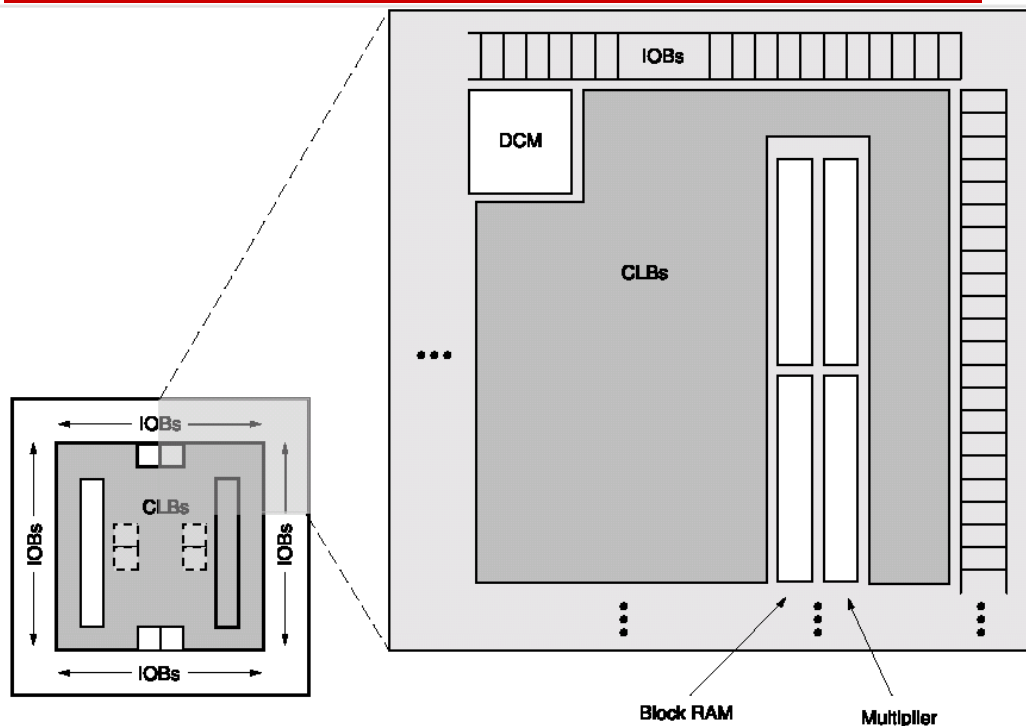
DS312-2_31_021205

Figure 11: CLB Locations

FPGA全体の構成

FPGAの構成ブロック

- DCM
- CLB
- IOB
- Block RAM
- Multiplier



Notes:

1. The XC3S1200E and XC3S1600E have two additional DCMs on both the left and right sides as indicated by the dashed lines. The XC3S100E has only one DCM at the top and one at the bottom.

Figure 1: Spartan-3E Family Architecture

データシートを参照してください

ここでは複雑になるので、これ以上説明しません。

まとめ：FPGA

- LUT+DFFの集合体
- 回路を書き込むとは
 - LUTデータを設定すること
- SRAMベースのFPGA
 - 電源を切るとデータが消える
 - 電源投入後に外部からデータをダウンロードする必要あり
 - JTAG
 - 外部ROM
- 不揮発性のフラッシュROMベースのFPGAもある

SRAMベースのFPGAが良く使われます

デジタル回路開発の流れ

目的: 回路設計とは何をするのかを知る

この業界でデジタル回路設計と言えば

- 既存デバイスの応用とその制御回路の設計
- ボード設計+FPGA回路設計が多い
 - ボード設計: デバイスを使う方法
 - FPGA設計: 論理回路
- アナデジ混載ASICは珍しい

必要なデバイスを選択して、それらを外部I/Fに接続できるように

仲介回路を設計する事

回路の勉強をするより、具体的な装置を設計する方が効率が良い

開発手順

1. 実験から入出力(I/F)が決まる
2. I/F処理デバイスの選定
 - 信号をデジタルへ、またはデジタルから、変換するデバイス
3. FPGA選定
 - サイズ、動作スピード見積り
4. ボード開発
5. FPGA回路開発
6. デバック

このセクションのまとめ

デジタル回路設計でやる事は・・・

- フロントエンドとバックエンド間の仲介回路開発
 - I/Fデバイスの選定
 - ボード開発
 - I/Fデバイスの制御回路の開発

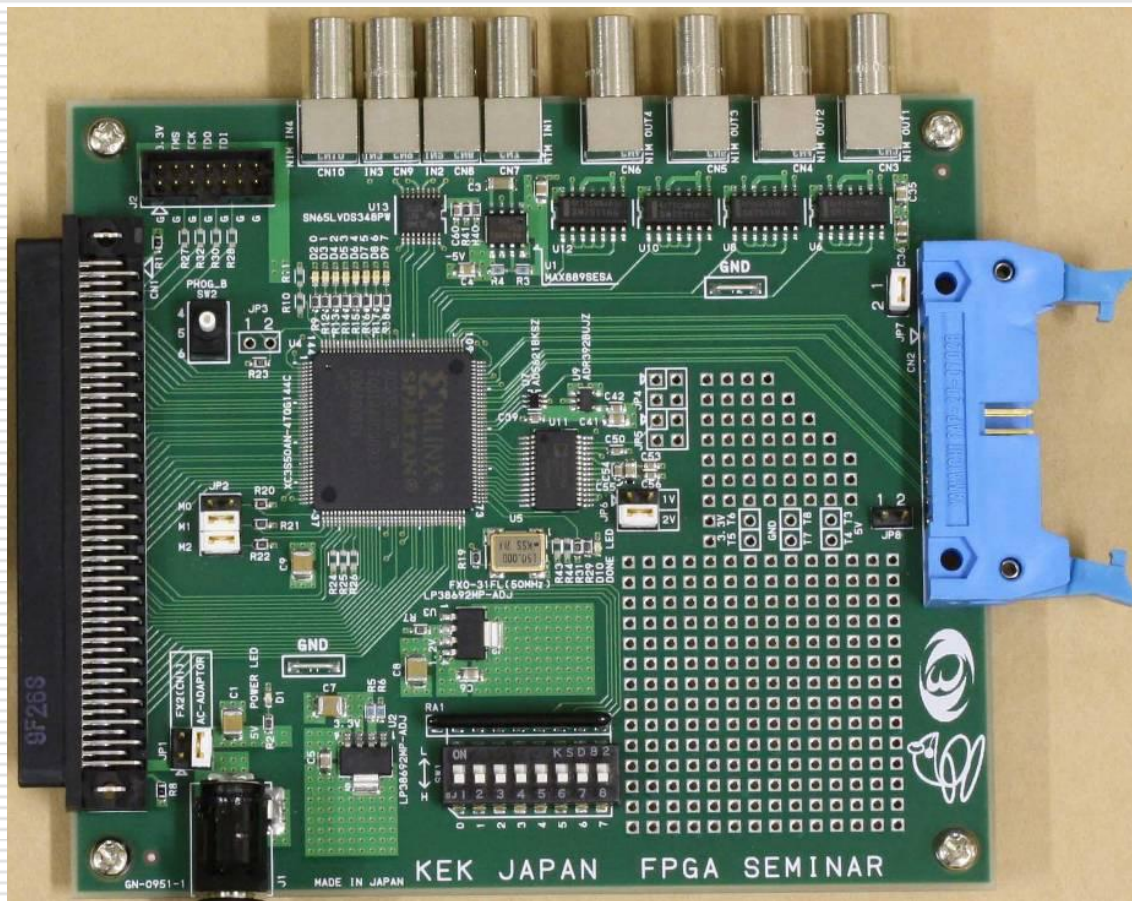
今後の学習方法(まとめに代えて)

- 本を読んでも設計できるようにはならない

- FPGA等を使って実際の回路を設計する
 - 例えば、NIM入力のスケーラ、コインシデンス回路、テスト実験などで使用する回路

しかし、ボードを開発する余力はない、という事なので
要望にこたえNIM入出力できるFPGAボードを開発しました！

FPGAセミナー用に開発したボード



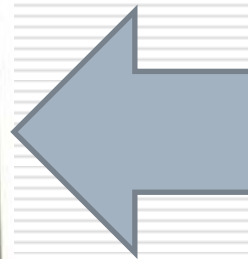
NIM 4 IN+4 OUT
FPGA: Spartan3AN50
ADC 20Msps
DAC
LED
DIPSW

Xilinx社Spartan3E/A/AN
スターターキットと接続できる

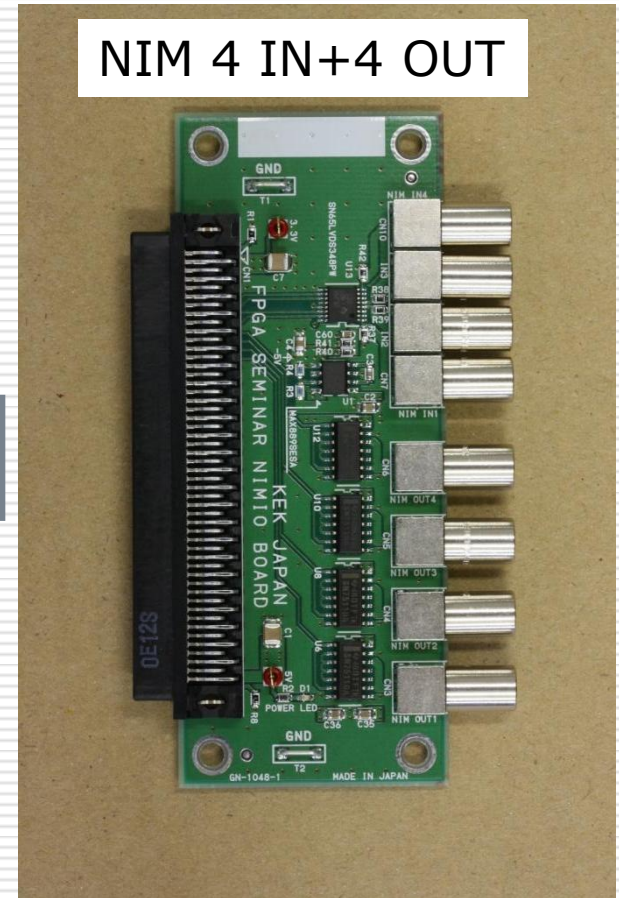
<http://gn-d.com/>
GN-0951-1 FPGA 講習会用基板

半田付け実習用に開発したボード

Xilinx社Spartan3E/A/AN
スターターキットと接続できる



NIM 4 IN+4 OUT



<http://gn-d.com/> から GN-1048-1について問い合わせ



終了です

参考文献

- コンピュータとデジタル回路を学びたい方
 - パターソン、ヘネシー、“コンピュータの構成と設計(上・下)” 日経BP社
- Verilog-HDLを学びたい方
 - 小林優、“入門Verilog記述” CQ出版社
- ASIC設計を目指す方
 - 國枝博昭、“集積回路設計入門” コロナ社
- 雑誌
 - 月刊Design Wave、CQ出版社
- 無料セミナー
 - FPGA販売代理店が主催している