

# FPGAと計量器ソフトウェア認証

工学計測標準部門

データサイエンス研究グループ

松岡聡

# 内容

1. FPGAとは
2. FPGAを用いたシステム開発の概略
3. FPGAを用いた簡単な計量器のデモ
4. FPGAを計量器に用いた場合の計量器認証

# FPGA とは？

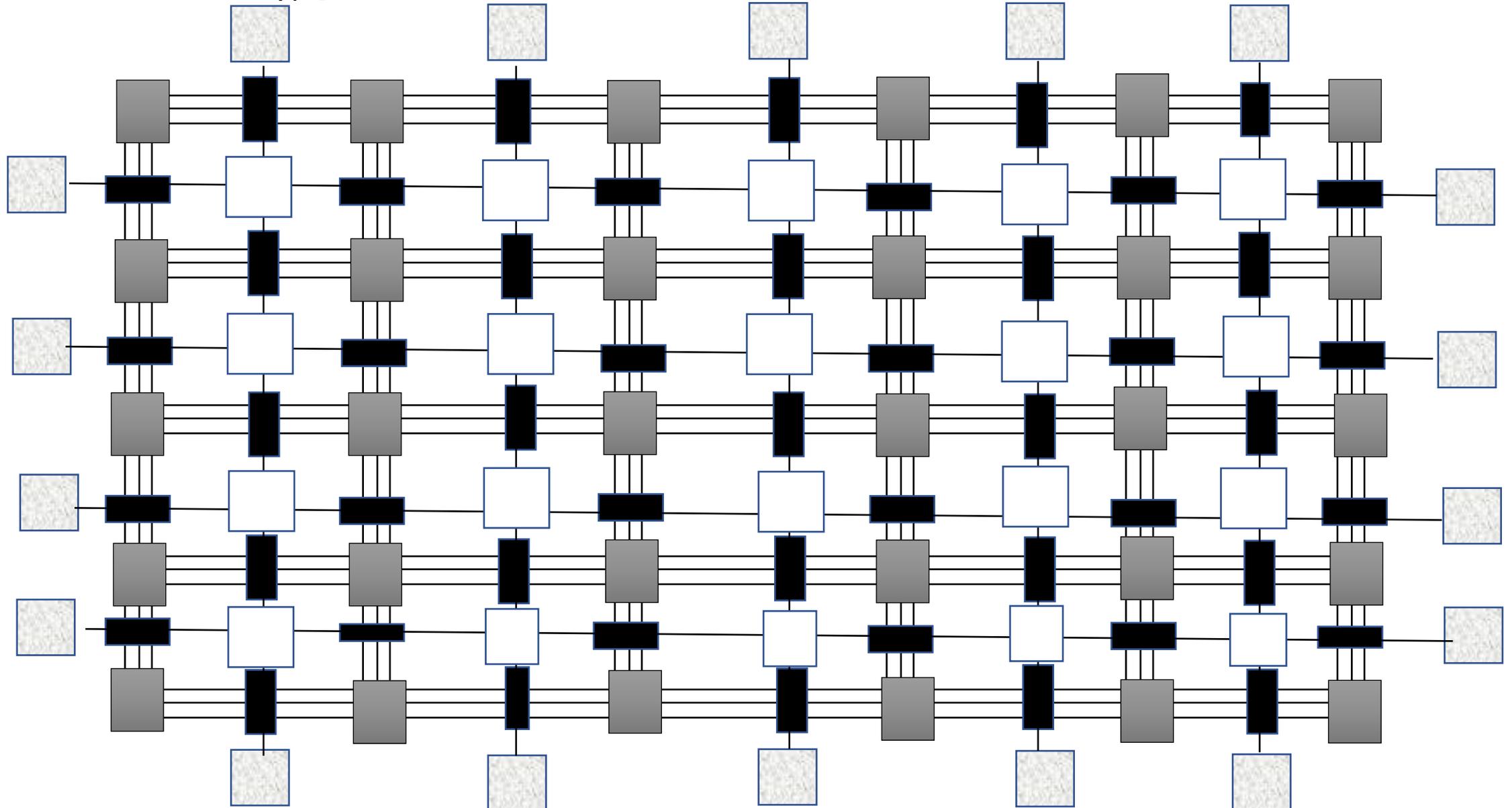
- Field Programmable Gate Array の略
- 大雑把に言えば、その場で書き換えられて設計変更ができる論理回路
- 産業界用としては、初期段階の製品や小規模生産の製品に組み込まれたりする

# FPGAの利点

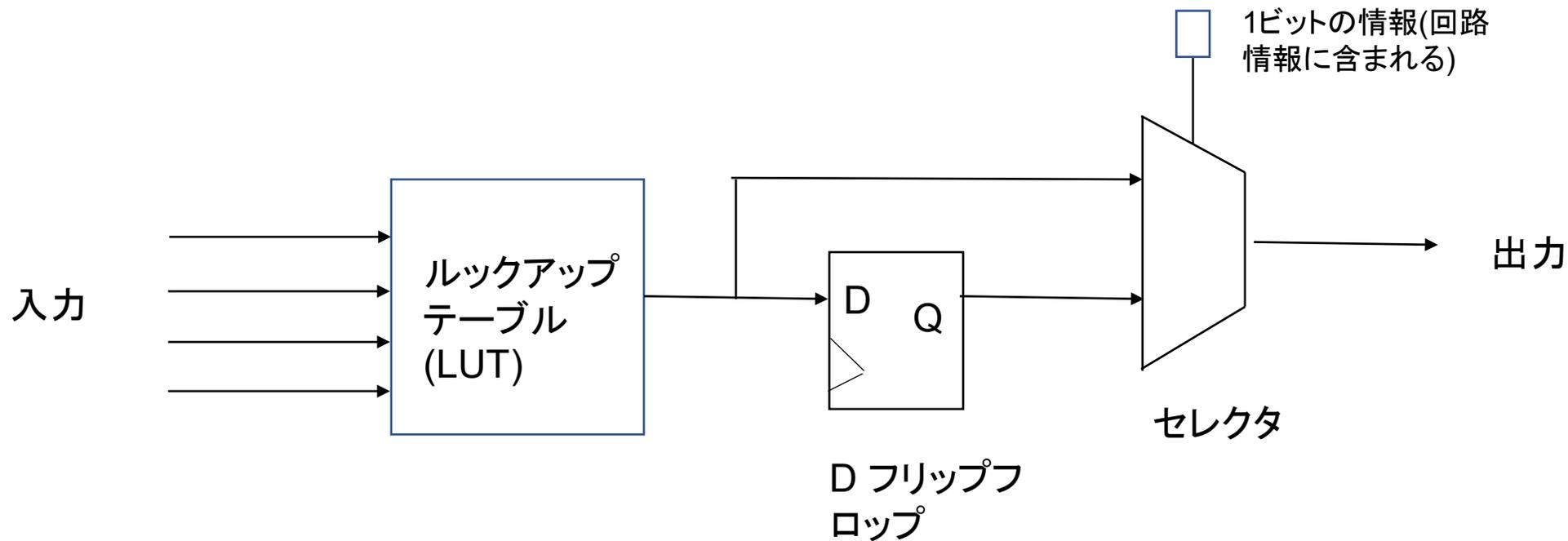
- バグを発見した際、書き換えて修正できる
- 少量生産ならば ASIC (Application Specific Integrated Circuit) よりもコスト面で有利 → **計量器**用途向き



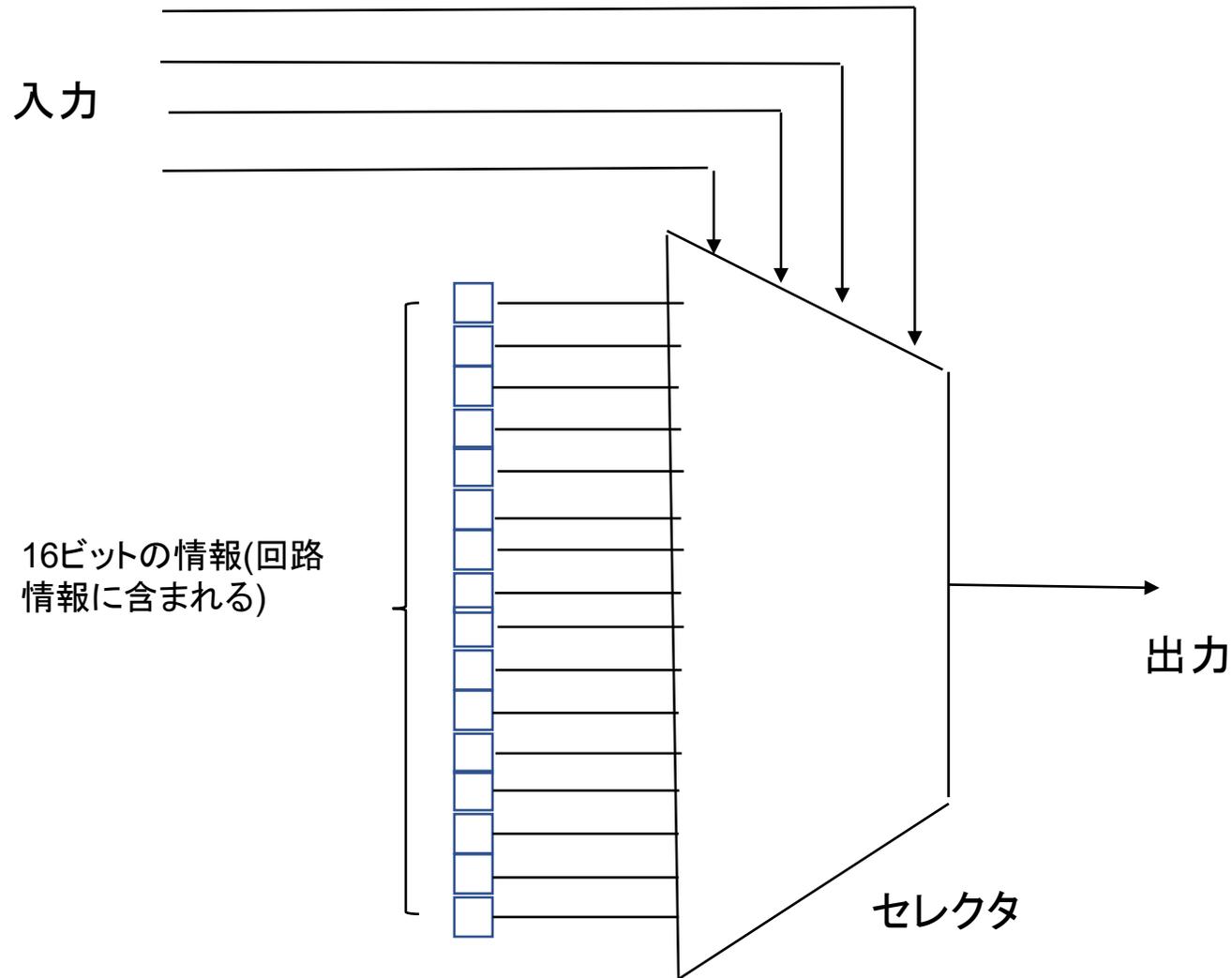
# FPGAの構造



# FPGAの構造 (Basic Logic Element, BE)



# FPGAの構造 (ルックアップテーブル, LUT)



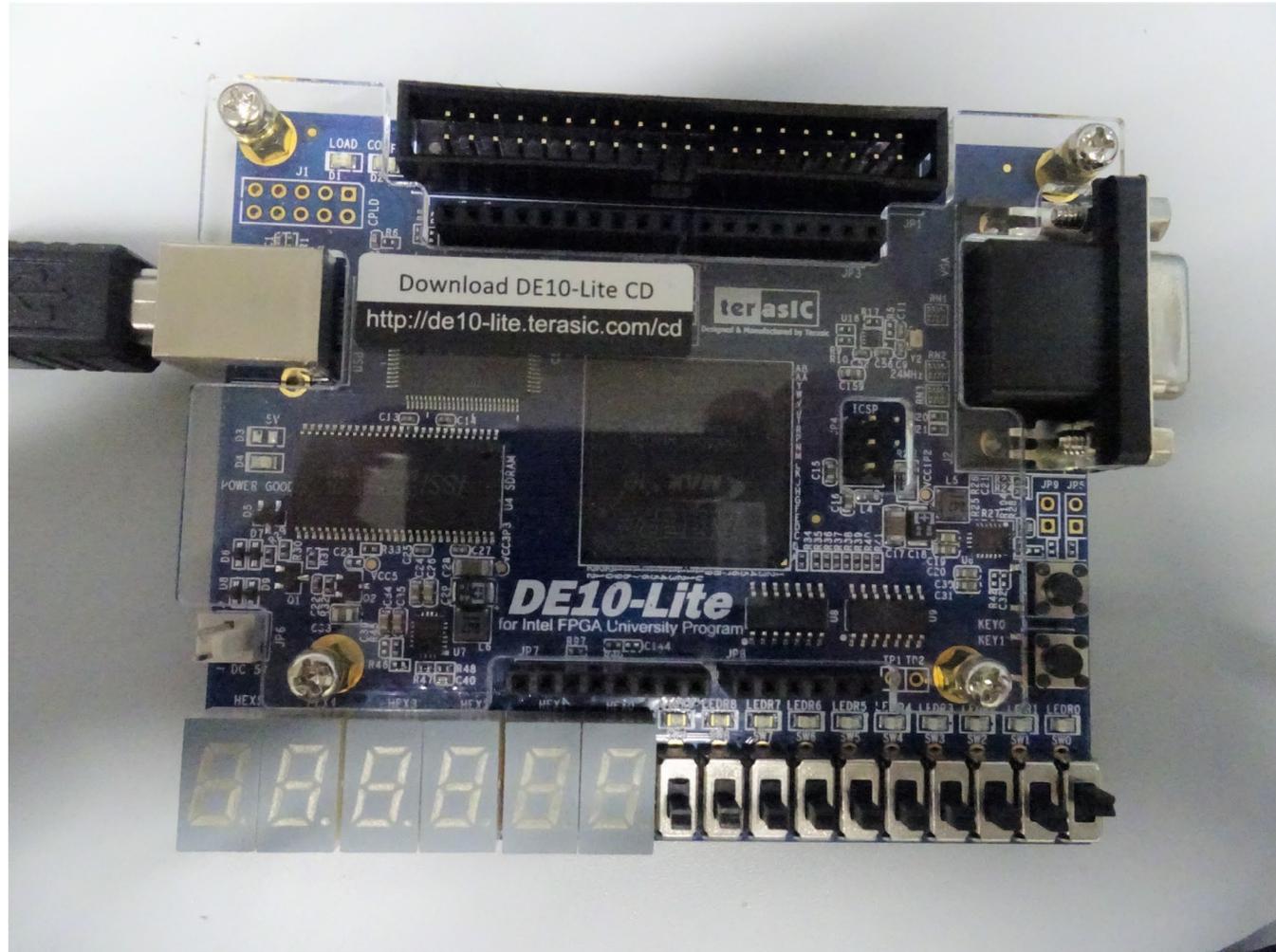
# FPGAの動作原理

- 各論理ブロックに共通の同期用クロックが提供される
- 各クロックの立ち上がり時、あるいは立ち下がり時に、論理ブロックは、入力にしたがって出力を変化させる
- また、論理ブロックは状態(記憶)を保持することができる
- 同期式順序回路が実現可能

# マイコン(microprocessor)との違い

- データ処理の単位
  - マイコン: オペランドのデータ長が固定(16ビット、32ビット)
  - FPGA: 半端なデータ長も可能(13ビットとか、21ビットとか)  
最適なデータ長での演算の設計が可能
- データ処理のやり方
  - マイコン: 基本的にはシーケンシャルな処理(プログラムカウンタで指された命令を順番に実行)
  - FPGA: (使用可能な論理回路がある限り)クロックに同期して並列に処理を実行可能
- 設計の難易度
  - マイコン: PC上でのプログラミングよりは制約が多いので難しいが、Cでプログラムが書けて相対的には容易、動作確認(検証)も相対的には容易
  - FPGA: 複数のプロセスが通信し合う形で動作する形で設計する必要性があり、本質的な並行処理が含まれるので動作確認(検証)が難しい

# FPGAの評価ボード



# FPGA上でのシステムの構築

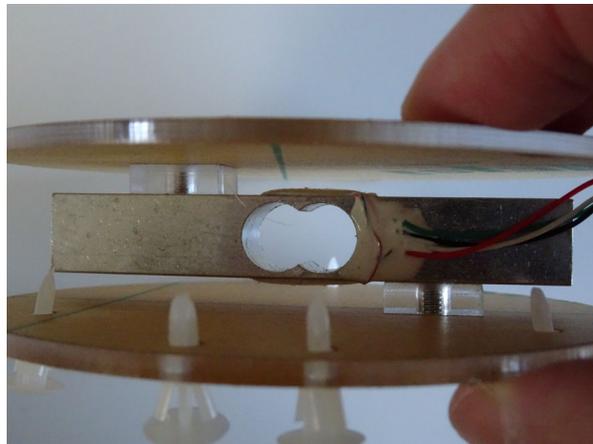
- 配線接続の情報(コンフィギュレーション)を一つ決めることが、プログラミングに相当
- プログラマは通常は、直接的にこのような情報を操作するではなく、ハードウェア記述言語(HDL)を用いて設計を行う
- 専用のソフトウェア(Intel Quartus Prime, Xilinx Vivadoなど)を用いて、パソコン上でHDLで書かれた仕様をコンフィギュレーションに変換し、FPGAにこれを書き込む

# FPGA開発ソフトウェア Quartus Prime Lite

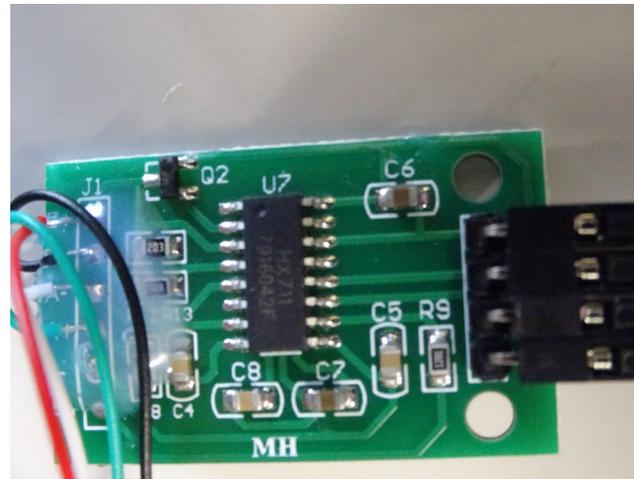
[動画再生はここをクリックしてください。](#)

# 簡単なはかりの製作(使用機材)

- FPGAボード(DE10-Lite) (1万6千円程度)
- Intel Quartus Prime Lite Edition (無料)
- ロードセル + A/D コンバータ (1千500円程度)



ロードセル: MAX 5kg



A/Dコンバータ HX711

# 簡単なはかりの製作(A/Dコンバータ HX711)

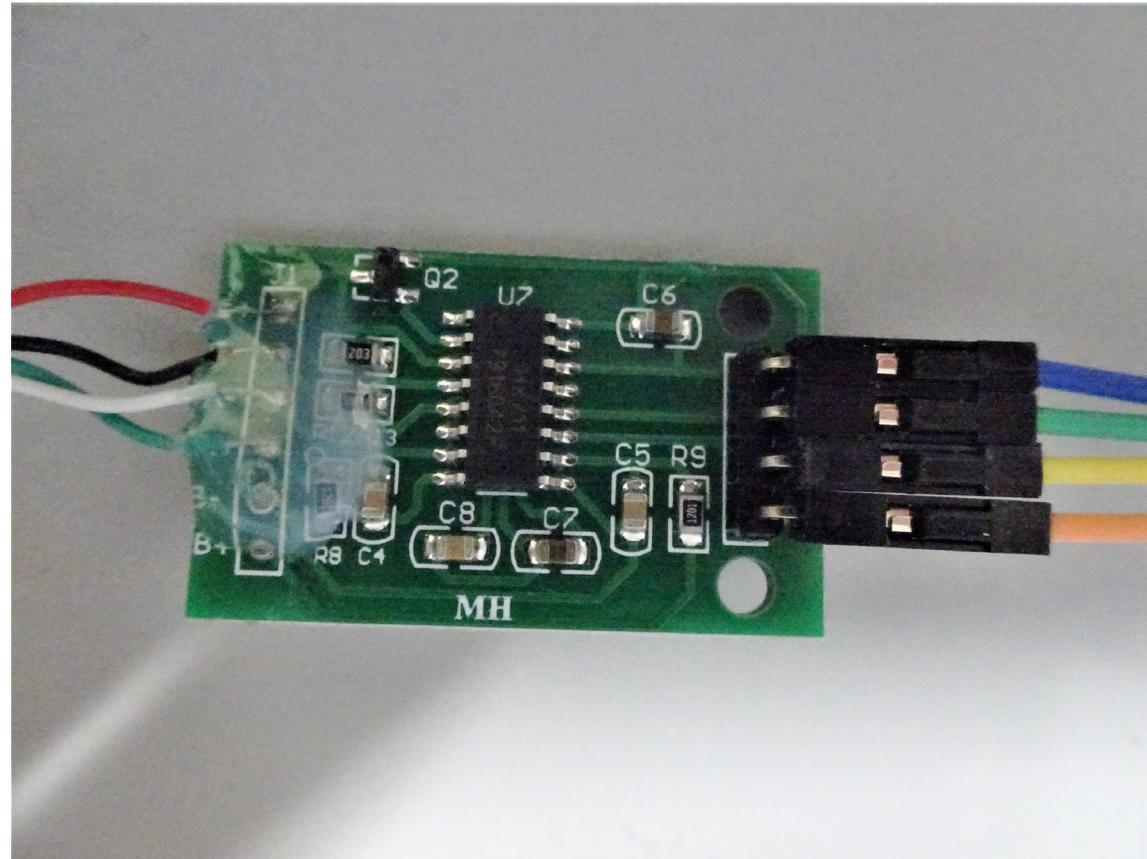
ロードセル側:

赤: VCC

黒: GND

白: 入力信号(マイナス)

緑: 入力信号(プラス)



FPGA側:

青: VCC

緑: データ出力

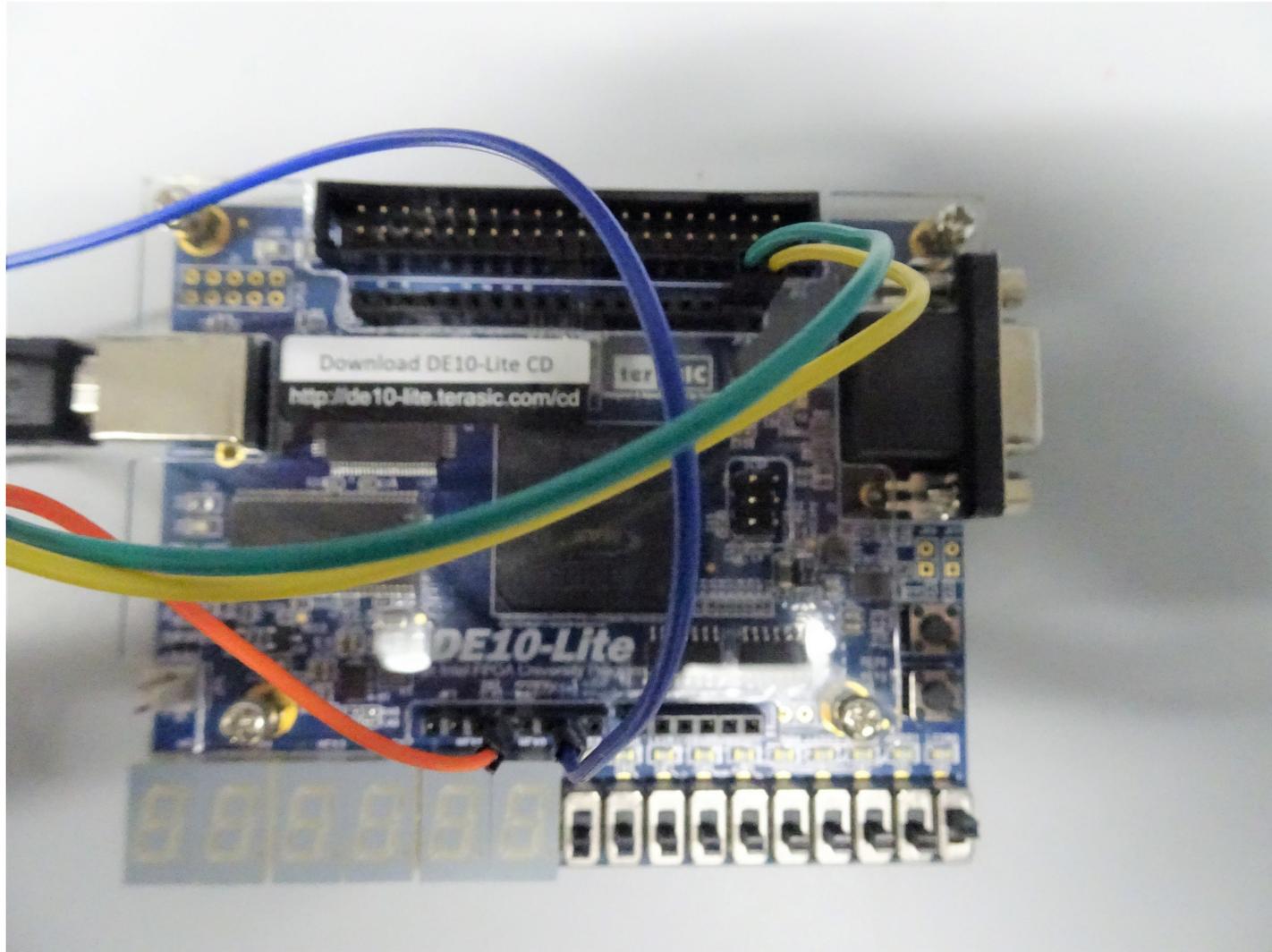
黄: クロック入力

橙: GND

ロードセルから得られる重量の情報を、FPGA側から受け取るクロックに合わせてFPGA側に送信する

電源はFPGAから受け取る

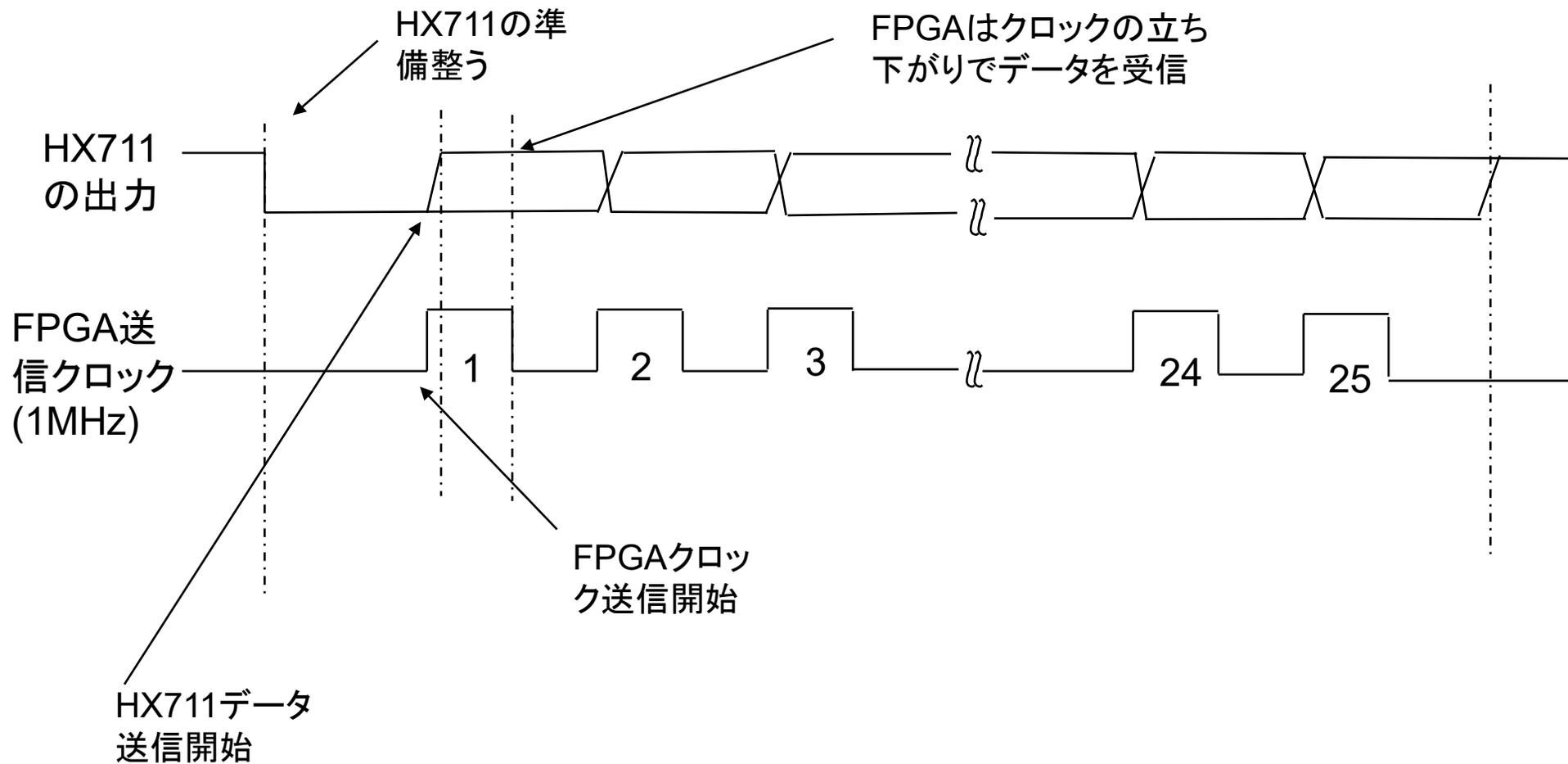
# 簡単なはかりの製作(コネクタの接続)



## 簡単なはかりの製作(重量表示部の基本設計)

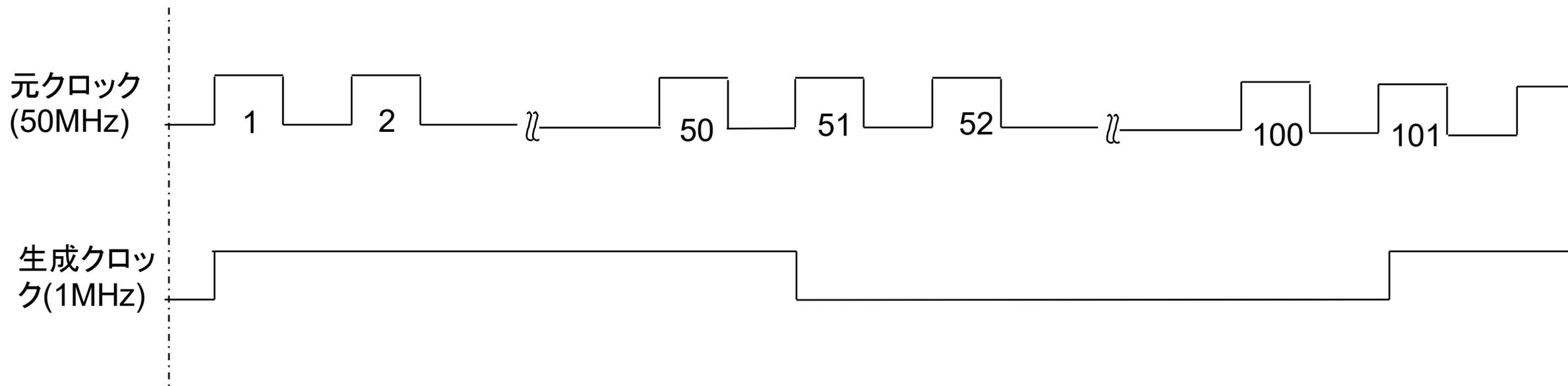
1. 初期化の段階でHX711側の準備が整うまでFPGA側は待つ  
(データ線がHigh→Lowになる)
2. その後で、FPGA側は1MHzのクロックを必要な回数(25回)分、HX711側に送信する
3. HX711側は送られてきたクロックに合わせて、重量信号を1ビットずつ送信する(最後にHighになる)
4. FPGA側は信号を受け取り、全部揃ったところで、得られたデータ値に定数を掛け合わせることで重量を得て、これを表示部に表示させる
5. 1.に戻る

# 簡単なはかりの製作(タイミングチャート)

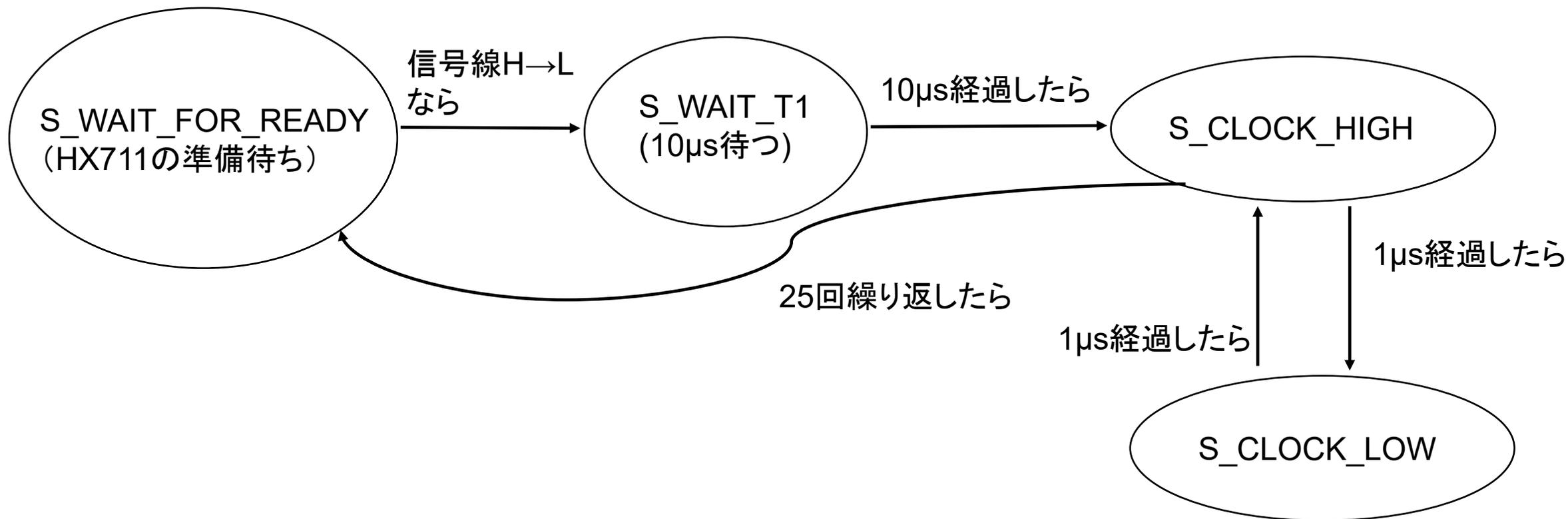


# 簡単なはかりの製作(1MHzのクロックの生成)

- 当該のFPGAは50MHzのクロックで動作
- 1MHzのクロックを作るには、FPGAのクロックの50回ごとに信号値を1→0あるいは0→1に反転させてやればよい



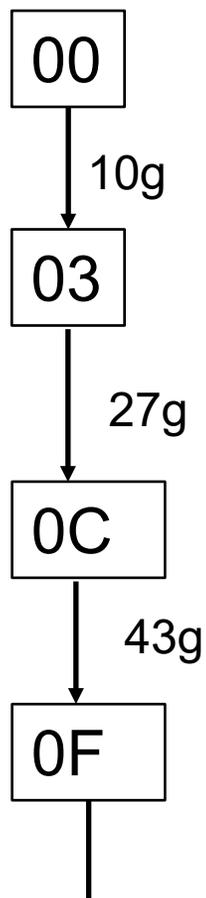
# 簡単なはかりの製作(状態遷移図の簡略版)



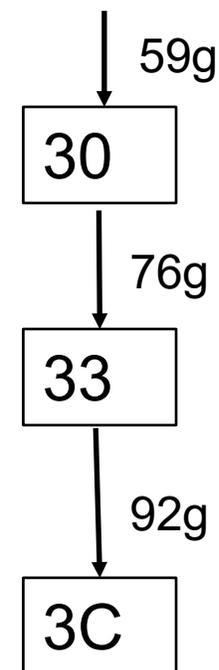
# 簡単なはかりの製作(Verilogコードの一部)

```
always @ (posedge clk) begin
  if (rst)
    begin
      next_state <= S_WAIT_FOR_READY;
    end
  else
    begin
      case (cur_state)
        S_WAIT_FOR_READY: // dout が立ち下がるまで待つ
          begin
            if (dout_nedge) // dout が立ち下がったら
              next_state <= S_WAIT_T1; // S_WAIT_T1 に移行
            else
              next_state <= S_WAIT_FOR_READY;
          end
      end
    end
  end
```

# 簡単なはかりの製作(生データの表示)



[動画再生はここをクリックしてください。](#)



1回のインクリメントが16.6g程度に対応

# 簡単なはかりの製作(生データの表示の注)

- 各桁の値

1.  $0_{16} = 0000_2$

2.  $3_{16} = 0011_2$

3.  $C_{16} = 1100_2$

4.  $F_{16} = 1111_2$

- 冗長性を取除くと、初期値の $3FF300_{16}$ は $11111010000_2$ を表す

- さらに最上位の桁は負の数を表しているので、実際の値は

$$1111010000_2 = 976_{10}$$

- 976を引き、16.6g を掛けると重量値が得られる

# 簡単なはかりの製作(重量値に変換後)

[動画再生はここをクリックしてください。](#)

# 簡単なはかりの製作(現状の課題)

- 初期値は個別のロードセルごとに異なるが、それぞれのために補正を行う機能がない
- HX711との通信に失敗したとき、リセットする機能がない

# 簡単なはかりの製作(FPGAの長所)

- タイミング制約を満足させることはマイコンよりはやりやすい
- マイコンだと、「△△ミリ秒待ち(wait命令)、その後で○○を行う」というようなコードを、プログラムファイルの中に散りばめる必要がある
- FPGAだと、「△△ミリ秒待った後の**状態**にいるときに、○○を行う」という形の記述になる

# FPGAと計量器ソフトウェア認証(組み込み)

- マイコンの代わりにFPGAを非自動はかりとして用いた場合に OIML R76-1に適合させるには
- 「組み込み」の場合には以下が必要
  1. バージョン番号の表示
  2. 封印を解かずにFPGAを更新できなくするような機械的な保護方法が別途必要

# FPGAと計量器ソフトウェア認証(PC)

- マイコンの代わりにFPGAを非自動はかりとして用いた場合にOIML R76-1に適合させるには
  - 「PC」の場合にはいくつか問題が生じる
1. ソフトウェア識別
    - マイコンの場合には法定計量に関連するプログラムの部分のチェックサムを取り、表示させる必要がある
    - 一方で、FPGAは回路と経路の情報しか持っていない
    - FPGAにロードする回路情報のファイルのチェックサムを取ることは可能

# FPGAと計量器ソフトウェア認証(PC)

## 2. ソフトウェア分離

- マイコンの場合には法定計量に関連するプログラムとそれ以外のプログラムに分離する
- 一方で、FPGA は回路と経路の情報しか持っていない
- 回路内で法定計量部分と非部分の切り分けは非常に難しいはず

# FPGAと計量器ソフトウェア認証(PC)

## 3. ソフトウェア・インターフェース

- FPGA は回路と経路の情報しか持っていない
- インターフェース部分の切り分けは難しい

# FPGAと計量器ソフトウェア認証

- ソフトウェア保護
  - FPGA は回路と経路の情報しか持っていない
  - 回路情報は逆アSEMBルしやすいので、OIML D76-1とは直接関係ないが、別途、暗号化などの形で、回路情報を保護する必要がある

# FPGAと計量器ソフトウェア認証

- 高度な品質保証: モデル検査(Model Checking)を用いる
- モデル検査: 時制論理式(temporal formula)を用いてシステム(状態遷移図)において成立している性質を検証する
- 時制論理式を用いて検証できる性質
  1. Safety: ある性質が常に成り立つ
  2. Liveness: 将来、必ずあるイベントが成立する
  3. インターフェースにおける非干渉性も検証可能?
- 性質が成り立っていないければ、反例を返す
- FPGA は回路と経路の情報しか持っていない
- マイコンと比較して、モデル検査を行った結果、保証される性質はFPGAでは、実機でも保証される可能性が高い → 高度な品質保証

# 他の話題

- 近似計算: 可能な場合は、必要最小限まで計算の精度を落として、消費電力を少なくする

# まとめ

- FPGAを用いて計量器を開発することは可能
- タイミングに依存するようなシステムはマイコンよりも開発しやすい
- 「組み込み」ではない形でFPGAを用いる場合は、議論の余地がある
- 高度な品質保証を与えることができる