同期回路とクロック

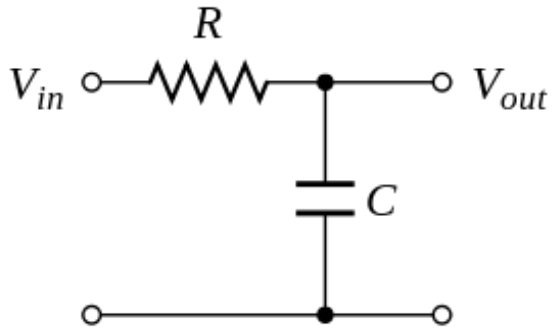
蜂谷　崇
奈良女子大　＆　理研BNL

アナログ回路(Analog)とデジタル回路(Digital)

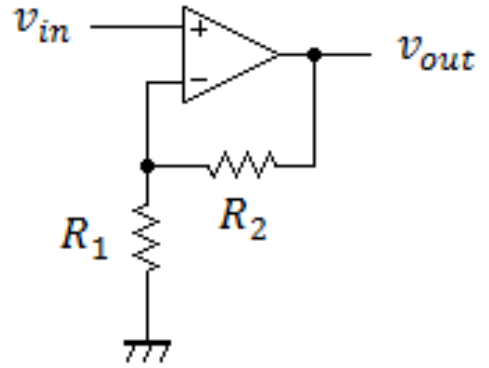
- アナログ回路:
  - 電圧Vの値に意味がある回路。
  - 例：
    - RLC回路(passive)：フィルタ回路
    - トランジスタ、オペアンプ(active)
      - 増幅器など

- デジタル回路
  - 2値(H/L)扱う、主に論理回路
    - 電圧値で規格がある。
      - シングル：TTL(0 －5V)、LVCMOS(0－3.3V), 2.5V, 1.8V, 1.2V
      - 差動：　LVDS(+-0.2V + 1.2V), ECL(3.4-4.2V)
  - 論理回路
    - AND、OR、NOT、XOR、
      - 論理素子のことをゲートと呼ぶ
    - ADDR、FF
      - ゲートの組み合わせ

ローパスフィルタ(passive)

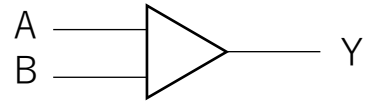


非反転増幅器(active)

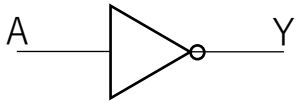


論理回路

A	B	AND Y	OR Y	XOR Y
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

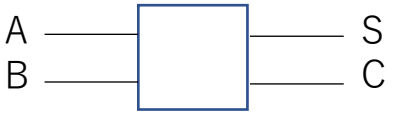


A	NOT Y
0	1
1	0



ADDR(加算器)

A	B	S(和)	C桁上
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



フリップフロップ：FF

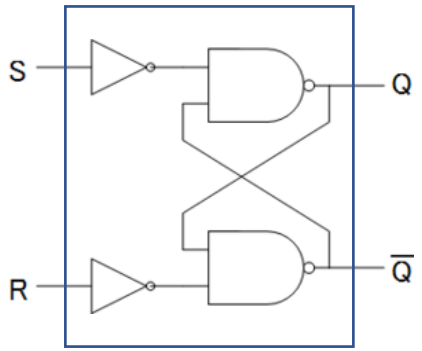
- フリップフロップ：　FlipFlop
  - 1ビットの情報を"0"または"1"の状態として保持することができる論理回路

- RS –FF (Reset-Set)
  - Set=1 でQ=1, Reset＝ 1 でQ=0

- D-FF
  - Cの立上りエッジでDの値を取り込んでQに出力する。

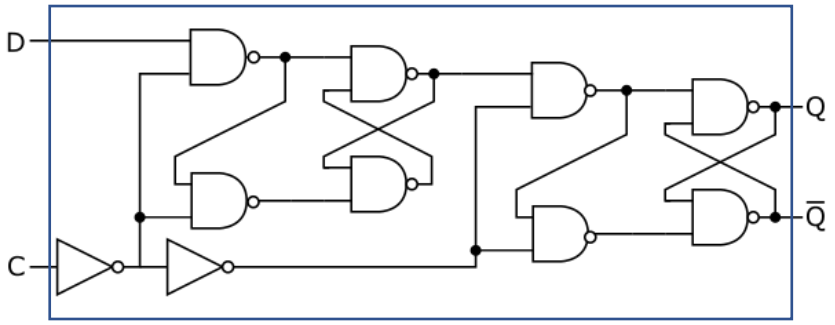
D－FFにより回路の動作タイミングを制御できる
- Cのエッジが基準
  - 非同期(これまで)とはちがう。
    - 入力値が変わると出力が変わる。
- 実行順序を制御(シーケンス)できるようになる。
  - 1→2→3→4

RS –FF



入力		出力
S	R	Q
0	0	保持
0	1	0
1	0	1
1	1	禁止

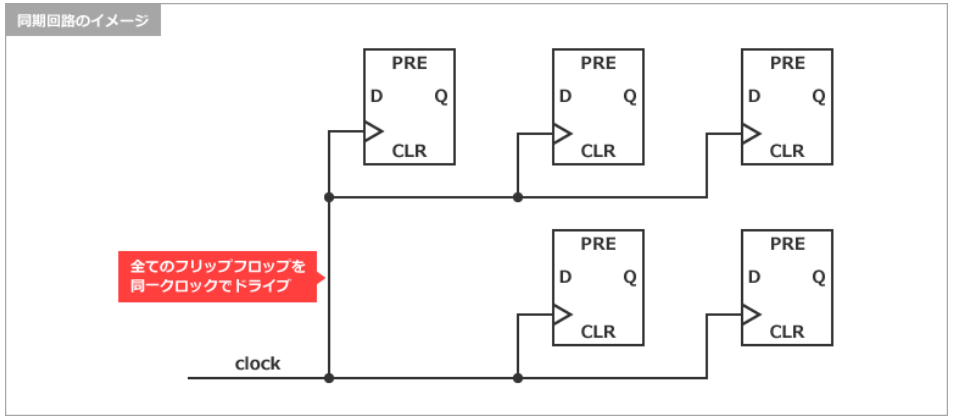
D-FF



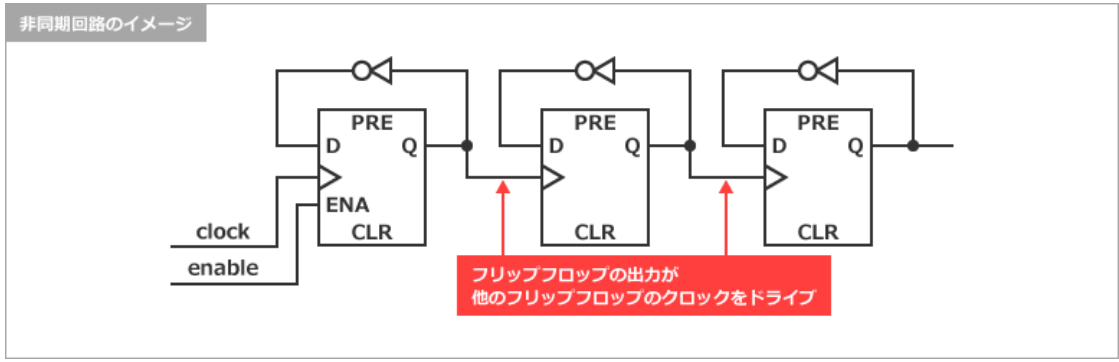
D	C	次のステートのQ
0	↑	0
1	↑	1
X	↓	前ステートのQを保持

同期回路と非同期回路

- 同期回路
  - 『同一クロックの同一エッジに同期して動作する回路系』を言います。
  - D-FFをつなげた回路など



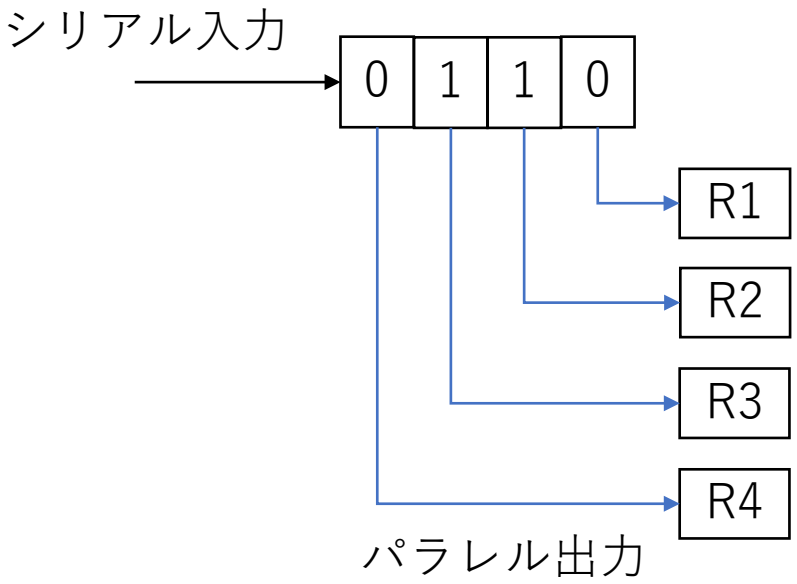
- 非同期回路
  - クロックを使わない。
  - 同一クロックでも、立上り・立下りが違うと、非同期という扱いになる。

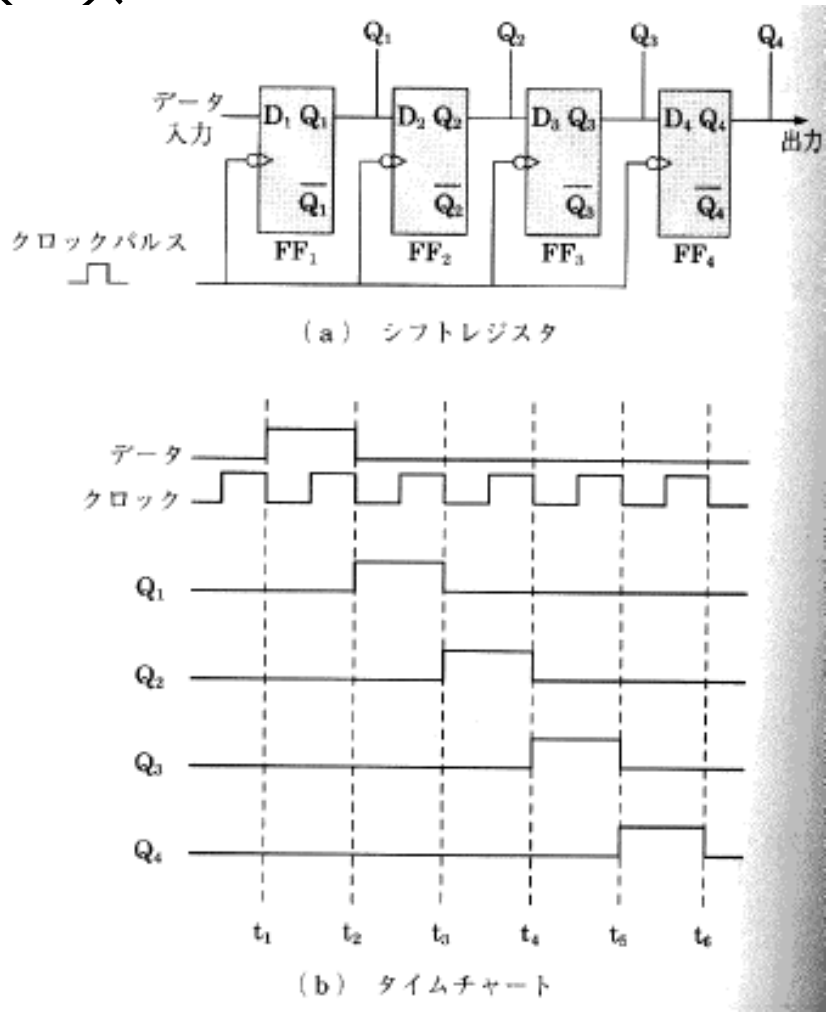


同期回路の例：シフトレジスタ

- 入力データを1クロックごとに次のDFFに送る。

- パラレル・シリアル変換
  - データ通信　：　1ビット入力→4ビット出力に変える
    - ハードウェア制御(リレーとか)



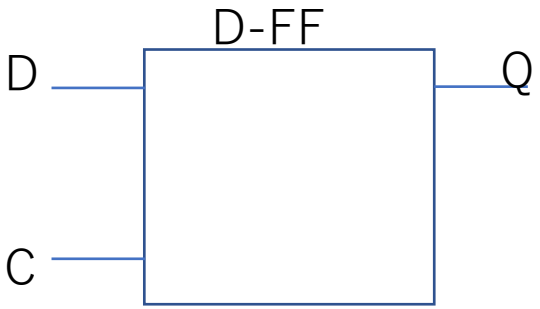


（a）　シフトレジスタ

（b）　タイムチャート

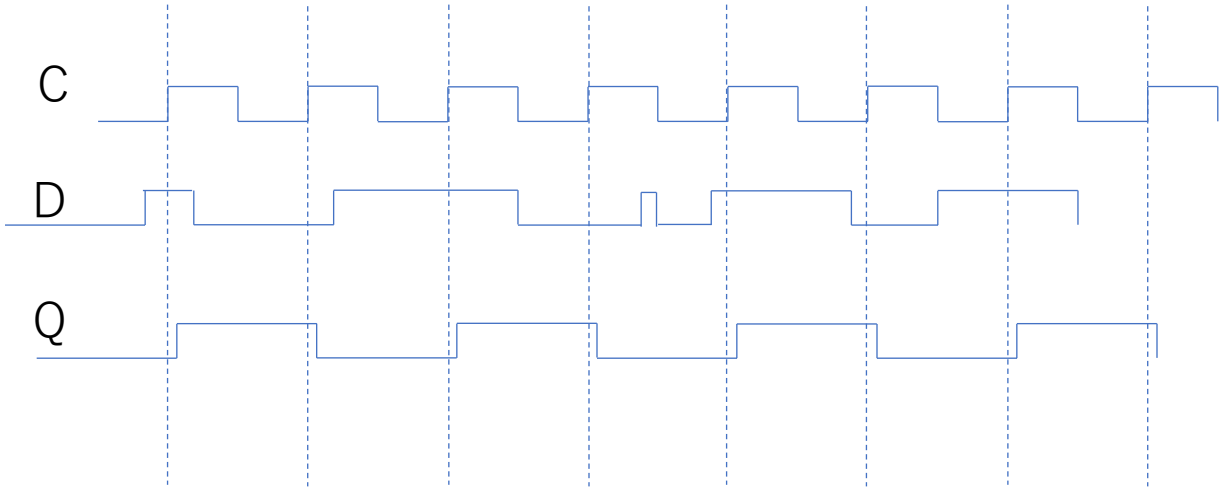
FFのデータ取り込みタイミング

- Cのエッジ(↑か↓)時のDを取り込みQに出力する。
  - 取り込み→出力に時間がかかる。
- 特徴
  - DはCに同期していなくてもよい
    - 非同期→同期回路変換
  - Cのエッジ以外でDが変化しても無視する。
    - データの整形
- タイミングで注意すること
  - CLKのエッジでデータが変化しないようにする。



VHDLコード

```
D : in
clk : in
Q : out

process(clk)
begin
   if(rising_edge(clk)) then
      Q <= D;
   endif;
end;
```

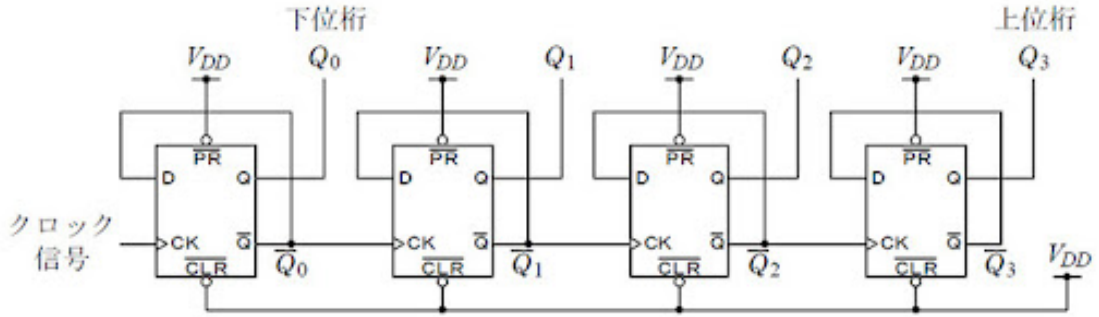


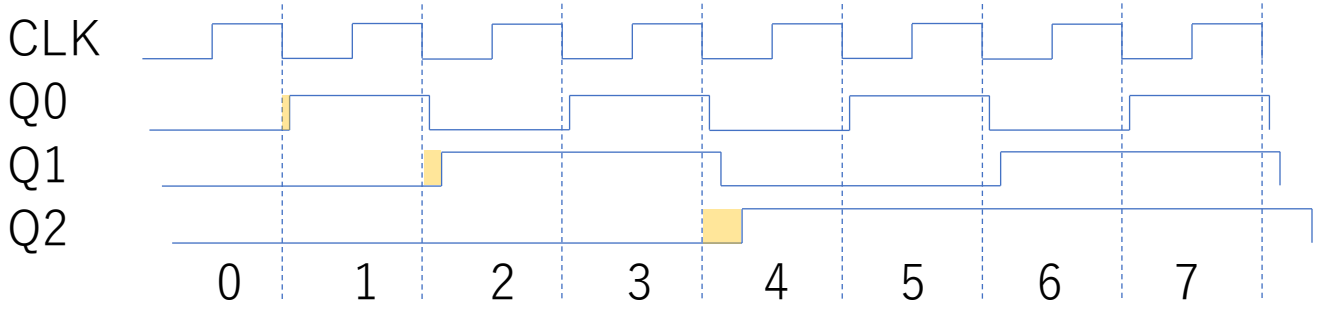
カウンタの例

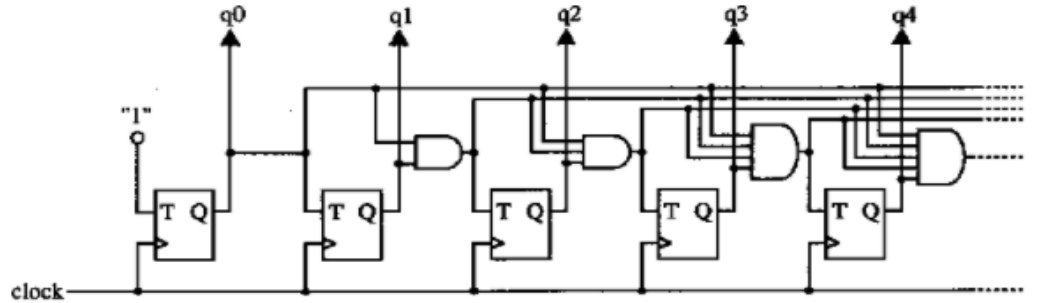
- カウンタ
  - クロックごとに0→1→2→ 3 →　と変わっていく。

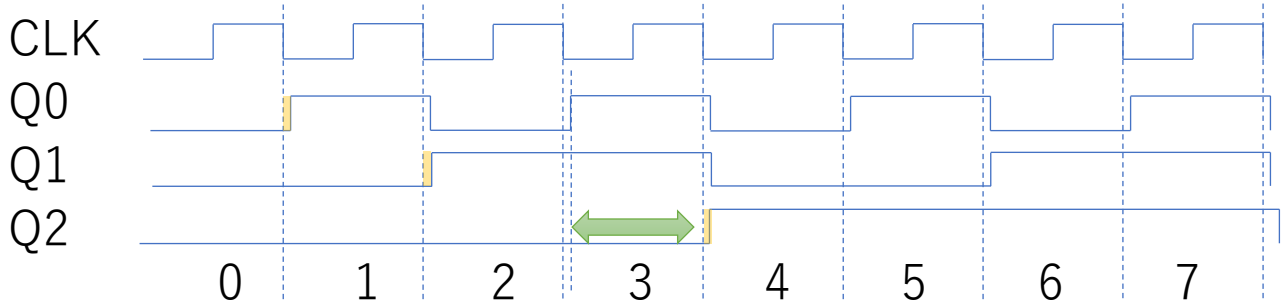
- 非同期式カウンタ
  - 直前のビットの出力をトリガー(CLKに入れて)にしてカウントしていく。
  - 高位ビットほどタイミングが遅れていく。
    - 非同期だからデータが壊れることはない。

- 同期式カウンタ
  - 高位ビットほどタイミングが遅れていく。
    - ゲートが多段になるため。
  - 遅れすぎるとデータが壊れる
    - 次のCLKに間に合わなくなる



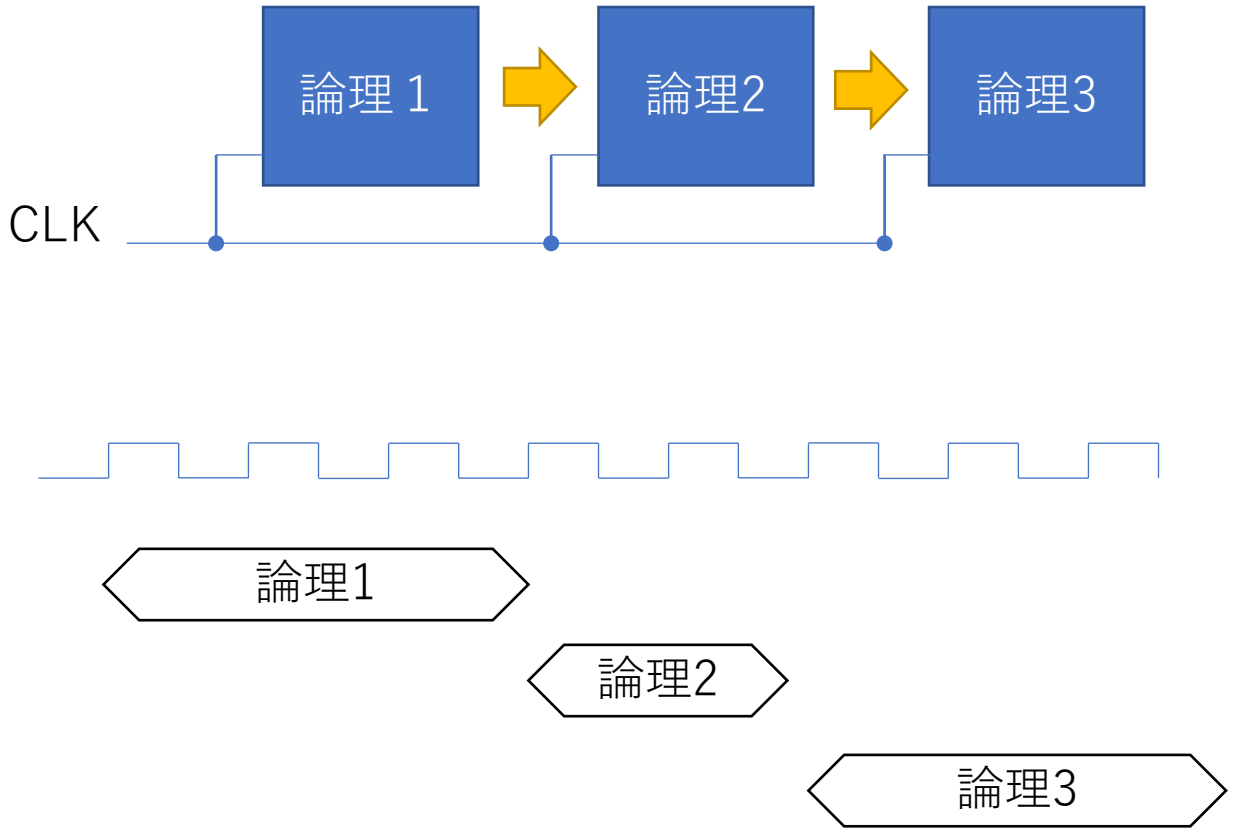






論理回路の構成

- 同期回路として作る。
- 例：　DAC設定
  - 論理１：　設定値受信
  - 論理２：　データ変換
  - 論理３：　DAC設定

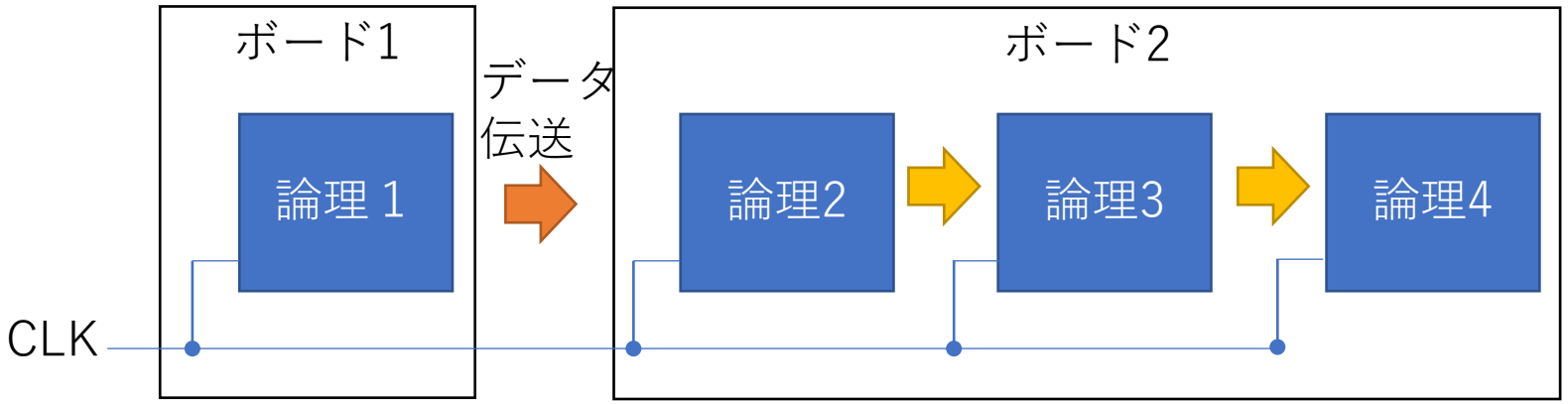


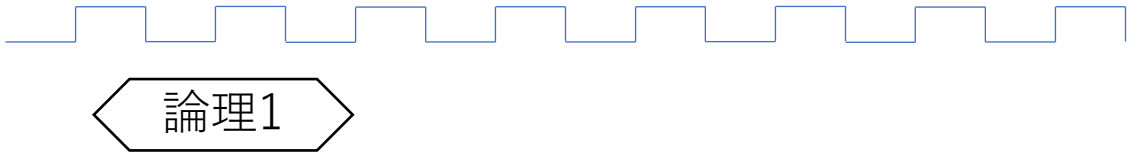
タイミング制御にCLKを使うので**CLKが重要！！**

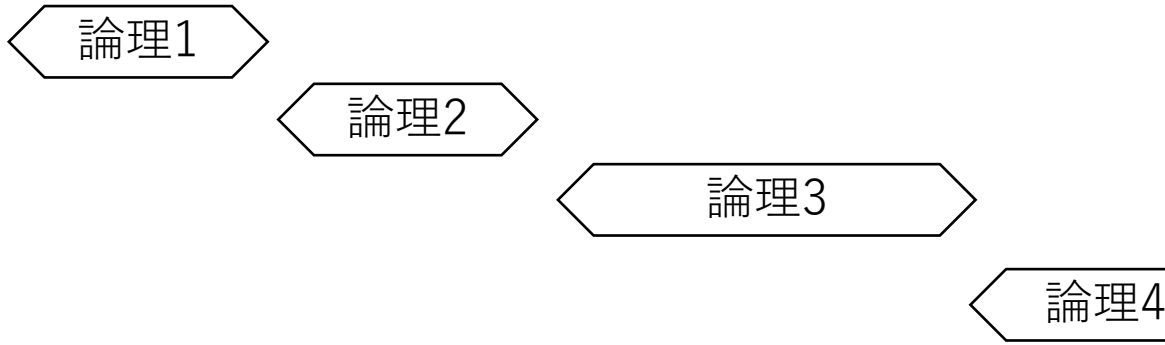
- 遅延があるとずれる。
  - 配線長をそろえる必要がある。：　遅延制約を設定
  - FPGA内のCLK配線は遅延の少ない特別なラインを使う
- ジッターがあるとタイミングがずれる。

論理回路の構成2：　異なるボード間で連携

- 同期回路として作る。
- 例：　DAC設定
  - 論理１：　設定値送信
  - 論理２：　設定値受信
  - 論理３：　データ変換
  - 論理４：　DAC設定







ボード間でCLKを共有することで回路を同期する。

- データ伝送
  - ボード間のデータの受渡し
    - 論理間の受渡と論理的には同じ。
  - 方式にはいろいろある。
    - 伝送方式：　パラレル、シリアル
    - ハードウェア(配線)の選択：イーサネット、同軸ケーブル、WIFI
    - 　　：　シングル、差動
    - シンクロナス：クロック伝送も同時か？、SDRAMのS
    - シリアル：UART, I2C、SPI、パラレル：PCI

PLL：(ボード間通信で)CLKを安定化する

- ボード間の長い配線を通すとCLKが歪む。
  - 同期が崩れる原因になる。
  - **CLKを安定化する。**
    → **PLL (Phase Locked Loop)**

- PLL:　位相差を電圧に変換し、VCOの周波数を変える　→　周波数＋位相を保つ。
  - 追従の感度(時定数)を決めることができる。：入力CLKが揺れる場合でもCLKを安定化できる。

- PLLのその他の機能
  - 入力周波数に対して、分周、逓倍が可能：　周波数が変えられる。
    - INTT: BCO(10MHz) -> 200MHz (20倍)
  - FPGA内部では、PLL, DLL, DCMなどいろいろな名前で呼ばれる

■ 1. PLL の基本構成 と 動作原理

PLL とは、**Phase Locked Loop** の略語であり、周波数負帰還回路 を構成する。
図-1 には その 基本構成図 を示す。

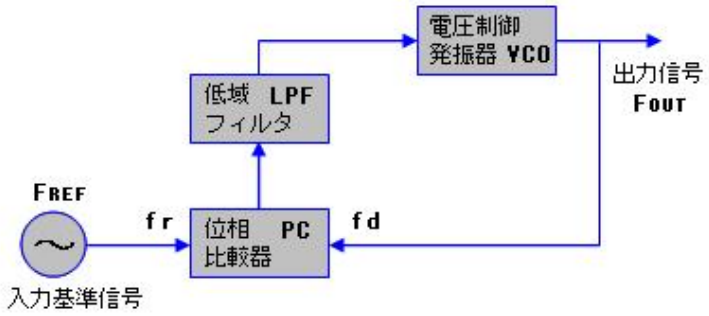


図-1. PLL の基本構成図

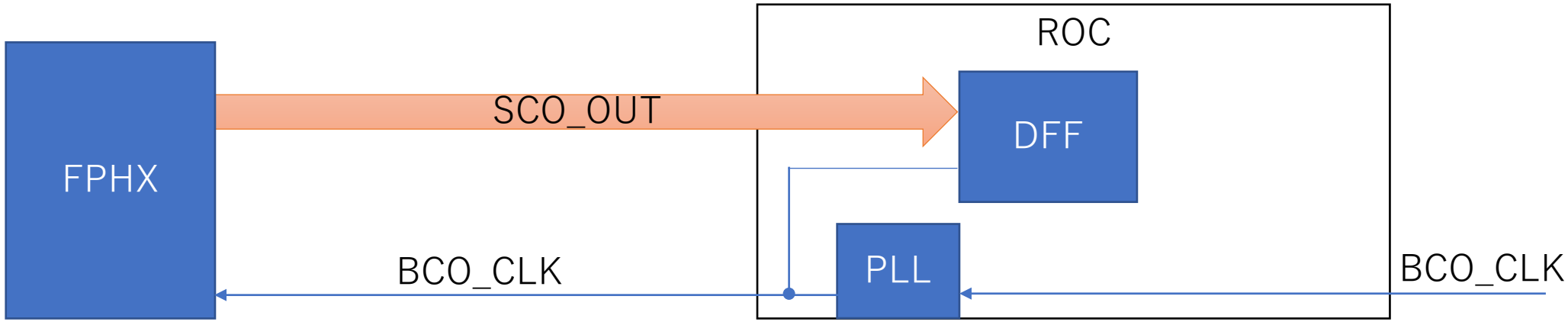
PLL は、位相比較器（**PC**：Phase Compalator），ループフィルタ（または ローパスフィルタ **LPF**），電圧制御発振器（**VCO**：Voltage Contorolled Oscillator）及び 入力基準信号 **FREF** から構成される。

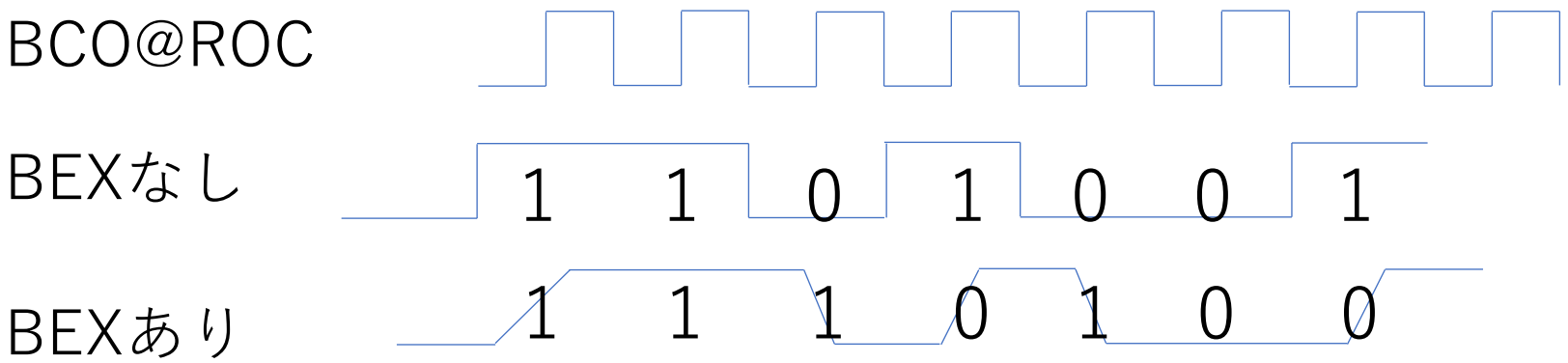
ここで、PLL の基本動作を 文書 で 説明すると 次のようになる。

1. PLL が ロックしている状態 **fr** = **fd** から 入力基準信号より VCO の出力周波数が 高くなる。
   ↓
2. 位相比較器 **PC** の出力に 誤差信号パルス が 発生する。
   ↓
3. 低域フィルタ **LPF** を通過することによって 直流電圧となる。
   ↓
4. この直流電圧は、誤差信号に比例し VCO の出力周波数が 低くなる値となる。
   ↓
5. VCO の出力周波数が 下がり、**fr** = **fd** の状態に戻る。

このように PLL は、常に **fr** = **fd** の状態を 保つように働いてくれる 優れものである。

FPHXのReadbackerが動かない原因

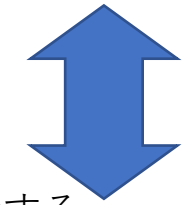




- 20ns程度遅れる
- 最初のスロープが緩やか　(Z -> ロジック　に切り替わる)

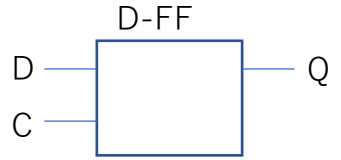
順序回路：ハードウェアのソフトウェア化

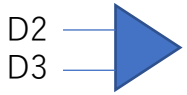
- ハードウェア：
  - 入力に応じて、すぐに動作する。
    - 瞬間的に動く。
    - 全ての回路が同時に動く



全く異なる動作仕様

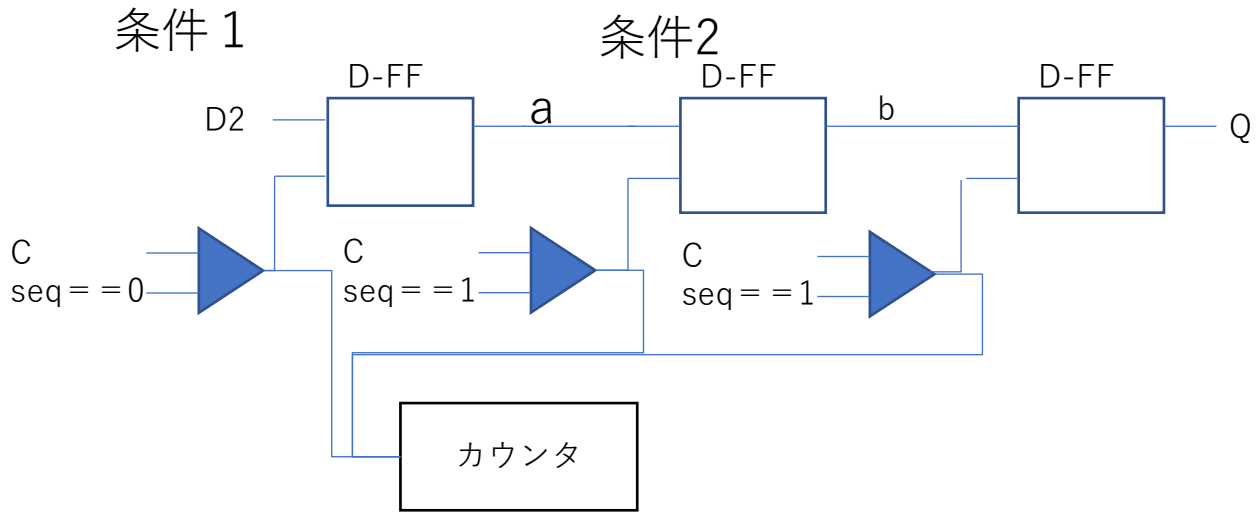
- ソフトウェア
  - 上から順番に実行する。





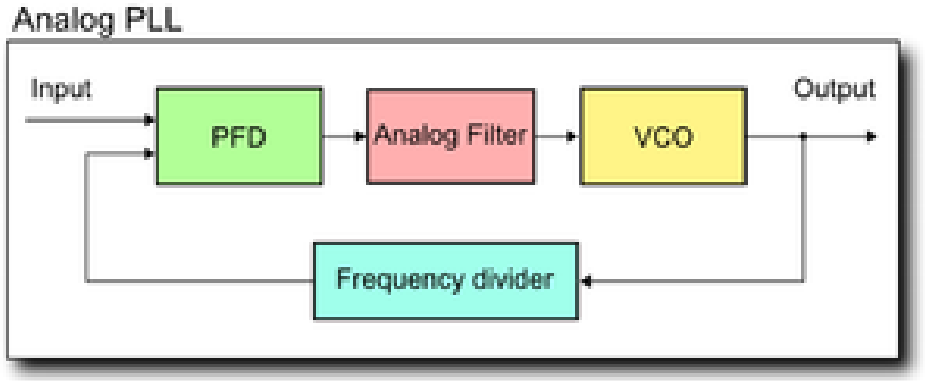

```
TCanvas *c1 = new TCanvas("c1", "c1", 1000, 800);
c1->Divide(2,2);
c1->cd(1);
h2->Draw();

c1->cd(2);
h2->Draw("colz");

c1->cd(3);
h2->Draw("lego1");
```

- 順序回路：順番に実行する。　→　ハードウェアをソフト的に動かす




```
seq=0;
If(seq==0) {a = D2;   seq==1; }
If(seq==1) {b = a;    seq==2; }
```

Phase Locked Loop (位相同期回路)

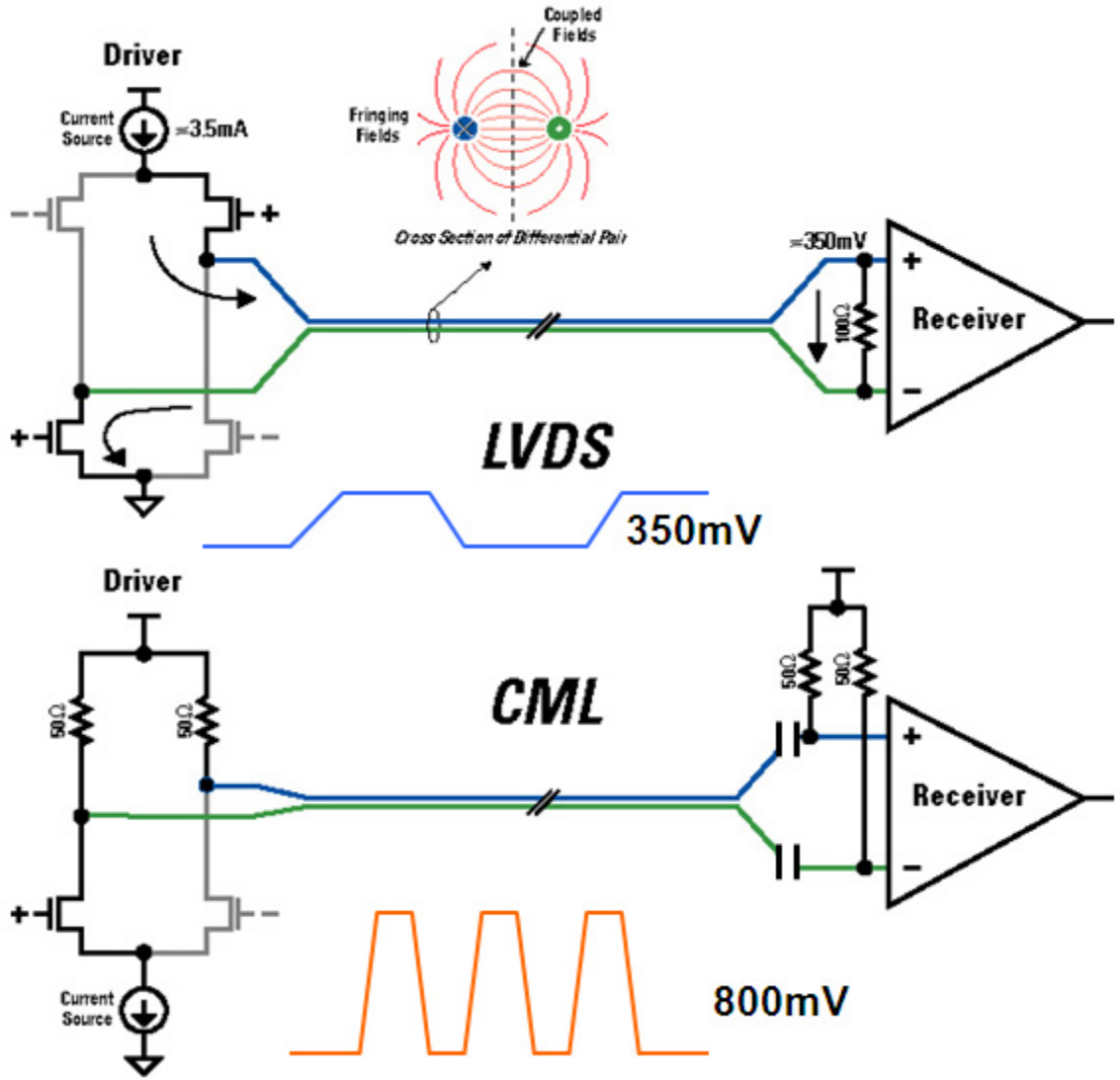


位相比較器
入力された2つの信号の位相差を電圧に変換し出力する回路である。アナログPLLではアナログ乗算器が良く使われ、デジタルPLLでは排他的論理和とチャージポンプなどから構成される。

ループ・フィルタ
帰還ループのフィルタとしてローパスフィルタを使用する。フィードバックを含む回路では短周期の信号変動が増幅されることで無用な発振が起きることがあり、アナログPLLとデジタルPLLではこれを避けるためにローパスフィルタによって不要な短周期の変動を遮断する。 入力周波数の

VCO (Voltage Controlled Oscilator)
入力された電圧によって出力周波数を制御することができる回路である。一般的にはバリキャップ（バラクタ、可変容量ダイオード）に入力電圧を加え、その静電容量の変化で発振周波数を制御するものが多い。

分周器 (Frequency divider)
分周器は入力された周波数を整数分の1にして出力する回路である。PLLに入力された基準となる信号の周波数を精確な倍率で高めて出力する。この分周する比率を外部制御によって可変にすることで出力する周波数を制御することができる。PLLとしての出力周波数を入力周波数より低くする場合には基準周波数となる入力信号を分周してから位相比較器に与えることで容易に実現出来る。FM復調器のように周波数を変更しない場合には分周器は必要ない。



- LVDS is TIA/EIA-644-A Standard
- LVDS operates in multiple configurations and up to 3Gbps depend on device
- No Standard for Current Mode Logic (CML)
- CML will operates up to 28 Gbps as I/O but limited to point-to-point applications

1. PLL の基本構成 と 動作原理

PLL とは、**Phase Locked Loop** の略語であり、周波数負帰還回路 を構成する。
図-1 には その 基本構成図 を示す。

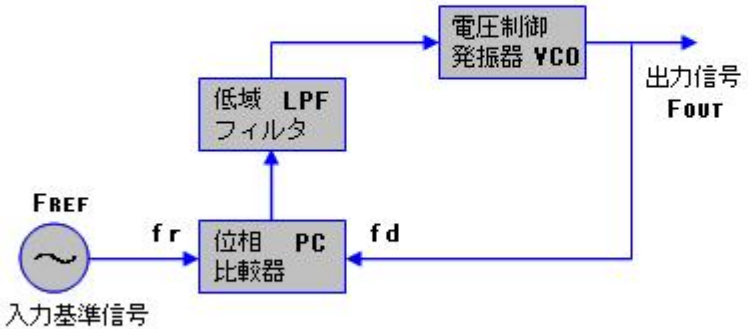


図-1. PLL の基本構成図

PLL は、位相比較器（ **PC**：Phase Compalator ），ループフィルタ（ または ローパスフィルタ **LPF** ），
電圧制御発振器（ **VCO**：Voltage Contorolled Oscillator ）及び 入力基準信号 **Fref** から構成される。

ここで、PLL の基本動作を 文書 で 説明すると 次のようになる。

1. PLL が ロックしている状態 **fr** = **fd** から 入力基準信号より VCO の出力周波数が 高くなる。
   ↓
2. 位相比較器 **PC** の出力に 誤差信号パルス が 発生する。
   ↓
3. 低域フィルタ **LPF** を通過することによって 直流電圧となる。
   ↓
4. この直流電圧は、誤差信号に比例し VCO の出力周波数が 低くなる値となる。
   ↓
5. VCO の出力周波数が 下がり、**fr** = **fd** の状態に戻る。

このように PLL は、常に **fr** = **fd** の状態を 保つように働いてくれる 優れものである。